UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 7,181,743 B2
APPLICATION NO. : 09/864825
DATED : February 20, 2007
INVENTOR(S) : Werme et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In the Claims

In Column 57, Line 2 (Claim 10), please delete “QS” and insert -- QoS --.

Signed and Sealed this
Eighteenth Day of February, 2020

Andrei Iancu
Director of the United States Patent and Trademark Office

PTO/SB/44 (09-07)
Approved for use through 01/31/2020. OMB 0651-0033
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.
(Also Form PTO-1050)

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,181,743

APPLICATION NO.: 09/864,825

ISSUE DATE : February 20, 2007

INVENTOR(S) : Werme et al.

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In column 57, line 2 (claim 10), please delete “QS” and insert -- QoS --.

MAILING ADDRESS OF SENDER (Please do not use Customer Number below):

KOWERT, HOOD, MUNYON, RANKIN & GOETZEL, P.C.
P.O. Box 398
Austin, Texas 78767-0398

This collection of information is required by 37 CFR 1.322, 1.323, and 1.324. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1.0 hour to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Attention Certificate of Corrections Branch, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor:	Werme et al.	Filing Date:	May 24, 2001
Serial Number:	09/864,825	Issued:	February 20, 2007
Patent No.:	7,181,743	Group/Art Unit:	2171
Title:	Resource allocation decision function for resource management architecture and corresponding programs therefor	Atty. Dkt. No:	6757-51203

TRANSMITTAL LETTER

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Transmitted herewith for filing in the captioned case is the following:

(1) Form PTO/SB/44. Errors which occur at important points in the captioned patent or which may otherwise affect the understanding or interpretation of the patent are thereon corrected.

All of the errors shown in PTO/SB/44 are due to Patent Office oversights. A Certificate of Correction is requested under 35 U.S.C. § 254. Applicants believe that no fees are required, however, should any fees be required, please deduct them from Kowert, Hood, Munyon, Rankin & Goetzel, P.C. Deposit Account No. 501505/6757-51203/DMM.

Respectfully submitted,

/Dean M. Munyon/
Dean M. Munyon
Reg. No. 42,914

Kowert, Hood, Munyon, Rankin & Goetzel, PC
P.O. Box 398
Austin, Texas 78767-0398
(512) 853-8800

Date: January 10, 2019

Electronic Acknowledgement Receipt	
EFS ID:	38259174
Application Number:	09864825
International Application Number:	
Confirmation Number:	2372
Title of Invention:	RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR
First Named Inventor/Applicant Name:	Paul V. Werme
Customer Number:	23501
Filer:	Dean M. Munyon/Danielle Kramer
Filer Authorized By:	Dean M. Munyon
Attorney Docket Number:	NC 83017
Receipt Date:	10-JAN-2020
Filing Date:	24-MAY-2001
Time Stamp:	10:24:33
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Request for Certificate of Correction	Certificate_of_Correction_PTO SB44.pdf	165100 d1335cca4ac64ee41483981fbc8399add25ec2c9	no	2
Warnings:					

Information:					
2	Transmittal Letter	Cert_Corr_Transltr.pdf	100617 827da6c0394076539a47e19f14fa79c97c689184	no	1
Warnings:					
Information:					
			Total Files Size (in bytes):	265717	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.
National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.
New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
09/864,825	05/24/2001	Paul V. Werme	NC 83017

CONFIRMATION NO. 2372

23501
NAVAL SURFACE WARFARE CENTER DAHLGREN DIVISION
OFFICE OF COUNSEL, CODE XDC1
17632 DAHLGREN ROAD
SUITE 121
DAHLGREN, VA 22448-5110

Date Mailed: 05/03/2007

DENIAL OF REQUEST FOR WITHDRAWAL OF ATTORNEY OR AGENT

The request for withdrawal of attorney or agent filed __04/02/07_____ is acknowledged. However, the request cannot be granted at this time for the reason stated below.

☐ The request is untimely because there is an outstanding Office Action in the application to which a response is due within 30 days.

☐ The request is not being made by the attorney or agent who wishes to be withdrawn, or the attorney or agent signing the request has not stated that he/she is signing on behalf of himself/herself and the withdrawing attorney or agent.

☒ The person signing the request did not provide his/her name and registration number.

☐ There is no alternate correspondence address, either in the application file or the request. Please provide a correspondence address to which further correspondence can be directed.

☐ No Power of Attorney has been established in the above-identified application. Accordingly, the withdrawal cannot be accepted because the practitioner seeking to withdraw is not of record. The change of correspondence address will be considered separately.

☒ The attorney or agent requesting to withdraw has not been appointed attorney of record in the application (i.e., there is no power of attorney). Accordingly, the withdrawal cannot be accepted because the practitioner seeking to withdraw is not of record. Moreover, a patent practitioner cannot withdraw another practitioner of record unless they also withdraw himself or herself, and affirm that they are acting on behalf of the other patent practitioner(s). To revoke the power of attorney given to another patent practitioner, the applicant must comply with 37 CFR 1.36(a).

☐ The Request for Withdrawal of Attorney or Agent does not properly identify the attorney(s)/agent(s) to be withdrawn.

Questions relating to this Notice should be directed to the Office of Initial Patent Examination, Customer Service at the number below.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Customer Service Center
Office of Initial Patent Examination
(571) 272-4000

OIPE IAP44 APR 02 2007 PATENT & TRADEMARK OFFICE

IFW

Please type a plus sign (+) inside this box → ☐

PTO/SB/83 (2/00)
Approved for use through 10/31/02. OMB 0651-0035
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

REQUEST FOR WITHDRAWAL AS ATTORNEY OR AGENT

Application Number	09/864825
Filing Date	
First Named Inventor	Fontenot
Group Art Unit	
Examiner Name	
Attorney Docket Number	83017

To: Assistant Commissioner for Patents
Washington, DC 20231

I hereby apply to withdraw as attorney or agent for the above identified patent application.

The reasons for this request are:

Mandatory withdrawal due to my employment by the United States Patent and Trademark Office beginning April 1, 2007.

1. ☑ The correspondence address is NOT affected by this withdrawal.
2. ☐ Change the correspondence address and direct all future correspondence to:

CORRESPONDENCE ADDRESS

☐ Customer Number [ ] →

OR

Place Customer Number Bar Code Label here

☐ Firm *or* Individual Name					
Address					
Address					
City		**State**		**ZIP**	
Country					
Telephone		**Fax**			

This request is enclosed in triplicate.

Name	Scott R. Boalick
Signature	
Date	March 30, 2007

NOTE: Withdrawal is effective when approved rather than when received.
Unless there are at least 30 days between approval of withdrawal and the expiration date of a time period for response or possible extension period, the request to withdraw is normally disapproved.

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/864,825	02/20/2007	7181743	NC 83017	2372

23501 7590 01/31/2007

NAVAL SURFACE WARFARE CENTER, DAHLGREN DIVISION
OFFICE OF COUNSEL, CODE XDC1
17320 DAHLGREN ROAD
DAHLGREN, VA 22448-5110

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment is 742 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site http://pair.uspto.gov for additional applicants):

Paul V. Werme, King George, VA;
Michael W. Masters, Fredericksburg, VA;
Larry A. Fontenot, King George, VA;
Lonnie R. Welch, Athens, OH;

IR103 (Rev. 11/05)

OIPE
DEC 18 2006
PATENT & TRADEMARK OFFICE

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

23501 7590 11/02/2006

NAVAL SURFACE WARFARE CENTER, DAHLGREN DIVISION
OFFICE OF COUNSEL, CODE XDC1
17320 DAHLGREN ROAD
DAHLGREN, VA 22448-5110

12/18/2006 EAYALEW2 00000074 500967 09864825

01 FC:1501 1400.00 DA
02 FC:1504 300.00 DA

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

DEBORAH G. BLAKLEY	(Depositor's name)
Deborah G. Blakley	(Signature)
11/20/06	(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/864,825	05/24/2001	Paul V. Werme	NC 83017	2372

TITLE OF INVENTION: RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$300	$0	$1700	02/02/2007

EXAMINER	ART UNIT	CLASS-SUBCLASS
TANG, KENNETH	2195	718-104000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 GERHARD W. THIELMAN, ESQ.
2 SCOTT R. BOALICK, ESQ.
3 JAMES B. BECHTEL, ESQ.

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE
THE UNITED STATES OF AMERICA
AS REPRESENTED BY THE SECRETARY
OF THE NAVY

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☐ Corporation or other private group entity XX Government

4a. The following fee(s) are submitted:
XX Issue Fee
XX Publication Fee (No small entity discount permitted)
XX Advance Order - # of Copies 10

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
☐ A check is enclosed.
☐ Payment by credit card. Form PTO-2038 is attached.
☒ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number 50-0967 (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____ Date November 20, 2006

Typed or printed name GERHARD W. THIELMAN Registration No. 43,186

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

OIPE MAP21 DEC 18 2006 PATENT & TRADEMARK OFFICE

PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission	3

Application Number	09/864,825
Filing Date	May 24, 2001
First Named Inventor	Paul V. Werme et al.
Art Unit	2195
Examiner Name	Kenneth Tang
Attorney Docket Number	Navy Case 83017

ENCLOSURES *(Check all that apply)*

- [x] Fee Transmittal Form
  - [x] Fee Attached
- [ ] Amendment/Reply
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Reply to Missing Parts/ Incomplete Application
  - [ ] Reply to Missing Parts under 37 CFR 1.52 or 1.53
- [x] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
  - [ ] Landscape Table on CD
- [ ] After Allowance Communication to TC
- [ ] Appeal Communication to Board of Appeals and Interferences
- [ ] Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- [ ] Proprietary Information
- [ ] Status Letter
- [ ] Other Enclosure(s) (please Identify below):

Remarks

Replacement of Fig 2A as per draftsperson's instructions on PTO-948
Issue Fee/Publ. Fee

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Naval Surface Warfare Center - Dahlgren Division		
Signature			
Printed name	Gerhard W. Thielman		
Date	November 20, 2006	Reg. No.	43,186

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature	Deborah G. Blakley		
Typed or printed name	Deborah G. Blakley	Date	11/20/06

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

OIPE
DEC 18 2006
PATENT & TRADEMARK OFFICE

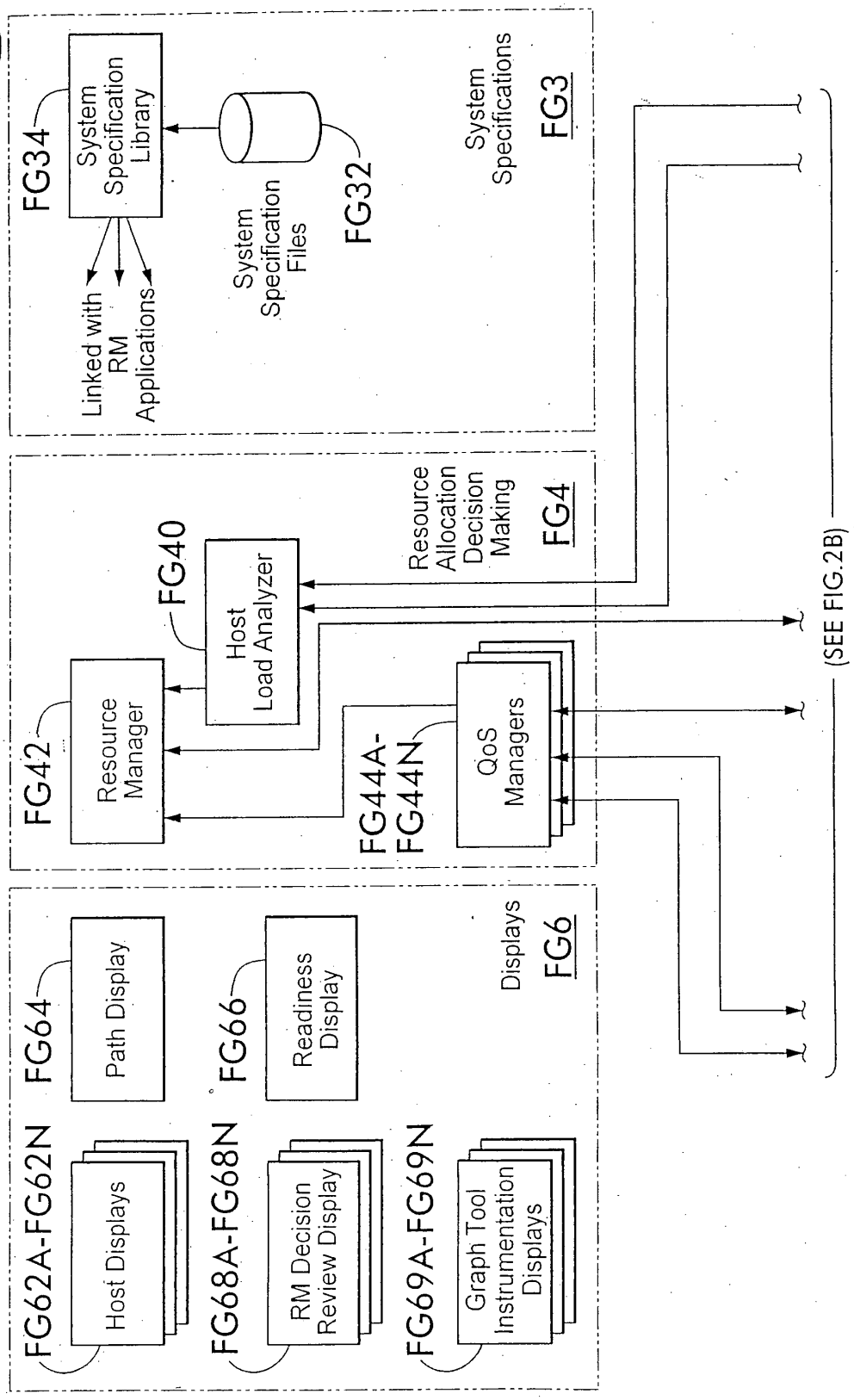


FIG.2A

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

23501 7590 11/02/2006

NAVAL SURFACE WARFARE CENTER, DAHLGREN DIVISION
OFFICE OF COUNSEL, CODE XDC1
17320 DAHLGREN ROAD
DAHLGREN, VA 22448-5110

EXAMINER	
TANG, KENNETH	
ART UNIT	PAPER NUMBER
2195	

DATE MAILED: 11/02/2006

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/864,825	05/24/2001	Paul V. Werme	NC 83017	2372

TITLE OF INVENTION: RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$300	$0	$1700	02/02/2007

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PTOL-85 (Rev. 07/06) Approved for use through 04/30/2007.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

23501 7590 11/02/2006

NAVAL SURFACE WARFARE CENTER, DAHLGREN DIVISION
OFFICE OF COUNSEL, CODE XDC1
17320 DAHLGREN ROAD
DAHLGREN, VA 22448-5110

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)

(Signature)

(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/864,825	05/24/2001	Paul V. Werme	NC 83017	2372

TITLE OF INVENTION: RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$300	$0	$1700	02/02/2007

EXAMINER	ART UNIT	CLASS-SUBCLASS
TANG, KENNETH	2195	718-104000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) the names of up to 3 registered patent attorneys or agents OR, alternatively, 1_____

(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed. 2_____ 3_____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:

☐ Issue Fee

☐ Publication Fee (No small entity discount permitted)

☐ Advance Order - # of Copies _____

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**

☐ A check is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☐ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)

☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.

☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/864,825	05/24/2001	Paul V. Werme	NC 83017	2372

23501 7590 11/02/2006

NAVAL SURFACE WARFARE CENTER, DAHLGREN DIVISION
OFFICE OF COUNSEL, CODE XDC1
17320 DAHLGREN ROAD
DAHLGREN, VA 22448-5110

EXAMINER	
TANG, KENNETH	
ART UNIT	PAPER NUMBER
2195	

DATE MAILED: 11/02/2006

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 807 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 807 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

PTOL-85 (Rev. 07/06) Approved for use through 04/30/2007.

Notice of Allowability	Application No.	Applicant(s)	
	09/864,825	WERME ET AL.	
	Examiner	Art Unit	
	Kenneth Tang	2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *10/17/06*.

2. ☒ The allowed claim(s) is/are *16-20 and 22-36; now renumbered as 1-20*.

3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☒ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☒ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☒ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☒ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☐ Examiner's Amendment/Comment
8. ☐ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Form PTO-948 (Rev. 06/03)
Application No. 09/864825

U.S. DEPARTMENT OF COMMERCE
U.S. Patent and Trademark Office

NOTICE OF DRAFTSPERSON'S PATENT DRAWING REVIEW

The drawing(s) filed (insert date) 09/19/2001 & 04/26/2006 are:

A. _____ approved by the Draftsperson under 37 CFR 1.84 or 1.152.

B. ✓ objected to by the Draftsperson under 37 CFR 1.84 or 1.152 for the reasons indicated below. Corrected drawings are required.

1. DRAWINGS. 37 CFR 1.84(a): Acceptable categories of drawings: Black ink or Color (3 sets required).

___ Color drawings are not acceptable until petition is granted. Fig(s)_____

___ Pencil and non black ink not permitted. Fig(s)____

2. PHOTOGRAPHS. 37 CFR 1.84(b)

___ One (1) full-tone set is required. Fig(s)_____

___ Photographs may not be mounted. 37 CFR 1.84(e)

___ Photographs must meet paper size requirements of 37 CFR 1.84(f). Fig(s)_____

___ Poor quality (half-tone). Fig(s)_____

3. TYPE OF PAPER. 37 CFR 1.84(e)

___ Paper not flexible, strong, white, and durable. Fig(s)_____

✓ Erasures, alterations, overwritings. interlineations, folds, copy machine marks not accepted. Fig(s) *2A (Copy marks)*

4. SIZE OF PAPER. 37 CFR 1.84(f): Acceptable sizes:

21.0 cm by 29.7 cm (DIN size A4) or
21.6 cm by 27.9 cm (8 1/2x 11 inches)

___ All drawing sheets not the same size. Sheet(s)_____

___ Drawings sheets not an acceptable size. Fig(s)____

5. MARGINS. 37 CFR 1.84(g): Acceptable margins:

Top 2.5 cm Left 2.5 cm Right 1.5 cm Bottom 1.0 cm

___ Margins not acceptable. Fig(s)_____

_____Top (T) _____ Left (L)

_____Right (R) _____ Bottom (B)

6. VIEWS. 37 CFR 1.84(h)

REMINDER: Specification may require revision to correspond to drawing changes, e.g., if Fig. 1 is changed to Fig. 1A, Fig 1B and Fig. 1C, etc., the specification, at the Brief Description of the Drawings, must likewise be changed.

___Views not labeled separately or properly. Fig(s)_____

7. SECTIONAL VIEWS. 37 CFR 1.84(h)(3)

____ Sectional designation should be noted with Arabic or Roman numbers. Fig(s)_____

8. ARRANGEMENT OF VIEWS. 37 CFR 1.84(i)

___ Words do not appear on a horizontal, left-to-right fashion when page is either upright or turned so that the top becomes the right side, except for graphs. Fig(s)_____

9. SCALE. 37 CFR 1.84(k)

___ Scale not large enough to show mechanism without crowding when drawing is reduced in size to two-thirds in reproduction. Fig(s)_____

10. CHARACTER OF LINES, NUMBERS, & LETTERS. 37 CFR 1.84(l)

___ Lines, numbers & letters not uniformly thick and well defined, clean, durable, and black (poor line quality). Fig(s)_____

11. SHADING. 37 CFR 1.84(m)

___ Solid black areas pale. Fig(s) _____

___ Solid black shading not permitted. Fig(s)_____

12. NUMBERS, LETTERS, & REFERENCE CHARACTERS. 37 CFR 1.84(p)

___ Numbers and reference characters not plain and legible. Fig(s)_____

___ Figure legends are poor. Fig(s)_____

___ Numbers and reference characters not oriented in the same direction as the view. 37 CFR 1.84(p)(1) Fig(s)_____

___ English alphabet not used. 37 CFR 1.84(p)(2) Fig(s)_____

___ Numbers, letters and reference characters must be at least 32 cm (1/8 inch) in height. 37 CFR 1.84(p)(3). Fig(s)_____

13. LEAD LINES. 37 CFR 1.84(q)

___ Lead lines missing. Fig(s)_____

14. NUMBERING OF SHEETS OF DRAWINGS. 37 CFR 1.84(t)

___ Sheets not numbered consecutively, and in Arabic numbers beginning with number 1. Sheet(s)_____

15. NUMBERING OF VIEWS. 37 CFR 1.84(u)

___ Views not numbered consecutively, and in Arabic numerals, beginning with number 1. Fig(s)_____

16. DESIGN DRAWINGS. 37 CFR 1.152

___ Surface shading shown not appropriate. Fig(s)_____

___ Solid black surface shading is not permitted except when used to represent the color black as well as color contrast. Fig(s)_____

COMMENTS:

Reviewer SON LAM

Date 10/24/2006

If you have questions, call (703) ~~305-8404~~ 305-0333 (EXT. 133)

Attachment to Paper No. _____

☆ U.S. GOVERNMENT PRINTING OFFICE: 2003-300-153

RECEIVED
CENTRAL FAX CENTER
OCT 17 2006

PATENT APPLICATION

RESPONSE UNDER 37 CFR §1.116
TECHNOLOGY CENTER ART UNIT 2195

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants :	Paul V. Werme *et al.*	Group Art Unit :	2195
Serial No. :	09/864,825	Examiner :	Kenneth Tang
Filed :	May 24, 2001	Docket :	Navy Case 83017
Title :	RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR		

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT AFTER FINAL REJECTION UNDER 37 CFR §1.116

In response to the August 23, 2006 Final Office Action, the shortened statutory period for reply effectively ending November 24, 2006, please consider the following:

OK to enter
10/23/06
KT

Amendments to the Claims are reflected in the listing of claims at pages 2-7; and

Remarks at pages 8-9.

CERTIFICATE OF TRANSMISSION / MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, Mail Stop: Amendment, P.O. Box 1450, Alexandria VA 22313-1450 on the date identified below:

Date: 10/17/2006 Name: Laurie A. Muir
Laurie A. Muir

Issue Classification	Application/Control No. 09/864,825	Applicant(s)/Patent under Reexamination WERME ET AL.
	Examiner Kenneth Tang	Art Unit 2195

ISSUE CLASSIFICATION

ORIGINAL		INTERNATIONAL CLASSIFICATION									
CLASS	SUBCLASS	CLAIMED					NON-CLAIMED				
718	104	G	06	F	11	/00				/	
CROSS REFERENCES		G	06	F	15	/16				/	
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)	G	08	B	5	/22				/	
709	219	G	06	F	9	/46				/	
340	7.2					/				/	
714	2 25					/				/	
						/				/	
						/				/	

Kenneth Tang 10/24/06
(Assistant Examiner) (Date)

Sharon Blizzard 10/30/06
(Legal Instruments Examiner) (Date)

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100
(Primary Examiner) (Date) 10/24/06

Total Claims Allowed: 20

O.G. Print Claim(s)	O.G. Print Fig.
1	1

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
	1	15	31		61		91		121		151		181
	2	16	32		62		92		122		152		182
	3	17	33		63		93		123		153		183
	4	18	34		64		94		124		154		184
	5	19	35		65		95		125		155		185
	6	20	36		66		96		126		156		186
	7		37		67		97		127		157		187
	8		38		68		98		128		158		188
	9		39		69		99		129		159		189
	10		40		70		100		130		160		190
	11		41		71		101		131		161		191
	12		42		72		102		132		162		192
	13		43		73		103		133		163		193
	14		44		74		104		134		164		194
	15		45		75		105		135		165		195
1	16		46		76		106		136		166		196
2	17		47		77		107		137		167		197
3	18		48		78		108		138		168		198
4	19		49		79		109		139		169		199
5	20		50		80		110		140		170		200
	21		51		81		111		141		171		201
6	22		52		82		112		142		172		202
7	23		53		83		113		143		173		203
8	24		54		84		114		144		174		204
9	25		55		85		115		145		175		205
10	26		56		86		116		146		176		206
11	27		57		87		117		147		177		207
12	28		58		88		118		148		178		208
13	29		59		89		119		149		179		209
14	30		60		90		120		150		180		210

Search Notes	Application No.	Applicant(s)	
	09/864,825	WERME ET AL.	
	Examiner	Art Unit	
	Kenneth Tang	2127	

SEARCHED			
Class	Subclass	Date	Examiner
718	100-108	6/27/2004	KT
705	8	6/27/2004	KT
370	217	6/27/2004	KT
719	311	10/23/2006	KT
709	219	10/23/2006	KT
340	7.2	10/23/2006	KT
714	2, 25	10/26/2006	KT
718	104	10/26/2006	KT

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner
718	**104**	10/26/2006	**KT**
709	219	10/23/2006	KT
340	7.2	10/23/2006	KT
714, 2, 25 PG Pub search		10/23/2006	KT

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
EAST	6/27/2004	KT
EAST	2/15/2005	KT
EAST, text search	1/22/2006	KT
EAST, text search	8/15/2006	KT
EAST, updated search	10/23/2006	KT
Inventor search	10/23/2006	KT

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	942	714/2.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/10/26 11:30
L2	1843	714/25.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/10/26 11:30
L3	31	1 and 2	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/10/26 11:30

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	76	(((characteristic attribute) near3 (program application))) and (user with (fault adj toleran$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/23 13:39
L2	65	@ad<="20010524" and L1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/23 13:39
L3	811	(((characteristic attribute) with (program application))) with priority	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/23 13:39
L4	4	L2 and L3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/23 13:39
L5	11	(@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))) and (resource adj manag$5) and network$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/23 13:39
L6	1	(US-5826239-$).did.	USPAT	OR	OFF	2006/10/23 13:39
L7	0	L6 and (weigh$4 priority rank$4)	US-PGPUB; USPAT	OR	ON	2006/10/23 13:39
L8	79	(scalable adj application) and @ad<="20010524"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/23 13:39
L9	0	L6 and (weigh$4 priority rank$4)	US-PGPUB; USPAT	OR	ON	2006/10/23 13:42
L10	155	719/311.ccls.	US-PGPUB; USPAT	OR	ON	2006/10/23 13:42
L11	4056	709/219.ccls.	US-PGPUB; USPAT	OR	ON	2006/10/23 13:42
L12	402614	"340"/("7.2").ccls.	US-PGPUB; USPAT	OR	ON	2006/10/23 13:42
L13	946	(werme masters fontenot).inv.	US-PGPUB; USPAT	OR	OFF	2006/10/23 13:43
L14	1677	qos.clm.	US-PGPUB; USPAT	OR	OFF	2006/10/23 13:44

EAST Search History

L15	0	13 and 14	US-PGPUB; USPAT	OR	OFF	2006/10/23 13:44
L16	780	@ad<="20010524" and (scalable and (resource adj manag$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/23 13:44
L17	1844	scalable and (resource adj manag$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/23 13:44
L18	1844	scalable and (resource adj manag$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/23 13:44

RECEIVED
CENTRAL FAX CENTER
OCT 17 2006

FAX TRANSMITTAL

Date: 10/17/2006

Number of Pages (including cover sheet): 11

From: Laurie Muir
Office of Counsel, Code XDC1
Naval Surface Warfare Center
Dahlgren Division
17320 Dahlgren Road
Dahlgren, VA 22448-5110
Phone: (540) 653-7578
FAX: (540) 653-8879

To: EXAMINER: Kenneth Tang
FAX: 571-273-8300

Certificate of Transmission under 37 CFR 1.8
I hereby certify that this correspondence is being facsimile transmitted to the United States Patent and Trademark Office (Fax No. (571)273-8300) on Tuesday, October 17, 2006.

Laurie A. Muir

Remarks:

In re application of	:	
Paul V. Werme et al	:	CONFIRMATION NO. 2372
Attorney Docket No. 83017	:	
Serial No. 09/864,825	:	Art Unit: 2195
Filed: May 24, 2001	:	
For: RESOURCE ALLOCATION DECISION	:	Examiner: Kenneth Tang
FUNCTION FOR RESOURCE MANAGEMENT	:	
ARCHITECTURE AND CORRESPONDING	:	
PROGRAMS THEREFO	:	

Faxed herewith please find response to Final Office Action.

PAGE 1/11 * RCVD AT 10/17/2006 10:38:51 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/9 * DNIS:2738300 * CSID:5406538879 * DURATION (mm-ss):03-20

RECEIVED
CENTRAL FAX CENTER

OCT 17 2006

PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission: 10

Application Number	09/884,825
Filing Date	May 24, 2001
First Named Inventor	Paul V. Werme et al.
Art Unit	2195
Examiner Name	Kenneth Tang
Attorney Docket Number	Navy Case 83017

ENCLOSURES *(Check all that apply)*

- ☐ Fee Transmittal Form
  - ☐ Fee Attached
- ☑ Amendment/Reply
  - ☑ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Reply to Missing Parts/ Incomplete Application
  - ☐ Reply to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) _____
  - ☐ Landscape Table on CD
- ☐ After Allowance Communication to TC
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to TC (Appeal Notice, Brief, Reply Brief)
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Other Enclosure(s) (please identify below):

Remarks

Response to August 23, 2006 Final Office Action

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Naval Surface Warfare Center - Dahlgren Division		
Signature			
Printed name	Gerhard W. Thielman		
Date	October 17, 2006	Reg. No.	43,186

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature	Laurie A. Muir		
Typed or printed name	Laurie A. Muir	Date	October 17, 2006

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: **Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

RECEIVED
CENTRAL FAX CENTER

OCT 17 2006

PATENT APPLICATION

RESPONSE UNDER 37 CFR §1.116
TECHNOLOGY CENTER ART UNIT 2195

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants :	Paul V. Werme *et al.*	Group Art Unit :	2195
Serial No. :	09/864,825	Examiner :	Kenneth Tang
Filed :	May 24, 2001	Docket :	Navy Case 83017
Title :	RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR		

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT AFTER FINAL REJECTION UNDER 37 CFR §1.116

In response to the August 23, 2006 Final Office Action, the shortened statutory period for reply effectively ending November 24, 2006, please consider the following:

Amendments to the Claims are reflected in the listing of claims at pages 2-7; and

Remarks at pages 8-9.

CERTIFICATE OF TRANSMISSION/ MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, Mail Stop: Amendment, P.O. Box 1450, Alexandria VA 22313-1450 on the date identified below:

Date: 10/17/2006 Name: Laurie A. Muir

Laurie A. Muir

PAGE 3/11 * RCVD AT 10/17/2006 10:38:51 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/9 * DNIS:2738300 * CSID:5406538879 * DURATION (mm-ss):03-20

Amendments to the Claims:

The following listing of claims will replace all prior versions, and listings, of claims in the application:

1-15. (Cancelled)

16. (Previously Presented) In a distributed environment comprised of N hosts operating in a distributed environment instantiating M managed characteristic application computer programs managed by the N hosts, resource allocation control software instantiated by at least the N hosts, M and N each being a positive integer, where M may be equal to, less than, or greater than N, the software comprising:

a first function which determines a state and health of the N hosts, a network operatively coupling the N hosts to one another, and the M managed characteristic application computer programs in the distributed environment;

a second function which determines required allocation and reallocation actions needed to maintain a plurality of Quality of Service (QoS) requirements established for the M managed characteristic application computer programs, the QoS requirements dictating parameters regarding service quality of the M managed characteristic application computer programs; and

a third function which generates automatic control signal requests corresponding to the actions dictated by the QoS requirements, such that the managed characteristic application computer programs are moved, shutdown, and started in accordance with satisfaction of the QoS requirements, wherein the second function determines the required allocation and reallocation actions needed to maintain the Quality of Service (QoS) requirements established for the M managed characteristic applications by:

responding to application and host failures by determining if and what recovery actions should be taken;

determining if and where to place new copies of one of the M managed characteristic application computer programs or which of the M managed characteristic application computer programs should be shutdown when QoS Manager functions indicate that scale up or scale down

actions are indicated based on measured application performance and application QoS specifications;

determining where new application computer programs should be placed when requested to do so by a program control device; and

determining which and how many application computer programs should run based on application system priorities.

17. (Original) The software as recited in claim 16, wherein the first function receives system specification information comprising selected ones of host configuration and capabilities, application capabilities, survivability requirements, scalability characteristics, application startup and shutdown dependencies, and application and path performance requirements.

18. (Previously Presented) The software as recited in claim 16, wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic application computer programs, and detected failures regarding the N hosts.

19. (Previously Presented) The software as recited in claim 16, wherein the first function receives application performance data representing each of the M managed characteristic application computer programs.

20. (Previously Presented) The software as recited in claim 16, wherein the first function receives application performance data on one or more applications instantiated by the N hosts including performance data representing each of the M managed characteristic application computer programs.

21. (Cancelled)

22. (Previously Presented) The software as recited in claim 16, wherein at least one of the M managed characteristic application computer programs comprises a scalable application computer program.

23. (Previously Presented) The software as recited in claim 16, wherein at least one of the M managed characteristic application computer programs comprises a fault tolerant application computer program, where the degree of fault tolerance is selectable by a user.

24. (Previously Presented) The software as recited in claim 16, wherein one of the M managed characteristic application computer programs comprises a selectable priority application computer program.

25. (Previously Presented) The software as recited in claim 16, wherein the M managed characteristic application computer programs comprise M copies of a single managed characteristic application computer program.

26. (Previously Presented) Software stored on at least one host for converting N networked hosts into a resource managed system instantiating M managed characteristic application computer programs, each managed characteristic application computer program managed by one of the N networked hosts, the software comprising:

a first function group which monitors the host and network resources;

a second function group which provides application computer program event reporting and event correlation capabilities;

a third function group which provides reasoning and decision making capabilities for the resource managed system, wherein the third function group comprises:

a first function which determines ~~the~~ a state and health of the N hosts, a network operatively coupling the N hosts to one another and the M managed characteristic application computer programs in the distributed environment;

a second function which determines required allocation and reallocation actions needed to maintain a plurality of Quality of Service (QoS) requirements established for the M managed characteristic application computer programs, the QoS requirements dictating parameters regarding service quality of the M management characteristic application programs; and

a third function which generates automatic control signal requests corresponding to the actions dictated by the QoS requirements, such that the managed characteristic application computer programs are moved, shutdown, and started in accordance with satisfaction of the QoS requirements; and

a fourth function group which provides program control capabilities permitting starting, stopping, and configuring of selected ones of the M managed characteristic application computer programs on respective ones of the N hosts in the resource managed system,

where M and N are positive integers and where M may be equal to, greater than, or less than N.

27. (Previously Presented) The software as recited in claim 26, wherein the first function receives system specification information comprising host configuration and capabilities.

28. (Previously Presented) The software as recited in claim 26, wherein the first function receives system specification information comprising selected ones of capabilities, survivability requirements, scalability characteristics, startup and shutdown dependencies, and performance requirements for at least one of the M managed characteristic application computer programs.

29. (Original) The software as recited in claim 26, wherein the first function receives system specification information comprising path performance requirements regarding communication between at least two of the N hosts.

30. (Previously Presented) The software as recited in claim 26, wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic application computer programs, and detected failures regarding the N hosts.

31. (Original) The software as recited in claim 26, wherein the first function receives historical data regarding statuses, configuration, and loads of the N hosts and link statuses and loads regarding the network.

32. (Previously Presented) The software as recited in claim 26, wherein the first function receives application performance data representing each one of the M managed characteristic application computer programs.

33. (Previously Presented) The software as recited in claim 26, wherein the first function receives application performance data on one or more applications instantiated by the N hosts including performance data representing each of the N copies of the managed characteristic application computer programs.

34. (Previously Presented) The software as recited in claim 26, wherein the second function which determines the required allocation and reallocation actions established for the M managed characteristic application computer programs by:

responding to application and host failures by determining if and what recovery actions should be taken;

determining if and where to place new copies of managed characteristic application computer programs or which managed characteristic application computer programs should be shutdown when QoS Manager functions indicate that scale up or scale down actions should be taken based on measured application performance and QoS specifications established for the M managed characteristic application computer programs;

determining where new application computer programs should be placed when requested to do so by the fourth function group; and

determining which and how many application computer programs should run based on application system priorities.

35. (Previously Presented) The software as recited in claim 26, wherein the third function group makes decisions by one of:

based on requests from one or more of the hosts, determining where new application computer programs should be started;

based on indication of application failure from the fourth function group, determining whether and where ~~the~~ a failed application computer program should be restarted;

based on indication of host failure from the fourth function group, determining whether and where the failed application computer program previously instantiated by the failed one of the N hosts should be restarted;

based on startup and shutdown dependency resolution requests from the fourth function group, determine whether and where additional application computer programs should to be one of started and shut down prior to starting or shutting down another application computer program; and

based on changes to application system priorities, determining whether and where new application computer programs need to be started and/or determine whether and which existing application computer programs need to be shutdown.

36. (Previously Presented) The software as recited in claim 26, wherein the third function group makes decisions by one of:

based on application computer program inter-dependencies defined in system specification files, determining whether and where additional application computer programs should be one of started and shut down prior to starting or shutting down of another application computer program;

based on application computer program instrumentation data generated by the second function group and performance requirements defined in the system specification files, determining whether application computer programs are meeting performance requirements and whether an application computer program can be scaled up or moved to attempt to improve performance; and

based on the application computer program instrumentation data and performance requirements defined in the system specification files, determining whether application computer programs are performing well within performance requirements and can be scaled down.

REMARKS

By this Amendment, claims 8-15 are cancelled without prejudice to or disclaimer of the subject matter contained therein. Accordingly, claims 16-20 and 22-36 are pending in this application. No new matter is added by any of these amendments.

Reconsideration based on the following remarks is respectfully requested.

I. Amendment Entry After Final Rejection

Entry of this amendment is proper under 37 CFR §1.116 because the amendments: a) place the application in condition for allowance for all the reasons discussed herein; b) do not raise any new issues requiring further search or consideration; and c) address formal requirements of the Final Rejection and preceding Office Action. Accordingly, Applicants respectfully request entry of this Amendment.

II. Allowable Subject Matter

Applicants gratefully acknowledge that the Final Office Action indicates that claims 16-20 and 21-36 are allowed.

III. Anticipatory Rejection under 35 U.S.C. §102

The Office Action rejects claims 8-9, 12, 14 and 15 as being allegedly anticipated under 35 U.S.C. §102(b) over U.S. Patent 6,041,306 to Du *et al.* (hereinafter "Du"). This rejection is rendered moot by the cancellation of the rejected claims. Applicants respectfully request that the rejection under 35 U.S.C. §102 be withdrawn.

IV. Obviousness Rejections under 35 U.S.C. §103

The Office Action further rejects claims 10, 11 and 13 as being allegedly obvious and thereby unpatentable under 35 U.S.C. §103(a) over Du in view of U.S. Patent 6,654,029 to Chiu *et al.* (hereinafter "Chiu"). This rejection is rendered moot by the cancellation of the rejected claims. Applicants respectfully request that the rejection under 35 U.S.C. §103 be withdrawn.

Applicants : Paul V. Werme *et al.*
Serial No. : 09/864,825
Filed : May 24, 2001
Page : 9 of 9

Attorney Docket No.: Navy Case 83017

V. Conclusion

In view of the foregoing, Applicants respectfully submit that this application is in condition for allowance. Favorable reconsideration and prompt allowance are earnestly solicited.

Should the Examiner believe that anything further is desirable in order to place this application in even better condition for allowance, the Examiner is invited to contact Applicants' undersigned representative at the telephone number listed below.

Respectfully submitted,

Gerhard W. Thielman
Registration No. 43,186

Date: October 17, 2006

DEPARTMENT OF THE NAVY
Naval Surface Warfare Center – Dahlgren Division
Office of Counsel – Code XDC1
17320 Dahlgren Road
Dahlgren, Virginia 22448-5100
Telephone: (540) 653-8061 **Customer No. 23501**

DEPOSIT ACCOUNT USE AUTHORIZATION
Please grant any extension necessary for entry;
Charge any fee due to our Deposit Account No. 50-0967

PAGE 11/11 * RCVD AT 10/17/2006 10:38:51 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/9 * DNIS:2738300 * CSID:5406538879 * DURATION (mm-ss):03-20

PTO/SB/06 (08-03)
Approved for use through 7/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 09/867825

CLAIMS AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA
BASIC FEE (37 CFR 1.16(a))		
TOTAL CLAIMS (37 CFR 1.16(c))	minus 20 =	*
INDEPENDENT CLAIMS (37 CFR 1.16(b))	minus 3 =	*
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(d))		

* If the difference in column 1 is less than zero, enter "0" in column 2.

SMALL ENTITY RATE	SMALL ENTITY FEE
	$
X $ =	
X $ =	
+ $ =	
TOTAL	

OR

OTHER THAN SMALL ENTITY RATE	OTHER THAN SMALL ENTITY FEE
	$
X $ =	
X $ =	
+ $ =	
TOTAL	

CLAIMS AS AMENDED – PART II

AMENDMENT A (4/04/6)

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total (37 CFR 1.16(c))	* 28	Minus	** 36	=
Independent (37 CFR 1.16(b))	* 3	Minus	*** 3	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(d))				

SMALL ENTITY RATE	SMALL ENTITY ADDITIONAL FEE
X $ =	
X $ =	
+ $ =	
TOTAL ADD'L FEE	

OR

OTHER THAN SMALL ENTITY RATE	OTHER THAN SMALL ENTITY ADDITIONAL FEE
X $ =	
X $ =	
+ $ =	
TOTAL ADD'L FEE	

AMENDMENT B (10/17/6)

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total (37 CFR 1.16(c))	* 20	Minus	** 36	=
Independent (37 CFR 1.16(b))	* 3	Minus	*** 3	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(d))				

SMALL ENTITY RATE	SMALL ENTITY ADDITIONAL FEE
X $ =	
X $ =	
+ $ =	
TOTAL ADD'L FEE	

OR

OTHER THAN SMALL ENTITY RATE	OTHER THAN SMALL ENTITY ADDITIONAL FEE
X $ =	
X $ =	
+ $ =	
TOTAL ADD'L FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total (37 CFR 1.16(c))	*	Minus	**	=
Independent (37 CFR 1.16(b))	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(d))				

SMALL ENTITY RATE	SMALL ENTITY ADDITIONAL FEE
X $ =	
X $ =	
+ $ =	
TOTAL ADD'L FEE	

OR

OTHER THAN SMALL ENTITY RATE	OTHER THAN SMALL ENTITY ADDITIONAL FEE
X $ =	
X $ =	
+ $ =	
TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/864,825	05/24/2001	Paul V. Werme	NC 83017	2372

23501 7590 08/23/2006

NAVAL SURFACE WARFARE CENTER, DAHLGREN DIVISION
OFFICE OF COUNSEL, CODE XDC1
17320 DAHLGREN ROAD
DAHLGREN, VA 22448-5110

EXAMINER	
TANG, KENNETH	

ART UNIT	PAPER NUMBER
2195	

DATE MAILED: 08/23/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)	
	09/864,825	WERME ET AL.	
	Examiner	Art Unit	
	Kenneth Tang	2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 April 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *8-20 and 22-36* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *16-20 and 22-36* is/are allowed.

6)☒ Claim(s) *8-15* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the Amendment filed on 4/26/06. Applicant's arguments have been fully considered but are not found to be persuasive.

2. Claims 8-20 and 22-36 are presented for examination.

Allowable Subject Matter

3. Claims 16-20 and 22-36 are allowed.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. **Claims 8-9, 12, and 14-15 are rejected under 35 U.S.C. 102(b) as being unpatentable over Du et al. (hereinafter Du) (US 6,041,306).**

5. As to claim 8, Du teaches on a host instantiating a managed characteristic computer program *(attribute, state, instances) (col. 2, lines 59-65, col. 5, lines 59-67 through col. 6, lines 1-10)*, the managed characteristic application computer program *(col. 2, lines 59-65, col. 5, lines 59-67 through col. 6, lines 1-10)* being managed by at least the host, a resource management device of the host, the resource management device *(see Fig. 2)* generating signals responsive to

first information received by the resource management device regarding performance *(monitor and measure progress of the process instances created) (col. 8, lines 3-6, col. 10, lines 1-4)* of a plurality of application computer programs including the managed characteristic application computer program and second information received by the resource management device regarding performance of the host *(system monitoring and network communication, etc.) (col. 2, lines 7-10)*, the signals including:

a signal that starts up an additional copy (instance) of the managed characteristic (attribute or state) application computer program on one of the host and a second networked host (database, etc) *(col. Fig. 2, 21, 34a, or 34b, etc.)*;

a signal shuts down *(stopped)* and restarts (restart or reset) *(col. 7, lines 17-24, 56-57)* the managed characteristic on the host; and

a signal that moves the managed characteristic to the second host (database, etc.) *(col. Fig. 2, 12a, 20, 23b, 26, 27, 21, 34a, or 34b, etc.)*.

In addition, an M+1 copy is taught in Du where it is disclosed that there is scalability *(col. 5, lines 16-34, etc)*. If something is scalable, it can do the same for an extra/additional or an M+1 amount, etc. Un addition, Du teaches configuration management by of one of the hosts *(col. 10, lines 5-28)*.

6. As to claim 9, Du teaches wherein the managed characteristic application computer program comprises a scalable application program *(col. 5, lines 16-34)*.

7. As to claim 12, Du teaches the resource management device wherein the resource management device further generates signals responsive to third information received by the resource management device regarding the performance of hardware operatively coupling the networked hosts *(col. 8, lines 3-6, col. 10, lines 1-4,col. 2, lines 7-10)*.

8. As to claim 14, Du teaches the resource management device wherein the managed characteristic application computer program further responds to user-initiated control actions *(users respond and interactive through the user interface, see Fig. 3, etc.)*.

9. As to claim 15, Du teaches where the action requests are generated by an operator *(col. 4, lines 9-33)*.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 10-11 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Du et al. (hereinafter Du) (US 6,041,306), and further in view of Chiu et al. (hereinafter Chiu) (US 6,654,029 B1).

11. As to claims 10-11 and 13, Du teaches the resource management device wherein the managed characteristic application computer program comprises a fault tolerant application *(col. 4, lines 58-67, col. 7, lines 55-62)*. Du fails to explicitly teach where the degree of fault tolerance is selectable by a user by priority. However, Chiu teaches a scalable, object-oriented architecture with a user selecting the priority (importance) level of fault tolerance *(col. 5, lines 20-25)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Du and Chiu because the user could react based on importance *(col. 5, lines 20-25)*.

Response to Arguments

12. During patent examination, the pending claims must be "given their broadest reasonable interpretation consistent with the specification." *In re Hyatt*, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000). Applicant always has the opportunity to amend the claims during prosecution, and broad interpretation by the examiner reduces the possibility that the claim, once issued, will be interpreted more broadly than is justified. *In re Prater*, 415 F.2d 1393, 1404-05, 162 USPQ 541, 550-51 (CCPA 1969).

13. Applicant argues that Du does not teach having a first signal that controls a configuration of an Mth copy and having a second signal for an M+1th copy, a third signal that controls shutdown and restart, and a fourth signal that controls movement, for example.

In response, the office action has shown how Du has taught the claimed invention. Applicant ignores the cited portions that the Examiner has pointed for this teaching in the

rejection. Instead, Applicant points to other areas of the reference. Applicant's arguments were not found to be persuasive.

14. Applicant argues the 35 USC 103 rejection by stating that Chiu does not compensate for the deficiencies of Du.

Again, the office action has shown how Du has taught the claimed invention. Applicant ignores the cited portions that the Examiner has pointed for this teaching in the rejection. Instead, Applicant points to other areas of the reference. Applicant's arguments were not found to be persuasive.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kenneth Tang whose telephone number is (571) 272-3772. The examiner can normally be reached on 8:30AM - 6:00PM, Every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Kt
8/15/06

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Search Notes	Application No.	Applicant(s)	
	09/864,825	WERME ET AL.	
	Examiner	Art Unit	
	Kenneth Tang	2127	

SEARCHED			
Class	Subclass	Date	Examiner
718	100-108	6/27/2004	KT
705	8	6/27/2004	KT
370	217	6/27/2004	KT

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
EAST	6/27/2004	KT
EAST	2/15/2005	KT
EAST, text search	1/22/2006	KT
EAST, text search	8/15/2006	KT

Index of Claims

Application No.	Applicant(s)
09/864,825	WERME ET AL.
Examiner	**Art Unit**
Kenneth Tang	2127

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim		Date								
Final	Original		2/15/05	1/22/06	8/15/06					
	1	√	√	√						
	2	√	√	√						
	3	√	√	√						
	4	√	√	√						
	5	√	√	√						
	6	√	√	√						
	7	√	√	√						
	8	√	√	√	√					
	9	√	√	√	√					
	10	√	√	√	√					
	11	√	√	√	√					
	12	√	√	√	√					
	13	√	√	√	√					
	14	√	√	√	√					
	15	√	√	√	√					
	16	√	√	√	=					
	17	√	√	√	=					
	18	√	√	√	=					
	19	√	√	√	=					
	20	√	√	√	=					
	21	√	√	o						
	22	√	√	√	=					
	23	√	√	√	=					
	24	√	√	√	=					
	25	√	√	√	=					
	26	√	√	=	=					
	27	√	√	=	=					
	28	√	√	=	=					
	29	√	√	=	=					
	30	√	√	=	=					
	31	√	√	=	=					
	32	√	√	=	=					
	33	√	√	=	=					
	34	√	√	=	=					
	35	√	√	=	=					
	36	√	√	√	=					
	37									
	38									
	39									
	40									
	41									
	42									
	43									
	44									
	45									
	46									
	47									
	48									
	49									
	50									

Claim		Date								
Final	Original									
	51									
	52									
	53									
	54									
	55									
	56									
	57									
	58									
	59									
	60									
	61									
	62									
	63									
	64									
	65									
	66									
	67									
	68									
	69									
	70									
	71									
	72									
	73									
	74									
	75									
	76									
	77									
	78									
	79									
	80									
	81									
	82									
	83									
	84									
	85									
	86									
	87									
	88									
	89									
	90									
	91									
	92									
	93									
	94									
	95									
	96									
	97									
	98									
	99									
	100									

Claim		Date								
Final	Original									
	101									
	102									
	103									
	104									
	105									
	106									
	107									
	108									
	109									
	110									
	111									
	112									
	113									
	114									
	115									
	116									
	117									
	118									
	119									
	120									
	121									
	122									
	123									
	124									
	125									
	126									
	127									
	128									
	129									
	130									
	131									
	132									
	133									
	134									
	135									
	136									
	137									
	138									
	139									
	140									
	141									
	142									
	143									
	144									
	145									
	146									
	147									
	148									
	149									
	150									

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1758	scalable and (resource adj manag$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L2	1695	scalable and (resource adj manag$5) and network$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L3	742	@ad<="20010524" and (scalable and (resource adj manag$5) and network$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L4	262	(start$3 same (shut$3 ends) same restart$3) with (application program)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L5	1	(scalable and (resource adj manag$5)) and @ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L6	1	(scalable and (resource adj manag$5) and network$3) and @ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L7	174	@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L8	1	(scalable and (resource adj manag$5)) and (@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L9	34	(start$3 same (shut$3 ends) same restart$3) with (copy instance)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24

EAST Search History

L10	240	resumption with (copy instance)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L11	767	@ad<="20010524" and (scalable and (resource adj manag$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L12	1	(scalable and (resource adj manag$5)) and (resumption with (copy instance))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L13	151	@ad<="20010524" and (resumption with (copy instance))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L14	0	((US-20020064126-$).did.) and scalable	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24
L15	0	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020064126-$).did.) and scalable	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24
L16	180	scalable adj application	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24
L17	20810169	@ad<="20010524"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L18	0	((US-20020064126-$).did.) and performance	US-PGPUB	OR	ON	2006/08/15 15:24
L19	11	(@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))) and (resource adj manag$5) and network$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L20	25	@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (copy instance))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24

EAST Search History

L21	41	(@ad<="20010524" and (resumption with (copy instance))) and network	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L22	34	((starts and resum$) with (copy instance)) same (manager controller)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L23	25	@ad<="20010524" and (((starts and resum$) with (copy instance)) same (manager controller))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L24	1	("20020064126").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2006/08/15 15:24
L25	3	quality adj of adj service	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L26	4	(US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020064126-$).did.	US-PGPUB; USPAT	OR	OFF	2006/08/15 15:24
L27	76	(scalable adj application) and @ad<="20010524"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L28	1	((US-20020174420-$).did.) and performance	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L29	1	((US-20020064126-$).did.) and perform$	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L30	1	((US-20020064126-$).did.) and application	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L31	5	(US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.	US-PGPUB; USPAT	OR	OFF	2006/08/15 15:24

EAST Search History

L32	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and monitor$3	USPAT	OR	ON	2006/08/15 15:24
L33	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and monitor$3 and (status state)	USPAT	OR	ON	2006/08/15 15:24
L34	1	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and qos	USPAT	OR	ON	2006/08/15 15:24
L35	1	((US-20020174420-$).did.) and states	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L36	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and automat$	USPAT	OR	ON	2006/08/15 15:24
L37	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and scal$	USPAT	OR	ON	2006/08/15 15:24
L38	1	("20030167270").PN.	US-PGPUB; USPAT	OR	OFF	2006/08/15 15:24
L39	76	(resource adj manag$3) with scal$6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L41	35	L39 and L40	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L42	29	L41 and (instance instantiat$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24

EAST Search History

L43	30	L41 and (instance instantiat$4 image)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L44	30	L41 and (instance instantiat$4 image copy)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L45	151	(resource adj (manager management)) with scal$6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L46	62	L40 and L45	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L47	31	L46 and qos	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L48	31	L47 and L40	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L49	31	L48 and (host network server client)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L50	19151236	@ad<="20000425"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L51	151	(resource adj (manager management)) with scal$6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24

EAST Search History

L52	62	L50 and L51	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L53	31	L52 and qos	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L54	31	L53 and L50	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L55	31	L54 and (host network server client)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L56	30	L55 and (application program)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L57	28	L55 and signal	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L58	1	(US-5826239-$).did.	USPAT	OR	OFF	2006/08/15 15:24
L59	1	L58 and (configur$4 configuration)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L60	100886	"9" and ((application program) with instance)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L61	1	(US-5826239-$).did.	USPAT	OR	OFF	2006/08/15 15:24

EAST Search History

L62	0	L61 and ((application program) with instance)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L63	1	L61 and ((application program) same instance)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L64	0	L61 and (mov$3 transfer$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L65	1	L61 and (communicat$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L66	1	L61 and attribute	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L67	1	L61 and restart$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L68	1	(US-20030167270-$).did.	US-PGPUB	OR	OFF	2006/08/15 15:24
L69	0	L68 and characteristic	US-PGPUB; USPAT	OR	OFF	2006/08/15 15:24
L70	1	L68 and characteristic	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24
L71	1	L68 and fault	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24
L72	0	L61 and fault	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24
L73	1	L61 and fail$4	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24
L74	0	L61 and (weigh$4 priority rank$4)	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24

EAST Search History

L75	1	L61 and user	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24
L77	1	(US-6041306-$).did.	USPAT	OR	OFF	2006/08/15 15:24
L78	1	L77 and scalab$6	USPAT	OR	OFF	2006/08/15 15:24
L79	1	L77 and table	USPAT	OR	OFF	2006/08/15 15:24
L80	1	L77 and configuration	USPAT	OR	ON	2006/08/15 15:24
L81	1	L77 and instance	USPAT	OR	ON	2006/08/15 15:24
L82	1	L77 and (instance with (transfer$4 sending sent send transmit$3 network host client server))	USPAT	OR	ON	2006/08/15 15:24
L83	1	L77 and ((instance attribute) with (transfer$4 sending sent send transmit$3 network host client server))	USPAT	OR	ON	2006/08/15 15:24
L84	539	((instance copy image) near3 program) with (transfer$4 sending sent send transmit$3) with (network host client server)	USPAT	OR	ON	2006/08/15 15:24
L85	425	@ad<="20010524" and L84	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L86	71	L85 and ((application program) with priority)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L87	10	L86 and ((scalable scalability) with (application program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L88	1	L85 and (user with (fault adj toleran$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L89	2	(((characteristic attribute) near3 (program application)) with priority) and (user with (fault adj toleran$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L90	73	(((characteristic attribute) near3 (program application))) and (user with (fault adj toleran$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24

EAST Search History

L91	63	@ad<="20010524" and L90	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L92	769	(((characteristic attribute) with (program application))) with priority	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L93	4	L91 and L92	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L94	51	(resource adj allocation) and (user with (fault adj toleran$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L95	31	@ad<="20010524" and L94	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L96	13	(resource adj allocation) and (user with (fault adj toleran$4) with (select$4 control$3 manag$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L97	4	@ad<="20010524" and L96	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L98	102	@ad<="20010524" and (user with (fault adj toleran$4) with (select$4 control$3 manag$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L99	13	@ad<="20010524" and (user with (fault adj toleran$4) with (select$4 control$3 manag$4) with (level degree))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L100	1	(US-6654029-$).did.	USPAT	OR	OFF	2006/08/15 15:24

EAST Search History

L101	1	L100 and fault	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L102	1	L100 and level	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L103	1	L100 and user	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L104	1	L77 and (device with user)	USPAT	OR	ON	2006/08/15 15:24
L105	613137	"1" and (fault fail$6)	USPAT	OR	ON	2006/08/15 15:24
L106	1	(US-6041306-$).did.	USPAT	OR	OFF	2006/08/15 15:24
L107	1	L106 and (fault fail$6)	USPAT	OR	ON	2006/08/15 15:24
L108	1758	scalable and (resource adj manag$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L109	1695	scalable and (resource adj manag$5) and network$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L110	742	@ad<="20010524" and (scalable and (resource adj manag$5) and network$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L111	262	(start$3 same (shut$3 ends) same restart$3) with (application program)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L112	1	(scalable and (resource adj manag$5)) and @ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L113	1	(scalable and (resource adj manag$5) and network$3) and @ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24

EAST Search History

L114	174	@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L115	1	(scalable and (resource adj manag$5)) and (@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L116	34	(start$3 same (shut$3 ends) same restart$3) with (copy instance)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L117	240	resumption with (copy instance)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L118	767	@ad<="20010524" and (scalable and (resource adj manag$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L119	1	(scalable and (resource adj manag$5)) and (resumption with (copy instance))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L120	151	@ad<="20010524" and (resumption with (copy instance))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L121	0	((US-20020064126-$).did.) and scalable	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24
L122	0	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020064126-$).did.) and scalable	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24
L123	180	scalable adj application	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24
L124	20810169	@ad<="20010524"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24

EAST Search History

L125	0	((US-20020064126-$).did.) and performance	US-PGPUB	OR	ON	2006/08/15 15:24
L126	151	(resource adj (manager management)) with scal$6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L127	19151236	@ad<="20000425"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L128	62	L127 and L126	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L129	31	L128 and qos	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L130	19151236	@ad<="20000425"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L131	151	(resource adj (manager management)) with scal$6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L132	62	L130 and L131	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L133	31	L132 and qos	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24

EAST Search History

L134	31	L133 and L130	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L135	100886	"9" and ((application program) with instance)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L136	1	(US-5826239-$).did.	USPAT	OR	OFF	2006/08/15 15:24
L137	0	L136 and ((application program) with instance)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L138	0	L136 and (mov$3 transfer$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L139	1	(US-20030167270-$).did.	US-PGPUB	OR	OFF	2006/08/15 15:24
L140	0	L139 and characteristic	US-PGPUB; USPAT	OR	OFF	2006/08/15 15:24
L141	0	L136 and fault	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24
L142	0	L136 and (weigh$4 priority rank$4)	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24
L143	539	((instance copy image) near3 program) with (transfer$4 sending sent send transmit$3) with (network host client server)	USPAT	OR	ON	2006/08/15 15:24
L144	425	@ad<="20010524" and L143	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L145	1	L144 and (user with (fault adj toleran$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24

EAST Search History

L146	73	(((characteristic attribute) near3 (program application))) and (user with (fault adj toleran$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L147	63	@ad<="20010524" and L146	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L148	769	(((characteristic attribute) with (program application))) with priority	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L149	51	(resource adj allocation) and (user with (fault adj toleran$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L150	13	(resource adj allocation) and (user with (fault adj toleran$4) with (select$4 control$3 manag$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L151	102	@ad<="20010524" and (user with (fault adj toleran$4) with (select$4 control$3 manag$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L152	1	("20020064126").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2006/08/15 15:24
L153	1	((US-20020174420-$).did.) and performance	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L154	1	((US-20020064126-$).did.) and perform$	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L155	1	((US-20020064126-$).did.) and application	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24

EAST Search History

L156	1	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and qos	USPAT	OR	ON	2006/08/15 15:24
L157	1	((US-20020174420-$).did.) and states	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L158	1	("20030167270").PN.	US-PGPUB; USPAT	OR	OFF	2006/08/15 15:24
L159	1	(US-5826239-$).did.	USPAT	OR	OFF	2006/08/15 15:24
L160	1	L159 and (configur$4 configuration)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L161	1	(US-5826239-$).did.	USPAT	OR	OFF	2006/08/15 15:24
L162	1	L136 and ((application program) same instance)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L163	1	L136 and (communicat$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L164	1	L136 and attribute	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L165	1	L136 and restart$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L166	1	(US-20030167270-$).did.	US-PGPUB	OR	OFF	2006/08/15 15:24
L167	1	L139 and characteristic	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24
L168	1	L139 and fault	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24

EAST Search History

L169	1	L136 and fail$4	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24
L170	1	L136 and user	US-PGPUB; USPAT	OR	ON	2006/08/15 15:24
L171	1	(US-6041306-$).did.	USPAT	OR	OFF	2006/08/15 15:24
L172	1	L171 and scalab$6	USPAT	OR	OFF	2006/08/15 15:24
L173	1	L171 and table	USPAT	OR	OFF	2006/08/15 15:24
L174	1	L171 and configuration	USPAT	OR	ON	2006/08/15 15:24
L175	1	L171 and instance	USPAT	OR	ON	2006/08/15 15:24
L176	1	L171 and (instance with (transfer$4 sending sent send transmit$3 network host client server))	USPAT	OR	ON	2006/08/15 15:24
L177	1	L171 and ((instance attribute) with (transfer$4 sending sent send transmit$3 network host client server))	USPAT	OR	ON	2006/08/15 15:24
L178	2	(((characteristic attribute) near3 (program application)) with priority) and (user with (fault adj toleran$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L179	1	(US-6654029-$).did.	USPAT	OR	OFF	2006/08/15 15:24
L180	1	L179 and fault	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L181	1	L179 and level	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L182	1	L179 and user	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L183	1	L171 and (device with user)	USPAT	OR	ON	2006/08/15 15:24
L184	11	(@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))) and (resource adj manag$5) and network$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24

EAST Search History

L185	25	@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (copy instance))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L186	41	(@ad<="20010524" and (resumption with (copy instance))) and network	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L187	34	((starts and resum$) with (copy instance)) same (manager controller)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L188	25	@ad<="20010524" and (((starts and resum$) with (copy instance)) same (manager controller))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L189	3	quality adj of adj service	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L190	4	(US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020064126-$).did.	US-PGPUB; USPAT	OR	OFF	2006/08/15 15:24
L191	76	(scalable adj application) and @ad<="20010524"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L192	5	(US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.	US-PGPUB; USPAT	OR	OFF	2006/08/15 15:24
L193	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and monitor$3	USPAT	OR	ON	2006/08/15 15:24
L194	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and monitor$3 and (status state)	USPAT	OR	ON	2006/08/15 15:24

EAST Search History

L195	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and automat$	USPAT	OR	ON	2006/08/15 15:24
L196	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and scal$	USPAT	OR	ON	2006/08/15 15:24
L197	76	(resource adj manag$3) with scal$6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L198	35	L197 and L127	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L199	29	L198 and (instance instantiat$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L200	30	L198 and (instance instantiat$4 image)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L201	30	L198 and (instance instantiat$4 image copy)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L202	31	L129 and L127	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L203	31	L202 and (host network server client)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24

EAST Search History

L204	31	L134 and (host network server client)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L205	30	L204 and (application program)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L206	28	L204 and signal	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L207	71	L144 and ((application program) with priority)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L208	10	L207 and ((scalable scalability) with (application program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L209	73	(((characteristic attribute) near3 (program application))) and (user with (fault adj toleran$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L210	4	L147 and L148	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L211	31	@ad<="20010524" and L149	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24
L212	4	@ad<="20010524" and L150	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24

EAST Search History

L213	13	@ad<="20010524" and (user with (fault adj toleran$4) with (select$4 control$3 manag$4) with (level degree))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/08/15 15:24

OIPE IAP46
APR 26 2006
PATENT & TRADEMARK OFFICE

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants :	Paul V. Werme *et al.*	Group Art Unit :	2195
Serial No. :	09/864,825	Examiner :	Kenneth Tang
Filed :	May 24, 2001	Docket :	Navy Case 83017
Title :	RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR		

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT UNDER 37 CFR §1.111

In response to the January 25, 2006 Office Action, the shortened statutory period for reply effectively ending April 25, 2006, please consider the following:

Amendments to the Specification are reflected in the paragraphs at pages 2-5;

Amendments to the Claims are reflected in the listing of claims at pages 6-13; and

Amendments to the Drawings include an attachment replacement sheet listed at page 14; and

Remarks at pages 15-18.

CERTIFICATE OF TRANSMISSION / MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, Mail Stop: Amendment, P.O. Box 1450, Alexandria VA 22313-1450 on the date identified below:

Date: April 20, 2006 Name: Deborah G. Blakley
Deborah G. Blakley

Amendments to the Specification:

Please replace the Abstract with the attached replacement Abstract.

Please replace the paragraph at page 6, lines 15-22 in the specification (corresponding to the API definition immediately following paragraph [0032] of U.S. Patent Application Publication 2003/0167270) with the following rewritten paragraph:

A set of subroutines or functions that a program, or *application*, can call to invoke ~~sone~~ some functionality contained in another software or hardware component. The Windows API consists of more than 1,000 functions that programs written in C, C++, Pascal, and other languages can call to create windows, open files, and perform other essential tasks. An application that wants to display an on-screen message can call Windows' *MessageBox* API function, for example.

Please replace the paragraphs at page 46, lines 1-27 in the specification (corresponding to paragraphs [0226] – [0228] of U.S. Patent Application Publication 2003/0167270) with the following rewritten paragraphs:

The ~~Proggram~~ Program Control function FG50 also receives periodic heartbeat messages, e.g., once per second, from each of the Program Control Agents FG52A-FG52N, as discussed above. If Fault Detection has been enabled at the Program Control Displays FG54A-FG54N, if three consecutive heartbeat messages from an Agent, e.g., FG52A, are missed, the host that the agent is running on is declared down and all linked functions, including the Resource Manager FG42 and the Displays FG54A-FG54N are notified.

As mentioned above, the ~~Proggram~~ Program Control function FG50 sends out periodic application status updates as well as immediate notification when applications are started up, are shutdown, or fail. These notifications are sent out to all linked functions.

It should be noted that the ~~Proggram~~ Program Control function FG50 uses the same message traffic and internal processing for handling application startup and shutdown orders received from either the Resource Manager FG42 or from the Program Control Displays FG54A-FG54N. However, if a startup order received from one of the Program Control Displays FG54A-FG54N indicates that the Resource Manager FG42 should determine where to run the application, a request to allocate the application is sent to the Resource Manager FG42. When no response is received from the Resource Manager FG42 within a predetermined timeout period, the ~~Proggram~~ Program Control function FG50 will automatically start the application on the default host. Moreover, when an application startup cannot proceed due to an unfulfilled application startup dependency, a request will be made to the Resource Manager FG42 to attempt to resolve the dependency. If the Resource Manager FG42 either cannot resolve the dependency or no response is received within a predetermined timeout period, the application startup will fail, and a "dependency failed" indication will be sent to the Display. It will be appreciated that this will cause the application status to be displayed in, for example, yellow and post an alert to the Alert window on one of the Program Control Displays FG54A-FG54N.

Please replace the paragraphs at page 47, line 28 – page 48, line 7 in the specification (corresponding to paragraph [0244] of U.S. Patent Application Publication 2003/0167270) with the following rewritten paragraph:

It will be appreciated from FIG. 2 that multiple Program Control Displays FG54A-FG54N advantageously can be run simultaneously. If this is done, any configuration change actions will be reflected on all the displays. Whenever application stop or start actions are taken by the display operator, a message is sent to the ~~Proggram~~ Program Control function FG50 which is responsible for enacting the start or stop action. The Program Control function FG50 also sends indications of any status changes to the Program Control Displays FG54A-FG54N as soon as the status changes are seen. In addition, periodic status updates are also sent to the Program Control Displays FG54A-FG54M.

Please replace the paragraphs at page 49, lines 7-9 in the specification (corresponding to paragraph [0251] of U.S. Patent Application Publication 2003/0167270) with the following rewritten paragraph:

The RM_Start flag, which identifies whether the ~~the~~ Resource Manager should be queried at run-time to determine what host the application should be started on. The valid values are "0" for "NO" and "1" for "YES".

Please replace the paragraphs at page 52, lines 12-17 in the specification (corresponding to paragraph [0270] of U.S. Patent Application Publication 2003/0167270) with the following rewritten paragraph:

FIGS. 10 and 11 are screen captures of the Graph Tool Instrumentation Displays FG69A-FG69N, which depict user-configurable displays capable of ~~receiveing~~ receiving data via standardized message formats and open interfaces. The Graph Tool Displays FG69A-FG69N allow the operator to select and configure various display widgets (line graphs, bar charts, pie charts, meters, and text boxes) to build a desired display layout. Data sources for driving the widgets can also be selected interactively.

Please replace the paragraphs at page 60, lines 9-22 in the specification (corresponding to paragraph [0332] of U.S. Patent Application Publication 2003/0167270) with the following rewritten paragraph:

It should also be noted that the Resource Manager FG42 sends application system and hardware system readiness and system (mission) priority information to the Readiness Broker, which is a translator ~~withing~~ within the Readiness Display FG66 and to the Globus Broker (another Broker (not shown)). The Readiness Broker is responsible for driving a GUI/display FG66, which shows the current readiness data and allows the system (mission) priorities to be changed and sent back to the Resource Manager FG42. The Globus Broker provides basically the same functionality except that only a high-level subset of the readiness data provided to the

Readiness Broker is provided to the Globus Broker. The readiness information sent to the Readiness Broker consists of readiness values for each application, application subsystem, and application system defined in the System Specification Files FG32. The scores advantageously can be based on the status (up/down) of the applications and the percentage of potential copies of scalable applications that are currently running. Host and network readiness scores are determined based on the host loads and host fitness scores received from the Hardware Broker FG40.

Please replace the paragraphs at page 96, lines 22-26 in the specification (corresponding to paragraph [0463] of U.S. Patent Application Publication 2003/0167270) with the following rewritten paragraph:

Table III provides a listing of the ~~Appendicies~~ Appendices included for all purposes in the application. It will be noted that the majority of the listed ~~Apendicies~~ Appendices are provided on the CD-ROM filed concurrently with the application. In addition, the CD-ROM also includes the source code listing for the Resource Management Architecture according to the present invention.

Amendments to the Claims:

The following listing of claims will replace all prior versions, and listings, of claims in the application:

1-7. (Cancelled)

8. (Previously Presented) In a grid system comprising networked hosts instantiating M managed characteristic application computer programs, a resource management device within one of the hosts generating signals responsive to first information received by the resource management device regarding performance of the M managed characteristic application computer programs and second information received by the resource management device regarding performance of the hosts, where M is a positive integer, the signals including:

a first signal that controls a configuration of an Mth copy of the M managed characteristic application computer programs, each of the M managed characteristic application computer programs being managed by one of the hosts;

a second signal that controls a start up of an $(M+1)^{th}$ copy of the M managed characteristic application computer programs;

a third signal that controls a shutdown and restart of the Mth copy of the M managed characteristic application computer programs; and

a fourth signal that controls a movement of the Mth copy of the M managed characteristic application computer programs from a first to a second host.

9. (Previously Presented) The resource management device as recited in claim 8, wherein at least one of the M managed characteristic application computer programs comprises a scalable application computer program.

10. (Previously Presented) The resource management device as recited in claim 8, wherein at least one of the M managed characteristic application computer programs comprise a fault

tolerant application computer program, where the degree of fault tolerance is selectable by a user.

11. (Previously Presented) The resource management device as recited in claim 8, wherein at least one of the M managed characteristic application computer programs comprises a selectable priority application computer program.

12. (Previously Presented) The resource management device as recited in claim 8, wherein the resource management device further generates signals responsive to third information received by the resource management device regarding the performance of hardware operatively coupling the networked hosts.

13. (Previously Presented) The resource management device as recited in claim 8, wherein the resource management device further generates signals responsive to fourth information received by the resource management device establishing one of the priority assigned to the managed characteristic application computer program and multiple copies of the managed characteristic application computer program.

14. (Previously Presented) The resource management device as recited in claim 8, wherein the resource management device further generates signals responsive to externally generated action requests received by the resource management device.

15. (Original) The resource management device as recited in claim 14, where the action requests are generated by an operator.

16. (Currently Amended) In a distributed environment comprised of N hosts operating in a distributed environment instantiating M managed characteristic application computer programs managed by the N hosts, resource allocation control software instantiated by at least the N hosts,

M and N each being a positive integer, where M may be equal to, less than, or greater than N, the software comprising:

a first function which determines a state and health of the N hosts, a network operatively coupling the N hosts to one another, and the M managed characteristic application computer programs in the distributed environment;

a second function which determines required allocation and reallocation actions needed to maintain a plurality of Quality of Service (QoS) requirements established for the M managed characteristic application computer programs, the QoS requirements dictating parameters regarding service quality of the M managed characteristic application computer programs; and

a third function which generates automatic control signal requests corresponding to the actions dictated by the QoS requirements, such that the managed characteristic application computer programs are moved, shutdown, and started in accordance with satisfaction of the QoS requirements, wherein the second function determines the required allocation and reallocation actions needed to maintain the Quality of Service (QoS) requirements established for the M managed characteristic applications by:

responding to application and host failures by determining if and what recovery actions should be taken;

determining if and where to place new copies of one of the M managed characteristic application computer programs or which of the M managed characteristic application computer programs should be shutdown when QoS Manager functions indicate that scale up or scale down actions are indicated based on measured application performance and application QoS specifications;

determining where new application computer programs should be placed when requested to do so by a program control device; and

determining which and how many application computer programs should run based on application system priorities.

17. (Original) The software as recited in claim 16, wherein the first function receives system specification information comprising selected ones of host configuration and capabilities,

application capabilities, survivability requirements, scalability characteristics, application startup and shutdown dependencies, and application and path performance requirements.

18. (Previously Presented) The software as recited in claim 16, wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic application computer programs, and detected failures regarding the N hosts.

19. (Previously Presented) The software as recited in claim 16, wherein the first function receives application performance data representing each of the M managed characteristic application computer programs.

20. (Previously Presented) The software as recited in claim 16, wherein the first function receives application performance data on one or more applications instantiated by the N hosts including performance data representing each of the M managed characteristic application computer programs.

21. (Cancelled)

22. (Previously Presented) The software as recited in claim 16, wherein at least one of the M managed characteristic application computer programs comprises a scalable application computer program.

23. (Previously Presented) The software as recited in claim 16, wherein at least one of the M managed characteristic application computer programs comprises a fault tolerant application computer program, where the degree of fault tolerance is selectable by a user.

24. (Previously Presented) The software as recited in claim 16, wherein one of the M managed characteristic application computer programs comprises a selectable priority application computer program.

25. (Previously Presented) The software as recited in claim 16, wherein the M managed characteristic application computer programs comprise M copies of a single managed characteristic application computer program.

26. (Previously Presented) Software stored on at least one host for converting N networked hosts into a resource managed system instantiating M managed characteristic application computer programs, each managed characteristic application computer program managed by one of the N networked hosts, the software comprising:

a first function group which monitors the host and network resources;

a second function group which provides application computer program event reporting and event correlation capabilities;

a third function group which provides reasoning and decision making capabilities for the resource managed system, wherein the third function group comprises:

a first function which determines ~~the~~ a state and health of the N hosts, a network operatively coupling the N hosts to one another and the M managed characteristic application computer programs in the distributed environment;

a second function which determines required allocation and reallocation actions needed to maintain a plurality of Quality of Service (QoS) requirements established for the M managed characteristic application computer programs, the QoS requirements dictating parameters regarding service quality of the M management characteristic application programs; and

a third function which generates automatic control signal requests corresponding to the actions dictated by the QoS requirements, such that the managed characteristic application computer programs are moved, shutdown, and started in accordance with satisfaction of the QoS requirements; and

a fourth function group which provides program control capabilities permitting starting, stopping, and configuring of selected ones of the M managed characteristic application computer programs on respective ones of the N hosts in the resource managed system,

where M and N are positive integers and where M may be equal to, greater than, or less than N.

27. (Previously Presented) The software as recited in claim 26, wherein the first function receives system specification information comprising host configuration and capabilities.

28. (Previously Presented) The software as recited in claim 26, wherein the first function receives system specification information comprising selected ones of capabilities, survivability requirements, scalability characteristics, startup and shutdown dependencies, and performance requirements for at least one of the M managed characteristic application computer programs.

29. (Original) The software as recited in claim 26, wherein the first function receives system specification information comprising path performance requirements regarding communication between at least two of the N hosts.

30. (Previously Presented) The software as recited in claim 26, wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic application computer programs, and detected failures regarding the N hosts.

31. (Original) The software as recited in claim 26, wherein the first function receives historical data regarding statuses, configuration, and loads of the N hosts and link statuses and loads regarding the network.

32. (Previously Presented) The software as recited in claim 26, wherein the first function receives application performance data representing each one of the M managed characteristic application computer programs.

33. (Previously Presented) The software as recited in claim 26, wherein the first function receives application performance data on one or more applications instantiated by the N hosts including performance data representing each of the N copies of the managed characteristic application computer programs.

34. (Previously Presented) The software as recited in claim 26, wherein the second function which determines the required allocation and reallocation actions established for the M managed characteristic application computer programs by:

responding to application and host failures by determining if and what recovery actions should be taken;

determining if and where to place new copies of managed characteristic application computer programs or which managed characteristic application computer programs should be shutdown when QoS Manager functions indicate that scale up or scale down actions should be taken based on measured application performance and QoS specifications established for the M managed characteristic application computer programs;

determining where new application computer programs should be placed when requested to do so by the fourth function group; and

determining which and how many application computer programs should run based on application system priorities.

35. (Previously Presented) The software as recited in claim 26, wherein the third function group makes decisions by one of:

based on requests from one or more of the hosts, determining where new application computer programs should be started;

based on indication of application failure from the fourth function group, determining whether and where ~~the~~ a failed application computer program should be restarted;

based on indication of host failure from the fourth function group, determining whether and where the failed application computer program previously instantiated by the failed one of the N hosts should be restarted;

based on startup and shutdown dependency resolution requests from the fourth function group, determine whether and where additional application computer programs should to be one of started and shut down prior to starting or shutting down another application computer program; and

based on changes to application system priorities, determining whether and where new application computer programs need to be started and/or determine whether and which existing application computer programs need to be shutdown.

36. (Currently Amended) The software as recited in claim 26, wherein the third function group makes decisions by one of:

based on application computer program ~~inter dependencies~~ inter-dependencies defined in system specification files, determining whether and where additional application computer programs should ~~to~~ be one of started and shut down prior to starting or shutting down of another application computer program;

based on application computer program instrumentation data generated by the second function group and performance requirements defined in the system specification files, determining whether application computer programs are meeting performance requirements and whether an application computer program can be scaled up or moved to attempt to improve performance; and

based on the application computer program instrumentation data and performance requirements defined in the system specification files, determining whether application computer programs are performing well within performance requirements and can be scaled down.

Amendments to the Drawings:

The attached sheet of drawings includes changes to Fig. 2A. This sheet, which includes Fig. 2A, replaces the original sheet including Fig. 2A.

Attachment: Replacement Sheet: Fig. 2A

REMARKS

By this Amendment, the Abstract is replaced, Fig. 2A is replaced pursuant to the attached drawing sheet, claims 1-7 and 21 are cancelled without prejudice to or disclaimer of the subject matter contained therein, and claims 16 and 36 are amended. Accordingly, claims 8-20 and 22-36 are pending in this application. No new matter is added by any of these amendments.

Reconsideration based on the following remarks is respectfully requested.

I. Amendments to the Specification and the Drawings

The specification is amended to correct minor inconsistencies. The corresponding paragraphs of U.S. Patent Application Publication 2003/0167270 are also identified. The substitute drawings also filed September 19, 2001 are corrected to replace an identification label "FG43" in Fig. 2A for the Resource Manager with --FG42-- to be consistent with the specification and Fig. 14. No new matter is added by these amendments.

II. Objection to the Specification

The Office Action objects to the specification, and in particular to the Abstract for minor informalities. The Abstract is replaced as attached correcting the informalities. Applicants respectfully request that the objection to the specification be withdrawn.

III. Objection to the Claims

The Office Action objects to claim 36 for minor informalities. Claim 36 is amended accordingly. Applicants respectfully request that the objection to the claims be withdrawn.

IV. Indefiniteness Rejection under 35 U.S.C. §112, second paragraph

The Office Action rejects claim 36 as being allegedly indefinite under 35 U.S.C. §112, second paragraph. As such, Applicants amend claim 36 to obviate the rejection. Applicants respectfully request that the rejection under 35 U.S.C. §112, second paragraph, be withdrawn.

V. Allowable Subject Matter

Applicants gratefully acknowledge that the Office Action indicates that claims 26-35 are allowed and that claims 21 and 36 contain allowable subject matter. As such, Applicants amend

claim 16 to incorporate the allowable features from claim 21. Applicants also amend claim 36 to obviate the objection and indefiniteness rejection.

VI. Anticipatory Rejection under 35 U.S.C. §102

The Office Action rejects claims 1-3, 6-9, 12, 14 and 15 as being allegedly anticipated under 35 U.S.C. §102(b) over U.S. Patent 6,041,306 to Du *et al.* (hereinafter "Du"). This rejection is rendered moot for claims 1-3, 6 and 7 and is respectfully traversed for the remaining claims.

Applicants' claims are directed generally, for example, to a resource management device generating signals responsive to information on performance of the M managed characteristic application computer programs and the hosts. For example, Applicants' independent claim 8 recites the signals including, *inter alia*, a first signal that controls a configuration of an Mth copy of the programs, each of the M managed characteristic application computer programs being managed by one of the hosts; a second signal that controls a start up of an $(M+1)^{th}$ copy of the programs; a third signal that controls a shutdown and restart of the Mth copy; and a fourth signal that controls a movement of the Mth copy from first to second hosts. Applicants respectfully submit that Du does not describe or suggest at least these claimed features. This also applies by extension to claims 9, 12, 14 and 15 based on their dependence from claim 8.

Instead, Du discloses a workflow process for a distributed management system 10. In particular, Du teaches a process manager engine 20 in communication with a database 21 and switchware 26 for resource managers 28. See col. 5, lines 44-58 and Figs. 2 and 5 of Du. The portions of Du cited in the Office Action merely describe example Hewlett Packard middleware and configuration management systems. These examples do not appear to have any association with starting and moving application program copies, as provided in Applicants' claims.

A claim must be literally disclosed for a proper rejection under §102(a), (b) or (e). This requirement is satisfied "only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference" (MPEP §2131). Applicants assert that the Office Action fails to satisfy this requirement with Du. For at least these reasons, Applicants respectfully request that the rejection under 35 U.S.C. §102 be withdrawn.

VII. Obviousness Rejections under 35 U.S.C. §103

A. The Office Action further rejects claims 4, 5, 10, 11 and 13 as being allegedly obvious and thereby unpatentable under 35 U.S.C. §103(a) over Du in view of U.S. Patent 6,654,029 to Chiu *et al.* (hereinafter "Chiu"). This rejection is rendered moot for claims 4 and 5 and is respectfully traversed with respect to the remaining claims.

Chiu does not compensate for the deficiencies of Du outlined above for claim 8. Nor does Chiu teach, disclose or suggest the additional features recited in claims 10, 11 and 13, including those relating to fault tolerance and priority assignments. Instead, Chiu discloses an asset management system. In particular, Chiu teaches a tools 102, library 104 and program resources 106 for assets stored on servers 110. See col. 7, lines 49-67 and Fig. 1 of Chiu.

A *prima facie* case of obviousness for a §103 rejection requires satisfaction of three basic criteria: there must be some suggestion or motivation either in the references or knowledge generally available to modify the references or combine reference teachings, a reasonable expectation of success, and the references must teach or suggest all the claim limitations. See MPEP §706.02(j). Applicants assert that the Office Action fails to teach or suggest the features recited in claims 10, 11 and 13 with Du and Chiu.

B. The Office Action also rejects claims 16-20, 22 and 25 as being allegedly obvious and thereby unpatentable under 35 U.S.C. §103(a) over Du in view of U.S. Patent 6,742,020 to Dimitroff *et al.* (hereinafter "Dimitroff"). The Office Action additionally rejects claims 23 and 24 as being allegedly obvious and thereby unpatentable under 35 U.S.C. §103(a) over Du in view of Dimitroff and further in view of Chiu. These rejections are rendered moot by the incorporation of allowable subject matter of claim 21 into claim 16. This reason applies by extension to claims 27-20 and 22-25 based on their dependence from claim 16.

For at least these reasons, Applicants respectfully request that the rejections under 35 U.S.C. §103 be withdrawn.

VIII. Conclusion

In view of the foregoing amendments and remarks, Applicants respectfully submit that this application is in condition for allowance. Favorable reconsideration and prompt allowance are earnestly solicited.

Attorney Docket No.: Navy Case 83017

Should the Examiner believe that anything further is desirable in order to place this application in even better condition for allowance, the Examiner is invited to contact Applicants' undersigned representative at the telephone number listed below.

Respectfully submitted,

Gerhard W. Thielman
Registration No. 43,186

Attachments:
Substitute Abstract
Replacement Drawing Sheet (Fig. 2A)

Date: April 19, 2006

DEPARTMENT OF THE NAVY
Naval Surface Warfare Center – Dahlgren Division
Office of Counsel – Code XDC1
17320 Dahlgren Road
Dahlgren, Virginia 22448-5100
Telephone: (540) 653-8061 **Customer No. 23501**

DEPOSIT ACCOUNT USE AUTHORIZATION
Please grant any extension necessary for entry;
Charge any fee due to our Deposit Account No. 50-0967

OIPE IAP41
APR 26 2006
PATENT & TRADEMARK OFFICE

SUBSTITUTE ABSTRACT

A resource manager for a distributed environment including hosts instantiating copies of a scalable application, generates signals which start up, shutdown or move a selected one of the copies responsive to first information regarding performance of all copies of the scalable application and second information regarding performance of the hosts.

OIPE IAP46
APR 26 2006
PATENT & TRADEMARK OFFICE

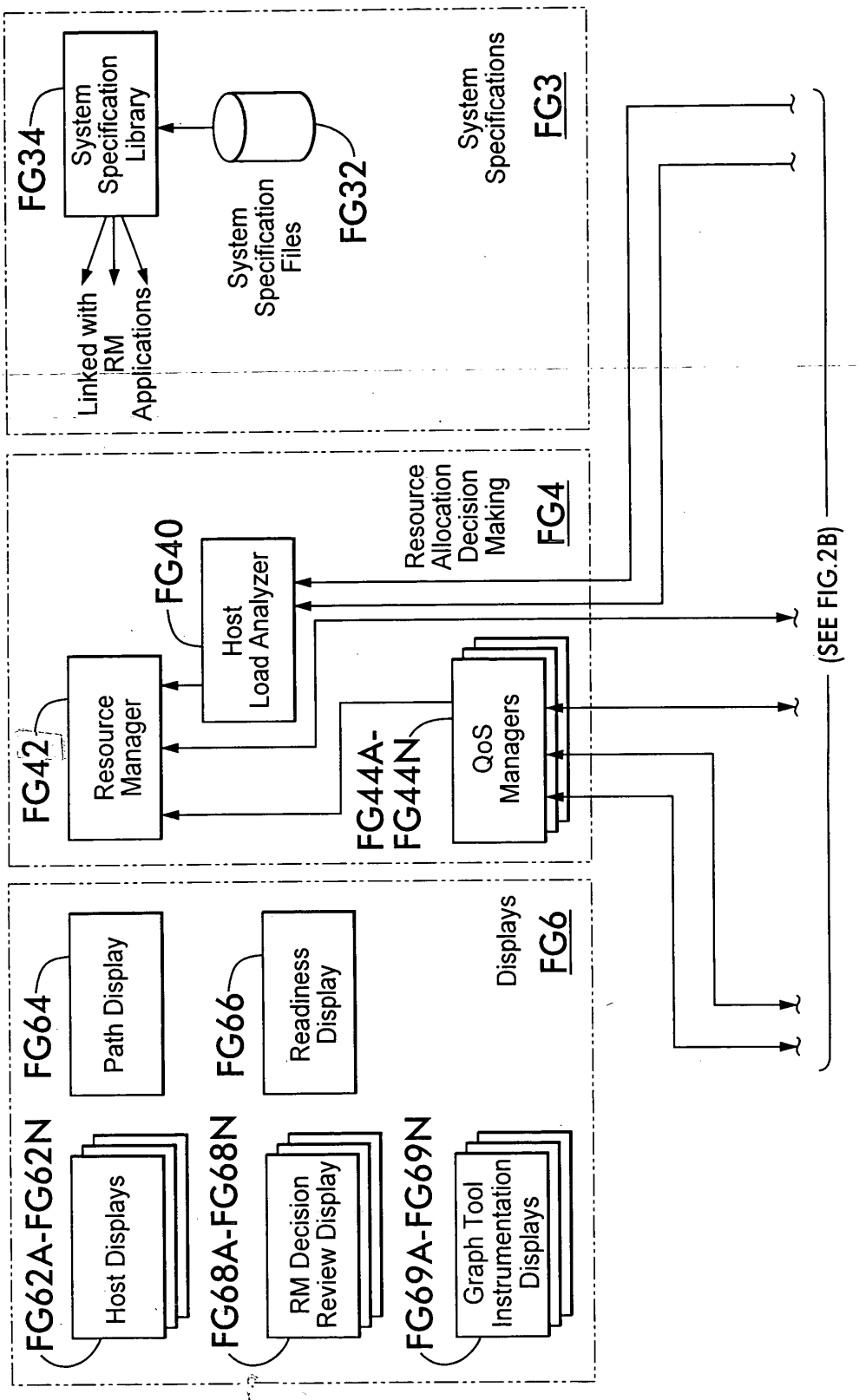


FIG.2A

OIPE IAP4C
APR 26 2006
PATENT & TRADEMARK OFFICE

IFW 2195

PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission: 21

Application Number	09/864,825
Filing Date	May 24, 2001
First Named Inventor	Paul V. Werme et al.
Art Unit	2195
Examiner Name	Kenneth Tang
Attorney Docket Number	Navy Case 83017

ENCLOSURES *(Check all that apply)*

- ☐ Fee Transmittal Form
  - ☐ Fee Attached
- ☑ Amendment/Reply
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Reply to Missing Parts/ Incomplete Application
  - ☐ Reply to Missing Parts under 37 CFR 1.52 or 1.53
- ☑ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) _____
  - ☐ Landscape Table on CD
- ☐ After Allowance Communication to TC
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- ☐ Proprietary Information
- ☐ Status Letter
- ☑ Other Enclosure(s) (please Identify below):
  Substitute Abstract

Remarks

Response to January 25, 2006 Office Action

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Naval Surface Warfare Center - Dahlgren Division		
Signature			
Printed name	Gerhard W. Thielman		
Date	April 19, 2006	Reg. No.	43,186

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature	Deborah G. Blakley		
Typed or printed name	Deborah G. Blakley	Date	April 20, 2006

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2000

Application or Docket Number: 09/864,825
NAV #D917

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	1	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	1 minus 20=	* 0
INDEPENDENT CLAIMS	1 minus 3 =	* 0
MULTIPLE DEPENDENT CLAIM PRESENT		☐

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
BASIC FEE	355.00	OR	BASIC FEE	710.00
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL		OR	TOTAL	710

CLAIMS AS AMENDED - PART II

10/5/4

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 35	Minus	** 36	= 0
Independent	* 3	Minus	*** 3	= 0
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=	X	OR	X80=	✓
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

5/25/5

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 35	Minus	** 36	= 0
Independent	* 3	Minus	*** 3	= 0
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

14-26-6

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 28	Minus	** 36	= 0
Independent	* 3	Minus	*** 3	= 0
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

FORM PTO-875 (Rev. 8/00)

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
*U.S. GPO: 2000-480-708/60169

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/864,825	05/24/2001	Paul V. Werme	NC 83017	2372

23501 7590 01/25/2006

NAVAL SURFACE WARFARE CENTER, DAHLGREN DIVISION
OFFICE OF COUNSEL, CODE XDC1
17320 DAHLGREN ROAD
DAHLGREN, VA 22448-5110

EXAMINER	
TANG, KENNETH	

ART UNIT	PAPER NUMBER
2195	

DATE MAILED: 01/25/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)	
	09/864,825	WERME ET AL.	
	Examiner	Art Unit	
	Kenneth Tang	2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 May 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-36* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *26-35* is/are allowed.

6)☒ Claim(s) *1-20,22-25 and 36* is/are rejected.

7)☒ Claim(s) *21* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the Amendment filed on 5/25/05. Applicant's arguments have been fully considered but are moot in view of the new grounds of rejections.

2. Claims 1-36 are presented for examination.

Specification

3. The abstract of the disclosure is objected to because "reseource" is misspelled on line 1. Correction is required. See MPEP § 608.01(b).

Claim Objections

4. Claim 36 is objected to because of the following informalities: "inter dependencies" should be amended to "inter-dependencies". Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claim 36 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

a. In claim 36, lines 4-6, "determining whether and where additional application computer programs should to be one of.." does not make any sense because it is grammatically incorrect and therefore, indefinite.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. **Claims 1-3, 6-9, 12, and 14-15 are rejected under 35 U.S.C. 102(b) as being unpatentable over Du et al. (hereinafter Du) (US 6,041,306).**

7. As to claim 1, Du teaches in a distributed environment comprised of hosts *(network node 12a, etc.)* instantiating copies *(creating instances)* of an application computer program *(col. 2, lines 59-65, col. 5, lines 59-67 through col. 6, lines 1-10)*, a resource management device within one of the hosts of the distributed environment *(see Fig. 2)*, the resource management device generating signals which start up *(started or restart)*, shutdown *(stopped)* or move *(controlled or data exchange or communication, etc.) (col. 5, lines 59-67, col. 1, line 49)* a selected one of the copies *(instance)* responsive to first information received by the resource management device regarding performance *(monitor and measure progress of the process instances created) (col. 8, lines 3-6, col. 10, lines 1-4)* of one or more copies *(instances)* of the application computer program *(col. 2, lines 59-65, col. 5, lines 59-67 through col. 6, lines 1-10)* and second information received by the resource management device regarding performance of the hosts *(system monitoring and network communication, etc.) (col. 2, lines 7-10)*. The central processing unit (CPU) of the computer 12a generates the signals or "firing" *(col. 7, line 20, etc.)*.

8. As to claim 2, Du teaches on a host instantiating a managed characteristic computer program *(attribute, state, instances) (col. 2, lines 59-65, col. 5, lines 59-67 through col. 6, lines 1-10)*, the managed characteristic application computer program *(col. 2, lines 59-65, col. 5, lines 59-67 through col. 6, lines 1-10)* being managed by at least the host, a resource management device of the host, the resource management device *(see Fig. 2)* generating signals responsive to first information received by the resource management device regarding performance *(monitor and measure progress of the process instances created) (col. 8, lines 3-6, col. 10, lines 1-4)* of a plurality of application computer programs including the managed characteristic application computer program and second information received by the resource management device regarding performance of the host *(system monitoring and network communication, etc.) (col. 2, lines 7-10)*, the signals including:

a first signal that starts up an additional copy (instance) of the managed characteristic (attribute or state) application computer program on one of the host and a second networked host (database, etc) *(col. Fig. 2, 21, 34a, or 34b, etc.)*;

a second signal shuts down *(stopped)* and restarts (restart or reset) *(col. 7, lines 17-24, 56-57)* the managed characteristic on the host; and

a third signal that moves the managed characteristic to the second host (database, etc.) *(col. Fig. 2, 12a, 20, 23b, 26, 27, 21, 34a, or 34b, etc.)*.

9. As to claim 3, Du teaches wherein the managed characteristic application computer program comprises a scalable application program *(col. 5, lines 16-34)*.

10. As to claim 6, Du teaches the resource management device wherein the managed characteristic application computer program further responds to user-initiated control actions *(users respond and interactive through the user interface, see Fig. 3, etc.)*.

11. As to claim 7, Du teaches the resource management device wherein the resource arrangement device generates signals instructing a program control device to modify the configuration of the managed characteristic application computer program *(configuration management) (col. 10, lines 5-27, etc)*.

12. As to claim 8, it is rejected for the same reasons as stated in the rejection of claim 2. An M+1 copy is taught in Du where it is disclosed that there is scalability *(col. 5, lines 16-34, etc)*. If something is scalable, it can do the same for an extra/additional or an M+1 amount, etc. Un addition, Du teaches configuration management by of one of the hosts *(col. 10, lines 5-28)*.

13. As to claim 9, it is rejected for the same reasons as stated in the rejection of claim 3.

14. As to claim 12, Du teaches the resource management device wherein the resource management device further generates signals responsive to third information received by the resource management device regarding the performance of hardware operatively coupling the networked hosts *(col. 8, lines 3-6, col. 10, lines 1-4,col. 2, lines 7-10)*.

15. As to claims 14, it is rejected for the same reasons as stated in the rejection of claims 6, respectively.

16. As to claim 15, Du teaches where the action requests are generated by an operator *(col. 4, lines 9-33)*.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

17. Claims 4-5, 10-11 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Du et al. (hereinafter Du) (US 6,041,306), and further in view of Chiu et al. (hereinafter Chiu) (US 6,654,029 B1).

18. As to claims 4-5, Du teaches the resource management device wherein the managed characteristic application computer program comprises a fault tolerant application *(col. 4, lines 58-67, col. 7, lines 55-62)*. Du fails to explicitly teach where the degree of fault tolerance is selectable by a user by priority. However, Chiu teaches a scalable, object-oriented architecture with a user selecting the priority (importance) level of fault tolerance *(col. 5, lines 20-25)*. It

would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Du and Chiu because the user could react based on importance *(col. 5, lines 20-25)*.

19. As to claims 10-11 and 13, they are rejected for the same reasons as stated in the rejection of claims 4-5, respectively.

20. **Claims 16-20 and 22, and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Du et al. (hereinafter Du) (US 6,041,306), and further in view of Dimitroff et al. (hereinafter Dimitroff) (US 6,742,020 B1).**

As to claim 16, it is rejected for the same reasons as stated in the rejection of claim 8. In addition, Du teaches using managers and state machines that contains states of every condition and determines the state and health based on the conditions automatically *(col. 6, lines 65-67 through col. 7, lines 1-15 and col. 12, lines 31-47 and col. 10, line 3.* In addition, Du teaches using QoS requirements for the applications *(col. 11, lines 16-18)* and shutting down, restarting and moving managed characteristic application programs (see rejection of claim 2). Du fail to explicitly teach that the computer programs are moved, shutdown and started in accordance with satisfaction of the QoS requirements. However, Dimitroff teaches using (opening, closing, moving, etc.) application programs in accordance to satisfaction of QoS requirements *(col. 4, lines 7-18)*. It would have been obvious to one of ordinary skill in the art at the time the

invention was made to include the feature of the computer programs are moved, shutdown and started in accordance with satisfaction of the QoS requirements to the existing networking system with QoS requirements of Du because ensuring that QOS requirements are met insofar as possible is desirable *(col.1, lines 57-65)*.

21. As to claim 17, Du teaches the software wherein the first function receives system specification information comprising selected ones of host configuration and capabilities, application capabilities, survivability requirements, scalability characteristics, application startup and shutdown dependencies, and application and path performance requirements *(col. 10, lines 5-28, col. 1, lines 56-67, col. 8, lines 53-56, col. 5, lines 48-67, col. 7, lines 17-24, 56-57, etc.)*.

22. As to claim 18, Du teaches wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic application computer programs, and detected failures regarding the N hosts *(col. 15, lines 3-5, col. 16, lines 33-52)*.

23. As to claim 19, Du teaches wherein the first function receives application performance data representing each of the M managed characteristic application computer programs *(col. 1, liens 56-67, col. 8, lines 53-56, col. 5, lines 48-67, etc.)*.

24. As to claim 20, Du teaches wherein the first function receives application performance data on one or more applications instantiated by the N hosts including performance data

representing each one of the M managed characteristic application computer programs *(col. 1, liens 56-67, col. 8, lines 53-56, col. 5, lines 48-67, etc.)*.

25. As to claim 22, Du teaches the resource management device wherein the managed characteristic application computer programs comprises a scalable application *(col. 5, line 34, etc.)*.

26. As to claim 25, Du teaches wherein the M managed characteristic application computer programs comprise M copies of a single managed characteristic application computer program *(col. 2, lines 59-65, col. 5, lines 59-67 through col. 6, lines 1-10)*.

27. Claims 23-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Du et al. (hereinafter Du) (US 6,041,306), and further in view of Dimitroff et al. (hereinafter Dimitroff) (US 6,742,020 B1), and further in view of Chiu et al. (hereinafter Chiu) (US 6,654,029 B1).

28. As to claims 23-24, Du teaches the resource management device wherein the managed characteristic application computer program comprises a fault tolerant application *(col. 4, lines 58-67, col. 7, lines 55-62)*. Du and Dimitroff fail to explicitly teach where the degree of fault tolerance is selectable by a user by priority. However, Chiu teaches a scalable, object-oriented architecture with a user selecting the priority (importance) level of fault tolerance *(col. 5, lines*

20-25). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Du, Dimitroff, and Chiu because the user could react based on importance *(col. 5, lines 20-25)*.

Allowable Subject Matter

29. Claims 26-35 are allowed.

30. Claim 36 would be allowable if rewritten to overcome the rejection(s) under 35 U.S.C. 112, 2nd paragraph, set forth in this Office action.

