13146 U.S.PTO
123004

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

I hereby certify that this transmittal of the below described documents is being deposited with the United States Postal Service in an envelope bearing Express Mail Postage and an Express Mail label, with the below serial number, addressed to the Commissioner for Patents P.O. Box 1450 Alexandria, VA 22313-1450, on the below date of deposit.

Express Mail Label No.:	EV561339198US	Name of Person Making the Deposit:	Anthony Chou
Date of Deposit:	12/30/04	Signature of the Person Making the Deposit:	Anthony Chou

113010 U.S.PTO
11/026622
123004

Inventor(s): Tung M. NGUYEN, Richard AU, Eric POUYOUL

Title: BRANCHING STORE FILE SYSTEM

Mail Stop Patent Application
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450
Sir:

Transmittal of a Patent Application
(Under 37 CFR §1.53)

Transmitted herewith is the above identified patent application, including:

- x Specification, claims and abstract, totaling 29 pages.
- x Formal drawings, totaling 7 pages.
- ____ Informal drawings, totaling ____ pages.
- x Declaration and Power of Attorney.
- ____ Information Disclosure statement.
- ____ Form 1449
- x Assignment(s)
- x Assignment Recordation Form (duplicate)
- x Request and Certification under 35 U.S.C. 122(b)(2)(B)(i)

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	$1000.00
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		Small Entity
	Fee ($)	Fee ($)
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 ($)	Fee Paid ($)	Multiple Dependent Claims
38	-20 or HP=	18	x	$50.00	$900.00	$900.00
HP= highest number of total claims paid for, if greater than 20						
Indep. Claims		Extra Claims		Fee ($)	Fee Paid ($)	
4	-3 or HP=	1	x	$200	$200.00	$200.00
HP= highest number of independent claims paid for, if greater than 3						

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 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets

Total Sheets		Extra Sheets		Number of each additional 50 or fraction thereof	Fee $		Fee Paid $
	-100		/50			=	$0.00

4. OTHER FEE(S)

Non-English specification, $130 fee (no small entity discount)
Other: Assignment Recordation Fee: $40.00

Total Fees Due ($)
$2,140.00

5. PAYMENT OF FEES

The full fee due in connection with this communication is provided as follows:

1. Not enclosed

   [ ] No filing fee is to be paid at this time.

2. Enclosed

   [X] Filing fee

   [X] Recording assignment

   [X] The Commissioner is hereby authorized to charge any additional fees associated with this communication or credit any overpayment to Deposit Account No.: 23-0085. A duplicate copy of this authorization is enclosed.

   [X] A check in the amount of $2,140.00

This application is filed pursuant to 37 C.F.R. § 1.53 in the name of the above-identified Inventor(s).

[X] Customer No: 41066

Please direct all correspondence concerning the above-identified application to the following address:

WAGNER, MURABITO & HAO LLP
Two North Market Street, Third Floor
San Jose, California 95113
(408) 938-9060

[X] This transmittal ends with this page.

Respectfully submitted,

Date: December 30, 2004

By: Ronald M. Pomerenke
Ronald M. Pomerenke
Reg. No. 43,009

UNITED STATES PATENT APPLICATION FOR

BRANCHING STORE FILE SYSTEM

Inventors:

Tung M. Nguyen

Richard Au

Eric Pouyoul

Prepared by:

WAGNER, MURABITO & HAO, LLP

TWO NORTH MARKET STREET

THIRD FLOOR

SAN JOSE, CALIFORNIA 95113

(408) 938-9060

BRANCHING STORE FILE SYSTEM

FIELD OF THE INVENTION

[0001] Embodiments of the present invention relate to the field of networked computing. Specifically, embodiments of the present invention relate to distributed application deployment in a multi-computer system.

BACKGROUND ART

[0002] In large-scale systems, a trend in software deployment is to centralize Operating System (OS) software image management on a globally accessible file system with stateless computing nodes. The compute nodes are activated with the distributed application by either diskless booting protocols or remote software installation to local storage.

[0003] One approach to software image management is to use a single image with global process dispatch to lightweight OS node computing environments. A second approach is to dedicate one image per compute node.

[0004] Neither of the two general approaches provides fault isolation between different instances of the distributed application while being scalable and efficient. For example, the single OS image approach does not provide fault isolation between different instances of the distributed application. Moreover, a single OS image is a bottleneck that becomes worse as the size of the system grows.