31. Claim 21 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Response to Arguments

32. Applicant's arguments have been fully considered but are moot in view of the new grounds of rejections.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kenneth Tang whose telephone number is (571) 272-3772. The examiner can normally be reached on 8:30AM - 6:00PM, Every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

1/11/05
kt

MENG-AI T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Notice of References Cited	Application/Control No. 09/864,825	Applicant(s)/Patent Under Reexamination WERME ET AL.	
	Examiner Kenneth Tang	Art Unit 2195	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,654,029 B1	11-2003	Chiu et al.	715/717
	B	US-			
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Search Notes	Application No.	Applicant(s)	
	09/864,825	WERME ET AL.	
	Examiner	Art Unit	
	Kenneth Tang	2127	

SEARCHED			
Class	Subclass	Date	Examiner
718	100-108	6/27/2004	KT
705	8	6/27/2004	KT
370	217	6/27/2004	KT

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
EAST	6/27/2004	KT
EAST	2/15/2005	KT
EAST, text search	1/22/2006	KT

Index of Claims

Application No.	Applicant(s)
09/864,825	WERME ET AL.
Examiner	**Art Unit**
Kenneth Tang	2127

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim		Date		
Final	Original		2/15/05	1/22/06
	1	√	√	√
	2	√	√	√
	3	√	√	√
	4	√	√	√
	5	√	√	√
	6	√	√	√
	7	√	√	√
	8	√	√	√
	9	√	√	√
	10	√	√	√
	11	√	√	√
	12	√	√	√
	13	√	√	√
	14	√	√	√
	15	√	√	√
	16	√	√	√
	17	√	√	√
	18	√	√	√
	19	√	√	√
	20	√	√	√
	21	√	√	o
	22	√	√	√
	23	√	√	√
	24	√	√	√
	25	√	√	√
	26	√	√	=
	27	√	√	=
	28	√	√	=
	29	√	√	=
	30	√	√	=
	31	√	√	=
	32	√	√	=
	33	√	√	=
	34	√	√	=
	35	√	√	=
	36	√	√	√
	37			
	38			
	39			
	40			
	41			
	42			
	43			
	44			
	45			
	46			
	47			
	48			
	49			
	50			

Claim		Date
Final	Original	
	51	
	52	
	53	
	54	
	55	
	56	
	57	
	58	
	59	
	60	
	61	
	62	
	63	
	64	
	65	
	66	
	67	
	68	
	69	
	70	
	71	
	72	
	73	
	74	
	75	
	76	
	77	
	78	
	79	
	80	
	81	
	82	
	83	
	84	
	85	
	86	
	87	
	88	
	89	
	90	
	91	
	92	
	93	
	94	
	95	
	96	
	97	
	98	
	99	
	100	

Claim		Date
Final	Original	
	101	
	102	
	103	
	104	
	105	
	106	
	107	
	108	
	109	
	110	
	111	
	112	
	113	
	114	
	115	
	116	
	117	
	118	
	119	
	120	
	121	
	122	
	123	
	124	
	125	
	126	
	127	
	128	
	129	
	130	
	131	
	132	
	133	
	134	
	135	
	136	
	137	
	138	
	139	
	140	
	141	
	142	
	143	
	144	
	145	
	146	
	147	
	148	
	149	
	150	

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	588817	"1" and (fault fail$6)	USPAT	OR	ON	2006/01/22 10:32
L2	1	(US-6041306-$).did.	USPAT	OR	OFF	2006/01/22 10:32
L3	1	2 and (fault fail$6)	USPAT	OR	ON	2006/01/22 10:32
L4	1513	scalable and (resource adj manag$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L5	1457	scalable and (resource adj manag$5) and network$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L6	689	@ad<="20010524" and (scalable and (resource adj manag$5) and network$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L7	243	(start$3 same (shut$3 ends) same restart$3) with (application program)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L8	0	(scalable and (resource adj manag$5)) and @ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L9	0	(scalable and (resource adj manag$5) and network$3) and @ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L10	172	@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L11	0	(scalable and (resource adj manag$5)) and (@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56

L12	31	(start$3 same (shut$3 ends) same restart$3) with (copy instance)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L13	216	resumption with (copy instance)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L14	713	@ad<="20010524" and (scalable and (resource adj manag$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L15	0	(scalable and (resource adj manag$5)) and (resumption with (copy instance))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L16	151	@ad<="20010524" and (resumption with (copy instance))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L17	0	((US-20020064126-$).did.) and scalable	US-PGPUB; USPAT	OR	ON	2006/01/22 10:56
L18	0	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020064126-$).did.) and scalable	US-PGPUB; USPAT	OR	ON	2006/01/22 10:56
L19	153	scalable adj application	US-PGPUB; USPAT	OR	ON	2006/01/22 10:56
L20	20776748	@ad<="20010524"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L21	0	((US-20020064126-$).did.) and performance	US-PGPUB	OR	ON	2006/01/22 10:56
L22	9	(@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))) and (resource adj manag$5) and network$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L23	24	@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (copy instance))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56

L24	41	(@ad<="20010524" and (resumption with (copy instance))) and network	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L25	30	((starts and resum$) with (copy instance)) same (manager controller)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L26	25	@ad<="20010524" and (((starts and resum$) with (copy instance)) same (manager controller))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L27	1	("20020064126").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2006/01/22 10:56
L28	3	quality adj of adj service	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L29	4	(US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020064126-$).did.	US-PGPUB; USPAT	OR	OFF	2006/01/22 10:56
L30	75	(scalable adj application) and @ad<="20010524"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L31	1	((US-20020174420-$).did.) and performance	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L32	1	((US-20020064126-$).did.) and perform$	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L33	1	((US-20020064126-$).did.) and application	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L34	5	(US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.	US-PGPUB; USPAT	OR	OFF	2006/01/22 10:56

L35	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and monitor$3	USPAT	OR	ON	2006/01/22 10:56
L36	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and monitor$3 and (status state)	USPAT	OR	ON	2006/01/22 10:56
L37	1	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and qos	USPAT	OR	ON	2006/01/22 10:56
L38	1	((US-20020174420-$).did.) and states	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L39	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and automat$	USPAT	OR	ON	2006/01/22 10:56
L40	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and scal$	USPAT	OR	ON	2006/01/22 10:56
L41	1	("20030167270").PN.	US-PGPUB; USPAT	OR	OFF	2006/01/22 10:56
L42	70	(resource adj manag$3) with scal$6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L44	35	L42 and L43	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L45	29	L44 and (instance instantiat$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56

L46	30	L44 and (instance instantiat$4 image)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L47	30	L44 and (instance instantiat$4 image copy)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L48	131	(resource adj (manager management)) with scal$6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L49	60	L43 and L48	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L50	31	L49 and qos	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L51	31	L50 and L43	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L52	31	L51 and (host network server client)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L53	19132534	@ad<="20000425"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L54	131	(resource adj (manager management)) with scal$6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56

L55	60	L53 and L54	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L56	31	L55 and qos	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L57	31	L56 and L53	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L58	31	L57 and (host network server client)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L59	30	L58 and (application program)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L60	28	L58 and signal	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L61	1	(US-5826239-$).did.	USPAT	OR	OFF	2006/01/22 10:56
L62	1	L61 and (configur$4 configuration)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L63	92843	"9" and ((application program) with instance)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L64	1	(US-5826239-$).did.	USPAT	OR	OFF	2006/01/22 10:56

L65	0	L64 and ((application program) with instance)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L66	1	L64 and ((application program) same instance)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L67	0	L64 and (mov$3 transfer$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L68	1	L64 and (communicat$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L69	1	L64 and attribute	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L70	1	L64 and restart$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L71	1	(US-20030167270-$).did.	US-PGPUB	OR	OFF	2006/01/22 10:56
L72	0	L71 and characteristic	US-PGPUB; USPAT	OR	OFF	2006/01/22 10:56
L73	1	L71 and characteristic	US-PGPUB; USPAT	OR	ON	2006/01/22 10:56
L74	1	L71 and fault	US-PGPUB; USPAT	OR	ON	2006/01/22 10:56
L75	0	L64 and fault	US-PGPUB; USPAT	OR	ON	2006/01/22 10:56
L76	1	L64 and fail$4	US-PGPUB; USPAT	OR	ON	2006/01/22 10:56
L77	0	L64 and (weigh$4 priority rank$4)	US-PGPUB; USPAT	OR	ON	2006/01/22 10:56

L78	1	L64 and user	US-PGPUB; USPAT	OR	ON	2006/01/22 10:56
L80	1	(US-6041306-$).did.	USPAT	OR	OFF	2006/01/22 10:56
L81	1	L80 and scalab$6	USPAT	OR	OFF	2006/01/22 10:56
L82	1	L80 and table	USPAT	OR	OFF	2006/01/22 10:56
L83	1	L80 and configuration	USPAT	OR	ON	2006/01/22 10:56
L84	1	L80 and instance	USPAT	OR	ON	2006/01/22 10:56
L85	1	L80 and (instance with (transfer$4 sending sent send transmit$3 network host client server))	USPAT	OR	ON	2006/01/22 10:56
L86	1	L80 and ((instance attribute) with (transfer$4 sending sent send transmit$3 network host client server))	USPAT	OR	ON	2006/01/22 10:56
L87	478	((instance copy image) near3 program) with (transfer$4 sending sent send transmit$3) with (network host client server)	USPAT	OR	ON	2006/01/22 10:56
L88	402	@ad<="20010524" and L87	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L89	68	L88 and ((application program) with priority)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L90	9	L89 and ((scalable scalability) with (application program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L91	1	L88 and (user with (fault adj toleran$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L92	1	(((characteristic attribute) near3 (program application)) with priority) and (user with (fault adj toleran$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L93	69	(((characteristic attribute) near3 (program application))) and (user with (fault adj toleran$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56

L94	60	@ad<="20010524" and L93	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L95	625	(((characteristic attribute) with (program application))) with priority	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L96	3	L94 and L95	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L97	46	(resource adj allocation) and (user with (fault adj toleran$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L98	29	@ad<="20010524" and L97	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L99	9	(resource adj allocation) and (user with (fault adj toleran$4) with (select$4 control$3 manag$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L100	2	@ad<="20010524" and L99	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L101	98	@ad<="20010524" and (user with (fault adj toleran$4) with (select$4 control$3 manag$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L102	12	@ad<="20010524" and (user with (fault adj toleran$4) with (select$4 control$3 manag$4) with (level degree))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L103	1	(US-6654029-$).did.	USPAT	OR	OFF	2006/01/22 10:56
L104	1	L103 and fault	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56

L105	1	L103 and level	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L106	1	L103 and user	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/01/22 10:56
L107	1	L80 and (device with user)	USPAT	OR	ON	2006/01/22 10:56

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/864,825	05/24/2001	Paul V. Werme	NC 83017	2372

23501 7590 11/14/2005

NAVAL SURFACE WARFARE CENTER, DAHLGREN DIVISION
OFFICE OF COUNSEL, CODE XDC1
17320 DAHLGREN ROAD
DAHLGREN, VA 22448-5110

EXAMINER	
ART UNIT	PAPER NUMBER

DATE MAILED: 11/14/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450
www.uspto.gov

Paper No.

Letter Withdrawing a Notice of Non-Compliant Amendment

The Notice of Non-Compliant Amendment mailed on 6/7/05 was sent in error, and is hereby withdrawn. The application is being forwarded to the examiner for appropriate action. (Note: this letter does not apply to any Notice of Non-Compliant Amendment where the amendment was a reply to a final Office action.)

Legal Instruments Examiner (LIE)

571-272-3580
Telephone No.

Previous notice has been withdrawn.
Claim numbering has been updated
per rule 1.126. See attached.

Rev. 6/04

OIPE JC130 JUL 08 2005 PATENT & TRADEMARK OFFICE

2127 DW

PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission	4

Application Number	09/864,825
Filing Date	24 May 2001
First Named Inventor	PAUL V. WERME
Art Unit	2127
Examiner Name	KENNETH TANG
Attorney Docket Number	83017

ENCLOSURES *(Check all that apply)*

- ☐ Fee Transmittal Form
  - ☐ Fee Attached
- ☐ Amendment/Reply
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Reply to Missing Parts/ Incomplete Application
  - ☐ Reply to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) _____
  - ☐ Landscape Table on CD
- ☐ After Allowance Communication to TC
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- ☐ Proprietary Information
- ☐ Status Letter
- ☒ Other Enclosure(s) (please Identify below):

RESPONSE TO NOTICE OF NON-COMPLIANT AMENDMENT

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	NAVAL SURFACE WARFARE CENTER DAHLGREN DIVISION		
Signature			
Printed name	SCOTT R. BOALICK		
Date	July 6, 2005	Reg. No.	42,337

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature	Deborah G. Blakley		
Typed or printed name	DEBORAH G. BLAKLEY	Date	July 6, 2005

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

OIPE JC130 JUL 08 2005 PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Applicant: Paul V. Werme	
Application No.: 09/864,825	Group Art Unit: 2127
Filed: 5/24/2001	Examiner: Kenneth Tang
Title: Resource allocation decision function for resource management architecture and corresponding programs therefor	
Attorney Docket No.: 83017	

Mail Stop RCE
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

RESPONSE TO NOTICE OF NON-COMPLIANT AMENDMENT

In response to the Notice of Non-Compliant Amendment mailed June 7, 2005 for the Request for Continuing Examination filed May 25, 2005, please consider the following remarks.

REMARKS

The Notice of Non-Compliant Amendment mailed June 7, 2005, stated that "Claim 18 can't be both cancelled and presented?" In reply, Applicant notes that a typographical error caused two claims labeled as claim 18 to be presented in the application simultaneously. The second one of the redundantly labeled claims was later cancelled by amendment. Only the first claim labeled as claim 18 is now pending. For consistency, Applicant has continued to list the currently pending claim 18 along with the cancelled redundant claim 18 in the listing of claims.

In particular, two claims labeled as "Claim 18" were added at pages 5 and 6 of the preliminary amendment filed September 19, 2001. This redundant claim numbering was a typographical error. In the Office Action mailed July 2, 2004, at page 5, the Examiner stated that "As to claim 18, there are two different claims for claim 18 (see page 5 and 6). It is not known which of the two claims represents claim 18." Thereafter, in the Amendment and Response to Office Action filed October 5, 2004, the second claim labeled as claim 18 was canceled and, at page 13, Applicant stated that "a second claim 18 has been canceled by this response." Again, for consistency, both of the redundantly labeled claims appear in the listing of claims. If the Examiner desires a change to the listing of claims, the Examiner is invited to contact the attorney listed below.

Applicant has made a diligent effort to place the pending claims in condition for allowance, and request that they so be allowed. For these reasons, this application is now considered to be in condition for allowance and such action is earnestly solicited. Please apply any charges or credits to deposit account 05-0967.

Respectfully Submitted,

July 6, 2005
Date

SCOTT R. BOALICK
Registration No. 42,337
Attorney/Agent for Applicant(s)

Office of Counsel, Code XDC1
Naval Surface Warfare Center
Dahlgren Division
17320 Dahlgren Road
Dahlgren, VA 22448-5110
Phone: (540) 653-8061
Fax: (540) 653-8879

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/864,825	05/24/2001	Paul V. Werme	NC 83017	2372

23501 7590 06/07/2005

NAVAL SURFACE WARFARE CENTER, DAHLGREN DIVISION
OFFICE OF COUNSEL, CODE XDC1
17320 DAHLGREN ROAD
DAHLGREN, VA 22448-5110

EXAMINER	
ART UNIT	PAPER NUMBER

DATE MAILED: 06/07/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450
www.uspto.gov

Paper No.

Notice of Non-Compliant Amendment (37 CFR 1.121)

The amendment document filed on May 25, 2005 is considered non-compliant because it has failed to meet the requirements of 37 CFR 1.121, as amended on June 30, 2003 (see *68 Fed. Reg. 38611*, Jun. 30, 2003). In order for the amendment document to be compliant, correction of the following item(s) is required. **Only the corrected section of the non-compliant amendment document must be resubmitted (in its entirety), e.g., the entire "Amendments to the claims" section of applicant's amendment document must be re-submitted.** 37 CFR 1.121(h).

THE FOLLOWING CHECKED (X) ITEM(S) CAUSE THE AMENDMENT DOCUMENT TO BE NON-COMPLIANT:

☐ 1. Amendments to the specification:
- ☐ A. Amended paragraph(s) do not include markings.
- ☐ B. New paragraph(s) should not be underlined.
- ☐ C. Other_____

☐ 2. Abstract:
- ☐ A. Not presented on a separate sheet. 37 CFR 1.72.
- ☐ B. Other_____

☐ 3. Amendments to the drawings: _____

☒ 4. Amendments to the claims:
- ☐ A. A complete listing of all of the claims is not present.
- ☐ B. The listing of claims does not include the text of all claims (including withdrawn claims)
- ☐ C. Each claim has not been provided with the proper status identifier, and as such, the individual status of each claim cannot be identified.
- ☐ D. The claims of this amendment paper have not been presented in ascending numerical order.
- ☒ E. Other: Claim 18 can't be both cancelled and presented.?

For further explanation of the amendment format required by 37 CFR 1.121, see MPEP Sec. 714 and the USPTO website at http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/officeflyer.pdf.

If the non-compliant amendment is a **PRELIMINARY AMENDMENT**, applicant is given ONE MONTH from the mail date of this letter to supply the corrected section which complies with 37 CFR 1.121. Failure to comply with 37 CFR 1.121 will result in non-entry of the preliminary amendment and examination on the merits will commence without consideration of the proposed changes in the preliminary amendment(s). This notice is not an action under 35 U.S.C. 132, and **this ONE MONTH time limit is not extendable**.

If the non-compliant amendment is a reply to a **NON-FINAL OFFICE ACTION (including a submission for an RCE),** and since the amendment appears to be a *bona fide* attempt to be a reply (37 CFR 1.135(c)), applicant is given a TIME PERIOD of ONE MONTH from the mailing of this notice within which to re-submit the corrected section which complies with 37 CFR 1.121 in order to avoid abandonment. **EXTENSIONS OF THIS TIME PERIOD ARE AVAILABLE UNDER 37 CFR 1.136(a).**

If the amendment is a reply to a **FINAL REJECTION,** this form may be an attachment to an Advisory Action. **The period for response to a final rejection continues to run from the date set in the final rejection**, and is not affected by the non-compliant status of the amendment.

C. Barnes
Legal Instruments Examiner (LIE)

571-272-3588
Telephone No.

Rev. 10/03

OIPE JC130 MAY 25 2005 PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Applicant: Paul V. Werme	
Application No.: 09/864,825	Group Art Unit: 2127
Filed: 5/24/2001	Examiner: Kenneth Tang
Title: Resource allocation decision function for resource management architecture and corresponding programs therefor	
Attorney Docket No.: 83017	

Mail Stop RCE
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT

Dear Sir:

Under the present Request for Continuing Examination and in response to the Final Office Action of March 1, 2005, please amend the application as follows, and consider the following remarks.

In the claims

1. (currently amended) In a distributed environment comprised of hosts instantiating copies of a scalable application computer program, a resource management device within one of the hosts of the distributed environment, the resource management device generating signals which start up, shutdown [[or]] and move a selected one of the copies responsive to first information received by the resource management device regarding performance of one or more copies of the scalable application computer program and second information received by the resource management device regarding performance of the hosts.

2. (previously presented) On a host instantiating a managed characteristic application computer program, the managed characteristic application computer program being managed by at least the host, a resource management device of the host, the resource management device generating signals responsive to first information received by the resource management device regarding performance of a plurality of application computer programs including the managed characteristic application computer program and second information received by the resource management device regarding performance of the host, the signals including:

a first signal that starts up a copy of the managed characteristic application computer program on one of the host and a second networked host;

a second signal that shuts down and restarts the managed characteristic application computer program on the host; and

a third signal that moves the managed characteristic application computer program to the second host.

3. (previously presented) The resource management device as recited in claim 2, wherein the managed characteristic application computer program comprises a scalable application computer program.

4. (previously presented) The resource management device as recited in claim 2, wherein the managed characteristic application computer program comprises a fault tolerant application computer program, where the degree of fault tolerance is selectable by a user.

5. (previously presented) The resource management device as recited in claim 2, wherein the managed characteristic application computer program comprises a selectable priority application computer program.

6. (previously presented) The resource management device as recited in claim 2, wherein the managed characteristic application computer program further responds to user initiated control actions.

7. (previously presented) The resource management device as recited in claim 2, wherein the resource management device generates signals instructing a program control device to modifies the configuration of the managed characteristic application computer program.

8. (currently amended) In a grid system comprising networked hosts instantiating M managed characteristic application computer programs, numbered from 1 to M, a resource management device within one of the hosts generating signals responsive to first information received by the resource management device regarding performance of the M managed characteristic application computer programs and second information received by the resource management device regarding performance of the hosts, where M is a positive integer, the signals including:

a first signal that controls a configuration of an M th copy of the M managed characteristic application computer programs, each of the M managed characteristic application computer programs being managed by one of the hosts;

a second signal that controls a start up of an (M+1) th copy of the M managed characteristic application computer programs;

a third signal that controls a shutdown and restart of the M th copy of the M managed characteristic application computer programs; and

a fourth signal that controls a movement of the M th copy of the M managed characteristic application computer programs from a first to a second host.

9. (previously presented) The resource management device as recited in claim 8, wherein one of the M managed characteristic application computer programs comprises a scalable application computer program.

10. (previously presented) The resource management device as recited in claim 8, wherein one of the M managed characteristic application computer programs comprise a fault tolerant application computer program, where the degree of fault tolerance is selectable by a user.

11. (previously presented) The resource management device as recited in claim 8, wherein at least one of the M managed characteristic application computer programs comprises a selectable priority application computer program.

12. (previously presented) The resource management device as recited in claim 8, wherein the resource management device further generates signals responsive to third information received by the resource management device regarding the performance of hardware operatively coupling the networked hosts.

13. (previously presented) The resource management device as recited in claim 8, wherein the resource management device further generates signals responsive to fourth information received by the resource management device establishing one of the priority assigned to the managed characteristic application computer program and multiple copies of the managed characteristic application computer program.

14. (previously presented) The resource management device as recited in claim 8, wherein the resource management device further generates signals responsive to externally generated action requests received by the resource management device.

15. (original) The resource management device as recited in claim 14, where the action requests are generated by an operator.

16. (currently amended) In a distributed environment comprised of N hosts, numbered 1 to N, operating in a distributed environment instantiating M managed characteristic application computer programs, numbered 1 to M, managed by the N hosts, resource allocation control software instantiated by at least the N hosts, M and N each being a positive integer, where M may be equal to, less than, or greater than N, the software comprising:

a first function which determines a state and health of the N hosts, a network operatively coupling the N hosts to one another, and operatively coupling the M managed characteristic application computer programs to one another in the distributed environment, the M managed characteristic application computer programs also operatively coupled to the N hosts via the network;

a second function which determines required allocation and reallocation actions needed to maintain a plurality of Quality of Service (QoS) requirements established for the M managed characteristic application computer programs, the QoS requirements dictating parameters regarding service quality of the M management characteristic application programs; and

a third function interacting with the first and the second functions to ~~which~~ generate[[s]] automatic control signal requests corresponding to the actions dictated by the QoS requirements, such that the managed characteristic application computer programs are moved, shutdown, and started in accordance with satisfaction of the QoS requirements.

17. (original) The software as recited in claim 16, wherein the first function receives system specification information comprising selected ones of host configuration and capabilities, application capabilities, survivability requirements, scalability characteristics, application startup and shutdown dependencies, and application and path performance requirements.

18. (previously presented) The software as recited in claim 16, wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic application computer programs, and detected failures regarding the N hosts.

18. (cancelled)

19. (previously presented) The software as recited in claim 16, wherein the first function receives application performance data representing each one of the M managed characteristic application computer programs.

20. (previously presented) The software as recited in claim 16, wherein the first function receives application performance data on one or more applications instantiated by the N hosts including performance data representing each of the M managed characteristic application computer programs.

21. (previously presented) The software as recited in claim 16, wherein the second function which determines the required allocation and reallocation actions needed to maintain the Quality of Service (QoS) requirements established for the M managed characteristic applications by:

responding to application and host failures by determining if and what recovery actions should be taken;

determining if and where to place new copies of one of the M managed characteristic application computer programs or which of the M managed characteristic application computer programs should be shutdown when QoS Manager functions indicate that scale up or scale down actions are indicated based on measured application performance and application QoS specifications;

determining where new application computer programs should be placed when requested to do so by a program control device; and

determining which and how many application computer programs should run based on application system priorities.

22. (previously presented) The software as recited in claim 16, wherein at least one of the M managed characteristic application computer programs comprises a scalable application computer program.

23. (previously presented) The software as recited in claim 16, wherein at least one of the M managed characteristic application computer programs comprises a fault tolerant application computer program, where the degree of fault tolerance is selectable by a user.

24. (currently amended) The software as recited in claim 16, wherein at least one of the M managed characteristic application computer programs comprises a selectable priority application computer program.

25. (previously presented) The software as recited in claim 16, wherein the M managed characteristic application computer programs comprise M copies of a single managed characteristic application computer program.

26. (currently amended) Software stored on at least one host for converting N networked hosts, numbered 1 to N, into a resource managed system instantiating M managed characteristic application computer programs, numbered 1 to M, each managed characteristic application computer program managed by one of the N networked hosts, the software comprising:

a first function group which monitors the host and network resources;

a second function group which provides application computer program event reporting and event correlation capabilities;

a third function group interacting with the first and the second function groups and which provides reasoning and decision making capabilities for the resource managed system, wherein the third function group comprises:

a first function which determines a state and health of the N hosts, a network operatively coupling the N hosts to one another and the M managed characteristic application computer programs in the distributed environment;

a second function which determines required allocation and reallocation actions needed to maintain a plurality of Quality of Service (QoS) requirements established for the M managed characteristic application computer programs, the QoS requirements dictating parameters regarding service quality of the M management characteristic application programs; and

a third function interacting with the first and the second functions to ~~which~~ generate[[s]] automatic control signal requests corresponding to the actions dictated by the QoS requirements, such that the managed characteristic application computer programs are moved, shutdown, and started in accordance with satisfaction of the QoS requirements; and

a fourth function group which provides program control capabilities permitting starting, stopping, and configuring of selected ones of the M managed characteristic application computer programs on respective ones of the N hosts in the resource managed system,

where M and N are positive integers and where M may be equal to, greater than, or less than N.

27. (previously presented) The software as recited in claim 26, wherein the first function receives system specification information comprising host configuration and capabilities.

28. (previously presented) The software as recited in claim 26, wherein the first function receives system specification information comprising selected ones of capabilities, survivability requirements, scalability characteristics, startup and shutdown dependencies, and performance requirements for at least one of the M managed characteristic application computer programs.

29. (original) The software as recited in claim 26, wherein the first function receives system specification information comprising path performance requirements regarding communication between at least two of the N hosts.

30. (previously presented) The software as recited in claim 26, wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic application computer programs, and detected failures regarding the N hosts.

31. (original) The software as recited in claim 26, wherein the first function receives historical data regarding statuses, configuration, and loads of the N hosts and link statuses and loads regarding the network.

32. (previously presented) The software as recited in claim 26, wherein the first function receives application performance data representing each one of the M managed characteristic application computer programs.

33. (previously presented) The software as recited in claim 26, wherein the first function receives application performance data on one or more applications instantiated by the N hosts including performance data representing each of the N copies of the managed characteristic application computer programs.

34. (previously presented) The software as recited in claim 26, wherein the second function which determines the required allocation and reallocation actions established for the M managed characteristic application computer programs by:

responding to application and host failures by determining if and what recovery actions should be taken;

determining if and where to place new copies of managed characteristic application computer programs or which managed characteristic application computer programs should be shutdown when QoS Manager functions indicate that scale up or scale down actions should be taken based on measured application performance and QoS specifications established for the M managed characteristic application computer programs;

determining where new application computer programs should be placed when requested to do so by the fourth function group; and

determining which and how many application computer programs should run based on application system priorities.

35. (previously presented) The software as recited in claim 26, wherein the third function group makes decisions by one of:

based on requests from one or more of the hosts, determining where new application computer programs should be started;

based on indication of application failure from the fourth function group, determining whether and where a failed application computer program should be restarted;

based on indication of host failure from the fourth function group, determining whether and where the failed application computer program previously instantiated by the failed one of the N hosts should be restarted;

based on startup and shutdown dependency resolution requests from the fourth function group, determine whether and where additional application computer programs should to be one of started and shut down prior to starting or shutting down another application computer program; and

based on changes to application system priorities, determining whether and where new application computer programs need to be started and/or determine whether and which existing application computer programs need to be shutdown.

36. (previously presented) The software as recited in claim 26, wherein the third function group makes decisions by one of:

based on application computer program inter dependencies defined in system specification files, determining whether and where additional application computer programs should to be one of started and shut down prior to starting or shutting down of another application computer program;

based on application computer program instrumentation data generated by the second function group and performance requirements defined in the system specification files, determining whether application computer programs are meeting performance requirements and whether an application computer program can be scaled up or moved to attempt to improve performance; and

based on the application computer program instrumentation data and performance requirements defined in the system specification files, determining whether application computer programs are performing well within performance requirements and can be scaled down.

REMARKS

Claim rejections under 35 USC 112

Claims 8-36 were rejected under 35 USC 112, second paragraph, as being indefinite or incomplete. Each basis of rejection is now discussed in order (and is identified by the number and optionally the letter with which it was identified in the Final Office Action).

3.a. In claims 8, 16, and 26, "M" and "N" have been indicated as being indefinite because a definite range is not specified. Per the Examiner's suggestion, these claims have been amended to recite "numbered 1 to M" and/or "numbered 1 to N".

3.b. In claim 16, "coupling the N hosts to one another, and the M managed characteristic application computer programs" has been indicated as being indefinite because it is not clear in the claim language whether the computer programs are coupled and connected to the hosts as well as to the other computer programs, or just to the hosts. Applicant has amended claim 16 so that it recites: "a network operatively coupling the N hosts to one another, and operatively coupling the M managed characteristic application computer programs to one another . . . the M managed characteristic application computer programs also operatively coupled to the N hosts via the network". Applicant submits that this amendment overcomes the rejection.

4. In claims 16 and 26, the Examiner has indicated that there is a lack of an essential relationship between the first, second, and third functions. In particular, the Examiner indicates that the third function has to interact with the first and the second functions to perform. Applicant has amended claims 16 and 26 so that the third function is recited as "interacting with the first and the second functions." Furthermore, the Examiner indicates that there is a lack of an essential relationship between the first, second, and third function groups of claim 26. Applicant has likewise amended claim 26 so that the third function group is recited as "interacting with the

first and the second function groups." Applicant submits that the Examiner's concerns are now overcome.

Claim rejections under 35 USC 103

Claims 1, 3, and 9 have been rejected under 35 USC 103(a) as being unpatentable over Bhattal (2002/0064126) in view of Du (6,041,306), whereas claims 2, 4-8, and 10-15 have been rejected under 35 USC 103(a) as being unpatentable over Bhattal. Claims 16-36 have been rejected under 35 USC 103(a) as being unpatentable over Bhattal in view of Du, and further in view of Dimitroff (6,742,020). Claims 1, 2, 8, 16, and 26 are independent claims, from which the remaining of the claims rejected ultimately depend. Applicant asserts that the independent claims 1, 2, 8, 16, and 26 are patentable over Bhattal in view of Du, over Bhattal alone, or over Bhattal in view of Du and further in view of Dimitroff, such that all of the pending claims are patentable. There are at least three independent and separate reasons why all the independent claims are patentable, as are now discussed in detail.

First, however, it is noted that Applicant particularly discusses Bhattal as the primary reference relied upon by the Examiner in rejecting the claims under 35 USC 103(a), such that the claimed invention is not obvious over Bhattal alone, Bhattal in view of Du, or Bhattal in view of Du and further in view of Dimitroff. Furthermore, Applicant duplicates the primary portion of Bhattal relied upon by the Examiner, paragraphs [0014]-[0016] thereof, here for ease of reference.

> According to a preferred embodiment of the invention, recovery from failures of outgoing message transmissions is achieved as follows. Each queue manager in a queue sharing group has access to a shared outgoing-message queue. Each of these queue managers (or its communication manager component) is provided with a copy of a definition of a sender channel between the shared queue and a destination queue manager, such that each queue manager in the queue sharing group (or its communication manager component) is able to start an instance of the channel. Only a single channel instance is allowed to be active at any one time. Certain state information for a channel is stored whenever the channel is active, and a subset of that state information is held in shared access

storage so as to be available to any queue manager within the queue sharing group. If the queue manager which is using a channel experiences a failure, another queue manager or communication manager component in the queue sharing group uses the state information held in shared access storage together with its copy of the channel definition to start a new instance of the channel. Thus, a queue manager continues message transmission on behalf of the queue manager which experienced the failure.

In a second aspect, the invention provides a data communications system comprising: a data storage repository accessible by any one of a set of communication managers; a set of communication managers, each adapted to start an instance of a communication channel for transmitting data from the data storage repository to a remote communication manager, and each adapted to transmit data via said communication channel; a storage repository for storing current state information for the communication channel, the storage repository being accessible by any one of the set of communication managers; wherein the set of communication managers are responsive to a failure affecting a first communication manager of said set which has a first active instance of a communications channel, to start a second instance of the channel using the stored current channel state information and to resume transmission of data from the data storage repository to the remote communication manager via the second channel instance.

In a third aspect, the invention provides a computer program comprising computer readable program code for controlling the operation of a data communication apparatus on which it runs to perform the steps of a method of managing communications between a set of communication managers and a remote communication manager, the method comprising: starting a communication channel between a first communication manager of the set and the remote communication manager for transmitting data from a data storage repository to the remote communication manager, the data storage repository being accessible by any one of the set of communication managers; storing state information for the communication channel in a storage repository accessible by any one of the set of communication managers (which may be the same data storage repository from which data is transmitted); in response to a failure affecting the first communication manager, a second one of the set of communication managers using the stored channel state information to start a new channel instance and resuming transmission of data from the data storage repository to the remote communication manager via the new channel instance.

(Emphasis added)

*Bhattal does not disclose **shutting down** (a copy of) an application computer program*

Claims 1 and 2 are limited to a resource management device that generates signals, including a signal to *shut down* a copy of an application computer program, and claims 8, 16, and 26 are similarly limited to the *shutting down* of a copy of an application program. However, as recited above, Bhattal does not disclose the *shutting down* of an application computer program. Bhattal discloses *starting* an instance of a channel, and thus could be considered at best to disclose generating a signal to start an instance of a channel. The channels of Bhattal are never indicated as being shut down after being started up. Because Bhattal does not disclose shutting down a copy of an application program, the claimed invention is not disclosed within Bhattal alone or in combination with any other reference.

Indeed, there is no motivation within Bhattal to include purposefully shutting down one of its channels. Bhattal is concerned with what happens when a communication manager, and thus the channel associated with the communication manager, fails. In such instance, Bhattal says that what you can do is have a different communication manager start up a new channel, and resume communications with this new channel. Bhattal does not suggest it is ever desirable to *generate a signal to shut down* the channel (i.e., to purposefully shut down the channel), only that a channel may fail.

*Bhattal does not disclose **moving** (a copy of) an application computer program*

Claims 1 and 2 are limited to a resource management device that generates signals, including a signal to *move* a copy of an application computer program, and claims 8, 16, and 26 are similarly limited to the *moving* of a copy of an application program. However, as recited above, Bhattal does not disclose the *moving* of an application computer program. Bhattal discloses *starting* an instance of a channel only. The channels of Bhattal are never indicated as being *moved*. Because Bhattal does not disclose moving a copy of an application program, the

claimed invention is not disclosed within Bhattal alone or in combination with any other reference.

Indeed, there is no motivation within Bhattal to move one of its channels. A channel is defined between the communication managers of two hosts. The channel does not need *moving*, because if two other communication managers desire to communicate, a new channel instance is simply started, rather than moving an existing channel. Therefore, Bhattal does not suggest it is ever desirable to *move* any of its channels.

*Bhattal does not disclose starting up (a copy of) **an application computer program** based on performance-oriented information*

As noted in the previous office action response, claims 1 and 2 are limited to a resource management device that generates signals, including a signal to starting up a copy of *an application computer program*, and claims 8, 16, and 26 are similarly limited to the starting up of a copy of *an application computer program*. However, Bhattal does not disclose the starting up of *an application computer program*, but rather discloses starting an instance of a *communication channel* only. As also noted in the previous office action response, though, a channel is not an application computer program, but rather "is a one-way communication link between two queue managers." (Bhattal, para. [0034]) Thus, in the previous office action response, Applicant had argued that Bhattal discloses starting up a channel, not an application computer program as in the claimed invention.

In response, the Examiner in the Final Office Action noted that "the control of the channel is part of the application program," such that "the channel can't be started unless the application for it is started, for example." (P. 14, para. 42) However, this argument fails to consider all the limitations of the claims *as a whole*. For instance, the claims are specifically limited to the starting up of an application program *based on various information*, such as performance of all copies of the application program (claims 1, 2, and 8), quality of service requirements (claims 16 and 26), and so on. For sake of convenience only, this various

information is referred to as shorthand in this office action response as "performance-oriented information."

Bhattal, however, never discusses the starting up of *its application programs* and application queues (identified as 20, 40, 30' in FIG. 1 by the Examiner) *based on such performance-oriented information*. Rather, Bhattal only discusses the starting up of a *channel* based on performance-oriented information, such as specifically the starting up of a channel when a previous one has failed. The application programs and the application queues of Bhattal have already been started, and in turn *they* start a *channel* based on performance-oriented information. Therefore, Bhattal, either alone or in combination with one or more other references, does not disclose the claimed invention.

In other words, in considering the claimed invention relative to the prior art, Applicant emphasizes that the claimed invention must be considered as a whole. By comparison, the statement in the Final Office action suggests the following reasoning process:

1. Bhattal discloses starting up of channels based on performance-oriented information
2. The channels are created by application computer programs and application queues.
3. These programs and queues must be started up in order to start up the channels.
4. *Thus,* – Bhattal thus discloses starting up of application computer programs, as in the claimed invention.

This reasoning process is flawed because it distorts what Bhattal is actually disclosing. Bhattal's application computer programs are *never started up based on performance-oriented information*, as in the claimed invention. Therefore, even though Bhattal's application computer programs do have to be first started up before these programs start up channels based on performance-oriented information, the application computer programs *themselves* are *not* started up based on performance-oriented information, as in the claimed invention – only the *channels* are.

The reasoning process in the Final Office Action is thus flawed because it finds *A* (starting up channels) based on *B* (performance-oriented information) in Bhattal, and then improperly substitutes *C* (starting up application computer programs) for *A* (starting up channels)

to read on the claimed invention, which is *C* (starting up application computer programs) based on *B* (performance-oriented information). However, in so doing, the Examiner ignores what Bhattal is actually disclosing, which is *C* (starting up application computer programs), and [*A* (starting up channels) based on B (performance-oriented information)]. Turning "C, and [A based on B]" into "C based on B," mischaracterizes and misinterprets what Bhattal discloses.

Conclusion

Applicant has made a diligent effort to place the pending claims in condition for allowance, and request that they so be allowed. For these reasons, this application is now considered to be in condition for allowance and such action is earnestly solicited.

Respectfully Submitted,

May 23, 2005
Date

SCOTT R. BOALICK
Registration No. 42,337
Attorney/Agent for Applicant(s)

Office of Counsel, Code XDC1
Naval Surface Warfare CenterZ
Dahlgren Division
17320 Dahlgren Road
Dahlgren, VA 22448-5110
Phone: (540) 653-8061
Fax: (540) 653-8879

RCE

OIPE JC1780 MAY 25 2005 PATENT & TRADEMARK OFFICE

PTO/SB/30 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Request for Continued Examination (RCE) Transmittal

Address to:
Mail Stop RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Application Number	09/864,825
Filing Date	May 24, 2001
First Named Inventor	Paul V. Werme
Art Unit	2127
Examiner Name	Kenneth Tang
Attorney Docket Number	83017

This is a Request for Continued Examination (RCE) under 37 CFR 1.114 of the above-identified application.
Request for Continued Examination (RCE) practice under 37 CFR 1.114 does not apply to any utility or plant application filed prior to June 8, 1995, or to any design application. See Instruction Sheet for RCEs (not to be submitted to the USPTO) on page 2.

1. Submission required under 37 CFR 1.114 Note: If the RCE is proper, any previously filed unentered amendments and amendments enclosed with the RCE will be entered in the order in which they were filed unless applicant instructs otherwise. If applicant does not wish to have any previously filed unentered amendment(s) entered, applicant must request non-entry of such amendment(s).

   a. ☐ Previously submitted. If a final Office action is outstanding, any amendments filed after the final Office action may be considered as a submission even if this box is not checked.

      i. ☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on _____

      ii. ☐ Other _____

   b. ☑ Enclosed

      i. ☑ Amendment/Reply

      ii. ☐ Affidavit(s)/ Declaration(s)

      iii. ☐ Information Disclosure Statement (IDS)

      iv. ☐ Other _____

2. Miscellaneous

   a. ☐ Suspension of action on the above-identified application is requested under 37 CFR 1.103(c) for a period of _____ months. (Period of suspension shall not exceed 3 months; Fee under 37 CFR 1.17(i) required)

   b. ☐ Other _____

3. Fees The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.

   a. ☑ The Director is hereby authorized to charge the following fees, or credit any overpayments, to Deposit Account No. 50-0967. I have enclosed a duplicate copy of this sheet.

      i. ☑ RCE fee required under 37 CFR 1.17(e)

      ii. ☐ Extension of time fee (37 CFR 1.136 and 1.17)

      iii. ☑ Other Any other charges or credits

   05/25/2005 CCHAU1 00000045 500967 09864825
   01 FC:1801 790.00 DA

   b. ☐ Check in the amount of $ _____ enclosed

   c. ☐ Payment by credit card (Form PTO-2038 enclosed)

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

Signature		Date	May 23, 2005
Name (Print/Type)	SCOTT R. BOALICK	Registration No.	42,337

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop RCE, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450 or facsimile transmitted to the U.S. Patent and Trademark Office on the date shown below.

Signature	Deborah G. Blakley		
Name (Print/Type)	DEBORAH G. BLAKLEY	Date	May 23 2005

This collection of information is required by 37 CFR 1.114. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Mail Stop RCE, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

OIPE JC190 MAY 25 2005 PATENT & TRADEMARK OFFICE

PTO/SB/17 (12-04)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Effective on 12/08/2004.
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL
For FY 2005

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 790.00

Complete if Known	
Application Number	09/864,825
Filing Date	May 24, 2001
First Named Inventor	Paul V. Werme
Examiner Name	Kenneth Tang
Art Unit	2127
Attorney Docket No.	83017

METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): _____

☒ Deposit Account Deposit Account Number: 50-0967 Deposit Account Name: NSWC Dahlgren

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☒ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, **except for the filing fee**

☒ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☒ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

	FILING FEES		SEARCH FEES		EXAMINATION FEES		
		Small Entity		Small Entity		Small Entity	
Application Type	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fees Paid ($)
Utility	300	150	500	250	200	100	
Design	200	100	100	50	130	65	
Plant	200	100	300	150	160	80	
Reissue	300	150	500	250	600	300	
Provisional	200	100	0	0	0	0	

2. EXCESS CLAIM FEES

Fee Description	Fee ($)	Small Entity Fee ($)
Each claim over 20 or, for Reissues, each claim over 20 and more than in the original patent	50	25
Each independent claim over 3 or, for Reissues, each independent claim more than in the original patent	200	100
Multiple dependent claims	360	180

Total Claims	Extra Claims	Fee ($)	Fee Paid ($)
_____ - 20 or HP =	_____	x _____	= _____

HP = highest number of total claims paid for, if greater than 20

Indep. Claims	Extra Claims	Fee ($)	Fee Paid ($)
_____ - 3 or HP =	_____	x _____	= _____

HP = highest number of independent claims paid for, if greater than 3

Multiple Dependent Claims	
Fee ($)	Fee Paid ($)
_____	_____

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets	Extra Sheets	Number of each additional 50 or fraction thereof	Fee ($)	Fee Paid ($)
_____ - 100 =	_____	/ 50 = _____ (round up to a whole number)	x _____	= _____

4. OTHER FEE(S)

	Fees Paid ($)
Non-English Specification, $130 fee (no small entity discount)	
Other: Request for Continued Examination (RCE)	790.00

SUBMITTED BY

Signature		Registration No. (Attorney/Agent) 42,337	Telephone 540-653-8061
Name (Print/Type)	SCOTT R. BOALICK		Date May 23, 2005

This collection of information is required by 37 CFR 1.136. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

OIPE JC130 MAY 25 2005 PATENT & TRADEMARK OFFICE

PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission: 22

Application Number	09/864,825
Filing Date	May 24, 2001
First Named Inventor	Paul V. Werme
Art Unit	2127
Examiner Name	Kenneth Tang
Attorney Docket Number	83017

ENCLOSURES *(Check all that apply)*

- [X] Fee Transmittal Form
- [ ] Fee Attached
- [X] Amendment/Reply
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Reply to Missing Parts/ Incomplete Application
  - [ ] Reply to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation, Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
  - [ ] Landscape Table on CD
- [ ] After Allowance Communication to TC
- [ ] Appeal Communication to Board of Appeals and Interferences
- [ ] Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- [ ] Proprietary Information
- [ ] Status Letter
- [X] Other Enclosure(s) (please identify below):

Request for Continued Examination (RCE)

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	NAVAL SURFACE WARFARE CENTER DAHLGREN DIVISION		
Signature			
Printed name	SCOTT R. BOALICK		
Date	May 23, 2005	Reg. No.	42,337

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature	Deborah G. Blakley		
Typed or printed name	DEBORAH G. BLAKLEY	Date	May 23, 2005

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

In the claims

Please amend the claims as follows:

1. (currently amended) In a distributed environment comprised of hosts instantiating copies of a scalable application computer program, a resource management device within one of the hosts of the distributed environment, the resource management device generating signals which start up, shutdown or move a selected one of the copies responsive to first information received by the resource management device regarding performance of ~~all~~ one or more copies of the scalable application computer program and second information received by the resource management device regarding performance of the hosts.

2. (currently amended) On a host instantiating a managed characteristic application computer program, the managed characteristic application computer program being managed by at least the host, a resource management device of the host, the resource management device generating signals responsive to first information received by the resource management device regarding performance of a plurality of application computer programs including the managed characteristic application computer program and second information received by the resource management device regarding performance of the host, the signals including:

a first signal that ~~which one of~~ starts up ~~an additional~~ a copy of the managed characteristic application computer program on one of the host and a second networked host[[,]];

a second signal that shuts down and restarts the managed characteristic application computer program on the host[[,]]; and

a third signal that moves the managed characteristic application computer program to the second host ~~responsive to first information regarding performance of all applications including the managed characteristic application and second information regarding performance of the host~~.

3. (currently amended) The resource management device as recited in claim 2, wherein the managed characteristic application computer program comprises a scalable application computer program.

4. (currently amended) The resource management device as recited in claim 2, wherein the managed characteristic application computer program comprises a fault tolerant application computer program, where the degree of fault tolerance is selectable by a user.

5. (currently amended) The resource management device as recited in claim 2, wherein the managed characteristic application computer program comprises a selectable priority application computer program.

6. (currently amended) The resource management device as recited in claim 2, wherein the managed characteristic application computer program further responds to user initiated control actions.

7. (currently amended) The ~~program control~~ resource management device as recited in claim 2, wherein the resource management device generates signals instructing a program control device to modifies the configuration of the managed characteristic application computer program.

8. (currently amended) In a grid system comprising networked hosts instantiating M managed characteristic application[[s]] computer programs, a resource management device within one of the hosts generating signals responsive to first information received by the resource management device regarding performance of the M managed characteristic application computer programs and second information received by the resource management device regarding performance of the hosts, where M is a positive integer, ~~which one of~~ the signals including:

a first signal that controls a configuration of an M th copy of the M managed characteristic application computer programs, each of the M managed characteristic application computer programs being managed by one of the hosts;

a second signal that controls a start up of an (M+1) th copy of the M managed characteristic application[[s]] computer programs;[[,]]

a third signal that controls a shutdown and restart of the M th copy of the M managed characteristic application[[s]] computer programs;[[,]] and

a fourth signal that controls a movement of the M th copy of the M managed characteristic application[[s]] computer programs from a first to a second host ~~responsive to first information regarding performance of the M managed characteristic applications and second information regarding performance of the hosts, where M is a positive integer~~.

9. (currently amended) The resource management device as recited in claim 8, wherein at least one of the M managed characteristic application[[s]] computer programs comprises a scalable application computer program.

10. (currently amended) The resource management device as recited in claim 8, wherein at least one of the M managed characteristic application[[s]] computer programs comprise a fault tolerant application computer program, where the degree of fault tolerance is selectable by a user.

11. (currently amended) The resource management device as recited in claim 8, wherein at least one of the M managed characteristic application[[s]] computer programs comprises a selectable priority application computer program.

12. (currently amended) The resource management device as recited in claim 8, wherein the resource management device further ~~responds~~ generates signals responsive to third information

received by the resource management device regarding the performance of hardware operatively coupling the networked hosts.

13. (currently amended) The resource management device as recited in claim 8, wherein the resource management device further ~~responds~~ generates signals responsive to fourth information received by the resource management device establishing one of the priority assigned to the managed characteristic application computer program and multiple copies of the managed characteristic application computer program.

14. (currently amended) The resource management device as recited in claim 8, wherein the resource management device further ~~responds~~ generates signals responsive to externally generated action requests received by the resource management device.

15. (original) The resource management device as recited in claim 14, where the action requests are generated by an operator.

16. (currently amended) In a distributed environment comprised of N hosts operating in a distributed environment instantiating M managed characteristic application[[s]] computer programs managed by the N hosts, resource allocation control software instantiated by at least the N hosts, M and N each being a positive integer, where M may be equal to, less than, or greater than N, the software comprising:

a first function which determines ~~the~~ a state and health of the N hosts, a network operatively coupling the N hosts to one another, and the M managed characteristic application[[s]] computer programs in the distributed environment;

a second function which determines required allocation and reallocation actions needed to maintain a plurality of Quality of Service (QoS) requirements established for the M managed

characteristic application[[s]] computer programs, the QoS requirements dictating parameters regarding service quality of the M managed characteristic application computer programs; and

a third function which generates automatic control signal requests corresponding to the actions dictated by the QoS requirements, such that the managed characteristic application computer programs are moved, shutdown, and started in accordance with satisfaction of the QoS requirements.

17. (original) The software as recited in claim 16, wherein the first function receives system specification information comprising selected ones of host configuration and capabilities, application capabilities, survivability requirements, scalability characteristics, application startup and shutdown dependencies, and application and path performance requirements.

18. (currently amended) The software as recited in claim 16, wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic application[[s]] computer programs, and detected failures regarding the N hosts.

~~18~~. (cancelled)

~~19~~. (currently amended) The software as recited in claim 16, wherein the first function receives application performance data representing each ~~one~~ of the M managed characteristic application[[s]] computer programs.

~~20~~. (currently amended) The software as recited in claim 16, wherein the first function receives application performance data on ~~all~~ one or more applications instantiated by the N hosts including performance data representing each ~~one~~ of the M managed characteristic application[[s]] computer programs.

22 21. (currently amended) The software as recited in claim 16, wherein the second function ~~which~~ determines the required allocation and reallocation actions needed to maintain the Quality of Service (QoS) requirements established for the M managed characteristic applications by:

responding to application and host failures by determining if and what recovery actions should be taken;

determining if and where to place new copies of one of the M managed characteristic application[[s]] computer programs or which of the M managed characteristic application[[s]] computer programs should be shutdown when QoS Manager functions indicate that scale up or scale down actions are indicated based on measured application performance and application QoS specifications;

determining where new application[[s]] computer programs should be placed when requested to do so by a program control device; and

determining which and how many application[[s]] computer programs should run based on application system priorities.

23 22. (currently amended) The software as recited in claim 16, wherein at least one of the M managed characteristic application[[s]] computer programs comprises a scalable application computer program.

24 23. (currently amended) The software as recited in claim 16, wherein at least one of the M managed characteristic application[[s]] computer programs comprises a fault tolerant application computer program, where the degree of fault tolerance is selectable by a user.

25 24. (currently amended) The software as recited in claim 16, wherein one of the M managed characteristic application[[s]] computer programs comprises a selectable priority application computer program.

26

~~25~~. (currently amended) The software as recited in claim 16, wherein the M managed characteristic application[[s]] computer programs comprise M copies of a single managed characteristic application computer program.

27

~~26~~. (currently amended) Software stored on at least one host for converting N networked hosts into a resource managed system instantiating M managed characteristic application[[s]] computer programs, each managed characteristic application computer program managed by one of the N networked hosts, the software comprising:

a first function group which monitors the host and network resources;

a second function group which provides ~~general-purpose~~ application computer program event reporting and event correlation capabilities;

a third function group which provides ~~the~~ reasoning and[[,]] decision making capabilities for the resource managed system, wherein the third function group comprises:

a first function which determines ~~the~~ a state and health of the N hosts, a network operatively coupling the N hosts to one another and the M managed characteristic application[[s]] computer programs in the distributed environment;

a second function which determines required allocation and reallocation actions needed to maintain a plurality of Quality of Service (QoS) requirements established for the M managed characteristic application[[s]] computer programs, the QoS requirements dictating parameters regarding service quality of the M management characteristic application programs; and

a third function which generates automatic control signal requests corresponding to the actions dictated by the QoS requirements, such that the managed characteristic application computer programs are moved, shutdown, and started in accordance with satisfaction of the QoS requirements; and

a fourth function group which provides program control capabilities permitting starting, stopping, and configuring of selected ones of the M managed characteristic application[[s]] computer programs on respective ones of the N hosts in the resource managed system,

where M and N are positive integers and where M may be equal to, greater than, or less than N.

28 27. (currently amended) The software as recited in claim 26, wherein the first function receives system specification information ~~comprises~~ comprising host configuration and capabilities.

29 28. (currently amended) The software as recited in claim 27, wherein the first function receives system specification information comprising selected ones of capabilities, survivability requirements, scalability characteristics, startup and shutdown dependencies, and performance requirements for at least one of the M managed characteristic application[[s]] computer programs.

30 29. (original) The software as recited in claim 27, wherein the first function receives system specification information comprising path performance requirements regarding communication between at least two of the N hosts.

31 30. (currently amended) The software as recited in claim 27, wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic application[[s]] computer programs, and detected failures regarding the N hosts.

32 31. (original) The software as recited in claim 27, wherein the first function receives historical data regarding statuses, configuration, and loads of the N hosts and link statuses and loads regarding the network.

33 32. (currently amended) The software as recited in claim 27 ~~26~~, wherein the first function receives application performance data representing each one of the M managed characteristic application[[s]] computer programs.

34 33. (currently amended) The software as recited in claim 27 ~~26~~, wherein the first function receives application performance data on ~~all~~ one or more applications instantiated by the N hosts including performance data representing each ~~one~~ of the N copies of the managed characteristic application computer programs.

35 34. (currently amended) The software as recited in claim 27 ~~26~~, wherein the second function which determines the required allocation and reallocation actions established for the M managed characteristic application computer programs by:

responding to application and host failures by determining if and what recovery actions should be taken;

determining if and where to place new copies of managed characteristic application[[s]] computer programs or which managed characteristic application[[s]] computer programs should be shutdown when QoS Manager functions indicate that scale up or scale down actions should be taken based on measured application performance and QoS specifications established for the M managed characteristic application[[s]] computer programs;

determining where new application[[s]] computer programs should be placed when requested to do so by the fourth function group; and

determining which and how many application[[s]] computer programs should run based on application system priorities.

36 35. (currently amended) The software as recited in claim 27 ~~26~~, wherein the third function group makes decisions by one of:

based on requests ~~from Program Control~~ from one or more of the hosts, determining where new application[[s]] computer programs should be started;

based on indication of application failure from the fourth function group, determining whether and where ~~the~~ a failed application computer program should be restarted;

based on indication of host failure from the fourth function group, determining whether and where the failed application computer program previously instantiated by the failed one of the N hosts should be restarted;

based on startup and shutdown dependency resolution requests from the fourth function group, determine whether and where additional application[[s]] computer programs should to be one of started and shut down prior to starting or shutting down another application computer program; and

based on changes to application system priorities, determining whether and where new application[[s]] computer programs need to be started and/or determine whether and which existing application[[s]] computer programs need to be shutdown.

37 ~~36~~. (currently amended) The software as recited in claim 27 ~~26~~, wherein the third function group makes decisions by one of:

based on application computer program inter dependencies defined in system specification files, determining whether and where additional application[[s]] computer programs should to be one of started and shut down prior to starting or shutting down of another application computer program;

based on application computer program instrumentation data generated by the second function group and performance requirements defined in the system specification files, determining whether application[[s]] computer programs are meeting performance requirements and whether an application computer program can be scaled up or moved to attempt to improve performance; and

based on the application computer program instrumentation data and performance requirements defined in the system specification files, determining whether application[[s]] computer programs are performing well within performance requirements and can be scaled down.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	PATENT NUMBER	GROUP ART UNIT	FILE WRAPPER LOCATION
09/864,825		2127	2111

Change of Address/Power of Attorney

The following fields have been set to Customer Number 23501 on 03/03/2005

- Correspondence Address
- Power of Attorney
- Maintenance Fee Address

The address of record for Customer Number 23501 is:
NAVAL SURFACE WARFARE CENTER, DAHLGREN DIVISION
OFFICE OF COUNSEL, CODE XDC1
17320 DAHLGREN ROAD
DAHLGREN, VA 22448-5110

The Practitioners of record for Customer Number 23501 are:

PTO INSTRUCTIONS:

Please take the following action when the correspondence address has been changed to a customer number:
1) Add 'ADDRESS CHANGE TO CUSTOMER NUMBER' on the next available content line of the File Jacket.
2) Put a line through the old address on the File Jacket and enter the Customer Number as the new address.
3) File this Notice in the File Jacket.

Please take the following action when the correspondence address has NOT been changed:
1) File this Notice in the File Jacket

59

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/864,825	05/24/2001	Paul V. Werme	NC 83017	2372

23501 7590 03/01/2005

NAVAL SURFACE WARFARE CENTER, DAHLGREN DIVISION
OFFICE OF COUNSEL, CODE XDC1
17320 DAHLGREN ROAD
DAHLGREN, VA 22448-5110

EXAMINER	
TANG, KENNETH	
ART UNIT	**PAPER NUMBER**
2127	

DATE MAILED: 03/01/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)	
	09/864,825	WERME ET AL.	
	Examiner	Art Unit	
	Kenneth Tang	2127	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 October 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-36* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-36* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the Amendment on 10/5/04.

2. Claims 1-36 are presented for examination.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 8-36 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention:

a. In claims 8, 16, and 26, "M" and "N" are indefinite because a definite range is not specified. The Examiner suggest amending the claims to reflect that the number of hosts is from 1 to N, and the number of managed characteristic applications is from 1 to M, for example.

b. In claims 16, "coupling the N hosts to one another, and the M managed characteristic application computer programs" (lines 7-8) is indefinite because it is not made explicitly clear in the claim language whether the managed character applications are coupled and connected to the hosts as well as the other managed character applications or just to the hosts.

4. Claims 16 and 26 are rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential structural cooperative relationships of elements, such omission amounting to a gap between the necessary structural connections. See MPEP § 2172.01. The

omitted structural cooperative relationships are: In claims 16 and 26, there lacks an essential relationship between the "first function", the "second function", and the "third function". For the third function to occur, it needs to interact with the first function to learn the state of health and also interact with the second function to learn the QoS requirements. There is no relationship established in the claim language for this. In addition, claim 26 also omits an essential relationship element between the "first function group", the "second function group", and the "third function group".

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. **Claims 1 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bhattal et al. (hereinafter Bhattal) (US 20020064126) in view of Du et al. (hereinafter Du) (US 6,041,306).**

6. As to claim 1, Bhattal teaches in a distributed environment comprised of hosts instantiating copies of an application computer program, a resource management device within one of the hosts of the distributed environment, the resource management device generating signals which start up, shutdown or move a selected one of the copies responsive to first information received by the resource management device regarding performance of one or more

copies of the application computer program and second information received by the resource management device regarding performance of the hosts *([0014]-[0016])*. Bhattal fails to explicitly teach the application being scalable. However, Du teaches a scalable computing system being an advantageous feature *(col. 5, lines 16-34)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of scalable applications to the existing system of Bhattal because of the advantages described above.

7. **Claims 2, 4-8, and 10-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bhattal et al. (hereinafter Bhattal) (US 20020064126).**

8. As to claim 2, Bhattal teaches on a host instantiating a managed characteristic computer program, the managed characteristic application computer program being managed by at least the host, a resource management device of the host, the resource management device generating signals responsive to first information received by the resource management device regarding performance of a plurality of application computer programs including the managed characteristic application computer program and second information received by the resrouce management device regarding performance of the host, the signals including: a first signal that starts up an additional copy of the managed characteristic application computer program on one of the host and a second networked host, a second signal shuts down and restarts the managed characteristic on the host, and a third signal that moves the managed characteristic to the second host *([0014]-[0016])*. Bhattal teaches using applications and using instances of communication

channels but fails to explicitly teach the instantiating managed characteristic be of an application. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made that there could be applications associated with channels. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of applications associated with the channels for instantiating because it will increase the functionality of the channels from the applications performing functions for them.

9. As to claim 4, Bhattal teaches the resource management device wherein the managed characteristic application computer program comprises a fault tolerant application, where the degree of fault tolerance is selectable by a user *([0070] and [0080])*.

10. As to claim 5, it is rejected for the same reasons as stated in the rejection of claim 2. In addition, Bhattal fails to explicitly teach wherein the managed characteristic application computer program comprises a selectable priority application. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of a selectable priority application because it would increase the functionality of the system if applications can be selected by order of importance.

11. As to claim 6, Bhattal fails to explicitly teach the resource management device wherein the managed characteristic application computer program further responds to user-initiated control actions. However, it would have been obvious to one of ordinary skill in the art at the

time the invention was made to include the feature of user-initiated control actions because more control will be given to the user.

12. As to claim 7, Bhattal teaches the resource management device wherein the resource arrangement device generates signals instructing a program control device to modify the configuration of the managed characteristic application computer program *([0038])*.

13. As to claim 8, it is rejected for the same reason as stated in the rejection of claim 2. In addition, Bhattal teaches there is a control configuration for the signals for the management device *([0008] and [0036])*.

14. As to claims 10-11, they are rejected for the same reasons as stated in the rejection of claims 4-5, respectively.

15. As to claim 12, Bhattal teaches the resource management device wherein the resource management device further generates signals responsive to third information received by the resource management device regarding the performance of hardware operatively coupling the networked hosts *(see Abstract)*.

16. As to claims 13-14, they are rejected for the same reasons as stated in the rejection of claims 5-6, respectively.

17. As to claim 15, Bhattal teaches the resource management device where the action requests are generated by an operator *([0002])*.

18. **Claims 3 and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bhattal et al. (hereinafter Bhattal) (US 20020064126) in view of Du et al. (hereinafter Du) (US 6,041,306).**

19. As to claim 3, Bhattal teaches in a distributed environment comprised of hosts instantiating copies of a characteristic application computer program, a resource management device generating signals which start up, shutdown or move a selected one of the copies responsive to first information regarding performance of all copies of the application and second information regarding performance of the hosts *([0014]-[0016])*. Bhattal fails to explicitly teach the application program being scalable. However, Du teaches a scalable computing system being an advantageous feature because of increasing capabilities *(col. 5, lines 16-34)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of scalable applications to the existing system of Bhattal because of the advantages described above.

20. As to claim 9, it is rejected for the same reasons as stated in the rejection of claim 3.

21. **Claims 16-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bhattal et al. (hereinafter Bhattal) (US 20020064126) in view of Du et al. (hereinafter Du) (US 6,041,306), and further in view of Dimitroff et al. (hereinafter Dimitroff) (US 6,742,020 B1).**

22. As to claim 16, it is rejected for the same reasons as stated in the rejection of claim 8. In addition, Bhattal fails to explicitly teach having states to represent the health of the hosts and that the actions need to maintain Quality of Service (QoS) requirements, and that . However, Du teaches using managers and state machines that contains states of every condition and determines the state and health based on the conditions automatically *(col. 6, lines 65-67 through col. 7, lines 1-15 and col. 12, lines 31-47 and col. 10, line 3)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of using managers and state machines that contains states of every condition and determines the state and health based on the conditions because they increase the integrity of the system by being able to know when a "unhealthy" state (or state that is considered below requirements) occurs. In addition, Du teaches using QoS requirements for the applications *(col. 11, lines 16-18)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of QoS requirements for the applications in order to maintain or increase the integrity of the system. In addition, Bhattal and Du fail to explicitly teach that the computer programs are moved, shutdown and started in accordance with satisfaction of the QoS requirements. However, Dimitroff teaches using (opening, closing, moving, etc.) application programs in accordance to satisfaction of QoS requirements *(col. 4, lines 7-18)*. It would have

been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of the computer programs are moved, shutdown and started in accordance with satisfaction of the QoS requirements to the existing networking system with QoS requirements of Bhattal in view of Du because ensuring that QOS requirements are met insofar as possible is desirable *(col.1, lines 57-65)*.

23. As to claim 17, Bhattal teaches the software wherein the first function receives system specification information comprising selected ones of host configuration and capabilities, application capabilities, survivability requirements, scalability characteristics, application startup and shutdown dependencies, and application and path performance requirements *([0067])*.

24. As to claim 18, Bhattal teaches the software wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic application computer programs, and detected failures regarding the N hosts *([0070] and [0080])*.

25. As to claim 19, Bhattal teaches wherein the first function receives application performance data representing each of the M managed characteristic application computer programs *(see Abstract)*.

26. As to claim 20, Bhattal teaches wherein the first function receives application performance data on one or more applications instantiated by the N hosts including performance

data representing each one of the M managed characteristic application computer programs *(see Abstract)*.

27. As to claim 21, Bhattal teaches the software wherein the second function which determines the required allocation and reallocation actions need to maintain the Quality of Service (QoS) requirements established for the M managed characteristic application by responding to application and host failures by determining if and what recovery actions should be taken; determining if and where to place new copies of one of the M managed characteristic application computer programs or which of the M managed characteristic application computer programs should be shutdown when QoS Manager functions indicate that scale-up or scale-down actions are indicated based on measured application performance and application QoS specifications; determining where new applications should be placed when requested to do so by a program control device; and determining which and how many application computer programs should run based on application system priorities *([0070] and [0080])*.

28. As to claim 22, Bhattal teaches the resource management device wherein the managed characteristic application computer programs comprises a scalable application *([0070] and [0080])*.

29. As to claim 23, Bhattal teaches the resource management device wherein the managed characteristic application computer programs comprises a fault tolerant application computer programs, where the degree of fault tolerance is selectable by a user *([0070] and [0080])*.

30. As to claim 24, Bhattal fails to explicitly teach wherein the managed characteristic application computer programs comprises a selectable priority application computer programs. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of a selectable priority application computer programs because it would increase the functionality of the system if applications can be selected by order of importance.

31. As to claim 25, Bhattal teaches using applications and using instances of communication channels but fails to explicitly teach the instantiating managed characteristic be of an application computer programs. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made that there could be application computer programs associated with channels. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of applications associated with the channels for instantiating because it will increase the functionality of the channels from the applications performing functions for them.

32. As to claim 26, it is rejected for the same reasons as stated in the rejection of claim 16. In addition, Bhattal fails to explicitly teach monitoring the host and network resources and reporting. However, Du teaches monitoring the host and network resources and reporting *(col. 5, lines 59-67 through col. 6, lines 1-10, col. 8, lines 3-10)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of monitoring

the host and network resources and reporting because the system could then diagnose or track the status or "health" which actions then can be taken to maintain or increase the integrity of the system.

33. As to claim 27, Bhattal teaches the software wherein the first function receives system specification information comprising host configuration and capabilities *([0067])*.

34. As to claim 28, it is rejected for the same reasons as stated in the rejection of claims 2 and 17.

35. As to claim 29, Bhattal teaches the software wherein the first function receives system specification information comprising path performance requirements regarding communication between at least two of the N hosts *([0006])*.

36. As to claim 30, it is rejected for the same reasons as stated in the rejection of claim 18.

37. As to claim 31, Bhattal teaches the software wherein the first function receives historical data regarding statuses, configuration, and loads of the N hosts and link statuses and loads regarding the network *([0006])*.

38. As to claims 32-34, they are rejected for the same reasons as stated in the rejection of claims 19-21.

39. As to claim 35, Bhattal teaches the software wherein the third function group makes decisions by one of: based on requests from Program Control, determining where new application computer programs should be started; based on indication of application failure from the fourth function group, determining whether and where the failed application computer programs should be restarted; based on indication of host failure from the fourth function group, determining whether and where the failed application computer programs previously instantiated by the failed one of the N hosts should be restarted; based on startup and shutdown dependency resolution requests from the fourth function group, determine whether and where additional applications should to be one of started and shut down prior to starting or shutting down another application; determining whether and where new applications need to be started and/or determine whether and which existing applications need to be shutdown *([0038])*.

Bhattal fails to explicitly teach wherein the managed characteristic application computer programs comprise a selectable priority application. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of a selectable priority application computer programs because it would increase the functionality of the system if applications can be selected by order of importance.

40. As to claim 36, it is rejected for the same reasons as stated in the rejection of claim 35 above. In addition, Bhattal fails to explicitly teach the application computer programs being scalable. However, Du teaches a scalable computing system being an advantageous feature *(col. 5, lines 16-34)*. It would have been obvious to one of ordinary skill in the art at the time the

invention was made to include the feature of scalable applications to the existing system of Bhattal because of the advantages described above.

Response to Arguments

41. Applicant's arguments have been fully considered but are moot in view of the new grounds of rejections.

42. Applicant argues (pages 21 and 23) that Bhattal does not teach startup, shutdown or moving of an application computer program but instead of a channel.

In response, the examiner respectfully disagrees. Bhuttal does teach a channel to transmit from a host to a device. But the control of the channel is part of the application program (along with application queues) (Fig. 1, items 20, 40, 30', etc.). The channel can't be started unless the application for it is started, for example.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kenneth Tang whose telephone number is (571) 272-3772. The examiner can normally be reached on 8:30AM - 6:00PM, Every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

2/15/05

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Notice of References Cited	Application/Control No. 09/864,825	Applicant(s)/Patent Under Reexamination WERME ET AL.	
	Examiner Kenneth Tang	Art Unit 2127	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-6,742,020 B1	05-2004	Dimitroff et al.	709/217
	B	US-			
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims		**Application No.** 09/864,825	**Applicant(s)** WERME ET AL.
		Examiner Kenneth Tang	**Art Unit** 2127

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim Final	Claim Original	Date	Date 2/15/05	Claim Final	Claim Original	Claim Final	Claim Original
	1	√	√		51		101
	2	√	√		52		102
	3	√	√		53		103
	4	√	√		54		104
	5	√	√		55		105
	6	√	√		56		106
	7	√	√		57		107
	8	√	√		58		108
	9	√	√		59		109
	10	√	√		60		110
	11	√	√		61		111
	12	√	√		62		112
	13	√	√		63		113
	14	√	√		64		114
	15	√	√		65		115
	16	√	√		66		116
	17	√	√		67		117
	18	√	√		68		118
	19	√	√		69		119
	20	√	√		70		120
	21	√	√		71		121
	22	√	√		72		122
	23	√	√		73		123
	24	√	√		74		124
	25	√	√		75		125
	26	√	√		76		126
	27	√	√		77		127
	28	√	√		78		128
	29	√	√		79		129
	30	√	√		80		130
	31	√	√		81		131
	32	√	√		82		132
	33	√	√		83		133
	34	√	√		84		134
	35	√	√		85		135
	36	√	√		86		136
	37				87		137
	38				88		138
	39				89		139
	40				90		140
	41				91		141
	42				92		142
	43				93		143
	44				94		144
	45				95		145
	46				96		146
	47				97		147
	48				98		148
	49				99		149
	50				100		150

U.S. Patent and Trademark Office

Part of Paper No. 06262004

Search Notes	Application No.	Applicant(s)	
	09/864,825	WERME ET AL.	
	Examiner	Art Unit	
	Kenneth Tang	2127	

SEARCHED			
Class	Subclass	Date	Examiner
718	100-108	6/27/2004	KT
705	8	6/27/2004	KT
370	217	6/27/2004	KT

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
EAST	6/27/2004	KT
EAST	2/15/2005	KT

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	1024	scalable and (resource adj manag$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:39
S2	981	scalable and (resource adj manag$5) and network$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:39
S3	562	@ad<="20010524" and (scalable and (resource adj manag$5) and network$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:39
S4	202	(start$3 same (shut$3 ends) same restart$3) with (application program)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:39
S5	0	(scalable and (resource adj manag$5)) and @ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:39
S6	0	(scalable and (resource adj manag$5) and network$3) and @ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:40
S7	160	@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:40
S8	0	(scalable and (resource adj manag$5)) and (@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:40
S9	29	(start$3 same (shut$3 ends) same restart$3) with (copy instance)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:40

S10	170	resumption with (copy instance)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:40
S11	582	@ad<="20010524" and (scalable and (resource adj manag$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:40
S12	0	(scalable and (resource adj manag$5)) and (resumption with (copy instance))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:40
S13	148	@ad<="20010524" and (resumption with (copy instance))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:40
S14	0	((US-20020064126-$).did.) and scalable	US-PGPUB; USPAT	OR	ON	2004/06/28 06:40
S15	0	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020064126-$).did.) and scalable	US-PGPUB; USPAT	OR	ON	2004/06/28 06:40
S16	104	scalable adj application	US-PGPUB; USPAT	OR	ON	2004/06/28 06:40
S17	20660895	@ad<="20010524"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:40
S18	0	((US-20020064126-$).did.) and performance	US-PGPUB	OR	ON	2004/06/28 06:40
S19	2	(@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (application program))) and (resource adj manag$5) and network$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:40
S20	24	@ad<="20010524" and ((start$3 same (shut$3 ends) same restart$3) with (copy instance))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:40
S21	38	(@ad<="20010524" and (resumption with (copy instance))) and network	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:41

S22	25	((starts and resum$) with (copy instance)) same (manager controller)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:41
S23	23	@ad<="20010524" and (((starts and resum$) with (copy instance)) same (manager controller))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:41
S24	1	("20020064126").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2004/06/28 06:41
S25	3	quality adj of adj service	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:41
S26	4	(US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020064126-$).did.	US-PGPUB; USPAT	OR	OFF	2004/06/28 06:41
S27	64	(scalable adj application) and @ad<="20010524"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:41
S28	1	((US-20020174420-$).did.) and performance	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:59
S29	1	((US-20020064126-$).did.) and perform$	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:41
S30	1	((US-20020064126-$).did.) and application	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:41
S31	5	(US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.	US-PGPUB; USPAT	OR	OFF	2004/06/28 06:43
S32	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and monitor$3	USPAT	OR	ON	2004/06/28 06:52

S33	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and monitor$3 and (status state)	USPAT	OR	ON	2004/06/28 07:33
S34	1	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and qos	USPAT	OR	ON	2004/06/28 06:57
S35	1	((US-20020174420-$).did.) and states	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/06/28 06:59
S36	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and automat$	USPAT	OR	ON	2004/06/28 07:52
S37	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and scal$	USPAT	OR	ON	2004/06/28 07:52
S66	1	("20020064126").PN.	US-PGPUB; USPAT	OR	OFF	2005/02/15 11:24
S67	1	S66 and programs	US-PGPUB; USPAT	OR	ON	2005/02/15 11:33
S68	1370165	((start$3 execut$3 begin$3) and (stop$3 end$3 exit$3))	US-PGPUB; USPAT	OR	ON	2005/02/15 11:29
S69	20741	((start$3 execut$3 begin$3) and (stop$3 end$3 exit$3)) near2 (application program)	US-PGPUB; USPAT	OR	ON	2005/02/15 11:42
S70	12793	S69 and (network client server host)	US-PGPUB; USPAT	OR	ON	2005/02/15 11:47
S71	1	S66 and ((start$3 begin$3 end$3 exit$3) with (program function))	US-PGPUB; USPAT	OR	ON	2005/02/15 12:10
S72	8905	((start$3 execut$3 begin$3) and (stop$3 end$3 exit$3)) adj2 (application program)	US-PGPUB; USPAT	OR	ON	2005/02/15 11:46
S73	1	(US-6041306-$).did.	USPAT	OR	OFF	2005/02/15 13:00
S74	1	S73 and (application program)	USPAT	OR	ON	2005/02/15 11:44
S75	1	S73 and ((start$3 begin$3 end$3 exit$3) with (program function application))	US-PGPUB; USPAT	OR	ON	2005/02/15 11:45
S76	169	S72 and qos	US-PGPUB; USPAT	OR	ON	2005/02/15 11:46

S77	20720627	@ad<="20010524"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/15 11:46
S78	166	S76 and "12"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/15 11:47
S79	94	S76 and S77	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/15 11:47
S80	90	S79 and (network client server host)	US-PGPUB; USPAT	OR	ON	2005/02/15 11:47
S81	72	S80 and ((start$3 begin$3 end$3 exit$3) with (program function))	US-PGPUB; USPAT	OR	ON	2005/02/15 13:01
S82	1	S73 and qos	USPAT	OR	OFF	2005/02/15 13:00
S83	0	S81 and ((satisfaction compliance) with qos)	US-PGPUB; USPAT	OR	ON	2005/02/15 13:01
S84	8	S81 and ((satisfaction compliance requirement) with qos)	US-PGPUB; USPAT	OR	ON	2005/02/15 13:03
S85	1242	(reliance rel$4 compliance meet met) near5 qos	US-PGPUB; USPAT	OR	ON	2005/02/15 13:04
S86	1021	(reliance rel$4 compliance meet met) near3 qos	US-PGPUB; USPAT	OR	ON	2005/02/15 13:05
S87	261	S86 and ((application computer) adj program)	US-PGPUB; USPAT	OR	ON	2005/02/15 13:04
S88	13	((reliance rel$4 compliance meet met) near3 qos) same ((application computer) adj program)	US-PGPUB; USPAT	OR	ON	2005/02/15 13:05
S89	6	S88 and S77	US-PGPUB; USPAT	OR	ON	2005/02/15 13:06

OIPE JC46
OCT 05 2004
PATENT & TRADEMARK OFFICE

PTO/SB/21 (05-03)
Approved for use through 04/30/2003. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

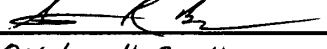
Application Number	09/864,825
Filing Date	24 May 2001
First Named Inventor	Paul V. Werme
Art Unit	2127
Examiner Name	Kenneth Tang
Total Number of Pages in This Submission: 25	Attorney Docket Number: 83017

ENCLOSURES *(Check all that apply)*

- [ ] Fee Transmittal Form
  - [ ] Fee Attached
- [X] Amendment/Reply
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Response to Missing Parts/ Incomplete Application
  - [ ] Response to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
- [ ] After Allowance communication to Group
- [ ] Appeal Communication to Board of Appeals and Interferences
- [ ] Appeal Communication to Group **(Appeal Notice, Brief, Reply Brief)**
- [ ] Proprietary Information
- [ ] Status Letter
- [ ] Other Enclosure(s) (please Identify below):

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	SCOTT R. BOALICK, CUSTOMER #23501
Signature	
Date	October 4, 2004

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below.

Typed or printed name	DEBORAH G. BLAKLEY		
Signature	Deborah G. Blakley	Date	10/4/04

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

OIPE JC49
OCT 05 2004
PATENT & TRADEMARK OFFICE

IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Paul V. Werme	
Application No.: 09/864,825	Group Art Unit: 2127
Filed: 5/24/2001	Examiner: Kenneth Tang
Title: RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR	
Attorney Docket No.: NC 83017	

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT AND RESPONSE TO OFFICE ACTION

Dear Sir:

In response to the Office Action of July 2, 2004, please amend the application as follows.

RECEIVED
OCT 14 2004
OIPE/JCW5

In the claims

Please amend the claims as follows:

1. (currently amended) In a distributed environment comprised of hosts instantiating copies of a scalable application computer program, a resource management device within one of the hosts of the distributed environment, the resource management device generating signals which start up, shutdown or move a selected one of the copies responsive to first information received by the resource management device regarding performance of ~~all~~ one or more copies of the scalable application computer program and second information received by the resource management device regarding performance of the hosts.

2. (currently amended) On a host instantiating a managed characteristic application computer program, the managed characteristic application computer program being managed by at least the host, a resource management device of the host, the resource management device generating signals responsive to first information received by the resource management device regarding performance of a plurality of application computer programs including the managed characteristic application computer program and second information received by the resource management device regarding performance of the host, the signals including:

a first signal that ~~which one of~~ starts up ~~an additional~~ a copy of the managed characteristic application computer program on one of the host and a second networked host[[,]];

a second signal that shuts down and restarts the managed characteristic application computer program on the host[[,]]; and

a third signal that moves the managed characteristic application computer program to the second host ~~responsive to first information regarding performance of all applications including the managed characteristic application and second information regarding performance of the host~~.

3. (currently amended) The resource management device as recited in claim 2, wherein the managed characteristic application computer program comprises a scalable application computer program.

4. (currently amended) The resource management device as recited in claim 2, wherein the managed characteristic application computer program comprises a fault tolerant application computer program, where the degree of fault tolerance is selectable by a user.

5. (currently amended) The resource management device as recited in claim 2, wherein the managed characteristic application computer program comprises a selectable priority application computer program.

6. (currently amended) The resource management device as recited in claim 2, wherein the managed characteristic application computer program further responds to user initiated control actions.

7. (currently amended) The ~~program control~~ resource management device as recited in claim 2, wherein the resource management device generates signals instructing a program control device to modifies the configuration of the managed characteristic application computer program.

8. (currently amended) In a grid system comprising networked hosts instantiating M managed characteristic application[[s]] computer programs, a resource management device within one of the hosts generating signals responsive to first information received by the resource management device regarding performance of the M managed characteristic application computer programs and second information received by the resource management device regarding performance of the hosts, where M is a positive integer, ~~which one of~~ the signals including:

a first signal that controls a configuration of an M th copy of the M managed characteristic application computer programs, each of the M managed characteristic application computer programs being managed by one of the hosts;

a second signal that controls a start up of an (M+1) th copy of the M managed characteristic application[[s]] computer programs;[[,]]

a third signal that controls a shutdown and restart of the M th copy of the M managed characteristic application[[s]] computer programs;[[,]] and

a fourth signal that controls a movement of the M th copy of the M managed characteristic application[[s]] computer programs from a first to a second host ~~responsive to first information regarding performance of the M managed characteristic applications and second information regarding performance of the hosts, where M is a positive integer~~.

9. (currently amended) The resource management device as recited in claim 8, wherein at least one of the M managed characteristic application[[s]] computer programs comprises a scalable application computer program.

10. (currently amended) The resource management device as recited in claim 8, wherein at least one of the M managed characteristic application[[s]] computer programs comprise a fault tolerant application computer program, where the degree of fault tolerance is selectable by a user.

11. (currently amended) The resource management device as recited in claim 8, wherein at least one of the M managed characteristic application[[s]] computer programs comprises a selectable priority application computer program.

12. (currently amended) The resource management device as recited in claim 8, wherein the resource management device further ~~responds~~ generates signals responsive to third information

received by the resource management device regarding the performance of hardware operatively coupling the networked hosts.

13. (currently amended) The resource management device as recited in claim 8, wherein the resource management device further ~~responds~~ generates signals responsive to fourth information received by the resource management device establishing one of the priority assigned to the managed characteristic application computer program and multiple copies of the managed characteristic application computer program.

14. (currently amended) The resource management device as recited in claim 8, wherein the resource management device further ~~responds~~ generates signals responsive to externally generated action requests received by the resource management device.

15. (original) The resource management device as recited in claim 14, where the action requests are generated by an operator.

16. (currently amended) In a distributed environment comprised of N hosts operating in a distributed environment instantiating M managed characteristic application[[s]] computer programs managed by the N hosts, resource allocation control software instantiated by at least the N hosts, M and N each being a positive integer, where M may be equal to, less than, or greater than N, the software comprising:

a first function which determines ~~the~~ a state and health of the N hosts, a network operatively coupling the N hosts to one another, and the M managed characteristic application[[s]] computer programs in the distributed environment;

a second function which determines required allocation and reallocation actions needed to maintain a plurality of Quality of Service (QoS) requirements established for the M managed

characteristic application[[s]] computer programs, the QoS requirements dictating parameters regarding service quality of the M managed characteristic application computer programs; and

a third function which generates automatic control signal requests corresponding to the actions dictated by the QoS requirements, such that the managed characteristic application computer programs are moved, shutdown, and started in accordance with satisfaction of the QoS requirements.

17. (original) The software as recited in claim 16, wherein the first function receives system specification information comprising selected ones of host configuration and capabilities, application capabilities, survivability requirements, scalability characteristics, application startup and shutdown dependencies, and application and path performance requirements.

18. (currently amended) The software as recited in claim 16, wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic application[[s]] computer programs, and detected failures regarding the N hosts.

18. (cancelled)

19. (currently amended) The software as recited in claim 16, wherein the first function receives application performance data representing each ~~one~~ of the M managed characteristic application[[s]] computer programs.

20. (currently amended) The software as recited in claim 16, wherein the first function receives application performance data on ~~all~~ one or more applications instantiated by the N hosts including performance data representing each ~~one~~ of the M managed characteristic application[[s]] computer programs.

21. (currently amended) The software as recited in claim 16, wherein the second function ~~which~~ determines the required allocation and reallocation actions needed to maintain the Quality of Service (QoS) requirements established for the M managed characteristic applications by:

responding to application and host failures by determining if and what recovery actions should be taken;

determining if and where to place new copies of one of the M managed characteristic application[[s]] computer programs or which of the M managed characteristic application[[s]] computer programs should be shutdown when QoS Manager functions indicate that scale up or scale down actions are indicated based on measured application performance and application QoS specifications;

determining where new application[[s]] computer programs should be placed when requested to do so by a program control device; and

determining which and how many application[[s]] computer programs should run based on application system priorities.

22. (currently amended) The software as recited in claim 16, wherein at least one of the M managed characteristic application[[s]] computer programs comprises a scalable application computer program.

23. (currently amended) The software as recited in claim 16, wherein at least one of the M managed characteristic application[[s]] computer programs comprises a fault tolerant application computer program, where the degree of fault tolerance is selectable by a user.

24. (currently amended) The software as recited in claim 16, wherein one of the M managed characteristic application[[s]] computer programs comprises a selectable priority application computer program.

25. (currently amended) The software as recited in claim 16, wherein the M managed characteristic application[[s]] computer programs comprise M copies of a single managed characteristic application computer program.

26. (currently amended) Software stored on at least one host for converting N networked hosts into a resource managed system instantiating M managed characteristic application[[s]] computer programs, each managed characteristic application computer program managed by one of the N networked hosts, the software comprising:

a first function group which monitors the host and network resources;

a second function group which provides ~~general purpose~~ application computer program event reporting and event correlation capabilities;

a third function group which provides ~~the~~ reasoning and[[,]] decision making capabilities for the resource managed system, wherein the third function group comprises:

a first function which determines ~~the~~ a state and health of the N hosts, a network operatively coupling the N hosts to one another and the M managed characteristic application[[s]] computer programs in the distributed environment;

a second function which determines required allocation and reallocation actions needed to maintain a plurality of Quality of Service (QoS) requirements established for the M managed characteristic application[[s]] computer programs, the QoS requirements dictating parameters regarding service quality of the M management characteristic application programs; and

a third function which generates automatic control signal requests corresponding to the actions dictated by the QoS requirements, such that the managed characteristic application computer programs are moved, shutdown, and started in accordance with satisfaction of the QoS requirements; and

a fourth function group which provides program control capabilities permitting starting, stopping, and configuring of selected ones of the M managed characteristic application[[s]] computer programs on respective ones of the N hosts in the resource managed system,

where M and N are positive integers and where M may be equal to, greater than, or less than N.

27. (currently amended) The software as recited in claim 26, wherein the first function receives system specification information ~~comprises~~ comprising host configuration and capabilities.

28. (currently amended) The software as recited in claim 26, wherein the first function receives system specification information comprising selected ones of capabilities, survivability requirements, scalability characteristics, startup and shutdown dependencies, and performance requirements for at least one of the M managed characteristic application[[s]] computer programs.

29. (original) The software as recited in claim 26, wherein the first function receives system specification information comprising path performance requirements regarding communication between at least two of the N hosts.

30. (currently amended) The software as recited in claim 26, wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic application[[s]] computer programs, and detected failures regarding the N hosts.

31. (original) The software as recited in claim 26, wherein the first function receives historical data regarding statuses, configuration, and loads of the N hosts and link statuses and loads regarding the network.

32. (currently amended) The software as recited in claim 26, wherein the first function receives application performance data representing each one of the M managed characteristic application[[s]] computer programs.

33. (currently amended) The software as recited in claim 26, wherein the first function receives application performance data on ~~all~~ one or more applications instantiated by the N hosts including performance data representing each ~~one~~ of the N copies of the managed characteristic application computer programs.

34. (currently amended) The software as recited in claim 26, wherein the second function which determines the required allocation and reallocation actions established for the M managed characteristic application computer programs by:

responding to application and host failures by determining if and what recovery actions should be taken;

determining if and where to place new copies of managed characteristic application[[s]] computer programs or which managed characteristic application[[s]] computer programs should be shutdown when QoS Manager functions indicate that scale up or scale down actions should be taken based on measured application performance and QoS specifications established for the M managed characteristic application[[s]] computer programs;

determining where new application[[s]] computer programs should be placed when requested to do so by the fourth function group; and

determining which and how many application[[s]] computer programs should run based on application system priorities.

35. (currently amended) The software as recited in claim 26, wherein the third function group makes decisions by one of:

based on requests ~~from Program Control~~ from one or more of the hosts, determining where new application[[s]] computer programs should be started;

based on indication of application failure from the fourth function group, determining whether and where ~~the~~ a failed application computer program should be restarted;

based on indication of host failure from the fourth function group, determining whether and where the failed application computer program previously instantiated by the failed one of the N hosts should be restarted;

based on startup and shutdown dependency resolution requests from the fourth function group, determine whether and where additional application[[s]] computer programs should to be one of started and shut down prior to starting or shutting down another application computer program; and

based on changes to application system priorities, determining whether and where new application[[s]] computer programs need to be started and/or determine whether and which existing application[[s]] computer programs need to be shutdown.

36. (currently amended) The software as recited in claim 26, wherein the third function group makes decisions by one of:

based on application computer program inter dependencies defined in system specification files, determining whether and where additional application[[s]] computer programs should to be one of started and shut down prior to starting or shutting down of another application computer program;

based on application computer program instrumentation data generated by the second function group and performance requirements defined in the system specification files, determining whether application[[s]] computer programs are meeting performance requirements and whether an application computer program can be scaled up or moved to attempt to improve performance; and

based on the application computer program instrumentation data and performance requirements defined in the system specification files, determining whether application[[s]] computer programs are performing well within performance requirements and can be scaled down.

REMARKS

Claims 1-36 are pending, with claims 1, 2, 8, 16, and 26 being independent. Claims 1-14, 16, 18-28, 30, and 32-36 have been amended and a second claim 18 has been canceled by this response. In view of the foregoing amendments and following remarks, reconsideration and allowance of this application are respectfully requested.

Claim rejections under 35 USC 112

Claims 1-36 have been rejected under 35 USC 112, second paragraph, as being indefinite. Each basis of rejection is now discussed in order.

a. In claim 1, the phrase "resource management device" has been indicated as being indefinite because it is not clear whether this device is in the host or not, nor which host it is in. This rejection has been rendered moot by the amendment to claim 1. Specifically, Applicant has amended claim 1 to recite that the resource management device is "within one of the hosts." Therefore, it is respectfully requested that this rejection be withdrawn.

b. In claim 1, the phrase "first information" recited in line 3 has been indicated as being indefinite due to not having sufficient antecedent basis. Applicant respectfully disagrees. The phrase "first information" was recited for the first time in line 3, without the prefix "said" or "the." However, to expedite prosecution of this claim, Applicant has amended claim 1 to recite that the first information is "received by the resource management device." Therefore, it is respectfully requested that this rejection be withdrawn.

c. In claim 1, the phrase "second information" recited in line 4 has been indicated as being indefinite due to not having sufficient antecedent basis. Applicant respectfully disagrees. The phrase "second information" was recited for the first time in line 4, without the prefix "said" or "the." However, to expedite prosecution of this claim, Applicant has amended claim 1 to recite

that the second information is "received by the resource management device." Therefore, it is respectfully requested that this rejection be withdrawn.

d. In claim 2, the phrase "resource management device" has been indicated as being indefinite because it is not clear whether this device is in the host or not, nor which host it is in. This rejection has been rendered moot by the amendment to claim 2. Specifically, Applicant has amended claim 2 to recite that the resource management device is "of the host." Therefore, it is respectfully requested that this rejection be withdrawn.

e. In claim 2, the phrase "managed characteristic application" has been indicated as being indefinite because it is not clear who manages the characteristic application program, and that the term is not defined in the claim language. This rejection has been rendered moot by the amendment to claim 2. Specifically, Applicant has amended claim 2 to recite, first, that the managed characteristic application is a "computer program," such that its definition is even more clear (i.e., the phrase refers to an actual computer program). Second, Applicant has amended claim 2 to recite that this computer program is managed "by at least the host." Therefore, it is respectfully requested that this rejection be withdrawn.

f. In claim 2, the phrase "which one of starts up" has been indicated as being indefinite as not making any sense. This rejection has been rendered moot by the amendment to claim 2. Applicant has removed the language "which one of" from the claim, and has amended the claim to recite that the signals include a first signal, a second signal, and a third signal that start up, shut down, and move compute programs as delineated in the claim. Therefore, it is respectfully requested that this rejection be withdrawn.

g. In claim 2, the phrase "additional copy" has been indicated as being indefinite because a first copy was never introduced. This rejection has been rendered moot by the amendment to

claim 2. Applicant has amended the claim to recite "a copy." instead of an "additional copy." Therefore, it is respectfully requested that this rejection be withdrawn.

h. In claim 2, the phrase "one of the host" has been indicated as being indefinite because it does not make any sense. Applicant respectfully disagrees. The claim recites "on one of the host and a second networked host." As such, the "one of" language refers to "one of" the host and a second networked host and therefore means either the host or the second networked host. Therefore, it is respectfully requested that this rejection be withdrawn.

i. In claim 2, the phrase "all applications" has been indicated as being indefinite because only one application was previously introduced. This rejection has been rendered moot by the amendment to claim 2. Applicant has amended claim 2 to recite "a plurality of application computer programs including the managed characteristic application computer program." Therefore, it is respectfully requested that this rejection be withdrawn.

j. In claim 2, the phrase "first information" in line 5 has been indicated as being indefinite due to insufficient antecedent basis. Applicant respectfully disagrees. The phrase "first information" was recited for the first time in line 5, without the prefix "said" or "the." However, to expedite prosecution of this claim, Applicant has amended claim 2 to recite that the first information is "received by the resource management device." Therefore, it is respectfully requested that this rejection be withdrawn.

k. In claim 2, the phrase "second information" recited in line 6 has been indicated as being indefinite due to not having sufficient antecedent basis. Applicant respectfully disagrees. The phrase "second information" was recited for the first time in line 6, without the prefix "said" or "the." However, expedite prosecution of this claim, Applicant has amended claim 2 to recite that

the second information is "received by the resource management device." Therefore, it is respectfully requested that this rejection be withdrawn.

l. In claim 8, the phrase "resource management device" has been indicated as being indefinite because it is not clear whether this device is in the house or not, nor which host. This rejection has been rendered moot by the amendment to claim 8. Applicant has amended claim 8 to recite that the device is "within one of the hosts." Therefore, it is respectfully requested that this rejection be withdrawn.

m. In claim 8, the phrase "managed characteristic application" has been indicated as being indefinite because it is not clear who manages the characteristic application program, and that the term is not defined in the claim language. This rejection has been rendered moot by the amendment to claim 8. Applicant has amended claim 8 to recite, first, that the managed characteristic application is a "computer program," such that its definition is even more clear (i.e., the phrase refers to an actual computer program). Second, Applicant has amended claim 8 to recite that this computer program is managed "by one of the hosts." Therefore, it is respectfully requested that this rejection be withdrawn.

n. In claim 8, the phrase "first information" in line 6 has been indicated as being indefinite due to insufficient antecedent basis. Applicant respectfully disagrees. The phrase "first information" was recited for the first time in line 6, without the prefix "said" or "the." However, to expedite prosecution of this claim, Applicant has amended claim 8 to recite that the first information is "received by the resource management device." Therefore, it is respectfully requested that this rejection be withdrawn.

o. In claim 8, the phrase "second information" recited in line 7 has been indicated as being indefinite due to not having sufficient antecedent basis. Applicant respectfully disagrees. The

phrase "second information" was recited for the first time in line 7, without the prefix "said" or "the." However, to expedite prosecution of this claim, Applicant has amended claim 8 to recite that the second information is "received by the resource management device." Therefore, it is respectfully requested that this rejection be withdrawn.

p. In claim 8, the phrase "control configuration" in line 2 has been indicated as being indefinite due to insufficient antecedent basis. This rejection has been rendered moot by the amendment to claim 8. Applicant has amended claim 8 to recite "a control configuration." Therefore, it is respectfully requested that this rejection be withdrawn.

q. In claim 16, the phrase "managed characteristic application" has been indicated as being indefinite because it is not clear who manages the characteristic application program, and that the term is not defined in the claim language. This rejection has been rendered moot by the amendment to claim 16. Applicant has amended claim 16 to recite, first, that the managed characteristic application is a "computer program," such that its definition is even more clear (i.e., the phrase refers to an actual computer program). Second, Applicant has amended claim 16 to recite that this computer program is managed "by the N hosts." Therefore, it is respectfully requested that this rejection be withdrawn.

r. In claim 16, the letters "N" and "M" have been indicated as being indefinite because it is not clear whether these are variables or whether they present static numbers, as well as whether N is different than M. This rejection has been rendered moot by the amendment to claim 16. Applicant has amended claim 16 to recite that "M and N" are each "a positive integer," and further that "M may be equal to, less than, or greater than N." Therefore, it is respectfully requested that this rejection be withdrawn.

s. In claim 16, the phrase "the state and health" in line 4 has been indicated as being indefinite because there is insufficient antecedent basis. This rejection has been rendered moot by the amendment to claim 16. Specifically, Applicant has amended the claim to recite "a state and health." Therefore, it is respectfully requested that this rejection be withdrawn.

t. In claim 16, the phrase "Quality of Service (QoS)" has been indicated as being indefinite for lack of sufficient antecedent basis. This rejection has been rendered moot by the amendment to claim 16. Specifically, Applicant has amended the claim to recite "a plurality of Quality of Service (QoS) requirements." Therefore, it is respectfully requested that this rejection be withdrawn.

u. In claim 16, the phrase "Quality of Service (QoS)" has been indicated as being indefinite because it is not clear what the requirements involving this term are. This rejection has been rendered moot by the amendment to claim 16. In particular, Applicant has amended claim 16 to recite that the QoS requirements specifically "dictat[e] parameters regarding service quality of the M management characteristic application programs." Therefore, it is respectfully requested that this rejection be withdrawn.

v. Claim 18 has been indicated as being indefinite because there are two different claim 18s. Applicant has cancelled one of the claim 18s, and submits that the remaining claim 18 is not indefinite.

w. In claim 26, the letters "N" and "M" have been indicated as being indefinite because it is not clear whether these are variables or whether they present static numbers, as well as whether N is different than M. This rejection has been rendered moot by the amendment to claim 26. First, Applicant notes that claim 26 continues to recite that "M and N are positive integers." Second,

Applicant has amended claim 26 to recite that "M may be equal to, less than, or greater than N." Therefore, it is respectfully requested that this rejection be withdrawn.

x. In claim 26, the phrase "managed characteristic application" has been indicated as being indefinite because it is not clear who manages the characteristic application program, and that the term is not defined in the claim language. This rejection has been rendered moot by the amendment to claim 26. Applicant has amended claim 26 to recite, first, that the managed characteristic application is a "computer program," such that its definition is even more clear (i.e., the phrase refers to an actual computer program). Second, Applicant has amended claim 26 to recite that this computer program is managed "by one of the N networked hosts." Therefore, it is respectfully requested that this rejection be withdrawn.

y. In claim 26, the phrase "the state and health" in line 8 has been indicated as being indefinite because there is insufficient antecedent basis. This rejection has been rendered moot by the amendment to claim 26. Applicant has amended the claim to recite "a state and health." Therefore, it is respectfully requested that this rejection be withdrawn.

z. In claim 26, the phrase "Quality of Service (QoS)" has been indicated as being indefinite for lack of sufficient antecedent basis. This rejection has been rendered moot by the amendment to claim 26. Applicant has amended the claim to recite "a plurality of Quality of Service (QoS) requirements." Therefore, it is respectfully requested that this rejection be withdrawn.

aa. In claim 26, the phrase "Quality of Service (QoS)" has been indicated as being indefinite because it is not clear what the requirements involving this term are. This rejection has been rendered moot by the amendment to claim 26. Applicant has amended claim 26 to recite that the QoS requirements specifically "dictat[e] parameters regarding service quality of the M

management characteristic application programs." Therefore, it is respectfully requested that this rejection be withdrawn.

bb. In claim 26, the phrase "Quality of Service (QoS)" has been indicated as being indefinite for lack of sufficient antecedent basis. This appears to be the same rejection as z. above. As with the rejection discussed above with respect to item z, this rejection has been rendered moot by the amendment to claim 26. Applicant has amended the claim to recite "a plurality of Quality of Service (QoS) requirements." Therefore, it is respectfully requested that this rejection be withdrawn.

cc. In claim 26, the phrase "general-purpose" has been indicated as a relative term that renders the claim indefinite. This rejection has been rendered moot by the amendment to claim 26. In particular, Applicant has amended claim 26 so that this phrase is no longer present. Therefore, it is respectfully requested that this rejection be withdrawn.

dd. In claim 35, the phrase "Program Control" has been indicated as being indefinite for lack of sufficient antecedent basis. This rejection has been rendered moot by the amendment to claim 35. Applicant has amended claim 35 so that this phrase is no longer present. Therefore, it is respectfully requested that this rejection be withdrawn.

Claim rejections under 35 USC 103

Claims 1 and 16-32

Claims 1 and 16-32 have been rejected under 35 USC 103(a) as being unpatentable over Bhattal (2002/0064126) in view of Du (6,041,306). Claims 1, 16, and 26 are independent claims, from which the remaining of the claims rejected over Bhattal in view of Du ultimately depend.

Applicant asserts that claims 1, 16, and 26, as amended, are patentable over Bhattal in view of Du, such that all of claims 1 and 16-32 are patentable over Bhattal in view of Du.

Claim 1 is limited to a resource management device generating signals that "startup, shutdown or move a selected one of the copies" of a scalable application *computer program.* The Examiner relies specifically on Bhattal in finding such a device that generates signal that startup, shutdown, or move a selected scalable application computer program. However, Applicant submits that Bhattal does not disclose the startup, shutdown, or moving of a scalable application *computer program* in particular, such that Bhattal in view of Du does not disclose the claimed invention of claim 1.

Paragraphs [0014]-[0016] of Bhattal, which the Examiner has relied upon, specifically relate to the starting of "an instance of a *channel*," not a computer program as recited in claim 1. An example of a channel is shown as the channel 60 of FIG. 1 of Bhattal, which allows application computer programs 20 and 40 to send messages to one another. Thus, it is not the application computer programs 20 and 40 that are started up, shutdown, or moved in Bhattal, as recited in claim 1, but rather the communication channel 60. A channel, however, is not an application computer program. Rather, it "is a one-way communication link between two queue managers." (Para. [0034]) As such, Bhattal in view of Du does not render claim 1 obvious.

Furthermore, Bhattal, and thus Bhattal in view of Du, has nothing to do with starting up, shutting down, or moving instances of computer programs. There is thus no motivation to modify Bhattal to substitute computer programs for its channels. Indeed, to substitute computer programs for the channels of Bhattal in view of Du destroys the operation of Bhattal. Simply put, a communication channel allows for communication between two computer programs; the same cannot be said of a computer program itself. Therefore, Bhattal in view of Du cannot be modified to yield the recited limitations of claim 1.

Similarly, independent claims 16 and 26 recite the movement, startup, and shutdown of application computer programs -- specifically, managed characteristic application programs.

Therefore, claims 16 and 26 are patentable over Bhattal in view of Du for at least the reasons discussed above with respect to claim 1.

Furthermore, claims 16 and 26 are patentable over Bhattal in view of Du because each recites Quality of Service (QoS) requirements ultimately resulting in the movement, startup, or shutdown of application computer programs. For instance, as amended, claims 16 and 26 are limited to "application computer programs [being] moved, shutdown, and started in accordance with satisfaction of the QoS requirements." Applicant submits that Bhattal in view of Du does not disclose such reliance on QoS requirements.

The Examiner specifically relies on Du to disclose QoS requirements, in column 11, lines 16-18 thereof. This part of Du reads as follows:

> 7. Get customer's approval of the selected configuration. *Once a suitable path is identified, the customer will review the offer, including available date, charges, quality of services (QoS), and so on.* Depending upon the business factors (e.g., cheapest service wanted), the customer may request that a new search be initiated, that is, loop back to step 5 to find another valid route.

(Col. 11, lines 15-21) This seventh step in Du is part of a process that relates to a customer requesting "to add or drop communication paths between certain endpoints in a private virtual network (PVN)." (Col. 10, ll. 42-43) The process is specifically for "[a]dding a new path[, which] may consist of the following activities and decision points" (col. 10, ll. 45-47), where the seventh step recited above is one of those activities and decision points. That is, Du, and thus Bhattal in view of Du, relies upon QoS requirements as to the process followed for *adding a new communication path* – not for the movement, startup, or shutdown of application computer programs, as in the claimed invention of claims 16 and 26. Therefore, Du, and thus Bhattal in view of Du, does not render the invention of claims 16 and 26 obvious.

Du, and thus Bhattal in view of Du, further is not properly modified to yield reliance upon QoS requirements for moving, starting up, or shutting down application computer programs, as in claims 16 and 26. There is no motivation for replacing the communication paths of Du for the application computer programs of the claimed invention insofar as reliance upon QoS

requirements is concerned. Indeed, Du, and thus Bhattal in view of Du, uses QoS requirements in responding to a customer's request as a business decision, performed by businesspeople (e.g., step 7 is to get the customer's approval) – Du, and thus Bhattal in view of Du, does not have its *software* use QoS requirements at all, unlike the invention of claims 16 and 26. Replacing the communication paths of Du with the application computer programs of claims 16 and 26 changes the focus of Du, still does not cause the computer software of Du to rely upon QoS requirements, and results in Du's process being unsuitable for responding to customer requests for new communication channels. Therefore, Du, and thus Bhattal in view of Du, cannot be modified to yield the limitations recited by claims 16 and 26.

Claims 2, 4-8, and 10-15

Claims 2, 4-8, and 10-15 have been rejected under 35 USC 103(a) as being unpatentable over Bhattal. Claims 2 and 8 are independent claims, from which claims 4-7 and 10-15 ultimately depend. Applicant submits that as amended, claims 2 and 8 are patentable, and claims 4-7 and 10-15 are patentable for at least the same reason as claims 2 and 8. Similar to claim 1, claims 2 and 8 each recite an application computer program being started up, shutdown, or moved. As discussed with respect to claim 1, however, Bhattal is limited to the startup, shutdown, or movement of *communication channels*, not computer programs. Therefore, for the reasons discussed above with respect to claim 1, Bhattal does not render claims 2 and 8 unpatentable.

Claims 3 and 9

Claims 3 and 9 have been rejected under 35 USC 103(a) as being unpatentable over Bhattal in view of Du. Claims 3 and 9 are dependent claims, depending ultimately from claims 2 and 8, and therefore are patentable for at least the reasons discussed above with respect to claims 2 and 8.

Conclusion

Applicant asks that all claims be allowed. No fee is believed to be due. Please apply any other charges or credits to Deposit Account No. 50-0967.

Respectfully Submitted,

October 4, 2004
Date

Scott R. Boalick
Reg. No. 42,337
Attorney/Agent for Applicant(s)

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2000

Application or Docket Number
09/864825
NAVY #79317

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	1	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	1 minus 20=	* 0
INDEPENDENT CLAIMS	1 minus 3 =	* 0
MULTIPLE DEPENDENT CLAIM PRESENT		☐

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
BASIC FEE	355.00	OR	BASIC FEE	710.00
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL		OR	TOTAL	710

CLAIMS AS AMENDED - PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 35	Minus	** 36	= 0
Independent	* 3	Minus	*** 3	= 0
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=	X	OR	X80=	✓
+135=		OR	+270=	✓
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

FORM PTO-875
(Rev. 8/00)

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
*U.S. GPO: 2000-490-709/30103

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/864,825	05/24/2001	Paul V. Werme	NC 83017	2372

23501 7590 07/02/2004

JAMES B BECHTEL, ESQ
NAVAL SURFACE WARFARE CENTER CODE CD222
17320 DAHLGREN ROAD, BLDG. 183, RM. 015
DAHLGREN, VA 224485100

EXAMINER	
TANG, KENNETH	
ART UNIT	PAPER NUMBER
2127	

DATE MAILED: 07/02/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)
	09/864,825	WERME ET AL.
	Examiner	Art Unit
	Kenneth Tang	2127

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 September 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-36* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-36* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-36 are presented for examination.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claims 1-36 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention:

a. In claim 1, "resource management device" is indefinite because it is not made explicitly clear in the claim language whether this device in the host or not. And if it is located in the host, it is not made clear which host.

b. Claim 1 recites the limitation "first information" in line 3. There is insufficient antecedent basis for this limitation in the claim.

c. Claim 1 recites the limitation "second information" in line 4. There is insufficient antecedent basis for this limitation in the claim.

d. In claim 2, "resource management device" is indefinite because it is not made explicitly clear in the claim language whether this device in the host or not. And if it is located in the host, it is not made clear which host.

e. In claim 2, "managed characteristic application" is indefinite because it is not made explicitly clear in the claim language who manages the characteristic application – whether it is managed by the resource management device or by something else, for

example. In addition, the term is not defined in the claim language and the definition is not made clear. Is this term an instance of a characteristic of an application, or an actual application?

f. In claim 2, "which one of starts up" is indefinite because it is grammatically incorrect and does not make any sense.

g. In claim 2, "additional copy" is indefinite because a first copy was never introduced so it is not made clear in the claim language why there is an additional copy.

h. In claim 2, "one of the host" is indefinite because it is grammatically incorrect and does not make any sense. In addition, only one host was introduced prior and the term implies that one host is being selected from a plurality of hosts.

i. In claim 2, "all applications" is indefinite because only one application was introduced prior.

j. Claim 2 recites the limitation "first information" in line 5. There is insufficient antecedent basis for this limitation in the claim.

k. Claim 2 recites the limitation "second information" in line 6. There is insufficient antecedent basis for this limitation in the claim.

l. In claim 8, "resource management device" is indefinite because it is not made explicitly clear in the claim language whether this device in the host or not. And if it is located in the host, it is not made clear which host.

m. In claim 8, "managed characteristic application" is indefinite because it is not made explicitly clear in the claim language who manages the characteristic application – whether it is managed by the resource management device or by something else, for

example. In addition, the term is not defined in the claim language and the definition is not made clear. Is this term an instance of a characteristic of an application, or an actual application?

n. Claim 8 recites the limitation "first information" in line 6. There is insufficient antecedent basis for this limitation in the claim.

o. Claim 8 recites the limitation "second information" in line 7. There is insufficient antecedent basis for this limitation in the claim.

p. Claim 8 recites the limitation "control configuration " in line 2. There is insufficient antecedent basis for this limitation in the claim.

q. In claim 16, "managed characteristic application" is indefinite because it is not made explicitly clear in the claim language who manages the characteristic application – whether it is managed by the resource management device or by something else, for example. In addition, the term is not defined in the claim language and the definition is not made clear. Is this term an instance of a characteristic of an application, or an actual application?

r. In claim 16, "N" and "M" are indefinite because it is not made explicitly clear in the claim language whether these are variables or whether they present static numbers or whether these are integer values, or whether N is a value that is different than M or not, etc.

s. Claim 16 recites the limitation "the state and health" in line 4. There is insufficient antecedent basis for this limitation in the claim.

t. Claim 16 recites the limitation "Quality of Service (QoS) " in line 8. There is insufficient antecedent basis for this limitation in the claim.

u. In claim 16, the term "Quality of Service (QoS) is indefinite because it is not made explicitly clear in the claim language the requirements involving this term.

v. As to claim 18, there are two different claims for claim 18 (see page 5 and 6). It is not known which of the two claims represents claim 18.

w. In claim 26, "N" and "M" are indefinite because it is not made explicitly clear in the claim language whether these integers are variables or whether they present static numbers or whether these are integer values, or whether N is a value that is different than M or not, etc.

x. In claim 26, "managed characteristic application" is indefinite because it is not made explicitly clear in the claim language who manages the characteristic application – whether it is managed by the resource management device or by something else, for example. In addition, the term is not defined in the claim language and the definition is not made clear. Is this term an instance of a characteristic of an application, or an actual application?

y. Claim 26 recites the limitation "the state and health" in line 8. There is insufficient antecedent basis for this limitation in the claim.

z. Claim 26 recites the limitation "Quality of Service (QoS) " in line 12. There is insufficient antecedent basis for this limitation in the claim.

aa. In claim 26, the term "Quality of Service (QoS) in line 12 is indefinite because it is not made explicitly clear in the claim language the requirements involving this term.

bb. Claim 26 recites the limitation "Quality of Service (QoS) " in line 12. There is insufficient antecedent basis for this limitation in the claim.

cc. The term "general-purpose" in claim 26 is a relative term which renders the claim indefinite. The term "general-purpose" is not defined by the claim, the specification does not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention.

dd. Claim 35 recites the limitation "Program Control" in line 3. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. **Claims 1 and 16-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bhattal et al. (hereinafter Bhattal) (US 20020064126) in view of Du et al. (hereinafter Du) (US 6,041,306).**

4. As to claim 1, Bhattal teaches in a distributed environment comprised of hosts instantiating copies of a application, a resource management device generating signals which

start up, shutdown or move a selected one of the copies responsive to first information regarding performance of all copies of the application and second information regarding performance of the hosts *([0014]-[0016])*. Bhattal fails to explicitly teach the application being scalable. However, Du teaches a scalable computing system being an advantageous feature *(col. 5, lines 16-34)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of scalable applications to the existing system of Bhattal because of the advantages described above.

5. As to claim 16, it is rejected for the same reasons as stated in the rejection of claim 8. In addition, Bhattal fails to explicitly teach having states to represent the health of the hosts and that the actions need to maintain Quality of Service (QoS) requirements, and that . However, Du teaches using managers and state machines that contains states of every condition and determines the state and health based on the conditions automatically *(col. 6, lines 65-67 through col. 7, lines 1-15 and col. 12, lines 31-47 and col. 10, line 3)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of using managers and state machines that contains states of every condition and determines the state and health based on the conditions because they increase the integrity of the system by being able to know when a "unhealthy" state (or state that is considered below requirements) occurs. In addition, Du teaches using QoS requirements for the applications *(col. 11, lines 16-18)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of QoS requirements for the applications in order to maintain or increase the integrity of the system.

6. As to claim 17, Bhattal teaches the software wherein the first function receives system specification information comprising selected ones of host configuration and capabilities, application capabilities, survivability requirements, scalability characteristics, application startup and shutdown dependencies, and application and path performance requirements *([0067])*.

7. As to claim 18, Bhattal teaches the software wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic applications, and detected failures regarding the N hosts *([0070] and [0080])*.

8. As to claim 19, Bhattal teaches wherein the first function receives application performance data representing each one of the M managed characteristic applications *(see Abstract)*.

9. As to claim 20, Bhattal teaches wherein the first function receives application performance data on all applications instantiated by the N hosts including performance data representing each one of the M managed characteristic applications *(see Abstract)*.

10. As to claim 21, Bhattal teaches the software wherein the second function which determines the required allocation and reallocation actions need to maintain the Quality of Service (QoS) requirements established for the M managed characteristic application by

responding to application and host failures by determining if and what recovery actions should be taken; determining if and where to place new copies of one of the M managed characteristic applications or which of the M managed characteristic applications should be shutdown when QoS Manager functions indicate that scale-up or scale-down actions are indicated based on measured application performance and application QoS specifications; determining where new applications should be placed when requested to do so by a program control device; and determining which and how many applications should run based on application system priorities *([0070] and [0080])*.

11. As to claim 22, Bhattal teaches the resource management device wherein the managed characteristic application comprises a scalable application *([0070] and [0080])*.

12. As to claim 23, Bhattal teaches the resource management device wherein the managed characteristic application comprises a fault tolerant application, where the degree of fault tolerance is selectable by a user *([0070] and [0080])*.

13. As to claim 24, Bhattal fails to explicitly teach wherein the managed characteristic application comprises a selectable priority application. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of a selectable priority application because it would increase the functionality of the system if applications can be selected by order of importance.

14. As to claim 25, Bhattal teaches using applications and using instances of communication channels but fails to explicitly teach the instantiating managed characteristic be of an application. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made that there could be applications associated with channels. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of applications associated with the channels for instantiating because it will increase the functionality of the channels from the applications performing functions for them.

15. As to claim 26, it is rejected for the same reasons as stated in the rejection of claim 16. In addition, Bhattal fails to explicitly teach monitoring the host and network resources and reporting. However, Du teaches monitoring the host and network resources and reporting *(col. 5, lines 59-67 through col. 6, lines 1-10, col. 8, lines 3-10)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of monitoring the host and network resources and reporting because the system could then diagnose or track the status or "health" which actions then can be taken to maintain or increase the integrity of the system.

16. As to claim 27, Bhattal teaches the software wherein the first function receives system specification information comprises host configuration and capabilities *([0067])*.

17. As to claim 28, it is rejected for the same reasons as stated in the rejection of claims 2 and 17.

18. As to claim 29, Bhattal teaches the software wherein the first function receives system specification information comprising path performance requirements regarding communication between at least two of the N hosts *([0006])*.

19. As to claim 30, it is rejected for the same reasons as stated in the rejection of claim 18.

20. As to claim 31, Bhattal teaches the software wherein the first function receives historical data regarding statuses, configuration, and loads of the N hosts and link statuses and loads regarding the network *([0006])*.

21. As to claim 32, it is rejected for the same reasons as stated in the rejection of claim 19.

22. As to claim 33, it is rejected for the same reasons as stated in the rejection of claim 20.

23. As to claim 34, it is rejected for the same reasons as stated in the rejection of claim 21.

24. As to claim 35, Bhattal teaches the software wherein the third function group makes decisions by one of: based on requests from Program Control, determining where new applications should be started; based on indication of application failure from the fourth function group, determining whether and where the failed application should be restarted; based on

indication of host failure from the fourth function group, determining whether and where the failed application previously instantiated by the failed one of the N hosts should be restarted; based on startup and shutdown dependency resolution requests from the fourth function group, determine whether and where additional applications should to be one of started and shut down prior to starting or shutting down another application; determining whether and where new applications need to be started and/or determine whether and which existing applications need to be shutdown *([0038])*.

Bhattal fails to explicitly teach wherein the managed characteristic application comprises a selectable priority application. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of a selectable priority application because it would increase the functionality of the system if applications can be selected by order of importance.

25. As to claim 36, it is rejected for the same reasons as stated in the rejection of claim 35 above. In addition, Bhattal fails to explicitly teach the application being scalable. However, Du teaches a scalable computing system being an advantageous feature *(col. 5, lines 16-34)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of scalable applications to the existing system of Bhattal because of the advantages described above.

26. **Claims 2, 4-8, and 10-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bhattal et al. (hereinafter Bhattal) (US 20020064126).**

27. As to claim 2, Bhattal teaches on a host instantiating a managed characteristic, a resource management device generating signals which one of starts up an additional copy of the managed characteristic on one of the host and a second networked host, shuts down and restarts the managed characteristic on the host, and moves the managed characteristic to the second host responsive to first information regarding performance of all applications including the managed characteristic application and second information regarding performance of the host *([0014]-[0016])*. Bhattal teaches using applications and using instances of communication channels but fails to explicitly teach the instantiating managed characteristic be of an application. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made that there could be applications associated with channels. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of applications associated with the channels for instantiating because it will increase the functionality of the channels from the applications performing functions for them.

28. As to claim 4, Bhattal teaches the resource management device wherein the managed characteristic application comprises a fault tolerant application, where the degree of fault tolerance is selectable by a user *([0070] and [0080])*.

29. As to claim 5, it is rejected for the same reasons as stated in the rejection of claim 2. In addition, Bhattal fails to explicitly teach wherein the managed characteristic application comprises a selectable priority application. However, it would have been obvious to one of

ordinary skill in the art at the time the invention was made to include the feature of a selectable priority application because it would increase the functionality of the system if applications can be selected by order of importance.

30. As to claim 6, Bhattal fails to explicitly teach the resource management device wherein the managed characteristic application further responds to user-initiated control actions. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of user-initiated control actions because more control will be given to the user.

31. As to claim 7, Bhattal teaches the program control device wherein the resource arrangement device generates signals instructing a program control device to modify the configuration of the managed characteristic application *([0038])*.

32. As to claim 8, it is rejected for the same reason as stated in the rejection of claim 2. In addition, Bhattal teaches there is a control configuration for the signals for the management device *([0008] and [0036])*.

33. As to claim 10, it is rejected for the same reasons as stated in the rejection of claim 4.

34. As to claim 11, it is rejected for the same reasons as stated in the rejection of claim 5.

35. As to claim 12, Bhattal teaches the resource management device wherein the resource management device further responds to third information regarding the performance of hardware operatively coupling the networked hosts *(see Abstract)*.

36. As to claim 13, it is rejected for the same reasons as stated in the rejection of claim 5.

37. As to claim 14, it is rejected for the same reasons as stated in the rejection of claim 6.

38. As to claim 15, Bhattal teaches the resource management device where the action requests are generated by an operator *([0002])*.

39. **Claims 3 and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bhattal et al. (hereinafter Bhattal) (US 20020064126) in view of Du et al. (hereinafter Du) (US 6,041,306).**

40. As to claim 3, Bhattal teaches in a distributed environment comprised of hosts instantiating copies of a application, a resource management device generating signals which start up, shutdown or move a selected one of the copies responsive to first information regarding performance of all copies of the application and second information regarding performance of the hosts *([0014]-[0016])*. Bhattal fails to explicitly teach the application being scalable.

However, Du teaches a scalable computing system being an advantageous feature *(col. 5, lines 16-34)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of scalable applications to the existing system of Bhattal because of the advantages described above.

41. As to claim 9, it is rejected for the same reasons as stated in the rejection of claim 3.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kenneth Tang whose telephone number is (703) 305-5334. The examiner can normally be reached on 8:30AM - 7:00PM, Monday through Thursday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (703) 305-9678. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MENG-AI T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Notice of References Cited	Application/Control No. 09/864,825	Applicant(s)/Patent Under Reexamination WERME ET AL.	
	Examiner Kenneth Tang	Art Unit 2127	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-2002/0064126	05-2002	Bhattal et al.	370/217
	B	US-2002/0174420 A1	11-2002	Kumar, Naveen	717/174
	C	US-6,041,306 A	03-2000	Du et al.	705/8
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

BEST AVAILABLE COPY

Index of Claims

Application No.	Applicant(s)
09/864,825	WERME ET AL.
Examiner	**Art Unit**
Kenneth Tang	2127

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim		Date
Final	Original	6/2/04
	1	√
	2	√
	3	√
	4	√
	5	√
	6	√
	7	√
	8	√
	9	√
	10	√
	11	√
	12	√
	13	√
	14	√
	15	√
	16	√
	17	√
	18	√
	19	√
	20	√
	21	√
	22	√
	23	√
	24	√
	25	√
	26	√
	27	√
	28	√
	29	√
	30	√
	31	√
	32	√
	33	√
	34	√
	35	√
	36	√
	37	
	38	
	39	
	40	
	41	
	42	
	43	
	44	
	45	
	46	
	47	
	48	
	49	
	50	

Claim		Date
Final	Original	
	51	
	52	
	53	
	54	
	55	
	56	
	57	
	58	
	59	
	60	
	61	
	62	
	63	
	64	
	65	
	66	
	67	
	68	
	69	
	70	
	71	
	72	
	73	
	74	
	75	
	76	
	77	
	78	
	79	
	80	
	81	
	82	
	83	
	84	
	85	
	86	
	87	
	88	
	89	
	90	
	91	
	92	
	93	
	94	
	95	
	96	
	97	
	98	
	99	
	100	

Claim		Date
Final	Original	
	101	
	102	
	103	
	104	
	105	
	106	
	107	
	108	
	109	
	110	
	111	
	112	
	113	
	114	
	115	
	116	
	117	
	118	
	119	
	120	
	121	
	122	
	123	
	124	
	125	
	126	
	127	
	128	
	129	
	130	
	131	
	132	
	133	
	134	
	135	
	136	
	137	
	138	
	139	
	140	
	141	
	142	
	143	
	144	
	145	
	146	
	147	
	148	
	149	
	150	

BEST AVAILABLE COPY

Search Notes	Application No.	Applicant(s)
	09/864,825	WERME ET AL.
	Examiner	Art Unit
	Kenneth Tang	2127

SEARCHED			
Class	Subclass	Date	Examiner
718	100-108	6/27/2004	KT
705	8	6/27/2004	KT
370	217	6/27/2004	KT

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
EAST	6/27/2004	KT

BEST AVAILABLE COPY

L Number	Hits	Search Text	DB	Time stamp
29	1024	scalable and (resource adj manag$5)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:39
30	981	scalable and (resource adj manag$5) and network$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:39
31	562	@ad<=20010524 and (scalable and (resource adj manag$5) and network$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:39
32	202	(start$3 same (shut$3 ends) same restart$3) with (application program)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:39
33	0	(scalable and (resource adj manag$5)) and @ad<=20010524 and ((start$3 same (shut$3 ends) same restart$3) with (application program))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:39
34	0	(scalable and (resource adj manag$5) and network$3) and @ad<=20010524 and ((start$3 same (shut$3 ends) same restart$3) with (application program))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:40
35	160	@ad<=20010524 and ((start$3 same (shut$3 ends) same restart$3) with (application program))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:40
36	0	(scalable and (resource adj manag$5)) and (@ad<=20010524 and ((start$3 same (shut$3 ends) same restart$3) with (application program)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:40
38	29	(start$3 same (shut$3 ends) same restart$3) with (copy instance)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:40
40	170	resumption with (copy instance)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:40
41	582	@ad<=20010524 and (scalable and (resource adj manag$5))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:40
42	0	(scalable and (resource adj manag$5)) and (resumption with (copy instance))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:40
43	148	@ad<=20010524 and (resumption with (copy instance))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:40
49	0	((US-20020064126-$).did.) and scalable	USPAT; US-PGPUB	2004/06/28 06:40
51	0	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020064126-$).did.) and scalable	USPAT; US-PGPUB	2004/06/28 06:40

BEST AVAILABLE COPY

52	104	scalable adj application	USPAT; US-PGPUB	2004/06/28 06:40
53	20660895	@ad<=20010524	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:40
55	0	((US-20020064126-$).did.) and performance	US-PGPUB	2004/06/28 06:40
37	2	(@ad<=20010524 and ((start$3 same (shut$3 ends) same restart$3) with (application program))) and (resource adj manag$5) and network$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:40
39	24	@ad<=20010524 and ((start$3 same (shut$3 ends) same restart$3) with (copy instance))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:40
44	38	(@ad<=20010524 and (resumption with (copy instance))) and network	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:41
45	25	((starts and resum$) with (copy instance)) same (manager controller)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:41
46	23	@ad<=20010524 and (((starts and resum$) with (copy instance)) same (manager controller))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:41
47	1	("20020064126").PN.	USPAT; US-PGPUB	2004/06/28 06:41
48	3	quality adj of adj service	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:41
50	4	(US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020064126-$).did.	USPAT; US-PGPUB	2004/06/28 06:41
54	64	(scalable adj application) and @ad<=20010524	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:41
56	1	((US-20020174420-$).did.) and performance	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:59
57	1	((US-20020064126-$).did.) and perform$	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:41
58	1	((US-20020064126-$).did.) and application	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:41
59	5	(US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.	USPAT; US-PGPUB	2004/06/28 06:43
60	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and monitor$3	USPAT	2004/06/28 06:52

BEST AVAILABLE COPY

61	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and monitor$3 and (status state)	USPAT	2004/06/28 07:33
62	1	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and qos	USPAT	2004/06/28 06:57
63	1	((US-20020174420-$).did.) and states	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/28 06:59
64	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and automat$	USPAT	2004/06/28 07:52
65	3	((US-6351744-$ or US-6041306-$ or US-6362836-$).did. or (US-20020174420-$ or US-20020064126-$).did.) and scal$	USPAT	2004/06/28 07:52

BEST AVAILABLE COPY

OIPE JC130 SEP 28 2001 PATENT & TRADEMARK OFFICE

3700 Receipt

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

2100

#5

In re Application of:

Paul V. Werme et al.

Serial No.: 09/864,825

Filing Date: May 24, 2001

For: RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR

Docket No. 83017

Group Art Unit: 2171

Examiner: not yet assigned

TC 2100 CSO STATUS INQUIRY RECEIVED
DEC 18 2001
Initials: PK

RECEIVED
DEC 17 2001
TC 3700 MAIL ROOM

REQUEST FOR "CORRECTED FILING RECEIPT"

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

The Receipt received for the above identified application contains an error. There should be four inventors listed. Please add Michael W. Masters, Fredericksburg, VA, Larry A. Fontenot, King George, VA and Lonnie R. Welch, Athens, OH to the list of inventors.

Please issue a "Corrected Filing Receipt" which reflects this correction.

Respectfully yours,

JAMES B. BECHTEL
Reg. No. 29,890
Customer No. 23501

RECEIVED
DEC 13 2001
TC 3700 MAIL ROOM

Attorney for Applicants
Office of Counsel, Code CD222
Naval Surface Warfare Center, Dahlgren Division
17320 Dahlgren Road, Building 183, Room 015
Dahlgren, VA 22448-5100
Tel: 540-653-8061
Fax: 540-653-7816

RECEIVED
FEB 25 2002
Technology Center 2100

Encl: Copy of Filing Receipt

CERTIFICATE OF MAILING

I hereby certify that this paper (along with any referred to as being attached or enclosed is being deposited with the United States Postal Service on the date shown below with sufficient postage as to be sent by First Class Mail in an envelope addressed to the Assistant Commissioner for Patents, Washington, DC 20231.

Date: September 25 2001

for Deborah A. Hughes
Beverly D. Cook

OIPE
SEP 28 2001
PATENT & TRADEMARK OFFICE

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

APPLICATION NUMBER	FILING DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	DRAWINGS	TOT CLAIMS	IND CLAIMS
09/864,825	05/24/2001	2171	710	NAVY CASE # 83017	17	1	1

CONFIRMATION NO. 2372

23501
JAMES B BECHTEL, ESQ
NAVAL SURFACE WARFARE CENTER CODE CD222
17320 DAHLGREN ROAD, BLDG. 183, RM. 015
DAHLGREN, VA 224485100

FILING RECEIPT

OC000000006330629

RECEIVED
FEB 25 2002
Technology Center 2100

Date Mailed: 07/23/2001

Receipt is acknowledged of this nonprovisional Patent Application. It will be considered in its order and you will be notified as to the results of the examination. Be sure to provide the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION when inquiring about this application. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please write to the Office of Initial Patent Examination's Customer Service Center. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections (if appropriate).**

Applicant(s)

Paul V. Werme, King George, VA;

Domestic Priority data as claimed by applicant

THIS APPLN CLAIMS BENEFIT OF 60/207,891 05/25/2000

Foreign Applications

If Required, Foreign Filing License Granted 07/23/2001

Projected Publication Date: To Be Determined - pending completion of Missing Parts

Non-Publication Request: No

Early Publication Request: No

RECEIVED
DEC 13 2001
TC 3700 MAIL ROOM
RECEIVED
DEC 17 2001
TC 3700 MAIL ROOM

Title

Resource allocation decision function for resource management architecture and corresponding programs therefor

Preliminary Class

707

Data entry by : LEE, KATY Team : OIPE Date: 07/23/2001

LICENSE FOR FOREIGN FILING UNDER
Title 35, United States Code, Section 184
Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Office of Export Administration, Department of Commerce (15 CFR 370.10 (j)); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

PLEASE NOTE the following information about the Filing Receipt:

- The articles such as "**a**," "**an**" and "**the**" are not included as the first words in the title of an application. They are considered to be unnecessary to the understanding of the title.
- The words "**new**," "**improved**," "**improvements in**" or "**relating to**" are not included as first words in the title of an application because a patent application, by nature, is a new idea or improvement.
- The title may be truncated if it consists of more than 500 characters (letters and spaces combined).
- The docket number allows a maximum of 25 characters.
- If your application was submitted under 37 CFR 1.10, your filing date should be the "date in" found on the Express Mail label. If there is a discrepancy, you should submit a request for a corrected Filing Receipt along with a copy of the Express Mail label showing the "date in."
- The title is recorded in sentence case.

Any corrections that may need to be done to your Filing Receipt should be directed to:

Assistant Commissioner for Patents
Office of Initial Patent Examination
Customer Service Center
Washington, DC 20231

SUBSTITUTE SPECIFICATION

OIPE JC139 SEP 19 2001 PATENT & TRADEMARK OFFICE

NCN-83017

RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR

STATEMENT OF GOVERNMENT INTEREST

The invention described herein was made in the performance of official duties by employees of the Department of the Navy or by researchers under contract to an agency of the United States government and, thus, the invention disclosed herein may be manufactured, used, licensed by or for the Government for governmental purposes without the payment of any royalties thereon.

BACKGROUND OF THE INVENTION

The present invention relates generally to resource management systems by which networked computers cooperate in performing at least one task too complex for a single computer to perform. More specifically, the present invention relates to a resource management system which dynamically and remotely controls networked computers to thereby permit them to cooperate in performing tasks that are too complex for any single computer to perform. Advantageously, software programs for converting a general purpose computer network into a resource managed network are also disclosed.

The instant application claims priority from Provisional Patent Application Serial No. 60/207,891, which was filed on May 25, 2000. The Provisional Patent Application is incorporated herein in its entirety by reference.

Resource Management consists of a set of cooperating computer programs that provides an ability to dynamically allocate computing tasks to a collection of networked computing resources (computer processors interconnected on a network) based on the

following measures:

- an application developer/user description of application computer program performance requirements;
- measured performance of each application programs;
- measured workload (CPU processing load, memory accesses, disk accesses) of each computer in the network; and
- measured inter-computer message communication traffic on the network.

Many attempts to form distributed systems and environments have been made in the past. For example, several companies and organizations have networked multiple computers to form a massively parallel supercomputer of sorts. One the best known of these efforts is SETI@home, which is organized by SETI (Search for Extraterrestrial Intelligence), a scientific effort aiming to determine if there is intelligent life out in the universe.

Typically, the search means the search of billions of radio frequencies that flood the universe in the hopes of finding another civilization that might be transmitting a radio signal. Most of the SETI programs in existence today, including those at UC Berkeley, build large computers that analyze that data from the telescope in real time. None of these computers look very deeply at the data for weak signals nor do they look for a large class of signal types. The reason for this is because they are limited by the amount of computer power available for data analysis. To extract the weakest signals, a great amount of computer power is necessary. It would take a monstrous supercomputer to get the job done. Moreover, SETI programs could never afford to build or buy that computing power. Thus, rather than use a huge computer to do the job, the SETI team developed software to use thousands of small computers, all working simultaneously on different parts of the analysis, to run the search routine. This is accomplished with a screen saver that can retrieve a data block over the internet, analyze that data, and then report the results back to SETI.

Several commercial companies are developing and implementing similar capabilities. Moreover, several companies, most notably IBM, have developed networks where each networked desktop computer becomes a parallel processor in a distributed computer system when the desktop computer is otherwise idle.

It will be appreciated that these approaches to computing in a distributed environment do not provide a system that is both flexible and adaptive (or at least easily adapted) to changes in system configuration, performance bottlenecks, survivability requirements, scalability, etc.

What is needed is a Resource Management Architecture which permits flexible control, i.e., allowing autonomous start up and shut down of application copies on host machines to accommodate changes in data processing requirements. What is also needed is functionality included in the Resource Management Architecture which permits the Resource Management Architecture to determine the near-optimal alignment of host and application resources in the distributed environment. It would be desirable to have a user-friendly technique with which to specify quality of service (QoS) requirements for each host, each application, and the network in which the hosts are connected. What is also needed is instrumentation to ensure that the specified QoS goals are being met.

SUMMARY OF THE INVENTION

Based on the above and foregoing, it can be appreciated that there presently exists a need in the art for a Resource Management Architecture, which overcomes the above-described deficiencies. The present invention was motivated by a desire to overcome the drawbacks and shortcomings of the presently available technology, and thereby fulfill this need in the art.

According to one aspect, the present invention provides, in a distributed environment

comprised of hosts instantiating copies of a scalable application, a resource management device generating signals which start up, shutdown or move a selected one of the copies responsive to first information regarding performance of all copies of the scalable application and second information regarding performance of the hosts.

BRIEF DESCRIPTION OF THE DRAWINGS

These and various other features and aspects of the present invention will be readily understood with reference to the following detailed description taken in conjunction with the accompanying drawings, in which like or similar numbers are used throughout, and in which:

FIGS. 1A, 1B collectively represent a high-level block diagram of hardware and software components implemented in the Resource Management System according to the present invention;

FIGS. 2A, 2B collectively represent a functional block diagram of the Resource Management Architecture according to the present invention;

FIG. 3 is a functional block diagram illustrating functional elements included in the system specification library (SSL) implementation of the Resource Management System according to the present invention;

FIG. 4 is a block diagram illustrating one technique for implementing the Resource (Application) Control functional group FG5 in FIGS. 2A, 2B using discrete software components;

FIGS. 5A, 5B represent a screen capture of a program control display FG54 generated by the software components illustrated in FIG. 4;

FIGS. 6A, 6B represent a screen capture of a host display generated by the Resource Management Architecture according to the present invention;

FIGS. 7A, 7B represent a screen capture of performance data regarding several of the hosts A - N included in FIGS. 6A, 6B;

FIGS. 8A, 8B represent a screen capture of a path display generated by the Resource Management Architecture according to the present invention;

FIGS. 9A, 9B represent a screen capture of the Resource Management Decision Review Display, which provides a summary of allocation and reallocation actions taken by the Resource Manager;

FIGS. 10A, 10B and 11A, 11B represent screen captures illustrating alternative, user-configurable displays generated from received data via standardized message formats and open interfaces;

FIGS. 12A, 12B represent a screen capture of an exemplary version of the Readiness Display FG66 according to the present invention;

Figs. 13A, 13B, and 13C are block diagrams which are useful in explaining various operational and functional aspects of the Resource Management Architecture according to the present invention; and

FIG. 14 is a high-level block diagram illustrating connectivity and data flow between the Hardware Broker and the other Resource Management and Resource Management-related functional elements in the Resource Management Architecture; and

FIG. 15 is a high-level block diagram of a CPU-based general computer which can act as a host in the Resource Management Architecture according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The Resource Management Architecture, which was and is being developed by the Naval Surface Warfare Center - Dahlgren Division (NSWC-DD), provides capabilities for monitoring hosts, networks, and applications within a distributed computing environment. Moreover, the Resource Management Architecture provides the capability of dynamically allocating, and reallocating, applications to hosts as needed in order to maintain user-specified system performance goals. Advantageously, the Resource Management architecture provides functionality for determining both how each component within the distributed environment is performing and what options are available for attempting to correct deficient performance, determining the proper actions that should be taken, and enacting the determined course of action. In addition to these capabilities, the architecture also allows for

operator control over creating and loading pre-defined static, dynamic, or combined static and dynamic system and/or host configurations. One particularly desirable feature of the Resource Management Architecture is that it provides capabilities for monitoring system performance along with the ability to dynamically allocate and reallocate system resources as required.

Before addressing the various features and aspects of the present invention, it would be useful to establish both terminology and the conventions that the instant application will follow throughout. In terms of terminology, a glossary section is presented below. In terms of conventions, this application includes information such as source code listing in an Appendix section. Since the source code itself is hundreds of pages, the Appendix section is divided into attached pages, e.g., Attached Appendix A, and an optical disk section, e.g., CD-Appendix N. Thus, while the appendices are listed in order, the reader must look to the signaling language to determine whether any particular appendix is actually provided in printed form.

API	**API (application programming interface)** A set of subroutines or functions that a program, or *application*, can call to invoke some functionality contained in another software or hardware component. The Windows API consists of more than 1,000 functions that programs written in C, C++, Pascal, and other languages can call to create windows, open files, and perform other essential tasks. An application that wants to display an on-screen message can call Windows' *MessageBox* API function, for example.

BNF	Acronym for `Backus Normal Form' (often incorrectly expanded as `Backus -Naur Form'), a metasyntactic notation used to specify the syntax of programming languages, command sets, and the like. Widely used for language descriptions but seldom documented anywhere, so that it must usually be learned by osmosis from other hackers.
DAEMON	A background process on a host or Web server (normally in a UNIX environment), waiting to perform tasks. Well-known examples of daemons are sendmail and HTTP daemon.
FUNCTION	A capability available on a host due to the presence of software (e.g., a program), a software module (e.g., an API), etc.
GLOBUS	Wide area network (WAN) enterprise management and control capability developed under DARPA sponsorship by USC/ISI.
HOST	A device including a central processor controlled by an operating system.
ICMP	Internet Control Message Protocol - ICMP is an extension to the Internet Protocol. It allows for the generation of error messages, test packets and informational messages related to IP. It is defined in STD 5, RFC 792.
JEWEL	An open-source instrumentation package produced by the German National Research Center for Computer Science
NFS	Network File System - A protocol developed by Sun Microsystems, and defined in RFC 1094, which allows a computer system to access files over a network as if they were on its local disks. This protocol has been incorporated in products by more than two hundred companies, and is now a *de facto* Internet standard.
QoS	Quality of Service

REMOS	Remos (REsource MOnitoring System)is a network bandwidth and topology monitoring system developed under DARPA sponsorship by CMU. Remos allows network-aware applications to obtain relevant information about their execution environment. The major challenges in defining a uniform interface are network heterogeneity, diversity in traffic requirements, variability of the information, and resource sharing in the network. Remos provides an API that addresses these issue by striking a compromise between accuracy (the information provided is best-effort, but includes statistical information if available) and efficiency (providing a query-based interface, so applications incur overhead only when they acquire information). Remos supports two classes of queries. "Flow queries" provide a portable way to describe a communication step to the Remos implementation, which uses its platform-dependent knowledge to return to the user the capacity of the network to meet this request. "Topology queries" reverse the process, with the Remos implementation providing a portable description of the network's behavior to the application.
SNMP	Simple Network Management Protocol Internet standard protocol defined in STD 15, RFC 1157; developed to manage nodes, e.g., hubs and switches, on an IP network.

An exemplary system for implementing the Resource Management Architecture according to the present invention is illustrate in FIGS. 1A, 1B, which includes a plurality of Host computers A, B, ..., N operatively connected to one another and Resource Management hardware RM via a Network 100. It will be appreciated that the hardware configuration illustrated in FIGS. 1a, 1B constitutes a so-called grid system. It will also be appreciated that the network 100 advantageously can be any known network, e.g., a local

area network (LAN) or a wide area network (WAN). It will also be appreciated that the hardware RM need not be a discrete piece of equipment; the hardware RM advantageously can be distributed across multiple platforms, e.g., the host computer(s), as discussed in detail below. In addressing the functional elements and applications in the distributed environment, it will be appreciated that hosts A-N each can instantiate applications 1-M. Thus, when all applications are being addressed, these applications will be denoted as A1-NM.

Still referring to FIGS. 1A, 1B, each of the hosts A, B, etc., preferably is controlled by an operating system (OSA, OSB, etc.), which permits Host A, for example, to execute applications A1 - AN, as well as an instrumentation daemon IDA, a Program Control (PC) agent PCA, and a Host Monitor HMA. It should be noted that instrumentation daemon IDA, PC agent PCA, and Host Monitor HMA are integral to the Resource Management Architecture while the operating system OSA and applications A1 - AN are well known to one of ordinary skill in the art.

In FIGS. 1A, 1B, the Resource Management Architecture RM advantageously includes an instrument collector 10 receiving data from all of the instrumentation daemons (IDA - IDN) and providing data to instrument correlator(s) 20, which, in turn, provide correlation data to corresponding quality of service (QoS) managers 30. Resource Management Architecture RM also receives data from host monitors HMA- HMN at history servers 40, which maintain status and performance histories on each of the hosts A - N and provide selected information to host load analyzer 50. Analyzer 50 advantageously determines the host and network loads for both hosts A-N and their connecting network 100 and provides that information to Resource Manager 60, which is the primary decision making component of the Resource Management Architecture. It will be appreciated that Resource Manager 60 also receives information from the QoS managers 30 and exchanges information with program controller 70. Program controller 70 sends startup and shutdown orders to the Program Control Agents based on operator or Resource Manager-initiated orders. It will be

appreciated that the operator-initiated orders are received via the one of the program control displays 80.

As will be discussed in greater detail below, the Resource Manager 60 is the primary decision-making component of the Resource Management Architecture. The Resource Manager 60 is responsible for determining:

- how to respond to host and application failures;
- where (i.e., which of hosts A -N) to place new applications;
- which applications to start up in response to the detection of a new host (host N+1);
- how to resolve application dependencies;
- what applications should be started, stopped, or moved in response to application system priority changes; and
- based on recommendations from the QoS Managers, when and where scalable application should be started or stopped.

Before leaving FIGS. 1A, 1B, is should be noted that the functions, e.g., instantiated programs or software program modules, in the Resource Management Architecture advantageously can be distributed across multiple platforms, e.g., multiple hosts (which may or may not be the illustrated Hosts A -N) or a grid system.

The major functional groups of the Resource Management Architecture according to the present invention are illustrated in FIGS. 2A, 2B. The functions illustrated as solid boxes are components of the Resource Management Architecture and are fully described below; the functions denoted by diagonal striping denote third-party software which has been integrated with the Resource Management Architecture but does not provide core functionality. Thus, the latter functions will be described only to the extent necessary to provide integration details. Moreover, it will be appreciated that the functions and

functionality of the Resource Management Architecture according to the present invention are interconnected to one another via middleware, which provides message passing interfaces between substantially all of the Resource Management functions. This middleware package, RMComms, is fully described below.

The major functional groups provided by the Resource Management architecture in an exemplary embodiment of the present invention are illustrated in FIGS. 2A, 2B. A summary of the functions provided by the Resource Management Architecture is available in Attached Appendix A. These functions, taken together, provide an integrated capability for monitoring and control of a distributed computing environment. In addition, many of the functions (and functional groups) within the Resource Management Architecture can also be run in a non-integrated configuration, thus providing subsets of the integrated Resource Management capabilities.

These function(al) groups illustrated in FIGS. 2A, 2B include:

FG1 - Host and Network Monitoring. This function group consists of software which monitors the host and network resources within the distributed environment. The function group collects extensive run-time information on host and network configuration, statuses, and performance. Run-time capabilities for discovering new hosts that have been started and for determining that existing hosts have gone down are also provided. Distribution of current and historical status and performance data to other components of the Resource Management Architecture is also provided. A more detailed discussion is provided below.

FG2 - Application-Level Instrumentation. The instrumentation function group provides general-purpose application event reporting and event correlation capabilities. Capabilities are provided for collecting and correlating application-provided data

such as application statuses, states, performance, and internally detected errors. Low-overhead (API) libraries are provided for applications to use in sending out key internal event and performance data. This application data is forwarded to other components of the instrumentation subsystem which collect data from applications on hosts throughout the distributed environment. The system also provides grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other Resource Management components.

FG3 - System Specifications. A specification language has been developed which allows the user to specify:

1) application software system structure, capabilities, dependencies, and requirements; and

2) hardware system (computer and network) structure, capabilities, and configuration.

Specification files, based on this specification language, are created by the user and provide the model of the software and hardware components of the distributed computing environment which is used by other Resource Management functions. The specification information is accessed by other Resource Management functions by linking in a specification parser library and making library calls to read in the files and convert them to an internal object model. Specific specification data items can then be retrieved via an object-oriented API. See the discussion below.

FG4 - Resource Allocation Decision-Making. This subsystem provides the reasoning and decision-making capabilities of the Resource Management architecture. The components of this subsystem use information from other subsystems in order to determine the health and state of the distributed environment and the options that are available for attempting to recover from faults or unacceptable performance. The

functions in this particular functional group make decisions regarding:

1) where new applications should be started;

2) whether and where failed applications should be restarted;

3) based on application inter-dependencies, whether and where additional applications should to be started prior to starting a particular application;

4) whether applications are meeting performance requirements and whether and where an application can be scaled up or moved when it is necessary to improve performance;

5) whether scalable applications are performing well within performance requirements and can be scaled down and which copy should be brought down; and

6) based on operator changes to application system priorities, whether and where new applications need to be started or whether and which existing applications need to be shut down.

FG5 - Application (Resource) Control. This subsystem provides application control (i.e., Program Control) capabilities which permit starting, stopping, and configuring applications on each of the hosts in the distributed environment. The subsystem provides both interactive operator control of the distributed environment as well as automatic control via configuration orders received from the Resource Allocation Decision-Making Subsystem (i.e., the Resource Manager component). The interactive controls allow an operator to create, load, save, and edit pre-defined system configurations (e.g., lists of applications that are to be run, with or without specific host mappings), determine the status and configuration of currently running programs, and start and stop any or all applications. Both static (operator-entered) mappings of applications to hosts and dynamic mappings of applications to hosts (where the Resource Allocation Decision-Making Subsystem will be queried to determine the proper mapping at run-time) can be defined. The subsystem also

provides application fault detection capabilities which are triggered by the unexpected death of an application that was started by the subsystem. A basic host fault detection capability is also provided which is triggered based on failure to receive heartbeat messages from subsystem components running on a particular host.

FG6 - **Displays.** The display subsystem provides capabilities for visualizing the status, performance, and health of the hosts, networks, and applications in the distributed environment. Capabilities are also provided for visualizing the status, performance, and health of the Resource Management components themselves.

As mentioned above, the RMComms middleware package provides the internal message passing interfaces between substantially all of the Resource Management functions both within each functional group and between the various functional groups. The middleware provides for automatic location-transparent many-to-many client-server connections. Low-overhead, reliable message passing capabilities are provided. Registration of message handler callback functions for specified requested message types is provided with the message handler functions being invoked when messages arrive. Registration of connection status callback functions, which are invoked when either new connections are made or existing connections are broken, is also provided. The middleware package also allows for multiple client and server objects to be instantiated in the same application, is thread-safe, and provides an easy-to-use object-oriented API through which all capabilities are accessed.

A detailed overview of each functional group and each function instantiated within each of the function groups FG1 - FG6 of the exemplary embodiment of the Resource Management Architecture illustrated in FIGS. 2A, 2B, including the capabilities provided by the functional group or function, will now be described in greater detail. The discussion below also includes an overview of the information flow between function blocks within the

same functional group and between function blocks in separate functional groups.

FG1 - Host and Network Monitoring Functional Group

Functional group FG1 provides extensive monitoring capabilities at the host and network levels. The information monitored includes statuses, configuration information, performance metrics, and detected fault conditions. By monitoring the individual hosts and network components within the distributed environment, the functional group FG1 determines:

- Accurate State and Performance Information, primarily by gathering the level of information necessary for accurately determining the state and health of each machine and network component.
- Distribution of Current Data to Resource Management Components by providing current performance and status information, either periodically or on request.
- Distribution of Historical Data to Resource Management Components, thus providing historical performance and status information on request.

It will be appreciated that the functional group FG1 makes these determinations by (or while) providing:

- Common Monitored Data Set and Formats, which permits functional group FG1 to gather the same set of statuses and statistics in the same formats for each host regardless of machine architecture or operating system.
- Minimally-Intrusive Data Collection Mechanisms, which permits functional group FG1 to gather the information in as non-intrusive a manner as possible (in terms of CPU utilization, network bandwidth utilization, etc...).
- Near Real-Time Data Collection Mechanisms, which permits functional group FG1 to gather the information in as timely a manner as possible.

The Host and Network functional group FG1 includes the four functions set forth below:

1) **Host Monitors** FG10A - FG10N, which reside on each respective machine in the distributed environment and collect extensive operating system-level data for each host A - N.

2) **History Servers** FG12A - FG12N, which collect data from the Host Monitors FG10A - FG10N, respectively, maintain status and performance histories on each host A - N in the distributed environment, i.e., in the Resource Management Architecture, and provide this information to displays and other functions with the Resource Management Architecture.

3) **Host Discovery Function** FG14, which uses Simple Network Management Protocol (SNMP) calls and ping Internet Control Message Protocol (ICMP) calls to determine when new hosts, e.g., host N+1, come on-line and if an existing host, e.g., host K, goes down.

4) **Remos Network Data Broker** Function FG16, which collects information on network link bandwidths from the SNMP-based Remos tool (developed by Carnegie Mellon University) and passes this information to the Host Load Analyzer function of the Resource Allocation Decision-Making functional group FG4, both of which are discussed in greater detail below.

Host monitors FG10A- FG10N, which monitor the status and performance of hosts A -N, respectively, are instantiated on each host machine within the distributed environment. Host Monitors FG10A - FG10N employ operating system-level mechanisms to retrieve status, configuration, and performance information on each host A - N. The information retrieved includes:

1) operating system version and machine configuration;

2) CPU configuration, status, and utilization;

3) memory configuration and usage;

4) network configuration, status, and utilization;

5) filesystem configuration, status, and utilization; and

6) process statuses including CPU, memory, network, and filesystem utilization for each process.

While Host Monitors FG10A - FG10N are primarily responsible for monitoring the status of a particular host, they also provide information on network load as seen by that particular host. In the same manner, the Host Monitors FG10A - FG10N also provide information and statistics concerning any remotely mounted filesystems, e.g., Network File System (NFS).

The information that the Host Monitors FG10A - FG10N collect advantageously can be formatted into operating system-independent message formats. These message formats provide a pseudo-standardized set of state, status, and performance information which is useful to other components of the Resource Management Architecture, i.e., other components do not have to be aware of or deal with the minor differences between data formats and semantics. It will be appreciated that since not all the state and performance data is available on every platform, in order to indicate which information is available, a group of flags are set in the host configuration message indicating whether specific data items are valid on a particular platform.

History Servers FG12A - FG12N are responsible for collecting information from the Host Monitors FG10A - FG10N and maintaining histories on the statuses, statistics, and performance of each host A - N in the distributed environment. This information advantageously can be requested by other functions instantiated in the Resource Management Architecture. Preferably, the primary consumers of the status information obtained by the History Servers FG12A - FG12N are the Host Load Analyzer (Hardware Broker) component of the Resource Allocation Decision-Making functional group FG4, the Host Display FG62A - FG62N and the Path Display FG64 of the Displays functional group FG6. The Host Load Analyzer FG40 receives information on host configuration and loads (primarily CPU,

memory, and network data) from History Servers FG12A - FG12N and employs this information to assign host fitness scores. Each Host Display, e.g., FG62A, receives and displays current status information on one of the hosts A - N, including process status information, and network connectivity information. Each Host Display can also request that a respective one of the History Servers FG12A - FG12N provide CPU load information, network load information, paging activity data, and memory utilization information, which is used to drive line graph charts for specific selected hosts.

It will be appreciated that History Servers FG12A - FG12N are designed so that multiple copies can be run simultaneously. Each of the History Servers FG12A - FG12N advantageously can be configured to either monitor all Host Monitors or to monitor only a selected set of Host Monitors. It should be mentioned at this point that the History Servers FG12A - FG12N determine the list of hosts in the distributed environment that could potentially be monitored from the System Specification Library. In this manner, the History Servers advantageously can be used to provide survivability (by having multiple History Servers connected to each Host Monitor) and/or to perform load-sharing (with the History Servers FG12A - FG12N each monitoring only a subset of the Host Monitors). It will also be appreciated that the History Servers FG12A - FG12N can be configured to periodically record history data to disk. These disk files can then be used for off-line analysis of the Resource Management Architecture.

The Host Discovery function FG14 employs Perl scripts in making SNMP and ICMP ping calls. These calls are used to periodically scan each subnet and host address in the distributed environment in an attempt to determine whether there have been any host status changes. In an exemplary case, the list of hosts and subnets that are to be monitored is read in from a file; alternatively, this information can reside in and be read from the System Specification Library, which is discussed in greater detail below.

It should be mentioned that when a new host is first detected, the new host's operating system configuration is queried by the Host Discovery function FG14 via SNMP calls. Information on the newly discovered host and its operating system configuration is then sent to the Program Control function FG50 in application control functional group FG5. Likewise, when a host fails to respond to multiple SNMP and ping queries, a message indicating that the host appears to have gone down is sent to the Program Control function FG50.

The final component of the Host and Network Monitoring functional group FG1 is the Remos Network Data Broker FG16, which receives information on network link bandwidth and network link bandwidth utilization from the SNMP-based Remos network monitoring tool mentioned above. The network information is accessed via the Remos application programming interface (API) library and is then sent on to the Host Load Analyzer (Hardware Broker) function FG40 of the Resource Allocation Decision-Making functional group FG4. The network information received from Remos consists of the maximum potential bandwidth and the current bandwidth utilization on specific host network links. As mentioned above, Remos network monitoring tool FG 16 is not a core component of the Resource Management Architecture; that being the case, no further details on either Remos or the Remos Network Data Broker are provided in the instant application.

FG2 - Application-Level Instrumentation Functional Group

The Instrumentation functional group FG2 advantageously provides general-purpose application event reporting and event correlation capabilities. The Instrumentation functional group permits instrumented application data to be easily accessible to other components of the Resource Management Architecture. The functional group provides capabilities for collecting and correlating application-provided data such as application statuses, states, performance, and internally detected errors. Low-overhead API's are provided that the

applications can use for sending internal event and performance data to the instrumentation components. The instrumentation functional group FG2 can collect data from applications on hosts A - N throughout the distributed environment. The functional group also provides grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other functional groups of the Resource Management Architecture.

The Instrumentation functional group provides:

- open API's and non-proprietary architecture
- near real-time monitoring support
- cross-language support: C, C++, Ada
- cross-platform support: Solaris, IRIX, Linux, etc...
- simple easy-to-use API's
- low-intrusive instrumentation interface
- instrumentation interface that does not significantly change the run-time behavior of the applications
- support for passing wide range of data types
- support for data marshalling / unmarshalling (system independent data formats)
- support for adding to or changing the information being instrumented without having to recompile portions of the architecture unaffected by the changes (preferably, no recompilation should be necessary expect for recompilation of the app being instrumented and any evaluation logic or displays that have been affected by the changes)
- scalable architecture (100+ hosts / 20+ apps per host / 5+ threads per app)
- ability for the architecture to perform auto-configuration as required
- ability to run multiple tests, multiple displays and multiple data logging components simultaneously

- ability to abstract away the underlying connectivity/communications between infrastructure components.
- ability for instrumentation infrastructure to be brought up and down while the application is running
- ability to easily build and configure new displays and data logging components (interactive configuration is preferable)
- ability to easily build and configure new performance and data correlation components (interactive configuration is preferable)
- backwards compatibility with existing Jewel Instrumentation displays (protect investments in existing display capabilities)
- backwards compatibility with existing Jewel Instrumentation function calls (provide ease of transition / backfit)

As illustrated in FIGS. 2A, 2B, the Instrumentation functional group FG2 includes the components enumerated below. In addition, Instrumentation APIs and Jewel Instrumentation will be addressed along with the Instrumentation functional group, i.e., the Instrumentation functional group includes:

1) **Instrumentation API Libraries** FG20 are linked with the applications and provide the function call interfaces by which these applications send instrumentation data.

2) **Instrumentation Daemons** FG22A - FG22N reside on each host in the distributed environment and are responsible for reading instrumentation data sent out by the applications, reformatting the data into instrumentation event messages and sending the messages to the Instrumentation Collectors.

3) **Instrumentation Collectors** FG24A -FG24N connect to the Instrumentation Daemons FG22A - FG22N on each host and receive instrumentation messages from host A - N. The Collectors forward received messages to the Instrumentation Correlators FG26A - FG26N and Instrumentation Brokers

FG28A - FG28N.

4) **Instrumentation Correlators** FG26A - FG26N receive instrumentation messages from the Instrumentation Collectors FG24A - FG24N and provide grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other functions of the Resource Management Architecture.

5) **Instrumentation Brokers** FG28A - FG28N receive instrumentation messages from the Instrumentation Collectors and perform task-specific reformatting and data manipulation for driving displays or other Resource Management components.

6) **Jewel Instrumentation Broker (QoS Monitor)** FG29 (a legacy component) receives instrumentation data from either the open source Jewel instrumentation package or from the Instrumentation Collectors. The QoS Monitor FG29 performs task-specific message reformatting and data manipulation for driving displays and the QoS Managers FG44A - FG44N.

The applications, e.g., A1-AN, link in the Instrumentation API Library FG20 and make API calls to construct and send out instrumentation event messages. Three separate APIs are provided for use by the applications:

1) a printf()-style API which allows the code to format, build, and send instrumentation data with a single function call;

2) a buffer-construction-style API where the multiple function calls are made to construct the instrumentation buffer iteratively, one data element per call; and

3) a Jewel function call API based on the existing API provided by the Jewel instrumentation package (an open-source package produced by the German National Research Center for Computer Science).

The first two APIs are the preferred programming interfaces and take advantage of several key instrumentation features while the Jewel API is provided solely for backwards

compatibility with existing instrumented application code and is implemented as a set of wrappers around the printf()-style API. All three APIs are supported for C and C++. ADA bindings have also been produced for the buffer-construction-style API and the Jewel function call API.

Preferably, the instrumented data is sent from the application to one of the Instrumentation Daemons FG22A -FG22N on a respective one of the hosts A - N where the application is running. The currently preferred mechanism for data transfer is via UNIX FIFO (first in - first out) IPC (inter-process communication) mechanisms. It will be appreciated that the FIFO mechanism was chosen based on reliability, low overhead, and ease of implementation. Alternative data passing mechanisms including shared message queues are considered to be within the scope of the present invention.

As mentioned above, an Instrumentation Daemon resides on each host in the distributed environment. The Instrumentation Daemon is interrupted whenever new data is written to the FIFO. The Instrumentation Daemon reads the data from the FIFO, reformats the data into the standard internal Instrumentation message format (discussed below), and sends the data to each of the respective Instrumentation Collectors FG24A -FG24N that are currently active. Alternatively, an event request filtering mechanism can be implemented so that specific event messages will only be sent to those ones of the Instrumentation Collectors FG24A -FG24N that have requested the message.

The standard instrumentation message format includes a header, a format string describing the application-provided data contained in the message, and the actual data values. The message components are illustrated in Attached Appendix B.

The Instrumentation Collectors FG24A- FG24N receive instrumentation messages from the Instrumentation Daemons FG22A - FG22N on each host A - N, respectively, in the

distributed environment. Currently, the Instrumentation Collectors FG24A- FG24N send every instrumentation message to all Instrumentation Brokers FG29A- FG29N and Instrumentation Correlators (Brokers) FG26A- FG26N that have connected to the Instrumentation Collectors FG24A- FG24N. The Instrumentation Collectors FG24A- FG24N serve as a pass-through server for instrumentation messages. The Instrumentation Collectors do support architecture scalability in the sense that without the Instrumentation Collectors, the Instrumentation Broker FG29 and Instrumentation Correlators FG26A- FG26N would need to maintain connections to the Instrumentation Daemons FG22A- FG22N on every host. As discussed above, an event request filtering mechanism advantageously can be implemented so that specific event messages will only be sent to those Instrumentation Brokers / Instrumentation Correlators that have requested the message.

Preferably, the Instrumentation Correlators FG26A- FG26N provide grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other components of the Resource Management Architecture. Each Correlator reads in a user-specified correlation grammar file which is interpreted at run-time by the Correlator's instrumentation correlation engine.

The Instrumentation Brokers FG28A- FG28N are task-specific applications built around a common code package. The Instrumentation Brokers FG28A- FG28N receive instrumentation messages from the Instrumentation Collectors FG24A- FG24N, filter all received instrumentation messages to find the messages of interest, and perform task-specific message data reformatting and manipulation for driving other components such as displays or other components of the Resource Management Architecture. This Instrumentation Broker approach permits instrumentation data sources to be quickly integrated for test, display, and debugging purposes.

It should be mentioned at this point that the Jewel Instrumentation Broker FG29

(hereafter referred to the QoS Monitor) is a legacy architecture component that served as a broker between the Jewel instrumentation package components and Resource Management components and displays. The QoS Monitor FG29 was responsible for polling the Jewel Collector components to retrieve application event messages. These messages were then reformatted and used to drive several displays and the QoS Managers FG44A - FG44N. The Jewel instrumentation package has now been replaced in all applications, however the message reformatting capabilities of the QoS Monitor have been maintained so that several displays and the existing QoS Manager interface do not have to be upgraded immediately. The QoS Monitor component has been modified so that it receives instrumentation data from both Jewel and the Instrumentation Collectors.

FG3 - SYSTEM SPECIFICATIONS FUNCTIONAL GROUP

Still referring to FIGS. 2A, 2B, it should be noted that a System Specification Language has been developed which allows the user to specify both (1) software system structure, capabilities, dependencies, and requirements, and (2) hardware system (computer and network) structure, capabilities, and configuration. System Specification Files, generally denoted FG32, which are based on this specification language, are created by the user and provide a model of the software and hardware components of the distributed computing environment which is used by the Resource Management Architecture. The language grammar advantageously can capture the following information related to the distributed environment and the applications that can run within the distributed environment:

- **Hardware and Operating Systems**
  - Hardware Configuration
  - Network Configuration
  - Operating Systems and Version
- **Software**
  - Systems, Subsystems, Paths, Applications, Processes

- Resource Requirements
- QoS Requirements (Events)
- Survivability Requirements
- Data Flow Path Information: Structure and QoS Requirements

It will be appreciated that the System Specification Language allows for grouping hardware and software components into systems and subsystems in order to create a hierarchy of components. Each application system and subsystem can be assigned a priority which is used at run-time to determine the relative importance of applications running in the distributed environment.

At the application level, the hardware, operating system, and other host requirements for each application can be specified along with information describing how to start up, configure, and shutdown the application. This information can include:

a) environment variables that need to be set;
b) the working directory for running the application;
c) the path(s) and file name of the application;
d) command-line arguments that should be set, including arguments that need to be resolved at run-time (e.g., the hostname where another application is running, the current date, the current userid, a unique run-time identifier number, etc...);
e) whether the application needs to run in an xterm;
f) whether a script file or signal should be run to shutdown the application; and
g) which script or signal should be used.

In addition, startup and shutdown dependencies between applications can be specified. Moreover, application states can be defined based on received instrumentation data values, the length of time an application has been running, and/or the set of processes that are currently running. Furthermore, for each application A1 - NM, the survivability and

scalability capabilities of the application can be specified. This latter information includes whether an application can be restarted if it fails, whether multiple copies of an application can be run, what type of scalability the application supports (e.g., Primary-Shadow, Load-Sharing, etc...), and the minimum and maximum number of copies that can be run. Moreover, an estimate of the amount of CPU, memory, and network resources that the application will use at run-time, advantageously can be specified.

At the host level, the operating system and version, the hardware architecture, the host's network interface name, and the SPEC organization's SPECfp95 and SPECint95 ratings for the host can be specified. At the network level, router and switch configurations and bandwidths can also be specified.

Moreover, application data flow paths can be defined including a graph of the data flow between applications along with performance requirements tied to one of more of the applications within the path. It should be mentioned that these defined requirements are named and are tied at run-time to Instrumentation Event data provided by the Instrumentation Correlators FG26A- FG26N. Monitoring of the performance requirements is the responsibility of the QoS Manager components FG44A - FG44N, as discussed in greater detail below.

As noted above, the System Specification Language provides a hierarchical structure for defining software and hardware systems. The current structure is shown below:

- **Software Specifications**
  - Application
    - Security
    - Configuration
      - Hardware Requirements
      - Startup Info

- Dynamic Arguments
- Shutdown Info
- States
- Dependencies
- Initial Load Estimate
- QoS Info
  - Survivability
  - Scalability

- **Hardware Specifications**
  - Host Info
  - Network Info
    - LANs
    - Network Devices (Interconnects)
- **Path Specifications**
  - Data Flow Graph
  - Data Flow Info
  - QoS Requirements

The specification information is accessed by linking in a specification parser library FG34 and making library calls to read in the files and convert them to an internal object model, and by making object access method calls to retrieve specific data items. The specification library is written in C++ and has been ported to all of the development platforms in the testbed. The library is currently being used by most of the Resource Management components, including Program Control FG50, the Resource Manager FG42, the QoS Managers FG44A -FG44N, the Hardware Broker FG40, and the History Servers FG12A - FG12N.

It should be mentioned that the software used to construct the API library consists of

(1) a parser file that defines the grammar (in BNF format), (2) a lexical analyzer file that defines the tokens of the language, and (3) a set of C++ System Specification classes for storing the specification file information. The lexical analyzer file is compiled with the GNU flex (lex) utility and the parser file is compiled using the GNU bison (yacc) utility. The flex and bison utilities create C source files which are then compiled along with the C++ System Specification object storage classes to create the System Specification Library (SSL) FG34. This library is then linked with the Resource Management applications. An overview of this structure is provided in FIG. 3; a more detailed discussion of the various functions are provided below.

FG4 - RESOURCE ALLOCATION DECISION-MAKING FUNCTIONAL GROUP

As illustrated in FIGS. 2A, 2B, the Resource Allocation Decision-Making functional group provides the reasoning and decision-making capabilities of the Resource Management architecture. The functions associated with this functional group employ information (listed below) to (1) determine the state and health of the distributed environment (hosts, networks, and applications), and (2) determine what allocation and reallocation actions need to be taken. The information provided to functional group FG4 includes:

- System Specifications:
  - Host configuration and capabilities
  - Application capabilities
    - Survivability
    - Scalability
    - Potential hosts to run on
  - Application startup and shutdown dependencies
  - Application and path performance requirements
- Program Control:
  - Application statuses

- Detected application faults
- Detected host failures
- Detection of new host
- Operator initiated requests
  - Resolution of application startup or shutdown dependencies
  - Selection of application-to-host mappings
- History Servers:
  - Host statuses, configuration, and loads
  - Network link statuses and loads
- Remos Network Data Broker:
  - Network link statuses and loads
- Instrumentation Subsystem:
  - Application performance information
- Readiness Display:
  - Run-time changes to application system priorities

The subsystem components make decisions based on the following triggers and data sources:

- Based on requests from Program Control, determine where new applications should be started
- Based on indication of application failure from Program Control, determine whether and where the failed applications should be restarted
- Based on indication of host failure from Program Control (or indirectly from Host Discovery), determine whether and where the failed applications should be restarted
- Based on application inter-dependencies defined in the System Specification Files, determine whether and where additional applications should to be started (or shut down) prior to starting (or shutting down) a particular

application

- Based on startup and shutdown dependency resolution requests from Program Control, determine whether and where additional applications should to be started (or shut down) prior to starting (or shutting down) a particular application
- Based on application instrumentation data and performance requirements defined in the System Specification Files, determine whether applications are meeting performance requirements and whether an application can be scaled up or moved to attempt to improve performance
- Based on application instrumentation data and performance requirements defined in the System Specification Files, determine whether applications are performing well within performance requirements and can be scaled down
- Based on operator changes to application system priorities, determine whether and where new applications need to be started and/or determine whether and which existing applications need to be shutdown
- Based on indication that a new host is on-line (from Host Discovery via Program Control), issue startup orders to bring up a Program Control Agent, Host Monitor, and Instrumentation Daemon on the new host which will bring the host under Resource Management control

The Resource Allocation Decision-Making functional group implements one of the three discrete functions listed below:

1) **Resource Manager** FG 42 is the primary decision-making component of the Resource Management Architecture. Resource Manager FG42 is responsible for determining (1) how to respond to host and application failures, (2) where to place new applications, (3) which applications to start up in response to the detection of a new host, (4) how to resolve application dependencies, (5) what applications should be started, stopped, or moved in response to

application system priority changes, and (6) based on recommendations from the QoS Managers FG44A - FG44N, when and where scalable application should be started or stopped.

2) **Host Load Analyzer** FG40 is responsible for assigning a set of fitness scores to each host based on host capabilities and loads.

3) **QoS Managers** FG44A -FG44N are responsible for monitoring application and path requirements as defined in the System Specification Files FG32 and recommending that applications be either scaled up, scaled down, or moved in order to maintain acceptable performance.

As mentioned above, the Resource Manager FG42 is the primary decision-making component of the Resource Management Architecture. It is responsible for:

(1) responding to application and host failures by determining if and what recovery actions should be taken;

(2) determining if and where to place new copies of scalable applications or which scalable applications should be shutdown when the QoS Managers indicate that scale-up or scale-down actions should be taken based on measured application performance;

(3) determining where new applications should be placed when requested to do so by Program Control; and

(4) determining which and how many applications should run based on application system (mission) priorities.

In order to accomplish these tasks, the Resource Manager FG42 maintains a global view of the state of the entire distributed environment including status information on all hosts A- N, network 100, and applications A1 -NM. In addition, the Resource Manager FG42 also calculates software and hardware readiness metrics and reports these readiness values, for display purposes, to the display functional group FG6.

It will be appreciated from FIGS. 2A, 2B that the Resource Manager FG42 receives status and failure information about hosts, networks, and applications from Program Control function FG50. This information includes both periodic status updates and immediate updates when statuses change such as a new host being detected or an application failing. In the case of application shutdown, information as to whether the application was shutdown intentionally or whether the application failed is also provided. Program Control function FG50 also issues requests to the Resource Manager FG42 when new applications need to be dynamically allocated and when the Program Control function FG50 determines that the Resource Manager FG42 needs to assess and attempt to resolve inter-application dependencies (such as one application which needs to be running prior to starting up another application).

The Resource Manager FG42 responds to faulted applications and hosts by determining whether the failed applications can and should be restarted and attempting to determine where (and if) there are hosts available that the application can run on. When a decision is made by the Resource Manager FG42, a message is sent to Program Control FG50 specifying what application to start and where to put it, i.e., which of hosts A - N to start the application on. The same general mechanism is used when Program Control FG50 requests that the Resource Manager FG42 determine where to start new applications and/or how to resolve inter-application dependencies; the Resource Manager FG42 responds with orders indicating what applications to start and where to start them. The Resource Manager FG42 advantageously can send application shutdown instructions to Program Control FG50 requesting that a certain application be stopped; this can occur when the QoS Managers FG44A-FG44N indicate that certain scalable applications have too many copies running or when application system priority changes (when an application changes from a high priority to a lower priority) occur resulting in scaling back the application system configuration.

The Resource Manager FG42 also receives host load and host fitness information on

all known hosts from the Hardware Broker (Host Load Analyzer) FG40. This information includes (1) overall host fitness scores, (2) CPU-based fitness scores, (3) network-based fitness scores, and (4) memory and paging-based fitness scores, along with (5) the SPEC95™ rating of the hosts. These scores are used by the Resource Manager FG42 for determining the "best" hosts for placing new applications when:

(1) responding to requests from the QoS Managers to scale up additional copies of an application;

(2) attempting to restart failed applications;

(3) responding to requests to dynamically allocate certain applications; and

(4) responding to application system (mission) priority changes which require scaling up additional applications.

The Resource Manager FG42 also receives requests from the QoS Managers FG44A-FG44N for scaling up, moving, or scaling down specific applications. The Resource Manager FG42 responds to these requests by determining whether the request should be acted upon and, if so, determines the specific action to take. The Resource Manager FG42 then issues orders to Program Control FG50 to start up or shutdown specific applications on specific hosts.

It should be noted that when the Resource Manager FG42 is first started, it reads in the System Specification Files FG32 (via calls to System Specification Library FG34) which contains the list of hosts that are known to be associated with the distributed environment and information on all applications that can be run in the distributed environment. The application-level information includes where, i.e., on which host, specific applications can be run, which applications are scalable, which applications can be restarted, and any dependencies between applications.

The Resource Manager FG42 currently responds to application system priority changes received from the Readiness Broker (translation software in or associated with the Readiness Display FG66) in the following manner:

(1) If the priority is changed to None, all applications associated with the specified system are shutdown.

(2) If the priority is changed to Low, all scalable applications within the specified system are scaled back to no more than 50% of potential maximum scalability and are not allowed to be scaled up past the 50% limit irregardless of performance.

(3) If the priority is changed to Medium, normal scaleup and scaledown functionality is allowed.

(4) If the priority is changed to High, all scalable applications are scaled up to at least 50% of potential maximum scalability and are not allowed to be scaled down to less than 50% irregardless of performance.

(5) If the priority is changed to Urgent, all scalable applications are scaled up to 100% (for maximum survivability) and are not allowed to be scaled down.

[Moreover, if the previous priority was None, and the new changed priority is higher than None, all required applications within the specified system are started up subject to the limitations outlined for each of the priority levels listed above.]

The Resource Manager FG42 also sends information about allocation and reallocation decisions to the Resource Management Decision Review Displays FG68A-FG68N, as discussed in greater detail below. Information on the decision that was made, what event the decision was in response to, and how long it took to both make the decision and implement the decision advantageously are also sent to the display functional group FG6. In addition, information about the alternative choices for where an application could have potentially been placed is also provided (if applicable); in an exemplary case, this information includes the host fitness scores for the selected host and the next best host choices which could have been selected.

As described above, the Resource Manager FG42 communicates with Program

Control FG50, the Hardware Broker FG40, the QoS Managers FG44A -FG44N, QoS Specification Control (not shown), the Readiness Broker of display FG66, the Globus Broker (e.g., message translation software (not shown)), and the RM Decision Review Displays FG68A- FG68N using the RMComms middleware, which will be discussed in greater detail below.

The Hardware Broker (Host Load Analyzer) FG40 is the host load analysis component of the Resource Management Architecture, which is primarily responsible for determining the host and network loads on each host A - N within the distributed computing environment. The Hardware Broker FG40 assigns a set of fitness scores for each host and periodically provides the list of fitness scores to the Resource Manager FG42.

The Hardware Broker FG40 advantageously receives operating system-level statuses and statistics for each host A- N from the History Server(s) FG12A-FG12N, respectively. This information can be employed for calculating CPU, network, memory, paging activity, and overall fitness scores for each of the hosts A-N. Preferably, the Hardware Broker FG40 periodically, e.g, once per second, provides the complete list of host fitness scores to the Resource Manager FG42.

It should be noted that when the Hardware Broker FG40 is first started, it reads in the System Specification Files FG32 (via calls to the System Specification Library (SSL) FG34), which files contain the list of hosts that are known to be in the distributed environment. The Hardware Broker FG40 also receives, e.g., reads in a file containing, information about the bandwidth and maximum packet sizes on all known network subnets in the distributed environment. It will be appreciated that this data advantageously can be used for converting host network load information based on packet counts to load information based on bytes per second and percentage of available bandwidth.

Periodically, e.g., approximately every three seconds, the Hardware Broker FG40 transmits a list of overall and network host fitness scores to the Hardware Broker Instrumentation Display which was constructed using the Graph Tool Instrumentation Display FG69A-FG69N. Moreover, the Hardware Broker FG40 advantageously can receive host-based network load data from the Remos Network Data Broker function FG16, which receives network data via the Remos Network Monitoring software 2. It should be noted that if Remos network data is available for any of the hosts A -N that are being monitored, the Remos reported network data advantageously can be used for calculating the network fitness score for that host, rather than using the host network data received from the History Server(s) FG12A-FG12N.

The QoS Managers FG44A - FG44N of functional group FG4 are responsible for monitoring application-level performance requirements. These requirements are defined in the System Specification Files FG32 and are monitored primarily via instrumentation data obtained directly from the application code. The QoS Managers FG44A - FG44N advantageously can determine if applications or application paths are meeting their assigned requirements. If an application is not meeting its performance requirements and the application is scalable (in the sense that multiple copies can be run and the copies will perform load-sharing across the copies), the QoS Managers FG44A - FG44N will either request that the Resource Manager FG42 scale up a new copy of the application or move the application to a new host (as an attempt to achieve better performance). Moreover, if there are multiple copies of a scalable application running, and all copies are performing well below the specified requirement threshold, the QoS Managers FG44A - FG44N will request that the Resource Manager FG42 shutdown a specific copy. It should be noted that the division of responsibility between the QoS Managers FG44A - FG44N and the Resource Manager FG42 is that the QoS Managers determine what actions would potentially improve performance, while the Resource Manager has final authority to determine whether to implement the requested action(s).

Each of the QoS Managers FG44A - FG44N can be scaled for both redundancy and for load-sharing. In an exemplary case, each copy of the QoS Manager monitors all of the requirements associated with a single application path defined in the System Specification Files FG32. It will be appreciated that the specific path to be monitored can be specified via command-line parameters. By default, without specifying a path via the command-line, the QoS Managers FG44A - FG44N will monitor all requirements for all paths defined in the System Specification Files FG32.

It should be mentioned that, in one exemplary embodiment, the QoS Managers FG44A - FG44N each employ a sliding window algorithm to determine when to declare that applications should be scaled up or scaled down. The inputs to the algorithm define both high and low sampling window sizes, the maximum number of allowed violations within the sampling window, and violation thresholds as a percentage of the actual specified requirement value. It should also be mentioned that the sliding window algorithm was selected in order to damp out unexpected "noise" or "spikes" in the measured performance data. Moreover, the threshold value as a percentage of the actual requirement value was selected in order to scale up, or scale down, prior to violating the specified hard requirement. The QoS Managers FG44A - FG44N provide application scale up and scale down requests to the Resource Manager FG42 when the measured performance data for an associated application violates either the high (scale up) or low (scale down) sliding window criteria for a specific requirement. A scale up request indicates which application on which host has violated the performance criteria, and a scale down request indicates which application on which host is recommended to be shutdown. It will be appreciated that the success of this algorithm is highly dependent on the rate of change and noisiness of the measured data.

Any of the QoS Managers FG44A - FG44N can also request that the Resource Manager FG42 move an application. This will occur in the case where one copy of an application is performing much worse than all other running copies of the same application.

In an exemplary case, the move request is implemented as a scale up request followed by a scale down request (of the badly performing copy). In that case, the scale down request does not get sent to the Resource Manager FG42 until the scale up action has been implemented. The QoS Managers FG44A - FG44N preferably employ application "settling times," defined in the System Specification Files FG32, to ensure that once a requested action has been sent to the Resource Manager FG42 that no additional actions are requested for that application until after the settling time has elapsed. It will be appreciated that this provides time for initialization and configuration among the application copies to occur. Alternatively, System Specification Language inter-application dependency definitions advantageously can be used instead of settling times.

The QoS Managers FG44A - FG44N also receive application status and state information from Program Control FG50, which periodically sends application status updates for all running applications and also sends immediate indications of any applications which have been started or stopped. This information is used by the QoS Managers FG44A - FG44N, along with the instrumented performance data being received via the QoS Monitor FG29 and Instrumentation Correlator FG34, to determine the exact state of all monitored applications that are running. This information is also used to determine when (and if) requested actions have been implemented by the Resource Manager FG42. The information is also used for setting up and discarding internal data structures used for monitoring the performance of each application A1-NM.

It will be appreciated that the QoS Managers FG44A - FG44N also receive application-level instrumentation data indicating current application performance values from the Instrumentation Correlators (Brokers) FG26A -FG26N, the Instrumentation Brokers FG28A-FG28N, and/or the Jewel Instrumentation Broker (QoS Monitor) FG29. The instrumentation data that is received contains (at a minimum) (1) the timetag when the data was generated, (2) the hostname and IP address of the host where the application that the data

is associated with is running, (3) the process id (pid) of the application that the data is associated with, and (4) the event number of the instrumentation message. Preferably, the event number of the instrumentation message specifies the type of instrumentation data that has been received; the hostname, IP address, and pid are used, in conjunction with the application data received from Program Control FG50, to determine the specific application that the data is associated with.

When the contents of the instrumentation message match any of the application performance requirements that are currently being monitored by the QoS Managers FG44A - FG44N, the data value is added to the proper requirement sliding window for the specified application. The sliding window algorithm is then checked to determine if the new sample triggered a violation of either the high or low sliding window. If a high threshold sliding window violation occurs and the application does not already have the maximum number of copies running, a determination is made as to whether performance can be best improved by starting a new application (scale up) or by moving an existing copy to a different host. The corresponding action recommendation will then be sent to the Resource Manager FG42. In an exemplary case, the criteria for determining whether an application should be moved rather than scaled up is based on relative performance of the replicated applications. More specifically, if one application is performing much worse [> 50%] than the other copies, the recommendation will be to move the application. Likewise, if the new sample triggers a low threshold sliding window violation and the application has more than the minimum number of copies running, a recommendation will be sent to the Resource Manager FG42 requesting that the copy of the application that is experiencing the worst performance be scaled down.

FG5 - RESOURCE (APPLICATION) CONTROL FUNCTIONAL GROUP

As discussed above, the Resource Control capabilities provided by the Resource Management Architecture consist of controlling application startup, configuration, and

shutdown on hosts within the distributed environment. This capability, known as Application Control or Program Control (hereafter referred to as Program Control) provides a powerful distributed configuration capability. The Program Control capabilities permit an operator to startup and control applications running on platforms throughout the distributed environment via an easy-to-use interactive display. These capabilities are provided by the Application Control functional group FG5.

More specifically, the Application Control functional group provides application control (i.e., Program Control) capabilities which permit starting, stopping, and configuring applications on each of the hosts in the distributed environment. The functional group provides both interactive operator control of the distributed environment as well as automatic control via configuration orders received from the Resource Allocation Decision-Making functional group FG4, i.e., the Resource Manager component. The interactive controls allow an operator to create, load, save, and edit pre-defined system configurations, e.g., lists of applications that are to be run, with or without specific host mappings, determine the status and configuration of currently running programs, and start and stop any or all applications. Both static (operator-entered) mappings of applications to hosts and dynamic mappings of applications to hosts (where the Resource Allocation Decision-Making functional group FG4 will be queried to determine the proper mapping at run-time) advantageously can be defined. The functional group also provides application fault detection capabilities which are triggered by the unexpected death, i.e., fault, of an application that was started by the functional group. A basic host fault detection capability is also provided which is triggered based on failure to receive heartbeat messages from functional group components running on a particular host.

A brief description of each function provided by the functional group FG5 is provided below; a detailed discussion of the Resource Control functional group FG5 and associated data flow will be provided in discussing FIG. 4.

1) **Program Control Agents** FG52A- FG52N: A Program Control agent generally denoted FG52 resides on each of the hosts A-N (i.e., PCA - PCN). Each agent is responsible for providing direct control over application startup and shutdown of applications on its respective host. The agent receives control orders from the Program Control function FG50 and is then responsible for implementing the orders. In an exemplary case, the agents implement the orders via system call mechanisms specific to the particular operating system. In addition, the agent also provides feedback to the Control function FG50 regarding the current status of all applications running on a particular host.

2) **Program Control FG50** - maintains the application state information for the Program Control functional group FG5. It also serves as the decision-making component of the Program Control functional group. The Control function FG50 receives application control (startup, shutdown, or configuration) requests from the Program Control Displays FG54A - FG54N and from the Resource Management functional group FG4. Using information from the Specification Files FG32, these high-level control function requests are dynamically translated into specific control orders which are sent to the individual Program Control agents FG52A -FG52N. The program Control FG 50 also provides application status and configuration information back to the Resource Manager FG42.

3) **Program Control Displays** FG54A - FG54N - serve as the GUI for interactive control of distributed applications. The Program Control Displays FG54A - FG54N allow an operator to see and control the status of applications running on each host in the distributed environment. The

Program Control Displays FG54A - FG54N also provide the user the ability to determine the status of each of the components of the Program Control architecture. Predefined scenario configurations defined in Program Control Configuration Files FG56 advantageously can be loaded and edited via the Displays. It should be mentioned that new Program Control Configuration Files can also be created and saved via the Displays. As illustrated in FIGS. 2A, 2B, Program Control Displays FG54A - FG54N can be run simultaneously with application status changes being reflected at each display.

4) **Configuration Files FG56** - contain an ordered set of applications that can be loaded at the Program Control display and then either edited or executed. The Configuration Files can contain both dynamic and static application-to-host mappings. For static application-to-host mappings, an application will, by default, be started on a specified host. For dynamic application-to-host mappings, the application will have a default host to start on but the Resource Manager FG42 will be queried at run-time to determine where the application actually should be placed. The Configuration Files FG56 also contain all information on how to start, stop, and configure an application, with the exception of environment variable settings for the application which are set based on the System Specification Files FG32.

It should be mentioned here that the Program Control functional group employs the application startup and shutdown information defined in the System Specification Files FG32. When an application entry is first created interactively at one of the Program Control Displays FG54A - FG54N, all of the startup and shutdown information for that application, as specified in the System Specification Files FG32, are loaded in as default settings. Once a configuration file entry has been created, all configuration information on the application

is read in from the configuration file except for the application environment variable settings which are still set based on the System Specification Files FG32.

As mentioned above, a Program Control agent resides on each host. The agent is responsible for providing direct control over application startup and shutdown. The agent receives control orders from the Control component and is then responsible for implementing the orders. Each of the PC Agents FG52A - FG52N implements application startup and shutdown orders via system call mechanisms specific to the particular operating system of the host. For example, on the Unix platforms, to start an application, the fork() and execv() function calls are used to create the application. The csh command is executed to start up the applications. Moreover, if the application needs to run in a console, an xterm is configured for the application to run in. In addition, if logging of either stdout or stderr is specified, the proper redirection operators are configured and the output log file is set to "/usr/tmp/<userid>_<appname>_<pid>.log". All environment variables needed by the application are also configured and passed in at the execv() call. The current working directory is also set by the chdir() command, and the new application is made a process group leader via the setpgid() function. Other operating systems invoke applications using different calls.

In order to stop an application on the Unix platforms, if a signal is to be sent to the application, the killpg() function is used, or else if a script or command is to be executed to shutdown the application, the csh command is executed (via the system() function) specifying the full path and executable name of the command along with any arguments for the command. It should be noted that if the application default shutdown time elapses and the application has not died, the respective one of the Program Control Agents FG52A-FG52N advantageously sends a SIGKILL signal to the application by calling killpg().

As illustrated in FIGS. 1A, 1B, the Program Control Agents (PCA-PCN)

advantageously can be instantiated on stand-alone hosts A - N. In that case, the Program Control Agents PCA-PCN (FG52A-FG52N in FIGS. 2A, 2B) send heartbeat messages to Program Control FG50 approximately once per second to indicate that they are still "up and running." Moreover, every ten seconds, the Program Control Agents PCA-PCN (FG52A-FG52N) send complete configuration information on all running applications to Program Control FG50. It should be noted that the terminology employed in FIGS. 1A, 1B differs from that in FIGS. 2A, 2B to emphasize the distinction between software instantiated on a host and a function provided by the Resource Management Architecture.

The Program Control function FG50 is the decision-making component of the Program Control functional group FG5. It maintains complete information on everything that is running across all platforms in the distributed environment. The Program Control function FG50 receives input data from PCA-PCN (FG52A-FG52N), the Program Control Displays FG54A-FG54N, the Resource Manager FG42, and the Host Discovery function FG14.

It will be appreciated from the preceding discussion that the Program Control FG50 provides startup and shutdown orders to the Program Control Agents FG52A-FG52N based on operator or Resource Manager-initiated orders. If the Program Control Agents report that an application has terminated abnormally, the Program Control FG50 provides a notification to the Resource Manager FG42, to the Program Control Displays FG54A - FG54N, and to any other component to which it is connected. When the Program Control function FG50 is first brought up, it can be configured to attempt to start Program Control agents on every host defined in the System Specification Files. The Program Control function FG50 will also attempt to start a Program Control Agent on a newly discovered host (discovered via the Host Discovery function FG14) if Host Discovery has been enabled on the Program Control Displays FG54A-FG54N.

The Program Control function FG50 also receives periodic heartbeat messages, e.g.,

once per second, from each of the Program Control Agents FG52A-FG52N, as discussed above. If Fault Detection has been enabled at the Program Control Displays FG54A-FG54N, if three consecutive heartbeat messages from an Agent, e.g., FG52A, are missed, the host that the agent is running on is declared down and all linked functions, including the Resource Manager FG42 and the Displays FG54A-FG54N are notified.

As mentioned above, the Program Control function FG50 sends out periodic application status updates as well as immediate notification when applications are started up, are shutdown, or fail. These notifications are sent out to all linked functions.

It should be noted that the Program Control function FG50 uses the same message traffic and internal processing for handling application startup and shutdown orders received from either the Resource Manager FG42 or from the Program Control Displays FG54A-FG54N. However, if a startup order received from one of the Program Control Displays FG54A-FG54N indicates that the Resource Manager FG42 should determine where to run the application, a request to allocate the application is sent to the Resource Manager FG42. When no response is received from the Resource Manager FG42 within a predetermined timeout period, the Program Control function FG50 will automatically start the application on the default host. Moreover, when an application startup cannot proceed due to an unfulfilled application startup dependency, a request will be made to the Resource Manager FG42 to attempt to resolve the dependency. If the Resource Manager FG42 either cannot resolve the dependency or no response is received within a predetermined timeout period, the application startup will fail, and a "dependency failed" indication will be sent to the Display. It will be appreciated that this will cause the application status to be displayed in, for example, yellow and post an alert to the Alert window on one of the Program Control Displays FG54A-FG54N.

Preferably, Program Control function FG50 also handles simple startup timing

dependencies between applications and will reorder a list of applications that were selected to be started simultaneously if doing so will resolve startup order dependencies between the applications. Otherwise, the Program Control function FG50 sends a request to the Resource Manager to attempt to resolve the dependencies.

The Program Control Display serves as the operator console for controlling the distributed environment. From the Display, shown in FIGS. 5A, 5B, the operator can:

1) see the status and configuration of currently executing applications A1-NM;
2) see the status of Program Control Agents PCA-PCN on each host A-N;
3) see and browse the application system structure defined in the System Specification Files FG32;
4) load configuration files FG56
5) save configuration files FG56
6) edit the configuration of applications that are not currently running;
7) create new application entries by dragging an application, application system, or application subsystem icon onto the application status area;
8) manually start specific applications;
9) manually stop specific applications;
10) manually start all applications that have the "Start All" flag set;
11) manually stop all applications;
12) turn host fault detection on or off (if on, loss of 3 consecutive heartbeats from a Program Control Agent will result in declaring the host down); and
13) turn host discovery on or off (if on, a new host message from the Host Discovery component will result in attempting to start up a Program Control Agent on the new host).

It will be appreciated from FIGS. 2A, 2B that multiple Program Control Displays FG54A-FG54N advantageously can be run simultaneously. If this is done, any configuration

change actions will be reflected on all the displays. Whenever application stop or start actions are taken by the display operator, a message is sent to the Program Control function FG50 which is responsible for enacting the start or stop action. The Program Control function FG50 also sends indications of any status changes to the Program Control Displays FG54A-FG54N as soon as the status changes are seen. In addition, periodic status updates are also sent to the Program Control Displays FG54A-FG54N.

The Program Control Configuration Files are text files that are read in by the Program Control Display when the operator wishes to load a new application configuration. A Configuration File is an ASCII file containing a list of applications. The format of an entry in a Configuration File is shown in Table 1 below.

Table 1

```
Application     TACFIRE:tacfire
Host            electra1
Display         umbriel1:0.0
Auto_Start      0
RM_Start        0
Console         1
Time_Delay      1
StartupDir      "$ENV_SIM_VERSION/TACFIREprocessor"
StartupExe      "$ENV_SIM_VERSION/TACFIREprocessor/tacfire"
StartupArgs     "-disport $DIS_PORT_NUM -cffhost %(HOSTNAME,
                AAW:Tactical_Sims:CFF_Broker)"
ShutdownExe     SIGINT
LogType         STDOUT
LogDir          "/usr/tmp"
```

The Configuration file advantageously can include the following fields:

1) The **Application** field, which identifies the full application name as defined in the System Spec. Files FG32 (i.e., System:Subsystem:Application).

2) The **Host** field, which is the desired or default host that this application should be started on.

3) The **Display** field, which is an optional field used when graphical display

output from an application needs to be rerouted to a display on a different host.

4) The **Auto_Start** flag, which identifies whether the application is to be started automatically if the "Start All" action is selected by the operator from the Program Control Display. (If the flag were set to "1", then the application would be started. If the flag were set to "0," it would not be started.)

5) The **RM_Start** flag, which identifies whether the Resource Manager should be queried at run-time to determine what host the application should be started on. The valid values are "0" for "NO" and "1" for "YES".

6) The **Console** flag, which identifies whether the application needs to be started in an Xterm window. The valid values are 0 for "NO" and 1 for "YES".

7) The **Time_Delay** field, which identifies how many seconds to wait after the previous application has been started before starting this application.

8) The **StartupDir** field, which identifies the current working directory that is to be set prior to starting up the application. This directory is usually the same as the directory where the executable for the application resides but does not have to be. As this example shows, environment variables may be used in the path.

9) The **StartupExe** field identifies the entire path and name of the application executable.

10) The **StartupArgs** field, which contains all the argument values needed for this particular application. As this example indicates, the argument values can be dynamically set at run time if needed. Environment variables may also be used within the argument list. In this example, the %(UNIQUE, 1, 40, Isis) argument would yield a number from 1 to 40 which is unique within a context named "Isis". Another resolution of %(UNIQUE, 1, 40, Isis) would yield a different number.

11) The **ShutdownExe** field, which identifies which signal defined within the

application that program control is to use to shutdown this application. Some examples would be SIGINT, SIGTERM, or SIGKILL. A shutdown script can also be used to shutdown the application. (In that case, there would be **ShutdownDir, ShutdownExe,** and **ShutdownArgs** fields listed. The usage for the shutdown fields would be used exactly the same as the startup fields.)

12) The **LogType** field, which identifies which outputs are to be written to the specified log file. The valid values are STDOUT, STDERR, and LOG_ALL. STDOUT is the normal output of the application (stdout). STDERR is the error output of the application (stderr). LOG_ALL writes both stdout and stderr outputs to the file.

13) The **LogDir** indicates the directory where the log file will be written. Again, environment variables may be used here. The log file name will be "<userid>_<appname>_<pid>.log" where <appname> is the full application name as specified in the **Application** field, <userid> is the userid of the current user under which the program control application is running, and <pid> is the system assigned process id of the application being executed.

FG6 - DISPLAY FUNCTIONAL GROUP

A number of displays which show system configuration data and instrumentation data in near real-time are included as part of the Resource Management Architecture. These displays support operator and user monitoring of the operation of the distributed environment including host and network statuses and performance, application system statuses and performance, as well as the status and performance of the other Resource Management architecture functions. Most of the displays use OpenGL and Motif, the latter being built with ICS's Builder Xcessory toolkit, and run on Silicon Graphics (SGI) platforms in an exemplary case. Several of the displays can also run on the Sun Solaris platforms. The displays that make up the display functional group FG6 include:

1) **Host Displays FG62A -FG62N.** Show layout of hosts along with host status, network connectivity, and process statuses.

2) **Path Display FG64.** Shows the status of applications in key end-to-end data flow paths along with performance and load graphs.

3) **Resource Management Decision Review Display FG68.** Shows a summary of allocation decisions made by the Resource Management system along with timing information and host fitness scores.

4) **Graph Tool Instrumentation Displays FG69A-FG69N.** Provides a user-configurable set of display widgets used for run-time monitoring of instrumented status and performance information.

5) **System Readiness Display FG66.** Shows the status of each hardware and software system, subsystem, and application defined in the System Specification Files and allow the operator to interactively change system and subsystem priorities.

FIGS. 6A, 6B represent a screen capture of an exemplary one of the Host Displays FG62A-FG62N, which provide graphical representations of various sets of the hosts A- N in the distributed environment. The Host Displays show the status of each host, host network connectivity, and the status of interesting processes running on the hosts. The Host Display operator can also select hosts shown on the Host Display and bring up real-time graphs of system performance for the selected hosts including CPU utilization, memory utilization, network packets in, network packets out, and paging activity. A screen capture of host specific performance information is provided in FIGS. 7A, 7B.

FIGS. 8A, 8B represent a screen capture of a representative Path Display FG64, generated by the Resource Management architecture, which shows the status of key system data flow paths consisting of multiple application stages. The number of copies of each application in the path is shown labeled with the host on which the application is running.

In addition, it should be mentioned that as many as three real-time graphs can be produced to depict run-time performance and load metrics related to the applications in the selected data path.

FIGS. 9A, 9B represent a screen capture of the Resource Management Decision Review Display FG68, which advantageously can provide a summary of allocation and reallocation actions taken by the Resource Manager FG42. For each action, timing information regarding how long it took the Resource Management functions, e.g., the Resource Manager FG42 and the Program Controller FG50, to both arrive at a decision and to enact the decided action are shown along with host fitness scores that were used in arriving at the allocation decision.

FIGS. 10A, 10B and 11A, 11B are screen captures of the Graph Tool Instrumentation Displays FG69A-FG69N, which depict user-configurable displays capable of receiving data via standardized message formats and open interfaces. The Graph Tool Displays FG69A-FG69N allow the operator to select and configure various display widgets (line graphs, bar charts, pie charts, meters, and text boxes) to build a desired display layout. Data sources for driving the widgets can also be selected interactively.

FIGS. 12A, 12B represent a screen capture of the System Readiness Display FG66, which advantageously can be a Java™ display with a CORBA™ interface. The display FG66 shows the status of each hardware system, host, application system, application subsystem, and application defined in the System Specification Files. The top portion of the display shows a summary status for each defined application system. It should be noted that the display operator can also change system and subsystem priorities and send the changed priorities to the Resource Manager function FG42.

As mentioned above, the RMComms middleware package provides object-oriented

client-server services for message communication between distributed applications and function modules. The middleware provides location transparency and automatic socket connections and reconnections between client and server applications. These services advantageously can be accessed through an object-oriented API which allows client and server objects to be easily created and exchange user-defined message data. The abstraction provided by the API allows the user to quickly and easily create distributed applications without needing to be aware of the details of the underlying network mechanisms. The RMComms middleware provides the following functions:

- provides location transparency between clients and servers
- provides a simple powerful object-oriented client-server API
- supports reliable transport of user-defined message data
- based on Berkeley sockets
    - uses TCP for message transport
    - uses UDP multicast for identification of new clients or servers
    - servers identified by unique assigned UDP/TCP port numbers
- provides general purpose callback function registration capabilities
    - user-specified message callback functions invoked when specified messages arrive
    - user-specified connection status callback function invoked when new client-server connections are established or existing connections are broken
- support for multi-threading
    - supports both polled and asynchronous I/O
    - thread-safe
- provides automatic connections between clients and servers
    - supports multiple client and server connections within the same application
    - provides automatic connections to new clients / new servers

- supports simultaneous many-to-many client-server connections
- no separate "naming service" or "application registration" components
- provides automatic client-server connection fault detection and recovery
  - provides fault detection mechanisms based on timeouts and broken connections
  - supports fault recovery via automatic reconnections between clients and servers
- provides basic support for data marshalling between machine architectures
  - byte-swapping
  - explicit message data type specification
  - all message data sent out using network byte order
- provides basic capabilities for reading the system clock and performing time conversions
- allows registration of user-defined signal (interrupt) handler functions
- layered object-oriented design and implementation
- cross-platform support:
  - SGI IRIX 6.3/6.4/6.5
  - Sun Solaris 2.5.1/2.6/2.7/.2.8
  - HP HP-UX 10.20
  - Linux 2.1/2.2
  - Windows NT 4.0
  - Windows 95/98/2000
  - Solarisx86 2.7
- C++ language support using native and GNU compilers

The RMComms middleware is implemented as a shareable object-oriented C++ library. The library provides four primary object classes, which are detailed in Attached Appendix C. It will be appreciated that the applications link with this library and can then

instantiate client and server objects for communicating with other local or remote applications. It should be mentioned that the application source code must also include a set of header files that allow connections between client and server objects, where each server type is assigned a server port number. For clients and servers that want to communicate, both the client and the server objects are created specifying the same server port number. Multiple servers of the same type can also be created, which all use the same server port number. This advantageously provides the ability for many-to-many client-server connections to be established, as illustrated in FIG. 4. Control of which servers the clients actually connect to is handled on the client side; clients can specify whether they wish to establish connections with all servers in the distributed environment, with a particular set of servers, or with all servers running on a particular set of hosts.

The operation of the Resource Management Architecture will now be described while referring to Figs. 13A-13C, which illustrate various operations in the distributed environment. More specifically, the Resource Management Architecture of the system illustrated in Figs. 13A includes hosts A-N, where host A provides a video source server application A-1, host B provides a video distribution application B-1, a contract application B-2, and a host load monitor B-3, and host C provides a display broker application C-1 applying video signals to a display driver C-2. It will be appreciated that host D is idle and that the connections between the various hosts constitute the network 100'. In addition, the Resource Management Architecture of FIG. 13A instantiates various functions, e.g., an instrumentation broker FG26', a QoS manager FG44', a resource manager FG42' and a program control FG50'. The instrumentation broker FG26' receives data from each of the applications running in the distributed environment, although only the lines of communication between the applications running on host B are actually depicted. From the discussion above, it will be appreciated that each of the applications is linked to an Instrumentation API.

Referring now to FIG. 13B, a QoS violation and its consequences is depicted. In particular, the Instrumentation broker FG26' provides data to the QoS manager FG44' which is indicative of a QoS violation. The QoS manager FG44' notifies the resource manager FG42' of the violation; the resource manager determines that duplicate copies of the applications running on host B are required and that these copies should be placed on host D. The resource manager FG42' transmits instructions to the Program Control function FG50', which starts copies of the running applications, i.e., a video distribution application D-1, a contract application D-2, and a host load monitor D-3, on host D. FIG. 13C illustrates shutdown of the application copies running on host B. It will be appreciated that this shutdown may be initiated responsive to the original QoS violation, another QoS violation, or a query from the user.

Having discussed the various functions and features of the Resource Management Architecture in gross, selected functions and features will now be described in detail. It will be appreciated that the discussion of the various functions will be signaled using the designations established with respect to FIGS. 2A, 2B.

FG42 - Resource Manager Function

As mentioned above, the Resource Manager FG42 is the primary decision-making component of the Resource Management functional group. It is responsible for:

(1) responding to application and host failures by determining if and what recovery actions should be taken;

(2) determining if and where to place new copies of scalable applications or which scalable applications should be shutdown when the QoS Managers indicate that scale-up or scale-down actions should be taken based on measured application performance:

(3) determining where new applications should be placed when requested to do

so by Program Control: and

(4) determining which and how many applications should run based on application system (mission) priorities.

In order to accomplish these tasks, the Resource Manager FG42 maintains a global view of the state of the entire distributed environment including status information on all hosts, networks, and applications. In addition, the Resource Manager FG42 also calculates software and hardware readiness metrics and reports these readiness values for display purposes.

The Resource Manager FG42 is an object-oriented multi-threaded application written in C++, which uses the RMComms middleware for all external communication. The Resource Manager FG42 communicates with the various software components instantiating the (1) Program Control FG50, 2) Hardware Broker FG40, 3) QoS Managers FG44A - FG44N, 4) QoS Specification Control FG29, 5) Readiness Broker in Readiness Display FG66, 6) Globus Broker (not shown), and 7) RM Decision Review Displays FG68A-FG68N.

It will be appreciated that the Resource Manager FG42 receives status and failure information about hosts and networks from the Host and Network Monitoring functional group FG1, and applications from the Program Control functional group FG5. This information includes periodic status updates as well as immediate updates when statuses change, e.g., when a new host is detected or an application fails. In the case of any application shutdown, information as to whether the applications were intentionally shutdown or whether the application actually failed advantageously can be provided. The Program Control function FG50 also issues requests to the Resource Manager FG42 whenever new applications need to be dynamically allocated and whenever the Program Control function FG50 determines that the Resource Manager FG42 needs to assess and attempt to resolve inter-application dependencies (e.g., one application which needs to be running prior to starting up another application).

The Resource Manager FG42 responds to applications faults and host failures by determining whether the failed applications can and should be restarted and attempting to determine where (and if) there are hosts available that the application can run on. When a decision is made by the Resource Manager FG42, a message is sent to Program Control function FG50 specifying what application to start and where to put it. The same general mechanism is used when the Program Control function requests that the Resource Manager FG42 determine where to start new applications and/or how to resolve inter-application dependencies; the Resource Manager FG42 responds with orders indicating what applications to start and where to start them. The Resource Manager FG42 advantageously can send application shutdown orders to the Program Control function FG50 requesting that a certain running application be stopped; this can occur when the QoS Managers indicate that certain scalable applications have too many copies running or when application system priority changes (to lower priorities) occur resulting in scaling back the application system configuration. See Figs. 13B and 13C and the associated discussion above.

The Resource Manager FG42 receives host load and host fitness information from the Hardware Broker (Host Load Analyzer) function FG40. This information includes overall host fitness scores, CPU-based fitness scores, network-based fitness scores, and memory and paging-based fitness scores along with the SPEC95 rating of the hosts. This information is received approximately once a second and includes information on all known hosts in the distributed system. These scores are used by the Resource Manager FG42 for determining the "best" hosts for placing new applications when:

(1) responding to requests from the QoS Managers to scale up additional copies of an application;

(2) attempting to restart failed applications;

(3) responding to requests to dynamically allocate certain applications; and

(4) responding to application system (mission) priority changes which require scaling up additional applications.

Advantageously, the Resource Manager FG42 also receives requests from the QoS Managers FG44A - FG44N for scaling up, moving, or scaling down specific applications. The Resource Manager FG42 responds to these requests by determining whether the request should be acted upon and, if so, determines the specific action to take and issues orders to the Program Control function FG50 to start up or shutdown specific applications on specific hosts. The QoS Managers FG44A - FG44N are responsible for monitoring specific system performance metrics (e.g., quality of service, or QoS, requirements) via instrumentation and determining if performance can be improved by scaling up or moving certain applications. When this occurs, the QoS Managers send a request to the Resource Manager FG42 indicating that a new copy of a specific application should be started. If the QoS Managers determine that the performance of a scalable application can be improved by moving an application, a scale up request is first sent to the Resource Manager FG42 and when the new application has been started, a scaledown request is then sent to the Resource Manager FG42. Moreover, when the QoS Managers FG44A -FG44N determine that there are more copies of scalable application running then are needed, requests to shutdown specific applications are sent to the Resource Manager FG42.

It will be appreciated that the Resource Management Architecture distributes functionality between the QoS Managers FG44A-FG44N and the Resource Manager FG42. Thus, the QoS Managers determine what actions would potentially improve performance, while the Resource Manager FG42 has final authority to determine whether to implement the requested actions.

It should be noted that when the Resource Manager FG42 is first started, it reads in the System Specification Files FG32 (via calls to the System Specification Library (SSL) FG34) which contains the list of hosts that are known to be (operating) in the distributed environment and information on all applications that can be run in the distributed

environment. The application-level information includes where specific applications can be run, which applications are scalable, which applications can be restarted, and any dependencies between applications. In addition, the Resource Manager FG42 receives updated application survivability specifications from the QoS Specification Control function. This information overrides the application survivability information that was initially loaded in from the System Specification Files FG32 for the specified application. The information is used by the Resource Manager FG42 to determine whether the specific application will be restarted if it fails at run-time.