[0005] The multiple OS image approach does not use system resources efficiently and is also difficult and time-consuming to install and maintain. For example, multiple OS images result in duplicated storage; therefore, system resources are not used efficiently. Moreover, using multiple OS images adds overhead to software installation and maintenance. Therefore, while multiple OS images provide isolation between instances of the distributed application, system resources are not used efficiently and installation and maintenance is time-consuming.

SUMMARY OF THE INVENTION

[0006] A branching store file system is disclosed. In one embodiment in accordance with the present invention, the file system includes a root file system containing the operating environment or base image of a distributed application, any number of shared branch images, and at least one dedicated leaf image for each compute node. The base image and shared branch images are read-only accessible by compute nodes. Shared branch images contain changes made to the base image which are visible to all compute nodes. The dedicated leaf images include further changes made to the base or branch images, visible only by respective compute nodes. The file system also has a filter that controls accesses of the distributed application by the compute nodes. A compute node's instance of the distributed application is based on the base image, branch image(s), and the compute node's leaf image(s).

[0007] In another embodiment of the present invention, the branching store file system has branch modification logs for changes to the distributed application. The system has a root file system with a root metadata structure and a root data structure. The entries in the root metadata structure point to associated entries in the root data structure. The branching store file system also has branch modification logs for respective compute nodes. A given branch modification log has a branch data structure comprising modified data files reflecting changes made to

an instance of the root data structure by a compute node. The branch modification log also has a branch metadata structure comprising unmodified and modified entries. The modified entries reflect changes made by the compute node to its instance of the root data structure. Further, the unmodified entries point to associated entries in the root data structure and the modified entries point to associated ones of the modified data files.

[0008] In another embodiment of the present invention, the branching store file system comprises a base instance of a distributed application and change volumes for changes made by compute nodes to their respective instances of the distributed application. The system also has a filter program code that directs accesses of the compute nodes' respective instances of the distributed application.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The accompanying drawings, which are incorporated in and form a part of this specification, illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention:

[0010] Figure 1 is a diagram of a branching store file system with leaf images, in accordance with an embodiment of the present invention.

[0011] Figure 2 is a diagram of a branching store file system using branch modification logs, in accordance with an embodiment of the present invention.

[0012] Figure 3 illustrates an exemplary branching store architecture using branch modification logs, in accordance with an embodiment of the present invention.

[0013] Figure 4 is a diagram of branching store file system employing copy on write to change volumes, in accordance with an embodiment of the present invention.

[0014] Figure 5 is a tree structure that illustrates a branching file system, in accordance with an embodiment of the present invention.

[0015] Figure 6 is diagram illustrating modules and structures of a file system, in accordance with an embodiment of the present invention.

[0016] Figure 7 is flowchart illustrating a process of a branching file system, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0017] In the following detailed description of embodiments of the present invention, a branching store file system, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be recognized by one skilled in the art that the present invention may be practiced without these specific details or with equivalents thereof. In other instances, well-known methods, procedures, and components have not been described in detail as not to unnecessarily obscure aspects of the present invention.

NOTATION AND NOMENCLATURE

[0018] Some portions of the detailed descriptions that follow are presented in terms of procedures, steps, logic blocks, processing, and other symbolic representations of operations on data bits that can be performed on computer memory. These descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. A procedure, computer executed step, logic block, process, etc., is here, and generally, conceived to be a self-consistent sequence of steps or instructions leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated in a computer system. It

has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

[0019] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussions, it is appreciated that throughout the present invention, discussions utilizing terms such as "removing" or "creating" or "generating" or "freezing" or "restoring" or "computing" or "resolving" or "communicating" or "determining" or "maintaining" or " providing" or "storing" or "constructing" or "performing" or "initializing" or "selecting" or "forming" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0020] Embodiments of the present invention provide methods and systems for distributed application deployment in a multi-compute node environment. Embodiments of the present invention provide isolation and

fault containment typical of multiple OS image approaches with operation performance typical of a single OS image approach. Furthermore, embodiments of the present invention allow the flexibility to customize the application environment that is unmatched by either approach. Embodiments of the present invention are highly scalable. Embodiments of the present invention provide these advantages and others not specifically mentioned above but described in the sections to follow.