It should also be noted that the Resource Manager FG42 sends application system and hardware system readiness and system (mission) priority information to the Readiness Broker, which is a translator withing the Readiness Display FG66 and to the Globus Broker (another Broker (not shown)). The Readiness Broker is responsible for driving a GUI/display FG66, which shows the current readiness data and allows the system (mission) priorities to be changed and sent back to the Resource Manager FG42. The Globus Broker provides basically the same functionality except that only a high-level subset of the readiness data provided to the Readiness Broker is provided to the Globus Broker. The readiness information sent to the Readiness Broker consists of readiness values for each application, application subsystem, and application system defined in the System Specification Files FG32. The scores advantageously can be based on the status (up/down) of the applications and the percentage of potential copies of scalable applications that are currently running. Host and network readiness scores are determined based on the host loads and host fitness scores received from the Hardware Broker FG40.

The Resource Manager FG42 also sends information about allocation and reallocation decisions to the RM Decision Review Display FG68 (FIGS. 9A, 9B). Information on the decision that was made, what event the decision was in response to, and how long it took to both make the decision and implement the decision are sent to the

display. In addition, information about the top choices for where an application could have potentially been placed is also sent (if applicable); this information includes the host fitness scores for the selected host and other hosts which could have been selected.

As described above, the Resource Manager function FG42 communicates with Program Control FG50, the Hardware Broker FG40, the QoS Managers FG44A -FG44N, QoS Specification Control (not shown - legacy function), the Readiness Broker of the Readiness Display FG66, the Globus Broker (not shown), and the RM Decision Review Display FG68 using the RMComms middleware. The message formats and contents of each message that is exchanged between the Resource Manager function FG42 and other functional elements of the Resource Management architecture are described in CD-Appendix D. The timing and/or event trigger for each message is also described.

FG40 - Host Load Analyzer (Hardware Broker) Function

The Hardware Broker FG40 provides the host load analysis function of the Resource Management functional group FG4. It is responsible primarily for determining the host and network loads on each host within the distributed computing environment. The Hardware Broker FG40 assigns a set of fitness scores for each host and periodically provides the list of fitness scores to the Resource Manager FG42. FIG. 14 illustrates the connectivity and high-level data flow between the Hardware Broker and the other Resource Management and Resource Management-related components.

The Hardware Broker FG40 is an object-oriented multi-threaded application written in C++, which uses the RMComms middleware for all external communication. It receives operating system-level statuses and statistics for each host from the History Server(s) FG12A –FG12N. This information is used for calculating CPU, network, memory, paging activity, and overall fitness scores for each host. The Hardware Broker periodically (once per second)

sends the list of host fitness scores to the Resource Manager FG42.

When the Hardware Broker FG40 is first started, it reads in the System Specification Files FG32 (via calls to System Specification Library (SSL) FG34) which contain the list of hosts that are known to be in the distributed environment. The Hardware Broker also reads in the file *networks.dat* which contains a list of information about the bandwidth and maximum packet sizes on known network subnets. It should be mentioned that this data is used for converting host network load information based on packet counts to load information based on bytes per second and percentage of available bandwidth.

It should be mentioned that there are two other RMComms interfaces that the Hardware Broker FG40 uses. Periodically (approximately every three seconds), the Hardware Broker FG40 sends a list of overall and network host fitness scores to the Hardware Broker Instrumentation Display FG69A – FG69N. As mentioned above, these displays were constructed using the Graph Tool described in the Instrumentation Graph Tool Display. Additionally, the Hardware Broker FG40 can receive host-based network load data from the Remos Broker FG16, which receives network data via the Remos Network Monitoring software (denoted 2 in FIGS. 2A, 2B). If Remos network data is available for any of the hosts that are being monitored, the Remos data is used for the network fitness score calculation for that host rather than the host network data received from the History Server(s).

The exemplary instance of the Hardware Broker FG40 is an object-oriented multi-threaded application. At the highest level, the Hardware Broker object contains the elements listed in Table II below. It will be noted that Table II contains a brief description of each of these objects. Additional details are provided in CD-Appendix E.

Table II

No.	Title	Description
1	Host Fitness Database object (FitnessDB class)	The Host Fitness Database object stores load history data and fitness score information for each host. The Host Fitness Database is updated and fitness scores are recalculated when new History Server Host Status Response Messages are received. For each host, a circular queue of host load history data (HostInstance class) is maintained with the newest data being placed at the end of the queue; this history data is used for recalculating host fitness scores. The Host Fitness Database also contains a System Specification Library (SSL) object which is used to access SPEC rating information for the hosts.
2	Signal Registration object (SignalRegistry class)	The Signal Registration object allows for a user-defined SIGINT signal handler to be registered in order to permit the Hardware Broker FG40 to be shutdown gracefully.
3	Network Subnet Information Database object (SubnetDB class)	The Network Subnet Information Database object is used to store IP address, maximum bandwidth, and MTU size for each network specified in the *networks.dat* file. This information is used for converting network packet load information to bytes/second network load information.

4	Remos Host Network Bandwidth Database object (RemosDB class)	The Remos Host Network Bandwidth Database object stores the latest Remos-reported network bandwidth information for each host being monitored. The information stored consists of available bandwidth as well as maximum potential bandwidth on a specific host network link. If Remos bandwidth information is available for a host and the latest data is less than 5 seconds old, the Remos data will be used for calculating the network fitness score for the host.
5	History Server Interface object (HistServInterface class)	The History Server Interface object inherits from the RMComms TCPCommClient class and is responsible for maintaining connections to the History Server(s), for registering status and message handler callback functions, for sending messages to the History Server(s), and for invoking the status and message handler callback functions when connections to History Servers are either established or broken or new messages are received from a History Server.

6	Instrumentation Graph Tool Display Interface object (InstrInterface class)	The Instrumentation Graph Tool Display Interface object inherits from the RMComms TCPCommServer class and is responsible for maintaining connections to the Graph Tool Display(s), for registering status and message handler callback functions, for sending messages to the Graph Tool Display(s), and for invoking the status and message handler callback functions when connections to Graph Tool Displays are either established or broken or new messages are received from a Graph Tool Display.
7	Resource Manager Interface object (ResMgrInterface class)	The Resource Manager Interface object inherits from the RMComms TCPCommServer class and is responsible for maintaining connections to the Resource Manager for registering status and message handler callback functions, for sending messages to the Resource Manager, and for invoking the status and message handler callback functions when connections the Resource Manager are either established or broken or new messages are received from the Resource Manager.

8	Remos Broker Interface object (RemosInterface class)	The Remos Broker Interface object inherits from the RMComms TCPCommClient class and is responsible for maintaining connections to the Remos Broker for registering status and message handler callback functions, for sending messages to the Remos Broker, and for invoking the status and message handler callback functions when connections the Remos Broker are either established or broken or new messages are received from the Remos Broker.

FG44: Quality-of-Service (QoS) Manager Function

The QoS Managers FG44A - FG44N are responsible for monitoring application-level performance requirements, which requirements are defined in the System Specification Files FG32 and are monitored primarily via instrumentation data obtained directly from the application code. The QoS Managers FG44A - FG44N advantageously determine if applications or application paths are satisfying their assigned requirements. When an application is not meeting its performance requirements and the application is scalable (in the sense that multiple copies can be run and the copies will perform load-sharing across the copies), the QoS Managers FG44A - FG44N will either request that the Resource Manager FG42 scale up a new copy of the application or move the application to a new host (which hopefully will result in better performance). Moreover, if there are multiple copies of a scalable application running, and all copies are performing below the specified requirement threshold, the QoS Managers FG44A - FG44N will request that the Resource Manager shutdown a specific copy.

The QoS Manager is a single-threaded application written in C/C++. It should be

noted that the application can be scaled for both redundancy and/or load-sharing. In an exemplary case, each copy of the QoS Manager monitors all of the requirements associated with a single application path defined in the System Specification Files FG32. It will be appreciated that the specific path to be monitored can be specified via command-line parameters. By default, without specifying a path via the command-line, the QoS Manager will monitor all requirements for all defined paths.

As mentioned above, the QoS Manager advantageously uses a sliding window algorithm to determine when to declare that applications should be scaled up or scaled down. The inputs to the algorithm define both high and low sampling window sizes, the maximum number of allowed violations within the sampling window, and violation thresholds as a percentage of the actual specified requirement value. It will be appreciated that the sliding window algorithm was selected in an effort to damp out unexpected "noise" or "spikes" in the measured performance data. Use of threshold value states as a percentage of the actual requirement value was selected in order to scale up, or scale down, prior to violating the specified hard requirement. It will be understood that the success of this approach is highly dependent on the rate of change and noisiness of the measured data..

Again, the QoS Manager uses the RMComms middleware for all external communication. Each copy of the Resource Manager talks to (1) Resource Manager FG42, (2) Program Control FG50, (3) QoS Specification Control (not shown), (4) QoS Monitor FG29, (5) Instrumentation Correlators FG26A –FG26N, (6) Graph Tool Instrumentation Displays FG69A-FG69N, and (7) History Servers FG12A-FG12N. In an exemplary case, the QoS Managers FG44A - FG44N advantageously can receive configuration orders from the Resource Manager FG42, which allows the Resource Manager FG42 to configure each QoS Manager to monitor specific application paths and also set the sliding window criteria to be used by each respective QoS Manager.

Each copy of the QoS Manager advantageously can transmit application scale up and scale down requests to the Resource Manager FG42 when the measured performance data for a respective application violates either the high (scale up) or low (scale down) sliding window criteria for a specific requirement. A scale up request indicates which application on which host has violated the performance criteria, and a scale down request indicates which application on which host is recommended to be shutdown. Each copy of the QoS Manager can also request that the Resource Manager move an application. This will occur in the case where one copy of an application is performing much worse than all other running copies. The move request is implemented as a scale up request followed by a scale down request (of the badly performing copy); the scale down request is not transmitted to the Resource Manager FG42 until the scale up action has been implemented.

The QoS Managers FG44A - FG44N use the application "settling times" defined in the System Specification Files to ensure that once a requested action has been sent to the Resource Manager that no additional actions are requested until after the application settling time has elapsed. This provides time for initialization and configuration among the application copies to occur. In future releases, the inter-application dependencies will be used instead.

The division of responsibility between the QoS Managers FG44A - FG44N and the Resource Manager FG42 is as follows:

(1) the QoS Managers FG44A - FG44N determine what actions would potentially improve performance; and

(2) the Resource Manager FG42 has final authority to determine whether to implement the requested actions

It should be mentioned that there is a Request Acknowledge message from the Resource Manager FG42 which has been defined and implemented within the QoS Manager

code. This message is intended to provide feedback to the QoS Manager indicating that the request had been successfully received and whether the Resource Manager FG42 intends to implement the request.

As previously mentioned, the QoS Managers FG44A - FG44N receive application status and state information from the Program Control function FG50. Program Control periodically sends application status updates for all running applications and also sends immediate indications of any applications which have been started or stopped. This information is used by the QoS Managers FG44A-FG44N, along with the instrumented performance data being received via the QoS Monitor FG29 and Instrumentation Correlators FG26A-FG26N, to determine the exact state of the monitored applications A1-NM that are running. This information is also used to determine when (and if) requested actions have been implemented by the Resource Manager FG42. The information is also used for setting up and discarding internal data structures used for monitoring the performance of each application.

The QoS Managers FG44A - FG44N also receive application-level instrumentation data indicating current application performance values from the Instrumentation Correlators FG24A-FG24N, the Instrumentation Brokers FG26A-FG26N, and/or the Jewel Instrumentation Broker (QoS Monitor) FG29. The instrumentation data that is received contains (at a minimum):

(1) the timetag regarding when the data was generated;

(2) the hostname and IP address of the host where the application that the data is associated with is running;

(3) the process id (pid) of the application that the data is associated with; and

(4) the event number of the instrumentation message.

The event number of the instrumentation message specifies the type of instrumentation data that has been received and the hostname, IP address, and pid are used, in conjunction with

the application data received from Program Control, to determine the specific application that the data is associated with.

If the contents of the instrumentation message match any of the application performance requirements that are currently being monitored by the QoS Manager, the data value is added to the proper requirement sliding window for the specified application. The sliding window algorithm is then checked to determine if the new sample triggered a violation of either the high or low sliding window. If a high threshold sliding window violation occurs and the application does not already have the maximum number of copies running, a determination is made as to whether performance can be best improved by starting a new application (scale up) or by moving an existing copy to a different host. The corresponding action recommendation will then be sent to the Resource Manager. In an exemplary case, the criteria for determining whether an application should be moved rather than scaled up is based on relative performance of the replicated applications. Thus, if one application is performing much worse [> 50%] than the other copies, the recommendation will be to move the application. Likewise, if the new sample triggers a low threshold sliding window violation, and the application has more than the minimum number of copies running, a recommendation will be sent to the Resource Manager FG42 requesting that the copy of the application that is experiencing the worst performance be scaled down.

It will be appreciated from the discussion above that when a copy of the QoS Manager is first started, it reads in the System Specification Files FG32 (via calls to System Specification Library (SSL) FG34), which contain the list of hosts that are known to be in the distributed environment and information on all applications that can be run in the distributed environment. The application-level information includes where specific applications can be run, which applications are scalable, which applications can be restarted, and any dependencies between applications.

It should also be mentioned that the Resource Manager FG42 receives updated application survivability specifications from the QoS Specification Control component. This information overrides the application survivability information that was initially loaded in from the System Specification Files for the specified application. The information is used by the Resource Manager FG42 to determine whether the specific application will be restarted if it fails at run-time.

As described above, the QoS Managers FG44A - FG44N communicates with the Resource Manager FG42, Program Control FG50, the QoS Specification Control (not shown), the QoS Monitor FG29, an Instrumentation Correlator (generally denoted FG24), a Graph Tool Instrumentation Display (generally denoted FG69), and the History Servers FG12A-FG12N using the RMComms middleware. The message formats and contents of each message that is exchanged between the QoS Managers FG44A - FG44N and these other functional components are described in greater detail in CD-Appendix F. Additional details regarding the timing and/or event trigger for each message is also described in the Appendix.

FG3: SYSTEM SPECIFICATION LANGUAGE & SYSTEM SPECIFICATION LIBRARY (SSL) FUNCTIONS

In order to effectively manage a pool of computing resources, the Resource Manager FG42 requires some means or mechanism of determining the capabilities and configuration of the computing resources under its control, as well as the software components that need to be executed and the dependencies of these software components on both hardware and software resources. Additionally, the Resource Manager FG42 requires the capability to determine the expected mission-level and application-level requirements. Furthermore, the Resource Manager FG42 must be able to determine what control capabilities are available to be used to attempt to recover from fault or QoS violation conditions.

In order to address these needs, a System and Software Specification Grammar has been developed to capture the "static" information needed by the Resource Manager FG42 for effectively managing a pool of distributed resources. The grammar captures the following information:

- **Hardware and Operating Systems**
  - Hardware Configuration
  - Network Configuration
  - Operating System and Version
- **Software**
  - Systems, Subsystems, Applications, Processes
  - Resource Requirements
  - QoS Requirements (Events)
  - Survivability Requirements
  - Path Information: Structure and QoS Requirements

As part of the grammar development effort, a specification library has also been developed that parses the specification files and provides an API for accessing the specification information. It will be noted that the specification library was written in C++ and has been ported for all development platforms including Solaris 2.6, Solaris 2.7, Irix 6.5, HP-UX 10.20, Red Hat Linux 6.0, and Windows NT 4.0. The library advantageously can be used by substantially all of the Resource Management functional elements, including Program Control FG50, Resource Manager FG42, Path QoS Managers, Hardware Broker FG40, and History Servers FG12A-FG12N.

As illustrated in FIG. 3, the API library consists of a yacc file FG302 that defines the BNF grammar, a lex file FG304 that defines the tokens of the language, and a set of C++ classes FG306 that store the spec file information. The lex file FG304 is compiled with the

GNU tool flex FG310 and it creates a C++ source file FG320. The GNU tool bison FG 312 compiles the yacc file FG302 and creates C++ source and header files FG322 and FG324. It will be noted that the lex source file FG304 includes the yacc header file FG322. The C++ compiler FG314 then compiles these two source files to create lex and yacc objects FG330 and FG332. The C++ compiler FG314 also compiles the C++ storage classes FG334. All of these objects are linked into a single library FG34 to be utilized by an application. FIG. 3 illustrates this process flow.

The Software Specifications Grammar (SSG) provides techniques for describing the characteristics and requirements of dynamic, path-based real-time systems as well as providing abstractions to describe the properties of the software, such as hierarchical structure, inter-connectivity relationships, and run-time execution constraints. The SSG also allows description of the physical structure or composition of the hardware such as LANs, hosts, interconnecting devices or ICs (such as bridges, hubs, and routers), and their statically known properties (e.g., peak capacities). Furthermore, the Quality-of-Service (QoS) requirements on various system components advantageously can be described.

At the highest level, a specification consists of a collection of software systems, hardware systems, and network systems. The language rules for specifying systems are described generally below and in detail in CD-Appendix G. The system specification language hierarchy is shown below; selected details will be presented immediately following.

- **Software Specifications**
    - Application
        - Security
        - Configuration
            - Hardware Requirements
            - Startup Info

- Dynamic Arguments
- Shutdown Info
  - States
- Dependencies
- Initial Load Estimate
- QoS Info
  - Survivability
  - Scalability

- **Hardware Specifications**
  - Host Info
  - Network Info
    - LANs
    - Network Devices (Interconnects)

- **Path Specifications**
  - Data Flow Graph
  - Data Flow Info
  - QoS Requirements

It will be appreciated that a software specification is a collection of software systems, each of which consists of one or more software subsystems. Specification files are provided by the developer to capture as much knowledge about their software system as possible. These files provide a model of the actual systems which can be used by the Resource Manager FG42 at run-time.

In contrast, an application is an executable program that can be started as an autonomous process on a host. Application attributes include all information necessary to

startup and shutdown the application. Associated startup block and the shutdown blocks describe how to start and stop the application and includes information such as the directory and name of the application, command line options, and environment variable settings.

An application instantiates an SSL object by calling its constructor. This parses the spec files in the specified directory and populates the object hierarchy to provide the data to the application. The SSL class contains an SSL_Container member, that holds the spec file data in its lists and maps. All the systems from the spec files are contained in the appropriate list, software systems in the swSysList, hardware systems in hwSysList, and network systems in nwSysList. The pathList contains all the paths in the spec files. The hostList contains all the hosts in the spec files; this list is also available from the entries in hwSysList. The processList contains a list of processes from the CONFIGURATION block. Moreover, it should be noted that one or more configuration blocks can exist per application. For example, an application that runs on more than one platform would have multiple CONFIGURATION blocks with different platforms in each HARDWARE block.

The application startup block contains all the information necessary to, automatically or manually, start an application. This information includes supported hardware (host) type, operating-system type, and operating-system version(s). This may be further constrained by an optional list of the names of hosts that can run the application. The startup information also includes the working directory for reading and writing data files, the name of the executable, and an ordered list of arguments that must be passed on the command line when the application is started. Last is a list of processes expected to be seen on the system when the application is running.

An application shutdown block indicates the command(s) to be used for termination of the application. A shutdown command may be a POSIX signal name or may be a shell script or batch file. Supported signals include SIGKILL, SIGQUIT, SIGHUP, SIGUSR1,

SIGUSR2, SIGSTOP, SIGINT, and SIGTERM. The ShutdownTime parameter is the maximum time to wait for the an application to die gracefully before forcing the application to terminate via the SIGKILL signal.

Other blocks are available. For example, a dependency block indicates any dependencies the application may have with the startup and/or shutdown of other applications (e.g., it may be required that a particular application be started before another application can be started). It will be noted that the dependency block is used by both Application Control FG50 and the Resource Manager FG42 to determine whether or not it is safe to start an application, stop an application, or let an application continue to run.

The scalability specification for an application indicates whether an application can be scaled via replication. Scalable applications are programmed to exploit load sharing among replicas, and can adapt dynamically to varying numbers of replicas. The specification also indicates whether an application combines its input stream (which may be received from different predecessor applications and/or devices), and splits its output stream (which may be distributed to different successor applications and/or devices) are also specified. "Combining" and "splitting" are commonly called "forking" and "joining" in parallel computing paradigms.

Specification files advantageously can be provided to describe a given set of networks that exist in a distributed runtime environment. A network system specification describes the LANs and ICs (interconnection devices such as switches, hubs and routers). A system consists of one or more subsystems. A subsystem may contain LANs (each with an associated peak bandwidth specification) and ICs (each containing a description of network membership).

Advantageously, a real-time QoS requirement specification includes timing

constraints such as simple deadlines, inter-processing times, and throughputs. A simple deadline is defined as the maximum end-to-end path latency during a cycle from the beginning to the end of the path. Inter-processing time is defined as a maximum allowable time between processing of a particular element in the path. The throughput requirement is defined as the minimum number of data items that the path must process during a unit period of time. Each timing constraint specification may also include items that relate to the dynamic monitoring of the constraint. These include minimum and maximum slack values (that must be maintained at run-time), the size of a moving window of measured samples that should be observed, and the maximum tolerable number of violations (within the window).

CD-Appendix G described a specification grammar for declaring requirements on applications in a dynamic, distributed, heterogeneous resource pool. The grammar allows the description of environment-dependent application features, which allows for the modeling and dynamic resource management of such systems.

A common API was developed to allow Resource Management functions access to the information contained in the spec files. This is an object oriented API is, in an exemplary case, written in C++, with libraries ported to all supported platforms. The object is populated by parsing the spec files using the BNF grammar defined by lex and yacc syntax and compiled with GNU tools flex and bison, as discussed above. Actual population occurs in the semantic actions of the yacc file.

The SSL_System class is a generic class that can hold data for a software system, hardware system, or network system. The type member describes the type of system it contains. It also contains a pointer to its parent (it allows for nested systems of the same type), and a name of the system. The sysList contains its SSL_System children, and compList contains a list of the system's components (a list of hosts, for a hardware system for example).

Preferably, the Application Program Interface (API) for the System Specification Library (SSL) FG34 uses the C++ Standard Template Library for data structures such as linked lists and hash tables (maps). An application first instantiates the SSL object by calling its constructor with the name of the directory where the specification files reside. This object contains functions that allow setting this directory after calling its constructor (setSpecDir(directory name)), clearing the object of all currently held data (clear()), parsing a specific file (parseSpec(filename)), and rebuilding the object (rebuild(), implicitly clears the object first). Once instantiated, this object provides access to the data in the specification files. CD-Appendix G provides additional discussion regarding this aspect of the SSL. It will be appreciated that the SSL object provides methods that return all the data it contains. For example, the getSWSystems returns an STL list of all the software systems specified in the specification files. Each entry in this list provides its data by methods such as getSysName(), and the set of application components (ApplicationSpec) that make up the system. All data can be retrieved in this manner.

FG1: HOST AND NETWORK MONITORING FUNCTIONAL GROUP

As mentioned above, extensive monitoring capabilities are provided in the Resource Management architecture at the host and network levels. The information monitored includes statuses, configuration information, performance metrics, and detected fault conditions. Moreover, the Host and Network functional group FG1 consists of four components including:

1) **Host Monitors** FG10A-FG10N, that reside on each machine in the distributed environment and collect extensive operating system-level data for each host (CPU and memory usage, etc) and provides it to the History Servers via the RMComms TCPCommServer middleware.

2) **History Servers** FG12A-FG12N that collect data from the Host Monitors, maintain status and performance histories on each host in the distributed

environment via an RMComms TCPCommClient, and provide this information to displays and other Resource Management components using an RMComms TCPCommServer.

3) A **Host Discovery** function FG14 that uses SNMP (Simple Network Management Protocol) calls and ping ICMP calls to determine when new hosts come on-line and if existing hosts go down and providing this information to Program Control via an RMComms TCPCommServer.

4) A **Remos Network Data Broker** FG16 that collects information on network link bandwidths from Carnegie Mellon University's SNMP-based Remos tool and passes this information by way of an RMComms TCPCommServer to the Host Load Analyzer component of the Resource Allocation Decision-Making subsystem.

It will be appreciated that Network information is collected by both the Remos broker FG16 and indirectly via the Host Monitors FG10A-FG10N. See FIGS. 2A, 2B. The Remos Broker FG16 accesses the Remos network information via the Remos API. As mentioned previously, Remos uses SNMP calls to the LAN switches and hosts. The Host Discovery function FG14 uses both SNMP and ICMP (ping) calls to each host A-N to determine if a new host(s) has (have) come on-line or previously discovered hosts have gone down. The Host Monitors FG10A-FG10N employ Operating System calls to gather host and network performance statistics. Internally, the History Servers FG12A-FG12N collect data from the Host Monitors FG10A-FG210N. The Monitoring functional group provides its information to the rest of the Resource Management components using RMComms TCPCommServer objects, which are discussed in detail elsewhere. The Remos Broker FG16 sends data to the Host Load Analyzer FG40, the History Servers FG12A-FG12N send data to the Display functional group FG6 and Host Load Analyzer FG40, and the Host Discovery function FG14 provides Program Control FG50 with information on detected or faulted hosts. Additional details on these functional elements are provided immediately below.

FG10A-FG10N Host Monitors

For monitoring the status and performance of hosts, a Host Monitor process runs on each machine within the distributed environment. These Host Monitors FG10A-FG10N use operating system-level mechanisms to retrieve status, configuration, and performance information of each host A-N. The information retrieved includes 1) operating system version and machine configuration, 2) CPU configuration, status, and utilization, 3) memory configuration and usage, 4) network configuration, status, and utilization, 5) filesystem configuration, status, and utilization, and 6) process statuses including CPU, memory, network, and filesystem utilization for each process. While the Host Monitors are primarily responsible for monitoring the status of a particular host, they also provide information on network load as seen by a particular host. In the same manner, the Host Monitors FG10A-FG10N also provide information and statistics concerning any remotely mounted filesystems (e.g., NFS).

Preferably, the information the Host Monitors FG10A-FG10N collect is formatted into operating system-independent message formats. These message formats attempt to provide a pseudo-standardized set of state, status, and performance information which is useful to other components of the Resource Management architecture and such that other components do not have to be aware of or deal with the minor deltas between data formats and semantics. Since not all the state and performance data is available on every platform, to indicate which information is available, a group of flags are set in the host configuration message indicating whether specific data items are valid on a particular platform.

It will be appreciated that the Host Monitors FG10A-FG10N have a very specific interface with the History Servers FG12A-FG12N. It periodically (once a second) sends its data to all History Servers connected to it (this is transparent, a property of the RMComms TCPCommServer); the History Server makes no requests to the Host Monitors.

More specifically, the Host Monitors FG10A-FG10N have been designed and implemented in C++. This decision allows for a completely modular design in which platform-specific code can be restricted to a small number of modules. This approach alleviates any of the problems associated with porting to various platforms. Currently there is support for Sun SPARC based architectures running Solaris 2.6 and 2.7, Silicon Graphics MIPS based architectures running IRIX 6.5., Hewlett Packard PA-RISC based architectures running HP1020, and Pentium based architecture running both WinNT 4.0 Workstation and Red Hat Linux 6.0. The Host Monitor source compiles under the native compilers provided by Sun Microsystems and Silicon Graphics for their respective platforms. The Gnu C++ compiler (version 2.8.1) may also be used on Hewlett Packard PA-RISC based architectures under HP-UX 10.20 and Red Hat Linux. Microsoft Visual C++ compiles the Windows NT Host Monitor. All Host Monitors utilize the I/O library package supported by the Resource Management (RM) group under the NSWC's High Performance Distributed Computing (HiperD) initiative.

The Host Monitors FG10A-FG10N accumulate data on a periodic interval specified at invocation. System process table data is accumulated and then filtered to eliminate "uninteresting" processes (usually meaning processes belonging to user ID 0 or 1). It is important to note that system-wide data is accumulated and processed *before* the filtering stage, so as to insure a complete picture of system-wide performance. This system-wide data, along with the filtered process list, is then made available to the I/O module for subsequent transmission to client applications.

FG12A-FG12N: History Servers

The History Servers FG12A-FG12N are responsible for collecting information from the Host Monitors and maintaining histories on the statuses, statistics, and performance of each host in the distributed environment. This information can be requested by other

Resource Management functional group. Currently, the primary consumers of the status information are the Host Load Analyzer (Hardware Broker) FG40 component of the Resource Allocation Decision-Making functional group FG4, the Host Display(s) FG62A-FG62N, and the Path Display FG64. The Host Load Analyzer FG40 receives information on host configuration and loads (primarily CPU, memory, and network data) and uses this to assign host fitness scores. The Host Displays FG62A-FG62N receive and display current host status information, process status information, and network connectivity information. It should be mentioned that the Host Display can also request that the History Servers provide CPU load information, network load information, paging activity data, and memory utilization information which is used to drive line graph charts for specific hosts selected at the Host Display.

The History Servers FG12A-FG12N are designed so that multiple copies can be run simultaneously. Each History Server can be configured to either monitor all Host Monitors FG10A-FG10N or to monitor only a selected subset of the Host Monitors. It will be noted that the History Servers FG12A-FG12N determine the list of hosts in the distributed environment that could potentially be monitored from the System Specification Library (SSL). In this manner, the History Servers FG12A-FG12N can be used to provide survivability (by having multiple History Servers FG12A-FG12N connected to each Host Monitor) and/or to perform load-sharing (with the History Servers FG12A-FG12N each monitoring only a subset of the Host Monitors). The History Servers FG12A-FG12N can also be configured to periodically record history data to disk. These disk files can then be used for off-line analysis.

The History Server function of Resource Management acts as a data broker between daemons monitoring individual hosts, known as host monitors FG10A-FG10N, and other functional components of Resource Management. The host monitors collect performance information (such as CPU utilization and process status data) from hosts of various platforms

(SGI, SUN, HP, Windows NT, and Linux). The host monitors use a RMComms TCPCommServer object to distribute this data. For further information, refer to the host monitor and RMComms documentation. The History Server s FG12A-FG12N collect and store this data from the host monitors FG10A-FG10N and distribute it to other Resource Management Clients, such as the Host Displays FG62A-FG62N, Graph Display FG69A-FG69N, Path Display FG64, and the Hardware Broker. FG40

Each History Server has two modes of operation relating to fault tolerance, scalability, and workload distribution between multiple instances of History Servers. The first mode determines at initialization (through command line arguments or default) the set of hosts to monitor, and this set remains static for the life of the History Server process. The second mode recognizes the existence of other History Server processes and coordinates between them. It allows for dynamic changing of the set of hosts each History Server monitors (example: two History Servers each monitoring half of the hosts, a third History Server starts, and all three History Servers reconfigure to each monitor one third of the hosts.) This also allows History Servers to preserve the data it collected by sending it to the others, providing fault tolerance.

The History Server function is written in C++ with an object-oriented design. The main routine processes the command line arguments, retrieves the list of hosts to monitor using an SSL object, instantiates the main History_Server object, and spawns the Collector, Distributor, Communicator, and Display thread. These threads share the main History_Server object. The Collector thread is responsible for collecting and storing data from the host monitors. The Distributor thread processes requests from RM Clients. The Communicator thread waits for events with other History Servers and takes appropriate actions, including triggering the Display thread to update the History Server Display.

FG14 - Host Discovery

The Host Discovery function FG14 advantageously can use a Perl script that makes SNMP (Simple Network Management Protocol) calls and ICMP ping calls. These calls are used to periodically scan each subnet and host address in the distributed environment to attempt to determine whether there have been any host status changes. In an exemplary case, the list of hosts and subnets that are to be monitored is read in from a file.

The host discovery FG14 issues MIB-II SNMP queries to obtain information on the hosts A-N on the network. When a new host is first detected, the new host's operating system configuration is queried via SNMP calls. Information on the newly discovered host and its operating system configuration is then sent to the Program Control function FG50. Likewise, when a host fails to respond to multiple SNMP and ping queries, a message indicating that the host appears to have gone down is sent to the Program Control function.

The Host Discovery function FG14 interfaces with Program Control FG50 using a C++ wrapper class around the Perl script. This wrapper class contains an RMComms TCPCommServer, making the data collected by the SNMP calls available to the rest of the Resource Management components.

FG16 - Remos Network Data Broker

The final functional component of the Host and Network Monitoring functional group is the Remos Network Data Broker FG16 which receives information on network link bandwidth and network link bandwidth utilization from the SNMP-based Remos network monitoring tool, as shown in FIGS. 2A, 2B and/or FIG. 14. The network information is accessed via the Remos API library and is then sent on to the Host Load Analyzer (Hardware Broker) function FG40 of the Resource Allocation Decision-Making functional group FG4 using an RMComms TCPCommServer. Remos works by using SNMP to query the switches (via the bridge collector) to collect information on network configuration as well as

bandwidth utilization on each link and also issues SNMP MIB-II queries to each host to collect the host's view of network utilization. The network information received from Remos consists of the maximum potential bandwidth and the current bandwidth utilization on specific host network links.

The Remos Broker FG16 provides the following information about the network link for each host. The data is sent to the Host Load Analyzer (Hardware Broker) approximately every 2 seconds. The Remos Broker FG16 uses configuration files listing specific hosts and switches that should be queried.

The functions implemented by Host Monitor functional group FG1 have been designed to provide a system monitoring capability not normally supplied by standard SVR4 or BSD Unix services. Such services include cross-platform reporting of system process loading, CPU performance, network performance and periodic status summary reporting. The Host Monitors were developed to support efforts by the HiperD Resource Management group, attempting to provide a common set of OS level parameters useful for assessing host and network load and status, for supporting resource allocation/reallocation algorithms, and attempting to provide a minimally intrusive, close to real-time capability for gathering this data.

Host Discovery Design

The Host Discovery function FG14 of the Resource Management architecture provides resource discovery of hosts on a network. It identifies new hosts that come online or previously known hosts that have gone offline. The Host Discovery component can determine the hostname, the operating system name and version, and in some cases the machine architecture and manufacturer of a newly discovered host. This information is sent to Program Control so the new host can be added to the pool of resources.

The Host Discovery functional element FG14 consists of a Perl script that contains the resource discovery functionality, and a C++ object that receives the output of the Perl script and provides this information to Program Control via an RMComms TCPCommServer connection. This is described in CD-Appendix H. More specifically, the Perl script *host_discovery.pl* issues ICMP (ping) calls and MIB-II SNMP queries to discover new hosts. On initialization, the script populates a data structure called *Net_info* for each of the networks (subnets) it needs to monitor. Currently this information is hard-coded, the subnet is defined as 172.30.1, and the lower and upper limits for the host are 1 and 254 respectively. It then initializes the global variables for the server host and port, network domain, and the executable path for the ping (fping) command

The *host_discovery.pl* script establishes a baseline of existing hosts using the current set of hosts that answer the fping call. For each network/subnet defined in its list of *Net_info* (Net_info.pm) data structures, it calls fping and builds a list of IP addresses of hosts that answered the ping, known as reachable hosts, and a list for those hosts that did not answer the ping. For each reachable host, a *Host_info* (Host_info.pm) data structure is populated to store the host's information. (Key fields in the *Host_info* data structure include IP address, hostname, operating system and version, architecture class, and manufacturer.) Since the IP address of the reachable host is known, a call to gethostbyaddr() is used to get the hostname. Other information for the host is obtained by making a MIB-II (Management Interface Base version 2) system Group (Object ID 1.3.6.1.2.1.1.1.0) SNMP call to the SNMP agent on each reachable host. This SNMP query returns information on the configuration of a specific network device (in this case, the configuration of each reachable host).

The *host_discovery.pl* script makes SNMP calls by using subroutines freely available for public use (freeware), created by Simon Leinen. These subroutines are contained in the files BER.pm and SNMP_Session.pm. The *SNMP_Session* is configurable for specifying timeouts and number of retries before declaring a host unavailable, and for specifying the

SNMP Object Id (OID).

Additional general and specific details regarding functional elements of the Host and Networking functional group FG1 are provided in CD-Appendix H.

FG2: Instrumentation functional group

As mentioned above, the NSWC-DD Instrumentation System provides general-purpose application event reporting and event correlation capabilities. The Instrumentation system forms an architecture that allows instrumented application data to be easily accessible by other components of the Resource Management architecture. The major functional components of the Instrumentation System architecture are the following:

1) The **Instrumentation API Libraries,** which are linked with the applications and provide the function call interfaces by which the application sends instrumentation data.
2) An **Instrumentation Daemon**, one copy of which resides on each host in the distributed environment and is responsible for reading instrumentation data sent by the applications, reformatting the data into instrumentation event messages and sending the messages to the Instrumentation Collectors.
3) The **Instrumentation Collectors**, which connect to the Instrumentation Daemons on each host and receive instrumentation messages from all hosts. The Collectors forward received messages to the Instrumentation Correlators and Instrumentation Brokers.
4) The **Instrumentation Correlators,** which receive instrumentation messages from the Instrumentation Collectors and provide grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other Resource Management components.

5) The **Instrumentation Brokers**, which receive instrumentation messages from the Instrumentation Collectors and perform task-specific reformatting and data manipulation for driving displays or other Resource Management components.

6) The **Jewel Instrumentation Broker (QoS Monitor)**, which is a legacy component that can receive instrumentation data from either the open source Jewel instrumentation package or from the Instrumentation Collectors. The QoS Monitor performs task-specific message reformatting and data manipulation for driving displays and the QoS Managers.

Instrumentation API Library

The applications link in the Instrumentation API Library and make API call to construct and send out instrumentation event messages. Three separate APIs are provided for use by the applications: 1) a printf()-style API which allows the code to format, build, and send instrumentation data with a single function call, 2) a buffer-construction-style API where the multiple function calls are made to construct the instrumentation buffer iteratively, one data element per call, and 3) a Jewel function call API based on the existing API provided by the Jewel instrumentation package (an open-source package produced by the German National Research Center for Computer Science). The first two APIs are the preferred programming interfaces and take advantage of several key new instrumentation features. It will be appreciated that the Jewel API is provided solely for backwards compatibility with existing instrumented application code and is implemented as a set of wrappers around the printf()-style API. All three APIs are supported for C and C++. Ada bindings have been produced for the buffer-construction-style API and the Jewel function call API.

The instrumented data is sent from the application to the Instrumentation Daemon on

the same host. The current mechanism for data transfer is via UNIX FIFO IPC (inter-process communication) mechanisms. The FIFO mechanism was chosen based on reliability, low overhead, and ease of implementation. Future implementations of the Instrumentation system may explore alternate data passing mechanisms including shared message queues.

Instrumentation Daemon

An Instrumentation Daemon resides on each host in the distributed environment. The Instrumentation Daemon is interrupted when new data is written to the FIFO. The Instrumentation Daemon reads the data from the FIFO and reformats the data into the standard internal Instrumentation message format and sends the data to each of the Instrumentation Collectors that are currently active. (For future implementations, an event request filtering mechanism will be implemented so that specific event messages will only be sent to those Instrumentation Collectors that have requested the message.)

Instrumentation Collectors

The Instrumentation Collectors receive instrumentation messages from the Instrumentation Daemons on each host in the distributed environment. Currently, the Instrumentation Collectors send every instrumentation message to all Instrumentation Brokers and Instrumentation Correlators that have connected to the Instrumentation Collector. (For future implementations, an event request filtering mechanism will be implemented so that specific event messages will only be sent to those Instrumentation Brokers and Instrumentation Correlators that have requested the message. For now, the Instrumentation Collector serves as a pass-through server for instrumentation messages. The Instrumentation Collector does supports architecture scalability in the sense that without the Instrumentation Collectors, each Instrumentation Broker and Instrumentation Correlators would need to maintain connections to the Instrumentation Daemons on every host.)

Instrumentation Correlators

The Instrumentation Correlators provide grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other Resource Management components. Each Correlator reads in a user-specified correlation grammar file that is interpreted at run-time by the Correlator's instrumentation correlation engine.

Instrumentation Brokers

The Instrumentation Brokers are task-specific applications built around a common code package. The Instrumentation Brokers receive instrumentation messages from the Instrumentation Collectors, filter all received instrumentation messages to find the messages of interest, and perform task-specific message data reformatting and manipulation for driving other components such as displays or other Resource Management components. The Instrumentation Broker approach allows for instrumentation data sources to be quickly integrated for test, display, and debugging purposes. (As the Instrumentation Correlator grammar and correlation engine mature in future releases, it is anticipated that the Instrumentation Broker approach will be used less frequently.)

Jewel Instrumentation Broker (QoS Monitor)

The Jewel Instrumentation Broker (hereafter referred to the QoS Monitor) is a legacy architecture component that served as a broker between the Jewel instrumentation package components and Resource Management components and displays. The QoS Monitor was responsible for polling the Jewel Collector components to retrieve application event messages. These messages were then reformatted and used to drive several displays and the QoS Managers. The Jewel instrumentation package has now been replaced in all

applications, however the message reformatting capabilities of the QoS Monitor have been maintained so that several displays and the existing QoS Manager interface do not have to be upgraded immediately. The QoS Monitor component has been modified so that it receives instrumentation data from both Jewel and the Instrumentation Collectors.

Middleware

The RMComms middleware package, which is described in the RMComms Middleware Design Report, provides the internal message passing interfaces between the Resource Management components connected via the network. The middleware provides for automatic location-transparent many-to-many client-server connections. Low overhead, reliable message passing capabilities are provided. Registration of message handler callback functions for specified requested message types are provided with the message handler functions being invoked when messages arrive. Registration of connection status callback functions which are invoked when either new connections are made or existing connections are broken is also provided. The middleware package also allows for multiple client and server objects to be instantiated in the same application, is thread-safe, and provides an easy-to-use object-oriented API through which all capabilities are accessed.

Additional details regarding the Instrumentation functional group FG2 are provided in CD-Appendix I.

FG42: Resource Manager

The Resource Manager 42 is the primary decision-making component of the Resource Management toolkit. It is responsible for: 1) responding to application and host failures by determining if and what recovery actions should be taken, 2) determining if and where to place new copies of scalable applications or which scalable applications should be

shutdown when the QoS Managers FG44A-FG44N indicate that scale-up or scale-down actions should be taken based on measured application performance, 3) determining where new applications should be placed when requested to do so by Program Control, and 4) determining which and how many applications should run based on application system (mission) priorities. In order to accomplish these tasks, the Resource Manager 42 maintains a global view of the state of the entire distributed environment including status information on all hosts, networks, and applications. In addition, the Resource Manager 42 also calculates software and hardware readiness metrics and reports these readiness values for display purposes. FIGS. 1A, 1B show the connectivity and high-level data flow between the Resource Manager 42 and the other Resource Management-related components.

The Resource Manager 42 receives status and failure information about hosts, networks, and applications from Program Control. This information includes periodic status updates as well as immediate updates when statuses change such as a new host being detected or an application failing. In the case of applications going down, information as to whether the applications were shutdown on purpose or whether they failed is also sent. Program Control also issues requests to the Resource Manager 42 when new applications need to be dynamically allocated and when Program Control determines that the Resource Manager 42 needs to assess and attempt to resolve inter-application dependencies (such as an application which needs to be running prior to starting up another application).

The Resource Manager 42 responds to faulted applications and hosts by determining whether the failed applications can and should be restarted and attempting to determine where (and if) there are hosts available that the application can run on. When a decision is made by the Resource Manager 42, a message is sent to Program Control specifying what application to start and where to put it. The same general mechanism is used when Program Control requests that the Resource Manager 42 determine where to start new applications and/or how to resolve inter-application dependencies; the Resource Manager 42 responds

with orders indicating what applications to start and where to start them. The Resource Manager 42 also sends application shutdown orders to Program Control requesting that certain application be stopped; this can occur when the QoS Managers FG44A-FG44N indicate that certain scalable applications have too many copies running or when application system priority changes (to lower priorities) occur resulting in scaling back the application system configuration.

The Resource Manager 42 receives host load and host fitness information on all known hosts from the Hardware Broker 40 (Host Load Analyzer). This information include overall host fitness scores, CPU-based fitness scores, network-based fitness scores, and memory and paging-based fitness scores along with the SPEC95 ratings of the hosts. This information is received approximately once a second and includes information on all known hosts in the distributed system. These scores are used by the Resource Manager 42 for determining the "best" hosts for placing new applications when: 1) responding to requests from the QoS Managers FG44A-FG44N to scale up additional copies of an application, 2) attempting to restart failed applications, 3) responding to requests to dynamically allocate certain applications, and 4) responding to application system (mission) priority changes which require scaling up additional applications.

The Resource Manager 42 receives requests from the QoS Managers FG44A-FG44N for scaling up, moving, or scaling down specific applications. The Resource Manager FG42 responds to these requests by determining whether the request should be acted upon and, if so, determines the specific action to take and issues orders to Program Control to start up or shutdown specific applications on specific hosts. The QoS Managers FG44A-FG44N are responsible for monitoring specific system performance metrics (e.g., quality of service, or QoS, requirements) via instrumentation and determining if performance can be improved by scaling up or moving certain applications. When this occurs, the QoS Managers FG44A-FG44N send a request to the Resource Manager FG42 indicating that a new copy of a

specific application should be started. If the QoS Managers FG44A-FG44N determine that the performance of a scalable application can be improved by moving an application, a scale up request is first sent to the Resource Manager FG42 and when the new application has been started, a scaledown request is then sent to the Resource Manager. Also, when the QoS Managers FG44A-FG44N determine that there are more copies of scalable application running then are needed, requests to shutdown specific applications are sent to the Resource Manager FG42. The division of responsibility is that the QoS Managers FG44A-FG44N determine what actions would potentially improve performance, but the Resource Manager FG42 has final authority to determine whether to implement the requested actions.

When the Resource Manager FG42 is first started, it reads in the System Specification Files (via System Specification Library, SSL, calls) which contain the list of hosts that are known to be in the distributed environment and information on all applications that can be run in the distributed environment. The System Specification Files also include application-level information including where specific applications can be run, which applications are scalable, which applications can be restarted, and any dependencies between applications.

The Resource Manager FG42 can also receive updated application survivability specifications from the QoS Specification Control component. This information overrides the application survivability information that was initially loaded in from the System Specification Files for specified applications. The information is used by the Resource Manager FG42 to determine whether the specific applications will be restarted if they fail at run-time.

The Resource Manager FG42 sends application system and hardware system readiness and system (mission) priority information to the Readiness Broker and to the Globus Broker. The Readiness Broker is responsible for driving a GUI/display which shows

the current readiness data and allows the system (mission) priorities to be changed and sent back to the Resource Manager FG42. The Globus Broker provides basically the same functionality except that only a high-level subset of the readiness data provided to the Readiness Broker is provided to the Globus Broker. The readiness information sent to the Readiness Broker consists of readiness values for each application, application subsystem, and application system defined in the System Specification Files. The readiness scores are currently based on the status (up/down) of the applications within a system or subsystem along with the percentage of potential copies of scalable applications that are currently running. Host and network readiness scores are also calculated and are the scores are determined based on the host load information and host fitness scores received from the Hardware Broker 40.

The Resource Manager FG42 also sends information about allocation and reallocation decisions to the Resource Management Decision Review Display. Information on the decision that was made, what event the decision was in response to, and how long it took to both make the decision and implement the decision are sent to the display. In addition, information about the top choices for where an application could have potentially been placed is also sent (if applicable); this information includes the host fitness scores for the selected host and the next best host choices which could have been selected.

See CD-Appendix M for additional details regarding Resource Manager FG42.

In the Background Section of the application, the reader may have interpreted the sentence "The present invention relates generally to resource management systems by which networked computers cooperate in performing at least one task too complex for a single computer to perform" to indicate that the Resource Management Architecture is limited to such applications. Thus, while the Resource Management Architecture generally supports tasks distributed across multiple hosts, it is not limited to only those tasks that must be

distributed due to the inability to run them on a single machine. Moreover, the Resource Management functional elements advantageously could be used to control a set of applications which all run on the same machine while still providing monitoring, fault tolerance, etc. (albeit that this is not the normal or even the intended configuration). Furthermore, the Resource Management Architecture, as discussed above, deals with resource managed applications, where the managed characteristic may be one of scalability, survivability, fault tolerance or priority.

FIG. 15 is a block diagram of a CPU-based system 400, corresponding to one or more of the hosts A-N. The system 400 includes a central processing unit (CPU) 402, e.g., a microprocessor, that communicates with the RAM 412 and an I/O device 408 over a bus 420. It must be noted that the bus 420 may be a series of buses and bridges commonly used in a processor-based system, but for convenience purposes only, the bus 420 has been illustrated as a single bus. A second I/O device 410 is provided in an exemplary case. The processor-based system 400 also includes a primary memory 412, an additional memory 414, which could be either a read-only memory (ROM) or another memory device, e.g., a hard drive or the like. The CPU- based system may include peripheral devices such as a floppy disk drive 404, a compact disk (CD) ROM drive 406, a display (not shown), a key board (not shown), and a mouse (also not shown), that communicate with the CPU 402 over the bus 420 as is well known in the art. It will be appreciated that the either one of the memories 412 or 414 advantageously can be employed to store computer readable instructions for converting the general purpose system 400 into one of the host A-N. It will also be appreciated that the nature of the distributed environment permits the necessary application and API's needed to implement the Resource Management Architecture to be stored anywhere on the network.

Table III

APPENDIX	TYPE	DESCRIPTION
A	Attached	Resource Management Architecture Function List
B	Attached	Standard Instrumentation Message Format
C	Attached	API Listing for RMComms
D	CD	Resource Manager Interface Messages
E	CD	Host Load Analyzer (Hardware Broker) Function
F	CD	Quality-of-service (QoS) Manager Function
G	CD	FG3: System Specification Language & System Specification Library (SSL) Functions
H	CD	Host And Network Monitoring Functional Group
I	CD	Instrumentation Functional Group
J	CD	Display Functional Group
K	CD	RMComms Network Communication Middleware Design
L	CD	System Readiness Display
M	CD	Resource Manager FG42
N	CD	Instrumentation Graph Tool
O	CD	Host Discovery Function
P	CD	Instrumentation Application Programming Interface (API)
Q	CD	Program Control Application Controller
R	CD	Program Control Display
S	CD	Program Control Functional Group
T	CD	QoS Manager
U	CD	Resource Allocation Decision-making Functional Group

Table III provides a listing of the Appendices included for all purposes in the application. It will be noted that the majority of the listed Appendices are provided on the

CD-ROM filed concurrently with the application. In addition, the CD-ROM also includes the source code listing for the Resource Management Architecture according to the present invention.

Although presently preferred embodiments of the present invention have been described in detail herein, it should be clearly understood that many variations and/or modifications of the basic inventive concepts herein taught, which may appear to those skilled in the pertinent art, will still fall within the spirit and scope of the present invention, as defined in the appended claims.

RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR

STATEMENT OF GOVERNMENT INTEREST

The invention described herein was made in the performance of official duties by employees of the Department of the Navy or by researchers under contract to an agency of the United States government and, thus, the invention disclosed herein may be manufactured, used, licensed by or for the Government for governmental purposes without the payment of any royalties thereon.

BACKGROUND OF THE INVENTION

The present invention relates generally to resource management systems by which networked computers cooperate in performing at least one task too complex for a single computer to perform. More specifically, the present invention relates to a resource management system which dynamically and remotely controls networked computers to thereby permit them to cooperate in performing tasks that are too complex for any single computer to perform. Advantageously, software programs for converting a general purpose computer network into a resource managed network are also disclosed.

The instant application claims priority from Provisional Patent Application Serial No. 60/207,891, which was filed on May 25, 2000. The Provisional Patent Application is incorporated herein in its entirety by reference.

Resource Management consists of a set of cooperating computer programs that provides an ability to dynamically allocate computing tasks to a collection of networked computing resources (computer processors interconnected on a network) based on the

following measures:

- an application developer/user description of application computer program performance requirements;
- measured performance of each application programs;
- measured workload (CPU processing load, memory accesses, disk accesses) of each computer in the network; and
- measured inter-computer message communication traffic on the network.

Many attempts to form distributed systems and environments have been made in the past. For example, several companies and organizations have networked multiple computers to form a massively parallel supercomputer of sorts. One the best known of these efforts is SETI@home, which is organized by SETI (Search for Extraterrestrial Intelligence), a scientific effort aiming to determine if there is intelligent life out in the universe.

Typically, the search means the search of billions of radio frequencies that flood the universe in the hopes of finding another civilization that might be transmitting a radio signal. Most of the SETI programs in existence today, including those at UC Berkeley, build large computers that analyze that data from the telescope in real time. None of these computers look very deeply at the data for weak signals nor do they look for a large class of signal types. The reason for this is because they are limited by the amount of computer power available for data analysis. To extract the weakest signals, a great amount of computer power is necessary. It would take a monstrous supercomputer to get the job done. Moreover, SETI programs could never afford to build or buy that computing power. Thus, rather than use a huge computer to do the job, the SETI team developed software to use thousands of small computers, all working simultaneously on different parts of the analysis, to run the search routine. This is accomplished with a screen saver that can retrieve a data block over the internet, analyze that data, and then report the results back to SETI.

Several commercial companies are developing and implementing similar capabilities. Moreover, several companies, most notably IBM, have developed networks where each networked desktop computer becomes a parallel processor in a distributed computer system when the desktop computer is otherwise idle.

It will be appreciated that these approaches to computing in a distributed environment do not provide a system that is both flexible and adaptive (or at least easily adapted) to changes in system configuration, performance bottlenecks, survivability requirements, scalability, etc.

What is needed is a Resource Management Architecture which permits flexible control, i.e., allowing autonomous start up and shut down of application copies on host machines to accommodate changes in data processing requirements. What is also needed is functionality included in the Resource Management Architecture which permits the Resource Management Architecture to determine the near-optimal alignment of host and application resources in the distributed environment. It would be desirable to have a user-friendly technique with which to specify quality of service (QoS) requirements for each host, each application, and the network in which the hosts are connected. What is also needed is instrumentation to ensure that the specified QoS goals are being met.

SUMMARY OF THE INVENTION

Based on the above and foregoing, it can be appreciated that there presently exists a need in the art for a Resource Management Architecture, which overcomes the above-described deficiencies. The present invention was motivated by a desire to overcome the drawbacks and shortcomings of the presently available technology, and thereby fulfill this need in the art.

According to one aspect, the present invention provides, in a distributed environment

MACHINE GENERATED RED-LINE VERSION OF SPECIFICATION

comprised of hosts instantiating copies of a scalable application, a resource management device generating signals which start up, shutdown or move a selected one of the copies responsive to first information regarding performance of all copies of the scalable application and second information regarding performance of the hosts.

BRIEF DESCRIPTION OF THE DRAWINGS

These and various other features and aspects of the present invention will be readily understood with reference to the following detailed description taken in conjunction with the accompanying drawings, in which like or similar numbers are used throughout, and in which:

~~Fig~~FIGS. 1~~is~~A, 1B collectively represent a high-level block diagram of hardware and software components implemented in the Resource Management System according to the present invention;

~~Fig~~FIGS. 2~~is~~A, 2B collectively represent a functional block diagram of the Resource Management Architecture according to the present invention;

FigIG. 3 is a functional block diagram illustrating functional elements included in the system specification library (SSL) implementation of the Resource Management System according to the present invention;

FigIG. 4 is a block diagram illustrating one technique for implementing the Resource (Application) Control functional group FG5 in ~~Fig~~FIGS. ~~2~~2A, 2B using discrete software components;

~~Fig~~FIGS. 5~~is~~A, 5B represent a screen capture of a program control display FG54 generated by the software components illustrated in FigIG. 4;

~~Fig~~FIGS. 6~~is~~A, 6B represent a screen capture of a host display generated by the Resource Management Architecture according to the present invention;

~~Fig~~FIGS. 7~~is~~A, 7B represent a screen capture of performance data regarding several of the hosts A - N included in ~~Fig~~FIGS. 66A, 6B;

~~Fig~~FIGS. 8~~is~~A, 8B represent a screen capture of a path display generated by the Resource Management Architecture according to the present invention;

Ex.1002 / Page 331 of 541
TESLA, INC.

FigFIGS. 9isA, 9B represent a screen capture of the Resource Management Decision Review Display, which provides a summary of allocation and reallocation actions taken by the Resource Manager;

FigsFIGS. 10A, 10B and 11areA. 11B represent screen captures illustrating alternative, user-configurable displays generated from received data via standardized message formats and open interfaces;

FigFIGS. 12isA. 12B represent a screen capture of an exemplary version of the Readiness Display FG66 according to the present invention;

Figs. 13A, 13B, and 13C are block diagrams which are useful in explaining various operational and functional aspects of the Resource Management Architecture according to the present invention; and

FrgIG. 14 is a high-level block diagram illustrating connectivity and data flow between the Hardware Broker and the other Resource Management and Resource Management-related functional elements in the Resource Management Architecture; and

FrgIG. 15 is a high-level block diagram of a CPU-based general computer which can act as a host in the Resource Management Architecture according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The Resource Management Architecture, which was and is being developed by the Naval Surface Warfare Center - Dahlgren Division (NSWC-DD), provides capabilities for monitoring hosts, networks, and applications within a distributed computing environment. Moreover, the Resource Management Architecture provides the capability of dynamically allocating, and reallocating, applications to hosts as needed in order to maintain user-specified system performance goals. Advantageously, the Resource Management architecture provides functionality for determining both how each component within the distributed environment is performing and what options are available for attempting to correct deficient performance, determining the proper actions that should be taken, and enacting the

determined course of action. In addition to these capabilities, the architecture also allows for operator control over creating and loading pre-defined static, dynamic, or combined static and dynamic system and/or host configurations. One particularly desirable feature of the Resource Management Architecture is that it provides capabilities for monitoring system performance along with the ability to dynamically allocate and reallocate system resources as required.

Before addressing the various features and aspects of the present invention, it would be useful to establish both terminology and the conventions that the instant application will follow throughout. In terms of terminology, a glossary section is presented below. In terms of conventions, this application includes information such as source code listing in an Appendix section. Since the source code itself is hundreds of pages, the aAppendix section is divided into attached pages, e.g., Attached Appendix A, and an optical disk section, e.g., CD-Appendix N. Thus, while the appendices are listed in order, the reader must look to the signaling language to determine whether any particular appendix is actually provided in printed form.

API	**API (application programming interface)** A set of subroutines or functions that a program, or *application*, can call to invoke sonme functionality contained in another software or hardware component. The Windows API consists of more than 1,000 functions that programs written in C, C++, Pascal, and other languages can call to create windows, open files, and perform other essential tasks. An application that wants to display an on-screen message can call Windows' *MessageBox* API function, for example.

BNF	Acronym for 'Backus Normal Form' (-often- incorrectly expanded as 'Backus -Naur Form'), a metasyntactic notation -used to specify the syntax of programming languages-, command sets, and the like. Widely used for language descriptions but seldom documented anywhere-, so that it must usually be learned by osmosis- from other hackers.
DAEMON	A background process on a host or Web server (-normally in a UNIX environment-), waiting to perform tasks. Well-known examples of daemons are sendmail and HTTP daemon.
FUNCTION	A capability available on a host due to the presence of software (e.g., a program), a software module (e.g., an API), etc.
GLOBUS	Wide area network (WAN) enterprise management and control capability developed under DARPA sponsorship by USC/ISI.
HOST	A device including a central processor controlled by an operating system.
ICMP	Internet Control Message Protocol - ICMP is an extension to the Internet Protocol. It allows for the generation of error messages, test packets and informational messages related to IP. It is defined in STD 5, RFC 792.
JEWEL	An open-source instrumentation package produced by the German National Research Center for Computer Science
NFS	Network File System - A protocol developed by Sun Microsystems, and defined in RFC 1094-, which allows a computer system to access files over a network as if they were on its local disks. This protocol has been incorporated in products by more than two hundred companies, and is now a *de facto* Internet standard.
QoS	Quality of Service

REMOS	Remos (REsource MOnitoring System)is a network bandwidth and topology monitoring system developed under DARPA sponsorship -by CMU. Remos allows network-aware applications to obtain relevant information about their execution environment. The major challenges in defining a uniform interface are network heterogeneity, diversity in traffic requirements, variability of the information, and resource sharing in the network. Remos provides an API that addresses these issue by striking a compromise between accuracy (the information provided is best-effort, but includes statistical information if available) and efficiency (providing a query-based interface, so applications incur overhead only when they acquire information). Remos supports two classes of queries. "Flow queries" provide a portable way to describe a communication step to the Remos implementation, which uses its platform-dependent knowledge to return to the user the capacity of the network to meet this request. "Topology queries" reverse the process, with the Remos implementation providing a portable description of the network's behavior to the application.
SNMP	Simple Network Management Protocol Internet standard protocol defined in STD 15, RFC 1157; developed to manage nodes, e.g., hubs and switches, on an IP network.

An exemplary system for implementing the Resource Management Architecture according to the present invention is illustrate in FrgFIGS. 1A, 1B, which includes a plurality of Host computers A, B, ..., N operatively connected to one another and Resource Management hardware RM via a Network 100. It will be appreciated that the hardware configuration illustrated in FIGS. 1a, 1B consititutes a so-called grid system. It will also be appreciated that the network 100 advantageously can be any known network, e.g., a local

area network (LAN) or a wide area network (WAN). It will also be appreciated that the hardware RM need not be a discrete piece of equipment; the hardware RM advantageously can be distributed across multiple platforms, e.g., the host computer(s), as discussed in detail below.~ In addressing the functional elements and applications in the distributed environment, it will be appreciated that hosts A-N each can instantiate applications 1-M. Thus, when all applications are being addressed, these applications will be denoted as A1-NM.

Still referring to ~~Fig~~FIGS. ~~1~~1A, 1B, each of the hosts A, B, etc., preferably is controlled by an operating system (OSA, OSB, etc.), which permits Host A, for example, to execute applications A1 - AN, as well as an instrumentation daemon IDA, a Program Control (PC) agent PCA, and a Host Monitor HMA. It should be noted that instrumentation daemon IDA, PC agent PCA, and Host Monitor HMA are integral to the Resource Management Architecture while the operating system OSA and applications A1 - AN are well known to one of ordinary skill in the art.

In ~~Fig~~FIGS. ~~1~~1A, 1B, the Resource Management Architecture RM advantageously includes an instrument collector 10 receiving data from all of the instrumentation daemons (IDA - IDN) and providing data to instrument correlator(s) 20, which, in turn, provide correlation data to corresponding quality of service (QoS) managers 30. Resource Management Architecture RM also receives data from host monitors HMA- HMN at history servers 40, which maintain status and performance histories on each of the hosts A - N and provide selected information to host load analyzer 50. Analyzer 50 advantageously determines the host and network loads for both hosts A-N and their connecting network 100 and provides that information to Resource Manager 60, which is the primary decision making component of the Resource Management Architecture. It will be appreciated that Resource Manager 60 also receives information from the QoS managers 30 and exchanges information with program controller 70. Program controller 70 sends startup and shutdown orders to the

Program Control Agents based on operator or Resource Manager-initiated orders. It will be appreciated that the operator-initiated orders are received via the one of the program control displays 80.

As will be discussed in greater detail below, the Resource Manager 60 is the primary decision-making component of the Resource Management Architecture. The Resource Manager 60 is responsible for determining:

- how to respond to host and application failures;
- where (i.e., which of hosts A -N) to place new applications;
- which applications to start up in response to the detection of a new host (host N+1);
- how to resolve application dependencies;
- what applications should be started, stopped, or moved in response to application system priority changes; and
- based on recommendations from the QoS Managers, when and where scalable application should be started or stopped.

Before leaving ~~Fig~~FIGS. ~~1~~1A, 1B, is should be noted that the functions, e.g., instantiated programs or software program modules, in the Resource Management Architecture advantageously can be distributed across multiple platforms, e.g., multiple hosts (which may or may not be the illustrated Hosts A -N) or a grid system.

The major functional groups of the Resource Management Architecture according to the present invention are illustrated in ~~Fig~~FIGS. ~~2~~2A, 2B. The functions illustrated as solid boxes are components of the Resource Management Architecture and are fully described below; the functions denoted by diagonal striping denote third-party software which has been integrated with the Resource Management Architecture but does not provide core functionality. Thus, the latter functions will be described only to the extent necessary to

provide integration details. Moreover, it will be appreciated that the functions and functionality of the Resource Management Architecture according to the present invention are interconnected to one another via middleware, which provides message passing interfaces between substantially all of the Resource Management functions. This middleware package, RMComms, is fully described below.

The major functional groups provided by the Resource Management architecture in an exemplary embodiment of the present invention are illustrated in ~~Fig~~FIGS. ~~2~~2A, 2B. A summary of the functions provided by the Resource Management Architecture is available in Attached Appendix A. These functions, taken together, provide an integrated capability for monitoring and control of a distributed computing environment. In addition, many of the functions (and functional groups) within the Resource Management Architecture can also be run in a non-integrated configuration, thus providing subsets of the integrated Resource Management capabilities.

These function(al) groups illustrated in ~~Fig~~FIGS. ~~2~~2A, 2B include:

FG1 - Host and Network Monitoring. This function group consists of software which monitors the host and network resources within the distributed environment. The function group collects extensive run-time information on host and network configuration, statuses, and performance. Run-time capabilities for discovering new hosts that have been started and for determining that existing hosts have gone down are also provided. Distribution of current and historical status and performance data to other components of the Resource Management Architecture is also provided. A more detailed discussion is provided below.

FG2 - Application-Level Instrumentation. The instrumentation function group provides general-purpose application event reporting and event correlation capabilities.

Capabilities are provided for collecting and correlating application-provided data such as application statuses, states, performance, and internally detected errors. Low-overhead (API) libraries are provided for applications to use in sending out key internal event and performance data. This application data is forwarded to other components of the instrumentation subsystem which collect data from applications on hosts throughout the distributed environment. The system also provides grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other Resource Management components.

FG3 - System Specifications. A specification language has been developed which allows the user to specify:

1) application software system structure, capabilities, dependencies, and requirements; and

2) hardware system (computer and network) structure, capabilities, and configuration.

Specification files, based on this specification language, are created by the user and provide the model of the software and hardware components of the distributed computing environment which is used by other Resource Management functions. The specification information is accessed by other Resource Management functions by linking in a specification parser library and making library calls to read in the files and convert them to an internal object model. Specific specification data items can then be retrieved via an object-oriented API. See the discussion below.

FG4 - Resource Allocation Decision-Making. This subsystem provides the reasoning and decision-making capabilities of the Resource Management architecture. The components of this subsystem use information from other subsystems in order to determine the health and state of the distributed environment and the options that are

Ex.1002 / Page 333 of 541
TESLA, INC.

available for attempting to recover from faults or unacceptable performance. The functions in this particular functional group make decisions regarding:

1) where new applications should be started;

2) whether and where failed applications should be restarted;

3) based on application inter-dependencies, whether and where additional applications should to be started prior to starting a particular application;

4) whether applications are meeting performance requirements and whether and where an application can be scaled up or moved to when it is necessary to improve performance;

5) whether scalable applications are performing well within performance requirements and can be scaled down and which copy should be brought down; and

6) based on operator changes to application system priorities, whether and where new applications need to be started or whether and which existing applications need to be shut down.

FG5 - **Application (Resource) Control.** This subsystem provides application control (i.e., Program Control) capabilities which permit starting, stopping, and configuring applications on each of the hosts in the distributed environment. The subsystem provides both interactive operator control of the distributed environment as well as automatic control via configuration orders received from the Resource Allocation Decision-Making Subsystem (i.e., the Resource Manager component). The interactive controls allow an operator to create, load, save, and edit pre-defined system configurations (e.g., lists of applications that are to be run, with or without specific host mappings), determine the status and configuration of currently running programs, and start and stop any or all applications. Both static (operator-entered) mappings of applications to hosts and dynamic mappings of applications to hosts (where the Resource Allocation Decision-Making Subsystem will be queried to

determine the proper mapping at run-time) can be defined. The subsystem also provides application fault detection capabilities which are triggered by the unexpected death of an application that was started by the subsystem. A basic host fault detection capability is also provided which is triggered based on failure to receive heartbeat messages from subsystem components running on a particular host.

FG6 - **Displays.** The display subsystem provides capabilities for visualizing the status, performance, and health of the hosts, networks, and applications in the distributed environment. Capabilities are also provided for visualizing the status, performance, and health of the Resource Management components themselves.

As mentioned above, the RMComms middleware package provides the internal message passing interfaces between substantially all of the Resource Management functions both within each functional group and between the various functional groups. The middleware provides for automatic location-transparent many-to-many client-server connections. Low-overhead, reliable message passing capabilities are provided. Registration of message handler callback functions for specified requested message types is provided with the message handler functions being invoked when messages arrive. Registration of connection status callback functions, which are invoked when either new connections are made or existing connections are broken, is also provided. The middleware package also allows for multiple client and server objects to be instantiated in the same application, is thread-safe, and provides an easy-to-use object-oriented API through which all capabilities are accessed.

A detailed overview of each functional group and each function instantiated within each of the function groups FG1 - FG6 of the exemplary embodiment of the Resource Management Architecture illustrated in FigFIGS. 2A, 2B, including the capabilities provided by the functional group or function, will now be described in greater detail. The

discussion below also includes an overview of the information flow between function blocks within the same functional group and between function blocks in separate functional groups.

FG1 - Host and Network Monitoring Functional Group

Functional group FG1 provides extensive monitoring capabilities at the host and network levels. The information monitored includes statuses, configuration information, performance metrics, and detected fault conditions. By monitoring the individual hosts and network components within the distributed environment, the functional group FG1 determines:

- Accurate State and Performance Information, primarily by gathering the level of information necessary for accurately determining the state and health of each machine and network component.
- Distribution of Current Data to Resource Management Components by providing current performance and status information, either periodically or on request.
- Distribution of Historical Data to Resource Management Components, thus providing historical performance and status information on request.

It will be appreciated that the functional group FG1 makes these determinations by (or while) providing:

- Common Monitored Data Set and Formats, which permits functional group FG1 to gather the same set of statuses and statistics in the same formats for each host regardless of machine architecture or operating system.
- Minimally-Intrusive Data Collection Mechanisms, which permits functional group FG1 to gather the information in as non-intrusive a manner as possible (in terms of CPU utilization, network bandwidth utilization, etc...).
- Near Real-Time Data Collection Mechanisms, which permits functional

group FG1 to gather the information in as timely a manner as possible.

The Host and Network functional group FG1 includes the four functions set forth below:

1) **Host Monitors** FG10A - FG10N, which reside on each respective machine in the distributed environment and collect extensive operating system-level data for each host A - N.

2) **History Servers** FG12A - FG12N, which collect data from the Host Monitors FG10A - FG10N, respectively, maintain status and performance histories on each host A - N in the distributed environment, i.e., in the Resource Management Architecture, and provide this information to displays and other functions with the Resource Management Architecture.

3) **Host Discovery Function** FG14, which uses Simple Network Management Protocol (SNMP) calls and ping Internet Control Message Protocol (ICMP) calls to determine when new hosts, Ee.g., host N+1, come on-line and if an existing host, e.g., host K, goes down.

4) **Remos Network Data Broker** Function FG16, which collects information on network link bandwidths from the SNMP-based Remos tool (developed by Carnegie Mellon University) and passes this information to the Host Load Analyzer function of the Resource Allocation Decision-Making functional group FG4, both of which are discussed in greater detail below.

Host monitors FG10A- FG10N, which monitor the status and performance of hosts A -N, respectively, are instantiated on each host machine within the distributed environment. Host Monitors FG10A - FG10N employ operating system-level mechanisms to retrieve status, configuration, and performance information on each host A - N. The information retrieved includes:

1) operating system version and machine configuration;

2) CPU configuration, status, and utilization;

3) memory configuration and usage;
4) network configuration, status, and utilization;
5) filesystem configuration, status, and utilization; and
6) process statuses including CPU, memory, network, and filesystem utilization for each process.

While Host Monitors FG10A - FG10N are primarily responsible for monitoring the status of a particular host, they also provide information on network load as seen by that particular host. In the same manner, the Host Monitors FG10A - FG10N also provide information and statistics concerning any remotely mounted filesystems, e.g., Network File System (NFS).

The information that the Host Monitors FG10A - FG10N collect advantageously can be formatted into operating system-independent message formats. These message formats provide a pseudo-standardized set of state, status, and performance information which is useful to other components of the Resource Management Architecture, i.e., other components do not have to be aware of or deal with the minor differences between data formats and semantics. It will be appreciated that since not all the state and performance data is available on every platform, in order to indicate which information is available, a group of flags are set in the host configuration message indicating whether specific data items are valid on a particular platform.

History Servers FG12A - FG12N are responsible for collecting information from the Host Monitors FG10A - FG10N and maintaining histories on the statuses, statistics, and performance of each host A - N in the distributed environment. This information advantageously can be requested by other functions instantiated in the Resource Management Architecture. Preferably, the primary consumers of the status information obtained by the History Servers FG12A - FG12N are the Host Load Analyzer (Hardware Broker) component of the Resource Allocation Decision-Making functional group FG4, the Host Display FG62A - FG62N and the Path Display FG64 of the Displays functional group FG6. The Host Load

Analyzer FG40 receives information on host configuration and loads (primarily CPU, memory, and network data) from History Servers FG12A - FG12N and employs this information to assign host fitness scores. Each Host Display, e.g., FG62A, receives and displays current status information on one of the hosts A - N, including process status information, and network connectivity information. Each Host Display can also request that a respective one of the History Servers FG12A - FG12N provide CPU load information, network load information, paging activity data, and memory utilization information, which is used to drive line graph charts for specific selected hosts.

It will be appreciated that History Servers FG12A - FG12N are designed so that multiple copies can be run simultaneously. Each of the History Servers FG12A - FG12N advantageously can be configured to either monitor all Host Monitors or to monitor only a selected set of Host Monitors. It should be mentioned at this point that the History Servers FG12A - FG12N determine the list of hosts in the distributed environment that could potentially be monitored from the System Specification Library. In this manner, the History Servers advantageously can be used to provide survivability (by having multiple History Servers connected to each Host Monitor) and/or to perform load-sharing (with the History Servers FG12A - FG12N each monitoring only a subset of the Host Monitors). It will also be appreciated that the History Servers FG12A - FG12N can be configured to periodically record history data to disk. These disk files can then be used for off-line analysis of the Resource Management Architecture.

The Host Discovery function FG14 employs Perl scripts in making SNMP and ICMP ping calls. These calls are used to periodically scan each subnet and host address in the distributed environment in an attempt to determine whether there have been any host status changes. In an exemplary case, the list of hosts and subnets that are to be monitored is read in from a file; alternatively, this information can reside in and be read from the System Specification Library, which is discussed in greater detail below.

It should be mentioned that when a new host is first detected, the new host's operating system configuration is queried by the Host Discovery function FG14 via SNMP calls. Information on the newly discovered host and its operating system configuration is then sent to the Program Control function FG50 in application control functional group FG5. Likewise, when a host fails to respond to multiple SNMP and ping queries, a message indicating that the host appears to have gone down is sent to the Program Control function FG50.

The final component of the Host and Network Monitoring functional group FG1 is the Remos Network Data Broker FG16, which receives information on network link bandwidth and network link bandwidth utilization from the SNMP-based Remos network monitoring tool mentioned above. The network information is accessed via the Remos application programming interface (API) library and is then sent on to the Host Load Analyzer (Hardware Broker) function FG40 of the Resource Allocation Decision-Making functional group FG4. The network information received from Remos consists of the maximum potential bandwidth and the current bandwidth utilization on specific host network links. As mentioned above, Remos network monitoring tool FG 16 is not a core component of the Resource Management Architecture; that being the case, no further details on either Remos or the Remos Network Data Broker are provided in the instant application.

FG2 - Application-Level Instrumentation Functional Group

The Instrumentation functional group FG2 advantageously provides general-purpose application event reporting and event correlation capabilities. The Instrumentation functional group permits instrumented application data to be easily accessible to other components of the Resource Management Architecture. The functional group provides capabilities for collecting and correlating application-provided data such as application statuses, states, performance, and internally detected errors. Low-overhead API's are provided that the

applications can use for sending internal event and performance data to the instrumentation components. The instrumentation functional group FG2 can collect data from applications on hosts A - N throughout the distributed environment. The functional group also provides grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other functional groups of the Resource Management Architecture.

The Instrumentation functional group provides:

- open API's and non-proprietary architecture
- near real-time monitoring support
- cross-language support: C, C++, Ada
- cross-platform support: Solaris, IRIX, Linux, etc...
- simple easy-to-use API's
- low-intrusive instrumentation interface
- instrumentation interface that does not significantly change the run-time behavior of the applications
- support for passing wide range of data types
- support for data marshalling / unmarshalling (system independent data formats)
- support for adding to or changing the information being instrumented without having to recompile portions of the architecture unaffected by the changes (preferably, no recompilation should be necessary expect for recompilation of the app being instrumented and any evaluation logic or displays that have been affected by the changes)
- scalable architecture (100+ hosts / 20+ apps per host / 5+ threads per app)
- ability for the architecture to perform auto-configuration as required
- ability to run multiple tests, multiple displays and multiple data logging components simultaneously

- ability to abstract away the underlying connectivity/communications between infrastructure components.
- ability for instrumentation infrastructure to be brought up and down while the application is running
- ability to easily build and configure new displays and data logging components (interactive configuration is preferable)
- ability to easily build and configure new performance and data correlation components (interactive configuration is preferable)
- backwards compatibility with existing Jewel Instrumentation displays (protect investments in existing display capabilities)
- backwards compatibility with existing Jewel Instrumentation function calls (provide ease of transition / backfit)

As illustrated in FigFIGS. 2A, 2B, the Instrumentation functional group FG2 includes the components enumerated below. In addition, Instrumentation APIs and Jewel Instrumentation will be addressed along with the Instrumentation functional group, i.e., the Instrumentation functional group includes:

1) **Instrumentation API Libraries** FG20 are linked with the applications and provide the function call interfaces by which these applications send instrumentation data.
2) **Instrumentation Daemons** FG22A - FG22N reside on each host in the distributed environment and are responsible for reading instrumentation data sent out by the applications, reformatting the data into instrumentation event messages and sending the messages to the Instrumentation Collectors.
3) **Instrumentation Collectors** FG24A -FG24N connect to the Instrumentation Daemons FG22A - FG22N on each host and receive instrumentation messages from host A - N. The Collectors forward received messages to the Instrumentation Correlators FG26A - FG26N and Instrumentation Brokers

FG28A - FG28N.

4) **Instrumentation Correlators** FG26A - FG26N receive instrumentation messages from the Instrumentation Collectors FG24A - FG24N and provide grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other functions of the Resource Management Architecture.
5) **Instrumentation Brokers** FG28A - FG28N receive instrumentation messages from the Instrumentation Collectors and perform task-specific reformatting and data manipulation for driving displays or other Resource Management components.
6) **Jewel Instrumentation Broker (QoS Monitor)** FG29 (a legacy component) receives instrumentation data from either the open source Jewel instrumentation package or from the Instrumentation Collectors. The QoS Monitor FG29 performs task-specific message reformatting and data manipulation for driving displays and the QoS Managers FG44A - FG44N.

The applications, e.g., A1-AN, link in the Instrumentation API Library FG20 and make API calls to construct and send out instrumentation event messages. Three separate APIs are provided for use by the applications:

1) a printf()-style API which allows the code to format, build, and send instrumentation data with a single function call;
2) a buffer-construction-style API where the multiple function calls are made to construct the instrumentation buffer iteratively, one data element per call; and
3) a Jewel function call API based on the existing API provided by the Jewel instrumentation package (an open-source package produced by the German National Research Center for Computer Science).

The first two APIs are the preferred programming interfaces and take advantage of several key instrumentation features while the Jewel API is provided solely for backwards

compatibility with existing instrumented application code and is implemented as a set of wrappers around the printf()-style API. All three APIs are supported for C and C++. ADA bindings have also been produced for the buffer-construction-style API and the Jewel function call API.

Preferably, the instrumented data is sent from the application to one of the Instrumentation Daemons FG22A -FG22N on a respective one of the hosts A - N where the application is running. The currently preferred mechanism for data transfer is via UNIX FIFO (first in - first out) IPC (inter-process communication) mechanisms. It will be appreciated that the FIFO mechanism was chosen based on reliability, low overhead, and ease of implementation. Alternative data passing mechanisms including shared message queues are considered to be within the scope of the present invention.

As mentioned above, an Instrumentation Daemon resides on each host in the distributed environment. The Instrumentation Daemon is interrupted whenever new data is written to the FIFO. The Instrumentation Daemon reads the data from the FIFO, reformats the data into the standard internal Instrumentation message format (discussed below), and sends the data to each of the respective Instrumentation Collectors FG24A -FG24N that are currently active. Alternatively, an event request filtering mechanism can be implemented so that specific event messages will only be sent to those ones of the Instrumentation Collectors FG24A -FG24N that have requested the message.

The standard instrumentation message format includes a header, a format string describing the application-provided data contained in the message, and the actual data values. The message components are illustrated in Attached Appendix B.

The Instrumentation Collectors FG24A- FG24N receive instrumentation messages from the Instrumentation Daemons FG22A - FG22N on each host A - N, respectively, in the

distributed environment. Currently, the Instrumentation Collectors FG24A- FG24N send every instrumentation message to all Instrumentation Brokers FG29A- FG29N and Instrumentation Correlators (Brokers) FG26A- FG26N that have connected to the Instrumentation Collectors FG24A- FG24N. The Instrumentation Collectors FG24A- FG24N serve as a pass-through server for instrumentation messages. The Instrumentation Collectors do support architecture scalability in the sense that without the Instrumentation Collectors, the Instrumentation Broker FG29 and Instrumentation Correlators FG26A- FG26N would need to maintain connections to the Instrumentation Daemons FG22A- FG22N on every host. As discussed above, an event request filtering mechanism advantageously can be implemented so that specific event messages will only be sent to those Instrumentation Brokers / Instrumentation Correlators that have requested the message.

Preferably, the Instrumentation Correlators FG26A- FG26N provide grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other components of the Resource Management Architecture. Each Correlator reads in a user-specified correlation grammar file which is interpreted at run-time by the Correlator's instrumentation correlation engine.

The Instrumentation Brokers FG28A- FG28N are task-specific applications built around a common code package. The Instrumentation Brokers FG28A- FG28N receive instrumentation messages from the Instrumentation Collectors FG24A- FG24N, filter all received instrumentation messages to find the messages of interest, and perform task-specific message data reformatting and manipulation for driving other components such as displays or other components of the Resource Management Architecture. This Instrumentation Broker approach permits instrumentation data sources to be quickly integrated for test, display, and debugging purposes.

It should be mentioned at this point that the Jewel Instrumentation Broker FG29

(hereafter referred to the QoS Monitor) is a legacy architecture component that served as a broker between the Jewel instrumentation package components and Resource Management components and displays. The QoS Monitor FG29 was responsible for polling the Jewel Collector components to retrieve application event messages. These messages were then reformatted and used to drive several displays and the QoS Managers FG44A - FG44N. The Jewel instrumentation package has now been replaced in all applications, however the message reformatting capabilities of the QoS Monitor have been maintained so that several displays and the existing QoS Manager interface do not have to be upgraded immediately. The QoS Monitor component has been modified so that it receives instrumentation data from both Jewel and the Instrumentation Collectors.

FG3 - SYSTEM SPECIFICATIONS FUNCTIONAL GROUP

Still referring to FigFIGS. 22A, 2B, it should be noted that a System Specification Language has been developed which allows the user to specify both (1) software system structure, capabilities, dependencies, and requirements, and (2) hardware system (computer and network) structure, capabilities, and configuration. System Specification Files, generally denoted FG32, which are based on this specification language, are created by the user and provide a model of the software and hardware components of the distributed computing environment which is used by the Resource Management Architecture. The language grammar advantageously can capture the following information related to the distributed environment and the applications that can run within the distributed environment:

- **Hardware and Operating Systems**
  - Hardware Configuration
  - Network Configuration
  - Operating Systems and Version
- **Software**
  - Systems, Subsystems, Paths, Applications, Processes

- Resource Requirements
- QoS Requirements (Events)
- Survivability Requirements
- Data Flow Path Information: Structure and QoS Requirements

It will be appreciated that the System Specification Language allows for grouping hardware and software components into systems and subsystems in order to create a hierarchy of components. Each application system and subsystem can be assigned a priority which is used at run-time to determine the relative importance of applications running in the distributed environment.

At the application level, the hardware, operating system, and other host requirements for each application can be specified along with information describing how to start up, configure, and shutdown the application. This information can include:

a) environment variables that need to be set;
b) the working directory for running the application;
c) the path(s) and file name of the application;
d) command-line arguments that should be set, including arguments that need to be resolved at run-time (e.g., the hostname where another application is running, the current date, the current userid, a unique run-time identifier number, etc...);
e) whether the application needs to run in an xterm;
f) whether a script file or signal should be run to shutdown the application; and
g) which script or signal should be used.

In addition, startup and shutdown dependencies between applications can be specified. Moreover, application states can be defined based on received instrumentation data values, the length of time an application has been running, and/or the set of processes that are currently running. Furthermore, for each application A1 - NM, the survivability and

scalability capabilities of the application can be specified. This latter information includes whether an application can be restarted if it fails, whether multiple copies of an application can be run, what type of scalability the application supports (e.g., Primary-Shadow, Load-Sharing, etc...), and the minimum and maximum number of copies that can be run. Moreover, an estimate of the amount of CPU, memory, and network resources that the application will use at run-time, advantageously can be specified.

At the host level, the operating system and version, the hardware architecture, the host's network interface name, and the SPEC organization's SPECfp95 and SPECint95 ratings for the host can be specified. At the network level, router and switch configurations and bandwidths can also be specified.

Moreover, application data flow paths can be defined including a graph of the data flow between applications along with performance requirements tied to one of more of the applications within the path. It should be mentioned that these defined requirements are named and are tied at run-time to Instrumentation Event data provided by the Instrumentation Correlators FG26A- FG26N. Monitoring of the performance requirements is the responsibility of the QoS Manager components FG44A - FG44N, as discussed in greater detail below.

As noted above, the System Specification Language provides a hierarchical structure for defining software and hardware systems. The current structure is shown below:

- **Software Specifications**
  - Application
    - Security
    - Configuration
      - Hardware Requirements
      - Startup Info

      - Dynamic Arguments
      - Shutdown Info
      - States
    - Dependencies
    - Initial Load Estimate
    - QoS Info
      - Survivability
      - Scalability
- **Hardware Specifications**
  - Host Info
  - Network Info
    - LANs
    - Network Devices (Interconnects)
- **Path Specifications**
  - Data Flow Graph
  - Data Flow Info
  - QoS Requirements

The specification information is accessed by linking in a specification parser library FG34 and making library calls to read in the files and convert them to an internal object model, and by making object access method calls to retrieve specific data items. The specification library is written in C++ and has been ported to all of the development platforms in the testbed. The library is currently being used by most of the Resource Management components, including Program Control FG50, the Resource Manager FG42, the QoS Managers FG44A -FG44N, the Hardware Broker FG40, and the History Servers FG12A - FG12N.

It should be mentioned that the software used to construct the API library consists of

(1) a parser file that defines the grammar (in BNF format), (-2) a lexical analyzer file that defines the tokens of the language, and (3) a set of C++ System Specification classes for storing the specification file information. The lexical analyzer file is compiled with the GNU flex (lex) utility and the parser file is compiled using the GNU bison (yacc) utility. The flex and bison utilities create C source files which are then compiled along with the C++ System Specification object storage classes to create the System Specification Library (SSL) FG34. This library is then linked with the Resource Management applications. An overview of this structure is provided in FigIG. 3; a more detailed discussion of the various functions are provided below.

FG4 - RESOURCE ALLOCATION DECISION-MAKING FUNCTIONAL GROUP

As illustrated in ~~Fig~~FIGS. ~~2~~2A, 2B, the Resource Allocation Decision-Making functional group provides the reasoning and decision-making capabilities of the Resource Management architecture. The functions associated with this functional group employ information (listed below) to (1) determine the state and health of the distributed environment (hosts, networks, and applications), and (2) determine what allocation and reallocation actions need to be taken. The information provided to functional group FG4 includes:

- System Specifications:
  - Host configuration and capabilities
  - Application capabilities
    - Survivability
    - Scalability
    - Potential hosts to run on
  - Application startup and shutdown dependencies
  - Application and path performance requirements
- Program Control:

- Application statuses
- Detected application faults
- Detected host failures
- Detection of new host
- Operator initiated requests
  - Resolution of application startup or shutdown dependencies
  - Selection of application-to-host mappings
- History Servers:
  - Host statuses, configuration, and loads
  - Network link statuses and loads
- Remos Network Data Broker:
  - Network link statuses and loads
- Instrumentation Subsystem:
  - Application performance information
- Readiness Display:
  - Run-time changes to application system priorities

The subsystem components make decisions based on the following triggers and data sources:

- Based on requests from Program Control, determine where new applications should be started
- Based on indication of application failure from Program Control, determine whether and where the failed applications should be restarted
- Based on indication of host failure from Program Control (or indirectly from Host Discovery), determine whether and where the failed applications should be restarted
- Based on application inter-dependencies defined in the System Specification Files, determine whether and where additional applications should to be

started (or shut down) prior to starting (or shutting down) a particular application
- Based on startup and shutdown dependency resolution requests from Program Control, determine whether and where additional applications should to be started (or shut down) prior to starting (or shutting down) a particular application
- Based on application instrumentation data and performance requirements defined in the System Specification Files, determine whether applications are meeting performance requirements and whether an application can be scaled up or moved to attempt to improve performance
- Based on application instrumentation data and performance requirements defined in the System Specification Files, determine whether applications are performing well within performance requirements and can be scaled down
- Based on operator changes to application system priorities, determine whether and where new applications need to be started and/or determine whether and which existing applications need to be shutdown
- Based on indication that a new host is on-line (from Host Discovery via Program Control), issue startup orders to bring up a Program Control Agent, Host Monitor, and Instrumentation Daemon on the new host which will bring the host under Resource Management control

The Resource Allocation Decision-Making functional group implements one of the three discrete functions listed below:

1) **Resource Manager** FG 42 is the primary decision-making component of the Resource Management Architecture. Resource Manager FG42 is responsible for determining (1) how to respond to host and application failures, (2) where to place new applications, (3) which applications to start up in response to the detection of a new host, (4) how to resolve application dependencies, (5)

what applications should be started, stopped, or moved in response to application system priority changes, and (6) based on recommendations from the QoS Managers FG44A - FG44N, when and where scalable application should be started or stopped.

2) **Host Load Analyzer** FG40 is responsible for assigning a set of fitness scores to each host based on host capabilities and loads.

3) **QoS Managers** FG44A -FG44N are responsible for monitoring application and path requirements as defined in the System Specification Files FG32 and recommending that applications be either scaled up, scaled down, or moved in order to maintain acceptable performance.

As mentioned above, the Resource Manager FG42 is the primary decision-making component of the Resource Management Architecture. It is responsible for:

(1) responding to application and host failures by determining if and what recovery actions should be taken;

(2) determining if and where to place new copies of scalable applications or which scalable applications should be shutdown when the QoS Managers indicate that scale-up or scale-down actions should be taken based on measured application performance;

(3) determining where new applications should be placed when requested to do so by Program Control; and

(4) determining which and how many applications should run based on application system (mission) priorities.

In order to accomplish these tasks, the Resource Manager FG42 maintains a global view of the state of the entire distributed environment including status information on all hosts A- N, networks 100, and applications A1 -NM. In addition, the Resource Manager FG42 also calculates software and hardware readiness metrics and reports these readiness

values, for display purposes, to the display functional group FG6.

It will be appreciated from ~~Fig~~FIGS. ~~2~~2A, 2B that the Resource Manager FG42 receives status and failure information about hosts, networks, and applications from Program Control function FG50. This information includes both periodic status updates and immediate updates when statuses change such as a new host being detected or an application failing. In the case of application shutdown, information as to whether the application was shutdown intentionally or whether the application failed is also provided. Program Control function FG50 also issues requests to the Resource Manager FG42 when new applications need to be dynamically allocated and when the Program Control function FG50 determines that the Resource Manager FG42 needs to assess and attempt to resolve inter-application dependencies (such as one application which needs to be running prior to starting up another application).

The Resource Manager FG42 responds to faulted applications and hosts by determining whether the failed applications can and should be restarted and attempting to determine where (and if) there are hosts available that the application can run on. When a decision is made by the Resource Manager FG42, a message is sent to Program Control FG50 specifying what application to start and where to put it, i.e., which of hosts A - N to start the application on. The same general mechanism is used when Program Control FG50 requests that the Resource Manager FG42 determine where to start new applications and/or how to resolve inter-application dependencies; the Resource ~~Manager~~FG42Manager FG42 responds with orders indicating what applications to start and where to start them. The Resource Manager FG42 advantageously can send application shutdown instructions to Program Control FG50 requesting that a certain application be stopped; this can occur when the QoS Managers FG44A-FG44N indicate that certain scalable applications have too many copies running or when application system priority changes (when an application changes from a high priority to a lower priority) occur resulting in scaling back the application system

configuration.

The Resource Manager FG42 also receives host load and host fitness information on all known hosts from the Hardware Broker (Host Load Analyzer) FG40. This information includes (1) overall host fitness scores, (2) CPU-based fitness scores, (3) network-based fitness scores, and (4) memory and paging-based fitness scores, along with (5) the SPEC95™ rating of the hosts. These scores are used by the Resource Manager FG42 for determining the "best" hosts for placing new applications when:

(1) responding to requests from the QoS Managers to scale up additional copies of an application;
(2) attempting to restart failed applications;
(3) responding to requests to dynamically allocate certain applications; and
(4) responding to application system (mission) priority changes which require scaling up additional applications.

The Resource Manager ~~FG42 also~~FG42 also receives requests from the QoS Managers FG44A-FG44N for scaling up, moving, or scaling down specific applications. The Resource Manager FG42 responds to these requests by determining whether the request should be acted upon and, if so, determines the specific action to take. The Resource Manager FG42 then issues orders to Program Control FG50 to start up or shutdown specific applications on specific hosts.

It should be noted that when the Resource Manager FG42 is first started, it reads in the System Specification Files FG32 (via calls to System Specification Library FG34) which contains the list of hosts that are known to be associated with the distributed environment and information on all applications that can be run in the distributed environment. The application-level information includes where, i.e., on which host, specific applications can be run, which applications are scalable, which applications can be restarted, and any dependencies between applications.

The Resource Manager FG42 currently responds to application system priority changes received from the Readiness Broker (translation software in or associated with the Readiness Display FG66) in the following manner:

(1) If the priority is changed to None, all applications associated with the specified system are shutdown.
(2) If the priority is changed to Low, all scalable applications within the specified system are scaled back to no more than 50% of potential maximum scalability and are not allowed to be scaled up past the 50% limit irregardless of performance.
(3) If the priority is changed to Medium, normal scaleup and scaledown functionality is allowed.
(4) If the priority is changed to High, all scalable applications are scaled up to at least 50% of potential maximum scalability and are not allowed to be scaled down to less than 50% irregardless of performance.
(5) If the priority is changed to Urgent, all scalable applications are scaled up to 100% (for maximum survivability) and are not allowed to be scaled down.

[Moreover, if the previous priority was None, and the new changed priority is higher than None, all required applications within the specified system are started up subject to the limitations outlined for each of the priority levels listed above.]

The Resource Manager FG42 also sends information about allocation and reallocation decisions to the Resource Management Decision Review Displays FG68A-FG68N, as discussed in greater detail below. Information on the decision that was made, what event the decision was in response to, and how long it took to both make the decision and implement the decision advantageously are also sent to the display functional group FG6. In addition, information about the alternative choices for where an application could have potentially been placed is also provided (if applicable); in an exemplary case, this information includes the host fitness scores for the selected host and the next best host

choices which could have been selected.

As described above, the Resource Manager FG42 communicates with Program Control FG50, the Hardware Broker FG40, the QoS Managers FG44A -FG44N, QoS Specification Control (not shown), the Readiness Broker of display FG66, the Globus Broker (e.g., message translation software (not shown)), and the RM Decision Review Displays FG68A- FG68N using the RMComms middleware, which will be discussed in greater detail below.

The Hardware Broker (Host Load Analyzer) FG40 is the host load analysis component of the Resource Management Architecture, which is primarily responsible for determining the host and network loads on each host A - N within the distributed computing environment. The Hardware Broker FG40 assigns a set of fitness scores for each host and periodically provides the list of fitness scores to the Resource Manager FG42.

The Hardware Broker FG40 advantageously receives operating system-level statuses and statistics for each host A- N from the History Server(s) FG12A-FG12N, respectively. This information can be employed for calculating CPU, network, memory, paging activity, and overall fitness scores for each of the hosts A-N. Preferably, the Hardware ~~Broker FG40~~Broker FG40 periodically, e.g, once per second, provides the complete list of host fitness scores to the Resource Manager FG42.

It should be noted that when the Hardware Broker FG40 is first started, it reads in the System Specification Files FG32 (via calls to the System Specification Library (SSL) FG34), which files contain the list of hosts that are known to be in the distributed environment. The Hardware Broker FG40 also receives, e.g., reads in a file containing, information about the bandwidth and maximum packet sizes on all known network subnets in the distributed environment. It will be appreciated that this data advantageously can be used for converting

host network load information based on packet counts to load information based on bytes per second and percentage of available bandwidth.

Periodically, e.g., approximately every three seconds, the Hardware Broker FG40 transmits a list of overall and network host fitness scores to the Hardware Broker Instrumentation Display which was constructed using the Graph Tool Instrumentation Display FG69A-FG69N. Moreover, the Hardware Broker FG40 advantageously can receive host-based network load data from the Remos Network Data Broker function FG16, which receives network data via the Remos Network Monitoring software 2. It should be noted that if Remos network data is available for any of the hosts A -N that are being monitored, the Remos reported network data advantageously can be used for calculating the network fitness score for that host, rather than using the host network data received from the History Server(s) FG12A-FG12N.

The QoS Managers FG44A - FG44N of functional group FG4 are responsible for monitoring application-level performance requirements. These requirements are defined in the System Specification Files FG32 and are monitored primarily via instrumentation data obtained directly from the application code. The QoS Managers FG44A - FG44N advantageously can determine if applications or application paths are meeting their assigned requirements. If an application is not meeting its performance requirements and the application is scalable (in the sense that multiple copies can be run and the copies will perform load-sharing across the copies), the QoS Managers FG44A - FG44N will either request that the Resource Manager FG42 scale up a new copy of the application or move the application to a new host (as an attempt to achieve better performance). Moreover, if there are multiple copies of a scalable application running, and all copies are performing well below the specified requirement threshold, the QoS Managers FG44A - FG44N will request that the Resource Manager FG42 shutdown a specific copy. It should be noted that the division of responsibility between the QoS Managers FG44A - FG44N and the Resource

Manager FG42 is that the QoS Managers determine what actions would potentially improve performance, while the Resource Manager has final authority to determine whether to implement the requested action(s).

Each of the QoS Managers FG44A - FG44N can be scaled for both redundancy and for load-sharing. In an exemplary case, each copy of the QoS Manager monitors all of the requirements associated with a single application path defined in the System Specification Files FG32. It will be appreciated that the specific path to be monitored can be specified via command-line parameters. By default, without specifying a path via the command-line, the QoS Managers FG44A - FG44N will monitor all requirements for all paths defined in the System Specification Files FG32.

It should be mentioned that, in one exemplary embodiment, the QoS Managers FG44A - FG44N each employ a sliding window algorithm to determine when to declare that applications should be scaled up or scaled down. The inputs to the algorithm define both high and low sampling window sizes, the maximum number of allowed violations within the sampling window, and violation thresholds as a percentage of the actual specified requirement value. It should also be mentioned that the sliding window algorithm was selected in order to damp out unexpected "noise" or "spikes" in the measured performance data. Moreover, the threshold value as a percentage of the actual requirement value was selected in order to scale up, or scale down, prior to violating the specified hard requirement. The QoS Managers FG44A - FG44N provide application scale up and scale down requests to the Resource Manager FG42 when the measured performance data for an associated application violates either the high (scale up) or low (scale down) sliding window criteria for a specific requirement. A scale up request indicates which application on which host has violated the performance criteria, and a scale down request indicates which application on which host is recommended to be shutdown. It will be appreciated that the success of this algorithm is highly dependent on the rate of change and noisiness of the measured data.

Any of the QoS Managers FG44A - FG44N can also request that the Resource Manager FG42 move an application. This will occur in the case where one copy of an application is performing much worse than all other running copies of the same application. In an exemplary case, the move request is implemented as a scale up request followed by a scale down request (of the badly performing copy). In that case, the scale down request does not get sent to the Resource Manager FG42 until the scale up action has been implemented. The QoS Managers FG44A - FG44N preferably employ application "settling times," defined in the System Specification Files FG32, to ensure that once a requested action has been sent to the Resource Manager FG42 that no additional actions are requested for that application until after the settling time has elapsed. It will be appreciated that this provides time for initialization and configuration among the application copies to occur. Alternatively, System Specification Language inter-application dependency definitions advantageously can be used instead of settling times.

The QoS Managers FG44A - FG44N also receive application status and state information from Program Control FG50, which periodically sends application status updates for all running applications and also sends immediate indications of any applications which have been started or stopped. This information is used by the QoS Managers FG44A - FG44N, along with the instrumented performance data being received via the QoS Monitor FG29 and Instrumentation Correlator FG34, to determine the exact state of all monitored applications that are running. This information is also used to determine when (and if) requested actions have been implemented by the Resource Manager FG42. The information is also used for setting up and discarding internal data structures used for monitoring the performance of each ~~applicationA1-NM~~application A1-NM.

It will be appreciated that the QoS Managers FG44A - FG44N also receive application-level instrumentation data indicating current application performance values from the Instrumentation Correlators (Brokers) FG26A -FG26N, the Instrumentation Brokers

FG28A-FG28N, and/or the Jewel Instrumentation Broker (QoS Monitor) FG29. The instrumentation data that is received contains (at a minimum) (1) the timetag when the data was generated, (2) the hostname and IP address of the host where the application that the data is associated with is running, (3) the process id (pid) of the application that the data is associated with, and (4) the event number of the instrumentation message. Preferably, the event number of the instrumentation message specifies the type of instrumentation data that has been received; the hostname, IP address, and pid are used, in conjunction with the application data received from Program Control FG50, to determine the specific application that the data is associated with.

When the contents of the instrumentation message match any of the application performance requirements that are currently being monitored by the QoS Managers FG44A - FG44N, the data value is added to the proper requirement sliding window for the specified application. The sliding window algorithm is then checked to determine if the new sample triggered a violation of either the high or low sliding window. If a high threshold sliding window violation occurs and the application does not already have the maximum number of copies running, a determination is made as to whether performance can be best improved by starting a new application (scale up) or by moving an existing copy to a different host. The corresponding action recommendation will then be sent to the Resource Manager FG42. In an exemplary case, the criteria for determining whether an application should be moved rather than scaled up is based on relative performance of the replicated applications. More specifically, if one application is performing much worse [> 50%] than the other copies, the recommendation will be to move the application. Likewise, if the new sample triggers a low threshold sliding window violation and the application has more than the minimum number of copies running, a recommendation will be sent to the Resource Manager FG42 requesting that the copy of the application that is experiencing the worst performance be scaled down.

FG5 - RESOURCE (APPLICATION) CONTROL FUNCTIONAL GROUP

As discussed above, the Resource Control capabilities provided by the Resource Management Architecture consist of controlling application startup, configuration, and shutdown on hosts within the distributed environment. This capability, known as Application Control or Program Control (hereafter referred to as Program Control) provides a powerful distributed configuration capability. The Program Control capabilities permit an operator to startup and control applications running on platforms throughout the distributed environment via an easy-to-use interactive display. These capabilities are provided by the Application Control functional group FG5.

More specifically, the Application Control functional group provides application control (i.e., Program Control) capabilities which permit starting, stopping, and configuring applications on each of the hosts in the distributed environment. The functional group provides both interactive operator control of the distributed environment as well as automatic control via configuration orders received from the Resource Allocation Decision-Making functional group FG4, i.e., the Resource Manager component. The interactive controls allow an operator to create, load, save, and edit pre-defined system configurations, e.g., lists of applications that are to be run, with or without specific host mappings, determine the status and configuration of currently running programs, and start and stop any or all applications. Both static (operator-entered) mappings of applications to hosts and dynamic mappings of applications to hosts (where the Resource Allocation Decision-Making functional group FG4 will be queried to determine the proper mapping at run-time) advantageously can be defined. The functional group also provides application fault detection capabilities which are triggered by the unexpected death, i.e., fault, of an application that was started by the functional group. A basic host fault detection capability is also provided which is triggered based on failure to receive heartbeat messages from functional group components running on a particular host.

A brief description of each function provided by the functional group FG5 is provided below; a detailed discussion of the Resource Control functional group FG5 and associated data flow will be provided in discussing FIGURE 4.

1) **Program Control Agents** FG52A- FG52N: A Program Control agent generally denote~~DFG52~~d FG52 resides on each of the hosts A-N (i.e., PCA - PCN). Each agent is responsible for providing direct control over application startup and shutdown of applications on its respective host. The agent receives control orders from the Program Control function FG50 and is then responsible for implementing the orders. In an exemplary case, the agents implement the orders via system call mechanisms specific to the particular operating system. In addition, the agent also provides feedback to the Control function FG50 regarding the current status of all applications running on a particular host.

2) **Program Control FG50** - maintains the application state information for the Program Control functional group FG5. It also serves as the decision-making component of the Program Control functional group. The Control function FG50 receives application control (startup, shutdown, or configuration) requests from the Program Control Displays FG54A - FG54N and from the Resource Management functional group FG4. Using information from the Specification Files FG32, these high-level control function requests are dynamically translated into specific control orders which are sent to the individual Program Control agents FG52A -FG52N. The program Control FG 50 also provides application status and configuration information back to the

Resource Manager FG42.

3) **Program Control Displays** FG54A - FG54MN - serve as the GUI for interactive control of distributed applications. The Program Control Displays FG54A - FG54MN allow an operator to see and control the status of applications running on each host in the distributed environment. The Program Control Displays FG54A - FG54MN also provide the user the ability to determine the status of each of the components of the Program Control architecture. Predefined scenario configurations defined in Program Control Configuration Files FG56 advantageously can be loaded and edited via the Displays. It should be mentioned that new Program Control Configuration Files can also be created and saved via the Displays. As illustrated in FIGURES 2A, 2B, Program Control Displays FG54A - FG54MN can be run simultaneously with application status changes being reflected at each display.

4) **Configuration Files FG56** - contain an ordered set of applications that can be loaded at the Program Control display and then either edited or executed. The Configuration Files can contain both dynamic and static application-to-host mappings. For static application-to-host mappings, an application will, by default, be started on a specified host. For dynamic application-to-host mappings, the application will have a default host to start on but the Resource Manager FG42 will be queried at run-time to determine where the application actually should be placed. The Configuration Files FG56 also contain all information on how to start, stop, and configure an application, with the exception of environment variable settings for the application which are set based on the System Specification Files FG32.

It should be mentioned here that the Program Control functional group employs the application startup and shutdown information defined in the System Specification Files FG32. When an application entry is first created interactively at one of the Program Control Displays FG54A - FG54N, all of the startup and shutdown information for that application, as specified in the System Specification Files FG32, are loaded in as default settings. Once a configuration file entry has been created, all configuration information on the application is read in from the configuration file except for the application environment variable settings which are still set based on the System Specification Files FG32.

As mentioned above, a Program Control agent resides on each host. The agent is responsible for providing direct control over application startup and shutdown. The agent receives control orders from the Control component and is then responsible for implementing the orders. Each of the PC Agents FG52A - FG52N implements application startup and shutdown orders via system call mechanisms specific to the particular operating system of the host. For example, on the Unix platforms, to start an application, the fork() and execv() function calls are used to create the application. The csh command is executed to start up the applications. Moreover, if the application needs to run in a console, an xterm is configured for the application to run in. In addition, if logging of either stdout or stderr is specified, the proper redirection operators are configured and the output log file is set to "/usr/tmp/<userid>_<appname>_<pid>.log". All environment variables needed by the application are also configured and passed in at the execv() call. The current working directory is also set by the chdir() command, and the new application is made a process group leader via the setpgid() function. Other operating systems invoke applications using different calls.

In order to stop an application on the Unix platforms, if a signal is to be sent to the application, the killpg() function is used, or else if a script or command is to be executed to shutdown the application, the csh command is executed (via the system() function)

specifying the full path and executable name of the command along with any arguments for the command. It should be noted that if the application default shutdown time elapses and the application has not died, the respective one of the Program Control Agents FG52A-FG52N advantageously sends a SIGKILL signal to the application by calling killpg().

As illustrated in FigFIGS. 11A, 1B, the Program Control Agents (PCA-PCN) advantageously can be instantiated on stand-alone hosts A - N. In that case, the Program Control Agents PCA-PCN (FG52A-FG52N in FigFIGS. 22A, 2B) send heartbeat messages to Program Control FG50 approximately once per second to indicate that they are still "up and running." Moreover, every ten seconds, the Program Control Agents PCA-PCN (FG52A-FG52N) send complete configuration information on all running applications to Program Control FG50. It should be noted that the terminology employed in FigFIGS. 11A, 1B differs from that in FigFIGS. 22A, 2B to emphasize the distinction between software instantiated on a host and a function provided by the Resource Management Architecture.

The Program Control function FG50 is the decision-making component of the Program Control functional group FG5. It maintains complete information on everything that is running across all platforms in the distributed environment. The Program Control function FG50 receives input data from PCA-PCN (FG52A-FG52N), the Program Control Displays FG54A-FG54N, the Resource Manager FG42, and the Host Discovery function FG14.

It will be appreciated from the preceding discussion that the Program Control FG50 provides startup and shutdown orders to the Program Control Agents FG52A-FG52N based on operator or Resource Manager-initiated orders. If the Program Control Agents report that an application has terminated abnormally, the Program Control FG50 provides a notification to the Resource Manager FG42, to the Program Control Displays FG54A - FG54N, and to any other component to which it is connected. When the Program Control function FG50 is first brought up, it can be configured to attempt to start Program Control agents on every host

defined in the System Specification Files. The Program Control function FG50 will also attempt to start a Program Control Agent on a newly discovered host (discovered via the Host Discovery function FG14) if Host Discovery has been enabled on the Program Control Displays FG54A-FG54N.

The Proggram Control function FG50 also receives periodic heartbeat messages, e.g., once per second, from each of the Program Control Agents FG52A-FG52N, as discussed above. If Fault Detection has been enabled at the Program Control Displays FG54A-FG54N, if three consecutive heartbeat messages from an Agent, e.g., FG52A, are missed, the host that the agent is running on is declared down and all linked functions, including the Resource Manager FG42 and the Displays FG54A-FG54N are notified.

As mentioned above, the Proggram Control function FG50 sends out periodic application status updates as well as immediate notification when applications are started up, are shutdown, or fail. These notifications are sent out to all linked functions.

It should be noted that the Proggram Control function FG50 uses the same message traffic and internal processing for handling application startup and shutdown orders received from either the Resource Manager FG42 or from the Program Control Displays FG54A-FG54N. However, if a startup order received from one of the Program Control Displays FG54A-FG54N indicates that the Resource Manager FG42 should determine where to run the application, a request to allocate the application is sent to the Resource Manager FG42. When no response is received from the Resource Manager FG42 within a predetermined timeout period, the Proggram Control function FG50 will automatically start the application on the default host. Moreover, when an application startup cannot proceed due to an unfulfilled application startup dependency, a request will be made to the Resource Manager FG42 to attempt to resolve the dependency. If the Resource Manager FG42 either cannot resolve the dependency or no response is received within a predetermined timeout period,

the application startup will fail, and a "dependency failed" indication will be sent to the Display. It will be appreciated that this will cause the application status to be displayed in, for example, yellow and post an alert to the Alert window on one of the Program Control Displays FG54A-FG54N.

Preferably, Program Control function FG50 also handles simple startup timing dependencies between applications and will reorder a list of applications that were selected to be started simultaneously if doing so will resolve startup order dependencies between the applications. Otherwise, the Program Control function FG50 sends a request to the Resource Manager to attempt to resolve the dependencies.

The Program Control Display serves as the operator console for controlling the distributed environment. From the Display, shown in FigFIGS. 55A, 5B, the operator can:

1) see the status and configuration of currently executing applications A1-NM;
2) see the status of Program Control Agents PCA-PCN on each host A-N;
3) see and browse the application system structure defined in the System Specification Files FG32;
4) load configuration files FG56
5) save configuration files FG56
6) edit the configuration of applications that are not currently running;
7) create new application entries by dragging an application, application system, or application subsystem icon onto the application status area;
8) manually start specific applications;
9) manually stop specific applications;
10) manually start all applications that have the "Start All" flag set;
11) manually stop all applications;
12) turn host fault detection on or off (if on, loss of 3 consecutive heartbeats from a Program Control Agent will result in declaring the host down); and

13) turn host discovery on or off (if on, a new host message from the Host Discovery component will result in attempting to start up a Program Control Agent on the new host).

It will be appreciated from FigFIGS. 22A, 2B that multiple Program Control Displays FG54A-FG54N advantageously can be run simultaneously. If this is done, any configuration change actions will be reflected on all the displays. Whenever application stop or start actions are taken by the display operator, a message is sent to the Proggram Control function FG50 which is responsible for enacting the start or stop action. The Program Control function FG50 also sends indications of any status changes to the Program Control Displays FG54A-FG54N as soon as the status changes are seen. In addition, periodic status updates are also sent to the Program Control Displays FG54A-FG54MN.

The Program Control Configuration Files are text files that are read in by the Program Control Display when the operator wishes to load a new application configuration. A Configuration File is an ASCII file containing a list of applications. The format of an entry in a Configuration File is shown in Table 1 below.

Table 1

```
Application      TACFIRE:tacfire
Host             electral
Display      umbriel1:0.0
Auto_Start       0
RM_Start         0
Console      1
Time_Delay       1
StartupDir       "$ENV_SIM_VERSION/TACFIREprocessor"
StartupExe       "$ENV_SIM_VERSION/TACFIREprocessor/tacfire"
StartupArgs      "-disport $DIS_PORT_NUM -cffhost %(HOSTNAME,
                 AAW:Tactical_Sims:CFF_Broker)"
ShutdownExe      SIGINT
LogType          STDOUT
LogDir       "/usr/tmp"
```


Ex.1002 / Page 342 of 541
TESLA, INC.

The Configuration file advantageously can include the following fields:

1) The **Application** field, which identifies the full application name as defined in the System Spec. Files FG32 (i.e., System:Subsystem:Application).
2) The **Host** field, which is the desired or default host that this application should be started on.
3) The **Display** field, which is an optional field used when graphical display output from an application needs to be rerouted to a display on a different host.
4) The **Auto_Start** flag, which identifies whether the application is to be started automatically if the "Start All" action is selected by the operator from the Program Control Display. (If the flag were set to "1", then the application would be started. If the flag were set to "0," it would not be started.)
5) The **RM_Start** flag, which identifies whether the the Resource Manager should be queried at run-time to determine what host the application should be started on. The valid values are "0" for "NO" and "1" for "YES".
6) The **Console** flag, which identifies whether the application needs to be started in an Xterm window. The valid values are 0 for "NO" and 1 for "YES".
7) The **Time_Delay** field, which identifies how many seconds to wait after the previous application has been started before starting this application.
8) The **StartupDir** field, which identifies the current working directory that is to be set prior to starting up the application. This directory is usually the same as the directory where the executable for the application resides but does not have to be. As this example shows, environment variables may be used in the path.
9) The **StartupExe** field identifies the entire path and name of the application executable.
10) The **StartupArgs** field, which contains all the argument values needed for this particular application. As this example indicates, the argument values can

be dynamically set at run time if needed. Environment variables may also be used within the argument list. In this example, the %(UNIQUE, 1, 40, Isis) argument would yield a number from 1 to 40 which is unique within a context named "Isis". Another resolution of %(UNIQUE, 1, 40, Isis) would yield a different number.

11) The **ShutdownExe** field, which identifies which signal defined within the application that program control is to use to shutdown this application. Some examples would be SIGINT, SIGTERM, or SIGKILL. A shutdown script can also be used to shutdown the application. (In that case, there would be **ShutdownDir, ShutdownExe,** and **ShutdownArgs** fields listed. The usage for the shutdown fields would be used exactly the same as the startup fields.)
12) The **LogType** field, which identifies which outputs are to be written to the specified log file. The valid values are STDOUT, STDERR, and LOG_ALL. STDOUT is the normal output of the application (stdout). STDERR is the error output of the application (stderr). LOG_ALL writes both stdout and stderr outputs to the file.
13) The **LogDir** indicates the directory where the log file will be written. Again, environment variables may be used here. The log file name will be "<userid>_<appname>_<pid>.log" where <appname> is the full application name as specified in the **Application** field, <userid> is the userid of the current user under which the program control application is running, and <pid> is the system assigned process id of the application being executed.

FG6 - DISPLAY FUNCTIONAL GROUP

A number of displays which show system configuration data and instrumentation data in near real-time are included as part of the Resource Management Architecture. These displays support operator and user monitoring of the operation of the distributed environment

including host and network statuses and performance, application system statuses and performance, as well as the status and performance of the other Resource Management architecture functions. Most of the displays use OpenGL and Motif, the latter being built with ICS's Builder Xcessory toolkit, and run on Silicon Graphics (SGI) platforms in an exemplary case. Several of the displays can also run on the Sun Solaris platforms. The displays that make up the display functional group FG6 include:

1) **Host Displays FG62A-FG62N.** Show layout of hosts along with host status, network connectivity, and process statuses.
2) **Path Display FG64.** Shows the status of applications in key end-to-end data flow paths along with performance and load graphs.
3) **Resource Management Decision Review Display FG68.** Shows a summary of allocation decisions made by the Resource Management system along with timing information and host fitness scores.
4) **Graph Tool Instrumentation Displays FG69A-FG69N.** Provides a user-configurable set of display widgets used for run-time monitoring of instrumented status and performance information.
5) **System Readiness Display FG66.** Shows the status of each hardware and software system, subsystem, and application defined in the System Specification Files and allow the operator to interactively change system and subsystem priorities.

~~Fig~~FIGS. 6~~is~~A, 6B represent a screen capture of an exemplary one of the Host Displays FG62A-FG62N, which provide graphical representations of various sets of the hosts A- N in the distributed environment. The Host Displays show the status of each host, host network connectivity, and the status of interesting processes running on the hosts. The Host Display operator can also select hosts shown on the Host Display and bring up real-time graphs of system performance for the selected hosts including CPU utilization, memory utilization, network packets in, network packets out, and paging activity. A screen capture

of host specific performance information is provided in ~~Fig~~FIGS. 7A, 7B.

~~Fig~~FIGS. 8~~is~~A, 8B represent a screen capture of a representative Path Display FG64, generated by the Resource Management architecture, which shows the status of key system data flow paths consisting of multiple application stages. The number of copies of each application in the path is shown labeled with the host on which the application is running. In addition, it should be mentioned that as many as three real-time graphs can be produced to depict run-time performance and load metrics related to the applications in the selected data path.

~~Fig~~FIGS. 9~~is~~A, 9B represent a screen capture of the Resource Management Decision Review Display FG68, which advantageously can provide a summary of allocation and reallocation actions taken by the Resource Manager FG42. For each action, timing information regarding how long it took the Resource Management functions, e.g., the Resource Manager FG42 and the Program Controller FG50, to both arrive at a decision and to enact the decided action are shown along with host fitness scores that were used in arriving at the allocation decision.

~~Figs~~FIGS. ~~10~~10A, 10B and ~~11~~11A, 11B are screen captures of the Graph Tool Instrumentation Displays FG69A-FG69N, which depict user-configurable displays capable of receiveing data via standardized message formats and open interfaces. The Graph Tool Displays FG69A-FG69N allow the operator to select and configure various display widgets (line graphs, bar charts, pie charts, meters, and text boxes) to build a desired display layout. Data sources for driving the widgets can also be selected interactively.

~~Fig~~FIGS. 12~~is~~A, 12B represent a screen capture of the System Readiness Display FG66, which advantageously can be a Java™ display with a CORBA™ interface. The display FG66 shows the status of each hardware system, host, application system, application

subsystem, and application defined in the System Specification Files. The top portion of the display shows a summary status for each defined application system. It should be noted that the display operator can also change system and subsystem priorities and send the changed priorities to the Resource Manager function FG42.

As mentioned above, the RMComms middleware package provides object-oriented client-server services for message communication between distributed applications and function modules. The middleware provides location transparency and automatic socket connections and reconnections between client and server applications. These services advantageously can be accessed through an object-oriented API which allows client and server objects to be easily created and exchange user-defined message data. The abstraction provided by the API allows the user to quickly and easily create distributed applications without needing to be aware of the details of the underlying network mechanisms. The RMComms middleware provides the following functions:

- provides location transparency between clients and servers
- provides a simple powerful object-oriented client-server API
- supports reliable transport of user-defined message data
- based on Berkeley sockets
  - uses TCP for message transport
  - uses UDP multicast for identification of new clients or servers
  - servers identified by unique assigned UDP/TCP port numbers
- provides general purpose callback function registration capabilities
  - user-specified message callback functions invoked when specified messages arrive
  - user-specified connection status callback function invoked when new client-server connections are established or existing connections are broken
- support for multi-threading

- supports both polled and asynchronous I/O
- thread-safe
- provides automatic connections between clients and servers
  - supports multiple client and server connections within the same application
  - provides automatic connections to new clients / new servers
  - supports simultaneous many-to-many client-server connections
- no separate "naming service" or "application registration" components
- provides automatic client-server connection fault detection and recovery
  - provides fault detection mechanisms based on timeouts and broken connections
  - supports fault recovery via automatic reconnections between clients and servers
- provides basic support for data marshalling between machine architectures
  - byte-swapping
  - explicit message data type specification
  - all message data sent out using network byte order
- provides basic capabilities for reading the system clock and performing time conversions
- allows registration of user-defined signal (interrupt) handler functions
- layered object-oriented design and implementation
- cross-platform support:
  - SGI IrixIRIX 6.3/6.4/6.5
  - Sun Solaris 2.5.1/2.6/2.7/.2.8
  - HP HP-UX 10.20
  - Linux 2.1/2.2
  - Windows NT 4.0
  - Windows 95/98/2000

- Solarisx86 2.7
- C++ language support using native and GNU compilers

The RMComms middleware is implemented as a shareable object-oriented C++ library. The library provides four primary object classes, which are detailed in Attached Appendix C. It will be appreciated that the applications link with this library and can then instantiate client and server objects for communicating with other local or remote applications. It should be mentioned that the application source code must also include a set of header files that allow connections between client and server objects, where each server type is assigned a server port number. For clients and servers that want to communicate, both the client and the server objects are created specifying the same server port number. Multiple servers of the same type can also be created, which all use the same server port number. This advantageously provides the ability for many-to-many client-server connections to be established, as illustrated in FigIG. 4. Control of which servers the clients actually connect to is handled on the client side; clients can specify whether they wish to establish connections with all servers in the distributed environment, with a particular set of servers, or with all servers running on a particular set of hosts.

The operation of the Resource Management Architecture will now be described while referring to Figs. 13A-13C, which illustrate various operations in the distributed environment. More specifically, the Resource Management Architecture of the system illustrated in Figs. 13A includes hosts A-N, where host A provides a video source server application A-1, host B provides a video distribution application B-1, a contract application B-2, and a host load monitor B-3, and host C provides a display broker application C-1 applying video signals to a display driver C-2. It will be appreciated that host D is idle and that the connections between the various hosts constitute the network 100'. In addition, the Resource Management Architecture of FigIG. 13A instantiates various functions, e.g., an instrumentation broker FG26', a QoS manager FG44', a resource manager FgG42' and a

program control FG50'. The instrumentation broker FG26' receives data from each of the applications running in the distributed environment, although only the lines of communication between the applications running on host B are actually depicted. From the discussion above, it will be appreciated that each of the applications is linked to an Instrumentation API.

Referring now to FigIG. 13B, a QoS violation and its consequences is depicted. In particular, the Instrumentation broker FG26' provides data to the QoS manager FG44' which is indicative of a QoS violation. The QoS manager FG44' notifies the resource manager FgG42' of the violation; the resource manager determines that duplicate copies of the applications running on host B are required and that these copies should be placed on host D. The resource manager FG42' transmits instructions to the Program Control function FgG50', which starts copies of the running applications, i.e., a video distribution application D-1, a contract application D-2, and a host load monitor D-3, on host D. FigIG. 13C illustrates shutdown of the application copies running on host B. It will be appreciated that this shutdown may be initiated responsive to the original QoS violation, another QoS violation, or a query from the user.

Having discussed the various functions and features of the Resource Management Architecture in gross, selected functions and features will now be described in detail. It will be appreciated that the discussion of the various functions will be signaled using the designations established with respect to FigFIGS. 22A, 2B.

FG42 - Resource Manager Function

As mentioned above, the Resource Manager FG42 is the primary decision-making component of the Resource Management functional group. It is responsible for:

(1) responding to application and host failures by determining if and what

recovery actions should be taken;

(2) determining if and where to place new copies of scalable applications or which scalable applications should be shutdown when the QoS Managers indicate that scale-up or scale-down actions should be taken based on measured application performance:

(3) determining where new applications should be placed when requested to do so by Program Control: and

(4) determining which and how many applications should run based on application system (mission) priorities.

In order to accomplish these tasks, the Resource Manager FG42 maintains a global view of the state of the entire distributed environment including status information on all hosts, networks, and applications. In addition, the Resource Manager FG42 also calculates software and hardware readiness metrics and reports these readiness values for display purposes.

The Resource Manager FG42 is an object-oriented multi-threaded application written in C++, which uses the RMComms middleware for all external communication. The Resource Manager FG42 communicates with the various software components instantiating the (1) Program Control FG50, 2) Hardware Broker FG40, 3) QoS Managers FG44A - FG44N, 4) QoS Specification Control FG29, 5) Readiness Broker in Readiness Display FG66, 6) Globus Broker (not shown), and 7) RM Decision Review Displays FG68A-FG68N.

It will be appreciated that the Resource Manager FG42 receives status and failure information about hosts and networks from the Host and Network Monitoring functional group FG1, and applications from the Program Control functional group FG5. This information includes periodic status updates as well as immediate updates when statuses change, e.g., when a new host is detected or an application fails. In the case of any application shutdown, information as to whether the applications were intentionally

shutdown or whether the application actually failed advantageously can be provided. The Program Control function FG50 also issues requests to the Resource Manager FG42 whenever new applications need to be dynamically allocated and whenever the Program Control function FG50 determines that the Resource Manager FG42 needs to assess and attempt to resolve inter-application dependencies (e.g., one application which needs to be running prior to starting up another application).

The Resource Manager FG42 responds to applications faults and host failures by determining whether the failed applications can and should be restarted and attempting to determine where (and if) there are hosts available that the application can run on. When a decision is made by the Resource Manager FG42, a message is sent to Program Control function FG50 specifying what application to start and where to put it. The same general mechanism is used when the Program Control function requests that the Resource Manager FG42 determine where to start new applications and/or how to resolve inter-application dependencies; the Resource Manager FG42 responds with orders indicating what applications to start and where to start them. The Resource Manager FG42 advantageously can send application shutdown orders to the Program Control function FG50 requesting that a certain running application be stopped; this can occur when the QoS Managers indicate that certain scalable applications have too many copies running or when application system priority changes (to lower priorities) occur resulting in scaling back the application system configuration. See Figs. 13B and 13C and the associated discussion above.

The Resource Manager FG42 receives host load and host fitness information from the Hardware Broker (Host Load Analyzer) function FG40. This information includes overall host fitness scores, CPU-based fitness scores, network-based fitness scores, and memory and paging-based fitness scores along with the SPEC95 rating of the hosts. This information is received approximately once a second and includes information on all known hosts in the distributed system. These scores are used by the Resource Manager FG42 for determining

the "best" hosts for placing new applications when:

(1) responding to requests from the QoS Managers to scale up additional copies of an application;

(2) attempting to restart failed applications;

(3) responding to requests to dynamically allocate certain applications; and

(4) responding to application system (mission) priority changes which require scaling up additional applications.

Advantageously, the Resource Manager FG42 also receives requests from the QoS Managers FG44A - FG44N for scaling up, moving, or scaling down specific applications. The Resource Manager FG42 responds to these requests by determining whether the request should be acted upon and, if so, determines the specific action to take and issues orders to the Program Control function FG50 to start up or shutdown specific applications on specific hosts. The QoS Managers FG44A - FG44N are responsible for monitoring specific system performance metrics (e.g., quality of service, or QoS, requirements) via instrumentation and determining if performance can be improved by scaling up or moving certain applications. When this occurs, the QoS Managers send a request to the Resource Manager FG42 indicating that a new copy of a specific application should be started. If the QoS Managers determine that the performance of a scalable application can be improved by moving an application, a scale up request is first sent to the Resource Manager FG42 and when the new application has been started, a scaledown request is then sent to the Resource Manager FG42. Moreover, when the QoS Managers FG44A -FG44N determine that there are more copies of scalable application running then are needed, requests to shutdown specific applications are sent to the Resource Manager FG42.

It will be appreciated that the Resource Management Architecture distributes functionality between the QoS Managers FG44A-FG44N and the Resource Manager FG42. Thus, the QoS Managers determine what actions would potentially improve performance,

while the Resource Manager FG42 has final authority to determine whether to implement the requested actions.

It should be noted that when the Resource Manager FG42 is first started, it reads in the System Specification Files FG32 (via calls to the System Specification Library (SSL) FgG34) which contains the list of hosts that are known to be (operating) in the distributed environment and information on all applications that can be run in the distributed environment. The application-level information includes where specific applications can be run, which applications are scalable, which applications can be restarted, and any dependencies between applications. In addition, the Resource Manager FG42 receives updated application survivability specifications from the QoS Specification Control function. This information overrides the application survivability information that was initially loaded in from the System Specification Files FG32 for the specified application. The information is used by the Resource Manager FG42 to determine whether the specific application will be restarted if it fails at run-time.

It should also be noted that the Resource Manager FG42 sends application system and hardware system readiness and system (mission) priority information to the Readiness Broker, which is a translator withing the Readiness Display FG66 and to the Globus Broker (another Broker (not shown)). The Readiness Broker is responsible for driving a GUI/display FG66, which shows the current readiness data and allows the system (mission) priorities to be changed and sent back to the Resource Manager FG42. The Globus Broker provides basically the same functionality except that only a high-level subset of the readiness data provided to the Readiness Broker is provided to the Globus Broker. The readiness information sent to the Readiness Broker consists of readiness values for each application, application subsystem, and application system defined in the System Specification Files FG32. The scores advantageously can be based on the status (up/down) of the applications and the percentage of potential copies of scalable applications that are currently running.

Host and network readiness scores are determined based on the host loads and host fitness scores received from the Hardware Broker FG40.

The Resource Manager FG42 also sends information about allocation and reallocation decisions to the RM Decision Review Display FG68 (~~Fig~~FIGS. 9A, 9B). Information on the decision that was made, what event the decision was in response to, and how long it took to both make the decision and implement the decision are sent to the display. In addition, information about the top choices for where an application could have potentially been placed is also sent (if applicable); this information includes the host fitness scores for the selected host and other hosts which could have been selected.

As described above, the Resource Manager function FG42 communicates with Program Control FG50, the Hardware Broker FG40, the QoS Managers FG44A -FG44N, QoS Specification Control (not shown - legacy function), the Readiness Broker of the Readiness Display FG66, the Globus Broker (not shown), and the RM Decision Review Display FG68 using the RMComms middleware. The message formats and contents of each message that is exchanged between the Resource Manager function FG42 and other functional elements of the Resource Management architecture are described in CD-Appendix D. The timing and/or event trigger for each message is also described.

FG40 - Host Load Analyzer (Hardware Broker) Function

The Hardware Broker FG40 provides the host load analysis function of the Resource Management functional group FG4. It is responsible primarily for determining the host and network loads on each host within the distributed computing environment. The Hardware Broker FG40 assigns a set of fitness scores for each host and periodically provides the list of fitness scores to the Resource Manager FG42. ~~Fig~~FIG. 14 illustrates the connectivity and high-level data flow between the Hardware Broker and the other Resource Management and

Resource Management-related components.

The Hardware Broker FG40 is an object-oriented multi-threaded application written in C++, which uses the RMComms middleware for all external communication. It receives operating system-level statuses and statistics for each host from the History Server(s) FG12A -FG12N. This information is used for calculating CPU, network, memory, paging activity, and overall fitness scores for each host. The Hardware Broker periodically (once per second) sends the list of host fitness scores to the Resource Manager FG42.

When the Hardware Broker FG40 is first started, it reads in the System Specification Files FG32 (via calls to System Specification Library (SSL) FG34) which contain the list of hosts that are known to be in the distributed environment. The Hardware Broker also reads in the file *networks.dat* which contains a list of information about the bandwidth and maximum packet sizes on known network subnets. It should be mentioned that this data is used for converting host network load information based on packet counts to load information based on bytes per second and percentage of available bandwidth.

It should be mentioned that there are two other RMComms interfaces that the Hardware Broker FG40 uses. Periodically (approximately every three seconds), the Hardware Broker FG40 sends a list of overall and network host fitness scores to the Hardware Broker Instrumentation Display FG69A – FG69N. As mentioned above, these displays were constructed using the Graph Tool described in the Instrumentation Graph Tool Display. Additionally, the Hardware Broker FG40 can receive host-based network load data from the Remos Broker FG16, which receives network data via the Remos Network Monitoring software (denoted 2 in ~~Fig~~FIGS. 22A, 2B). If Remos network data is available for any of the hosts that are being monitored, the Remos data is used for the network fitness score calculation for that host rather than the host network data received from the History Server(s).

The exemplary instance of the Hardware Broker FG40 is an object-oriented multi-threaded application. At the highest level, the Hardware Broker object contains the elements listed in Table II below. It will be noted that Table II contains a brief description of each of these objects. Additional details are provided in CD-Appendix E.

Table II

No.	Title	Description
1	Host Fitness Database object (FitnessDB class)	The Host Fitness Database object stores load history data and fitness score information for each host. The Host Fitness Database is updated and fitness scores are recalculated when new History Server Host Status Response Messages are received. For each host, a circular queue of host load history data (HostInstance class) is maintained with the newest data being placed at the end of the queue; this history data is used for recalculating host fitness scores. The Host Fitness Database also contains a System Specification Library (SSL) object which is used to access SPEC rating information for the hosts.
2	Signal Registration object (SignalRegistry class)	The Signal Registration object allows for a user-defined SIGINT signal handler to be registered in order to permit the Hardware Broker ~~FG40~~toFG40 to be shutdown gracefully.

3	Network Subnet Information Database object (SubnetDB class)	The Network Subnet Information Database object is used to store IP address, maximum bandwidth, and MTU size for each network specified in the *networks.dat* file. This information is used for converting network packet load information to bytes/second network load information.
4	Remos Host Network Bandwidth Database object (RemosDB class)	The Remos Host Network Bandwidth Database object stores the latest Remos-reported network bandwidth information for each host being monitored. The information stored consists of available bandwidth as well as maximum potential bandwidth on a specific host network link. If Remos bandwidth information is available for a host and the latest data is less than 5 seconds old, the Remos data will be used for calculating the network fitness score for the host.
5	History Server Interface object (HistServInterface class)	The History Server Interface object inherits from the RMComms TCPCommClient class and is responsible for maintaining connections to the History Server(s), for registering status and message handler callback functions, for sending messages to the History Server(s), and for invoking the status and message handler callback functions when connections to History Servers are either established or broken or new messages are received from a History Server.

6	Instrumentation Graph Tool Display Interface object (InstrInterface class)	The Instrumentation Graph Tool Display Interface object inherits from the RMComms TCPCommServer class and is responsible for maintaining connections to the Graph Tool Display(s), for registering status and message handler callback functions, for sending messages to the Graph Tool Display(s), and for invoking the status and message handler callback functions when connections to Graph Tool Displays are either established or broken or new messages are received from a Graph Tool Display.
7	Resource Manager Interface object (ResMgrInterface class)	The Resource Manager Interface object inherits from the RMComms TCPCommServer class and is responsible for maintaining connections to the Resource Manager for registering status and message handler callback functions, for sending messages to the Resource Manager, and for invoking the status and message handler callback functions when connections the Resource Manager are either established or broken or new messages are received from the Resource Manager.

8	Remos Broker Interface object (RemosInterface class)	The Remos Broker Interface object inherits from the RMComms TCPCommClient class and is responsible for maintaining connections to the Remos Broker for registering status and message handler callback functions, for sending messages to the Remos Broker, and for invoking the status and message handler callback functions when connections the Remos Broker are either established or broken or new messages are received from the Remos Broker.

FG44: Quality-of-Service (QoS) Manager Function

The QoS Managers FG44A - FG44N are responsible for monitoring application-level performance requirements, which requirements are defined in the System Specification Files FG32 and are monitored primarily via instrumentation data obtained directly from the application code. The QoS Managers FG44A - FG44N advantageously determine if applications or application paths are satisfying their assigned requirements. When an application is not meeting its performance requirements and the application is scalable (in the sense that multiple copies can be run and the copies will perform load-sharing across the copies), the QoS Managers FG44A - FG44N will either request that the Resource Manager FG42 scale up a new copy of the application or move the application to a new host (which hopefully will result in better performance). Moreover, if there are multiple copies of a scalable application running, and all copies are performing below the specified requirement threshold, the QoS Managers FG44A - FG44N will request that the Resource Manager shutdown a specific copy.

The QoS Manager is a single-threaded application written in C/C++. It should be

noted that the application can be scaled for both redundancy and/or load-sharing. In an exemplary case, each copy of the QoS Manager monitors all of the requirements associated with a single application path defined in the System Specification Files FG32. It will be appreciated that the specific path to be monitored can be specified via command-line parameters. By default, without specifying a path via the command-line, the QoS Manager will monitor all requirements for all defined paths.

As mentioned above, the QoS Manager advantageously uses a sliding window algorithm to determine when to declare that applications should be scaled up or scaled down. The inputs to the algorithm define both high and low sampling window sizes, the maximum number of allowed violations within the sampling window, and violation thresholds as a percentage of the actual specified requirement value. It will be appreciated that the sliding window algorithm was selected in an effort to damp out unexpected "noise" or "spikes" in the measured performance data. Use of threshold values states as a percentage of the actual requirement value was selected in order to scale up, or scale down, prior to violating the specified hard requirement. It will be understood that the success of this approach is highly dependent on the rate of change and noisiness of the measured data..

Again, the QoS Manager uses the RMComms middleware for all external communication. Each copy of the Resource Manager talks to (1) Resource Manager FG42, (2) Program Control FG50, (3) QoS Specification Control (not shown), (4) QoS Monitor FG29, (5) Instrumentation Correlators FG26A –FG26N, (6) Graph Tool Instrumentation Displays FG69A-FG69N, and (7) History Servers FG12A-FG12N. iln an exemplary case, the QoS Managers FG44A - FG44N advantageously can receive configuration orders from the Resource Manager FG42, which allows the Resource Manager FG42 to configure each QoS Manager to monitor specific application paths and also set the sliding window criteria to be used by each respective QoS Manager.

Each copy of the QoS Manager advantageously can transmit application scale up and scale down requests to the Resource Manager FG42 when the measured performance data for a respective application violates either the high (scale up) or low (scale down) sliding window criteria for a specific requirement. A scale up request indicates which application on which host has violated the performance criteria, and a scale down request indicates which application on which host is recommended to be shutdown. Each copy of the QoS Manager can also request that the Resource Manager move an application. This will occur in the case where one copy of an application is performing much worse than all other running copies. The move request is implemented as a scale up request followed by a scale down request (of the badly performing copy); the scale down request is not transmitted to the Resource Manager FG42 until the scale up action has been implemented.

The QoS Managers FG44A - FG44N use the application "settling times" defined in the System Specification Files to ensure that once a requested action has been sent to the Resource Manager that no additional actions are requested until after the application settling time has elapsed. This provides time for initialization and configuration among the application copies to occur. In future releases, the inter-application dependencies will be used instead.

The division of responsibility between the QoS Managers FG44A - FG44N and the Resource Manager FG42 is as follows:

(1) the QoS Managers FG44A - FG44N determine what actions would potentially improve performance; and

(2) the Resource Manager ~~FG42has~~ FG42 has final authority to determine whether to implement the requested actions

It should be mentioned that there is a Request Acknowledge message from the Resource Manager FG42 which has been defined and implemented within the QoS Manager

code. This message is intended to provide feedback to the QoS Manager indicating that the request had been successfully received and whether the Resource Manager FG42 intends to implement the request.

As previously mentioned, the QoS Managers FG44A - FG44N receive application status and state information from the Program Control function FG50. Program Control periodically sends application status updates for all running applications and also sends immediate indications of any applications which have been started or stopped. This information is used by the QoS Managers FG44A-FG44N, along with the instrumented performance data being received via the QoS Monitor FG29 and Instrumentation Correlators FG26A-FG26N, to determine the exact state of the monitored applications A1-NM that are running. This information is also used to determine when (and if) requested actions have been implemented by the Resource Manager FG42. The information is also used for setting up and discarding internal data structures used for monitoring the performance of each application.

The QoS Managers FG44A - FG44N also receive application-level instrumentation data indicating current application performance values from the Instrumentation Correlators FG24A-FG24N, the Instrumentation Brokers FG26A-FG26N, and/or the Jewel Instrumentation Broker (QoS Monitor) FG29. The instrumentation data that is received contains (at a minimum):

~~39.~~ (1) the timetag regarding when the data was generated;

~~40.~~ (2) the hostname and IP address of the host where the application that the data is associated with is running;

~~41.~~ (3) the process id (pid) of the application that the data is associated with; and

~~(42.)~~ the event number of the instrumentation message.

The event number of the instrumentation message specifies the type of instrumentation data that has been received and the hostname, IP address, and pid are used, in conjunction with

the application data received from Program Control, to determine the specific application that the data is associated with.

If the contents of the instrumentation message match any of the application performance requirements that are currently being monitored by the QoS Manager, the data value is added to the proper requirement sliding window for the specified application. The sliding window algorithm is then checked to determine if the new sample triggered a violation of either the high or low sliding window. If a high threshold sliding window violation occurs and the application does not already have the maximum number of copies running, a determination is made as to whether performance can be best improved by starting a new application (scale up) or by moving an existing copy to a different host. The corresponding action recommendation will then be sent to the Resource Manager. In an exemplary case, the criteria for determining whether an application should be moved rather than scaled up is based on relative performance of the replicated applications. Thus, if one application is performing much worse [> 50%] than the other copies, the recommendation will be to move the application. Likewise, if the new sample triggers a low threshold sliding window violation, and the application has more than the minimum number of copies running, a recommendation will be sent to the Resource Manager FG42 requesting that the copy of the application that is experiencing the worst performance be scaled down.

It will be appreciated from the discussion above that when a copy of the QoS Manager is first started, it reads in the System Specification Files FG32 (via calls to System Specification Library (SSL) FG34), which contain the list of hosts that are known to be in the distributed environment and information on all applications that can be run in the distributed environment. The application-level information includes where specific applications can be run, which applications are scalable, which applications can be restarted, and any dependencies between applications.

It should also be mentioned that the Resource Manager FG42 receives updated application survivability specifications from the QoS Specification Control component. This information overrides the application survivability information that was initially loaded in from the System Specification Files for the specified application. The information is used by the Resource Manager FG42 to determine whether the specific application will be restarted if it fails at run-time.

As described above, the QoS Managers FG44A - FG44N communicates with the Resource Manager FG42, Program Control FG50, the QoS Specification Control (not shown), the QoS Monitor FG29, an Instrumentation Correlator (generally denoted FG24), a Graph Tool Instrumentation Display (generally denoted FG69), and the History Servers FG12A-FG12N using the RMComms middleware. The message formats and contents of each message that is exchanged between the QoS Managers FG44A - FG44N and these other functional components are described in greater detail in CD-Appendix F. Additional details regarding the timing and/or event trigger for each message is also described in the Appendix.

FG3: SYSTEM SPECIFICATION LANGUAGE & SYSTEM SPECIFICATION LIBRARY (SSL) FUNCTIONS

In order to effectively manage a pool of computing resources, the Resource Manager FG42 requires some means or mechanism of determining the capabilities and configuration of the computing resources under its control, as well as the software components that need to be executed and the dependencies of these software components on both hardware and software resources. Additionally, the Resource Manager FG42 requires the capability to determine the expected mission-level and application-level requirements. Furthermore, the Resource Manager FG42 must be able to determine what control capabilities are available to be used to attempt to recover from fault or QoS violation conditions.

In order to address these needs, a System and Software Specification Grammar has been developed to capture the "static" information needed by the Resource Manager FG42 for effectively managing a pool of distributed resources. The grammar captures the following information:

- **Hardware and Operating Systems**
  - Hardware Configuration
  - Network Configuration
  - Operating System and Version
- **Software**
  - Systems, Subsystems, Applications, Processes
  - Resource Requirements
  - QoS Requirements (Events)
  - Survivability Requirements
  - Path Information: Structure and QoS Requirements

As part of the grammar development effort, a specification library has also been developed that parses the specification files and provides an API for accessing the specification information. It will be noted that the specification library was written in C++ and has been ported for all development platforms including Solaris 2.6, Solaris 2.7, Irix 6.5, HP-UX 10.20, Red Hat Linux 6.0, and Windows NT 4.0. The library advantageously can be used by substantially all of the Resource Management functional elements, including Program Control FG50, Resource Manager FG42, Path QoS Managers, Hardware Broker FG40, and History Servers FG12A-FG12N.

As illustrated in FigIG. 3, the API library consists of a yaccs file FG302 that defines the BNF grammar, a lex file FG304 that defines the tokens of the language, and a set of C++ classes FG306 that store the spec file information. The lex file FG304 is compiled with the

GNU tool flex FG310 and it creates a C++ source file FG320. The GNU tool bison FG 312 compiles the yacca file FG302 and creates C++ source and header files FG322 and FG324. It will be noted that the lex source file FG304 includes the yacca header file FG322. The C++ compiler FG314 then compiles these two source files to create lex and yacca objects FG330 and FG332. The C++ compiler FG314 also compiles the C++ storage classes FG334. All of these objects are linked into a single library FG34 to be utilized by an application. FigIG. 3 illustrates this process flow.

The Software Specifications Grammar (S:SG:) provides techniques for describing the characteristics and requirements of dynamic, path-based real-time systems as well as providing abstractions to describe the properties of the software, such as hierarchical structure, inter-connectivity relationships, and run-time execution constraints. The S:SG: also allows description of the physical structure or composition of the hardware such as LANs, hosts, interconnecting devices or ICEs (such as bridges, hubs, and routers), and their statically known properties (e.g., peak capacities). Furthermore, the Quality-of-Service (QoS) requirements on various system components advantageously can be described.

At the highest level, a specification consists of a collection of software systems, hardware systems, and network systems. The language rules for specifying systems are described generally below and in detail in CD-Appendix G. The system specification language hierarchy is shown below; selected details will be presented immediately following.

- **Software Specifications**
  - Application
    - Security
    - Configuration
      - Hardware Requirements
      - Startup Info

      - Dynamic Arguments
    - Shutdown Info
      - States
    - Dependencies
    - Initial Load Estimate
    - QoS Info
      - Survivability
      - Scalability

- **Hardware Specifications**
  - Host Info
  - Network Info
    - LANs
    - Network Devices (Interconnects)

- **Path Specifications**
  - Data Flow Graph
  - Data Flow Info
  - QoS Requirements

It will be appreciated that a software specification is a collection of software systems, each of which consists of one or more software subsystems. Specification files are provided by the developer to capture as much knowledge about their software system as possible. These files provide a model of the actual systems which can be used by the Resource Manager FG42 at run-time.

In contrast, an application is an executable program that can be started as an autonomous process on a host. Application attributes include all information necessary to startup and shutdown the application. Associated startup block and the shutdown blocks describe how to start and stop the application and includes information such as the directory and name of the application, command line options, and environment variable settings.

An application instantiates an SSL object by calling its constructor. This parses the spec files in the specified directory and populates the object hierarchy to provide the data to the application. The SSL class contains an SSL_Container member, that holds the spec file data in its lists and maps. All the systems from the spec files are contained in the appropriate list, software systems in the swSysList, hardware systems in hwSysList, and network systems in nwSysList. The pathList contains all the paths in the spec files. The hostList contains all the hosts in the spec files; this list is also available from the entries in hwSysList. The processList contains a list of processes from the CONFIGURATION block. Moreover, it should be noted that one or more configuration blocks can exist per application. For example, an application that runs on more than one platform would have multiple CONFIGURATION blocks with different platforms in each HARDWARE block.

The application startup block contains all the information necessary to, automatically or manually, start an application. This information includes supported hardware (host) type, operating-system type, and operating-system version(s). This may be further constrained by an optional list of the names of hosts that can run the application. The startup information also includes the working directory for reading and writing data files, the name of the executable, and an ordered list of arguments that must be passed on the command line when the application is started. Last is a list of processes expected to be seen on the system when the application is running.

An application shutdown block indicates the command(s) to be used for termination

of the application. A shutdown command may be a POSIX signal name or may be a shell script or batch file. Supported signals include SIGKILL, SIGQUIT, SIGHUP, SIGUSR1, SIGUSR2, SIGSTOP, SIGINT, and SIGTERM. The ShutdownTime parameter is the maximum time to wait for the an application to die gracefully before forcing the application to terminate via the SIGKILL signal.

Other blocks are available. For example, a dependency block indicates any dependencies the application may have with the startup and/or shutdown of other applications (e.g., it may be required that a particular application be started before another application can be started). It will be noted that the dependency block is used by both Application Control FG50 and the Resource Manager FG42 to determine whether or not it is safe to start an application, stop an application, or let an application continue to run.

The scalability specification for an application indicates whether an application can be scaled via replication. Scalable applications are programmed to exploit load sharing among replicas, and can adapt dynamically to varying numbers of replicas. The specification also indicates whether an application combines its input stream (which may be received from different predecessor applications and/or devices), and splits its output stream (which may be distributed to different successor applications and/or devices) are also specified. "Combining" and "splitting" are commonly called "forking" and "joining" in parallel computing paradigms.

Specification files advantageously can be provided to describe a given set of networks that exist in a distributed runtime environment. A network system specification describes the LANs and ICEs (interconnection devices such as switches, hubs and routers). A system consists of one or more subsystems. A subsystem may contain LANs (each with an associated peak bandwidth specification) and ICEs (each containing a description of network membership).

Advantageously, a real-time QoS requirement specification includes timing constraints such as simple deadlines, inter-processing times, and throughputs. A simple deadline is defined as the maximum end-to-end path latency during a cycle from the beginning to the end of the path. Inter-processing time is defined as a maximum allowable time between processing of a particular element in the path. The throughput requirement is defined as the minimum number of data items that the path must process during a unit period of time. Each timing constraint specification may also include items that relate to the dynamic monitoring of the constraint. These include minimum and maximum slack values (that must be maintained at run-time), the size of a moving window of measured samples that should be observed, and the maximum tolerable number of violations (within the window).

CD-Appendix G described a specification grammar for declaring requirements on applications in a dynamic, distributed, heterogeneous Rresource pool. The grammar allows the description of environment-dependent application features, which allows for the modeling and dynamic Rresource management of such systems.

A common API was developed to allow Resource Management functions access to the information contained in the spec files. This is an object oriented API is, in an exemplary case, written in C++, with libraries ported to all supported platforms. The object is populated by parsing the spec files using the BNF grammar defined by lex and yacca syntax and compiled with GNU tools flex and bison, as discussed above. Actual population occurs in the semantic actions of the yacca file.

The SSL_System class is a generic class that can hold data for a software system, hardware system, or network system. The type member describes the type of system it contains. It also contains a pointer to its parent (it allows for nested systems of the same type), and a name of the system. The sysList contains its SSL_System children, and compList contains a list of the system's components (a list of hosts, for a hardware system for

example).

Preferably, the Application Program Interface (API) for the System Specification Library (SSL) FG34 uses the C++ Standard Template Library for data structures such as linked lists and hash tables (maps). An application first instantiates the SSL object by calling its constructor with the name of the directory where the specification files reside. This object contains functions that allow setting this directory after calling its constructor (setSpecDir(directory name)), clearing the object of all currently held data (clear()), parsing a specific file (parseSpec(filename)), and rebuilding the object (rebuild(), implicitly clears the object first). Once instantiated, this object provides access to the data in the specification files. CD-Appendix G provides additional discussion regarding this aspect of the SSL. It will be appreciated that the SSL object provides methods that return all the data it contains. For example, the getSWSystems returns an STL list of all the software systems specified in the specification files. Each entry in this list provides its data by methods such as getSysName(), and the set of application components (ApplicationSpec) that make up the system. All data can be retrieved in this manner.

FG1: HOST AND NETWORK MONITORING FUNCTIONAL GROUP

As mentioned above, extensive monitoring capabilities are provided in the Resource Management architecture at the host and network levels. The information monitored includes statuses, configuration information, performance metrics, and detected fault conditions. Moreover, the Host and Network functional group FG1 consists of four components including:

1) **Host Monitors** FG10A-FG10N, that reside on each machine in the distributed environment and collect extensive operating system-level data for each host (CPU and memory usage, etc) and provides it to the History Servers

via the RMComms TCPCommServer middleware.

2) **History Servers** FG12A-FG12N that collect data from the Host Monitors, maintain status and performance histories on each host in the distributed environment via an RMComms TCPCommClient, and provide this information to displays and other Resource Management components using an RMComms TCPCommServer.

3) **A Host Discovery** function FG14 that uses SNMP (Simple Network Management Protocol) calls and ping ICMP calls to determine when new hosts come on-line and if existing hosts go down and providing this information to Program Control via an RMComms TCPCommServer.

4) **A Remos Network Data Broker** FG16 that collects information on network link bandwidths from Carnegie Mellon University's SNMP-based Remos tool and passes this information by way of an RMComms TCPCommServer to the Host Load Analyzer component of the Resource Allocation Decision-Making subsystem.

It will be appreciated that Network information is collected by both the Remos broker FG16 and indirectly via the Host Monitors FG10A-FG10N. See FigFIGS. 22A, 2B. The Remos Broker FG16 accesses the Remos network information via the Remos API. As mentioned previously, Remos uses SNMP calls to the LAN switches and hosts. The Host Discovery function FG14 uses both SNMP and ICMP (ping) calls to each host A-N to determine if a new host(s) has (have) come on-line or previously discovered hosts have gone down. The Host Monitors FG10A-FG10N employ Operating System calls to gather host and network performance statistics. Internally, the History Servers FG12A-FG12N collect data from the Host Monitors FG10A-FG210N. The Monitoring functional group provides its information to the rest of the Resource Management components using RMComms TCPCommServer objects, which are discussed in detail elsewhere. The Remos Broker FG16 sends data to the Host Load Analyzer FG40, the History Servers FG12A-FG12N send data

to the Display functional group FG6 and Host Load Analyzer FG40, and the Host Discovery function FG14 provides Program Control FG50 with information on detected or faulted hosts. Additional details on these functional elements are provided immediately below.

FG10A-FG10N Host Monitors

For monitoring the status and performance of hosts, a Host Monitor process runs on each machine within the distributed environment. These Host Monitors FG10A-FG10N use operating system-level mechanisms to retrieve status, configuration, and performance information of each host A-N. The information retrieved includes 1) operating system version and machine configuration, 2) CPU configuration, status, and utilization, 3) memory configuration and usage, 4) network configuration, status, and utilization, 5) filesystem configuration, status, and utilization, and 6) process statuses including CPU, memory, network, and filesystem utilization for each process. While the Host Monitors are primarily responsible for monitoring the status of a particular host, they also provide information on network load as seen by a particular host. In the same manner, the Host Monitors FG10A-FG10N also provide information and statistics concerning any remotely mounted filesystems (e.g., NFS).

Preferably, the information the Host Monitors FG10A-FG10N collect is formatted into operating system-independent message formats. These message formats attempt to provide a pseudo-standardized set of state, status, and performance information which is useful to other components of the Resource Management architecture and such that other components do not have to be aware of or deal with the minor deltas between data formats and semantics. Since not all the state and performance data is available on every platform, to indicate which information is available, a group of flags are set in the host configuration message indicating whether specific data items are valid on a particular platform.

It will be appreciated that the Host Monitors FG10A-FG10N have a very specific interface with the History Servers FG12A-FG12N. It periodically (once a second) sends its data to all History Servers connected to it (this is transparent, a property of the RMComms TCPCommServer); the History Server makes no requests to the Host Monitors.

More specifically, the Host Monitors FG10A-FG10N have been designed and implemented in C++. This decision allows for a completely modular design in which platform-specific code can be restricted to a small number of modules. This approach alleviates any of the problems associated with porting to various platforms. Currently there is support for Sun SPARC based architectures running Solaris 2.6 and 2.7, Silicon Graphics MIPS based architectures running IRIX 6.5., Hewlett Packard ~~PA-Risc~~PA-RISC based architectures running HP1020, and Pentium based architecture running both WinNT 4.0 Workstation and Red Hat Linux 6.0. The Host Monitor source compiles under the native compilers provided by Sun Microsystems and Silicon Graphics for their respective platforms. The Gnu C++ compiler (version 2.8.1) may also be used on Hewlett Packard ~~PA-Risc~~PA-RISC based architectures under HP-UX 10.20 and Red Hat Linux. Microsoft Visual C++ compiles the Windows NT Host Monitor. All Host Monitors utilize the I/O library package supported by the Resource Management (RM) group under the NSWC's High Performance Distributed Computing (HiperD) initiative.

The Host Monitors FG10A-FG10N accumulate data on a periodic interval specified at invocation. System process table data is accumulated and then filtered to eliminate "uninteresting" processes (usually meaning processes belonging to user ID 0 or 1). It is important to note that system-wide data is accumulated and processed *before* the filtering stage, so as to insure a complete picture of system-wide performance. This system-wide data, along with the filtered process list, is then made available to the I/O module for subsequent transmission to client applications.

The History Server function of Resource Management acts as a data broker between daemons monitoring individual hosts, known as host monitors FG10A-FG10N, and other functional components of Resource Management. The host monitors collect performance information (such as CPU utilization and process status data) from hosts of various platforms (SGI, SUN, HP, Windows NT, and Linux). The host monitors use a RMComms TCPCommServer object to distribute this data. For further information, refer to the host monitor and RMComms documentation. The History Server s FG12A-FG12N collect and store this data from the host monitors FG10A-FG10N and distribute it to other Resource Management Clients, such as the Host Displays FG62A-FG62N, Graph Display FG69A-FG69N, Path Display FG64, and the Hardware Broker. FG40

Each History Server has two modes of operation relating to fault tolerance, scalability, and workload distribution between multiple instances of History Servers. The first mode determines at initialization (through command line arguments or default) the set of hosts to monitor, and this set remains static for the life of the History Server process. The second mode recognizes the existence of other History Server processes and coordinates between them. It allows for dynamic changing of the set of hosts each History Server monitors (example: two History Servers each monitoring half of the hosts, a third History Server starts, and all three History Servers reconfigure to each monitor one third of the hosts.) This also allows History Servers to preserve the data it collected by sending it to the others, providing fault tolerance.

The History Server function is written in C++ with an object-oriented design. The main routine processes the command line arguments, retrieves the list of hosts to monitor using an SSL object, instantiates the main History_Server object, and spawns the Collector, Distributor, Communicator, and Display thread. These threads share the main History_Server object. The Collector thread is responsible for collecting and storing data from the host monitors. The Distributor thread processes requests from RM Clients. The

FG12A-FG12N: History Servers

The History Servers FG12A-FG12N are responsible for collecting information from the Host Monitors and maintaining histories on the statuses, statistics, and performance of each host in the distributed environment. This information can be requested by other Resource Management functional group. Currently, the primary consumers of the status information are the Host Load Analyzer (Hardware Broker) FG40 component of the Resource Allocation Decision-Making functional group FG4, the Host Display(s) FG62A-FG62N, and the Path Display FG64. The Host Load Analyzer FG40 receives information on host configuration and loads (primarily CPU, memory, and network data) and uses this to assign host fitness scores. The Host Displays FG62A-FG62N receive and display current host status information, process status information, and network connectivity information. It should be mentioned that the Host Display can also request that the History Servers provide CPU load information, network load information, paging activity data, and memory utilization information which is used to drive line graph charts for specific hosts selected at the Host Display.

The History Servers FG12A-FG12N are designed so that multiple copies can be run simultaneously. Each History Server can be configured to either monitor all Host Monitors FG10A-FG10N or to monitor only a selected subset of the Host Monitors. It will be noted that the History Servers ~~FG12A-FG12Ndetermine~~FG12A-FG12N determine the list of hosts in the distributed environment that could potentially be monitored from the System Specification Library (SSL). In this manner, the History Servers FG12A-FG12N can be used to provide survivability (by having multiple History Servers FG12A-FG12N connected to each Host Monitor) and/or to perform load-sharing (with the History Servers FG12A-FG12N each monitoring only a subset of the Host Monitors). The History Servers FG12A-FG12N can also be configured to periodically record history data to disk. These disk files can then be used for off-line analysis.

Communicator thread waits for events with other History Servers and takes appropriate actions, including triggering the Display thread to update the History Server Display.

FG14 - Host Discovery

The Host Discovery function FG14 advantageously can use a Perl script that makes SNMP (Simple Network Management Protocol) calls and ICMP ping calls. These calls are used to periodically scan each subnet and host address in the distributed environment to attempt to determine whether there have been any host status changes. In an exemplary case, the list of hosts and subnets that are to be monitored is read in from a file.

The host discovery FG14 issues MIB-II SNMP queries to obtain information on the hosts A-N on the network. When a new host is first detected, the new host's operating system configuration is queried via SNMP calls. Information on the newly discovered host and its operating system configuration is then sent to the Program Control function FG50. Likewise, when a host fails to respond to multiple SNMP and ping queries, a message indicating that the host appears to have gone down is sent to the Program Control function.

The Host Discovery function FG14 interfaces with Program Control FG50 using a C++ wrapper class around the Perl script. This wrapper class contains an RMComms TCPCommServer, making the data collected by the SNMP calls available to the rest of the Resource Management components.

FG16 - Remos Network Data Broker

The final functional component of the Host and Network Monitoring functional group is the Remos Network Data Broker FG16 which receives information on network link bandwidth and network link bandwidth utilization from the SNMP-based Remos network

monitoring tool, as shown in FigFIGS. 22A, 2B and/or FigIG. H4l4.~ The network information is accessed via the Remos API library and is then sent on to the Host Load Analyzer (Hardware Broker) function FG40 of the Resource Allocation Decision-Making functional ~~groupFG4~~group FG4 using an RMComms TCPCommServer.~ Remos works by using SNMP to query the switches (via the bridge collector) to collect information on network configuration as well as bandwidth utilization on each link and also issues SNMP MIB-II queries to each host to collect the host's view of network utilization.~ The network information received from Remos consists of the maximum potential bandwidth and the current bandwidth utilization on specific host network links.

The Remos Broker FG16 provides the following information about the network link for each host. ~The data is sent to the Host Load Analyzer (Hardware Broker) approximately every 2 seconds. The Remos Broker FG16 uses configuration files listing specific hosts and switches that should be queried.

The functions implemented by Host Monitor functional group FG1 have been designed to provide a system monitoring capability not normally supplied by standard SVR4 or BSD Unix services. Such services include cross-platform reporting of system process loading~, CPU performance, network performance and periodic status summary reporting. The Host Monitors were developed to support efforts by the Hiper-D Resource Management group, attempting to provide a common set of OS level parameters useful for assessing host and network load and status, for supporting Rresource allocation/reallocation algorithms, and attempting to provide a minimally intrusive, close to real-time capability for gathering this data.

Host Discovery Design

The Host Discovery function FG14 of the Resource Management architecture

provides Rresource discovery of hosts on a network.~ It identifies new hosts that come online or previously known hosts that have gone offline.~ The Host Discovery component can determine the hostname, the operating system name and version, and in some cases the machine architecture and manufacturer of a newly discovered host.~ This information is sent to Program Control so the new host can be added to the pool of resources.

The Host Discovery functional element FG14 consists of a Perl script that contains the Rresource discovery functionality, and a C++ object that receives the output of the Perl script and provides this information to Program Control via an RMComms TCPCommServer connection. ~This is described in CD-Appendix H.~ More specifically, the Perl script *host_discovery.pl* issues ICMP (ping) calls and MIB-II SNMP queries to discover new hosts. On initialization, the script populates a data structure called *Net_info* for each of the networks (subnets) it needs to monitor.~ Currently this information is hard-coded, the subnet is defined as 172.30.1, and the lower and upper limits for the host are 1 and 254 respectively. It then initializes the global variables for the server host and port, network domain, and the executable path for the ping (fping) command

The *host_discovery.pl* script establishes a baseline of existing hosts using the current set of hosts that answer the fping call. ~For each network/subnet defined in its list of *Net_info* (Net_info.pm) data structures, it calls fping and builds a list of IP addresses of hosts that answered the ping, known as reachable hosts, and a list for those hosts that did not answer the ping. ~For each reachable host, a *Host_info* (Host_info.pm) data structure is populated to store the host's information.~ (Key fields in the *Host_info* data structure include IP address, hostname, operating system and version, architecture class, and manufacturer.) ~Since the IP address of the reachable host is known, a call to gethostbyaddr() is used to get the hostname. Other information for the host is obtained by making a MIB-II (Management Interface Base version 2) system Group (Object ID 1.3.6.1.2.1.1.1.0) SNMP call to the SNMP agent on each reachable host.~ This SNMP query returns information on the configuration of a specific

network device (in this case, the configuration of each reachable host).

The *host_discovery.pl* script makes SNMP calls by using subroutines freely available for public use (freeware), created by Simon Leinen. ~These subroutines are contained in the files BER.pm and SNMP_Session.pm.~ The *SNMP_Session* is configurable for specifying timeouts and number of retries before declaring a host unavailable, and for specifying the SNMP Object Id (OID).

Additional general and specific details regarding functional elements of the Host and Networking functional group FG1 are provided in CD-Appendix H.

FG2: Instrumentation functional group

As mentioned above, the NSWC-DD Instrumentation System provides general-purpose application event reporting and event correlation capabilities. The Instrumentation system forms an architecture that allows instrumented application data to be easily accessible by other components of the Resource Management architecture. The major functional components of the Instrumentation System architecture are the following:

1) The **Instrumentation API Libraries,** which are linked with the applications and provide the function call interfaces by which the application sends instrumentation data.
2) An **Instrumentation Daemon**, one copy of which resides on each host in the distributed environment and is responsible for reading instrumentation data sent by the applications, reformatting the data into instrumentation event messages and sending the messages to the Instrumentation Collectors.
3) The **Instrumentation Collectors**, which connect to the Instrumentation Daemons on each host and receive instrumentation messages from all hosts. The Collectors forward received messages to the Instrumentation Correlators

and Instrumentation Brokers.

4) The **Instrumentation Correlators**, which receive instrumentation messages from the Instrumentation Collectors and provide grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other Resource Management components.
5) The **Instrumentation Brokers**, which receive instrumentation messages from the Instrumentation Collectors and perform task-specific reformatting and data manipulation for driving displays or other Resource Management components.
~~(6)~~ 6) The **Jewel Instrumentation Broker (QoS Monitor)**, which is a legacy component that can receive instrumentation data from either the open source Jewel instrumentation package or from the Instrumentation Collectors. The QoS Monitor performs task-specific message reformatting and data manipulation for driving displays and the QoS Managers.

Instrumentation API Library

The applications link in the Instrumentation API Library and make API call to construct and send out instrumentation event messages. Three separate APIs are provided for use by the applications: 1) a printf()-style API which allows the code to format, build, and send instrumentation data with a single function call, 2) a buffer-construction-style API where the multiple function calls are made to construct the instrumentation buffer iteratively, one data element per call, and 3) a Jewel function call API based on the existing API provided by the Jewel instrumentation package (an open-source package produced by the German National Research Center for Computer Science). The first two APIs are the

preferred programming interfaces and take advantage of several key new instrumentation features. It will be appreciated that the Jewel API is provided solely for backwards compatibility with existing instrumented application code and is implemented as a set of wrappers around the printf()-style API. All three APIs are supported for C and C++. Ada bindings have been produced for the buffer-construction-style API and the Jewel function call API.

The instrumented data is sent from the application to the Instrumentation Daemon on the same host. The current mechanism for data transfer is via UNIX FIFO IPC (inter-process communication) mechanisms. The FIFO mechanism was chosen based on reliability, low overhead, and ease of implementation. Future implementations of the Instrumentation system may explore alternate data passing mechanisms including shared message queues.

Instrumentation Daemon

An Instrumentation Daemon resides on each host in the distributed environment. The Instrumentation Daemon is interrupted when new data is written to the FIFO. The Instrumentation Daemon reads the data from the FIFO and reformats the data into the standard internal Instrumentation message format and sends the data to each of the Instrumentation Collectors that are currently active. (For future implementations, an event request filtering mechanism will be implemented so that specific event messages will only be sent to those Instrumentation Collectors that have requested the message.)

Instrumentation Collectors

The Instrumentation Collectors receive instrumentation messages from the Instrumentation Daemons on each host in the distributed environment. Currently, the Instrumentation Collectors send every instrumentation message to all Instrumentation

Brokers and Instrumentation Correlators that have connected to the Instrumentation Collector. (For future implementations, an event request filtering mechanism will be implemented so that specific event messages will only be sent to those Instrumentation Brokers and Instrumentation Correlators that have requested the message. For now, the Instrumentation Collector serves as a pass-through server for instrumentation messages. The Instrumentation Collector does supports architecture scalability in the sense that without the Instrumentation Collectors, each Instrumentation Broker and Instrumentation Correlators would need to maintain connections to the Instrumentation Daemons on every host.)

Instrumentation Correlators

The Instrumentation Correlators provide grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other Resource Management components. Each Correlator reads in a user-specified correlation grammar file that is interpreted at run-time by the Correlator's instrumentation correlation engine.

Instrumentation Brokers

The Instrumentation Brokers are task-specific applications built around a common code package. The Instrumentation Brokers receive instrumentation messages from the Instrumentation Collectors, filter all received instrumentation messages to find the messages of interest, and perform task-specific message data reformatting and manipulation for driving other components such as displays or other Resource Management components. The Instrumentation Broker approach allows for instrumentation data sources to be quickly integrated for test, display, and debugging purposes. (As the Instrumentation Correlator grammar and correlation engine mature in future releases, it is anticipated that the Instrumentation Broker approach will be used less frequently.)

Jewel Instrumentation Broker (QoS Monitor)

The Jewel Instrumentation Broker (hereafter referred to the QoS Monitor) is a legacy architecture component that served as a broker between the Jewel instrumentation package components and Resource Management components and displays. The QoS Monitor was responsible for polling the Jewel Collector components to retrieve application event messages. These messages were then reformatted and used to drive several displays and the QoS Managers. The Jewel instrumentation package has now been replaced in all applications, however the message reformatting capabilities of the QoS Monitor have been maintained so that several displays and the existing QoS Manager interface do not have to be upgraded immediately. The QoS Monitor component has been modified so that it receives instrumentation data from both Jewel and the Instrumentation Collectors.

Middleware

The RMComms middleware package, which is described in the RMComms Middleware Design Report, provides the internal message passing interfaces between the Resource Management components connected via the network. The middleware provides for automatic location-transparent many-to-many client-server connections. Low overhead, reliable message passing capabilities are provided. Registration of message handler callback functions for specified requested message types are provided with the message handler functions being invoked when messages arrive. Registration of connection status callback functions which are invoked when either new connections are made or existing connections are broken is also provided. The middleware package also allows for multiple client and server objects to be instantiated in the same application, is thread-safe, and provides an easy-to-use object-oriented API through which all capabilities are accessed.

Additional details regarding the Instrumentation functional group FG2 are provided

in CD-Appendix I.

FG42: Resource Manager

The Resource Manager 42 is the primary decision-making component of the Resource Management toolkit. It is responsible for: 1) responding to application and host failures by determining if and what recovery actions should be taken, 2) determining if and where to place new copies of scalable applications or which scalable applications should be shutdown when the QoS Managers FG44A-FG44N indicate that scale-up or scale-down actions should be taken based on measured application performance, 3) determining where new applications should be placed when requested to do so by Program Control, and 4) determining which and how many applications should run based on application system (mission) priorities. In order to accomplish these tasks, the Resource Manager 42 maintains a global view of the state of the entire distributed environment including status information on all hosts, networks, and applications. In addition, the Resource Manager 42 also calculates software and hardware readiness metrics and reports these readiness values for display purposes. ~~Figure 1~~FIGS. 1A, 1B shows the connectivity and high-level data flow between the Resource Manager 42 and the other Resource Management-related components.

The Resource Manager 42 receives status and failure information about hosts, networks, and applications from Program Control. This information includes periodic status updates as well as immediate updates when statuses change such as a new host being detected or an application failing. In the case of applications going down, information as to whether the applications were shutdown on purpose or whether they failed is also sent. Program Control also issues requests to the Resource Manager 42 when new applications need to be dynamically allocated and when Program Control determines that the Resource Manager 42 needs to assess and attempt to resolve inter-application dependencies (such as an application which needs to be running prior to starting up another application).

The Resource Manager 42 responds to faulted applications and hosts by determining whether the failed applications can and should be restarted and attempting to determine where (and if) there are hosts available that the application can run on. When a decision is made by the Resource Manager 42, a message is sent to Program Control specifying what application to start and where to put it. The same general mechanism is used when Program Control requests that the Resource Manager 42 determine where to start new applications and/or how to resolve inter-application dependencies; the Resource Manager 42 responds with orders indicating what applications to start and where to start them. The Resource Manager 42 also sends application shutdown orders to Program Control requesting that certain application be stopped; this can occur when the QoS Managers FG44A-FG44N indicate that certain scalable applications have too many copies running or when application system priority changes (to lower priorities) occur resulting in scaling back the application system configuration.

The Resource Manager 42 receives host load and host fitness information on all known hosts from the Hardware Broker 40 (Host Load Analyzer). This information include overall host fitness scores, ~~cpu-based~~CPU-based fitness scores, network-based fitness scores, and memory and paging-based fitness scores along with the SPEC95 ratings of the hosts. This information is received approximately once a second and includes information on all known hosts in the distributed system. These scores are used by the Resource Manager 42 for determining the "best" hosts for placing new applications when: 1) responding to requests from the QoS Managers FG44A-FG44N to scale up additional copies of an application, 2) attempting to restart failed applications, 3) responding to requests to dynamically allocate certain applications, and 4) responding to application system (mission) priority changes which require scaling up additional applications.

The Resource Manager 42 receives requests from the QoS Managers FG44A-FG44N for scaling up, moving, or scaling down specific applications. The Resource Manager FG42

responds to these requests by determining whether the request should be acted upon and, if so, determines the specific action to take and issues orders to Program Control to start up or shutdown specific applications on specific hosts. The QoS Managers FG44A-FG44N are responsible for monitoring specific system performance metrics (e.g., quality of service, or QoS, requirements) via instrumentation and determining if performance can be improved by scaling up or moving certain applications. When this occurs, the QoS Managers FG44A-FG44N send a request to the Resource Manager FG42 indicating that a new copy of a specific application should be started. If the QoS Managers FG44A-FG44N determine that the performance of a scalable application can be improved by moving an application, a scale up request is first sent to the Resource Manager FG42 and when the new application has been started, a scaledown request is then sent to the Resource Manager. Also, when the QoS Managers FG44A-FG44N determine that there are more copies of scalable application running then are needed, requests to shutdown specific applications are sent to the Resource Manager FG42. The division of responsibility is that the QoS Managers FG44A-FG44N determine what actions would potentially improve performance, but the Resource Manager FG42 has final authority to determine whether to implement the requested actions.

When the Resource Manager FG42 is first started, it reads in the System Specification Files (via System Specification Library, SSL, calls) which contain the list of hosts that are known to be in the distributed environment and information on all applications that can be run in the distributed environment. The System Specification Files also include application-level information including where specific applications can be run, which applications are scalable, which applications can be restarted, and any dependencies between applications.

The Resource Manager FG42 can also receive updated application survivability specifications from the QoS Specification Control component. This information overrides the application survivability information that was initially loaded in from the System

Specification Files for specified applications. The information is used by the Resource Manager FG42 to determine whether the specific applications will be restarted if they fail at run-time.

The Resource Manager FG42 sends application system and hardware system readiness and system (mission) priority information to the Readiness Broker and to the Globus Broker. The Readiness Broker is responsible for driving a GUI/display which shows the current readiness data and allows the system (mission) priorities to be changed and sent back to the Resource Manager FG42. The Globus Broker provides basically the same functionality except that only a high-level subset of the readiness data provided to the Readiness Broker is provided to the Globus Broker. The readiness information sent to the Readiness Broker consists of readiness values for each application, application subsystem, and application system defined in the System Specification Files. The readiness scores are currently based on the status (up/down) of the applications within a system or subsystem along with the percentage of potential copies of scalable applications that are currently running. Host and network readiness scores are also calculated and are the scores are determined based on the host load information and host fitness scores received from the Hardware Broker 40.

The Resource Manager FG42 also sends information about allocation and reallocation decisions to the Resource Management Decision Review Display. Information on the decision that was made, what event the decision was in response to, and how long it took to both make the decision and implement the decision are sent to the display. In addition, information about the top choices for where an application could have potentially been placed is also sent (if applicable); this information includes the host fitness scores for the selected host and the next best host choices which could have been selected.

See CD-Appendix M for additional details regarding Resource Manager FG42.

In the Background Section of the application, the reader may have interpreted the sentence "The present invention relates generally to resource management systems by which networked computers cooperate in performing at least one task too complex for a single computer to perform" to indicate that the Resource Management Architecture is limited to such applications. Thus, while the Resource Management Architecture generally supports tasks distributed across multiple hosts, it is not limited to only those tasks that must be distributed due to the inability to run them on a single machine. Moreover, the Resource Management functional elements advantageously could be used to control a set of applications which all run on the same machine while still providing monitoring, fault tolerance, etc. (albeit that this is not the normal or even the intended configuration). Furthermore, the Resource Management Architecture, as discussed above, deals with resource managed applications, where the managed characteristic may be one of scalability, survivability, fault tolerance or priority.

FrgIG. 15 is a block diagram of a CPU-based system 400, corresponding to one or more of the hosts A-N. The system 400 includes a central processing unit (CPU) 402, e.g., a microprocessor, that communicates with the RAM 412 and an I/O device 408 over a bus 420. It must be noted that the bus 420 may be a series of buses and bridges commonly used in a processor-based system, but for convenience purposes only, the bus 420 has been illustrated as a single bus. A second I/O device 410 is provided in an exemplary case. The processor-based system 400 also includes a primary memory 412, an additional memory 414, which could be either a read-only memory (ROM) or another memory device, e.g., a hard drive or the like. The CPU- based system may include peripheral devices such as a floppy disk drive 404, a compact disk (CD) ROM drive 406, a display (not shown), a key board (not shown), and a mouse (also not shown), that communicate with the CPU 402 over the bus 420 as is well known in the art. It will be appreciated that the either one of the memories 412 or 414 advantageously can be employed to store computer readable instructions for converting the general purpose system 400 into one of the host A-N. It will also be

appreciated that the nature of the distributed environment permits the necessary application and API's needed to implement the Resource Management Architecture to9 be stored anywhere on the network.

provided on the CD-ROM filed concurrently with the application. In addition, the CD-ROM also includes the source code listing for the Resource Management Architecture according to the present invention.

Although presently preferred embodiments of the present invention have been described in detail herein, it should be clearly understood that many variations and/or modifications of the basic inventive concepts herein taught, which may appear to those skilled in the pertinent art, will still fall within the spirit and scope of the present invention, as defined in the appended claims.

Table III

APPENDIX	TYPE	DESCRIPTION
A	Attached	Resource Management Architecture Function List
B	Attached	Standard Instrumentation Message Format
C	Attached	API Listing for RMComms
D	CD	Resource Manager Interface Messages
E	CD	Host Load Analyzer (Hardware Broker) Function
F	CD	Quality-of-service (QoS) Manager Function
G	CD	FgG3:- System Specification Language & System Specification Library (SSL) Functions
H	CD	Host And Network Monitoring Functional Group
I	CD	Instrumentation Functional Group
J	CD	Display Functional Group
K	CD	RmcMComms Network Communication Middleware Design
L	CD	System Readiness Display
M	CD	Resource Manager Rm42FG42
N	CD	Instrumentation Graph Tool
O	CD	Host Discovery Function
P	CD	Instrumentation Application Programming Interface (API)
Q	CD	Program Control Application Controller
R	CD	Program Control Display
S	CD	Program Control Functional Group
T	CD	QoS Manager
U	CD	Resource Allocation Decision-making Functional Group

Table III provides a listing of the Appendicies included for all purposes in the application. It will be noted that the majority of the listed ~~Apendicies~~Appendices are

Response To Notice To File Missing Parts Of Application Filing Date Granted (PTO-1533)(Large Entity)	Docket No. NC 83017

In Re Application Of: **Paul V. Werme et al.**

OIPE JC139 SEP 19 2001 PATENT & TRADEMARK OFFICE

Serial No.	Filing Date	Examiner	Group Art Unit
09/864,825	05/24/2001	Unknown	2171

Invention: **RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT AND CORRESPONDING PROGRAMS THEREFOR**

TO THE ASSISTANT COMMISSIONER FOR PATENTS:

Box Missing Parts

This is a response to the Notice to File Missing Parts of Application - Filing Date Granted (PTO-1533) mailed on **07/23/2001** (*Date*).

Enclosed herewith for filing are the following:

☒ A copy of the Notice to File Missing Parts of Application - Filing Date Granted (PTO-1533). **(REQUIRED)**

☐ An oath or declaration in compliance with 37 CFR 1.63, including residence information and identifying the application by the above Application Number and Filing Date.

☒ A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date.

☐ An oath or declaration in compliance with 37 CFR 1.63 listing the names of all inventors and signed by the omitted inventor(s), identifying this application by the above Application Number and Filing Date.

☐ A verified English translation of the non-English language application papers as originally filed. It is requested that this translation be used as the copy for examination purposes in the United States Patent and Trademark Office.

☒ Other (list):

Preliminary Amendment (with Attached Substitute Specification and Red-lined Version of Specification)
Submission of Formal Drawing

Response To Notice To File Missing Parts Of Application Filing Date Granted (PTO-1533)(Large Entity)	Docket No. NC 83017

In Re Application Of: **Paul V. Werme et al.**

Serial No.	Filing Date	Examiner	Group Art Unit
09/864,825	**05/24/2001**	**Unknown**	**2171**

Invention: **RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT AND CORRESPONDING PROGRAMS THEREFOR**

TO THE ASSISTANT COMMISSIONER FOR PATENTS:

Box Missing Parts

☒ Completion of application fees as calculated below:

☐ Utility application filing fee	
☐ Design application filing fee	
☒ Total number of independent claims = 5	**$160.00**
☒ Total number of claims = 36	**$288.00**
☐ Multiple dependent claims	
☒ Surcharge for late payment of filing fee and/or late filing of original declaration or oath	**$130.00**
☐ Petition and fee for filing by other than all the inventors or a person not the inventor	
☐ Fee for processing an application filed with a non-English language specification	
☐ Fee for processing and retention of application	
Total completion of application fees	**$578.00**

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a response to the above-identified Notice to File Missing Parts of Application. The requested extension is as follows (check time period desired). If an additional time extension is required, please consider this a petition therefor.

☐ One month ☐ Two months ☐ Three months ☐ Four months ☐ Five months

from: _____ *Date* until: _____ *Date*

Total time extension fees	
Total fees due	**$578.00**

P31LARGE/REV02

Express Mail Label No. EL 523 415 956 US

Response To Notice To File Missing Parts Of Application Filing Date Granted (PTO-1533) (Large Entity)	Docket No. NC 83017

In Re Application Of: **Paul V. Werme et al.**

Serial No.	Filing Date	Examiner	Group Art Unit
09/864,825	05/24/2001	Unknown	2171

Invention: **RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT AND CORRESPONDING PROGRAMS THEREFOR**

TO THE ASSISTANT COMMISSIONER FOR PATENTS:

Box Missing Parts

The fee of **$578.00** is to be paid as follows:

- ☐ A check in the amount of the fee is enclosed.
- ☒ The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account No. **50-0967**
  A duplicate copy of this sheet is enclosed.
- ☐ If an additional extension of time is required, please consider this a petition therefor and charge any additional fees which may be required to Deposit Account No.
  A duplicate copy of this sheet is enclosed.

James B Bechtel
Signature

Dated: Sept 19, 2001

James B. Bechtel, Registration No. 29,890

I certify that this document and fee is being ~~deposited on~~ hand-carried ~~with the U.S. Postal Service as first class mail under 37 C.F.R. 1.8 and is addressed~~ to the Assistant Commissioner for Patents, Washington, D.C. 20231. on Sept 19, 2001.

James B Bechtel
Signature of Person Mailing Correspondence

James B Bechtel
Typed or Printed Name of Person Mailing Correspondence

cc:

P31LARGE/REV02

TRANSMITTAL OF FORMAL DRAWINGS	Docket No. NC 83017

OIPE
SEP 19 2001
PATENT & TRADEMARK OFFICE

In Re Application Of: **Paul V. Werme et al.**

Serial No.	Filing Date	Batch No.	Examiner	Art Unit
09/864,825	05/24/2001	NA	Unknown	2171

Invention: **SYSTEM SPECIFICATION LANGUAGE FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR**

Address to:
Assistant Commissioner for Patents
Washington, D.C. 20231

Transmitted herewith are:

27 sheets of formal drawing(s) for this application.

Each sheet of drawing indicates the identifying indicia suggested in 37 CFR Section 1.84(c) on the reverse side of the drawing.

Client/Matter: 83017.0005

James B Bechtel
Signature
James B. Bechtel, Registration No. 29,890

Dated: Sept 19, 2001

I certify that this document and attached formal drawings are being ~~deposited on~~ hand carried ~~with the U.S. Postal Service as first class mail under 37 C.F.R. 1.8 and addressed~~ to the Assistant Commissioner for Patents, Washington, D.C. 20231. on Sept 19, 2001.

James B Bechtel
Signature of Person Mailing Correspondence

James B Bechtel
Typed or Printed Name of Person Mailing Correspondence

P23B/REV01

OIPE JC139
SEP 19 2001
PATENT & TRADEMARK OFFICE

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Paul V. Werme et. al.

Serial No: 09/864,825 — Group Art Unit: 2171

Filed: May 24, 2001 — Examiner: Unknown

For: SYSTEM SPECIFICATION LANGUAGE FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR — Batch No. NA

Docket No: 83017

Box Issue Fee
ATTN: Box Missing Parts
Assistant Commissioner for Patents
Washington, DC 20231

TRANSMITTAL OF ADDITIONAL INFORMATION

Sir:

Enclosed herewith are copies of the following:

1. Transmittal of 27 sheets of formal drawings.
2. Response to Notice to file Missing Parts of Application.
3. Preliminary Amendment under 37 C.F.R. § 1.111.
4. Substitute Specification with Machine Generated Red-Line Version.

The Commissioner is hereby authorized to charge fees in the amount of $578.00 to Deposit Account No. 50-0967. The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account No. 50-0967.

Respectfully submitted,

James B. Bechtel

James B. Bechtel, Esq.
Reg. No. 29,890
Customer No. 23501

Naval Surface Warfare Center
Office of Counsel (Patents)(Code CD222)
17320 Dahlgren Road
Dahlgren, VA 22448-5100
Tel: 540-653-8061Fax: 540-653-7816

CERTIFICATE OF MAILING VIA EXPRESS MAIL

I hereby certify that this paper (along with any referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as to be sent by Express Mail in an envelope addressed to the Assistant Commissioner for Patents, Washington, DC 20231.

Date: _____ _____

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

APPLICATION NUMBER	FILING/RECEIPT DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NUMBER
09/864,825	05/24/2001	Paul V. Sjerme	NAVY CASE # 83017

OIPE JC139
SEP 19 2001
PATENT & TRADEMARK OFFICE

CONFIRMATION NO. 2372

23501
JAMES B BECHTEL, ESQ
NAVAL SURFACE WARFARE CENTER CODE CD222
17320 DAHLGREN ROAD, BLDG. 183, RM. 015
DAHLGREN, VA 224485100

FORMALITIES LETTER

OC000000006330630

Date Mailed 07/23/2001

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

09/21/2001 MWOLDER1 00000033 500967 09864825

01 FC:102 160.00 CH
02 FC:103 288.00 CH
03 FC:105 130.00 CH

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items and pay any fees required below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The oath or declaration is unsigned.
- To avoid abandonment, a late filing fee or oath or declaration surcharge as set forth in 37 CFR 1.16(e) of $130 for a non-small entity, must be submitted with the missing items identified in this letter.
- **The balance due by applicant is $ 130.**

The application is informal since it does not comply with the regulations for the reason(s) indicated below.

The required item(s) identified below must be timely submitted to avoid abandonment:

- Substitute drawings in compliance with 37 CFR 1.84 because:
  - drawing sheets do not have the appropriate margin(s) (see 37 CFR 1.84(g)) Each sheet must include a top margin of at least 2.5 cm. (1 inch), a left side margin of at least 2.5 cm. (1 inch), a right side margin of at least 1.5 cm. (5/8 inch), and a bottom margin of at least 1.0 cm. (3/8 inch);

11/26/2001 SSALEEKU 00000002 500967 09864825

01 FC:103 18.00 CH

A copy of this notice MUST be returned with the reply.

Customer Service Center

Initial Patent Examination Division (703) 308-1202
PART 2 - COPY TO BE RETURNED WITH RESPONSE

PTO/SB/21 (09-04)



PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of	: Atty. Docket No.: NC 83017
Paul V. WERME et al.	: Group Art Unit: 2171
Serial No.: 09/864,825	: Examiner: Unknown
Filed: May 24, 2001	:
For: **RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR**	:

Express Mail Label No. EL 695 525 954 US

PRELIMINARY AMENDMENT UNDER 37 C.F.R. §1.111

Honorable Assistant Commissioner
for Patents
Washington, D.C. 20231

Sir:

Prior to examination of the above-identified application on the merits, please amend the above-identified application as follows:

IN THE SPECIFICATION:

Attached hereto is a substitute specification and a machine generated red-line version of the specification showing the changes between the as-filed specification and the substitute specification. It is respectfully submitted that the substitute specification does not introduce prohibited new matter into the specification.

IN THE CLAIMS:

Please ADD new claims 2 - 36 as follows:

--2. On a host instantiating a managed characteristic application, a resource management device generating signals which one of starts up an additional copy of the managed characteristic application on one of the host and a second networked host, shuts down and restarts the managed characteristic application on the host, and moves the managed characteristic application to the second host responsive to first information regarding performance of all applications including the managed characteristic application and second information regarding performance of the host.--

--3. The resource management device as recited in claim 2, wherein the managed characteristic application comprises a scalable application.--

--4. The resource management device as recited in claim 2, wherein the managed characteristic application comprises a fault tolerant application, where the degree of fault tolerance is selectable by a user.--

--5. The resource management device as recited in claim 2, wherein the managed characteristic application comprises a selectable priority application.--

--6. The resource management device as recited in claim 2, wherein the managed characteristic application further responds to user-initiated control actions.--

--7. The program control device as recited in claim 2, wherein the resource management device generates signals instructing a program control device to modifies the configuration of the managed characteristic application.--

--8. In a grid system comprising networked hosts instantiating M managed characteristic applications, a resource management device generating signals which one of control configuration of an Mth copy of the M managed characteristic application, start up of an $(M+1)^{th}$ copy of the M managed characteristic applications, shutdown and restart of the Mth copy of the M managed characteristic applications, and movement of the Mth copy of the M managed characteristic applications from a first to a second host responsive to first information regarding performance of the M managed characteristic applications and second information regarding performance of the hosts, where M is a positive integer.--

--9. The resource management device as recited in claim 8, wherein one of the M managed characteristic applications comprises a scalable application.--

--10. The resource management device as recited in claim 8, wherein one of the M managed characteristic applications comprise fault tolerant application, where the degree of fault tolerance is selectable by a user.--

--11. The resource management device as recited in claim 8, wherein one of the M managed characteristic applications comprises a selectable priority application.--

--12. The resource management device as recited in claim 8, wherein the resource management device further responds to third information regarding the performance of hardware operatively coupling the networked hosts.--

--13. The resource management device as recited in claim 8, wherein the resource management device further responds to fourth information establishing one of the priority assigned to the managed characteristic application and multiple copies of the managed characteristic application.--

--14. The resource management device as recited in claim 8, wherein the resource management device further responds to externally generated action requests.--

--15. The resource management device as recited in claim 14, where the action requests are generated by an operator.--

--16. In a distributed environment comprised of N hosts operating in a distributed environment instantiating M managed characteristic applications, resource allocation control software instantiated by at least the N hosts, the software comprising:

a first function which determines the state and health of the N hosts, a network operatively coupling the N hosts to one another and the M managed characteristic applications in the distributed environment;

a second function which determines required allocation and reallocation actions need to maintain Quality of Service (QoS) requirements established for the M managed characteristic applications; and

a third function which generates automatic control signal requests corresponding to the actions dictated by the QoS requirements.--

--17. The software as recited in claim 16, wherein the first function receives system specification information comprising selected ones of host configuration and capabilities, application capabilities, survivability requirements, scalability characteristics, application startup and shutdown dependencies, and application and path performance requirements.--

--18. The software as recited in claim 16, wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic applications, and detected failures regarding the N hosts.--

Ex.1002 / Page 367 of 541
TESLA, INC.

--18. The software as recited in claim 16, wherein the first function receives historical data regarding statuses, configuration, and loads of the N hosts and link statuses and loads regarding the network.--

--19. The software as recited in claim 16, wherein the first function receives application performance data representing each one of the M managed characteristic applications.--

--20. The software as recited in claim 16, wherein the first function receives application performance data on all applications instantiated by the N hosts including performance data representing each one of the M managed characteristic applications.--

--21. The software as recited in claim 16, wherein the second function which determines the required allocation and reallocation actions need to maintain the Quality of Service (QoS) requirements established for the M managed characteristic applications by:

responding to application and host failures by determining if and what recovery actions should be taken;

determining if and where to place new copies of one of the M managed characteristic applications or which of the M managed characteristic applications should be shutdown when QoS

Manager functions indicate that scale-up or scale-down actions are indicated based on measured application performance and application QoS specifications;

determining where new applications should be placed when requested to do so by a program control device; and

determining which and how many applications should run based on application system priorities.--

--22. The software as recited in claim 16, wherein one of the M managed characteristic applications comprises a scalable application.--

--23. The software as recited in claim 16, wherein one of the M managed characteristic applications comprises a fault tolerant application, where the degree of fault tolerance is selectable by a user.--

--24. The software as recited in claim 16, wherein one of the M managed characteristic applications comprises a selectable priority application.--

--25. The software as recited in claim 16, wherein the M managed characteristic applications comprise M copies of a single managed characteristic application.--

Ex.1002 / Page 369 of 541
TESLA, INC.

--26. Software stored on at least one host for converting N networked hosts into a resource managed system instantiating M managed characteristic applications, the software comprising:

a first function group which monitors the host and network resources

a second function group which provides general-purpose application event reporting and event correlation capabilities;

a third function group which provides the reasoning and decision-making capabilities for the resource managed system, wherein the third function group comprises:

a first function which determines the state and health of the N hosts, a network operatively coupling the N hosts to one another and the M managed characteristic applications in the distributed environment;

a second function which determines required allocation and reallocation actions need to maintain Quality of Service (QoS) requirements established for the M managed characteristic applications; and

a third function which generates automatic control signal requests corresponding to the actions dictated by the QoS requirements; and

a fourth function group which provides program control capabilities permitting starting, stopping, and configuring of selected ones of the M managed characteristic applications on respective ones of the N hosts in the resource managed system,

where M and N are positive integers.--

--27. The software as recited in claim 26, wherein the first function receives system specification information comprises host configuration and capabilities.--

--28. The software as recited in claim 26, wherein the first function receives system specification information comprising selected ones of capabilities, survivability requirements, scalability characteristics, startup and shutdown dependencies, and performance requirements for at least one of the M managed characteristic applications.--

--29. The software as recited in claim 26, wherein the first function receives system specification information comprising path performance requirements regarding communication between at least two of the N hosts.--

--30. The software as recited in claim 26, wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic applications, and detected failures regarding the N hosts.--

--31. The software as recited in claim 26, wherein the first function receives historical data regarding statuses, configuration, and loads of the N hosts and link statuses and loads regarding the network.--

--32. The software as recited in claim 26, wherein the first function receives application performance data representing each one of the M managed characteristic applications.--

--33. The software as recited in claim 26, wherein the first function receives application performance data on all applications instantiated by the N hosts including performance data representing each one of the N copies of the managed characteristic application.--

--34. The software as recited in claim 26, wherein the second function which determines the required allocation and reallocation actions established for the M managed characteristic application by:

responding to application and host failures by determining if and what recovery actions should be taken;

determining if and where to place new copies of managed characteristic applications or which managed characteristic applications should be shutdown when QoS Manager functions indicate that scale-up or scale-down actions should be taken based on measured application performance and QoS specifications established for the M managed characteristic applications;

determining where new applications should be placed when requested to do so by the fourth function group; and

determining which and how many applications should run based on application system priorities.--

Ex.1002 / Page 372 of 541
TESLA, INC.

--35. The software as recited in claim 26, wherein the third function group makes decisions by one of:

based on requests from Program Control, determining where new applications should be started;

based on indication of application failure from the fourth function group, determining whether and where the failed application should be restarted;

based on indication of host failure from the fourth function group, determining whether and where the failed application previously instantiated by the failed one of the N hosts should be restarted;

based on startup and shutdown dependency resolution requests from the fourth function group, determine whether and where additional applications should to be one of started and shut down prior to starting or shutting down another application; and

based on changes to application system priorities, determining whether and where new applications need to be started and/or determine whether and which existing applications need to be shutdown.--

--36. The software as recited in claim 26, wherein the third function group makes decisions one of:

based on application inter-dependencies defined in system specification files, determining whether and where additional applications should to be one of started and shut down prior to starting or shutting down of another application;

based on application instrumentation data generated by the second function group and performance requirements defined in the system specification files, determining whether applications are meeting performance requirements and whether an application can be scaled up or moved to attempt to improve performance; and

based on the application instrumentation data and performance requirements defined in the system specification files, determining whether applications are performing well within performance requirements and can be scaled down.--

REMARKS

Claims 1-36 are pending in the Application. In the Preliminary Amendment, new claims 2-36 are added to recite features of the present invention that were previously disclosed but unclaimed in the application as originally filed. The attached Appendix contains a version of the claims without markings.

It is respectfully submitted that the instant Amendment does not introduce new matter into the application. It is also respectfully submitted that the Preliminary Amendment places the above-identified application in even better condition for initial examination.

In light of the amendments and remarks presented above, it is respectfully submitted that the application is in condition for allowance, and such action is hereby solicited.

If any points remain in issue which the Examiner feels may best be resolved through a personal or telephone interview, the Examiner is kindly requested to contact the undersigned at the telephone number listed below.

Respectfully submitted,

James B. Bechtel
Reg. No. 29,890
Phone: (540) 653-8061

Date: September __, 2001

Atty. Docket No.: NC 83017

Express Mail Label No. EV 045 525 345 US

APPENDIX

2. On a host instantiating a managed characteristic application, a resource management device generating signals which one of starts up an additional copy of the managed characteristic application on one of the host and a second networked host, shuts down and restarts the managed characteristic application on the host, and moves the managed characteristic application to the second host responsive to first information regarding performance of all applications including the managed characteristic application and second information regarding performance of the host.

3. The resource management device as recited in claim 2, wherein the managed characteristic application comprises a scalable application.

4. The resource management device as recited in claim 2, wherein the managed characteristic application comprises a fault tolerant application, where the degree of fault tolerance is selectable by a user.

5. The resource management device as recited in claim 2, wherein the managed characteristic application comprises a selectable priority application.

6. The resource management device as recited in claim 2, wherein the managed characteristic application further responds to user-initiated control actions.

7. The program control device as recited in claim 2, wherein the resource management device generates signals instructing a program control device to modifies the configuration of the managed characteristic application.

Ex.1002 / Page 376 of 541
TESLA, INC.

8. In a grid system comprising networked hosts instantiating M managed characteristic applications, a resource management device generating signals which one of control configuration of an Mth copy of the M managed characteristic application, start up of an $(M+1)^{th}$ copy of the M managed characteristic applications, shutdown and restart of the Mth copy of the M managed characteristic applications, and movement of the Mth copy of the M managed characteristic applications from a first to a second host responsive to first information regarding performance of the M managed characteristic applications and second information regarding performance of the hosts, where M is a positive integer.

9. The resource management device as recited in claim 8, wherein one of the M managed characteristic applications comprises a scalable application.

10. The resource management device as recited in claim 8, wherein one of the M managed characteristic applications comprise fault tolerant application, where the degree of fault tolerance is selectable by a user.

11. The resource management device as recited in claim 8, wherein one of the M managed characteristic applications comprises a selectable priority application.

12. The resource management device as recited in claim 8, wherein the resource management device further responds to third information regarding the performance of hardware operatively coupling the networked hosts.

13. The resource management device as recited in claim 8, wherein the resource management device further responds to fourth information establishing one of the priority assigned to the managed characteristic application and multiple copies of the managed characteristic application.

Ex.1002 / Page 377 of 541
TESLA, INC.

14. The resource management device as recited in claim 8, wherein the resource management device further responds to externally generated action requests.

15. The resource management device as recited in claim 14, where the action requests are generated by an operator.

16. In a distributed environment comprised of N hosts operating in a distributed environment instantiating M managed characteristic applications, resource allocation control software instantiated by at least the N hosts, the software comprising:

a first function which determines the state and health of the N hosts, a network operatively coupling the N hosts to one another and the M managed characteristic applications in the distributed environment;

a second function which determines required allocation and reallocation actions need to maintain Quality of Service (QoS) requirements established for the M managed characteristic applications; and

a third function which generates automatic control signal requests corresponding to the actions dictated by the QoS requirements.

17. The software as recited in claim 16, wherein the first function receives system specification information comprising selected ones of host configuration and capabilities, application capabilities, survivability requirements, scalability characteristics, application startup and shutdown dependencies, and application and path performance requirements.

18. The software as recited in claim 16, wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic applications, and detected failures regarding the N hosts.

Ex.1002 / Page 378 of 541
TESLA, INC.

18. The software as recited in claim 16, wherein the first function receives historical data regarding statuses, configuration, and loads of the N hosts and link statuses and loads regarding the network.

19. The software as recited in claim 16, wherein the first function receives application performance data representing each one of the M managed characteristic applications.

20. The software as recited in claim 16, wherein the first function receives application performance data on all applications instantiated by the N hosts including performance data representing each one of the M managed characteristic applications.

21. The software as recited in claim 16, wherein the second function which determines the required allocation and reallocation actions need to maintain the Quality of Service (QoS) requirements established for the M managed characteristic applications by:

responding to application and host failures by determining if and what recovery actions should be taken;

determining if and where to place new copies of one of the M managed characteristic applications or which of the M managed characteristic applications should be shutdown when QoS Manager functions indicate that scale-up or scale-down actions are indicated based on measured application performance and application QoS specifications;

determining where new applications should be placed when requested to do so by a program control device; and

determining which and how many applications should run based on application system priorities.

22. The software as recited in claim 16, wherein one of the M managed characteristic applications comprises a scalable application.

Ex.1002 / Page 379 of 541
TESLA, INC.

23. The software as recited in claim 16, wherein one of the M managed characteristic applications comprises a fault tolerant application, where the degree of fault tolerance is selectable by a user.

24. The software as recited in claim 16, wherein one of the M managed characteristic applications comprises a selectable priority application.

25. The software as recited in claim 16, wherein the M managed characteristic applications comprise M copies of a single managed characteristic application.

F:\56\3523\3523.amd

26. Software stored on at least one host for converting N networked hosts into a resource managed system instantiating M managed characteristic applications, the software comprising:

a first function group which monitors the host and network resources

a second function group which provides general-purpose application event reporting and event correlation capabilities;

a third function group which provides the reasoning and decision-making capabilities for the resource managed system, wherein the third function group comprises:

a first function which determines the state and health of the N hosts, a network operatively coupling the N hosts to one another and the M managed characteristic applications in the distributed environment;

a second function which determines required allocation and reallocation actions need to maintain Quality of Service (QoS) requirements established for the M managed characteristic applications; and

a third function which generates automatic control signal requests corresponding to the actions dictated by the QoS requirements; and

a fourth function group which provides program control capabilities permitting starting, stopping, and configuring of selected ones of the M managed characteristic applications on respective ones of the N hosts in the resource managed system,

where M and N are positive integers.

27. The software as recited in claim 26, wherein the first function receives system specification information comprises host configuration and capabilities.

28. The software as recited in claim 26, wherein the first function receives system specification information comprising selected ones of capabilities, survivability requirements, scalability characteristics, startup and shutdown dependencies, and performance requirements for at least one of the M managed characteristic applications.

29. The software as recited in claim 26, wherein the first function receives system specification information comprising path performance requirements regarding communication between at least two of the N hosts.

30. The software as recited in claim 26, wherein the first function receives program control information comprising application status and detected application faults for each of the M managed characteristic applications, and detected failures regarding the N hosts.

31. The software as recited in claim 26, wherein the first function receives historical data regarding statuses, configuration, and loads of the N hosts and link statuses and loads regarding the network.

32. The software as recited in claim 26, wherein the first function receives application performance data representing each one of the M managed characteristic applications.

Ex.1002 / Page 381 of 541
TESLA, INC.

33. The software as recited in claim 26, wherein the first function receives application performance data on all applications instantiated by the N hosts including performance data representing each one of the N copies of the managed characteristic application.

34. The software as recited in claim 26, wherein the second function which determines the required allocation and reallocation actions established for the M managed characteristic application by:

responding to application and host failures by determining if and what recovery actions should be taken;

determining if and where to place new copies of managed characteristic applications or which managed characteristic applications should be shutdown when QoS Manager functions indicate that scale-up or scale-down actions should be taken based on measured application performance and QoS specifications established for the M managed characteristic applications;

determining where new applications should be placed when requested to do so by the fourth function group; and

determining which and how many applications should run based on application system priorities.

35. The software as recited in claim 26, wherein the third function group makes decisions by one of:

based on requests from Program Control, determining where new applications should be started;

based on indication of application failure from the fourth function group, determining whether and where the failed application should be restarted;

based on indication of host failure from the fourth function group, determining whether and where the failed application previously instantiated by the failed one of the N hosts should be restarted;

Ex.1002 / Page 382 of 541
TESLA, INC.

based on startup and shutdown dependency resolution requests from the fourth function group, determine whether and where additional applications should to be one of started and shut down prior to starting or shutting down another application; and

based on changes to application system priorities, determining whether and where new applications need to be started and/or determine whether and which existing applications need to be shutdown.

36. The software as recited in claim 26, wherein the third function group makes decisions one of:

based on application inter-dependencies defined in system specification files, determining whether and where additional applications should to be one of started and shut down prior to starting or shutting down of another application;

based on application instrumentation data generated by the second function group and performance requirements defined in the system specification files, determining whether applications are meeting performance requirements and whether an application can be scaled up or moved to attempt to improve performance; and

based on the application instrumentation data and performance requirements defined in the system specification files, determining whether applications are performing well within performance requirements and can be scaled down.

Ex.1002 / Page 383 of 541
TESLA, INC.

Exhibit 1002 - Page 384 of 541

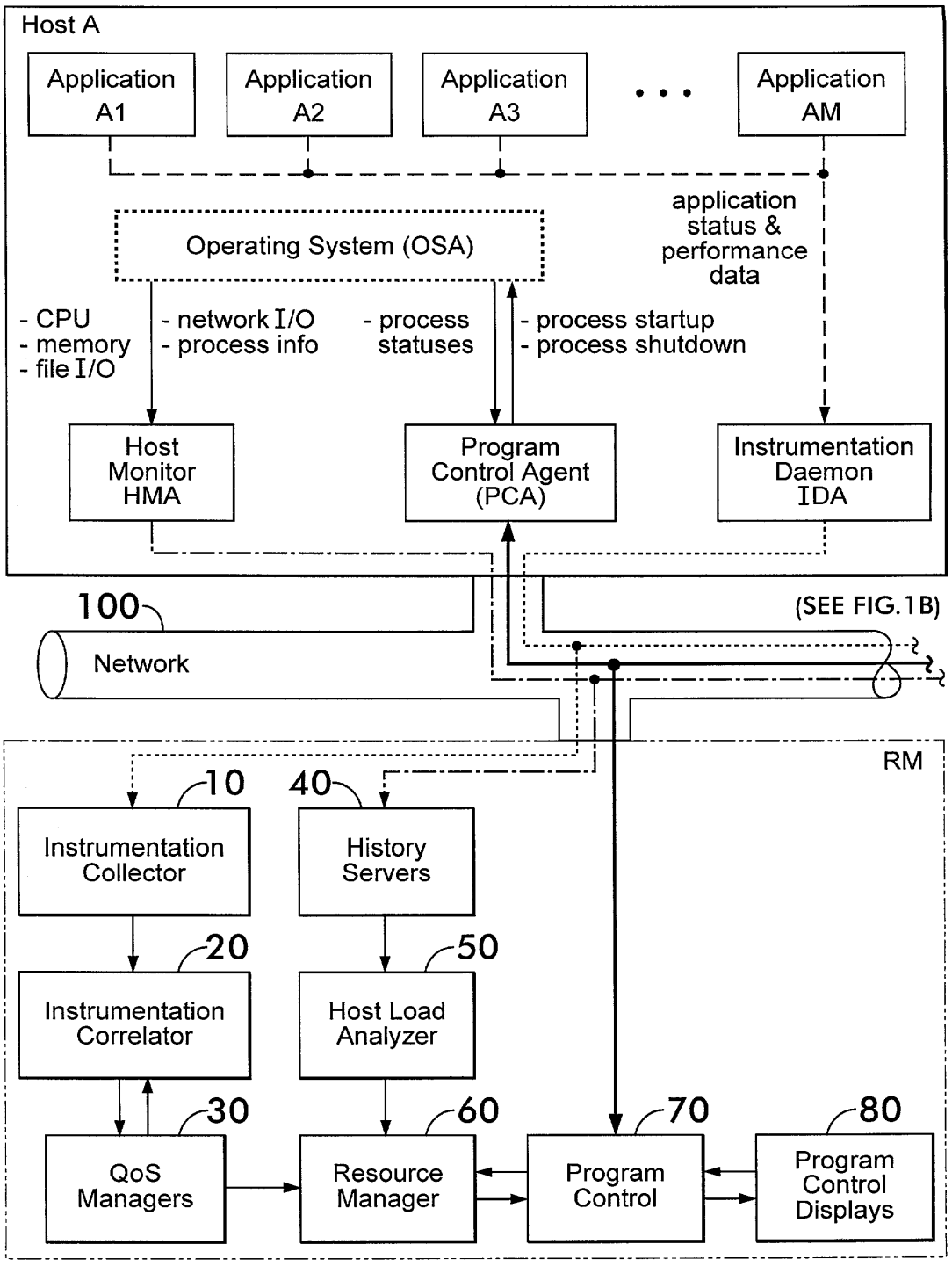
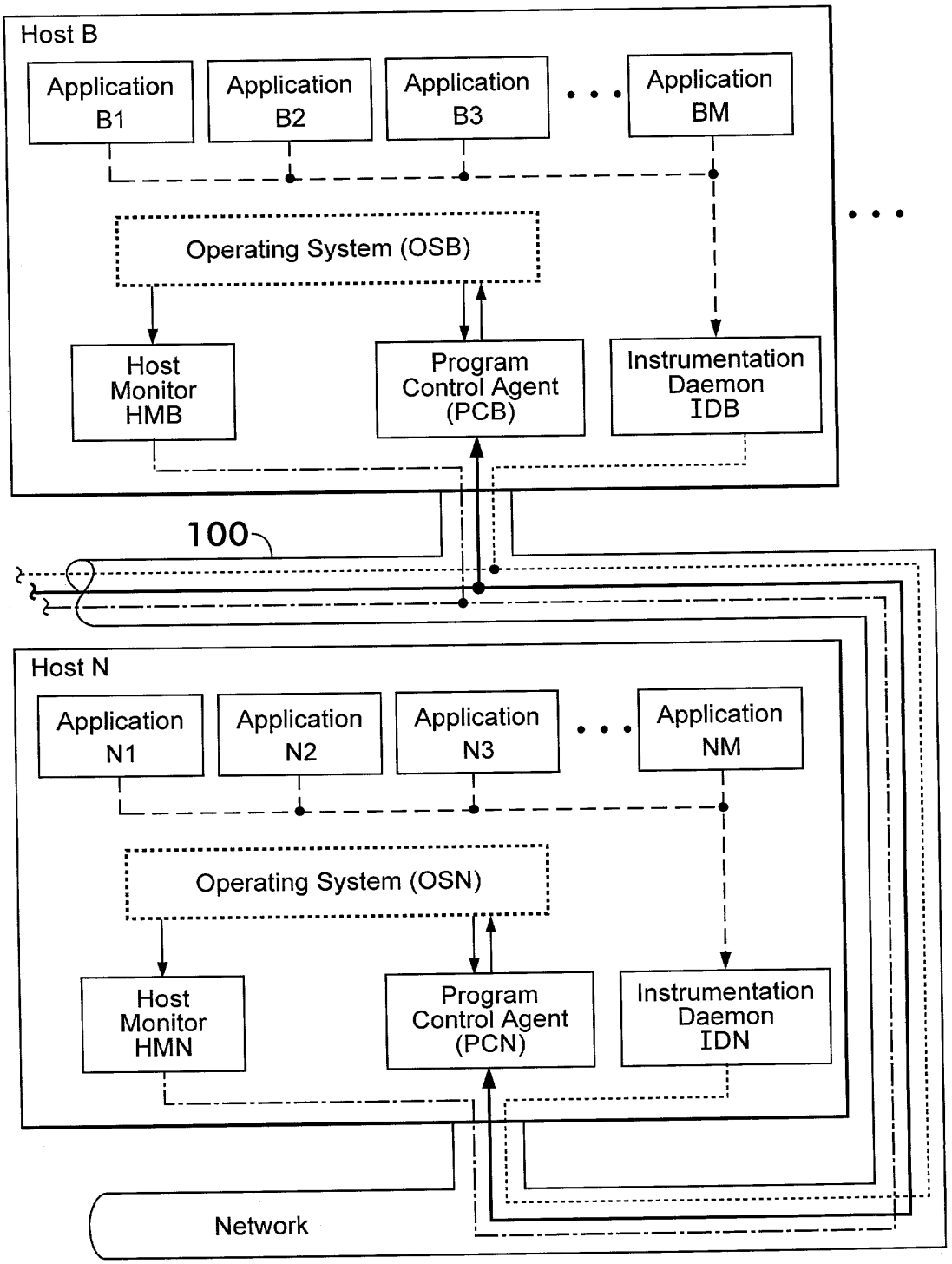


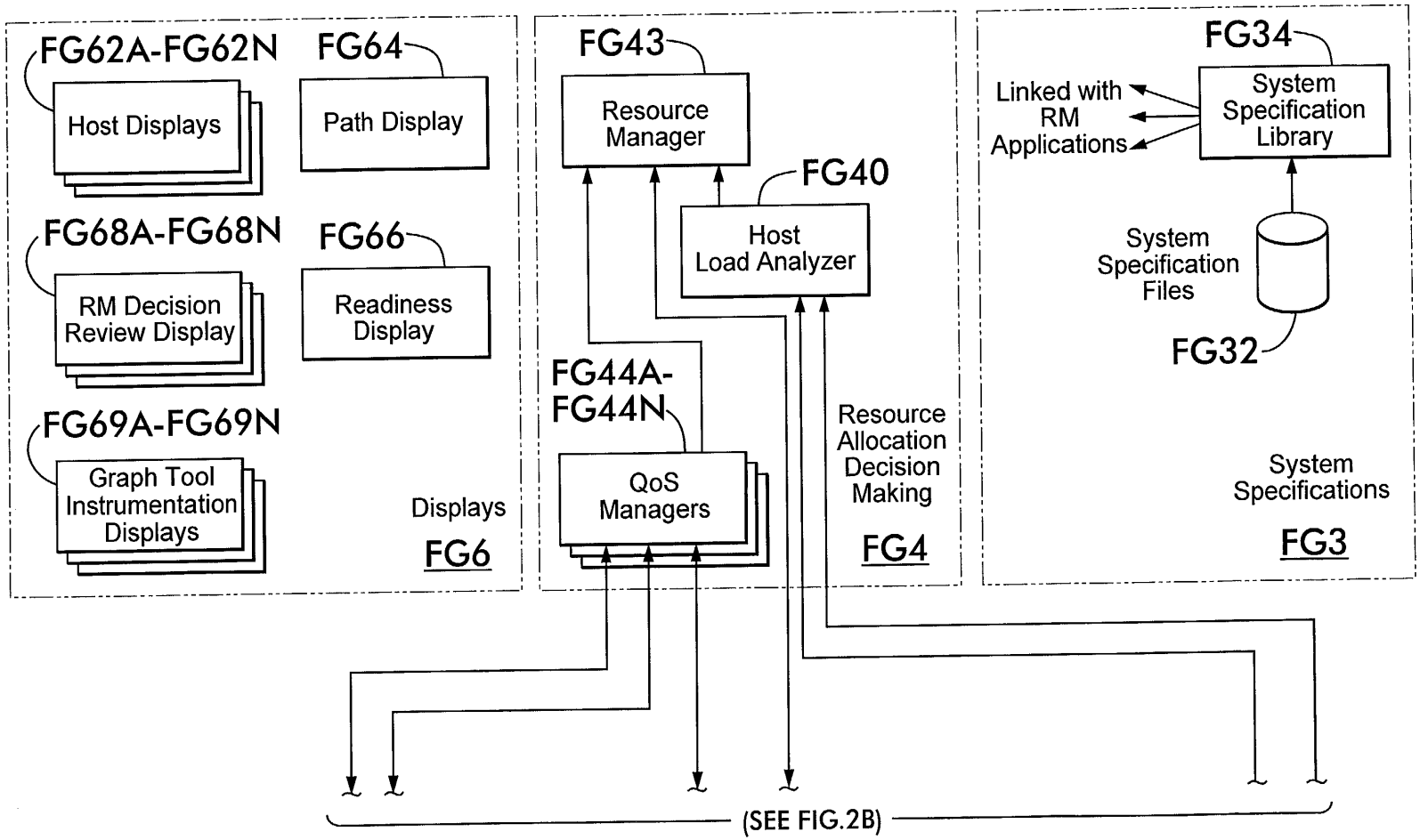
FIG. 1A

FIG.IB


Host B
Application B1
Application B2
Application B3
Application BM
Operating System (OSB)
Host Monitor HMB
Program Control Agent (PCB)
Instrumentation Daemon IDB
100
Host N
Application N1
Application N2
Application N3
Application NM
Operating System (OSN)
Host Monitor HMN
Program Control Agent (PCN)
Instrumentation Daemon IDN
Network

FG6T60" 5288999660

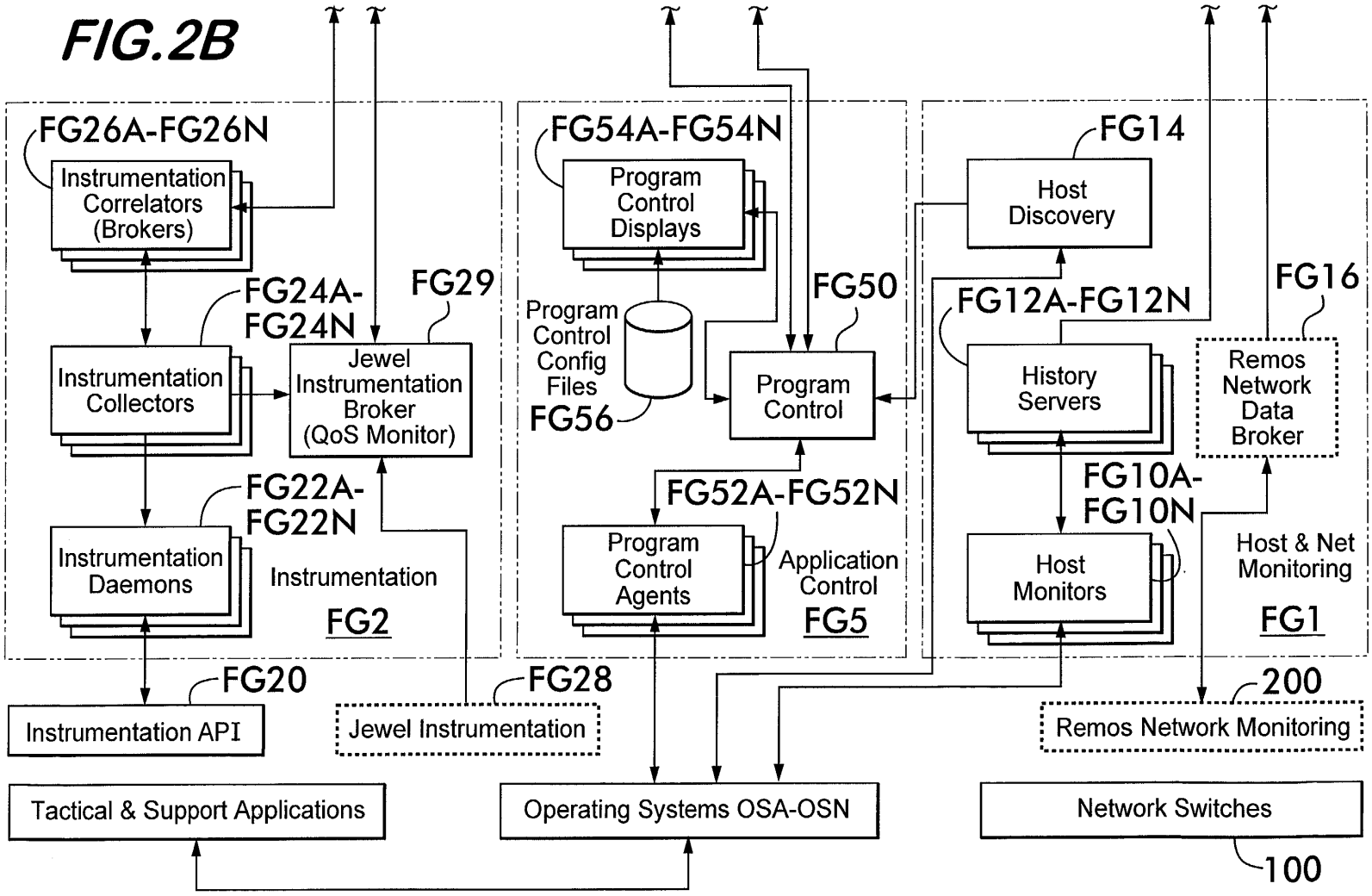
FIG.2A



Ex.1002 / Page 386 of 541
TESLA, INC.

T0670 S25h9860

FIG.2B



T06T60"529h9860

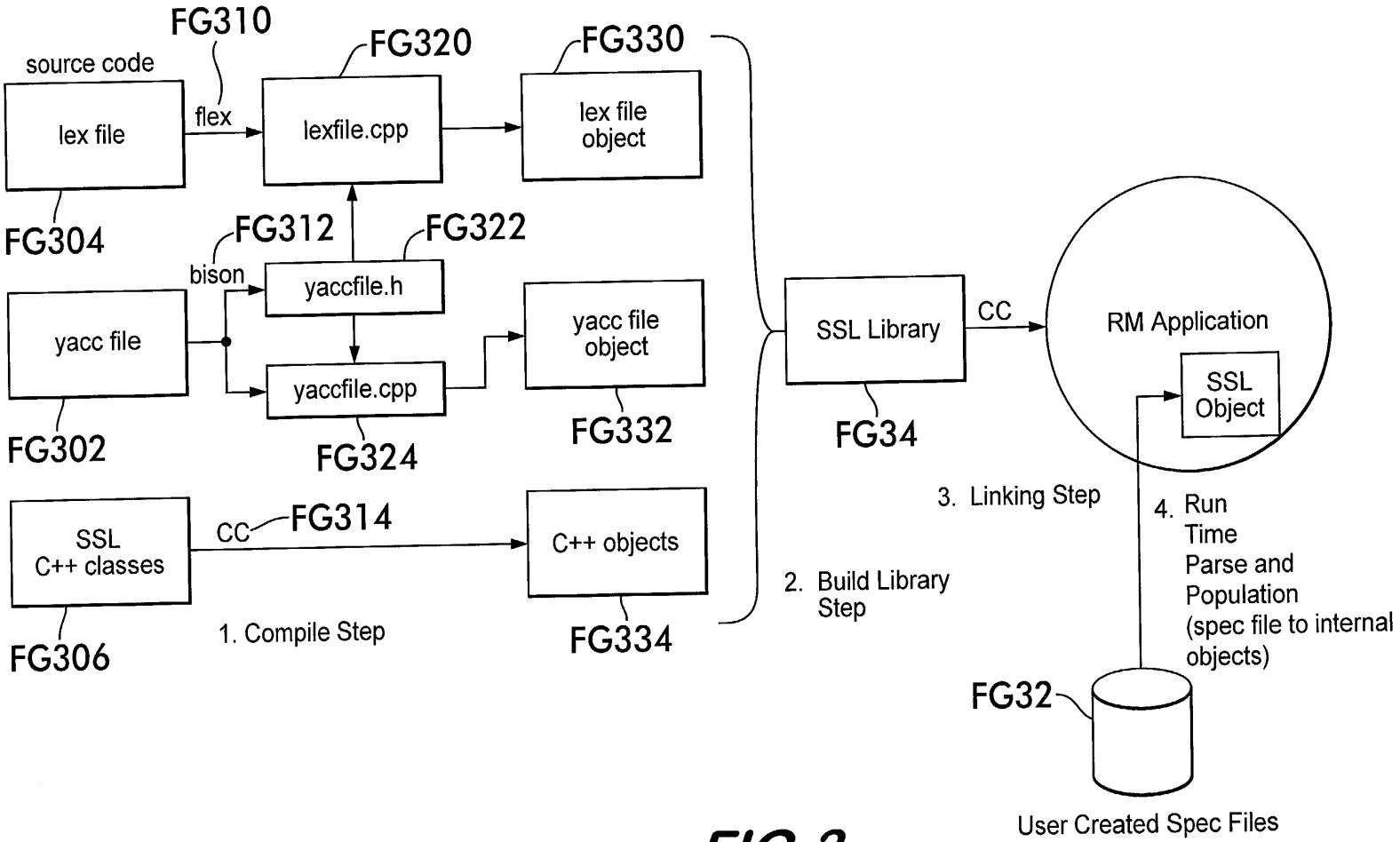
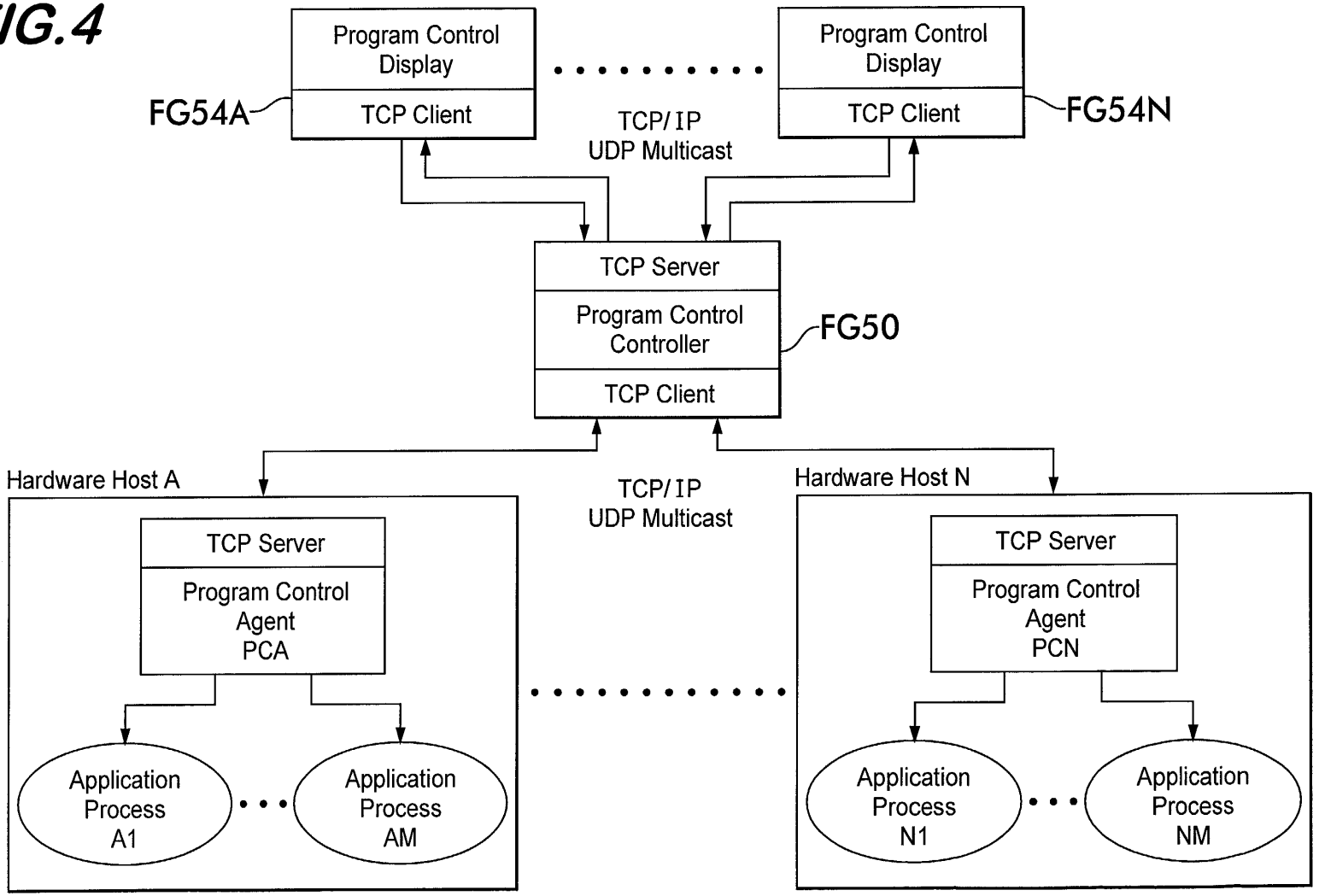


FIG.3

US 6,356,926 B1

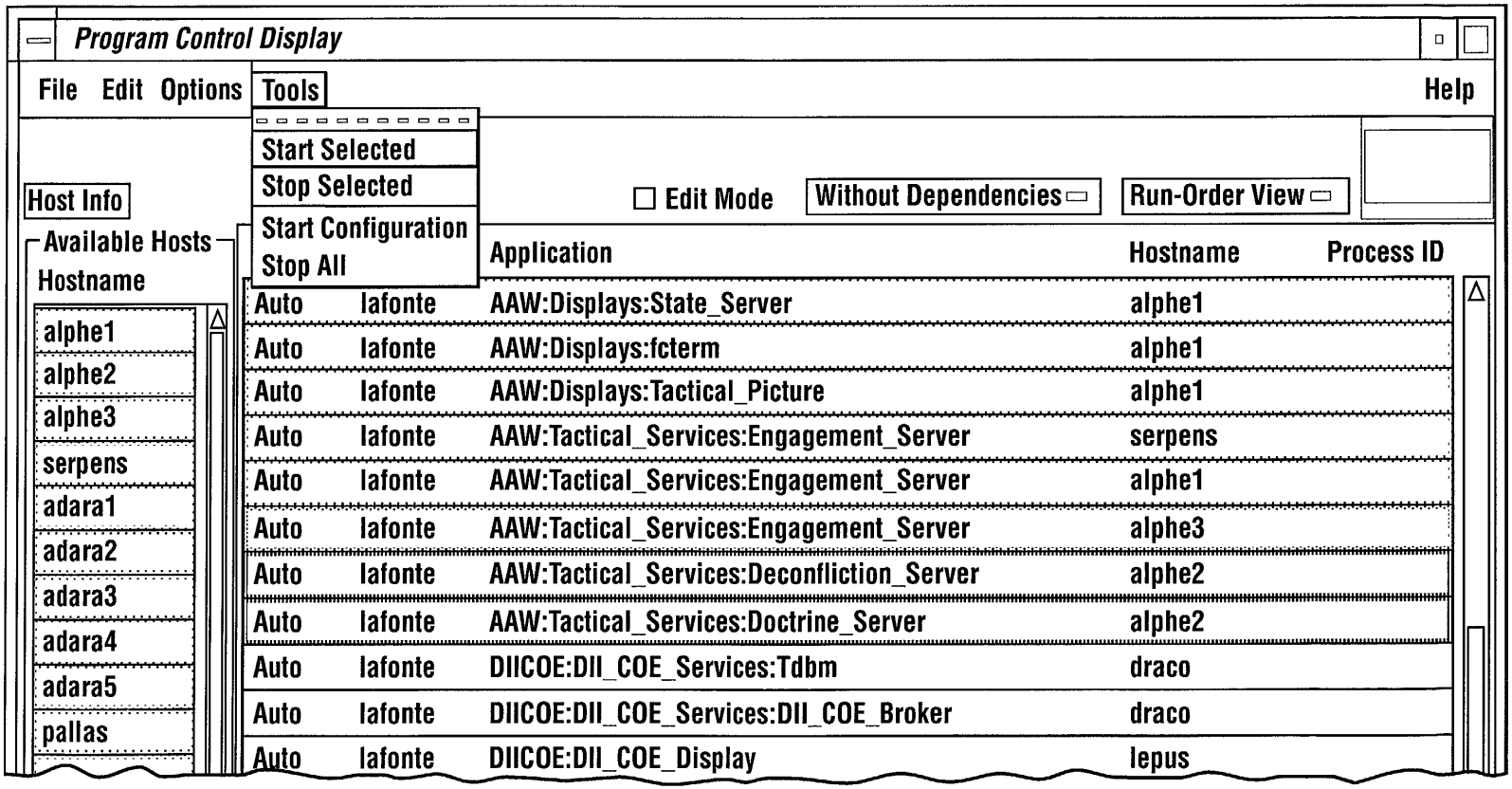
FIG.4



Ex.1002 / Page 389 of 541
TESLA, INC.

T06T60' 529+9860

FIG.5A



(SEE FIG.5B)

Ex.1002 / Page 390 of 541
TESLA, INC.

7661601 5394960

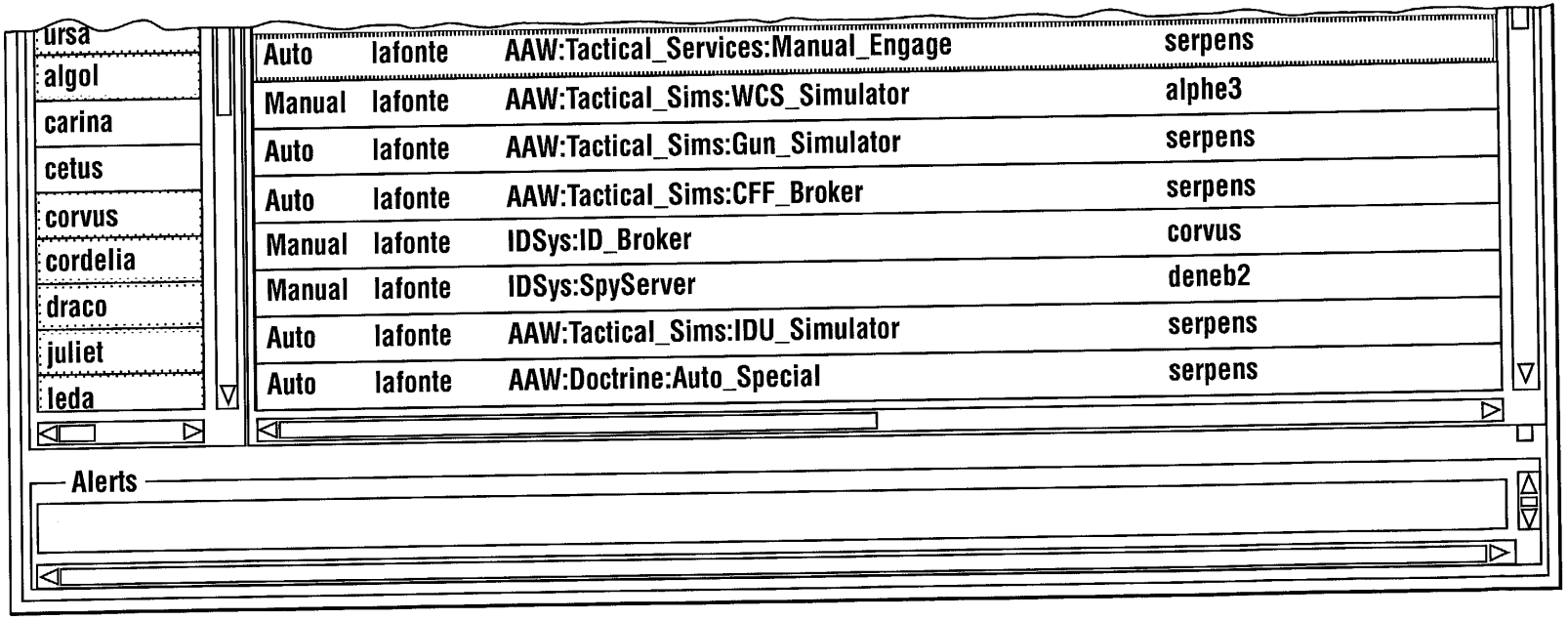
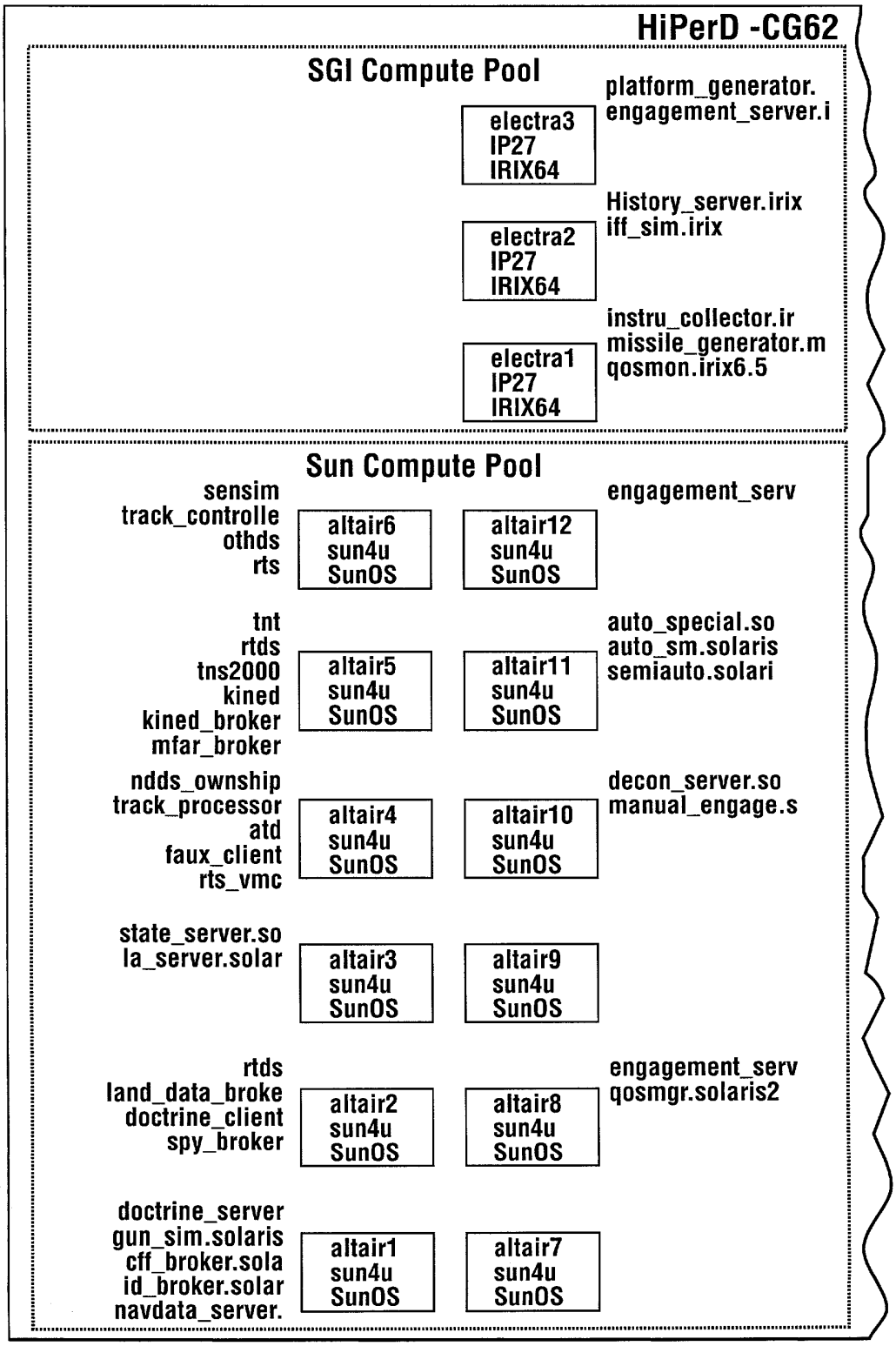


FIG.5B

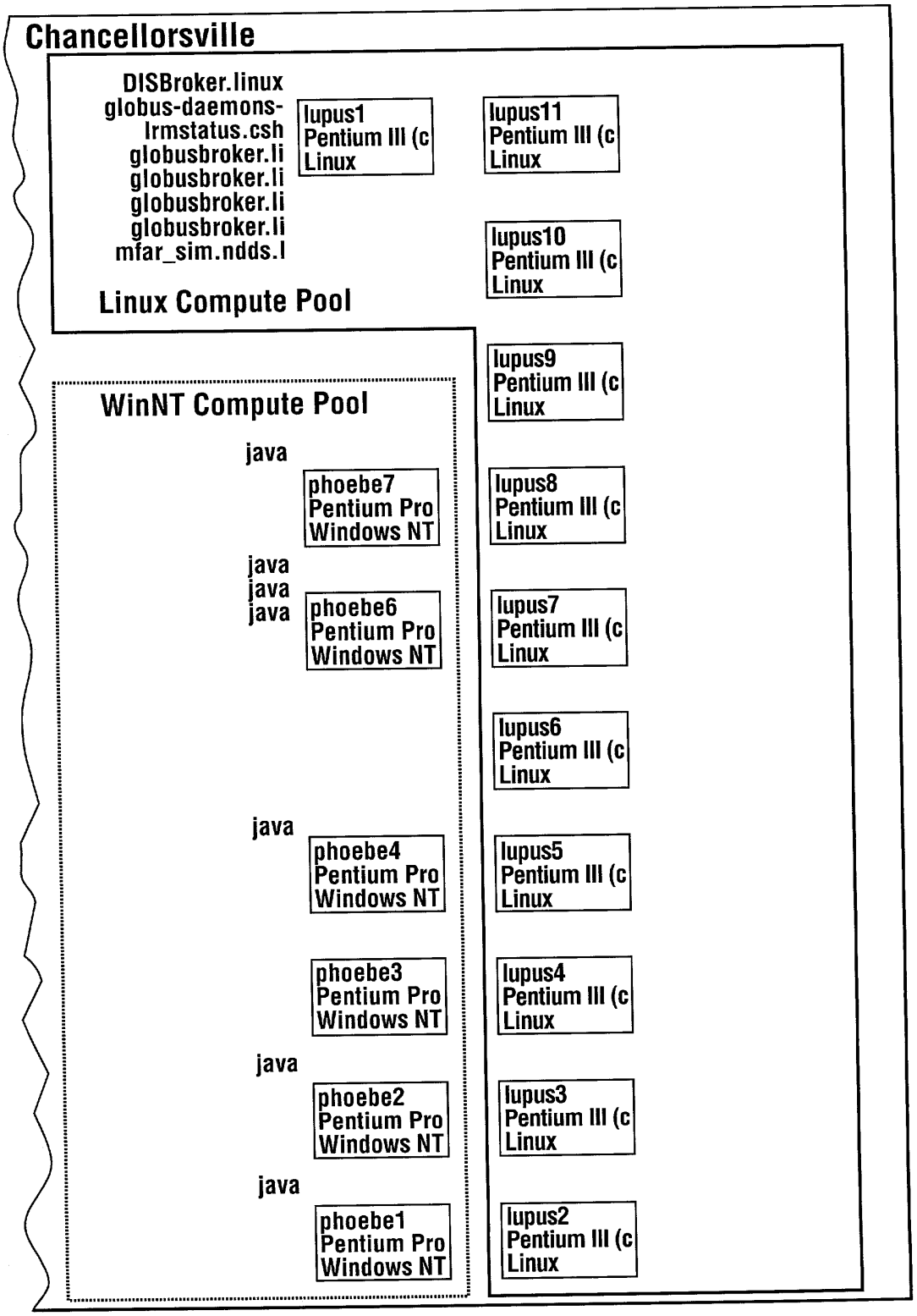
FIG. 6A

(SEE FIG.6B)



Ex.1002 / Page 392 of 541
TESLA, INC.

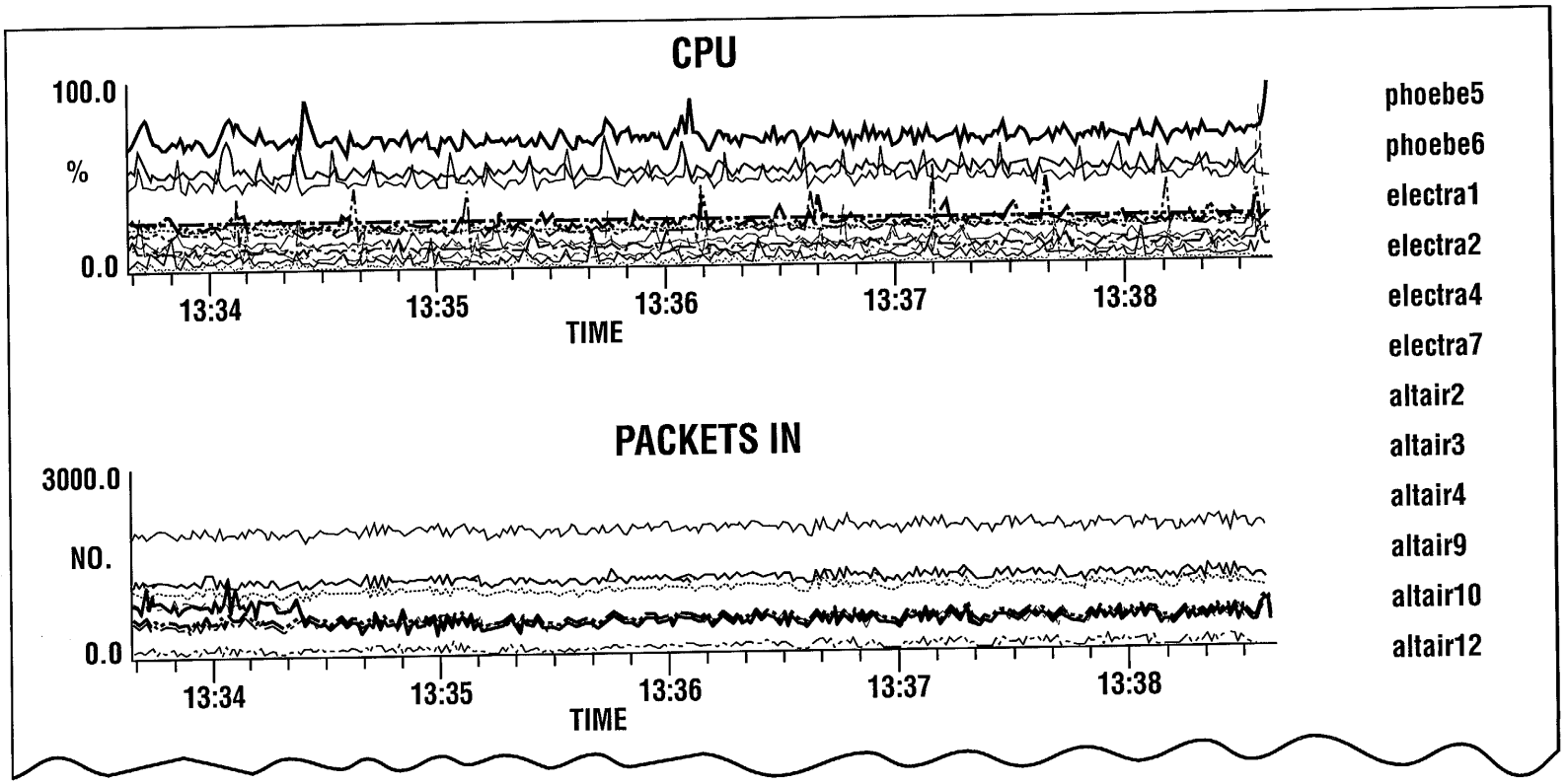
FIG.6B



Ex.1002 / Page 393 of 541
TESLA, INC.

T06T60"S29t9660

FIG. 7A



(SEE FIG.7B)

T06T60" 5294986D

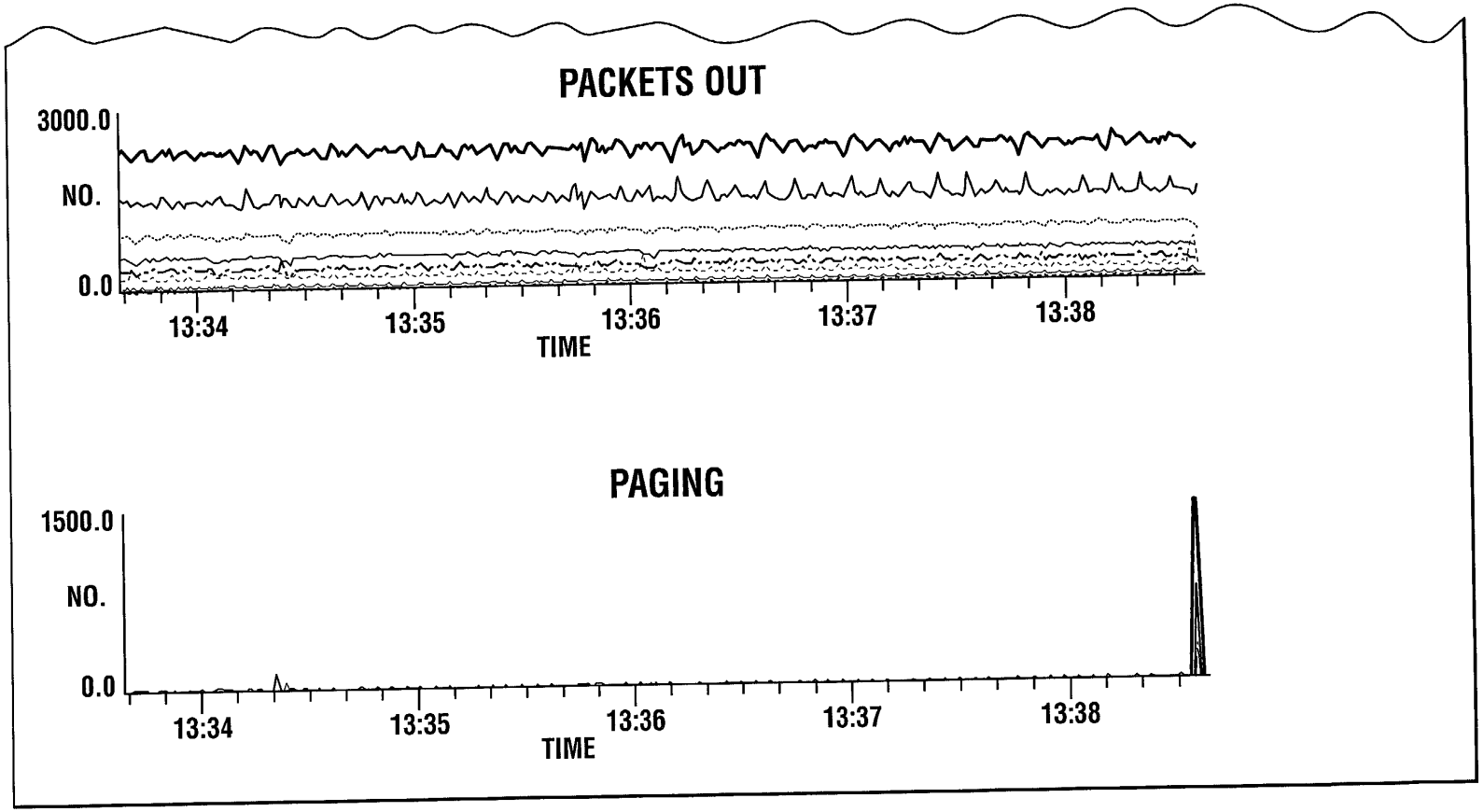
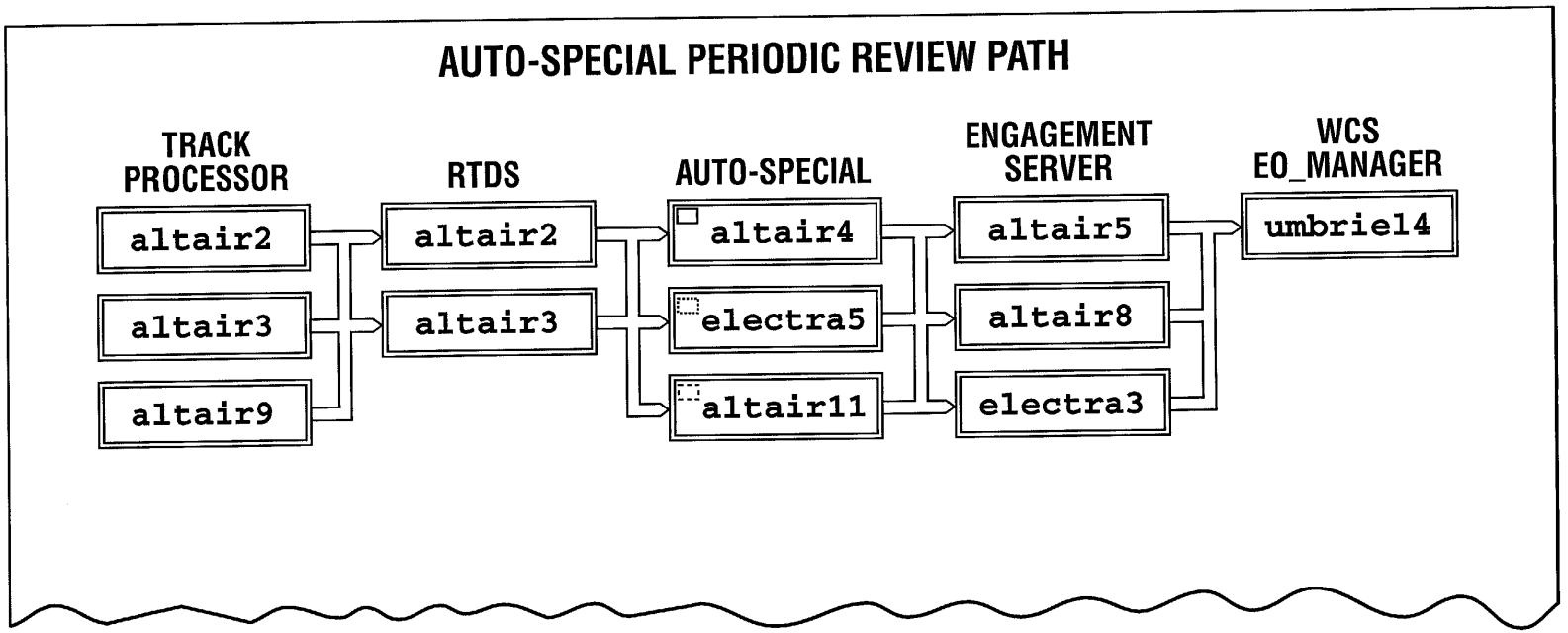


FIG. 7B

T06T60" 529h9860

FIG.8A



(SEE FIG.8B)

FOSTER SEISMIC

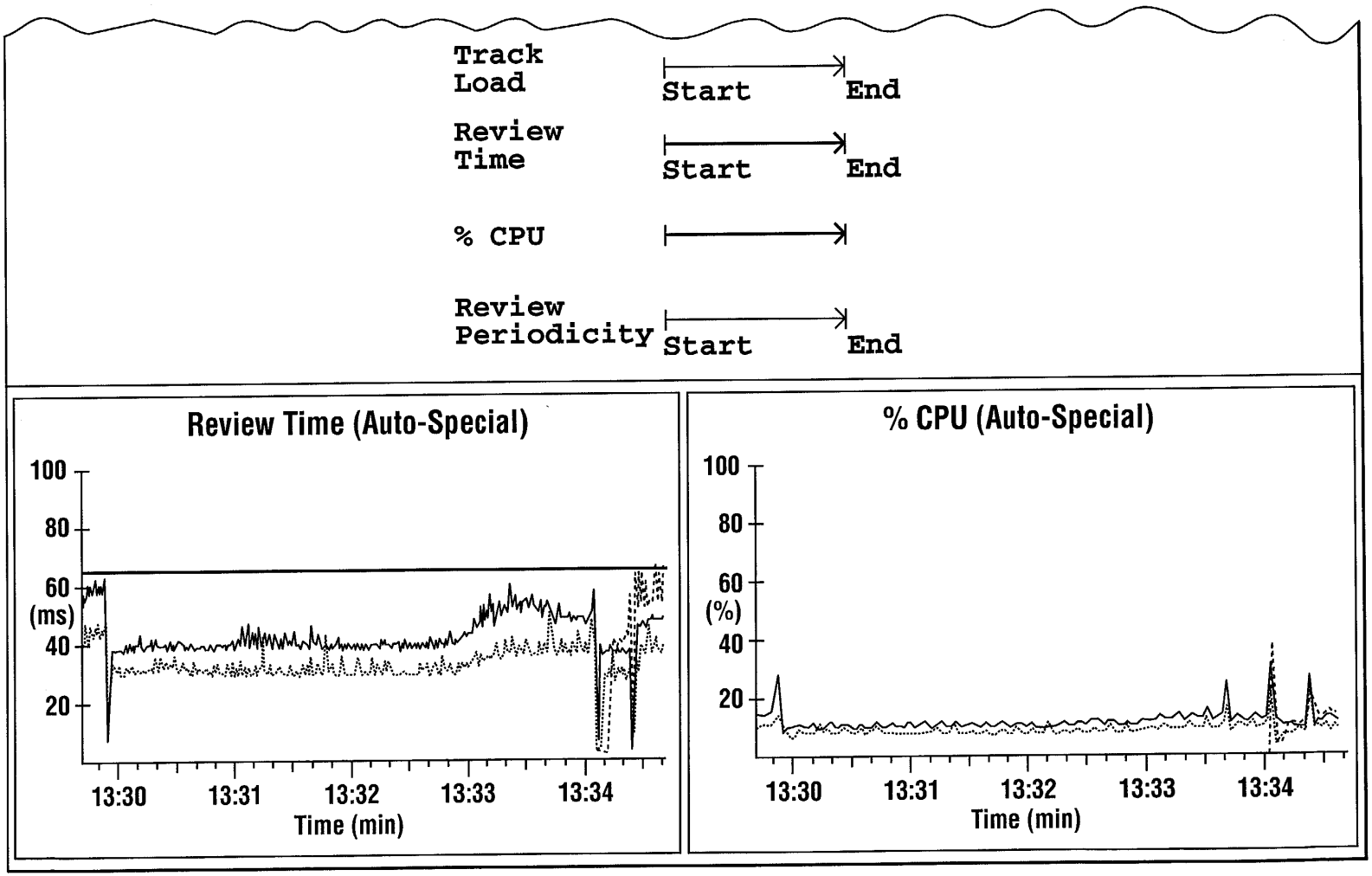
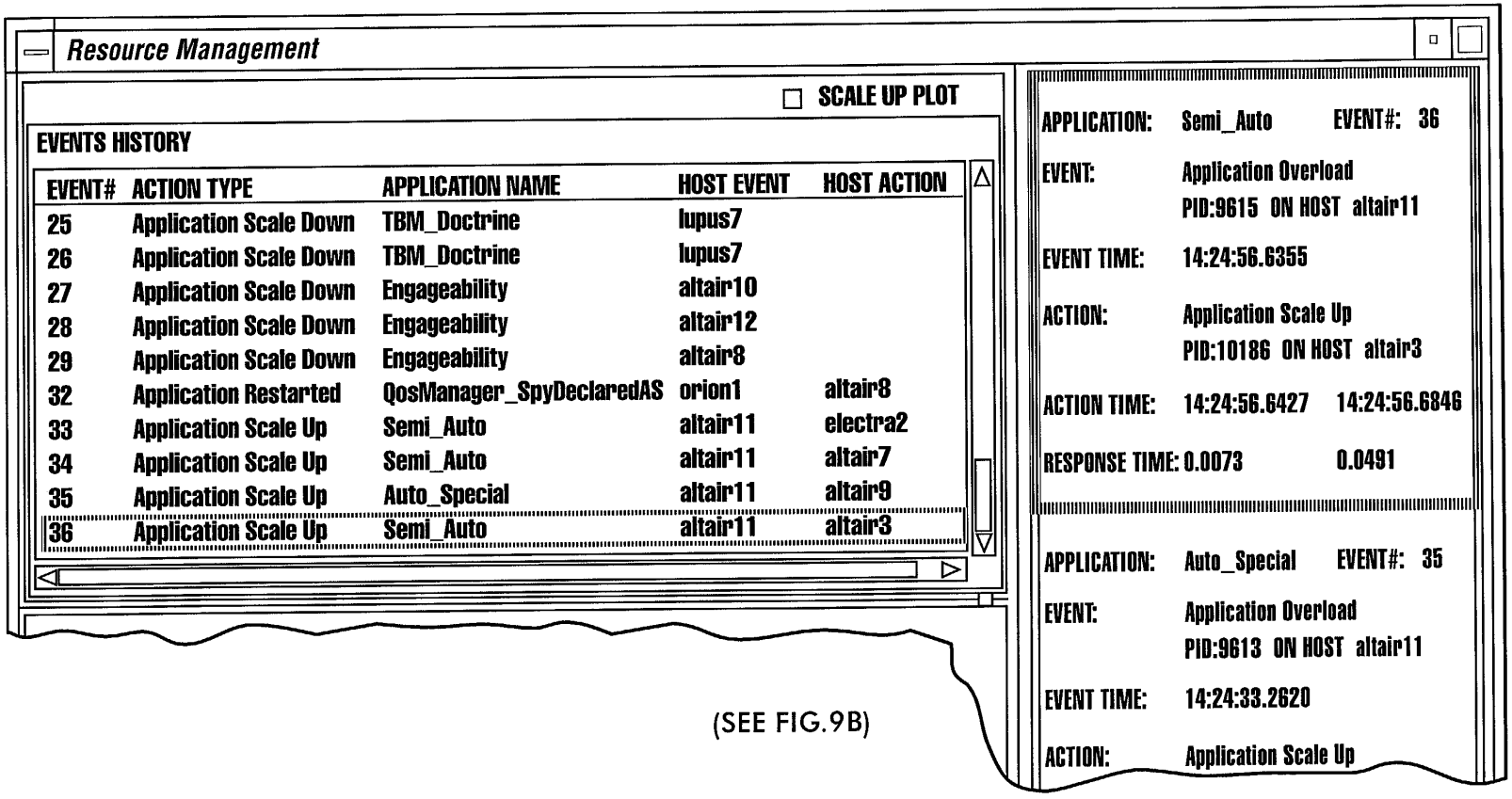


FIG.8B

7061601 5284960

FIG.9A



(SEE FIG.9B)

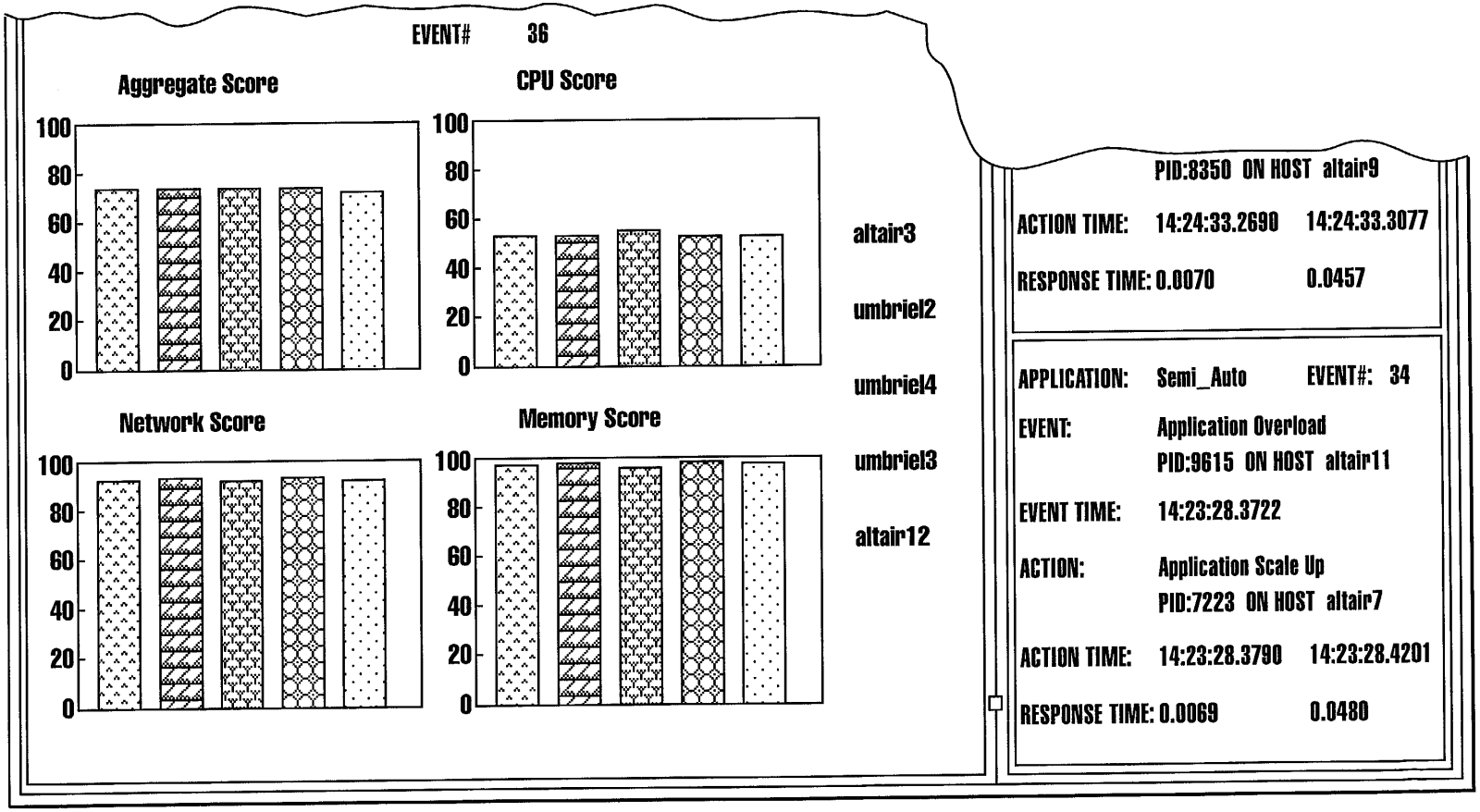
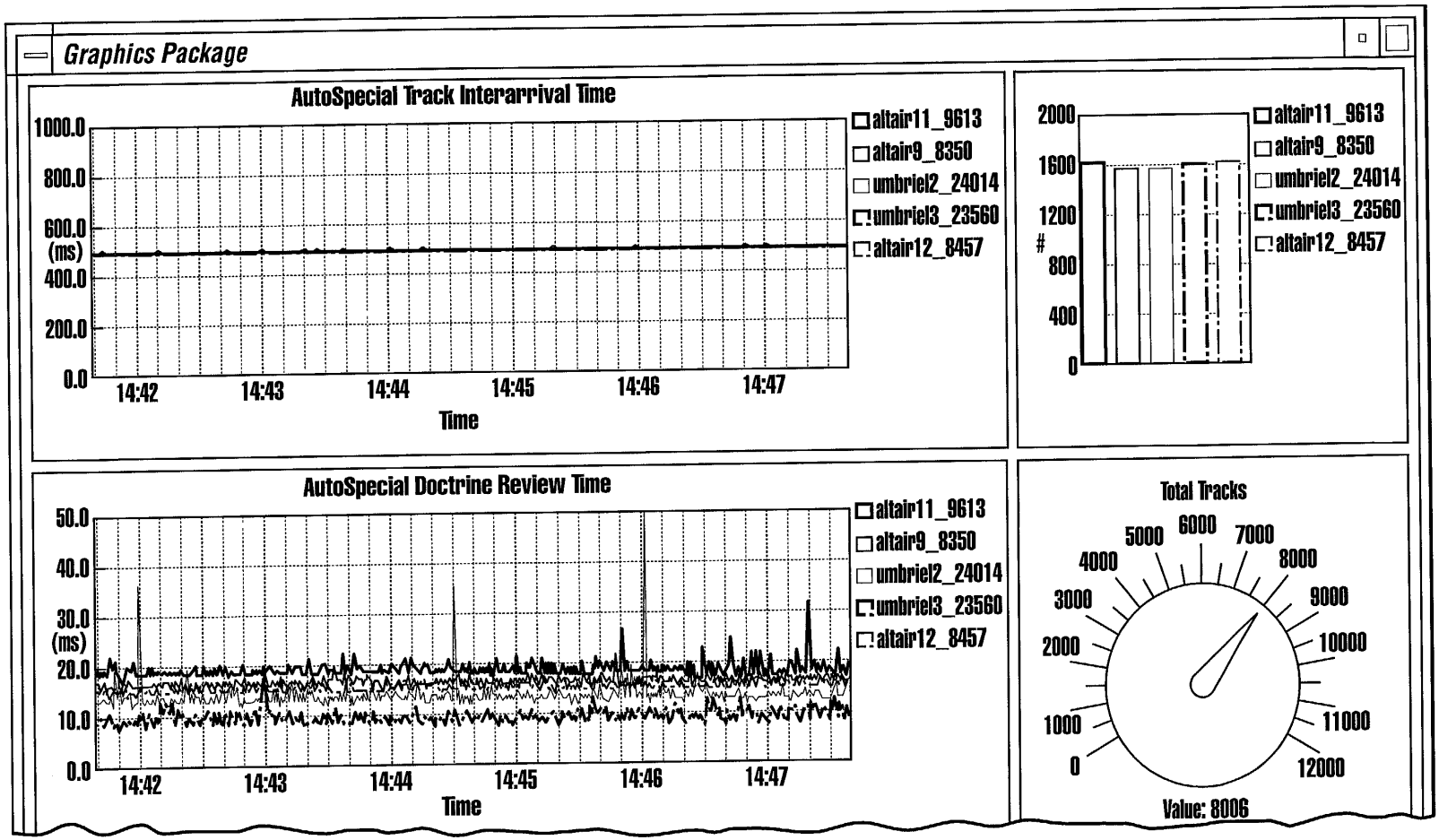

EVENT# 36
Aggregate Score
CPU Score
Network Score
Memory Score
100
80
60
40
20
0
altair3
umbriel2
umbriel4
umbriel3
altair12
PID:8350 ON HOST altair9
ACTION TIME: 14:24:33.2690 14:24:33.3077
RESPONSE TIME: 0.0070 0.0457
APPLICATION: Semi_Auto EVENT#: 34
EVENT: Application Overload
PID:9615 ON HOST altair11
EVENT TIME: 14:23:28.3722
ACTION: Application Scale Up
PID:7223 ON HOST altair7
ACTION TIME: 14:23:28.3790 14:23:28.4201
RESPONSE TIME: 0.0069 0.0480

FIG.9B

T06760 "5284,9860

FIG.10A



(SEE FIG.10B)

Ex.1002 / Page 400 of 541
TESLA, INC.

T06T60" 5295960

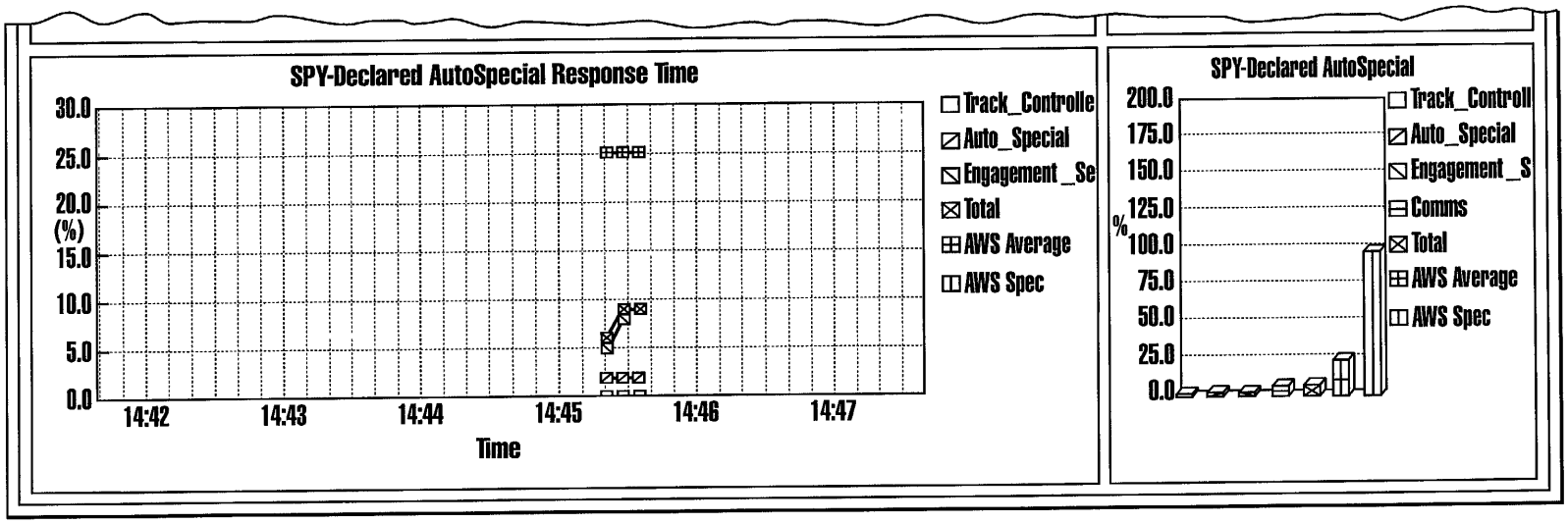
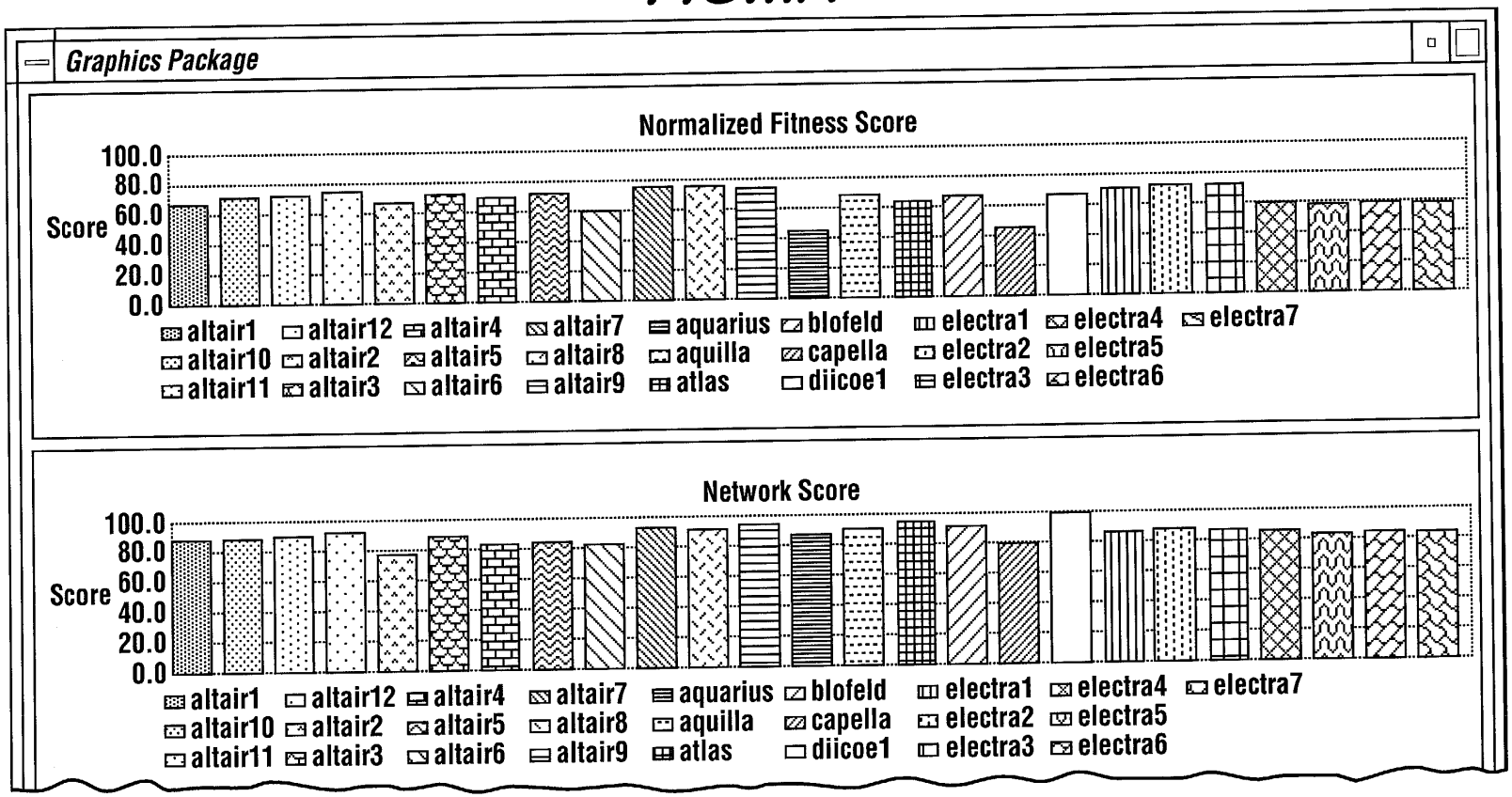


FIG.10B

U.S. Patent Jun. 16, 1998 Sheet 9 of 60 5,768,560

FIG. IIA



(SEE FIG. 11B)

Ex.1002 / Page 402 of 541
TESLA, INC.

T06T60"S38h9660

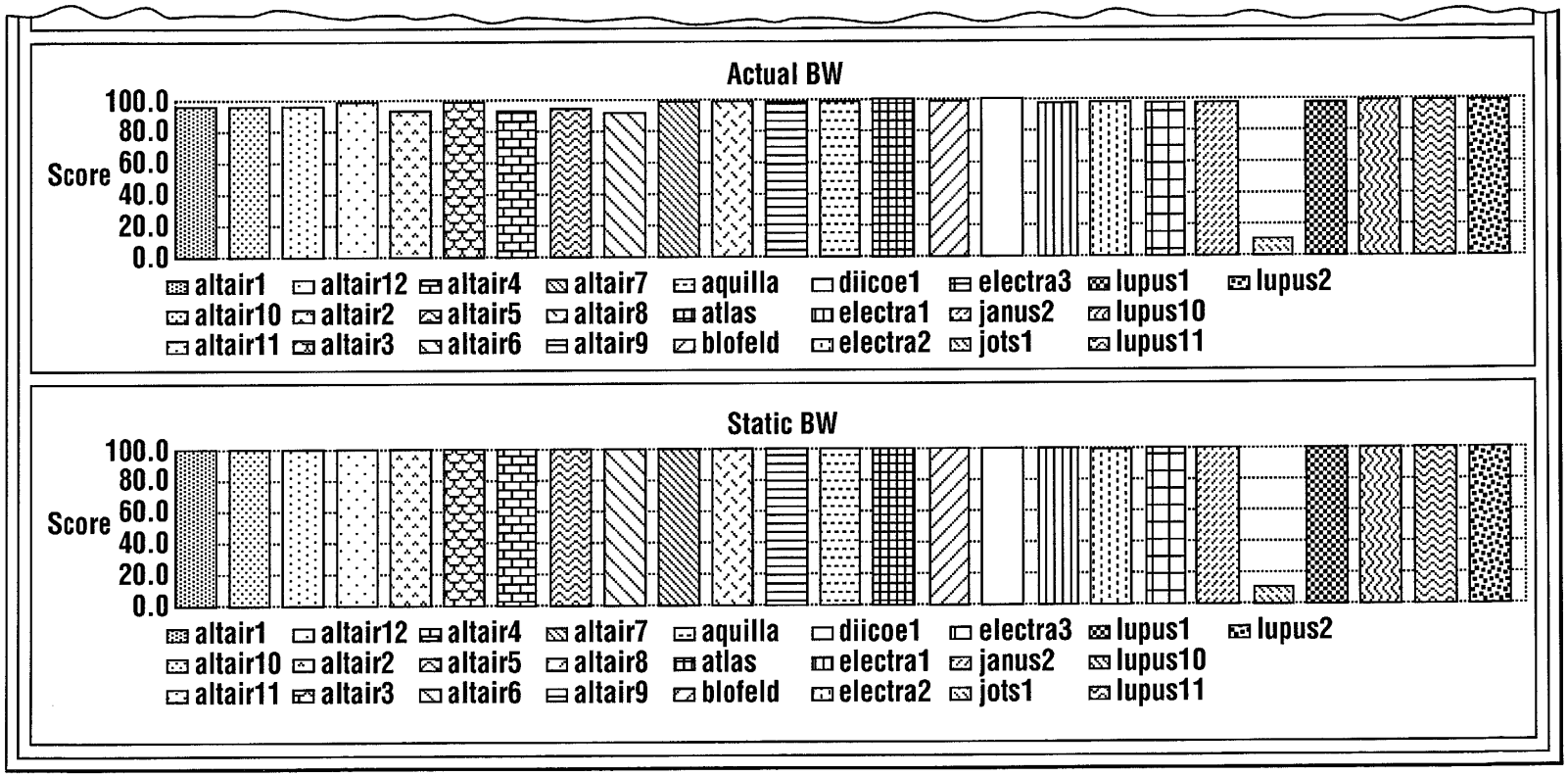
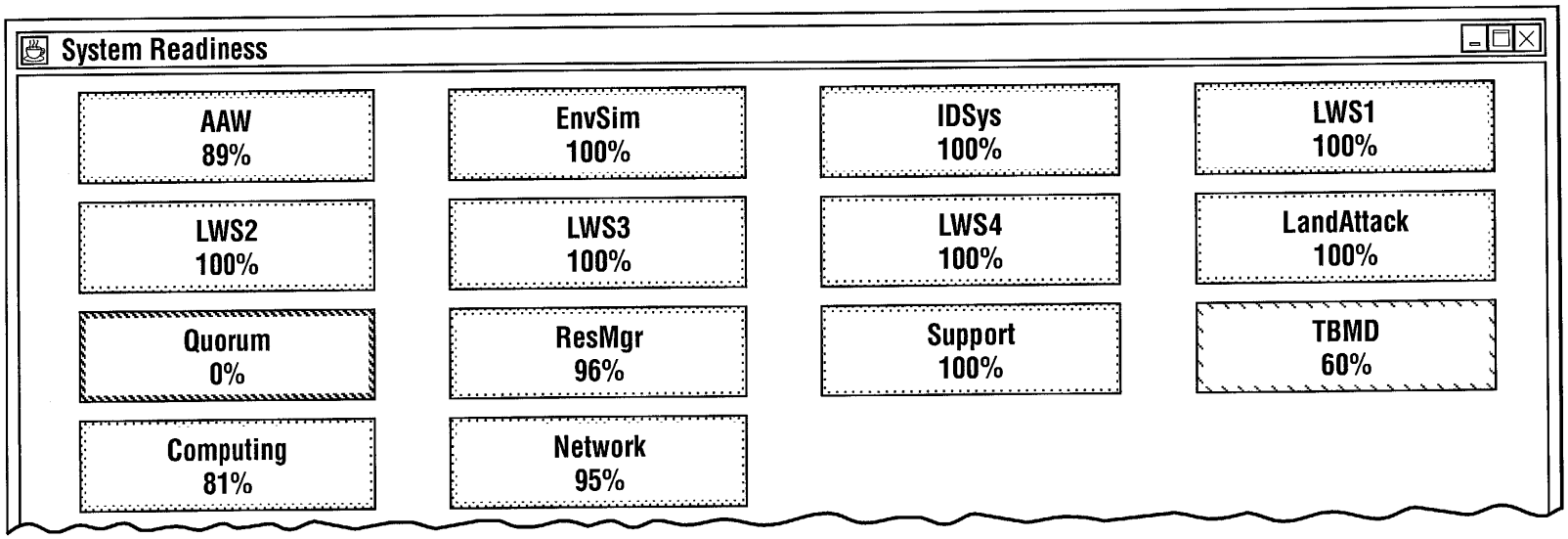


FIG.IIB

T06T60"539%9860

FIG.12A



(SEE FIG.12B)

T06760 "S259,9960

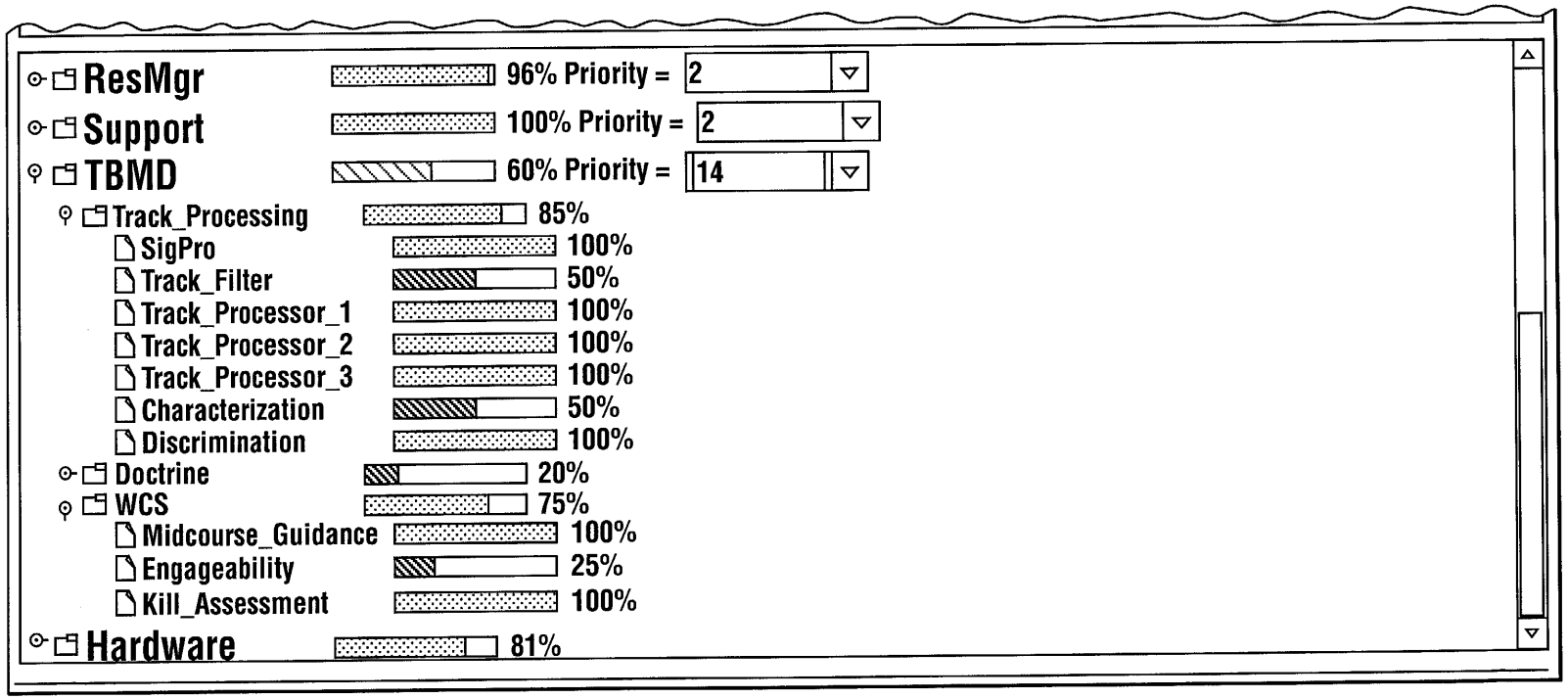
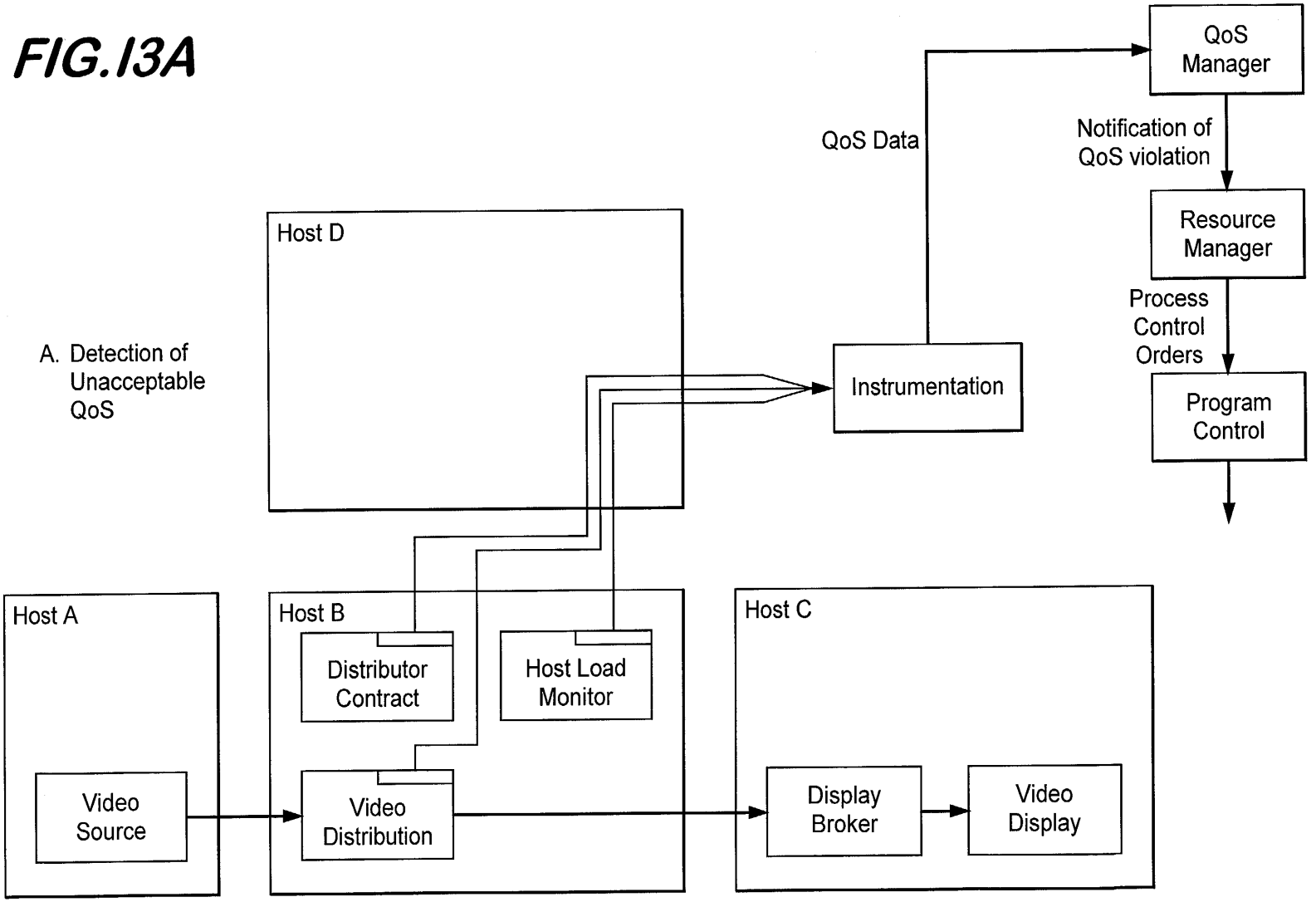


FIG.12B

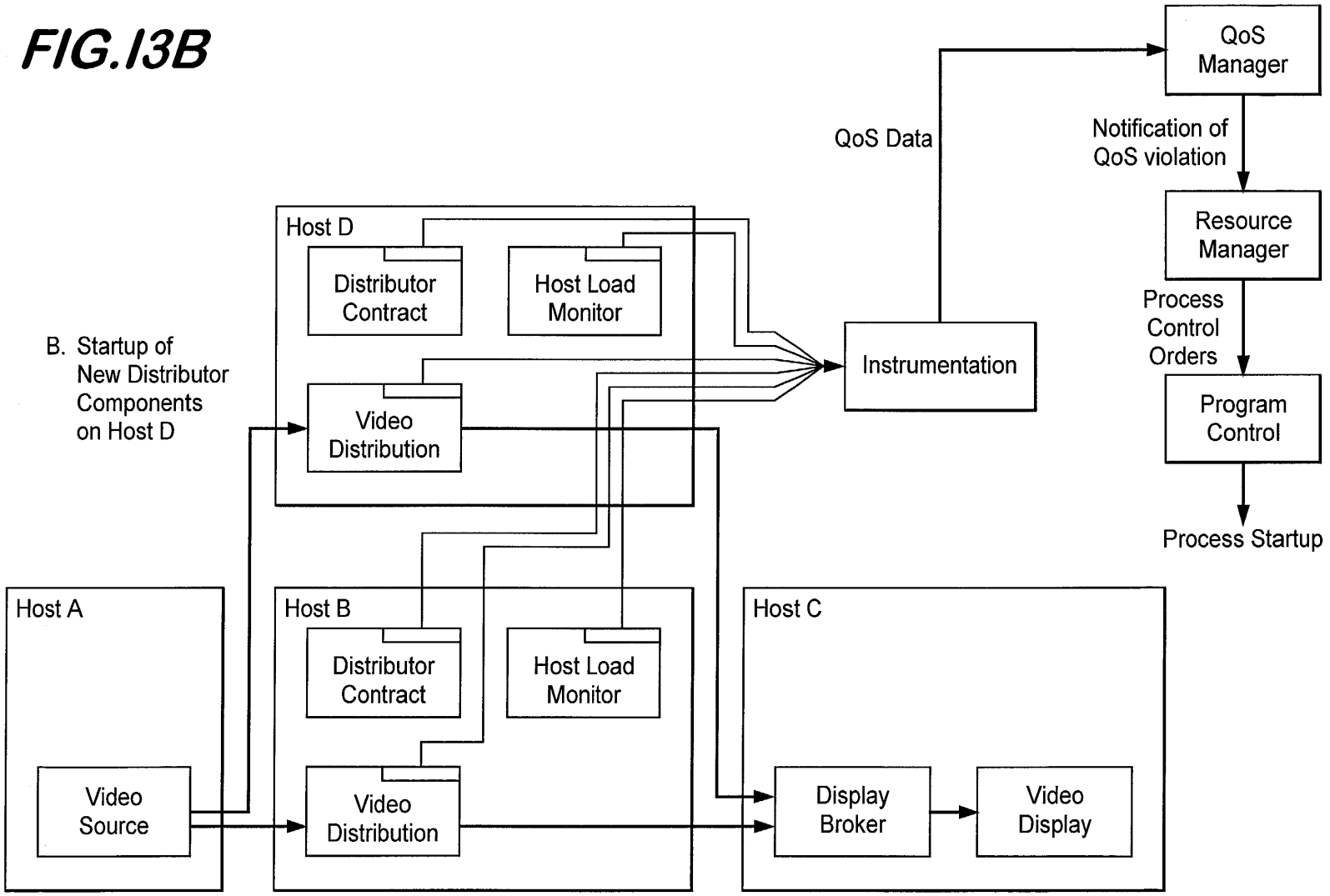
T06760 "5284960

FIG. 13A



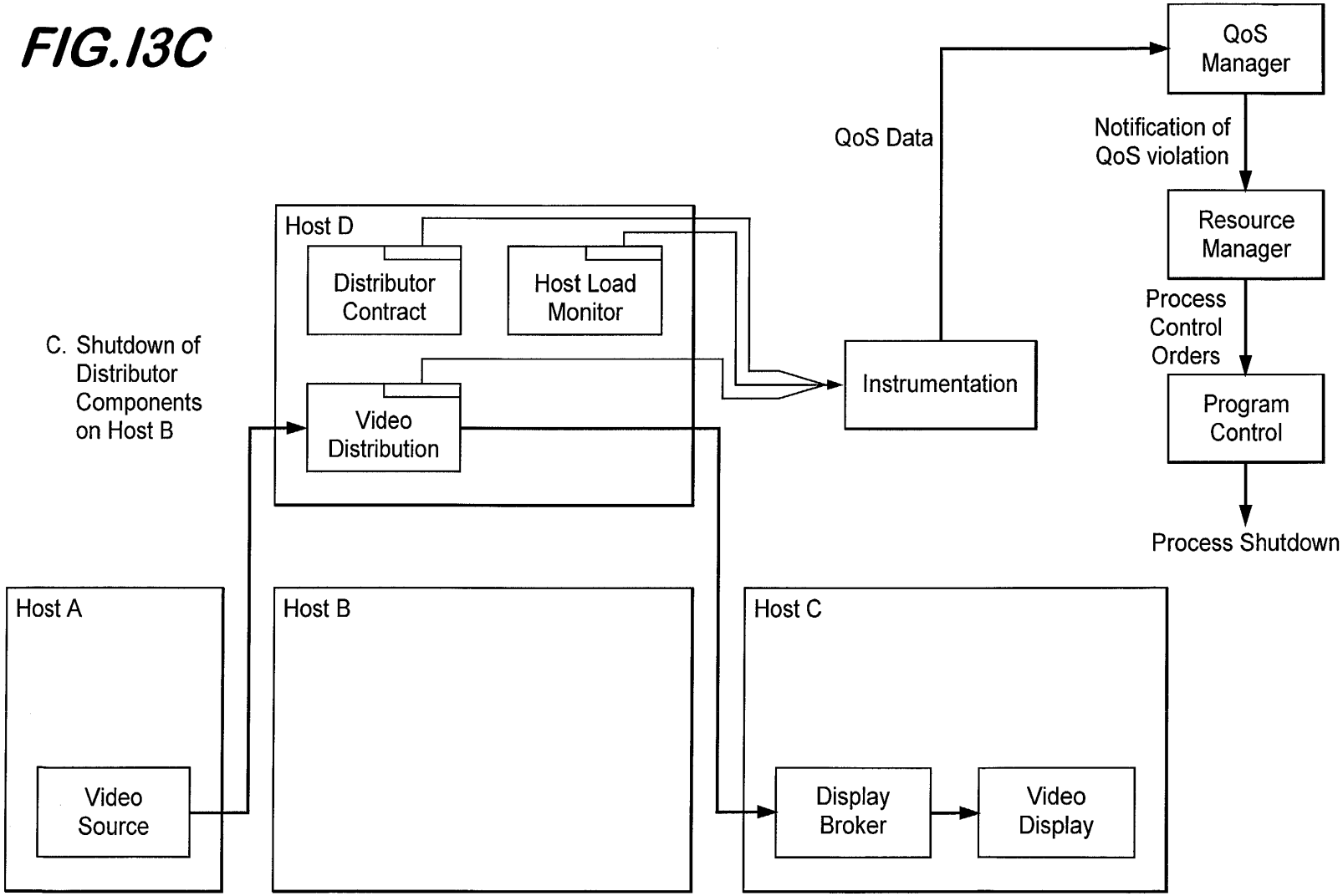
T06760 5256960

FIG.13B



706760 v2 9/9/98 9:59 AM

FIG. 13C



TD6T60" 529h9860

FIG.14

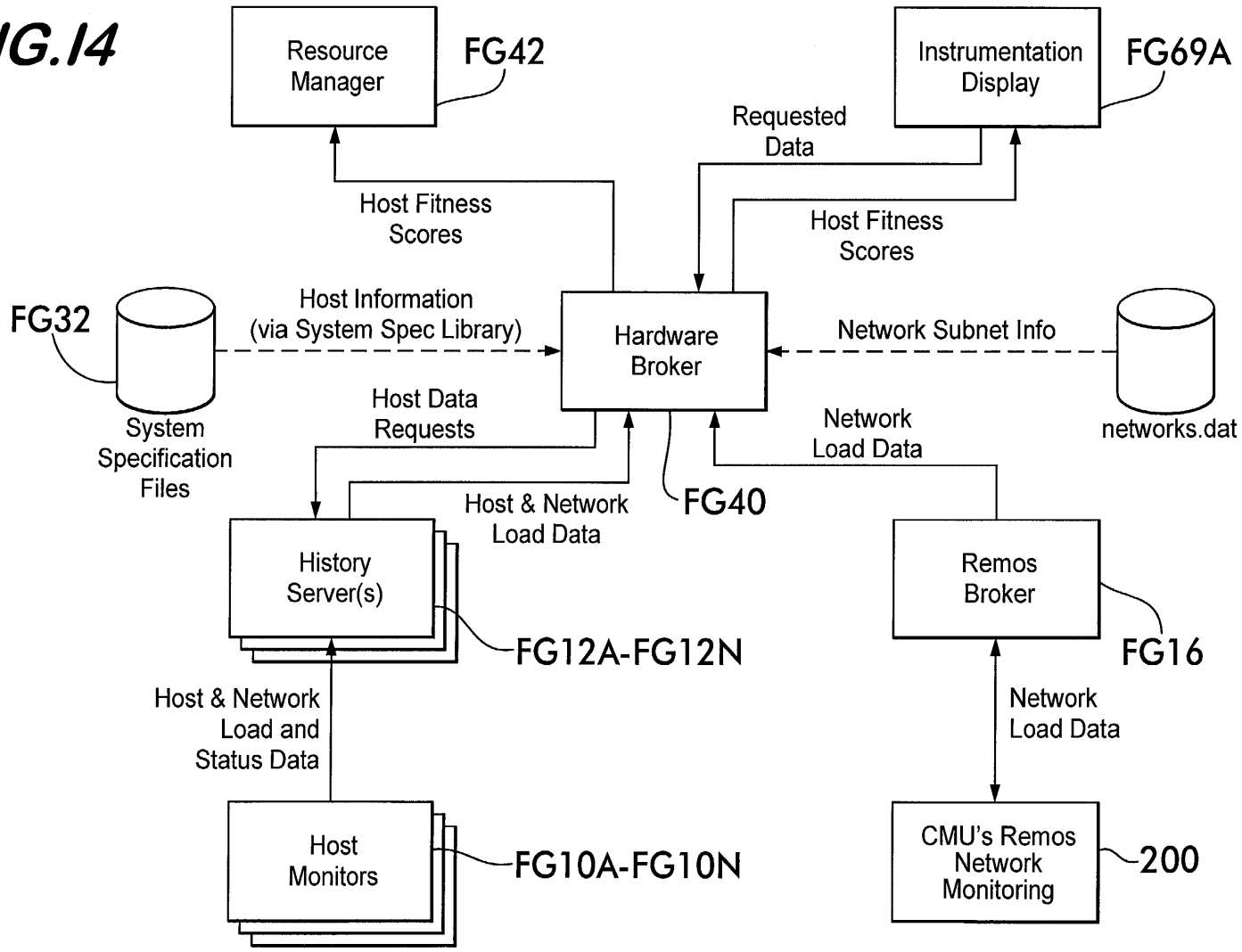
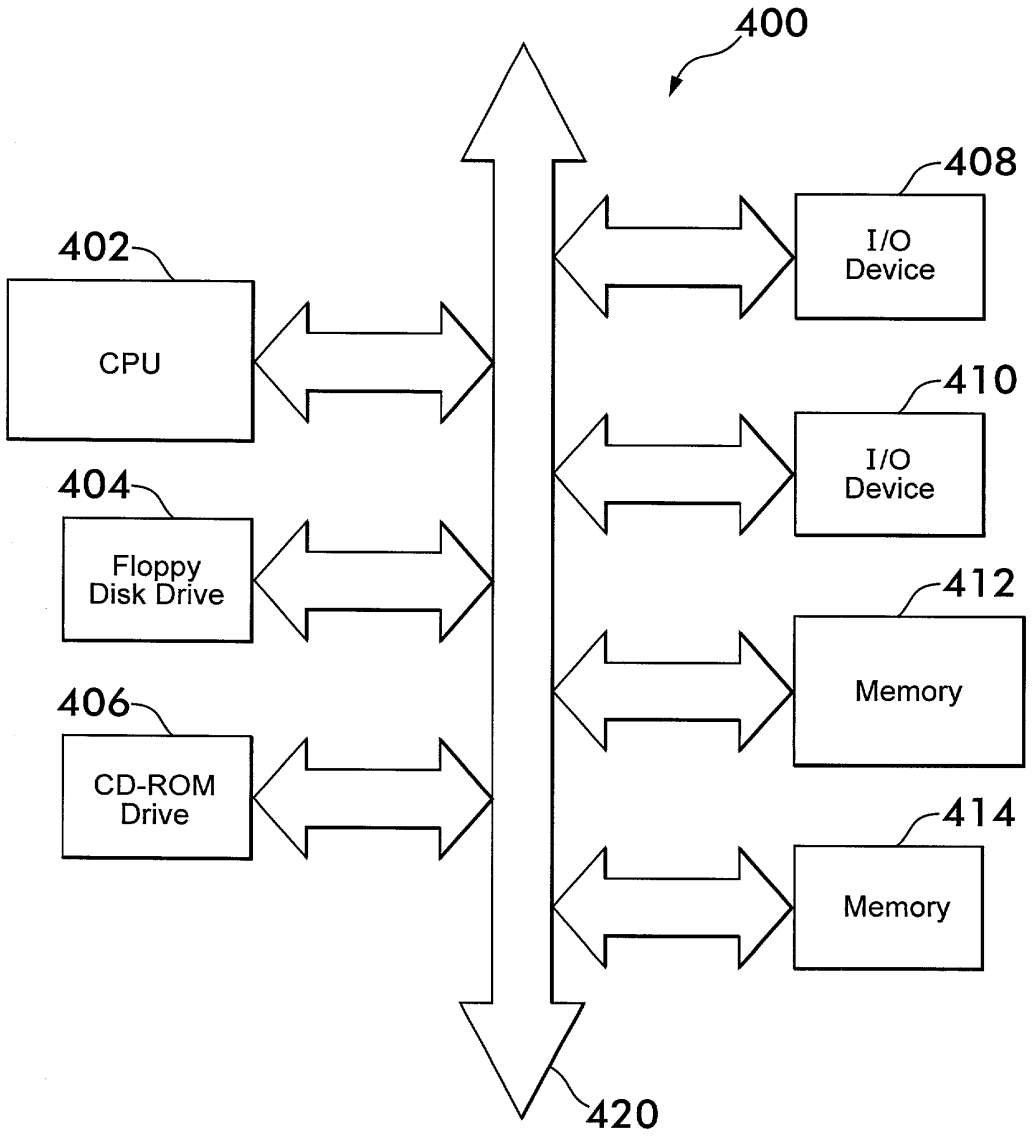


FIG.15



TESLA-EX1002-0410

Navy Case No. 83017

DECLARATION and POWER OF ATTORNEY for PATENT APPLICATION

As a below named inventor, I hereby declare that:
My residence, post office address and citizenship are as stated below next to my name,
I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled **RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR**, the specification of which (check one) ☐ is attached hereto ☑ was filed on 5/24/01 as Application Serial No. 09/864,825 and was amended on N/A.

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56(a).

I hereby claim foreign priority benefits under Title 35, United States Code, § 119 (a)-(d) or § 365(b) of any foreign application (s) for patent or inventor's certificate, or § 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application for patent or inventor's certificate, or of any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date:	Priority Not Claimed	Certified Copy Attached? Yes	No

OIPE JC139 SEP 19 2001 PATENT & TRADEMARK OFFICE

I hereby claim the benefit under Title 35, United States Code § 119(e) of any United States provisional application(s) listed below:

Application Number(s) **60/207,891** Filing Date (MM/DD/YYYY) **05/25/2000**

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s), or §365© of any PCT international application designating the United States of America listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56(a) which became available between the filing date of the prior application and the national or PCT international filing date of this application:

	U.S. Parent Application Number	PCT Parent Number	Parent Filing Date (MM/DD/YYYY)	Parent Patent Number (If applicable)

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith, (list name and registration number), and hereby certify that the Government of the United States has the irrevocable right to prosecute this application:

SEND CORRESPONDENCE TO: James B. Bechtel, Esq.
NSWCDD (CD222)
Dahlgren, VA 22448-5100

DIRECT TELEPHONE CALLS TO: James B. Bechtel, Esq.
Reg. No. 29,890
(540)653-8061

I hereby declare that all statements made herein of my knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of sole or first inventor Paul V. Werme

Inventor's signature Paul V. Werme Date 8/9/01
Residence 6172 Potomac Drive, King George, VA 22485
Citizenship United States Post Office Address Same as above

Full name of second inventor, if amy Michael W. Masters

Inventor's signature Michael W. Masters Date 8/9/01
Residence 1019 Hillcrest Terrace, Fredericksburg, VA 22405
Citizenship United States Post Office Address Same as above

Full name of third joint inventor, if any Larry A. Fontenot

Third Inventor's signature Larry A. Fontenot Date 8/23/01
Residence 12174 Canterbury Court, King George, VA 22485
Citizenship United States Post Office Address Same as above

Full name of fourth joint inventor, if any Lonnie R. Welch

Fourth Inventor's signature Lonnie R Welch Date 8/9/01
Residence 8007 Vore Ridge Road, Athens, OH 45701
Citizenship United States Post Office Address Same as above

Full name of fifth joint inventor, if any

Fifth Inventor's signature Date
Residence
Citizenship Post Office Address

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

APPLICATION NUMBER	FILING/RECEIPT DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NUMBER
09/864,825	05/24/2001	Paul V. Werme	NAVY CASE # 83017

CONFIRMATION NO. 2372

23501
JAMES B BECHTEL, ESQ
NAVAL SURFACE WARFARE CENTER CODE CD222
17320 DAHLGREN ROAD, BLDG. 183, RM. 015
DAHLGREN, VA 224485100

FORMALITIES LETTER

OC000000006330630

Date Mailed: 07/23/2001

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items and pay any fees required below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The oath or declaration is unsigned.
- To avoid abandonment, a late filing fee or oath or declaration surcharge as set forth in 37 CFR 1.16(e) of $130 for a non-small entity, must be submitted with the missing items identified in this letter.
- **The balance due by applicant is $ 130.**

The application is informal since it does not comply with the regulations for the reason(s) indicated below.

The required item(s) identified below must be timely submitted to avoid abandonment:

- Substitute drawings in compliance with 37 CFR 1.84 because:
  - drawing sheets do not have the appropriate margin(s) (see 37 CFR 1.84(g)). Each sheet must include a top margin of at least 2.5 cm. (1 inch), a left side margin of at least 2.5 cm. (1 inch), a right side margin of at least 1.5 cm. (5/8 inch), and a bottom margin of at least 1.0 cm. (3/8 inch);

---

*A copy of this notice **MUST** be returned with the reply.*

Customer Service Center

Initial Patent Examination Division (703) 308-1202
PART 3 - OFFICE COPY

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

05/30/2001 EHAILU 00000018 500967 09864825
01 FC:101 710.00 CH

PTO-1556
(5/87)
*U.S. GPO: 2000-468-987/39595

Application or Docket Number

NAVY #23917

PATENT APPLICATION FEE DETERMINATION RECORD

Effective October 1, 2000

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	1	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	1 minus 20=	* 0
INDEPENDENT CLAIMS	1 minus 3 =	* 0
MULTIPLE DEPENDENT CLAIM PRESENT		☐

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
BASIC FEE	355.00	OR	BASIC FEE	710.00
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL		OR	TOTAL	710

CLAIMS AS AMENDED - PART II

AMENDMENT A	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

CLAIMS ONLY

SERIAL NO. 09864825 | FILING DATE 05-24-01

APPLICANT(S)

CLAIMS

	AS FILED		AFTER 1st AMENDMENT		AFTER 2nd AMENDMENT	
	IND.	DEP.	IND.	DEP.	IND.	DEP.
1	1					
2	1					
3		1				
4		1				
5		1				
6		1				
7		1				
8	1					
9		1				
10		1				
11		1				
12		1				
13		1				
14		1				
15		1				
16	1					
17		1				
18		1				
19		1				
20		1				
21		1				
22		1				
23		1				
24		1				
25		1				
26	1					
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
48						
49						
50						
TOTAL IND.	5	⬇		⬇		⬇
TOTAL DEP.	21					
TOTAL CLAIMS	26					

	*		*		*	
	IND.	DEP.	IND.	DEP.	IND.	DEP.
51						
52						
53						
54						
55						
56						
57						
58						
59						
60						
61						
62						
63						
64						
65						
66						
67						
68						
69						
70						
71						
72						
73						
74						
75						
76						
77						
78						
79						
80						
81						
82						
83						
84						
85						
86						
87						
88						
89						
90						
91						
92						
93						
94						
95						
96						
97						
98						
99						
100						
TOTAL IND.		⬇		⬇		⬇
TOTAL DEP.						
TOTAL CLAIMS						

* MAY BE USED FOR ADDITIONAL CLAIMS OR ADMENDMENTS

FORM PTO-2022 (1-98)

U.S.DEPARTMENT OF COMMERCE
Patent and Trademark Office

*U.S. GPO: 1998-443-593/89152

ARTIFACT SHEET

Enter artifact number below. Artifact number is application number + artifact type code (see list below) + sequential letter (A, B, C ...). The first artifact folder for an artifact type receives the letter A, the second B, etc..
Examples: 59123456PA, 59123456PB, 59123456ZA, 59123456ZB

09864825 4A

Indicate quantity of a single type of artifact received but not scanned. Create individual artifact folder/box and artifact number for each Artifact Type.

- [x] CD(s) containing:
  - computer program listing — Doc Code: Computer — Artifact Type Code: P — [ ]
  - pages of specification and/or sequence listing and/or table — Doc Code: Artifact — Artifact Type Code: S — [ ]
  - content unspecified or combined — Doc Code: Artifact — Artifact Type Code: U — [x]
- [ ] Stapled Set(s) Color Documents or B/W Photographs — Doc Code: Artifact — Artifact Type Code: C
- [ ] Microfilm(s) — Doc Code: Artifact — Artifact Type Code: F
- [ ] Video tape(s) — Doc Code: Artifact — Artifact Type Code: V
- [ ] Model(s) — Doc Code: Artifact — Artifact Type Code: M
- [ ] Bound Document(s) — Doc Code: Artifact — Artifact Type Code: B
- [ ] Confidential Information Disclosure Statement or Other Documents marked Proprietary, Trade Secrets, Subject to Protective Order, Material Submitted under MPEP 724.02, etc. — Doc Code: Artifact — Artifact Type Code X
- [ ] Other, description: _____ — Doc Code: Artifact — Artifact Type Code: Z

March 8, 2004

Box Seq A

jc984 U.S. PTO
05/24/01

PATENT APPLICATION COVER SHEET

jc903 U.S. PTO
09/864825
05/24/01

NAVY CASE #83017

Honorable Commissioner of Patents and Applications
Washington, D.C. 20231

Sir:

Transmitted herewith for filing is the patent application of:

INVENTOR(S): Paul V. Werme et al.

FOR: RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR

Enclosed are:

☑ Seventeen sheet (s) of drawings.

☑ One hundred three sheet (s) of specification.

☐ An assignment of the invention to: Government of the United States of America.

☐ Information Disclosure Statement.

☑ Declaration and Power of Attorney.

☑ Other Appendix on CD-Rom.

☑ The filing fee has been calculated as shown below:

RECEIVED
MAY 29 2001
OIPE/JCMS

FOR:	No. Filed	No. Extra	OTHER THAN A SMALL ENTITY RATE	Fee
BASIC FEE				$710.00
TOTAL CLAIMS	1 - 20=	-0-	x $-0-	$-0-
INDEPENDENT CLAIMS	1 - 3 =	-0-	x $80.00	$-0-
MULTIPLE DEPENDENT CLAIMS PRESENTED			+$260.00	$ -0-
			TOTAL	$710.00

☑ Please charge Deposit Account 50-0967 in the amount of 710.00.

☑ The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account No. 50-0967.

Any additional fees required under 37 C.F.R.§ 1.16
Any patent application processing fees under 37 C.F.R. § 1.17

☑ The Commissioner is hereby authorized to charge payment of the following fees during the pendency of this application or credit any overpayment to Deposit Account No. 50-0967.

Any patent application processing fees under 37 C.F.R. § 1.17.
Any filing fees under 37 C.F.R. § 1.16 for presentation of extra claims.

May 24, 2001
Date

/James B. Bechtel/
James B. Bechtel, Esq.
Reg. No. 29,890
Customer No. 23501

Naval Surface Warfare Center
Office of Counsel (Patents) (Code CD222)
17320 Dahlgren Road
Dahlgren, VA 22448-5100
Phone: (540) 653-8061
Fax: (540) 653-7816

CERTIFICATE OF MAILING VIA EXPRESS MAIL

I hereby certify that this paper (along with any referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as to be sent by Express Mail in an envelope addressed to the Assistant Commissioner for Patents, Washington, DC 20231.

Date: _____

_____
Beverly D. Cook

23501
PATENT TRADEMARK OFFICE

RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR

STATEMENT OF GOVERNMENT INTEREST

The invention described herein was made in the performance of official duties by employees of the Department of the Navy or by researchers under contract to an agency of the United States government and, thus, the invention disclosed herein may be manufactured, used, licensed by or for the Government for governmental purposes without the payment of any royalties thereon.

BACKGROUND OF THE INVENTION

The present invention relates generally to resource management systems by which networked computers cooperate in performing at least one task too complex for a single computer to perform. More specifically, the present invention relates to a resource management system which dynamically and remotely controls networked computers to thereby permit them to cooperate in performing tasks that are too complex for any single computer to perform. Advantageously, software programs for converting a general purpose computer network into a resource managed network are also disclosed.

The instant application claims priority from Provisional Patent Application Serial No. 60/207,891, which was filed on May 25, 2000. The Provisional Patent Application is incorporated herein in its entirety by reference.

Resource Management consists of a set of cooperating computer programs that provides an ability to dynamically allocate computing tasks to a collection of networked computing resources (computer processors interconnected on a network) based on the

following measures:

- an application developer/user description of application computer program performance requirements;
- measured performance of each application programs;
- measured workload (CPU processing load, memory accesses, disk accesses) of each computer in the network; and
- measured inter-computer message communication traffic on the network.

Many attempts to form distributed systems and environments have been made in the past. For example, several companies and organizations have networked multiple computers to form a massively parallel supercomputer of sorts. One the best known of these efforts is SETI@home, which is organized by SETI (Search for Extraterrestrial Intelligence), a scientific effort aiming to determine if there is intelligent life out in the universe.

Typically, the search means the search of billions of radio frequencies that flood the universe in the hopes of finding another civilization that might be transmitting a radio signal. Most of the SETI programs in existence today, including those at UC Berkeley, build large computers that analyze that data from the telescope in real time. None of these computers look very deeply at the data for weak signals nor do they look for a large class of signal types. The reason for this is because they are limited by the amount of computer power available for data analysis. To extract the weakest signals, a great amount of computer power is necessary. It would take a monstrous supercomputer to get the job done. Moreover, SETI programs could never afford to build or buy that computing power. Thus, rather than use a huge computer to do the job, the SETI team developed software to use thousands of small computers, all working simultaneously on different parts of the analysis, to run the search routine. This is accomplished with a screen saver that can retrieve a data block over the internet, analyze that data, and then report the results back to SETI.

Several commercial companies are developing and implementing similar capabilities. Moreover, several companies, most notably IBM, have developed networks where each networked desktop computer becomes a parallel processor in a distributed computer system when the desktop computer is otherwise idle.

It will be appreciated that these approaches to computing in a distributed environment do not provide a system that is both flexible and adaptive (or at least easily adapted) to changes in system configuration, performance bottlenecks, survivability requirements, scalability, etc.

What is needed is a Resource Management Architecture which permits flexible control, i.e., allowing autonomous start up and shut down of application copies on host machines to accommodate changes in data processing requirements. What is also needed is functionality included in the Resource Management Architecture which permits the Resource Management Architecture to determine the near-optimal alignment of host and application resources in the distributed environment. It would be desirable to have a user-friendly technique with which to specify quality of service (QoS) requirements for each host, each application, and the network in which the hosts are connected. What is also needed is instrumentation to ensure that the specified QoS goals are being met.

SUMMARY OF THE INVENTION

Based on the above and foregoing, it can be appreciated that there presently exists a need in the art for a Resource Management Architecture, which overcomes the above-described deficiencies. The present invention was motivated by a desire to overcome the drawbacks and shortcomings of the presently available technology, and thereby fulfill this need in the art.

According to one aspect, the present invention provides, in a distributed environment

comprised of hosts instantiating copies of a scalable application, a resource management device generating signals which start up, shutdown or move a selected one of the copies responsive to first information regarding performance of all copies of the scalable application and second information regarding performance of the hosts.

BRIEF DESCRIPTION OF THE DRAWINGS

These and various other features and aspects of the present invention will be readily understood with reference to the following detailed description taken in conjunction with the accompanying drawings, in which like or similar numbers are used throughout, and in which:

Fig. 1 is a high-level block diagram of hardware and software components implemented in the Resource Management System according to the present invention;

Fig. 2 is a functional block diagram of the Resource Management Architecture according to the present invention;

Fig. 3 is a functional block diagram illustrating functional elements included in the system specification library (SSL) implementation of the Resource Management System according to the present invention;

Fig. 4 is a block diagram illustrating one technique for implementing the Resource (Application) Control functional group FG5 in Fig. 2 using discrete software components;

Fig. 5 is a screen capture of a program control display FG54 generated by the software components illustrated in Fig. 4;

Fig. 6 is a screen capture of a host display generated by the Resource Management Architecture according to the present invention;

Fig. 7 is a screen capture of performance data regarding several of the hosts A - N included in Fig. 6;

Fig. 8 is a screen capture of a path display generated by the Resource Management Architecture according to the present invention;

Fig. 9 is a screen capture of the Resource Management Decision Review Display,

which provides a summary of allocation and reallocation actions taken by the Resource Manager;

Figs. 10 and 11 are screen captures illustrating alternative, user-configurable displays generated from received data via standardized message formats and open interfaces;

Fig. 12 is a screen capture of an exemplary version of the Readiness Display FG66 according to the present invention;

Figs. 13A, 13B, and 13C are block diagrams which are useful in explaining various operational and functional aspects of the Resource Management Architecture according to the present invention; and

Fig. 14 is a high-level block diagram illustrating connectivity and data flow between the Hardware Broker and the other Resource Management and Resource Management-related functional elements in the Resource Management Architecture; and

Fig. 15 is a high-level block diagram of a CPU-based general computer which can act as a host in the Resource Management Architecture according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The Resource Management Architecture, which was and is being developed by the Naval Surface Warfare Center - Dahlgren Division (NSWCDD), provides capabilities for monitoring hosts, networks, and applications within a distributed computing environment. Moreover, the Resource Management Architecture provides the capability of dynamically allocating, and reallocating, applications to hosts as needed in order to maintain user-specified system performance goals. Advantageously, the Resource Management architecture provides functionality for determining both how each component within the distributed environment is performing and what options are available for attempting to correct deficient performance, determining the proper actions that should be taken, and enacting the determined course of action. In addition to these capabilities, the architecture also allows for operator control over creating and loading pre-defined static, dynamic, or

combined static and dynamic system and/or host configurations. One particularly desirable feature of the Resource Management Architecture is that it provides capabilities for monitoring system performance along with the ability to dynamically allocate and reallocate system resources as required.

Before addressing the various features and aspects of the present invention, it would be useful to establish both terminology and the conventions that the instant application will follow throughout. In terms of terminology, a glossary section is presented below. In terms of conventions, this application includes information such as source code listing in an Appendix section. Since the source code itself is hundreds of pages, the appendix section is divided into attached pages, e.g., Attached Appendix A, and an optical disk section, e.g., CD-Appendix N. Thus, while the appendices are listed in order, the reader must look to the signaling language to determine whether any particular appendix is actually provided in printed form.

API	**API (application programming interface)** A set of subroutines or functions that a program, or *application*, can call to invoke sone functionality contained in another software or hardware component. The Windows API consists of more than 1,000 functions that programs written in C, C++, Pascal, and other languages can call to create windows, open files, and perform other essential tasks. An application that wants to display an on-screen message can call Windows' *MessageBox* API function, for example.
BNF	Acronym for `Backus Normal Form' ( often incorrectly expanded as `Backus -Naur Form'), a metasyntactic notation used to specify the syntax of programming languages , command sets, and the like. Widely used for language descriptions but seldom documented anywhere , so that it must usually be learned by osmosis from other hackers.

DAEMON	A background process on a host or Web server (normally in a UNIX environment), waiting to perform tasks. Well-known examples of daemons are sendmail and HTTP daemon.
FUNCTION	A capability available on a host due to the presence of software (e.g., a program), a software module (e.g., an API), etc.
GLOBUS	Wide area network (WAN) enterprise management and control capability developed under DARPA sponsorship by USC/ISI.
HOST	A device including a central processor controlled by an operating system.
ICMP	Internet Control Message Protocol - ICMP is an extension to the Internet Protocol. It allows for the generation of error messages, test packets and informational messages related to IP. It is defined in STD 5, RFC 792.
JEWEL	An open-source instrumentation package produced by the German National Research Center for Computer Science
NFS	Network File System - A protocol developed by Sun Microsystems, and defined in RFC 1094 , which allows a computer system to access files over a network as if they were on its local disks. This protocol has been incorporated in products by more than two hundred companies, and is now a *de facto* Internet standard.
QoS	Quality of Service

REMOS	Remos (REsource MOnitoring System)is a network bandwidth and topology monitoring system developed under DARPA sponsorship by CMU. Remos allows network-aware applications to obtain relevant information about their execution environment. The major challenges in defining a uniform interface are network heterogeneity, diversity in traffic requirements, variability of the information, and resource sharing in the network. Remos provides an API that addresses these issue by striking a compromise between accuracy (the information provided is best-effort, but includes statistical information if available) and efficiency (providing a query-based interface, so applications incur overhead only when they acquire information). Remos supports two classes of queries. "Flow queries" provide a portable way to describe a communication step to the Remos implementation, which uses its platform-dependent knowledge to return to the user the capacity of the network to meet this request. "Topology queries" reverse the process, with the Remos implementation providing a portable description of the network's behavior to the application.
SNMP	Simple Network Management Protocol Internet standard protocol defined in STD 15, RFC 1157; developed to manage nodes, e.g., hubs and switches, on an IP network.

An exemplary system for implementing the Resource Management Architecture according to the present invention is illustrate in Fig. 1, which includes a plurality of Host computers A, B, ..., N operatively connected to one another and Resource Management hardware RM via a Network 1. It will be appreciated that the network 100 advantageously can be any known network, e.g., a local area network (LAN) or a wide area network (WAN). It will also be appreciated that the hardware RM need not be a discrete piece of

equipment; the hardware RM advantageously can be distributed across multiple platforms, as discussed in detail below. In addressing the functional elements and applications in the distributed environment, it will be appreciated that hosts A-N each can instantiate applications 1-M. Thus, when all applications are being addressed, these applications will be denoted as A1-NM.

Still referring to Fig. 1, each of the hosts A, B, etc., preferably is controlled by an operating system (OSA, OSB, etc.), which permits Host A, for example, to execute applications A1 - AN, as well as an instrumentation daemon IDA, a Program Control (PC) agent PCA, and a Host Monitor HMA. It should be noted that instrumentation daemon IDA, PC agent PCA, and Host Monitor HMA are integral to the Resource Management Architecture while the operating system OSA and applications A1 - AN are well known to one of ordinary skill in the art.

In Fig. 1, the Resource Management Architecture RM advantageously includes an instrument collector 10 receiving data from all of the instrumentation daemons (IDA - IDN) and providing data to instrument correlator(s) 20, which, in turn, provide correlation data to corresponding quality of service (QoS) managers 30. Resource Management Architecture RM also receives data from host monitors HMA- HMN at history servers 40, which maintain status and performance histories on each of the hosts A - N and provide selected information to host load analyzer 50. Analyzer 50 advantageously determines the host and network loads for both hosts A-N and their connecting network 100 and provides that information to Resource Manager 60, which is the primary decision making component of the Resource Management Architecture. It will be appreciated that Resource Manager 60 also receives information from the QoS managers 30 and exchanges information with program controller 70. Program controller 70 sends startup and shutdown orders to the Program Control Agents based on operator or Resource Manager-initiated orders. It will be appreciated that the operator-initiated orders are received via the one of the program control displays 80.

As will be discussed in greater detail below, the Resource Manager 60 is the primary decision-making component of the Resource Management Architecture. The Resource Manager 60 is responsible for determining:

- how to respond to host and application failures;
- where (i.e., which of hosts A -N) to place new applications;
- which applications to start up in response to the detection of a new host (host N+1);
- how to resolve application dependencies;
- what applications should be started, stopped, or moved in response to application system priority changes; and
- based on recommendations from the QoS Managers, when and where scalable application should be started or stopped.

Before leaving Fig. 1, is should be noted that the functions, e.g., instantiated programs or software program modules, in the Resource Management Architecture advantageously can be distributed across multiple platforms, e.g., multiple host (which may or may not be the illustrated Hosts A -N).

The major functional groups of the Resource Management Architecture according to the present invention are illustrated in Fig. 2. The functions illustrated as solid boxes are components of the Resource Management Architecture and are fully described below; the functions denoted by diagonal striping denote third-party software which has been integrated with the Resource Management Architecture but does not provide core functionality. Thus, the latter functions will be described only to the extent necessary to provide integration details. Moreover, it will be appreciated that the functions and functionality of the Resource Management Architecture according to the present invention are interconnected to one another via middleware, which provides message passing interfaces between substantially all of the Resource Management functions. This middleware package, RMComms, is fully

described below.

The major functional groups provided by the Resource Management architecture in an exemplary embodiment of the present invention are illustrated in Fig. 2. A summary of the functions provided by the Resource Management Architecture is available in Attached Appendix A. These functions, taken together, provide an integrated capability for monitoring and control of a distributed computing environment. In addition, many of the functions (and functional groups) within the Resource Management Architecture can also be run in a non-integrated configuration, thus providing subsets of the integrated Resource Management capabilities.

These function(al) groups illustrated in Fig. 2 include:

FG1 - Host and Network Monitoring. This function group consists of software which monitors the host and network resources within the distributed environment. The function group collects extensive run-time information on host and network configuration, statuses, and performance. Run-time capabilities for discovering new hosts that have been started and for determining that existing hosts have gone down are also provided. Distribution of current and historical status and performance data to other components of the Resource Management Architecture is also provided. A more detailed discussion is provided below.

FG2 - Application-Level Instrumentation. The instrumentation function group provides general-purpose application event reporting and event correlation capabilities. Capabilities are provided for collecting and correlating application-provided data such as application statuses, states, performance, and internally detected errors. Low-overhead (API) libraries are provided for applications to use in sending out key internal event and performance data. This application data is forwarded to other

components of the instrumentation subsystem which collect data from applications on hosts throughout the distributed environment. The system also provides grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other Resource Management components.

FG3 - System Specifications. A specification language has been developed which allows the user to specify:

1) application software system structure, capabilities, dependencies, and requirements; and

2) hardware system (computer and network) structure, capabilities, and configuration.

Specification files, based on this specification language, are created by the user and provide the model of the software and hardware components of the distributed computing environment which is used by other Resource Management functions. The specification information is accessed by other Resource Management functions by linking in a specification parser library and making library calls to read in the files and convert them to an internal object model. Specific specification data items can then be retrieved via an object-oriented API. See the discussion below.

FG4 - Resource Allocation Decision-Making. This subsystem provides the reasoning and decision-making capabilities of the Resource Management architecture. The components of this subsystem use information from other subsystems in order to determine the health and state of the distributed environment and the options that are available for attempting to recover from faults or unacceptable performance. The functions in this particular functional group make decisions regarding:

1) where new applications should be started;

2) whether and where failed applications should be restarted;

3) based on application inter-dependencies, whether and where additional applications should to be started prior to starting a particular application;

4) whether applications are meeting performance requirements and whether and where an application can be scaled up or moved to when it is necessary to improve performance;

5) whether scalable applications are performing well within performance requirements and can be scaled down and which copy should be brought down; and

6) based on operator changes to application system priorities, whether and where new applications need to be started or whether and which existing applications need to be shut down.

FG5 - Application (Resource) Control. This subsystem provides application control (i.e., Program Control) capabilities which permit starting, stopping, and configuring applications on each of the hosts in the distributed environment. The subsystem provides both interactive operator control of the distributed environment as well as automatic control via configuration orders received from the Resource Allocation Decision-Making Subsystem (i.e., the Resource Manager component). The interactive controls allow an operator to create, load, save, and edit pre-defined system configurations (e.g., lists of applications that are to be run, with or without specific host mappings), determine the status and configuration of currently running programs, and start and stop any or all applications. Both static (operator-entered) mappings of applications to hosts and dynamic mappings of applications to hosts (where the Resource Allocation Decision-Making Subsystem will be queried to determine the proper mapping at run-time) can be defined. The subsystem also provides application fault detection capabilities which are triggered by the unexpected death of an application that was started by the subsystem. A basic host fault detection capability is also provided which is triggered based on failure to

receive heartbeat messages from subsystem components running on a particular host.

FG6 - Displays. The display subsystem provides capabilities for visualizing the status, performance, and health of the hosts, networks, and applications in the distributed environment. Capabilities are also provided for visualizing the status, performance, and health of the Resource Management components themselves.

As mentioned above, the RMComms middleware package provides the internal message passing interfaces between substantially all of the Resource Management functions both within each functional group and between the various functional groups. The middleware provides for automatic location-transparent many-to-many client-server connections. Low-overhead, reliable message passing capabilities are provided. Registration of message handler callback functions for specified requested message types is provided with the message handler functions being invoked when messages arrive. Registration of connection status callback functions which are invoked when either new connections are made or existing connections are broken is also provided. The middleware package also allows for multiple client and server objects to be instantiated in the same application, is thread-safe, and provides an easy-to-use object-oriented API through which all capabilities are accessed.

A detailed overview of each functional group and each function instantiated within each of the function groups FG1 - FG6 of the exemplary embodiment of the Resource Management Architecture illustrated in Fig. 2, including the capabilities provided by the functional group or function, will now be described in greater detail. The discussion below also includes an overview of the information flow between function blocks within the same functional group and between function blocks in separate functional groups.

FG1 - Host and Network Monitoring Functional Group

Ex.1002 / Page 433 of 541
TESLA, INC.

Functional group FG1 provides extensive monitoring capabilities at the host and network levels. The information monitored includes statuses, configuration information, performance metrics, and detected fault conditions. By monitoring the individual hosts and network components within the distributed environment, the functional group FG1 determines:

- Accurate State and Performance Information, primarily by gathering the level of information necessary for accurately determining the state and health of each machine and network component.
- Distribution of Current Data to Resource Management Components by providing current performance and status information, either periodically or on request.
- Distribution of Historical Data to Resource Management Components, thus providing historical performance and status information on request.

It will be appreciated that the functional group FG1 makes these determinations by (or while) providing:

- Common Monitored Data Set and Formats, which permits functional group FG1 to gather the same set of statuses and statistics in the same formats for each host regardless of machine architecture or operating system.
- Minimally-Intrusive Data Collection Mechanisms, which permits functional group FG1 to gather the information in as non-intrusive a manner as possible (in terms of CPU utilization, network bandwidth utilization, etc...).
- Near Real-Time Data Collection Mechanisms, which permits functional group FG1 to gather the information in as timely a manner as possible.

The Host and Network functional group FG1 includes the four functions set forth below:

1) **Host Monitors** FG10A - FG10N, which reside on each respective machine in the distributed environment and collect extensive operating system-level

data for each host A - N.

2) **History Servers** FG12A - FG12N, which collect data from the Host Monitors FG10A - FG10N, respectively, maintain status and performance histories on each host A - N in the distributed environment, i.e., in the Resource Management Architecture, and provide this information to displays and other functions with the Resource Management Architecture.

3) **Host Discovery Function** FG14, which uses Simple Network Management Protocol (SNMP) calls and ping Internet Control Message Protocol (ICMP) calls to determine when new hosts, E.g., host N+1, come on-line and if an existing host, e.g., host K, goes down.

4) **Remos Network Data Broker** Function FG16, which collects information on network link bandwidths from the SNMP-based Remos tool (developed by Carnegie Mellon University) and passes this information to the Host Load Analyzer function of the Resource Allocation Decision-Making functional group FG4, both of which are discussed in greater detail below.

Host monitors FG10A- FG10N, which monitor the status and performance of hosts A -N, respectively, are instantiated on each host machine within the distributed environment. Host Monitors FG10A - FG10N employ operating system-level mechanisms to retrieve status, configuration, and performance information on each host A - N. The information retrieved includes:

1) operating system version and machine configuration;
2) CPU configuration, status, and utilization;
3) memory configuration and usage;
4) network configuration, status, and utilization;
5) filesystem configuration, status, and utilization; and
6) process statuses including CPU, memory, network, and filesystem utilization for each process.

While Host Monitors FG10A - FG10N are primarily responsible for monitoring the status of a particular host, they also provide information on network load as seen by that particular host. In the same manner, the Host Monitors FG10A - FG10N also provide information and statistics concerning any remotely mounted filesystems, e.g., Network File System (NFS).

The information that the Host Monitors FG10A - FG10N collect advantageously can be formatted into operating system-independent message formats. These message formats provide a pseudo-standardized set of state, status, and performance information which is useful to other components of the Resource Management Architecture, i.e., other components do not have to be aware of or deal with the minor differences between data formats and semantics. It will be appreciated that since not all the state and performance data is available on every platform, in order to indicate which information is available, a group of flags are set in the host configuration message indicating whether specific data items are valid on a particular platform.

History Servers FG12A - FG12N are responsible for collecting information from the Host Monitors FG10A - FG10N and maintaining histories on the statuses, statistics, and performance of each host A - N in the distributed environment. This information advantageously can be requested by other functions instantiated in the Resource Management Architecture. Preferably, the primary consumers of the status information obtained by the History Servers FG12A - FG12N are the Host Load Analyzer (Hardware Broker) component of the Resource Allocation Decision-Making functional group FG4, the Host Display FG62A - FG62N and the Path Display FG64 of the Displays functional group FG6. The Host Load Analyzer FG40 receives information on host configuration and loads (primarily CPU, memory, and network data) from History Servers FG12A - FG12N and employs this information to assign host fitness scores. Each Host Display, e.g., FG62A, receives and displays current status information on one of the hosts A - N, including process status information, and network connectivity information. Each Host Display can also

Ex.1002 / Page 436 of 541
TESLA, INC.

request that a respective one of the History Servers FG12A - FG12N provide CPU load information, network load information, paging activity data, and memory utilization information, which is used to drive line graph charts for specific selected hosts.

It will be appreciated that History Servers FG12A - FG12N are designed so that multiple copies can be run simultaneously. Each of the History Servers FG12A - FG12N advantageously can be configured to either monitor all Host Monitors or to monitor only a selected set of Host Monitors. It should be mentioned at this point that the History Servers FG12A - FG12N determine the list of hosts in the distributed environment that could potentially be monitored from the System Specification Library. In this manner, the History Servers advantageously can be used to provide survivability (by having multiple History Servers connected to each Host Monitor) and/or to perform load-sharing (with the History Servers FG12A - FG12N each monitoring only a subset of the Host Monitors). It will also be appreciated that the History Servers FG12A - FG12N can be configured to periodically record history data to disk. These disk files can then be used for off-line analysis of the Resource Management Architecture.

The Host Discovery function FG14 employs Perl scripts in making SNMP and ICMP ping calls. These calls are used to periodically scan each subnet and host address in the distributed environment in an attempt to determine whether there have been any host status changes. In an exemplary case, the list of hosts and subnets that are to be monitored is read in from a file; alternatively, this information can reside in and be read from the System Specification Library, which is discussed in greater detail below.

It should be mentioned that when a new host is first detected, the new host's operating system configuration is queried by the Host Discovery function FG14 via SNMP calls. Information on the newly discovered host and its operating system configuration is then sent to the Program Control function FG50 in application control functional group FG5.

Likewise, when a host fails to respond to multiple SNMP and ping queries, a message indicating that the host appears to have gone down is sent to the Program Control function FG50.

The final component of the Host and Network Monitoring functional group FG1 is the Remos Network Data Broker FG16, which receives information on network link bandwidth and network link bandwidth utilization from the SNMP-based Remos network monitoring tool mentioned above. The network information is accessed via the Remos application programming interface (API) library and is then sent on to the Host Load Analyzer (Hardware Broker) function FG40 of the Resource Allocation Decision-Making functional group FG4. The network information received from Remos consists of the maximum potential bandwidth and the current bandwidth utilization on specific host network links. As mentioned above, Remos network monitoring tool FG 16 is not a core component of the Resource Management Architecture; that being the case, no further details on either Remos or the Remos Network Data Broker are provided in the instant application.

FG2 - Application-Level Instrumentation Functional Group

The Instrumentation functional group FG2 advantageously provides general-purpose application event reporting and event correlation capabilities. The Instrumentation functional group permits instrumented application data to be easily accessible to other components of the Resource Management Architecture. The functional group provides capabilities for collecting and correlating application-provided data such as application statuses, states, performance, and internally detected errors. Low-overhead API's are provided that the applications can use for sending internal event and performance data to the instrumentation components. The instrumentation functional group FG2 can collect data from applications on hosts A - N throughout the distributed environment. The functional group also provides grammar-driven capabilities for correlating, combining, and reformatting application data

into higher-level metrics (composite events) for use by displays or other functional groups of the Resource Management Architecture.

The Instrumentation functional group provides:

- open API's and non-proprietary architecture
- near real-time monitoring support
- cross-language support: C, C++, Ada
- cross-platform support: Solaris, IRIX, Linux, etc...
- simple easy-to-use API's
- low-intrusive instrumentation interface
- instrumentation interface that does not significantly change the run-time behavior of the applications
- support for passing wide range of data types
- support for data marshalling / unmarshalling (system independent data formats)
- support for adding to or changing the information being instrumented without having to recompile portions of the architecture unaffected by the changes (preferably, no recompilation should be necessary expect for recompilation of the app being instrumented and any evaluation logic or displays that have been affected by the changes)
- scalable architecture (100+ hosts / 20+ apps per host / 5+ threads per app)
- ability for the architecture to perform auto-configuration as required
- ability to run multiple tests, multiple displays and multiple data logging components simultaneously
- ability to abstract away the underlying connectivity/communications between infrastructure components.
- ability for instrumentation infrastructure to be brought up and down while the application is running

- ability to easily build and configure new displays and data logging components (interactive configuration is preferable)
- ability to easily build and configure new performance and data correlation components (interactive configuration is preferable)
- backwards compatibility with existing Jewel Instrumentation displays (protect investments in existing display capabilities)
- backwards compatibility with existing Jewel Instrumentation function calls (provide ease of transition / backfit)

As illustrated in Fig. 2, the Instrumentation functional group FG2 includes the components enumerated below. In addition, Instrumentation APIs and Jewel Instrumentation will be addressed along with the Instrumentation functional group, i.e., the Instrumentation functional group includes:

1) **Instrumentation API Libraries** FG20 are linked with the applications and provide the function call interfaces by which these applications send instrumentation data.

2) **Instrumentation Daemons** FG22A - FG22N reside on each host in the distributed environment and are responsible for reading instrumentation data sent out by the applications, reformatting the data into instrumentation event messages and sending the messages to the Instrumentation Collectors.

3) **Instrumentation Collectors** FG24A -FG24N connect to the Instrumentation Daemons FG22A - FG22N on each host and receive instrumentation messages from host A - N. The Collectors forward received messages to the Instrumentation Correlators FG26A - FG26N and Instrumentation Brokers FG28A - FG28N.

4) **Instrumentation Correlators** FG26A - FG26N receive instrumentation messages from the Instrumentation Collectors FG24A - FG24N and provide grammar-driven capabilities for correlating, combining, and reformatting

application data into higher-level metrics (composite events) for use by displays or other functions of the Resource Management Architecture.

5) **Instrumentation Brokers** FG28A - FG28N receive instrumentation messages from the Instrumentation Collectors and perform task-specific reformatting and data manipulation for driving displays or other Resource Management components.

6) **Jewel Instrumentation Broker (QoS Monitor)** FG29 (a legacy component) receives instrumentation data from either the open source Jewel instrumentation package or from the Instrumentation Collectors. The QoS Monitor FG29 performs task-specific message reformatting and data manipulation for driving displays and the QoS Managers FG44A - FG44N.

The applications, e.g., A1-AN, link in the Instrumentation API Library FG20 and make API calls to construct and send out instrumentation event messages. Three separate APIs are provided for use by the applications:

1) a printf()-style API which allows the code to format, build, and send instrumentation data with a single function call;

2) a buffer-construction-style API where the multiple function calls are made to construct the instrumentation buffer iteratively, one data element per call; and

3) a Jewel function call API based on the existing API provided by the Jewel instrumentation package (an open-source package produced by the German National Research Center for Computer Science).

The first two APIs are the preferred programming interfaces and take advantage of several key instrumentation features while the Jewel API is provided solely for backwards compatibility with existing instrumented application code and is implemented as a set of wrappers around the printf()-style API. All three APIs are supported for C and C++. ADA bindings have also been produced for the buffer-construction-style API and the Jewel

function call API.

Preferably, the instrumented data is sent from the application to one of the Instrumentation Daemons FG22A -FG22N on a respective one of the hosts A - N where the application is running. The currently preferred mechanism for data transfer is via UNIX FIFO (first in - first out) IPC (inter-process communication) mechanisms. It will be appreciated that the FIFO mechanism was chosen based on reliability, low overhead, and ease of implementation. Alternative data passing mechanisms including shared message queues are considered to be within the scope of the present invention.

As mentioned above, an Instrumentation Daemon resides on each host in the distributed environment. The Instrumentation Daemon is interrupted whenever new data is written to the FIFO. The Instrumentation Daemon reads the data from the FIFO, reformats the data into the standard internal Instrumentation message format (discussed below), and sends the data to each of the respective Instrumentation Collectors FG24A -FG24N that are currently active. Alternatively, an event request filtering mechanism can be implemented so that specific event messages will only be sent to those ones of the Instrumentation Collectors FG24A -FG24N that have requested the message.

The standard instrumentation message format includes a header, a format string describing the application-provided data contained in the message, and the actual data values.
The message components are illustrated in Attached Appendix B.

The Instrumentation Collectors FG24A- FG24N receive instrumentation messages from the Instrumentation Daemons FG22A - FG22N on each host A - N, respectively, in the distributed environment. Currently, the Instrumentation Collectors FG24A- FG24N send every instrumentation message to all Instrumentation Brokers FG29A- FG29N and

Instrumentation Correlators (Brokers) FG26A- FG26N that have connected to the Instrumentation Collectors FG24A- FG24N. The Instrumentation Collectors FG24A-FG24N serve as a pass-through server for instrumentation messages. The Instrumentation Collectors do support architecture scalability in the sense that without the Instrumentation Collectors, the Instrumentation Broker FG29 and Instrumentation Correlators FG26A-FG26N would need to maintain connections to the Instrumentation Daemons FG22A-FG22N on every host. As discussed above, an event request filtering mechanism advantageously can be implemented so that specific event messages will only be sent to those Instrumentation Brokers / Instrumentation Correlators that have requested the message.

Preferably, the Instrumentation Correlators FG26A- FG26N provide grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other components of the Resource Management Architecture. Each Correlator reads in a user-specified correlation grammar file which is interpreted at run-time by the Correlator's instrumentation correlation engine.

The Instrumentation Brokers FG28A- FG28N are task-specific applications built around a common code package. The Instrumentation Brokers FG28A- FG28N receive instrumentation messages from the Instrumentation Collectors FG24A- FG24N, filter all received instrumentation messages to find the messages of interest, and perform task-specific message data reformatting and manipulation for driving other components such as displays or other components of the Resource Management Architecture. This Instrumentation Broker approach permits instrumentation data sources to be quickly integrated for test, display, and debugging purposes.

It should be mentioned at this point that the Jewel Instrumentation Broker FG29 (hereafter referred to the QoS Monitor) is a legacy architecture component that served as a broker between the Jewel instrumentation package components and Resource Management

components and displays. The QoS Monitor FG29 was responsible for polling the Jewel Collector components to retrieve application event messages. These messages were then reformatted and used to drive several displays and the QoS Managers FG44A - FG44N. The Jewel instrumentation package has now been replaced in all applications, however the message reformatting capabilities of the QoS Monitor have been maintained so that several displays and the existing QoS Manager interface do not have to be upgraded immediately. The QoS Monitor component has been modified so that it receives instrumentation data from both Jewel and the Instrumentation Collectors.

FG3 - SYSTEM SPECIFICATIONS FUNCTIONAL GROUP

Still referring to Fig. 2, it should be noted that a System Specification Language has been developed which allows the user to specify both (1) software system structure, capabilities, dependencies, and requirements, and (2) hardware system (computer and network) structure, capabilities, and configuration. System Specification Files, generally denoted FG32, which are based on this specification language, are created by the user and provide a model of the software and hardware components of the distributed computing environment which is used by the Resource Management Architecture. The language grammar advantageously can capture the following information related to the distributed environment and the applications that can run within the distributed environment:

- Hardware and Operating Systems
  - Hardware Configuration
  - Network Configuration
  - Operating Systems and Version
- Software
  - Systems, Subsystems, Paths, Applications, Processes
  - Resource Requirements
  - QoS Requirements (Events)

- Survivability Requirements
- Data Flow Path Information: Structure and QoS Requirements

It will be appreciated that the System Specification Language allows for grouping hardware and software components into systems and subsystems in order to create a hierarchy of components. Each application system and subsystem can be assigned a priority which is used at run-time to determine the relative importance of applications running in the distributed environment.

At the application level, the hardware, operating system, and other host requirements for each application can be specified along with information describing how to start up, configure, and shutdown the application. This information can include:

a) environment variables that need to be set;

b) the working directory for running the application;

c) the path(s) and file name of the application;

d) command-line arguments that should be set, including arguments that need to be resolved at run-time (e.g., the hostname where another application is running, the current date, the current userid, a unique run-time identifier number, etc...);

e) whether the application needs to run in an xterm;

f) whether a script file or signal should be run to shutdown the application; and

g) which script or signal should be used.

In addition, startup and shutdown dependencies between applications can be specified. Moreover, application states can be defined based on received instrumentation data values, the length of time an application has been running, and/or the set of processes that are currently running. Furthermore, for each application A1 - NM, the survivability and scalability capabilities of the application can be specified. This latter information includes whether an application can be restarted if it fails, whether multiple copies of an application

can be run, what type of scalability the application supports (e.g., Primary-Shadow, Load-Sharing, etc...), and the minimum and maximum number of copies that can be run. Moreover, an estimate of the amount of CPU, memory, and network resources that the application will use at run-time, advantageously can be specified.

At the host level, the operating system and version, the hardware architecture, the host's network interface name, and the SPEC organization's SPECfp95 and SPECint95 ratings for the host can be specified. At the network level, router and switch configurations and bandwidths can also be specified.

Moreover, application data flow paths can be defined including a graph of the data flow between applications along with performance requirements tied to one of more of the applications within the path. It should be mentioned that these defined requirements are named and are tied at run-time to Instrumentation Event data provided by the Instrumentation Correlators FG26A- FG26N. Monitoring of the performance requirements is the responsibility of the QoS Manager components FG44A - FG44N, as discussed in greater detail below.

As noted above, the System Specification Language provides a hierarchical structure for defining software and hardware systems. The current structure is shown below:

- **Software Specifications**
  - Application
    - Security
    - Configuration
      - Hardware Requirements
      - Startup Info
        - Dynamic Arguments
      - Shutdown Info

Ex.1002 / Page 446 of 541
TESLA, INC.

- States
- Dependencies
- Initial Load Estimate
- QoS Info
  - Survivability
  - Scalability

- **Hardware Specifications**
  - Host Info
  - Network Info
    - LANs
    - Network Devices (Interconnects)
- **Path Specifications**
  - Data Flow Graph
  - Data Flow Info
  - QoS Requirements

The specification information is accessed by linking in a specification parser library FG34 and making library calls to read in the files and convert them to an internal object model, and by making object access method calls to retrieve specific data items. The specification library is written in C++ and has been ported to all of the development platforms in the testbed. The library is currently being used by most of the Resource Management components, including Program Control FG50, the Resource Manager FG42, the QoS Managers FG44A -FG44N, the Hardware Broker FG40, and the History Servers FG12A - FG12N.

It should be mentioned that the software used to construct the API library consists of (1) a parser file that defines the grammar (in BNF format),(2) a lexical analyzer file that defines the tokens of the language, and (3) a set of C++ System Specification classes for

storing the specification file information. The lexical analyzer file is compiled with the GNU flex (lex) utility and the parser file is compiled using the GNU bison (yacc) utility. The flex and bison utilities create C source files which are then compiled along with the C++ System Specification object storage classes to create the System Specification Library (SSL) FG34. This library is then linked with the Resource Management applications. An overview of this structure is provided in Fig. 3; a more detailed discussion of the various functions are provided below.

FG4 - RESOURCE ALLOCATION DECISION-MAKING FUNCTIONAL GROUP

As illustrated in Fig. 2, the Resource Allocation Decision-Making functional group provides the reasoning and decision-making capabilities of the Resource Management architecture. The functions associated with this functional group employ information (listed below) to (1) determine the state and health of the distributed environment (hosts, networks, and applications), and (2) determine what allocation and reallocation actions need to be taken. The information provided to functional group FG4 includes:

- System Specifications:
  - Host configuration and capabilities
  - Application capabilities
    - Survivability
    - Scalability
    - Potential hosts to run on
  - Application startup and shutdown dependencies
  - Application and path performance requirements
- Program Control:
  - Application statuses
  - Detected application faults
  - Detected host failures

- Detection of new host
- Operator initiated requests
  - Resolution of application startup or shutdown dependencies
  - Selection of application-to-host mappings
- History Servers:
  - Host statuses, configuration, and loads
  - Network link statuses and loads
- Remos Network Data Broker:
  - Network link statuses and loads
- Instrumentation Subsystem:
  - Application performance information
- Readiness Display:
  - Run-time changes to application system priorities

The subsystem components make decisions based on the following triggers and data sources:

- Based on requests from Program Control, determine where new applications should be started
- Based on indication of application failure from Program Control, determine whether and where the failed applications should be restarted
- Based on indication of host failure from Program Control (or indirectly from Host Discovery), determine whether and where the failed applications should be restarted
- Based on application inter-dependencies defined in the System Specification Files, determine whether and where additional applications should to be started (or shut down) prior to starting (or shutting down) a particular application
- Based on startup and shutdown dependency resolution requests from

Program Control, determine whether and where additional applications should to be started (or shut down) prior to starting (or shutting down) a particular application

- Based on application instrumentation data and performance requirements defined in the System Specification Files, determine whether applications are meeting performance requirements and whether an application can be scaled up or moved to attempt to improve performance
- Based on application instrumentation data and performance requirements defined in the System Specification Files, determine whether applications are performing well within performance requirements and can be scaled down
- Based on operator changes to application system priorities, determine whether and where new applications need to be started and/or determine whether and which existing applications need to be shutdown
- Based on indication that a new host is on-line (from Host Discovery via Program Control), issue startup orders to bring up a Program Control Agent, Host Monitor, and Instrumentation Daemon on the new host which will bring the host under Resource Management control

The Resource Allocation Decision-Making functional group implements one of the three discrete functions listed below:

1) **Resource Manager** FG 42 is the primary decision-making component of the Resource Management Architecture. Resource Manager FG42 is responsible for determining (1) how to respond to host and application failures, (2) where to place new applications, (3) which applications to start up in response to the detection of a new host, (4) how to resolve application dependencies, (5) what applications should be started, stopped, or moved in response to application system priority changes, and (6) based on recommendations from the QoS Managers FG44A - FG44N, when and

where scalable application should be started or stopped.

2) **Host Load Analyzer** FG40 is responsible for assigning a set of fitness scores to each host based on host capabilities and loads.

3) **QoS Managers** FG44A -FG44N are responsible for monitoring application and path requirements as defined in the System Specification Files FG32 and recommending that applications be either scaled up, scaled down, or moved in order to maintain acceptable performance.

As mentioned above, the Resource Manager FG42 is the primary decision-making component of the Resource Management Architecture. It is responsible for:

(1) responding to application and host failures by determining if and what recovery actions should be taken;

(2) determining if and where to place new copies of scalable applications or which scalable applications should be shutdown when the QoS Managers indicate that scale-up or scale-down actions should be taken based on measured application performance;

(3) determining where new applications should be placed when requested to do so by Program Control; and

(4) determining which and how many applications should run based on application system (mission) priorities.

In order to accomplish these tasks, the Resource Manager FG42 maintains a global view of the state of the entire distributed environment including status information on all hosts A- N, networks 100, and applications A1 -NM. In addition, the Resource Manager FG42 also calculates software and hardware readiness metrics and reports these readiness values, for display purposes, to the display functional group FG6.

It will be appreciated from Fig. 2 that the Resource Manager FG42 receives status

and failure information about hosts, networks, and applications from Program Control function FG50. This information includes both periodic status updates and immediate updates when statuses change such as a new host being detected or an application failing. In the case of application shutdown, information as to whether the application was shutdown intentionally or whether the application failed is also provided. Program Control function FG50 also issues requests to the Resource Manager FG42 when new applications need to be dynamically allocated and when the Program Control function FG50 determines that the Resource Manager FG42 needs to assess and attempt to resolve inter-application dependencies (such as one application which needs to be running prior to starting up another application).

The Resource Manager FG42 responds to faulted applications and hosts by determining whether the failed applications can and should be restarted and attempting to determine where (and if) there are hosts available that the application can run on. When a decision is made by the Resource Manager FG42, a message is sent to Program Control FG50 specifying what application to start and where to put it, i.e., which of hosts A - N to start the application on. The same general mechanism is used when Program Control FG50 requests that the Resource Manager FG42 determine where to start new applications and/or how to resolve inter-application dependencies; the Resource ManagerFG42 responds with orders indicating what applications to start and where to start them. The Resource Manager FG42 advantageously can send application shutdown instructions to Program Control FG50 requesting that a certain application be stopped; this can occur when the QoS Managers FG44A-FG44N indicate that certain scalable applications have too many copies running or when application system priority changes (when an application changes from a high priority to a lower priority) occur resulting in scaling back the application system configuration.

The Resource Manager FG42 also receives host load and host fitness information on all known hosts from the Hardware Broker (Host Load Analyzer) FG40. This information

includes (1) overall host fitness scores, (2) CPU-based fitness scores, (3) network-based fitness scores, and (4) memory and paging-based fitness scores, along with (5) the SPEC95™ rating of the hosts. These scores are used by the Resource Manager FG42 for determining the "best" hosts for placing new applications when:

(1) responding to requests from the QoS Managers to scale up additional copies of an application;

(2) attempting to restart failed applications;

(3) responding to requests to dynamically allocate certain applications; and

(4) responding to application system (mission) priority changes which require scaling up additional applications.

The Resource Manager FG42also receives requests from the QoS Managers FG44A-FG44N for scaling up, moving, or scaling down specific applications. The Resource Manager FG42 responds to these requests by determining whether the request should be acted upon and, if so, determines the specific action to take. The Resource Manager FG42 then issues orders to Program Control FG50 to start up or shutdown specific applications on specific hosts.

It should be noted that when the Resource Manager FG42 is first started, it reads in the System Specification Files FG32 (via calls to System Specification Library FG34) which contains the list of hosts that are known to be associated with the distributed environment and information on all applications that can be run in the distributed environment. The application-level information includes where, i.e., on which host, specific applications can be run, which applications are scalable, which applications can be restarted, and any dependencies between applications.

The Resource Manager FG42 currently responds to application system priority changes received from the Readiness Broker (translation software in or associated with the Readiness Display FG66) in the following manner:

(1) If the priority is changed to None, all applications associated with the

specified system are shutdown.

(2) If the priority is changed to Low, all scalable applications within the specified system are scaled back to no more than 50% of potential maximum scalability and are not allowed to be scaled up past the 50% limit irregardless of performance.

(3) If the priority is changed to Medium, normal scaleup and scaledown functionality is allowed.

(4) If the priority is changed to High, all scalable applications are scaled up to at least 50% of potential maximum scalability and are not allowed to be scaled down to less than 50% irregardless of performance.

(5) If the priority is changed to Urgent, all scalable applications are scaled up to 100% (for maximum survivability) and are not allowed to be scaled down.

[Moreover, if the previous priority was None, and the new changed priority is higher than None, all required applications within the specified system are started up subject to the limitations outlined for each of the priority levels listed above.]

The Resource Manager FG42 also sends information about allocation and reallocation decisions to the Resource Management Decision Review Displays FG68A-FG68N, as discussed in greater detail below. Information on the decision that was made, what event the decision was in response to, and how long it took to both make the decision and implement the decision advantageously are also sent to the display functional group FG6. In addition, information about the alternative choices for where an application could have potentially been placed is also provided (if applicable); in an exemplary case, this information includes the host fitness scores for the selected host and the next best host choices which could have been selected.

As described above, the Resource Manager FG42 communicates with Program Control FG50, the Hardware Broker FG40, the QoS Managers FG44A -FG44N, QoS

Specification Control (not shown), the Readiness Broker of display FG66, the Globus Broker (e.g., message translation software (not shown)), and the RM Decision Review Displays FG68A- FG68N using the RMComms middleware, which will be discussed in greater detail below.

The Hardware Broker (Host Load Analyzer) FG40 is the host load analysis component of the Resource Management Architecture, which is primarily responsible for determining the host and network loads on each host A - N within the distributed computing environment. The Hardware Broker FG40 assigns a set of fitness scores for each host and periodically provides the list of fitness scores to the Resource Manager FG42.

The Hardware Broker FG40 advantageously receives operating system-level statuses and statistics for each host A- N from the History Server(s) FG12A-FG12N, respectively. This information can be employed for calculating CPU, network, memory, paging activity, and overall fitness scores for each of the hosts A-N. Preferably, the Hardware BrokerFG40 periodically, e.g, once per second, provides the complete list of host fitness scores to the Resource Manager FG42.

It should be noted that when the Hardware Broker FG40 is first started, it reads in the System Specification Files FG32 (via calls to the System Specification Library (SSL) FG34), which files contain the list of hosts that are known to be in the distributed environment. The Hardware Broker FG40 also receives, e.g., reads in a file containing, information about the bandwidth and maximum packet sizes on all known network subnets in the distributed environment. It will be appreciated that this data advantageously can be used for converting host network load information based on packet counts to load information based on bytes per second and percentage of available bandwidth.

Periodically, e.g., approximately every three seconds, the Hardware Broker FG40

transmits a list of overall and network host fitness scores to the Hardware Broker Instrumentation Display which was constructed using the Graph Tool Instrumentation Display FG69A-FG69N. Moreover, the Hardware Broker FG40 advantageously can receive host-based network load data from the Remos Network Data Broker function FG16, which receives network data via the Remos Network Monitoring software 2. It should be noted that if Remos network data is available for any of the hosts A -N that are being monitored, the Remos reported network data advantageously can be used for calculating the network fitness score for that host, rather than using the host network data received from the History Server(s) FG12A-FG12N.

The QoS Managers FG44A - FG44N of functional group FG4 are responsible for monitoring application-level performance requirements. These requirements are defined in the System Specification Files FG32 and are monitored primarily via instrumentation data obtained directly from the application code. The QoS Managers FG44A - FG44N advantageously can determine if applications or application paths are meeting their assigned requirements. If an application is not meeting its performance requirements and the application is scalable (in the sense that multiple copies can be run and the copies will perform load-sharing across the copies), the QoS Managers FG44A - FG44N will either request that the Resource Manager FG42 scale up a new copy of the application or move the application to a new host (as an attempt to achieve better performance). Moreover, if there are multiple copies of a scalable application running, and all copies are performing well below the specified requirement threshold, the QoS Managers FG44A - FG44N will request that the Resource Manager FG42 shutdown a specific copy. It should be noted that the division of responsibility between the QoS Managers FG44A - FG44N and the Resource Manager FG42 is that the QoS Managers determine what actions would potentially improve performance, while the Resource Manager has final authority to determine whether to implement the requested action(s).

Each of the QoS Managers FG44A - FG44N can be scaled for both redundancy and for load-sharing. In an exemplary case, each copy of the QoS Manager monitors all of the requirements associated with a single application path defined in the System Specification Files FG32. It will be appreciated that the specific path to be monitored can be specified via command-line parameters. By default, without specifying a path via the command-line, the QoS Managers FG44A - FG44N will monitor all requirements for all paths defined in the System Specification Files FG32.

It should be mentioned that, in one exemplary embodiment, the QoS Managers FG44A - FG44N each employ a sliding window algorithm to determine when to declare that applications should be scaled up or scaled down. The inputs to the algorithm define both high and low sampling window sizes, the maximum number of allowed violations within the sampling window, and violation thresholds as a percentage of the actual specified requirement value. It should also be mentioned that the sliding window algorithm was selected in order to damp out unexpected "noise" or "spikes" in the measured performance data. Moreover, the threshold value as a percentage of the actual requirement value was selected in order to scale up, or scale down, prior to violating the specified hard requirement. The QoS Managers FG44A - FG44N provide application scale up and scale down requests to the Resource Manager FG42 when the measured performance data for an associated application violates either the high (scale up) or low (scale down) sliding window criteria for a specific requirement. A scale up request indicates which application on which host has violated the performance criteria, and a scale down request indicates which application on which host is recommended to be shutdown. It will be appreciated that the success of this algorithm is highly dependent on the rate of change and noisiness of the measured data.

Any of the QoS Managers FG44A - FG44N can also request that the Resource Manager FG42 move an application. This will occur in the case where one copy of an application is performing much worse than all other running copies of the same application.

In an exemplary case, the move request is implemented as a scale up request followed by a scale down request (of the badly performing copy). In that case, the scale down request does not get sent to the Resource Manager FG42 until the scale up action has been implemented. The QoS Managers FG44A - FG44N preferably employ application "settling times," defined in the System Specification Files FG32, to ensure that once a requested action has been sent to the Resource Manager FG42 that no additional actions are requested for that application until after the settling time has elapsed. It will be appreciated that this provides time for initialization and configuration among the application copies to occur. Alternatively, System Specification Language inter-application dependency definitions advantageously can be used instead of settling times.

The QoS Managers FG44A - FG44N also receive application status and state information from Program Control FG50, which periodically sends application status updates for all running applications and also sends immediate indications of any applications which have been started or stopped. This information is used by the QoS Managers FG44A - FG44N, along with the instrumented performance data being received via the QoS Monitor FG29 and Instrumentation Correlator FG34, to determine the exact state of all monitored applications that are running. This information is also used to determine when (and if) requested actions have been implemented by the Resource Manager FG42. The information is also used for setting up and discarding internal data structures used for monitoring the performance of each applicationA1-NM.

It will be appreciated that the QoS Managers FG44A - FG44N also receive application-level instrumentation data indicating current application performance values from the Instrumentation Correlators (Brokers) FG26A -FG26N, the Instrumentation Brokers FG28A-FG28N, and/or the Jewel Instrumentation Broker (QoS Monitor) FG29. The instrumentation data that is received contains (at a minimum) (1) the timetag when the data was generated, (2) the hostname and IP address of the host where the application that the

data is associated with is running, (3) the process id (pid) of the application that the data is associated with, and (4) the event number of the instrumentation message. Preferably, the event number of the instrumentation message specifies the type of instrumentation data that has been received; the hostname, IP address, and pid are used, in conjunction with the application data received from Program Control FG50, to determine the specific application that the data is associated with.

When the contents of the instrumentation message match any of the application performance requirements that are currently being monitored by the QoS Managers FG44A - FG44N, the data value is added to the proper requirement sliding window for the specified application. The sliding window algorithm is then checked to determine if the new sample triggered a violation of either the high or low sliding window. If a high threshold sliding window violation occurs and the application does not already have the maximum number of copies running, a determination is made as to whether performance can be best improved by starting a new application (scale up) or by moving an existing copy to a different host. The corresponding action recommendation will then be sent to the Resource Manager FG42. In an exemplary case, the criteria for determining whether an application should be moved rather than scaled up is based on relative performance of the replicated applications. More specifically, if one application is performing much worse [> 50%] than the other copies, the recommendation will be to move the application. Likewise, if the new sample triggers a low threshold sliding window violation and the application has more than the minimum number of copies running, a recommendation will be sent to the Resource Manager FG42 requesting that the copy of the application that is experiencing the worst performance be scaled down.

FG5 - RESOURCE (APPLICATION) CONTROL FUNCTIONAL GROUP

As discussed above, the Resource Control capabilities provided by the Resource Management Architecture consist of controlling application startup, configuration, and

Ex.1002 / Page 459 of 541
TESLA, INC.

shutdown on hosts within the distributed environment. This capability, known as Application Control or Program Control (hereafter referred to as Program Control) provides a powerful distributed configuration capability. The Program Control capabilities permit an operator to startup and control applications running on platforms throughout the distributed environment via an easy-to-use interactive display. These capabilities are provided by the Application Control functional group FG5.

More specifically, the Application Control functional group provides application control (i.e., Program Control) capabilities which permit starting, stopping, and configuring applications on each of the hosts in the distributed environment. The functional group provides both interactive operator control of the distributed environment as well as automatic control via configuration orders received from the Resource Allocation Decision-Making functional group FG4, i.e., the Resource Manager component. The interactive controls allow an operator to create, load, save, and edit pre-defined system configurations, e.g., lists of applications that are to be run, with or without specific host mappings, determine the status and configuration of currently running programs, and start and stop any or all applications. Both static (operator-entered) mappings of applications to hosts and dynamic mappings of applications to hosts (where the Resource Allocation Decision-Making functional group FG4 will be queried to determine the proper mapping at run-time) advantageously can be defined. The functional group also provides application fault detection capabilities which are triggered by the unexpected death, i.e., fault, of an application that was started by the functional group. A basic host fault detection capability is also provided which is triggered based on failure to receive heartbeat messages from functional group components running on a particular host.

A brief description of each function provided by the functional group FG5 is provided below; a detailed discussion of the Resource Control functional group FG5 and associated data flow will be provided in discussing Fig. 4.

Ex.1002 / Page 460 of 541
TESLA, INC.

1) **Program Control Agents** FG52A- FG52N: A Program Control agent generally denote DFG52 resides on each of the hosts A-N. Each agent is responsible for providing direct control over application startup and shutdown of applications on its respective host. The agent receives control orders from the Program Control function FG50 and is then responsible for implementing the orders. In an exemplary case, the agents implement the orders via system call mechanisms specific to the particular operating system. In addition, the agent also provides feedback to the Control function FG50 regarding the current status of all applications running on a particular host.

2) **Program Control FG50** - maintains the application state information for the Program Control functional group FG5. It also serves as the decision-making component of the Program Control functional group. The Control function FG50 receives application control (startup, shutdown, or configuration) requests from the Program Control Displays FG54A - FG54N and from the Resource Management functional group FG4. Using information from the Specification Files FG32, these high-level control function requests are dynamically translated into specific control orders which are sent to the individual Program Control agents FG52A -FG52N. The program Control FG 50 also provides application status and configuration information back to the Resource Manager FG42.

3) **Program Control Displays** FG54A - FG54M - serve as the GUI for interactive control of distributed applications. The Program Control Displays FG54A - FG54M allow an operator to see and control the status of applications running on each host in the distributed environment. The

Program Control Displays FG54A - FG54M also provide the user the ability to determine the status of each of the components of the Program Control architecture. Predefined scenario configurations defined in Program Control Configuration Files FG56 advantageously can be loaded and edited via the Displays. It should be mentioned that new Program Control Configuration Files can also be created and saved via the Displays. As illustrated in Fig. 2, Program Control Displays FG54A - FG54M can be run simultaneously with application status changes being reflected at each display.

4) **Configuration Files FG56** - contain an ordered set of applications that can be loaded at the Program Control display and then either edited or executed. The Configuration Files can contain both dynamic and static application-to-host mappings. For static application-to-host mappings, an application will, by default, be started on a specified host. For dynamic application-to-host mappings, the application will have a default host to start on but the Resource Manager FG42 will be queried at run-time to determine where the application actually should be placed. The Configuration Files FG56 also contain all information on how to start, stop, and configure an application, with the exception of environment variable settings for the application which are set based on the System Specification Files FG32.

It should be mentioned here that the Program Control functional group employs the application startup and shutdown information defined in the System Specification Files FG32. When an application entry is first created interactively at one of the Program Control Displays FG54A - FG54N, all of the startup and shutdown information for that application, as specified in the System Specification Files FG32, are loaded in as default settings. Once a configuration file entry has been created, all configuration information on the application

is read in from the configuration file except for the application environment variable settings which are still set based on the System Specification Files FG32.

As mentioned above, a Program Control agent resides on each host. The agent is responsible for providing direct control over application startup and shutdown. The agent receives control orders from the Control component and is then responsible for implementing the orders. Each of the PC Agents FG52A - FG52N implements application startup and shutdown orders via system call mechanisms specific to the particular operating system of the host. For example, on the Unix platforms, to start an application, the fork() and execv() function calls are used to create the application. The csh command is executed to start up the applications. Moreover, if the application needs to run in a console, an xterm is configured for the application to run in. In addition, if logging of either stdout or stderr is specified, the proper redirection operators are configured and the output log file is set to "/usr/tmp/<userid>_<appname>_<pid>.log". All environment variables needed by the application are also configured and passed in at the execv() call. The current working directory is also set by the chdir() command, and the new application is made a process group leader via the setpgid() function. Other operating systems invoke applications using different calls.

In order to stop an application on the Unix platforms, if a signal is to be sent to the application, the killpg() function is used, or else if a script or command is to be executed to shutdown the application, the csh command is executed (via the system() function) specifying the full path and executable name of the command along with any arguments for the command. It should be noted that if the application default shutdown time elapses and the application has not died, the respective one of the Program Control Agents FG52A-FG52N advantageously sends a SIGKILL signal to the application by calling killpg().

As illustrated in Fig. 1, the Program Control Agents (PCA-PCN) advantageously can be instantiated on stand-alone hosts A - N. In that case, the Program Control Agents PCA-PCN (FG52A-FG52N in Fig. 2) send heartbeat messages to Program Control FG50 approximately once per second to indicate that they are still up and running. Moreover, every ten seconds, the Program Control Agents PCA-PCN (FG52A-FG52N) send complete configuration information on all running applications to Program Control FG50. It should be noted that the terminology employed in Fig. 1 differs from that in Fig. 2 to emphasize the distinction between software instantiated on a host and a function provided by the Resource Management Architecture.

The Program Control function FG50 is the decision-making component of the Program Control functional group FG5. It maintains complete information on everything that is running across all platforms in the distributed environment. The Program Control function FG50 receives input data from PCA-PCN (FG52A-FG52N), the Program Control Displays FG54A-FG54N, the Resource Manager FG42, and the Host Discovery function FG14.

It will be appreciated from the preceding discussion that the Program Control FG50 provides startup and shutdown orders to the Program Control Agents FG52A-FG52N based on operator or Resource Manager-initiated orders. If the Program Control Agents report that an application has terminated abnormally, the Program Control FG50 provides a notification to the Resource Manager FG42, to the Program Control Displays FG54A - FG54N, and to any other component to which it is connected. When the Program Control function FG50 is first brought up, it can be configured to attempt to start Program Control agents on every host defined in the System Specification Files. The Program Control function FG50 will also attempt to start a Program Control Agent on a newly discovered host (discovered via the Host Discovery function FG14) if Host Discovery has been enabled on the Program Control Displays FG54A-FG54N.

The Proggram Control function FG50 also receives periodic heartbeat messages, e.g., once per second, from each of the Program Control Agents FG52A-FG52N, as discussed above. If Fault Detection has been enabled at the Program Control Displays FG54A-FG54N, if three consecutive heartbeat messages from an Agent, e.g., FG52A, are missed, the host that the agent is running on is declared down and all linked functions , including the Resource Manager FG42 and the Displays FG54A-FG54N are notified.

As mentioned above, the Proggram Control function FG50 sends out periodic application status updates as well as immediate notification when applications are started up, are shutdown, or fail. These notifications are sent out to all linked functions.

It should be noted that the Proggram Control function FG50 uses the same message traffic and internal processing for handling application startup and shutdown orders received from either the Resource Manager FG42 or from the Program Control Displays FG54A-FG54N. However, if a startup order received from one of the Program Control Displays FG54A-FG54N indicates that the Resource Manager FG42 should determine where to run the application, a request to allocate the application is sent to the Resource Manager FG42. When no response is received from the Resource Manager FG42 within a predetermined timeout period, the Proggram Control function FG50 will automatically start the application on the default host. Moreover, when an application startup cannot proceed due to an unfulfilled application startup dependency, a request will be made to the Resource Manager FG42 to attempt to resolve the dependency. If the Resource Manager FG42 either cannot resolve the dependency or no response is received within a predetermined timeout period, the application startup will fail, and a "dependency failed" indication will be sent to the Display. It will be appreciated that this will cause the application status to be displayed in, for example, yellow and post an alert to the Alert window on one of the Program Control Displays FG54A-FG54N.

Preferably, Program Control function FG50 also handles simple startup timing dependencies between applications and will reorder a list of applications that were selected to be started simultaneously if doing so will resolve startup order dependencies between the applications. Otherwise, the Program Control function FG50 sends a request to the Resource Manager to attempt to resolve the dependencies.

The Program Control Display serves as the operator console for controlling the distributed environment. From the Display, shown in Fig. 5, the operator can:

1) see the status and configuration of currently executing applications A1-NM;
2) see the status of Program Control Agents PCA-PCN on each host A-N;
3) see and browse the application system structure defined in the System Specification Files FG32;
4) load configuration files FG56
5) save configuration files FG56
6) edit the configuration of applications that are not currently running;
7) create new application entries by dragging an application, application system, or application subsystem icon onto the application status area;
8) manually start specific applications;
9) manually stop specific applications;
10) manually start all applications that have the "Start All" flag set;
11) manually stop all applications;
12) turn host fault detection on or off (if on, loss of 3 consecutive heartbeats from a Program Control Agent will result in declaring the host down); and
13) turn host discovery on or off (if on, a new host message from the Host Discovery component will result in attempting to start up a Program Control Agent on the new host).

It will be appreciated from Fig. 2 that multiple Program Control Displays FG54A-

Ex.1002 / Page 466 of 541
TESLA, INC.

FG54N advantageously can be run simultaneously. If this is done, any configuration change actions will be reflected on all the displays. Whenever application stop or start actions are taken by the display operator, a message is sent to the Proggram Control function FG50 which is responsible for enacting the start or stop action. The Program Control function FG50 also sends indications of any status changes to the Program Control Displays FG54A-FG54N as soon as the status changes are seen. In addition, periodic status updates are also sent to the Program Control Displays FG54A-FG54M.

The Program Control Configuration Files are text files that are read in by the Program Control Display when the operator wishes to load a new application configuration. A Configuration File is an ASCII file containing a list of applications. The format of an entry in a Configuration File is shown in Table 1 below.

Table 1

```
Application     TACFIRE:tacfire
Host            electra1
Display         umbriel1:0.0
Auto_Start      0
RM_Start        0
Console         1
Time_Delay      1
StartupDir      "$ENV_SIM_VERSION/TACFIREprocessor"
StartupExe      "$ENV_SIM_VERSION/TACFIREprocessor/tacfire"
StartupArgs     "-disport    $DIS_PORT_NUM    -cffhost    %(HOSTNAME,
                AAW:Tactical_Sims:CFF_Broker)"
ShutdownExe     SIGINT
LogType         STDOUT
LogDir          "/usr/tmp"
```

The Configuration file advantageously can include the following fields:

1) The **Application** field, which identifies the full application name as defined in the System Spec. Files FG32 (i.e., System:Subsystem:Application).

2) The **Host** field, which is the desired or default host that this application should be started on.

3) The **Display** field, which is an optional field used when graphical display

output from an application needs to be rerouted to a display on a different host.

4) The **Auto_Start** flag, which identifies whether the application is to be started automatically if the "Start All" action is selected by the operator from the Program Control Display. (If the flag were set to "1", then the application would be started. If the flag were set to "0," it would not be started.)

5) The **RM_Start** flag, which identifies whether the the Resource Manager should be queried at run-time to determine what host the application should be started on. The valid values are "0" for "NO" and "1" for "YES".

6) The **Console** flag, which identifies whether the application needs to be started in an Xterm window. The valid values are 0 for "NO" and 1 for "YES".

7) The **Time_Delay** field, which identifies how many seconds to wait after the previous application has been started before starting this application.

8) The **StartupDir** field, which identifies the current working directory that is to be set prior to starting up the application. This directory is usually the same as the directory where the executable for the application resides but does not have to be. As this example shows, environment variables may be used in the path.

9) The **StartupExe** field identifies the entire path and name of the application executable.

10) The **StartupArgs** field, which contains all the argument values needed for this particular application. As this example indicates, the argument values can be dynamically set at run time if needed. Environment variables may also be used within the argument list. In this example, the %(UNIQUE, 1, 40, Isis) argument would yield a number from 1 to 40 which is unique within a context named "Isis". Another resolution of %(UNIQUE, 1, 40, Isis) would yield a different number.

11) The **ShutdownExe** field, which identifies which signal defined within the application that program control is to use to shutdown this application. Some examples would be SIGINT, SIGTERM, or SIGKILL. A shutdown script can also be used to shutdown the application. (In that case, there would be **ShutdownDir, ShutdownExe,** and **ShutdownArgs** fields listed. The usage for the shutdown fields would be used exactly the same as the startup fields.)

12) The **LogType** field, which identifies which outputs are to be written to the specified log file. The valid values are STDOUT, STDERR, and LOG_ALL. STDOUT is the normal output of the application (stdout). STDERR is the error output of the application (stderr). LOG_ALL writes both stdout and stderr outputs to the file.

13) The **LogDir** indicates the directory where the log file will be written. Again, environment variables may be used here. The log file name will be "<userid>_<appname>_<pid>.log" where <appname> is the full application name as specified in the **Application** field, <userid> is the userid of the current user under which the program control application is running, and <pid> is the system assigned process id of the application being executed.

FG6 - DISPLAY FUNCTIONAL GROUP

A number of displays which show system configuration data and instrumentation data in near real-time are included as part of the Resource Management Architecture. These displays support operator and user monitoring of the operation of the distributed environment including host and network statuses and performance, application system statuses and performance, as well as the status and performance of the other Resource Management architecture functions. Most of the displays use OpenGL and Motif , the latter being built with ICS's Builder Xcessory toolkit, and run on Silicon Graphics (SGI) platforms in an exemplary case. Several of the displays can also run on the Sun Solaris platforms. The

Ex.1002 / Page 469 of 541
TESLA, INC.

displays that make up the display functional group FG6 include:

1) **Host Displays FG62A -FG62N.** Show layout of hosts along with host status, network connectivity, and process statuses.

2) **Path Display FG64.** Shows the status of applications in key end-to-end data flow paths along with performance and load graphs.

3) **Resource Management Decision Review Display FG68.** Shows a summary of allocation decisions made by the Resource Management system along with timing information and host fitness scores.

4) **Graph Tool Instrumentation Displays FG69A-FG69N.** Provides a user-configurable set of display widgets used for run-time monitoring of instrumented status and performance information.

5) **System Readiness Display FG66.** Shows the status of each hardware and software system, subsystem, and application defined in the System Specification Files and allow the operator to interactively change system and subsystem priorities.

Fig. 6 is a screen capture of an exemplary one of the Host Displays FG62A-FG62N, which provide graphical representations of various sets of the hosts A- N in the distributed environment. The Host Displays show the status of each host, host network connectivity, and the status of interesting processes running on the hosts. The Host Display operator can also select hosts shown on the Host Display and bring up real-time graphs of system performance for the selected hosts including CPU utilization, memory utilization, network packets in, network packets out, and paging activity. A screen capture of host specific performance information is provided in Fig. 7.

Fig. 8 is a screen capture of a representative Path Display FG64, generated by the Resource Management architecture, which shows the status of key system data flow paths consisting of multiple application stages. The number of copies of each application in the

path is shown labeled with the host on which the application is running. In addition, it should be mentioned that as many as three real-time graphs can be produced to depict run-time performance and load metrics related to the applications in the selected data path.

Fig. 9 is a screen capture of the Resource Management Decision Review Display FG68, which advantageously can provide a summary of allocation and reallocation actions taken by the Resource Manager FG42. For each action, timing information regarding how long it took the Resource Management functions, e.g., the Resource Manager FG42 and the Program Controller FG50, to both arrive at a decision and to enact the decided action are shown along with host fitness scores that were used in arriving at the allocation decision.

Figs. 10 and 11 are screen captures of the Graph Tool Instrumentation Displays FG69A-FG69N, which depict user-configurable displays capable of receiveing data via standardized message formats and open interfaces. The Graph Tool Displays FG69A-FG69N allow the operator to select and configure various display widgets (line graphs, bar charts, pie charts, meters, and text boxes) to build a desired display layout. Data sources for driving the widgets can also be selected interactively.

Fig. 12 is a screen capture of the System Readiness Display FG66, which advantageously can be a Java™ display with a CORBA™ interface. The display FG66 shows the status of each hardware system, host, application system, application subsystem, and application defined in the System Specification Files. The top portion of the display shows a summary status for each defined application system. It should be noted that the display operator can also change system and subsystem priorities and send the changed priorities to the Resource Manager function FG42.

As mentioned above, the RMComms middleware package provides object-oriented client-server services for message communication between distributed applications and

function modules. The middleware provides location transparency and automatic socket connections and reconnections between client and server applications. These services advantageously can be accessed through an object-oriented API which allows client and server objects to be easily created and exchange user-defined message data. The abstraction provided by the API allows the user to quickly and easily create distributed applications without needing to be aware of the details of the underlying network mechanisms. The RMComms middleware provides the following functions:

- provides location transparency between clients and servers
- provides a simple powerful object-oriented client-server API
- supports reliable transport of user-defined message data
- based on Berkeley sockets
  - uses TCP for message transport
  - uses UDP multicast for identification of new clients or servers
  - servers identified by unique assigned UDP/TCP port numbers
- provides general purpose callback function registration capabilities
  - user-specified message callback functions invoked when specified messages arrive
  - user-specified connection status callback function invoked when new client-server connections are established or existing connections are broken
- support for multi-threading
  - supports both polled and asynchronous I/O
  - thread-safe
- provides automatic connections between clients and servers
  - supports multiple client and server connections within the same application
  - provides automatic connections to new clients / new servers
  - supports simultaneous many-to-many client-server connections

- no separate "naming service" or "application registration" components
- provides automatic client-server connection fault detection and recovery
  - provides fault detection mechanisms based on timeouts and broken connections
  - supports fault recovery via automatic reconnections between clients and servers
- provides basic support for data marshalling between machine architectures
  - byte-swapping
  - explicit message data type specification
  - all message data sent out using network byte order
- provides basic capabilities for reading the system clock and performing time conversions
- allows registration of user-defined signal (interrupt) handler functions
- layered object-oriented design and implementation
- cross-platform support:
  - SGI Irix 6.3/6.4/6.5
  - Sun Solaris 2.5.1/2.6/2.7/.2.8
  - HP HP-UX 10.20
  - Linux 2.1/2.2
  - Windows NT 4.0
  - Windows 95/98/2000
  - Solarisx86 2.7
- C++ language support using native and GNU compilers

The RMComms middleware is implemented as a shareable object-oriented C++ library. The library provides four primary object classes, which are detailed in Attached Appendix C. It will be appreciated that the applications link with this library and can then

instantiate client and server objects for communicating with other local or remote applications. It should be mentioned that the application source code must also include a set of header files that allow connections between client and server objects, where each server type is assigned a server port number. For clients and servers that want to communicate, both the client and the server objects are created specifying the same server port number. Multiple servers of the same type can also be created, which all use the same server port number. This advantageously provides the ability for many-to-many client-server connections to be established, as illustrated in Fig. 4. Control of which servers the clients actually connect to is handled on the client side; clients can specify whether they wish to establish connections with all servers in the distributed environment, with a particular set of servers, or with all servers running on a particular set of hosts.

The operation of the Resource Management Architecture will now be described while referring to Figs. 13A-13C, which illustrate various operations in the distributed environment. More specifically, the Resource Management Architecture of the system illustrated in Figs. 13A includes hosts A-N, where host A provides a video source server application A-1, host B provides a video distribution application B-1, a contract application B-2, and a host load monitor B-3, and host C provides a display broker application C-1 applying video signals to a display driver C-2. It will be appreciated that host D is idle and that the connections between the various hosts constitute the network 100'. In addition, the Resource Management Architecture of Fig. 13A instantiates various functions, e.g., an instrumentation broker FG26', a QoS manager FG44', a resource manager Fg42' and a program control FG50'. The instrumentation broker FG26' receives data from each of the applications running in the distributed environment, although only the lines of communication between the applications running on host B are actually depicted. From the discussion above, it will be appreciated that each of the applications is linked to an Instrumentation API.

Referring now to Fig. 13B, a QoS violation and its consequences is depicted. In particular, the Instrumentation broker FG26' provides data to the QoS manager FG44' which is indicative of a QoS violation. The QoS manager FG44' notifies the resource manager Fg42' of the violation; the resource manager determines that duplicate copies of the applications running on host B are required and that these copies should be placed on host D. The resource manager FG42' transmits instructions to the Program Control function Fg50', which starts copies of the running applications, i.e., a video distribution application D-1, a contract application D-2, and a host load monitor D-3, on host D. Fig. 13C illustrates shutdown of the application copies running on host B. It will be appreciated that this shutdown may be initiated responsive to the original QoS violation, another QoS violation, or a query from the user.

Having discussed the various functions and features of the Resource Management Architecture in gross, selected functions and features will now be described in detail. It will be appreciated that the discussion of the various functions will be signaled using the designations established with respect to Fig. 2.

FG42 - Resource Manager Function

As mentioned above, the Resource Manager FG42 is the primary decision-making component of the Resource Management functional group. It is responsible for:

(1) responding to application and host failures by determining if and what recovery actions should be taken;

(2) determining if and where to place new copies of scalable applications or which scalable applications should be shutdown when the QoS Managers indicate that scale-up or scale-down actions should be taken based on measured application performance:

(3) determining where new applications should be placed when requested to do

so by Program Control: and

(4) determining which and how many applications should run based on application system (mission) priorities.

In order to accomplish these tasks, the Resource Manager FG42 maintains a global view of the state of the entire distributed environment including status information on all hosts, networks, and applications. In addition, the Resource Manager FG42 also calculates software and hardware readiness metrics and reports these readiness values for display purposes.

The Resource Manager FG42 is an object-oriented multi-threaded application written in C++, which uses the RMComms middleware for all external communication. The Resource Manager FG42 communicates with the various software components instantiating the (1) Program Control FG50, 2) Hardware Broker FG40, 3) QoS Managers FG44A - FG44N, 4) QoS Specification Control FG29, 5) Readiness Broker in Readiness Display FG66, 6) Globus Broker (not shown), and 7) RM Decision Review Displays FG68A-FG68N.

It will be appreciated that the Resource Manager FG42 receives status and failure information about hosts and networks from the Host and Network Monitoring functional group FG1, and applications from the Program Control functional group FG5. This information includes periodic status updates as well as immediate updates when statuses change, e.g., when a new host is detected or an application fails. In the case of any application shutdown, information as to whether the applications were intentionally shutdown or whether the application actually failed advantageously can be provided. The Program Control function FG50 also issues requests to the Resource Manager FG42 whenever new applications need to be dynamically allocated and whenever the Program Control function FG50 determines that the Resource Manager FG42 needs to assess and attempt to resolve inter-application dependencies (e.g., one application which needs to be running prior to starting up another application).

The Resource Manager FG42 responds to applications faults and host failures by determining whether the failed applications can and should be restarted and attempting to determine where (and if) there are hosts available that the application can run on. When a decision is made by the Resource Manager FG42, a message is sent to Program Control function FG50 specifying what application to start and where to put it. The same general mechanism is used when the Program Control function requests that the Resource Manager FG42 determine where to start new applications and/or how to resolve inter-application dependencies; the Resource Manager FG42 responds with orders indicating what applications to start and where to start them. The Resource Manager FG42 advantageously can send application shutdown orders to the Program Control function FG50 requesting that a certain running application be stopped; this can occur when the QoS Managers indicate that certain scalable applications have too many copies running or when application system priority changes (to lower priorities) occur resulting in scaling back the application system configuration. See Figs. 13B and 13C and the associated discussion above.

The Resource Manager FG42 receives host load and host fitness information from the Hardware Broker (Host Load Analyzer) function FG40. This information includes overall host fitness scores, CPU-based fitness scores, network-based fitness scores, and memory and paging-based fitness scores along with the SPEC95 rating of the hosts. This information is received approximate once a second and includes information on all known hosts in the distributed system. These scores are used by the Resource Manager FG42 for determining the "best" hosts for placing new applications when:

(1) responding to requests from the QoS Managers to scale up additional copies of an application;

(2) attempting to restart failed applications;

(3) responding to requests to dynamically allocate certain applications; and

(4) responding to application system (mission) priority changes which require scaling up additional applications.

Advantageously, the Resource Manager FG42 also receives requests from the QoS Managers FG44A - FG44N for scaling up, moving, or scaling down specific applications. The Resource Manager FG42 responds to these requests by determining whether the request should be acted upon and, if so, determines the specific action to take and issues orders to the Program Control function FG50 to start up or shutdown specific applications on specific hosts. The QoS Managers FG44A - FG44N are responsible for monitoring specific system performance metrics (e.g., quality of service, or QoS, requirements) via instrumentation and determining if performance can be improved by scaling up or moving certain applications. When this occurs, the QoS Managers send a request to the Resource Manager FG42 indicating that a new copy of a specific application should be started. If the QoS Managers determine that the performance of a scalable application can be improved by moving an application, a scale up request is first sent to the Resource Manager FG42 and when the new application has been started, a scaledown request is then sent to the Resource Manager FG42. Moreover, when the QoS Managers FG44A -FG44N determine that there are more copies of scalable application running then are needed, requests to shutdown specific applications are sent to the Resource Manager FG42.

It will be appreciated that the Resource Management Architecture distributes functionality between the QoS Managers FG44A-FG44N and the Resource Manager FG42. Thus, the QoS Managers determine what actions would potentially improve performance, while the Resource Manager FG42 has final authority to determine whether to implement the requested actions.

It should be noted that when the Resource Manager FG42 is first started, it reads in the System Specification Files FG32 (via calls to the System Specification Library (SSL) Fg34) which contains the list of hosts that are known to be (operating) in the distributed environment and information on all applications that can be run in the distributed environment. The application-level information includes where specific applications can be

run, which applications are scalable, which applications can be restarted, and any dependencies between applications. In addition, the Resource Manager FG42 receives updated application survivability specifications from the QoS Specification Control function. This information overrides the application survivability information that was initially loaded in from the System Specification Files FG32 for the specified application. The information is used by the Resource Manager FG42 to determine whether the specific application will be restarted if it fails at run-time.

It should also be noted that the Resource Manager FG42 sends application system and hardware system readiness and system (mission) priority information to the Readiness Broker, which is a translator withing the Readiness Display FG66 and to the Globus Broker (another Broker (not shown)). The Readiness Broker is responsible for driving a GUI/display FG66, which shows the current readiness data and allows the system (mission) priorities to be changed and sent back to the Resource Manager FG42. The Globus Broker provides basically the same functionality except that only a high-level subset of the readiness data provided to the Readiness Broker is provided to the Globus Broker. The readiness information sent to the Readiness Broker consists of readiness values for each application, application subsystem, and application system defined in the System Specification Files FG32. The scores advantageously can be based on the status (up/down) of the applications and the percentage of potential copies of scalable applications that are currently running. Host and network readiness scores are determined based on the host loads and host fitness scores received from the Hardware Broker FG40.

The Resource Manager FG42 also sends information about allocation and reallocation decisions to the RM Decision Review Display FG68 (Fig. 9). Information on the decision that was made, what event the decision was in response to, and how long it took to both make the decision and implement the decision are sent to the display. In addition, information about the top choices for where an application could have potentially been placed is also sent (if applicable); this information includes the host fitness scores for the

selected host and other hosts which could have been selected.

As described above, the Resource Manager function FG42 communicates with Program Control FG50, the Hardware Broker FG40, the QoS Managers FG44A -FG44N, QoS Specification Control (not shown - legacy function), the Readiness Broker of the Readiness Display FG66, the Globus Broker (not shown), and the RM Decision Review Display FG68 using the RMComms middleware. The message formats and contents of each message that is exchanged between the Resource Manager function FG42 and other functional elements of the Resource Management architecture are described in CD-Appendix D. The timing and/or event trigger for each message is also described.

FG40 - Host Load Analyzer (Hardware Broker) Function

The Hardware Broker FG40 provides the host load analysis function of the Resource Management functional group FG4. It is responsible primarily for determining the host and network loads on each host within the distributed computing environment. The Hardware Broker FG40 assigns a set of fitness scores for each host and periodically provides the list of fitness scores to the Resource Manager FG42. Fig.14 illustrates the connectivity and high-level data flow between the Hardware Broker and the other Resource Management and Resource Management-related components.

The Hardware Broker FG40 is an object-oriented multi-threaded application written in C++, which uses the RMComms middleware for all external communication. It receives operating system-level statuses and statistics for each host from the History Server(s) FG12A –FG12N. This information is used for calculating CPU, network, memory, paging activity, and overall fitness scores for each host. The Hardware Broker periodically (once per second) sends the list of host fitness scores to the Resource Manager FG42.

When the Hardware Broker FG40 is first started, it reads in the System Specification Files FG32 (via calls to System Specification Library (SSL) FG34) which contain the list of hosts that are known to be in the distributed environment. The Hardware Broker also reads in the file *networks.dat* which contains a list of information about the bandwidth and maximum packet sizes on known network subnets. It should be mentioned that this data is used for converting host network load information based on packet counts to load information based on bytes per second and percentage of available bandwidth.

It should be mentioned that there are two other RMComms interfaces that the Hardware Broker FG40 uses. Periodically (approximately every three seconds), the Hardware Broker FG40 sends a list of overall and network host fitness scores to the Hardware Broker Instrumentation Display FG69A – FG69N. As mentioned above, these displays were constructed using the Graph Tool described in the Instrumentation Graph Tool Display. Additionally, the Hardware Broker FG40 can receive host-based network load data from the Remos Broker FG16, which receives network data via the Remos Network Monitoring software (denoted 2 in Fig. 2). If Remos network data is available for any of the hosts that are being monitored, the Remos data is used for the network fitness score calculation for that host rather than the host network data received from the History Server(s).

The exemplary instance of the Hardware Broker FG40 is an object-oriented multi-threaded application. At the highest level, the Hardware Broker object contains the elements listed in Table II below. It will be noted that Table II contains a brief description of each of these objects. Additional details are provided in CD-Appendix E.

Table II

No.	Title	Description

1	Host Fitness Database object (FitnessDB class)	The Host Fitness Database object stores load history data and fitness score information for each host. The Host Fitness Database is updated and fitness scores are recalculated when new History Server Host Status Response Messages are received. For each host, a circular queue of host load history data (HostInstance class) is maintained with the newest data being placed at the end of the queue; this history data is used for recalculating host fitness scores. The Host Fitness Database also contains a System Specification Library (SSL) object which is used to access SPEC rating information for the hosts.
2	Signal Registration object (SignalRegistry class)	The Signal Registration object allows for a user-defined SIGINT signal handler to be registered in order to permit the Hardware Broker FG40to be shutdown gracefully.
3	Network Subnet Information Database object (SubnetDB class)	The Network Subnet Information Database object is used to store IP address, maximum bandwidth, and MTU size for each network specified in the *networks.dat* file. This information is used for converting network packet load information to bytes/second network load information.

4	Remos Host Network Bandwidth Database object (RemosDB class)	The Remos Host Network Bandwidth Database object stores the latest Remos-reported network bandwidth information for each host being monitored. The information stored consists of available bandwidth as well as maximum potential bandwidth on a specific host network link. If Remos bandwidth information is available for a host and the latest data is less than 5 seconds old, the Remos data will be used for calculating the network fitness score for the host.
5	History Server Interface object (HistServInterface class)	The History Server Interface object inherits from the RMComms TCPCommClient class and is responsible for maintaining connections to the History Server(s), for registering status and message handler callback functions, for sending messages to the History Server(s), and for invoking the status and message handler callback functions when connections to History Servers are either established or broken or new messages are received from a History Server.
6	Instrumentation Graph Tool Display Interface object (InstrInterface class)	The Instrumentation Graph Tool Display Interface object inherits from the RMComms TCPCommServer class and is responsible for maintaining connections to the Graph Tool Display(s), for registering status and message handler callback functions, for sending messages to the Graph Tool Display(s), and for invoking the status and message handler callback functions when connections to Graph Tool Displays are either established or broken or new messages are received from a Graph Tool Display.

Ex.1002 / Page 483 of 541
TESLA, INC.

7	Resource Manager Interface object (ResMgrInterface class)	The Resource Manager Interface object inherits from the RMComms TCPCommServer class and is responsible for maintaining connections to the Resource Manager for registering status and message handler callback functions, for sending messages to the Resource Manager, and for invoking the status and message handler callback functions when connections the Resource Manager are either established or broken or new messages are received from the Resource Manager.
8	Remos Broker Interface object (RemosInterface class)	The Remos Broker Interface object inherits from the RMComms TCPCommClient class and is responsible for maintaining connections to the Remos Broker for registering status and message handler callback functions, for sending messages to the Remos Broker, and for invoking the status and message handler callback functions when connections the Remos Broker are either established or broken or new messages are received from the Remos Broker.

FG44: Quality-of-Service (QoS) Manager Function

The QoS Managers FG44A - FG44N are responsible for monitoring application-level performance requirements, which requirements are defined in the System Specification Files FG32 and are monitored primarily via instrumentation data obtained directly from the application code. The QoS Managers FG44A - FG44N advantageously determine if applications or application paths are satisfying their assigned requirements. When an

application is not meeting its performance requirements and the application is scalable (in the sense that multiple copies can be run and the copies will perform load-sharing across the copies), the QoS Managers FG44A - FG44N will either request that the Resource Manager FG42 scale up a new copy of the application or move the application to a new host (which hopefully will result in better performance). Moreover, if there are multiple copies of a scalable application running, and all copies are performing below the specified requirement threshold, the QoS Managers FG44A - FG44N will request that the Resource Manager shutdown a specific copy.

The QoS Manager is a single-threaded application written in C/C++. It should be noted that the application can be scaled for both redundancy and/or load-sharing. In an exemplary case, each copy of the QoS Manager monitors all of the requirements associated with a single application path defined in the System Specification Files FG32. It will be appreciated that the specific path to be monitored can be specified via command-line parameters. By default, without specifying a path via the command-line, the QoS Manager will monitor all requirements for all defined paths.

As mentioned above, the QoS Manager advantageously uses a sliding window algorithm to determine when to declare that applications should be scaled up or scaled down. The inputs to the algorithm define both high and low sampling window sizes, the maximum number of allowed violations within the sampling window, and violation thresholds as a percentage of the actual specified requirement value. It will be appreciated that the sliding window algorithm was selected in an effort to damp out unexpected "noise" or "spikes" in the measured performance data. Use of threshold values states as a percentage of the actual requirement value was selected in order to scale up, or scale down, prior to violating the specified hard requirement. It will be understood that the success of this approach is highly dependent on the rate of change and noisiness of the measured data..

Ex.1002 / Page 485 of 541
TESLA, INC.

Again, the QoS Manager uses the RMComms middleware for all external communication. Each copy of the Resource Manager talks to (1) Resource Manager FG42, (2) Program Control FG50, (3) QoS Specification Control (not shown), (4) QoS Monitor FG29, (5) Instrumentation Correlators FG26A –FG26N, (6) Graph Tool Instrumentation Displays FG69A-FG69N, and (7) History Servers FG12A-FG12N. in an exemplary case, the QoS Managers FG44A - FG44N advantageously can receive configuration orders from the Resource Manager FG42, which allows the Resource Manager FG42 to configure each QoS Manager to monitor specific application paths and also set the sliding window criteria to be used by each respective QoS Manager.

Each copy of the QoS Manager advantageously can transmit application scale up and scale down requests to the Resource Manager FG42 when the measured performance data for a respective application violates either the high (scale up) or low (scale down) sliding window criteria for a specific requirement. A scale up request indicates which application on which host has violated the performance criteria, and a scale down request indicates which application on which host is recommended to be shutdown. Each copy of the QoS Manager can also request that the Resource Manager move an application. This will occur in the case where one copy of an application is performing much worse than all other running copies. The move request is implemented as a scale up request followed by a scale down request (of the badly performing copy); the scale down request is not transmitted to the Resource Manager FG42 until the scale up action has been implemented.

The QoS Managers FG44A - FG44N use the application "settling times" defined in the System Specification Files to ensure that once a requested action has been sent to the Resource Manager that no additional actions are requested until after the application settling time has elapsed. This provides time for initialization and configuration among the application copies to occur. In future releases, the inter-application dependencies will be used instead.

The division of responsibility between the QoS Managers FG44A - FG44N and the Resource Manager FG42 is as follows:

(1) the QoS Managers FG44A - FG44N determine what actions would potentially improve performance; and

(2) the Resource Manager FG42has final authority to determine whether to implement the requested actions

It should be mentioned that there is a Request Acknowledge message from the Resource Manager FG42 which has been defined and implemented within the QoS Manager code. This message is intended to provide feedback to the QoS Manager indicating that the request had been successfully received and whether the Resource Manager FG42 intends to implement the request.

As previously mentioned, the QoS Managers FG44A - FG44N receive application status and state information from the Program Control function FG50. Program Control periodically sends application status updates for all running applications and also sends immediate indications of any applications which have been started or stopped. This information is used by the QoS Managers FG44A-FG44N, along with the instrumented performance data being received via the QoS Monitor FG29 and Instrumentation Correlators FG26A-FG26N, to determine the exact state of the monitored applications A1-NM that are running. This information is also used to determine when (and if) requested actions have been implemented by the Resource Manager FG42. The information is also used for setting up and discarding internal data structures used for monitoring the performance of each application.

The QoS Managers FG44A - FG44N also receive application-level instrumentation data indicating current application performance values from the Instrumentation Correlators FG24A-FG24N, the Instrumentation Brokers FG26A-FG26N, and/or the Jewel Instrumentation Broker (QoS Monitor) FG29. The instrumentation data that is received

contains (at a minimum):

39. the timetag regarding when the data was generated;
40. the hostname and IP address of the host where the application that the data is associated with is running;
41. the process id (pid) of the application that the data is associated with; and
42. the event number of the instrumentation message.

The event number of the instrumentation message specifies the type of instrumentation data that has been received and the hostname, IP address, and pid are used, in conjunction with the application data received from Program Control, to determine the specific application that the data is associated with.

If the contents of the instrumentation message match any of the application performance requirements that are currently being monitored by the QoS Manager, the data value is added to the proper requirement sliding window for the specified application. The sliding window algorithm is then checked to determine if the new sample triggered a violation of either the high or low sliding window. If a high threshold sliding window violation occurs and the application does not already have the maximum number of copies running, a determination is made as to whether performance can be best improved by starting a new application (scale up) or by moving an existing copy to a different host. The corresponding action recommendation will then be sent to the Resource Manager. In an exemplary case, the criteria for determining whether an application should be moved rather than scaled up is based on relative performance of the replicated applications. Thus, if one application is performing much worse [> 50%] than the other copies, the recommendation will be to move the application. Likewise, if the new sample triggers a low threshold sliding window violation, and the application has more than the minimum number of copies running, a recommendation will be sent to the Resource Manager FG42 requesting that the copy of the application that is experiencing the worst performance be scaled down.

It will be appreciated from the discussion above that when a copy of the QoS Manager is first started, it reads in the System Specification Files FG32 (via calls to System Specification Library (SSL) FG34), which contain the list of hosts that are known to be in the distributed environment and information on all applications that can be run in the distributed environment. The application-level information includes where specific applications can be run, which applications are scalable, which applications can be restarted, and any dependencies between applications.

It should also be mentioned that the Resource Manager FG42 receives updated application survivability specifications from the QoS Specification Control component. This information overrides the application survivability information that was initially loaded in from the System Specification Files for the specified application. The information is used by the Resource Manager FG42 to determine whether the specific application will be restarted if it fails at run-time.

As described above, the QoS Managers FG44A - FG44N communicates with the Resource Manager FG42, Program Control FG50, the QoS Specification Control (not shown), the QoS Monitor FG29, an Instrumentation Correlator (generally denoted FG24), a Graph Tool Instrumentation Display (generally denoted FG69), and the History Servers FG12A-FG12N using the RMComms middleware. The message formats and contents of each message that is exchanged between the QoS Managers FG44A - FG44N and these other functional components are described in greater detail in CD-Appendix F. Additional details regarding the timing and/or event trigger for each message is also described in the Appendix.

FG3: SYSTEM SPECIFICATION LANGUAGE & SYSTEM SPECIFICATION LIBRARY (SSL) FUNCTIONS

In order to effectively manage a pool of computing resources, the Resource Manager

FG42 requires some means or mechanism of determining the capabilities and configuration of the computing resources under its control, as well as the software components that need to be executed and the dependencies of these software components on both hardware and software resources. Additionally, the Resource Manager FG42 requires the capability to determine the expected mission-level and application-level requirements. Furthermore, the Resource Manager FG42 must be able to determine what control capabilities are available to be used to attempt to recover from fault or QoS violation conditions.

In order to address these needs, a System and Software Specification Grammar has been developed to capture the "static" information needed by the Resource Manager FG42 for effectively managing a pool of distributed resources. The grammar captures the following information:

- **Hardware and Operating Systems**
  - Hardware Configuration
  - Network Configuration
  - Operating System and Version
- **Software**
  - Systems, Subsystems, Applications, Processes
  - Resource Requirements
  - QoS Requirements (Events)
  - Survivability Requirements
  - Path Information: Structure and QoS Requirements

As part of the grammar development effort, a specification library has also been developed that parses the specification files and provides an API for accessing the specification information. It will be noted that the specification library was written in C++ and has been ported for all development platforms including Solaris 2.6, Solaris 2.7, Irid. 6.5, HP-

UX 10.20, Red Hat Linux 6.0, and Windows NT 4.0. The library advantageously can be used by substantially all of the Resource Management functional elements, including Program Control FG50, Resource Manager FG42, Path QoS Managers, Hardware Broker FG40, and History Servers FG12A-FG12N.

As illustrated in Fig. 3, the API library consists of a yacca file FG302 that defines the BNF grammar, a lex file FG304 that defines the tokens of the language, and a set of C++ classes FG306 that store the spec file information. The lex file FG304 is compiled with the GNU tool flex FG310 and it creates a C++ source file FG320. The GNU tool bison FG 312 compiles the yacca file FG302 and creates C++ source and header files FG322 and FG324. It will be noted that the lex source file FG304 includes the yacca header file FG322. The C++ compiler FG314 then compiles these two source files to create lex and yacca objects FG330 and FG332. The C++ compiler FG314 also compiles the C++ storage classes FG334. All of these objects are linked into a single library FG34 to be utilized by an application. Fig. 3 illustrates this process flow.

The Software Specifications Grammar (S.G.) provides techniques for describing the characteristics and requirements of dynamic, path-based real-time systems as well as providing abstractions to describe the properties of the software, such as hierarchical structure, inter-connectivity relationships, and run-time execution constraints. The S.G. also allows description of the physical structure or composition of the hardware such as LANs, hosts, interconnecting devices or ICD (such as bridges, hubs, and routers), and their statically known properties (e.g., peak capacities). Furthermore, the Quality-of-Service (QoS) requirements on various system components advantageously can be described.

At the highest level, a specification consists of a collection of software systems, hardware systems, and network systems. The language rules for specifying systems are described generally below and in detail in CD-Appendix G. The system specification language

hierarchy is shown below; selected details will be presented immediately following.

- **Software Specifications**
  - Application
    - Security
    - Configuration
      - Hardware Requirements
      - Startup Info
        - Dynamic Arguments
      - Shutdown Info
        - States
      - Dependencies
      - Initial Load Estimate
      - QoS Info
        - Survivability
        - Scalability
- **Hardware Specifications**
  - Host Info
  - Network Info
    - LANs
    - Network Devices (Interconnects)
- **Path Specifications**
  - Data Flow Graph
  - Data Flow Info
  - QoS Requirements

It will be appreciated that a software specification is a collection of software systems, each of which consists of one or more software subsystems. Specification files are provided by the developer to capture as much knowledge about their software system as possible. These files provide a model of the actual systems which can be used by the Resource Manager FG42 at run-time.

In contrast, an application is an executable program that can be started as an autonomous process on a host. Application attributes include all information necessary to startup and shutdown the application. Associated startup block and the shutdown blocks describe how to start and stop the application and includes information such as the directory and name of the application, command line options, and environment variable settings.

An application instantiates an SSL object by calling its constructor. This parses the spec files in the specified directory and populates the object hierarchy to provide the data to the application. The SSL class contains an SSL_Container member, that holds the spec file data in its lists and maps. All the systems from the spec files are contained in the appropriate list, software systems in the swSysList, hardware systems in hwSysList, and network systems in nwSysList. The pathList contains all the paths in the spec files. The hostList contains all the hosts in the spec files; this list is also available from the entries in hwSysList. The processList contains a list of processes from the CONFIGURATION block. Moreover, it should be noted that one or more configuration blocks can exist per application. For example, an application that runs on more than one platform would have multiple CONFIGURATION blocks with different platforms in each HARDWARE block.

The application startup block contains all the information necessary to, automatically or manually, start an application. This information includes supported hardware (host) type, operating-system type, and operating-system version(s). This may be further constrained by an optional list of the names of hosts that can run the application. The startup information also

Ex.1002 / Page 493 of 541
TESLA, INC.

includes the working directory for reading and writing data files, the name of the executable, and an ordered list of arguments that must be passed on the command line when the application is started. Last is a list of processes expected to be seen on the system when the application is running.

An application shutdown block indicates the command(s) to be used for termination of the application. A shutdown command may be a POSIX signal name or may be a shell script or batch file. Supported signals include SIGKILL, SIGQUIT, SIGHUP, SIGUSR1, SIGUSR2, SIGSTOP, SIGINT, and SIGTERM. The ShutdownTime parameter is the maximum time to wait for the an application to die gracefully before forcing the application to terminate via the SIGKILL signal.

Other blocks are available. For example, a dependency block indicates any dependencies the application may have with the startup and/or shutdown of other applications (e.g. it may be required that a particular application be started before another application can be started). It will be noted that the dependency block is used by both Application Control FG50 and the Resource Manager FG42 to determine whether or not it is safe to start an application, stop an application, or let an application continue to run.

The scalability specification for an application indicates whether an application can be scaled via replication. Scalable applications are programmed to exploit load sharing among replicas, and can adapt dynamically to varying numbers of replicas. The specification also indicates whether an application combines its input stream (which may be received from different predecessor applications and/or devices), and splits its output stream (which may be distributed to different successor applications and/or devices) are also specified. "Combining" and "splitting" are commonly called "forking" and "joining" in parallel computing paradigms.

Specification files advantageously can be provided to describe a given set of networks

that exist in a distributed runtime environment. A network system specification describes the LANs and ICD (interconnection devices such as switches, hubs and routers). A system consists of one or more subsystems. A subsystem may contain LANs (each with an associated peak bandwidth specification) and ICD (each containing a description of network membership).

Advantageously, a real-time QoS requirement specification includes timing constraints such as simple deadlines, inter-processing times, and throughputs. A simple deadline is defined as the maximum end-to-end path latency during a cycle from the beginning to the end of the path. Inter-processing time is defined as a maximum allowable time between processing of a particular element in the path. The throughput requirement is defined as the minimum number of data items that the path must process during a unit period of time. Each timing constraint specification may also include items that relate to the dynamic monitoring of the constraint. These include minimum and maximum slack values (that must be maintained at run-time), the size of a moving window of measured samples that should be observed, and the maximum tolerable number of violations (within the window).

CD-Appendix G described a specification grammar for declaring requirements on applications in a dynamic, distributed, heterogeneous Resource pool. The grammar allows the description of environment-dependent application features, which allows for the modeling and dynamic Resource management of such systems.

A common API was developed to allow Resource Management functions access to the information contained in the spec files. This is an object oriented API is, in an exemplary case, written in C++, with libraries ported to all supported platforms. The object is populated by parsing the spec files using the BNF grammar defined by lex and yacca syntax and compiled with GNU tools flex and bison, as discussed above. Actual population occurs in the semantic actions of the yacca file.

The SSL_System class is a generic class that can hold data for a software system, hardware system, or network system. The type member describes the type of system it contains. It also contains a pointer to its parent (it allows for nested systems of the same type), and a name of the system. The sysList contains its SSL_System children, and compList contains a list of the system's components (a list of hosts, for a hardware system for example).

Preferably, the Application Program Interface (API) for the System Specification Library (SSL) FG34 uses the C++ Standard Template Library for data structures such as linked lists and hash tables (maps). An application first instantiates the SSL object by calling its constructor with the name of the directory where the specification files reside. This object contains functions that allow setting this directory after calling its constructor (setSpecDir(directory name)), clearing the object of all currently held data (clear()), parsing a specific file (parseSpec(filename)), and rebuilding the object (rebuild(), implicitly clears the object first). Once instantiated, this object provides access to the data in the specification files. CD-Appendix G provides additional discussion regarding this aspect of the SSL. It will be appreciated that the SSL object provides methods that return all the data it contains. For example, the getSWSystems returns an STL list of all the software systems specified in the specification files. Each entry in this list provides its data by methods such as getSysName(), and the set of application components (ApplicationSpec) that make up the system. All data can be retrieved in this manner.

FG1: HOST AND NETWORK MONITORING FUNCTIONAL GROUP

As mentioned above, extensive monitoring capabilities are provided in the Resource Management architecture at the host and network levels. The information monitored includes statuses, configuration information, performance metrics, and detected fault conditions. Moreover, the Host and Network functional group FG1 consists of four components including:

1) **Host Monitors** FG10A-FG10N, that reside on each machine in the distributed environment and collect extensive operating system-level data for each host (CPU and memory usage, etc) and provides it to the History Servers via the RMComms TCPCommServer middleware.

2) **History Servers** FG12A-FG12N that collect data from the Host Monitors, maintain status and performance histories on each host in the distributed environment via an RMComms TCPCommClient, and provide this information to displays and other Resource Management components using an RMComms TCPCommServer.

3) A **Host Discovery** function FG14 that uses SNMP (Simple Network Management Protocol) calls and ping ICMP calls to determine when new hosts come on-line and if existing hosts go down and providing this information to Program Control via an RMComms TCPCommServer.

4) A **Remos Network Data Broker** FG16 that collects information on network link bandwidths from Carnegie Mellon University's SNMP-based Remos tool and passes this information by way of an RMComms TCPCommServer to the Host Load Analyzer component of the Resource Allocation Decision-Making subsystem.

It will be appreciated that Network information is collected by both the Remos broker FG16 and indirectly via the Host Monitors FG10A-FG10N. See Fig. 2. The Remos Broker FG16 accesses the Remos network information via the Remos API. As mentioned previously, Remos uses SNMP calls to the LAN switches and hosts. The Host Discovery function FG14 uses both SNMP and ICMP (ping) calls to each host A-N to determine if a new host(s) has (have) come on-line or previously discovered hosts have gone down. The Host Monitors FG10A-FG10N employ Operating System calls to gather host and network performance statistics. Internally, the History Servers FG12A-FG12N collect data from the Host Monitors FG10A-FG210N. The Monitoring functional group provides its information to the rest of the

Resource Management components using RMComms TCPCommServer objects, which are discussed in detail elsewhere. The Remos Broker FG16 sends data to the Host Load Analyzer FG40, the History Servers FG12A-FG12N send data to the Display functional group FG6 and Host Load Analyzer FG40, and the Host Discovery function FG14 provides Program Control FG50 with information on detected or faulted hosts. Additional details on these functional elements are provided immediately below.

FG10A-FG0N Host Monitors

For monitoring the status and performance of hosts, a Host Monitor process runs on each machine within the distributed environment. These Host Monitors FG10A_FG10N use operating system-level mechanisms to retrieve status, configuration, and performance information of each host A-N. The information retrieved includes 1) operating system version and machine configuration, 2) CPU configuration, status, and utilization, 3) memory configuration and usage, 4) network configuration, status, and utilization, 5) filesystem configuration, status, and utilization, and 6) process statuses including CPU, memory, network, and filesystem utilization for each process. While the Host Monitors are primarily responsible for monitoring the status of a particular host, they also provide information on network load as seen by a particular host. In the same manner, the Host Monitors FG10A_FG10N also provide information and statistics concerning any remotely mounted filesystems (e.g., NFS).

Preferably, the information the Host Monitors FG10A_FG10N collect is formatted into operating system-independent message formats. These message formats attempt to provide a pseudo-standardized set of state, status, and performance information which is useful to other components of the Resource Management architecture and such that other components do not have to be aware of or deal with the minor deltas between data formats and semantics. Since not all the state and performance data is available on every platform,

to indicate which information is available, a group of flags are set in the host configuration message indicating whether specific data items are valid on a particular platform.

It will be appreciated that the Host Monitors FG10A_FG10N have a very specific interface with the History Servers FG12A_FG12N. It periodically (once a second) sends its data to all History Servers connected to it (this is transparent, a property of the RMComms TCPCommServer); the History Server makes no requests to the Host Monitors.

More specifically, the Host Monitors FG10A-FG10N have been designed and implemented in C++. This decision allows for a completely modular design in which platform-specific code can be restricted to a small number of modules. This approach alleviates any of the problems associated with porting to various platforms. Currently there is support for Sun SPARC based architectures running Solaris 2.6 and 2.7, Silicon Graphics MIPS based architectures running IRID. 6.5., Hewlett Packard PA-Risc based architectures running HP1020, and Pentium based architecture running both WinNT 4.0 Workstation and Red Hat Linux 6.0. The Host Monitor source compiles under the native compilers provided by Sun Microsystems and Silicon Graphics for their respective platforms. The Gnu C++ compiler (version 2.8.1) may also be used on Hewlett Packard PA-Risc based architectures under HPUX 10.20 and Red Hat Linux. Microsoft Visual C++ compiles the Windows NT Host Monitor. All Host Monitors utilize the I/O library package supported by the Resource Management (RM) group under the NSWC's High Performance Distributed Computing (HiperD) initiative.

The Host Monitors FG10A-FG10N accumulate data on a periodic interval specified at invocation. System process table data is accumulated and then filtered to eliminate "uninteresting" processes (usually meaning processes belonging to user ID 0 or 1). It is important to note that system-wide data is accumulated and processed *before* the filtering stage, so as to insure a complete picture of system-wide performance. This system-wide data,

along with the filtered process list, is then made available to the I/O module for subsequent transmission to client applications.

FG12A-FG12N: History Servers

The History Servers FG12A_FG12N are responsible for collecting information from the Host Monitors and maintaining histories on the statuses, statistics, and performance of each host in the distributed environment. This information can be requested by other Resource Management functional group. Currently, the primary consumers of the status information are the Host Load Analyzer (Hardware Broker) FG40 component of the Resource Allocation Decision-Making functional group FG4, the Host Display(s) FG62A-FG62N, and the Path Display FG64. The Host Load Analyzer FG40 receives information on host configuration and loads (primarily CPU, memory, and network data) and uses this to assign host fitness scores. The Host Displays FG62A-FG62N receive and display current host status information, process status information, and network connectivity information. It should be mentioned that the Host Display can also request that the History Servers provide CPU load information, network load information, paging activity data, and memory utilization information which is used to drive line graph charts for specific hosts selected at the Host Display.

The History Servers FG12A-FG12N are designed so that multiple copies can be run simultaneously. Each History Server can be configured to either monitor all Host Monitors FG10A-FG10N or to monitor only a selected subset of the Host Monitors. It will be noted that the History Servers FG12A-FG12Ndetermine the list of hosts in the distributed environment that could potentially be monitored from the System Specification Library (SSL). In this manner, the History Servers FG12A-FG12N can be used to provide survivability (by having multiple History Servers FG12A-FG12N connected to each Host Monitor) and/or to perform load-sharing (with the History Servers FG12A-FG12N each monitoring only a subset

Ex.1002 / Page 500 of 541
TESLA, INC.

of the Host Monitors). The History Servers FG12A-FG12N can also be configured to periodically record history data to disk. These disk files can then be used for off-line analysis.

The History Server function of Resource Management acts as a data broker between daemons monitoring individual hosts, known as host monitors FG10A-FG10N, and other functional components of Resource Management. The host monitors collect performance information (such as CPU utilization and process status data) from hosts of various platforms (SGI, SUN, HP, Windows NT, and Linux). The host monitors use a RMComms TCPCommServer object to distribute this data. For further information, refer to the host monitor and RMComms documentation. The History Server s FG12A-FG12N collect and store this data from the host monitors FG10A-FG10N and distribute it to other Resource Management Clients, such as the Host Displays FG62A-FG62N, Graph Display FG69A-FG69N, Path Display FG64, and the Hardware Broker. FG40

Each History Server has two modes of operation relating to fault tolerance, scalability, and workload distribution between multiple instances of History Servers. The first mode determines at initialization (through command line arguments or default) the set of hosts to monitor, and this set remains static for the life of the History Server process. The second mode recognizes the existence of other History Server processes and coordinates between them. It allows for dynamic changing of the set of hosts each History Server monitors (example: two History Servers each monitoring half of the hosts, a third History Server starts, and all three History Servers reconfigure to each monitor one third of the hosts.) This also allows History Servers to preserve the data it collected by sending it to the others, providing fault tolerance.

The History Server function is written in C++ with an object-oriented design. The main routine processes the command line arguments, retrieves the list of hosts to monitor using an SSL object, instantiates the main History_Server object, and spawns the Collector,

Distributor, Communicator, and Display thread. These threads share the main History_Server object. The Collector thread is responsible for collecting and storing data from the host monitors. The Distributor thread processes requests from RM Clients. The Communicator thread waits for events with other History Servers and takes appropriate actions, including triggering the Display thread to update the History Server Display.

FG14 - Host Discovery

The Host Discovery function FG14 advantageously can use a Perl script that makes SNMP (Simple Network Management Protocol) calls and ICMP ping calls. These calls are used to periodically scan each subnet and host address in the distributed environment to attempt to determine whether there have been any host status changes. In an exemplary case, the list of hosts and subnets that are to be monitored is read in from a file.

The host discovery FG14 issues MIB-II SNMP queries to obtain information on the hosts A-N on the network. When a new host is first detected, the new host's operating system configuration is queried via SNMP calls. Information on the newly discovered host and its operating system configuration is then sent to the Program Control function FG50. Likewise, when a host fails to respond to multiple SNMP and ping queries, a message indicating that the host appears to have gone down is sent to the Program Control function.

The Host Discovery function FG14 interfaces with Program Control FG50 using a C++ wrapper class around the Perl script. This wrapper class contains an RMComms TCPCommServer, making the data collected by the SNMP calls available to the rest of the Resource Management components.

FG16 - Remos Network Data Broker

Ex.1002 / Page 502 of 541
TESLA, INC.

The final functional component of the Host and Network Monitoring functional group is the Remos Network Data Broker FG16 which receives information on network link bandwidth and network link bandwidth utilization from the SNMP-based Remos network monitoring tool, as shown in Fig. 2 and/or Fig. H4. The network information is accessed via the Remos API library and is then sent on to the Host Load Analyzer (Hardware Broker) function FG40 of the Resource Allocation Decision-Making functional groupFG4 using an RMComms TCPCommServer. Remos works by using SNMP to query the switches (via the bridge collector) to collect information on network configuration as well as bandwidth utilization on each link and also issues SNMP MIB-II queries to each host to collect the host's view of network utilization. The network information received from Remos consists of the maximum potential bandwidth and the current bandwidth utilization on specific host network links.

The Remos Broker FG16 provides the following information about the network link for each host. The data is sent to the Host Load Analyzer (Hardware Broker) approximately every 2 seconds. The Remos Broker FG16 uses configuration files listing specific hosts and switches that should be queried.

The functions implemented by Host Monitor functional group FG1 have been designed to provide a system monitoring capability not normally supplied by standard SVR4 or BSD Unix services. Such services include cross-platform reporting of system process loading , CPU performance, network performance and periodic status summary reporting. The Host Monitors were developed to support efforts by the Hiper-D Resource Management group, attempting to provide a common set of OS level parameters useful for assessing host and network load and status, for supporting Resource allocation/reallocation algorithms, and attempting to provide a minimally intrusive, close to real-time capability for gathering this data.

Host Discovery Design

The Host Discovery function FG14 of the Resource Management architecture provides Resource discovery of hosts on a network. It identifies new hosts that come online or previously known hosts that have gone offline. The Host Discovery component can determine the hostname, the operating system name and version, and in some cases the machine architecture and manufacturer of a newly discovered host. This information is sent to Program Control so the new host can be added to the pool of resources.

The Host Discovery functional element FG14 consists of a Perl script that contains the Resource discovery functionality, and a C++ object that receives the output of the Perl script and provides this information to Program Control via an RMComms TCPCommServer connection. This is described in CD-Appendix H. More specifically, the Perl script *host_discovery.pl* issues ICMP (ping) calls and MIB-II SNMP queries to discover new hosts. On initialization, the script populates a data structure called *Net_info* for each of the networks (subnets) it needs to monitor. Currently this information is hard-coded, the subnet is defined as 172.30.1, and the lower and upper limits for the host are 1 and 254 respectively. It then initializes the global variables for the server host and port, network domain, and the executable path for the ping (fping) command

The *host_discovery.pl* script establishes a baseline of existing hosts using the current set of hosts that answer the fping call. For each network/subnet defined in its list of *Net_info* (Net_info.pm) data structures, it calls fping and builds a list of IP addresses of hosts that answered the ping, known as reachable hosts, and a list for those hosts that did not answer the ping. For each reachable host, a *Host_info* (Host_info.pm) data structure is populated to store the host's information. (Key fields in the *Host_info* data structure include IP address, hostname, operating system and version, architecture class, and manufacturer.) Since the IP address of the reachable host is known, a call to gethostbyaddr() is used to get the hostname.

Other information for the host is obtained by making a MIB-II (Management Interface Base version 2) system Group (Object ID 1.3.6.1.2.1.1.1.0) SNMP call to the SNMP agent on each reachable host. This SNMP query returns information on the configuration of a specific network device (in this case, the configuration of each reachable host).

The *host_discovery.pl* script makes SNMP calls by using subroutines freely available for public use (freeware), created by Simon Leinen. These subroutines are contained in the files BER.pm and SNMP_Session.pm. The *SNMP_Session* is configurable for specifying timeouts and number of retries before declaring a host unavailable, and for specifying the SNMP Object Id (OID).

Additional general and specific details regarding functional elements of the Host and Networking functional group FG1 are provided in CD-Appendix H.

FG2: Instrumentation functional group

As mentioned above, the NSWCDD Instrumentation System provides general-purpose application event reporting and event correlation capabilities. The Instrumentation system forms an architecture that allows instrumented application data to be easily accessible by other components of the Resource Management architecture. The major functional components of the Instrumentation System architecture are the following:

1) The **Instrumentation API Libraries,** which are linked with the applications and provide the function call interfaces by which the application sends instrumentation data.

2) An **Instrumentation Daemon**, one copy of which resides on each host in the distributed environment and is responsible for reading instrumentation data sent by the applications, reformatting the data into instrumentation event messages and sending the messages to the Instrumentation Collectors.

Ex.1002 / Page 505 of 541
TESLA, INC.

3) The **Instrumentation Collectors**, which connect to the Instrumentation Daemons on each host and receive instrumentation messages from all hosts. The Collectors forward received messages to the Instrumentation Correlators and Instrumentation Brokers.

4) The **Instrumentation Correlators**, which receive instrumentation messages from the Instrumentation Collectors and provide grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other Resource Management components.

5) The **Instrumentation Brokers**, which receive instrumentation messages from the Instrumentation Collectors and perform task-specific reformatting and data manipulation for driving displays or other Resource Management components.

(6) The **Jewel Instrumentation Broker (QoS Monitor)**, which is a legacy component that can receive instrumentation data from either the open source Jewel instrumentation package or from the Instrumentation Collectors. The QoS Monitor performs task-specific message reformatting and data manipulation for driving displays and the QoS Managers.

Instrumentation API Library

The applications link in the Instrumentation API Library and make API call to construct and send out instrumentation event messages. Three separate APIs are provided for use by the applications: 1) a printf()-style API which allows the code to format, build, and send instrumentation data with a single function call, 2) a buffer-construction-style API where the multiple function calls are made to construct the instrumentation buffer iteratively, one data element per call, and 3) a Jewel function call API based on the existing API provided by the Jewel instrumentation package (an open-source package produced by the German National Research Center for Computer Science). The first two APIs are the

preferred programming interfaces and take advantage of several key new instrumentation features. It will be appreciated that the Jewel API is provided solely for backwards compatibility with existing instrumented application code and is implemented as a set of wrappers around the printf()-style API. All three APIs are supported for C and C++. Ada bindings have been produced for the buffer-construction-style API and the Jewel function call API.

The instrumented data is sent from the application to the Instrumentation Daemon on the same host. The current mechanism for data transfer is via UNIX FIFO IPC (inter-process communication) mechanisms. The FIFO mechanism was chosen based on reliability, low overhead, and ease of implementation. Future implementations of the Instrumentation system may explore alternate data passing mechanisms including shared message queues.

Instrumentation Daemon

An Instrumentation Daemon resides on each host in the distributed environment. The Instrumentation Daemon is interrupted when new data is written to the FIFO. The Instrumentation Daemon reads the data from the FIFO and reformats the data into the standard internal Instrumentation message format and sends the data to each of the Instrumentation Collectors that are currently active. (For future implementations, an event request filtering mechanism will be implemented so that specific event messages will only be sent to those Instrumentation Collectors that have requested the message.)

Instrumentation Collectors

The Instrumentation Collectors receive instrumentation messages from the Instrumentation Daemons on each host in the distributed environment. Currently, the Instrumentation Collectors send every instrumentation message to all Instrumentation Brokers and Instrumentation Correlators that have connected to the Instrumentation

Collector. (For future implementations, an event request filtering mechanism will be implemented so that specific event messages will only be sent to those Instrumentation Brokers and Instrumentation Correlators that have requested the message. For now, the Instrumentation Collector serves as a pass-through server for instrumentation messages. The Instrumentation Collector does supports architecture scalability in the sense that without the Instrumentation Collectors, each Instrumentation Broker and Instrumentation Correlators would need to maintain connections to the Instrumentation Daemons on every host.)

Instrumentation Correlators

The Instrumentation Correlators provide grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other Resource Management components. Each Correlator reads in a user-specified correlation grammar file that is interpreted at run-time by the Correlator's instrumentation correlation engine.

Instrumentation Brokers

The Instrumentation Brokers are task-specific applications built around a common code package. The Instrumentation Brokers receive instrumentation messages from the Instrumentation Collectors, filter all received instrumentation messages to find the messages of interest, and perform task-specific message data reformatting and manipulation for driving other components such as displays or other Resource Management components. The Instrumentation Broker approach allows for instrumentation data sources to be quickly integrated for test, display, and debugging purposes. (As the Instrumentation Correlator grammar and correlation engine mature in future releases, it is anticipated that the Instrumentation Broker approach will be used less frequently.)

Jewel Instrumentation Broker (QoS Monitor)

The Jewel Instrumentation Broker (hereafter referred to the QoS Monitor) is a legacy architecture component that served as a broker between the Jewel instrumentation package components and Resource Management components and displays. The QoS Monitor was responsible for polling the Jewel Collector components to retrieve application event messages. These messages were then reformatted and used to drive several displays and the QoS Managers. The Jewel instrumentation package has now been replaced in all applications, however the message reformatting capabilities of the QoS Monitor have been maintained so that several displays and the existing QoS Manager interface do not have to be upgraded immediately. The QoS Monitor component has been modified so that it receives instrumentation data from both Jewel and the Instrumentation Collectors.

Middleware

The RMComms middleware package, which is described in the RMComms Middleware Design Report, provides the internal message passing interfaces between the Resource Management components connected via the network. The middleware provides for automatic location-transparent many-to-many client-server connections. Low overhead, reliable message passing capabilities are provided. Registration of message handler callback functions for specified requested message types are provided with the message handler functions being invoked when messages arrive. Registration of connection status callback functions which are invoked when either new connections are made or existing connections are broken is also provided. The middleware package also allows for multiple client and server objects to be instantiated in the same application, is thread-safe, and provides an easy-to-use object-oriented API through which all capabilities are accessed.

Additional details regarding the Instrumentation functional group FG2 are provided in CD-Appendix I.

FG42: Resource Manager

The Resource Manager 42 is the primary decision-making component of the Resource Management toolkit. It is responsible for: 1) responding to application and host failures by determining if and what recovery actions should be taken, 2) determining if and where to place new copies of scalable applications or which scalable applications should be shutdown when the QoS Managers FG44A-FG44N indicate that scale-up or scale-down actions should be taken based on measured application performance, 3) determining where new applications should be placed when requested to do so by Program Control, and 4) determining which and how many applications should run based on application system (mission) priorities. In order to accomplish these tasks, the Resource Manager 42 maintains a global view of the state of the entire distributed environment including status information on all hosts, networks, and applications. In addition, the Resource Manager 42 also calculates software and hardware readiness metrics and reports these readiness values for display purposes. Figure 1 shows the connectivity and high-level data flow between the Resource Manager 42 and the other Resource Management-related components.

The Resource Manager 42 receives status and failure information about hosts, networks, and applications from Program Control. This information includes periodic status updates as well as immediate updates when statuses change such as a new host being detected or an application failing. In the case of applications going down, information as to whether the applications were shutdown on purpose or whether they failed is also sent. Program Control also issues requests to the Resource Manager 42 when new applications need to be dynamically allocated and when Program Control determines that the Resource Manager 42 needs to assess and attempt to resolve inter-application dependencies (such as an application which needs to be running prior to starting up another application).

The Resource Manager 42 responds to faulted applications and hosts by determining whether the failed applications can and should be restarted and attempting to determine where (and if) there are hosts available that the application can run on. When a decision is

made by the Resource Manager 42, a message is sent to Program Control specifying what application to start and where to put it. The same general mechanism is used when Program Control requests that the Resource Manager 42 determine where to start new applications and/or how to resolve inter-application dependencies; the Resource Manager 42 responds with orders indicating what applications to start and where to start them. The Resource Manager 42 also sends application shutdown orders to Program Control requesting that certain application be stopped; this can occur when the QoS Managers FG44A-FG44N indicate that certain scalable applications have too many copies running or when application system priority changes (to lower priorities) occur resulting in scaling back the application system configuration.

The Resource Manager 42 receives host load and host fitness information on all known hosts from the Hardware Broker 40 (Host Load Analyzer). This information include overall host fitness scores, cpu-based fitness scores, network-based fitness scores, and memory and paging-based fitness scores along with the SPEC95 ratings of the hosts. This information is received approximate once a second and includes information on all known hosts in the distributed system. These scores are used by the Resource Manager 42 for determining the "best" hosts for placing new applications when: 1) responding to requests from the QoS Managers FG44A-FG44N to scale up additional copies of an application, 2) attempting to restart failed applications, 3) responding to requests to dynamically allocate certain applications, and 4) responding to application system (mission) priority changes which require scaling up additional applications.

The Resource Manager 42 receives requests from the QoS Managers FG44A-FG44N for scaling up, moving, or scaling down specific applications. The Resource Manager FG42 responds to these requests by determining whether the request should be acted upon and, if so, determines the specific action to take and issues orders to Program Control to start up or shutdown specific applications on specific hosts. The QoS Managers FG44A-FG44N are responsible for monitoring specific system performance metrics (e.g., quality of service, or

QoS, requirements) via instrumentation and determining if performance can be improved by scaling up or moving certain applications. When this occurs, the QoS Managers FG44A-FG44N send a request to the Resource Manager FG42 indicating that a new copy of a specific application should be started. If the QoS Managers FG44A-FG44N determine that the performance of a scalable application can be improved by moving an application, a scale up request is first sent to the Resource Manager FG42 and when the new application has been started, a scaledown request is then sent to the Resource Manager. Also, when the QoS Managers FG44A-FG44N determine that there are more copies of scalable application running then are needed, requests to shutdown specific applications are sent to the Resource Manager FG42. The division of responsibility is that the QoS Managers FG44A-FG44N determine what actions would potentially improve performance, but the Resource Manager FG42 has final authority to determine whether to implement the requested actions.

When the Resource Manager FG42 is first started, it reads in the System Specification Files (via System Specification Library, SSL, calls) which contain the list of hosts that are known to be in the distributed environment and information on all applications that can be run in the distributed environment. The System Specification Files also include application-level information including where specific applications can be run, which applications are scalable, which applications can be restarted, and any dependencies between applications.

The Resource Manager FG42 can also receive updated application survivability specifications from the QoS Specfication Control component. This information overrides the application survivability information that was initially loaded in from the System Specification Files for specified applications. The information is used by the Resource Manager FG42 to determine whether the specific applications will be restarted if they fail at run-time.

The Resource Manager FG42 sends application system and hardware system readiness and system (mission) priority information to the Readiness Broker and to the Globus Broker. The Readiness Broker is responsible for driving a GUI/display which shows the current readiness data and allows the system (mission) priorities to be changed and sent back to the Resource Manager FG42. The Globus Broker provides basically the same functionality except that only a high-level subset of the readiness data provided to the Readiness Broker is provided to the Globus Broker. The readiness information sent to the Readiness Broker consists of readiness values for each application, application subsystem, and application system defined in the System Specification Files. The readiness scores are currently based on the status (up/down) of the applications within a system or subsystem along with the percentage of potential copies of scalable applications that are currently running. Host and network readiness scores are also calculated and are the scores are determined based on the host load information and host fitness scores received from the Hardware Broker 40.

The Resource Manager FG42 also sends information about allocation and reallocation decisions to the Resource Management Decision Review Display. Information on the decision that was made, what event the decision was in response to, and how long it took to both make the decision and implement the decision are sent to the display. In addition, information about the top choices for where an application could have potentially been placed is also sent (if applicable); this information includes the host fitness scores for the selected host and the next best host choices which could have been selected.

See CD-Appendix M for additional details regarding Resource Manager FG42.

In the Background Section of the application, the reader may have interpreted the sentence "The present invention relates generally to resource management systems by which networked computers cooperate in performing at least one task too complex for a single computer to perform" to indicate that the Resource Management Architecture is limited to

such appications. Thus, while the Resource Management Architecture generally supports tasks distributed across multiple hosts, it is not limited to only those tasks that must be distributed due to the inability to run them on a single machine. Moreover, the Resource Management functional elements advantageously could be used to control a set of applications which all run on the same machine while still providing monitoring, fault tolerance, etc. (albeit that this is not the normal or even the intended configuration).

Fig. 15 is a block diagram of a CPU-based system 400, corresponding to one or more of the hosts A-N. The system 400 includes a central processing unit (CPU) 402, e.g., a microprocessor, that communicates with the RAM 412 and an I/O device 408 over a bus 420. It must be noted that the bus 420 may be a series of buses and bridges commonly used in a processor-based system, but for convenience purposes only, the bus 420 has been illustrated as a single bus. A second I/O device 410 is provided in an exemplary case. The processor-based system 400 also includes a primary memory 412, an additional memory 414, which could be either a read-only memory (ROM) or another memory device, e.g., a hard drive or the like. The CPU- based system may include peripheral devices such as a floppy disk drive 404, a compact disk (CD) ROM drive 406, a display (not shown), a key board (not shown), and a mouse (also not shown), that communicate with the CPU 402 over the bus 420 as is well known in the art. It will be appreciated that the either one of the memories 412 or 414 advantageously can be employed to store computer readable instructions for converting the general purpose system 400 into one of the host A-N. It will also be appreciated that the nature of the distributed environment permits the necessary application and API's needed to implement the Resource Management Architecture to9 be stored anywhere on the network.

Table III

APPENDIX	TYPE	DESCRIPTION
A	Attached	Resource Management Architecture Function List

B	Attached	Standard Instrumentation Message Format
C	Attached	API Listing for RMComms
D	CD	Resource Manager Interface Messages
E	CD	Host Load Analyzer (Hardware Broker) Function
F	CD	Quality-of-service (QoS) Manager Function
G	CD	Fg3: System Specification Language & System Specification Library (SSL) Functions
H	CD	Host And Network Monitoring Functional Group
I	CD	Instrumentation Functional Group
J	CD	Display Functional Group
K	CD	Rmcomms Network Communication Middleware Design
L	CD	System Readiness Display
M	CD	Resource Manager Rm42
N	CD	Instrumentation Graph Tool
O	CD	Host Discovery Function
P	CD	Instrumentation Application Programming Interface (API)
Q	CD	Program Control Application Controller
R	CD	Program Control Display
S	CD	Program Control Functional Group
T	CD	QoS Manager
U	CD	Resource Allocation Decision-making Functional Group

Table III provides a listing of the Appendicies included for all purposes in the application. It will be noted that the majority of the listed Apendicies are provided on the CD-ROM filed concurrently with the application. In addition, the CD-ROM also includes the source code listing for the Resource Management Architecture according to the present invention.

Although presently preferred embodiments of the present invention have been described in detail herein, it should be clearly understood that many variations and/or modifications of the basic inventive concepts herein taught, which may appear to those skilled in the pertinent art, will still fall within the spirit and scope of the present invention, as defined in the appended claims.

Express Mail Label No. EV339908925US

WHAT IS CLAIMED IS:

1. In a distributed environment comprised of hosts instantiating copies of a scalable application, a resource management device generating signals which start up, shutdown or move a selected one of the copies responsive to first information regarding performance of all copies of the scalable application and second infomation regarding performance of the hosts.

Ex.1002 / Page 517 of 541
TESLA, INC.

ABSTRACT OF THE DISCLOSURE

A reseource manager for a distributed environment including hosts instantiating copies of a scalable application, generates signals which start up, shutdown or move a selected one of the copies responsive to first information regarding performance of all copies of the scalable application and second infomation regarding performance of the hosts.

Docket No. NCN-83017

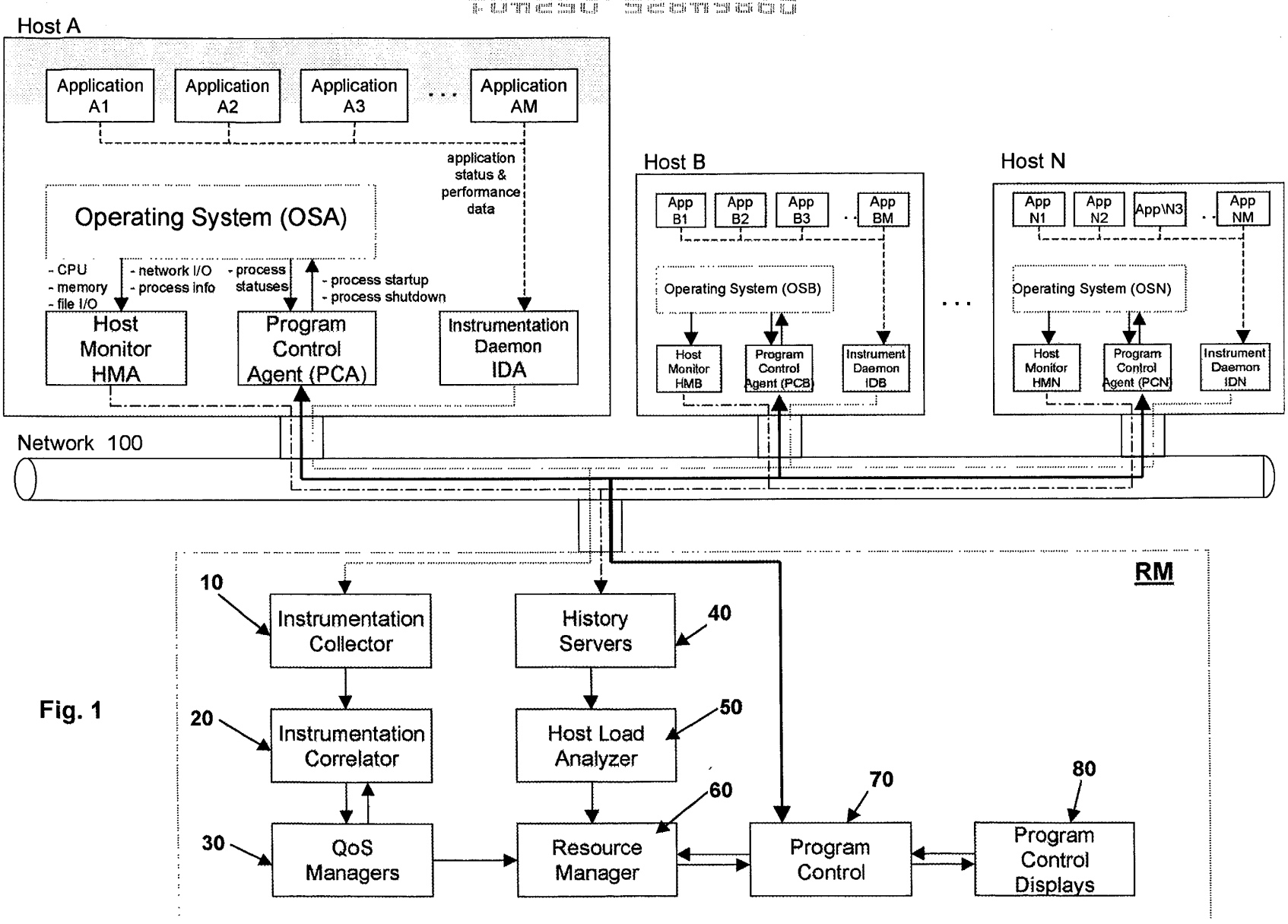

Host A
Application A1
Application A2
Application A3
Application AM
application status & performance data
Operating System (OSA)
- CPU
- memory
- file I/O
- network I/O
- process info
- process statuses
- process startup
- process shutdown
Host Monitor HMA
Program Control Agent (PCA)
Instrumentation Daemon IDA
Host B
App B1
App B2
App B3
App BM
Operating System (OSB)
Host Monitor HMB
Program Control Agent (PCB)
Instrument Daemon IDB
Host N
App N1
App N2
App\N3
App NM
Operating System (OSN)
Host Monitor HMN
Program Control Agent (PCN)
Instrument Daemon IDN
Network 100
RM
10
Instrumentation Collector
History Servers
40
20
Instrumentation Correlator
Host Load Analyzer
50
60
70
80
30
QoS Managers
Resource Manager
Program Control
Program Control Displays

Fig. 1

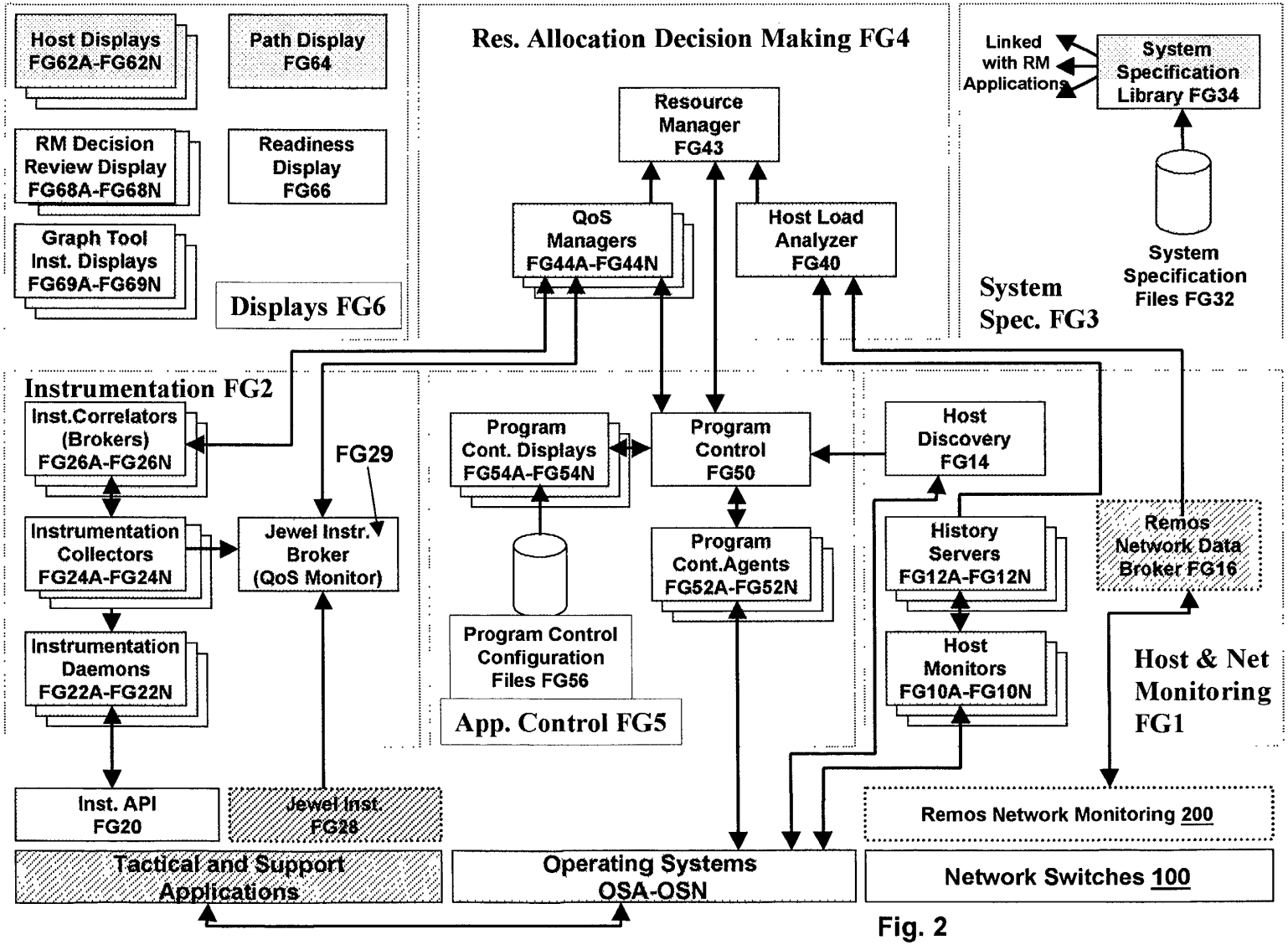

Host Displays FG62A-FG62N
Path Display FG64
RM Decision Review Display FG68A-FG68N
Readiness Display FG66
Graph Tool Inst. Displays FG69A-FG69N
Displays FG6
Res. Allocation Decision Making FG4
Resource Manager FG43
QoS Managers FG44A-FG44N
Host Load Analyzer FG40
Linked with RM Applications
System Specification Library FG34
System Specification Files FG32
System Spec. FG3
Instrumentation FG2
Inst.Correlators (Brokers) FG26A-FG26N
FG29
Instrumentation Collectors FG24A-FG24N
Jewel Instr. Broker (QoS Monitor)
Instrumentation Daemons FG22A-FG22N
Program Cont. Displays FG54A-FG54N
Program Control FG50
Program Cont.Agents FG52A-FG52N
Program Control Configuration Files FG56
App. Control FG5
Host Discovery FG14
History Servers FG12A-FG12N
Host Monitors FG10A-FG10N
Remos Network Data Broker FG16
Host & Net Monitoring FG1
Inst. API FG20
Jewel Inst. FG28
Tactical and Support Applications
Operating Systems OSA-OSN
Remos Network Monitoring 200
Network Switches 100

Fig. 2

Patented Application

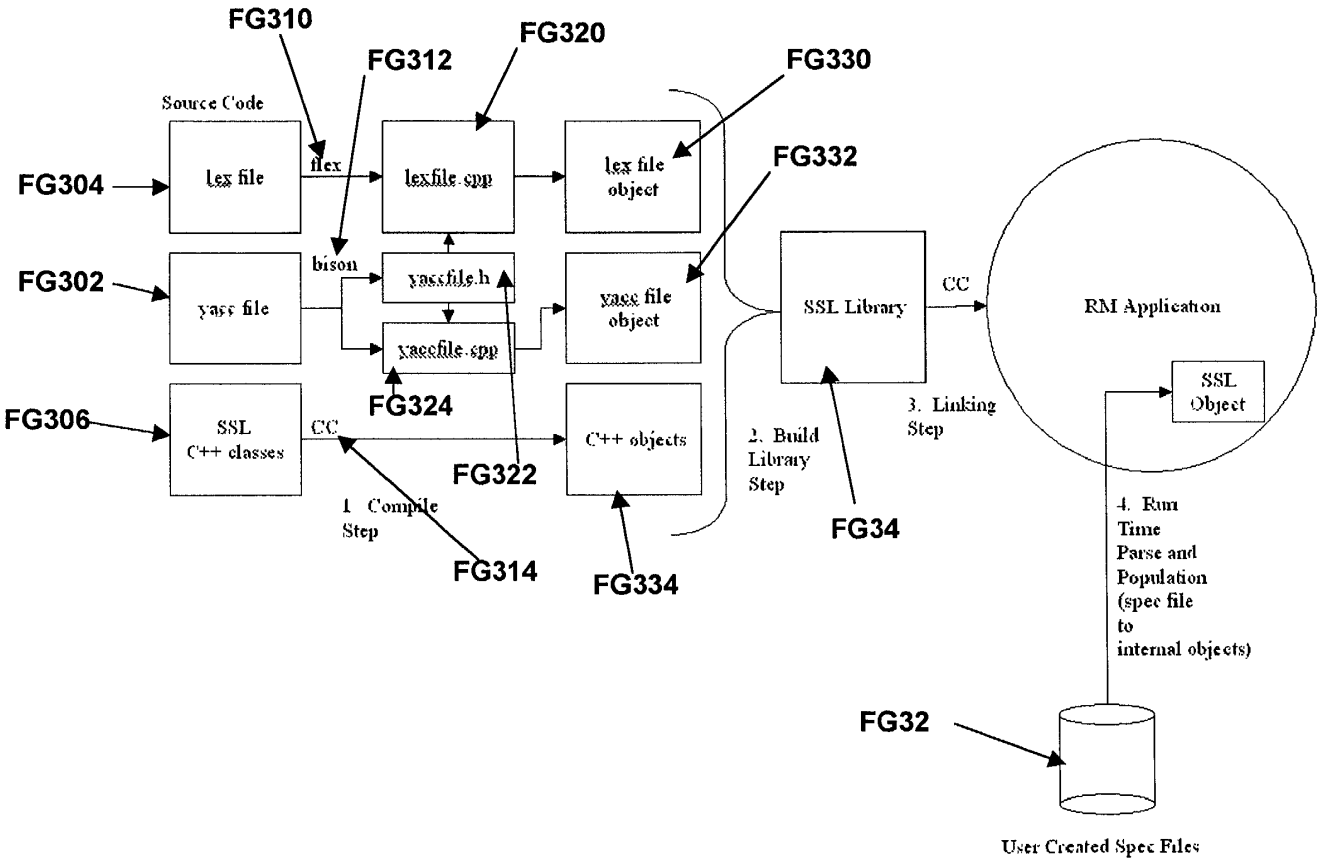


Fig. 3

Petitioner's Exhibit 1002

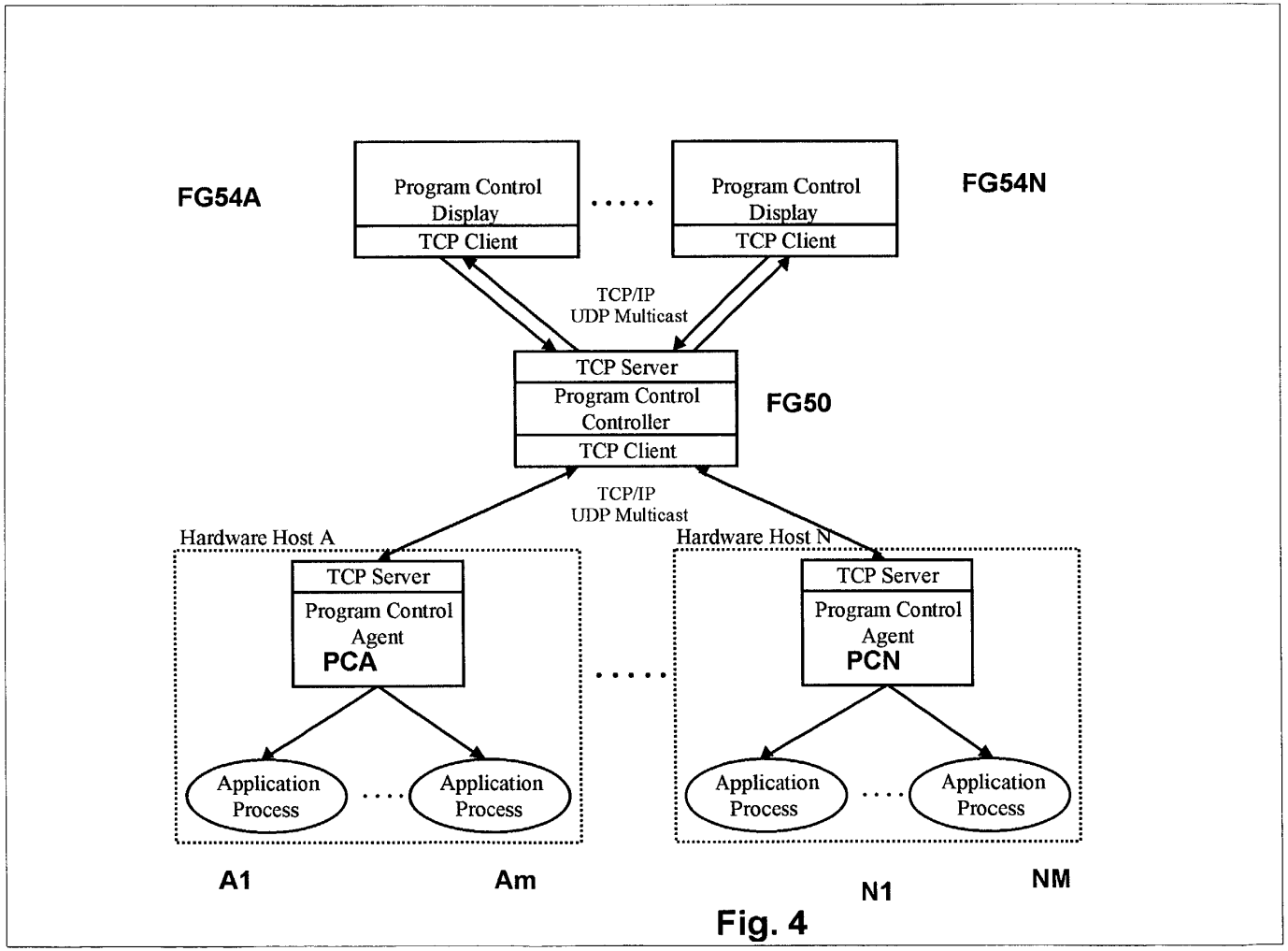


Fig. 4

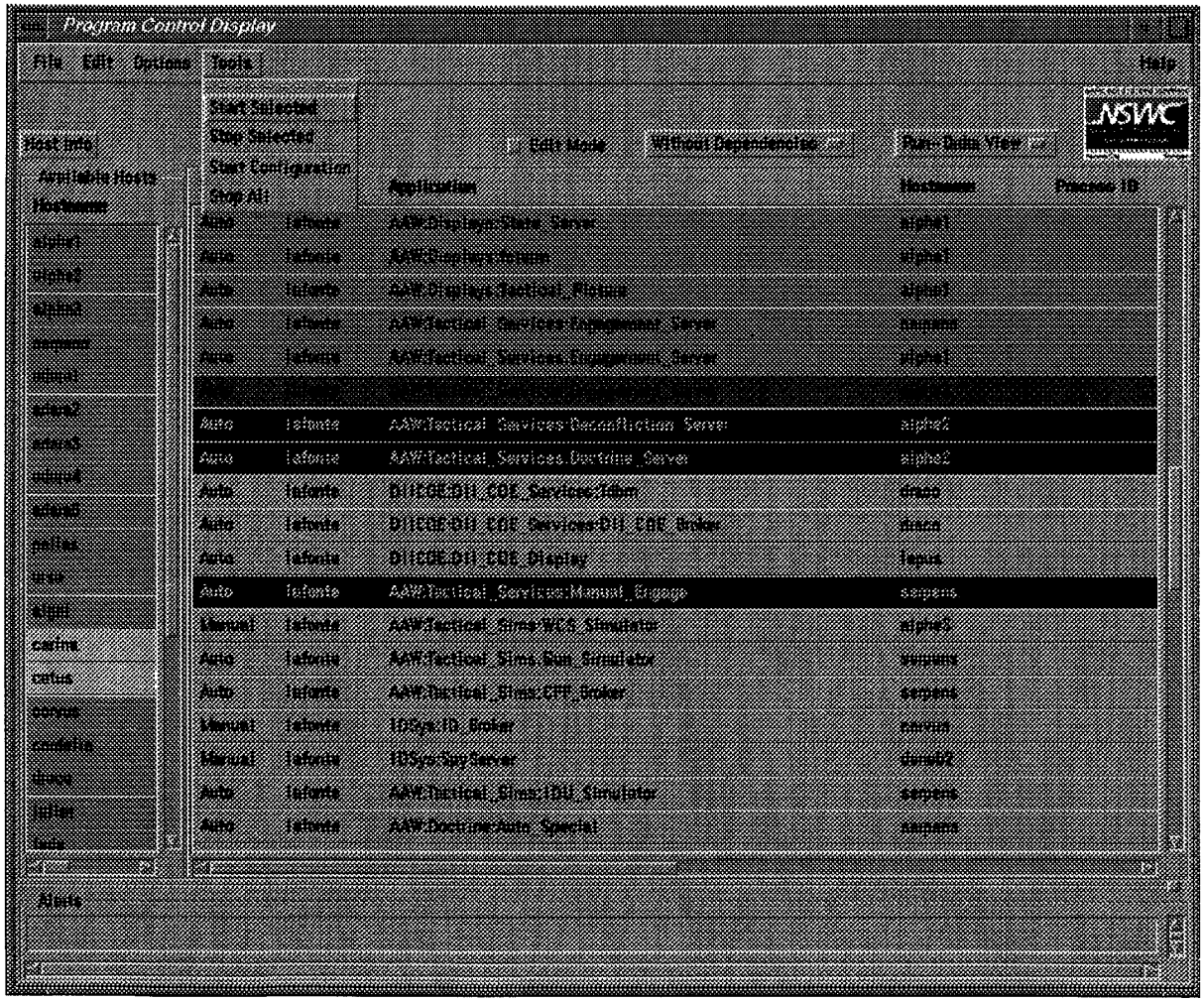

Program Control Display
File
Edit
Options
Tools
Help
Start Selected
Stop Selected
Start Configuration
Stop All
Host Info
Application
Hostname
Process ID
Alerts

Fig. 5

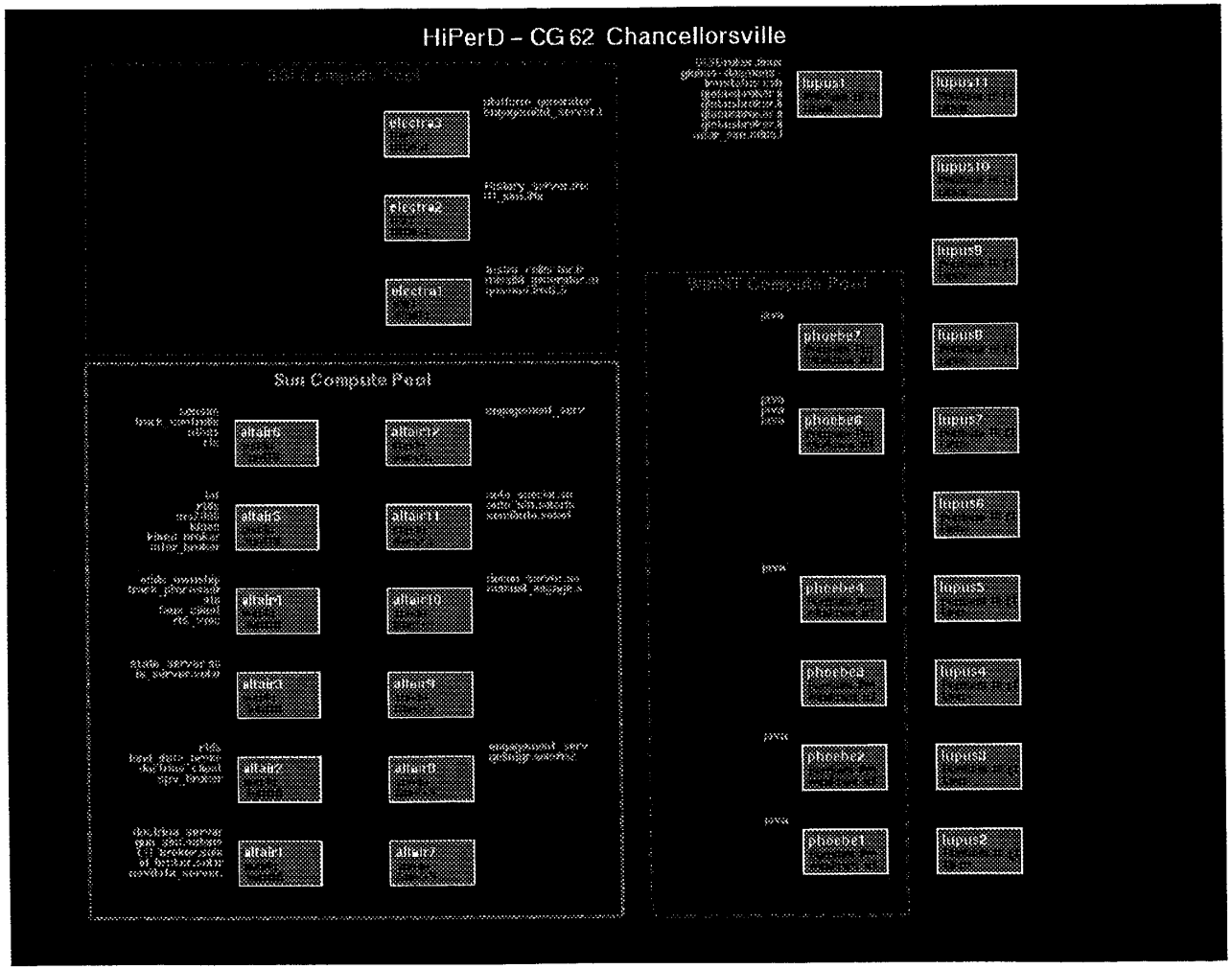

HiPerD – CG 62 Chancellorsville
Sun Compute Pool
electra3
electra2
electra1
altair6
altair7
altair5
altair11
altair1
altair10
altair4
altair9
altair2
altair8
altair3
altair12
phoebe7
phoebe6
phoebe4
phoebe3
phoebe2
phoebe1
lupus1
lupus11
lupus10
lupus9
lupus8
lupus7
lupus6
lupus5
lupus4
lupus3
lupus2

Fig. 6

Tesla Motors

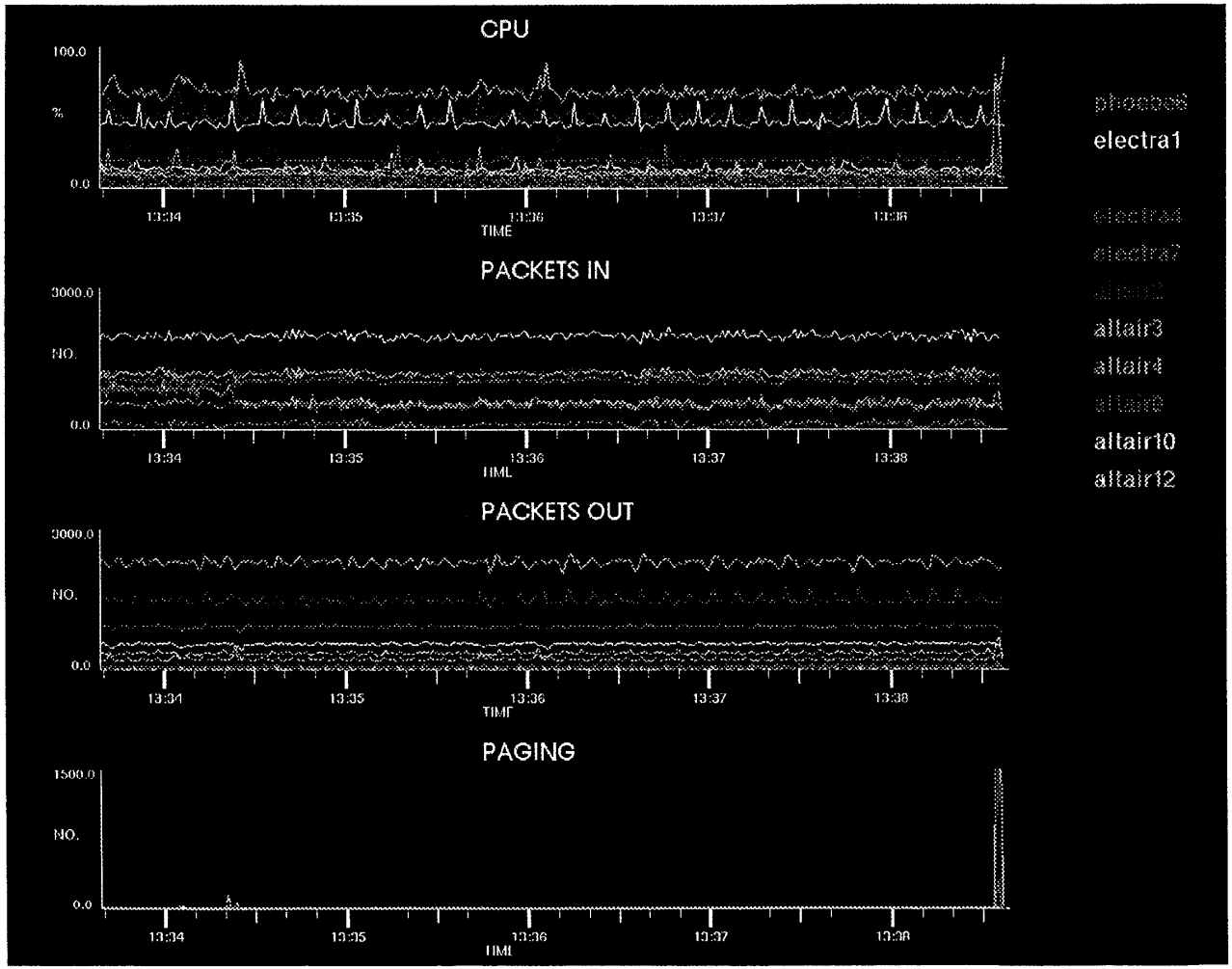


Fig. 7

Protected Specimen

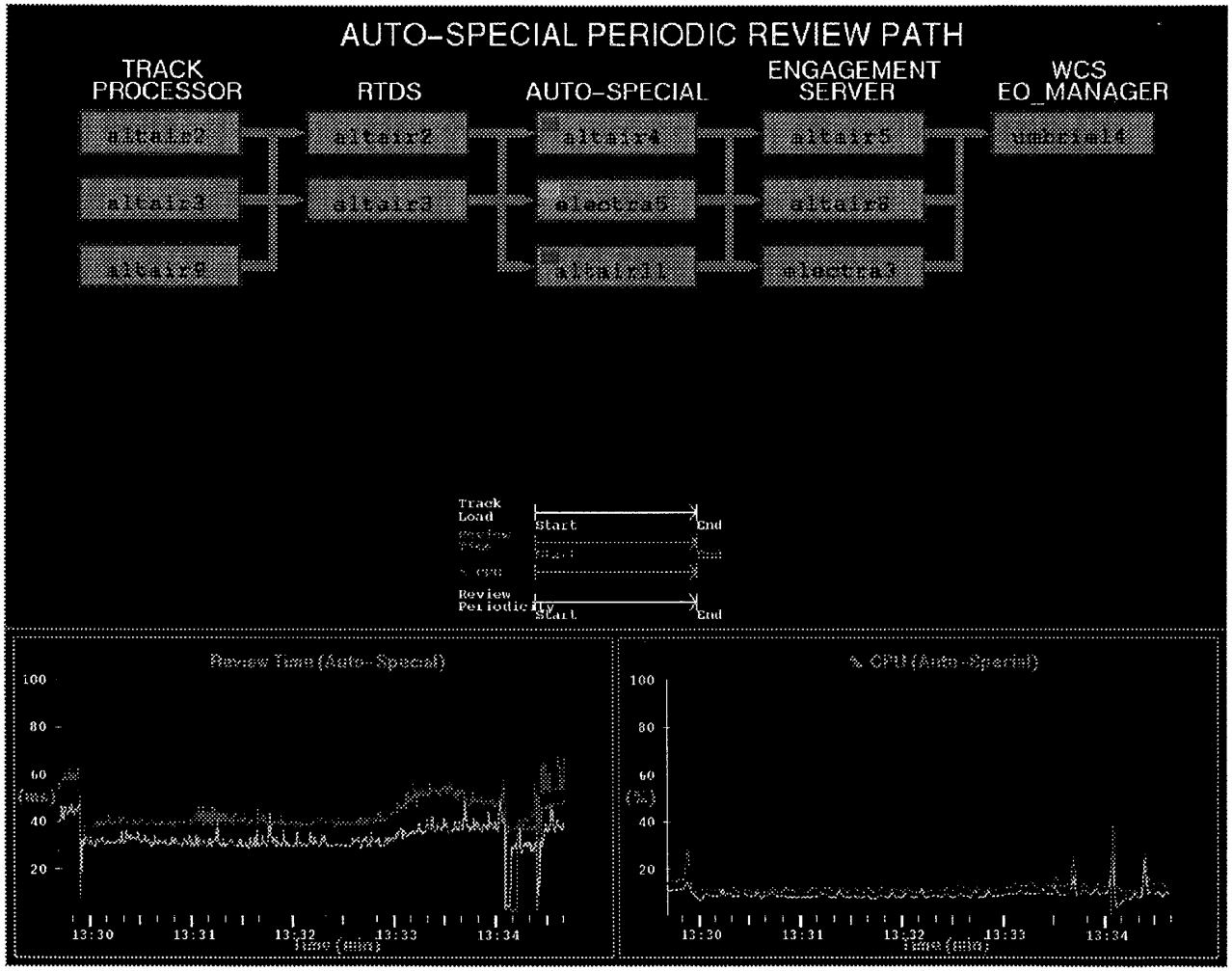


Fig. 8

Patent Application

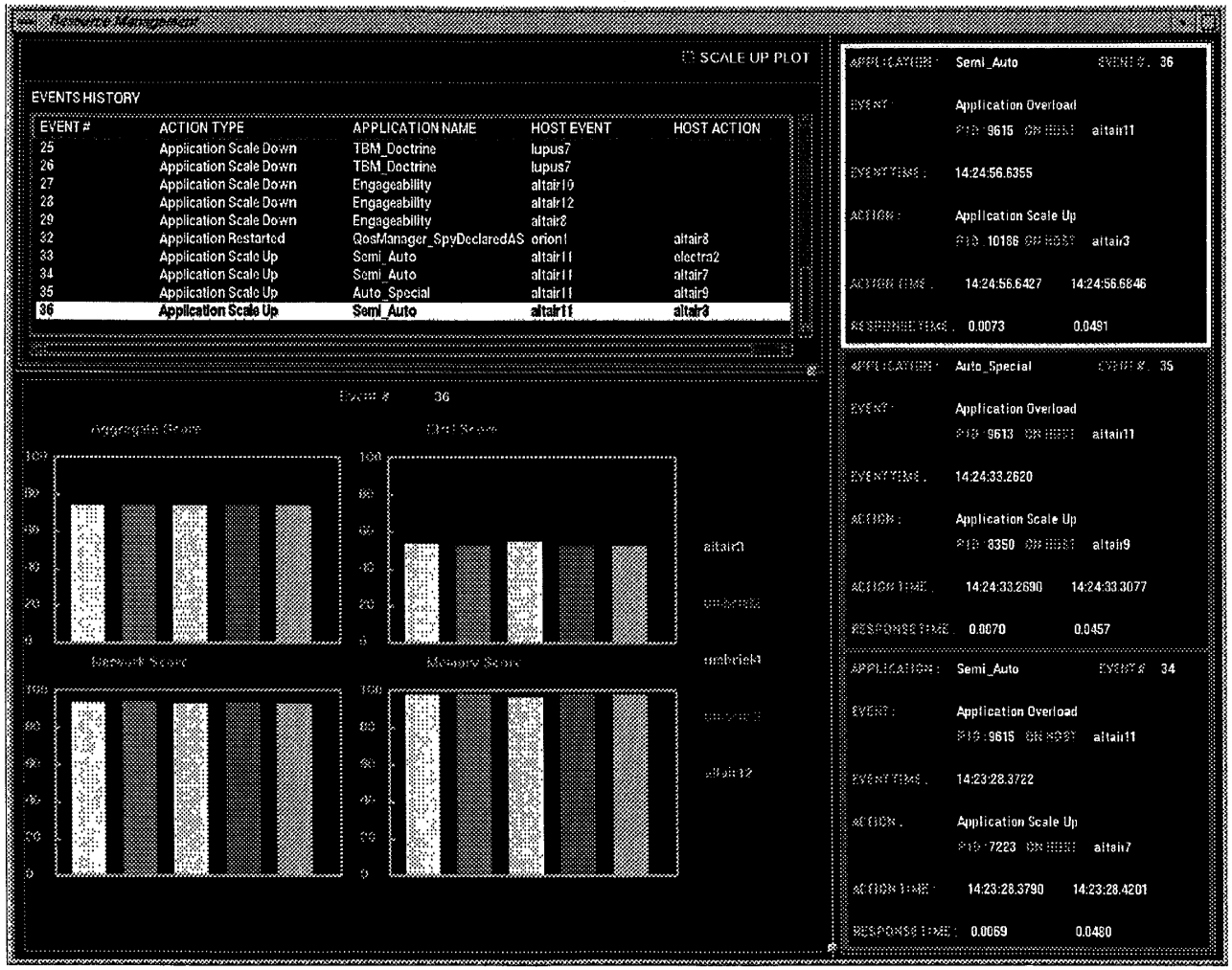


Fig. 9

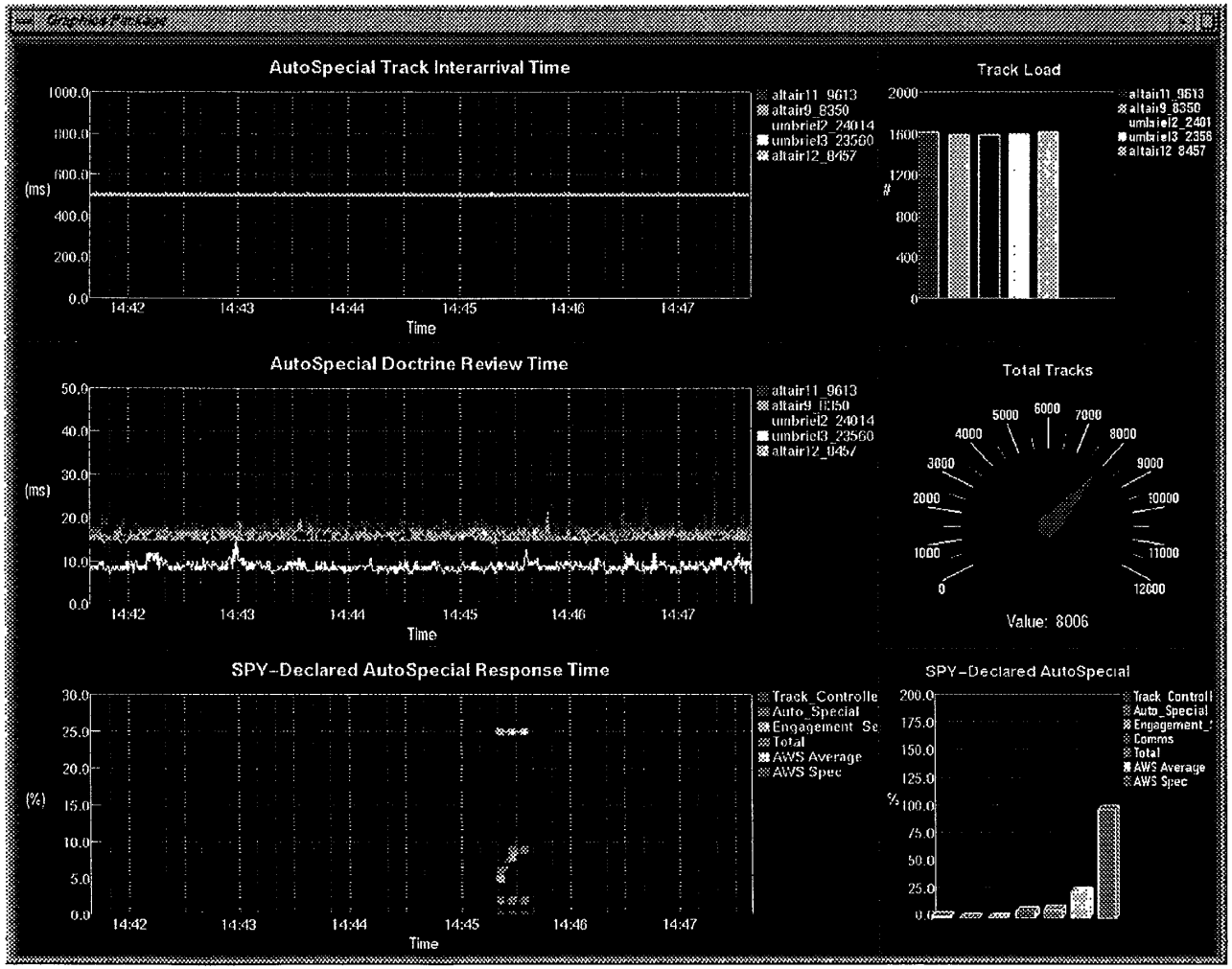

AutoSpecial Track Interarrival Time
1000.0
800.0
600.0
(ms)
400.0
200.0
0.0
14:42
14:43
14:44
14:45
14:46
14:47
Time
altair11_9613
altair9_8350
umbriel2_24014
umbriel3_23560
altair12_8457
Track Load
2000
1600
1200
#
800
400
0
altair11_9613
altair9_8350
umbriel2_2401
umbriel3_2356
altair12_8457
AutoSpecial Doctrine Review Time
50.0
40.0
30.0
(ms)
20.0
10.0
0.0
14:42
14:43
14:44
14:45
14:46
14:47
Time
altair11_9613
altair9_8350
umbriel2_24014
umbriel3_23560
altair12_8457
Total Tracks
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
12000
Value: 8006
SPY-Declared AutoSpecial Response Time
30.0
25.0
20.0
(%)
15.0
10.0
5.0
0.0
14:42
14:43
14:44
14:45
14:46
14:47
Time
Track_Controlle
Auto_Special
Engagement_Se
Total
AWS Average
AWS Spec
SPY-Declared AutoSpecial
200.0
175.0
150.0
125.0
%
100.0
75.0
50.0
25.0
0.0
Track_Controll
Auto_Special
Engagement_
Comms
Total
AWS Average
AWS Spec

Fig. 10

Patented September 1998

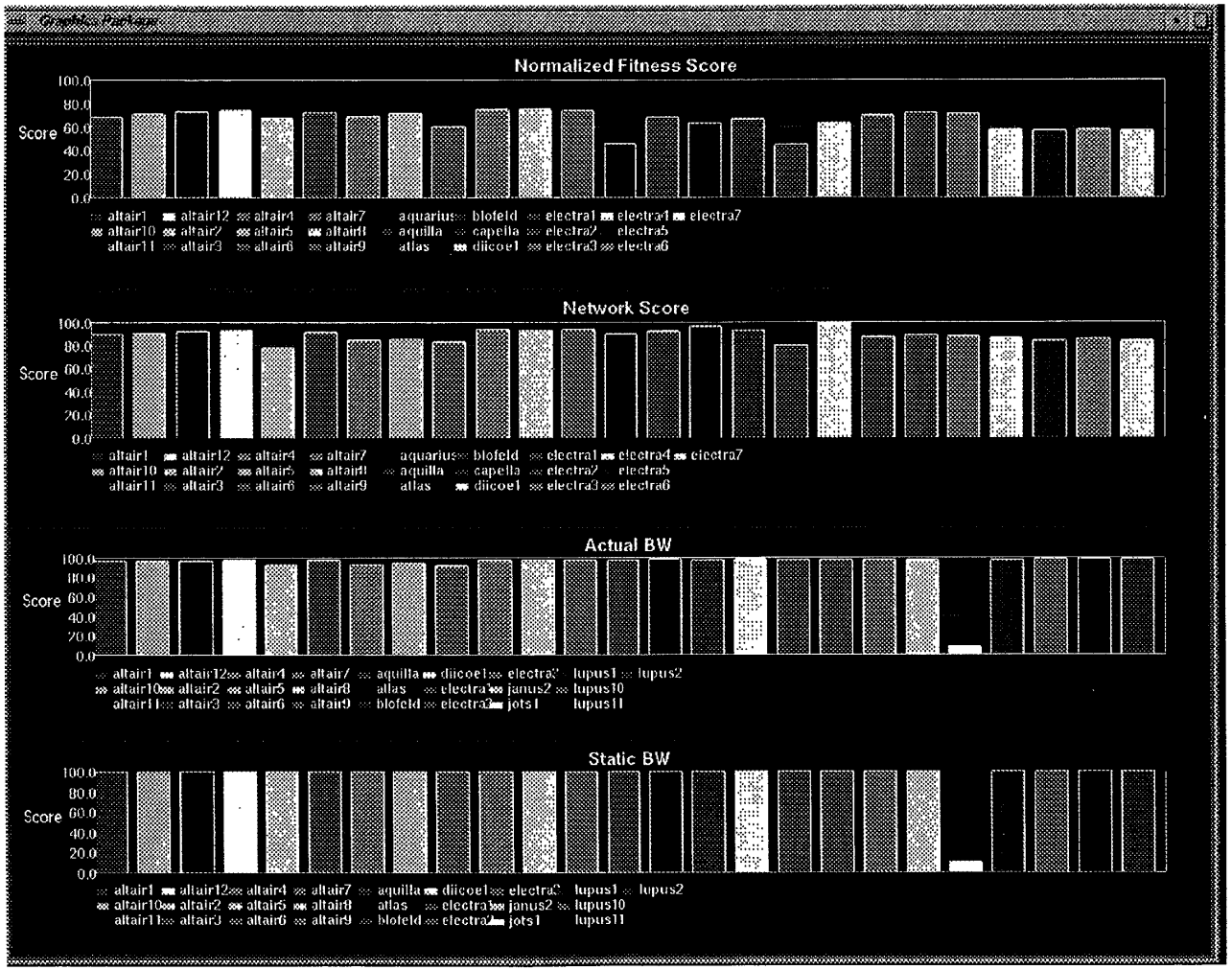


Fig. 11

Petitioner's Exhibit 1002

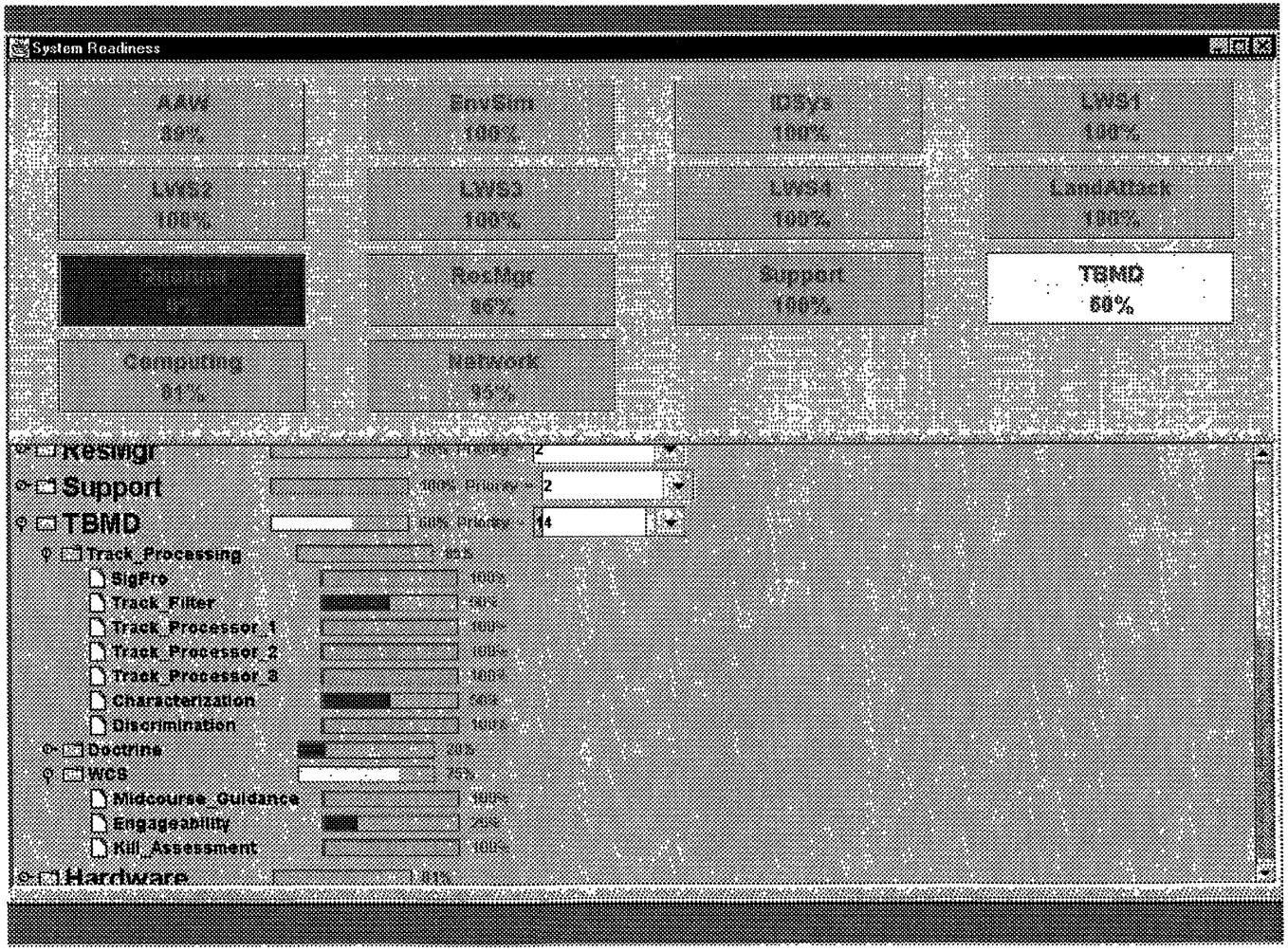


Fig. 12

Patent Application

Fig. 13A

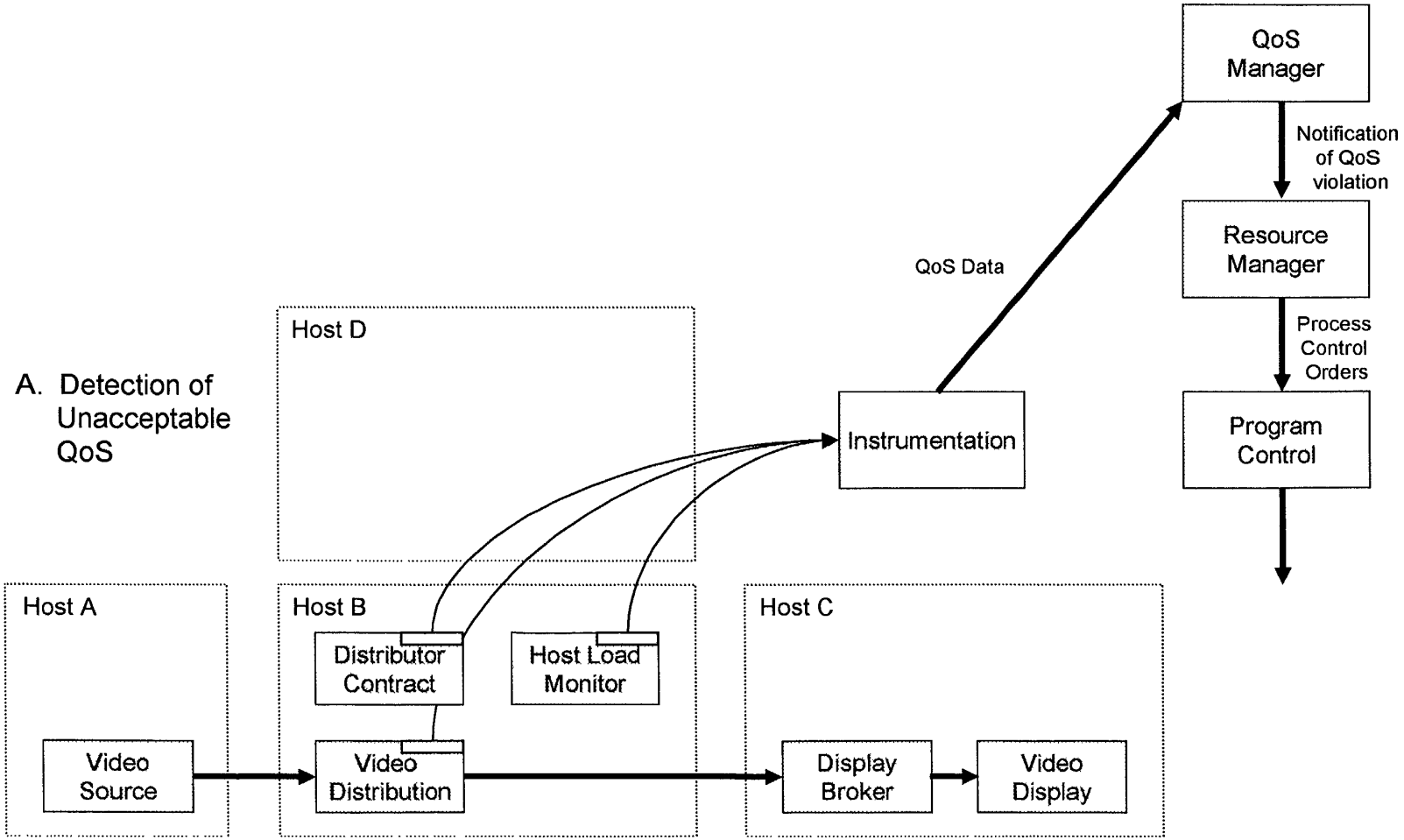
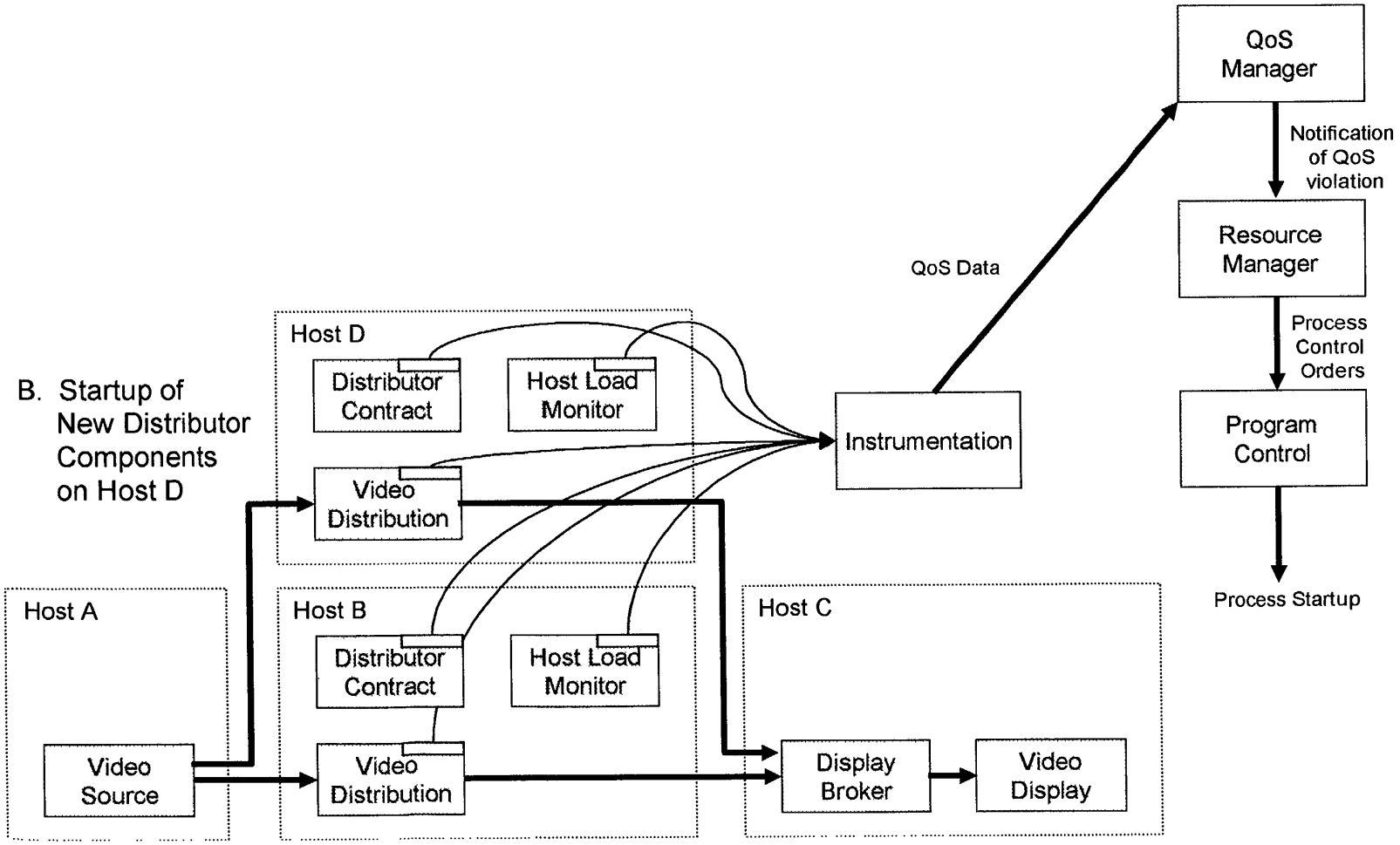
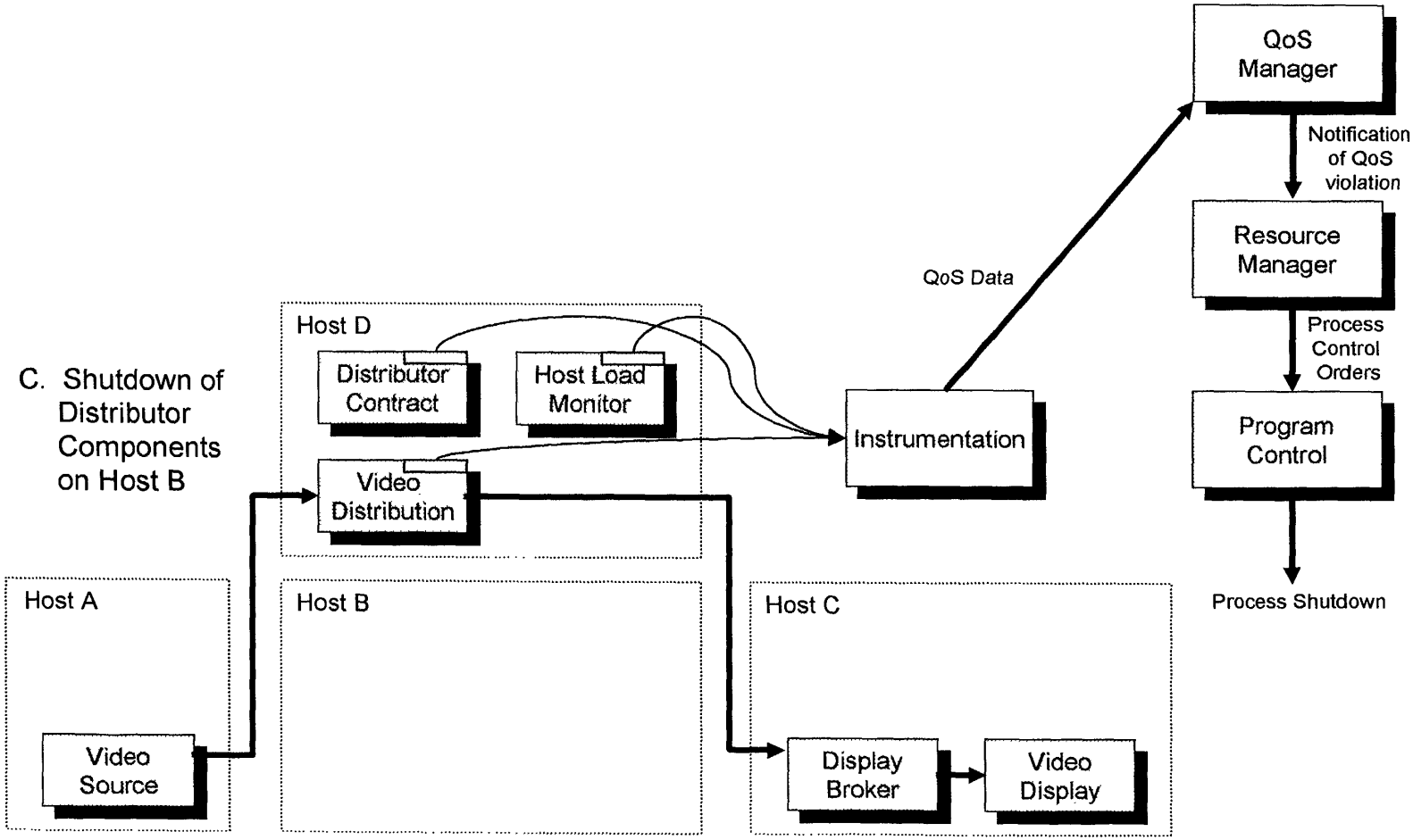


Fig. 13B


QoS Manager
Notification of QoS violation
Resource Manager
Process Control Orders
Program Control
Process Startup
QoS Data
Instrumentation
B. Startup of New Distributor Components on Host D
Host D
Distributor Contract
Host Load Monitor
Video Distribution
Host A
Video Source
Host B
Distributor Contract
Host Load Monitor
Video Distribution
Host C
Display Broker
Video Display

Petitioner's Exhibit

Fig. 13C



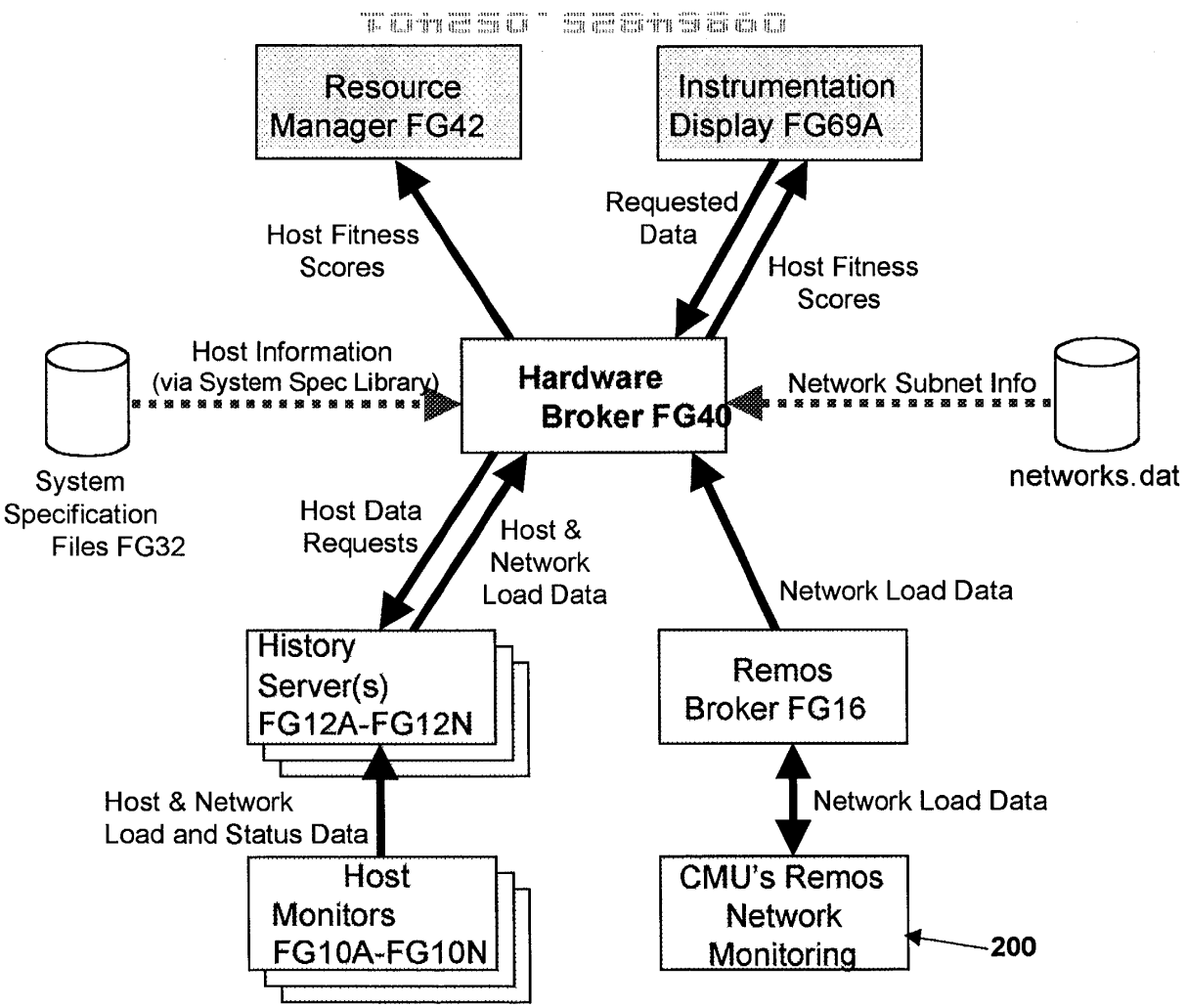

Resource Manager FG42
Instrumentation Display FG69A
Host Fitness Scores
Requested Data
Host Fitness Scores
Host Information (via System Spec Library)
Hardware Broker FG40
Network Subnet Info
System Specification Files FG32
networks.dat
Host Data Requests
Host & Network Load Data
Network Load Data
History Server(s) FG12A-FG12N
Remos Broker FG16
Host & Network Load and Status Data
Network Load Data
Host Monitors FG10A-FG10N
CMU's Remos Network Monitoring
200

Fig. 14

Patent Application

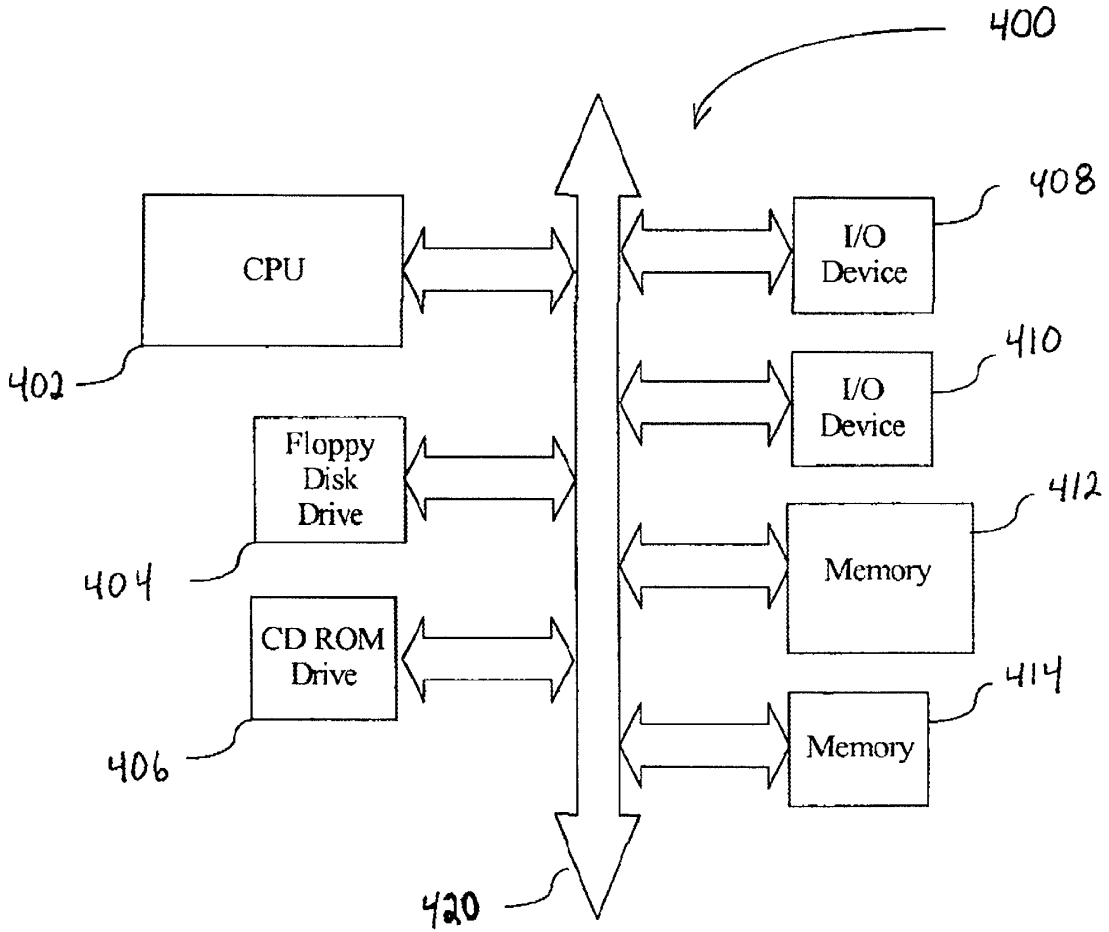


Fig. 15

Navy Case No. 83017

DECLARATION and POWER OF ATTORNEY for PATENT APPLICATION

As a below named inventor, I hereby declare that:
My residence, post office address and citizenship are as stated below next to my name,
I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled **RESOURCE ALLOCATION DECISION FUNCTION FOR RESOURCE MANAGEMENT ARCHITECTURE AND CORRESPONDING PROGRAMS THEREFOR**, the specification of which (check one) ☑ is attached hereto ☐ was filed on _____ as Application Serial No._____ and was amended on _____.

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56(a).

I hereby claim foreign priority benefits under Title 35, United States Code, § 119 (a)-(d) or § 365(b) of any foreign application (s) for patent or inventor's certificate, or § 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application for patent or inventor's certificate, or of any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date:	Priority Not Claimed	Certified Copy Attached? Yes	No

I hereby claim the benefit under Title 35, United States Code § 119(e) of any United States provisional application(s) listed below:

Application Number(s) 60/207,891 Filing Date (MM/DD/YYYY) 05/25/2000

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s), or §365© of any PCT international application designating the United States of America listed below and. insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56(a) which became available between the filing date of the prior application and the national or PCT international filing date of this application:

U.S. Parent Application Number	PCT Parent Number	Parent Filing Date (MM/DD/YYYY)	Parent Patent Number (If applicable)

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith, (list name and registration number), and hereby certify that the Government of the United States has the irrevocable right to prosecute this application:

SEND CORRESPONDENCE TO: James B. Bechtel, Esq.
NSWCDD (CD222)
Dahlgren, VA 22448-5100

DIRECT TELEPHONE CALLS TO: James B. Bechtel, Esq.
Reg. No. 29,890
(540)653-8061

I hereby declare that all statements made herein of my knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of sole or first inventor Paul V. Werme

Inventor's signature _____ Date _____
Residence 6172 Potomac Drive, King George, VA 22485
Citizenship United States Post Office Address Same as above

Full name of second joint inventor, if any _____

Second Inventor's signature _____ Date _____
Residence _____
Citizenship United States Post Office Address Same as above

Full name of third joint inventor, if any _____

Third Inventor's signature _____ Date _____
Residence _____
Citizenship United States Post Office Address Same as above

Full name of fourth joint inventor, if any _____

Fourth Inventor's signature _____ Date _____
Residence _____
Citizenship _____ Post Office Address _____

Full name of fifth joint inventor, if any _____

Fifth Inventor's signature _____ Date _____
Residence _____
Citizenship _____ Post Office Address _____

Attached Appendix A

- Host and Network Monitoring:
  - Collection of Data via Operating System Calls for Minimal Intrusiveness
  - Common Data Formats Across Platforms
  - Distribution of Host and Network Statuses and Performance Histories
  - Discovery of Distributed Environment Configuration Changes
    - Detection of Host Failure
    - Detection of Host Startup
- Application-Level Instrumentation:
  - Low-Overhead Application API's
  - Common User-Specified Instrumentation Data Formats
  - Data Collection and Distribution Architecture
  - Grammar-Driven Event Correlation
- System Specifications:
  - Modeling of Application Systems
    - Structure, Capabilities, and Configuration
    - Requirements and Inter-Dependencies
  - Modeling of Hardware and Network Systems
    - Structure, Capabilities, and Configuration
  - Run-Time Access to Specification Information
    - Run-Time Loading of Specification Information
    - Run-Time Access via Object-Oriented API
- Resource Allocation Decision-Making:
  - Determination of Application-to-Host Mappings
  - Recovery from Hardware and Software Failures
  - Detection of and Recovery from Software Performance Problems
    - Control of Application Scalability
    - Reallocation of Applications to Hosts
  - Reallocation of Applications to Hosts based on Priority Changes
    - Application-to-Host Mappings for New Required Applications
    - Selection of Applications to be Shutdown
  - Resolution of Inter-Application Startup Dependencies
- Resource Control:
  - Startup, Shutdown, and Configuration of Distributed Applications
    - Interactive Operator Control via Operator Display
      - Creation of Defined System Configurations
      - Loading of Pre-defined System Configurations
      - Startup, Shutdown, and Configuration of Individual Applications
    - Automatic Control via Resource Manager Orders
    - Failure Detection Capabilities
      - Application Failure Detection via Interrupt Notification
      - Host Failure Detection via Internal Heartbeat Mechanism
- Displays / Visualization:
  - Host Configuration and Performance
  - Network Configuration and Performance
  - Application Software Performance
  - *Resource Allocation Decisions and State Information*
  - Software Status and Configuration
  - User-Configurable Instrumentation Display
  - Near Real-Time Display of Information
- Middleware
  - Reliable Message Passing
  - Location-Transparent TCP Client-Server Configuration
  - Automated Connections and Reconnections
    - Client and Server detection via UDP multicast
  - Many-to-Many Client-Server Connections supported
  - Message Callback Function Registration
  - TCP Connection Status Change Callback Function Registration

Petitioner - Exhibit

Attached Appendix B:

1) Event data message header:

```
long total_bytes            // total number of bytes in the event data message
long message_type           // message type designator *
char version[8]             // version of Instrumentation APIs *
char test_name[24]          // test name for this event
double timetag                    // time stamp of when this event data message was sent
unsigned int gm_time        // GMT time stamp
unsigned int event_num      // event number
char process_name[24]       // name of application sending event data message
long pid                    // process id of application sending event data message
char host_name[64]          // host name that application is running on
long ip_addr                // ip address of host
long tid                    // task id of application sending event data message
unsigned int thread_type    // thread type
unsigned int sequence_num   // sequence number of the event data message
double time_in_client       // time the API was called to create event data message
double time_server_received // time the Instrumentation Daemon read in this event data message
double time_server_sent     // time the Instrumentation Daemon sent event data message to the Instrumentation Collector
```

2) The event data message format string contains data field names and format specifiers for each data field. The following data specifiers (borrowed from ANSI C) are supported:

```
%*r   : raw data, user defined, and number of bytes
%hi   : short signed 16 bit integer
%hd   : short signed 16 bit integer
%hu   : short unsigned 16 bit integer
%li   : long signed 32 bit integer
%ld   : long signed 32 bit integer
%lu   : long unsigned 32 bit integer
%lf   : IEEE double precision floating point - signed 64 bit floating point
```

%s : null-terminated string data
%c : character
%f : IEEE single precision floating point - signed 32 bit floating point
%i : signed 32 bit integer
%d : signed 32 bit integer
%u : unsigned 32 bit integer

Event string example: "StartTime %f StopTime %lf TrackNumber %u Hostname %s"

3) The data fields are then packed using the MessageBuffer class described in Appendix A of the RMComms Middleware Design Report.

Attached Appendix C

Class	Description
TCPCommClient	RMComms client server communication client services: ▪ client configuration - client name, server port number, network interface to use (optional) ▪ connection and disconnection to servers - all servers, specific servers, or servers on specific hosts ▪ sending user-defined messages to connected servers - send to all servers or only to specific servers ▪ receiving user-defined messages from connected servers - registration of message handler callback functions for specific messages - polled or asynchronous message delivery ▪ monitoring of server connection statuses - *queries to determine connected server statuses* - notification of new server connections or broken server connections
TCPCommServer	RMComms client-server communication server services: ▪ server configuration - server name, server port number, network interface to use (optional) ▪ connection to new clients ▪ sending user-defined messages to connected clients - send to all clients or only to specific clients ▪ receiving user-defined messages from connected clients - registration of message handler callback functions for specific messages - *polled or asynchronous message delivery* ▪ monitoring of client connection statuses - queries to determine connected client statuses - notification of new client connections or broken client connections
TimeUtils	Clock access and time conversion services: ▪ read system clock time ▪ time conversions between GMT and local time ▪ time conversions to hours, minutes, seconds, day, month, year
SignalRegistry	User-defined signal (interrupt) handler registration services: ▪ register a signal handler function for a specified signal - invoked when interrupt occurs ▪ unregister a signal handler function for a specified signal

United States Patent & Trademark Office

Office of Initial Patent Examination -- Scanning Division

PTO/SB/09 (08-00)

Application deficiencies found during scanning:

☐ Page(s)_____ of _____ were not present for scanning. (Document title)

☐ Page(s)_____ of _____ were not present for scanning. (Document title)

☒ ~~Scanned copy is best available.~~ Drawings.