[0021] Figure 1 is a diagram of branching store system 100 in accordance with an embodiment of the present invention for enabling distributed application deployment in a multi-computer system 116. The branching store system 100 has a number of compute nodes 114 coupled to file system 102 through a filter 110. To the compute nodes 114 it appears that they have access to their own version of a distributed application 117. However, the file system 102 does not store a separate and complete version of the distributed application 117 for each compute node 114.

[0022] The branching store system 100 has a file system 102 containing a base image or instance 104 of a distributed application. The base instance 104 is not changed by compute nodes 114. For example, the compute nodes 114 have read-only access to the base image 104. However, if changes are made to the base image 104, those changes are essentially instantaneously known to the compute nodes 114.

[0023] Moreover, the file system 102 has a number of leaf images 105. For example, each compute node 114a-n has a corresponding leaf image 105a-n. A given leaf image 105 describes changes made by the respective compute node 114 to its instance of the distributed application. Thus, when a compute node (e.g., node 114a) make changes involving the base image 104, modifications are made to that compute node's leaf image (e.g., leaf image 105a). A leaf image 105 may comprise versions of entire files of the distributed application; however, this is not required. In one embodiment, the changes in the leaf image 105 are performed on a data block-by-data block basis. In another embodiment, the changes in the leaf image 105 are performed on a file-by-file basis. There may also be intermediate images (not depicted in Fig. 1) between the base image 104 and the leaf images 105.

[0024] A compute node 114 mounts its instantiation of the distributed application via the filter 110. The filter 110 determines what leaf image 105 is needed for portions of the distributed application that the compute node 114 has changed. The filter 110 also locates the portions of the distributed application that are not changed by the compute node 114. These portions may reside in the base image 104. There may also be intermediate images (not depicted in Fig. 1) comprising versions of the base image 104 from which a compute node's instance of the distributed application is derived.

Further, the filter 110 creates new leaf images 105 when a compute node 114 makes changes to its instantiation of the distributed application.

[0025] By maintaining a separate instance of the distributed application for each compute node 114, embodiments of the present invention provide fault isolation between instances of the distributed application. Furthermore, by recording only changes in the leaf images, embodiments of the present invention conserve system resources. Furthermore, updates to the base image are provided to the compute nodes nearly instantaneously.

[0026] Figure 2 is a diagram of branching store system 200 in accordance with an embodiment of the present invention in which a branch modification log is used to record changes to the base instance of the distributed application. The system 200 has a number of compute nodes 114a-n coupled to network 112 that have access to a file system 102 through a filter 110. To the compute nodes 114a-n, it appears that they have access to their own version of a distributed application 117a-n, respectively. However, the file system 102 does not store a separate and complete version of the distributed application 117 for each compute node 114.

[0027] The branching store system 100 has a file system 102 containing a base image or instance 104 of a distributed application (e.g., a base operating system). Briefly, the base image 104 contains data and

metadata for the base image 104 of the distributed application. The base image 104 is not changed by compute nodes 114. For example, the compute nodes 114 have read-only access to the base image 104. However, if changes are made to the base image 104, those changes are essentially instantaneously known to the compute nodes 114.

[0028] Moreover, the file system 102 has a number of branch modification logs 205a-n. For example, each compute node 114a-n has a corresponding branch modification log 205a-n. The branch modification log 205 is one embodiment of a leaf image (Fig. 1, 105). A given branch modification log (e.g., 205a) describes changes made by the respective compute node (e.g., 114a) to its instance of the distributed application (e.g.,117). Thus, when a compute node 114 make changes involving the base image 104, modifications are made to that compute node's branch modification log 205. A compute node's instance is thus defined by at least a portion of the base image 104 and its branch modification log 205. This is indicated by the groupings 108a-n. It will be understood that a compute node's instance of the distributed application does not typically include all of the grouping 108. For example, some of the base image 104 may be copied into a branch modification log 205 and modified therein.

[0029] A branch modification 205 contains metadata and data for a respective compute node's instance of the distributed application. However,

the data in a given branch modification log 205 only reflects differences between the data in the base image 104 and the given compute node's instance of the distributed application, in this embodiment. A compute node 114 mounts its instance of the distributed application as pointed to by its branch modification log 205. For example, a compute node 117a accesses the base image 104 for data that is unchanged and accesses its branch modification log 205a for data that the compute node 114a has changed. Thus, this embodiment traverses inodes to determine the location of data for a compute node's instance of a distributed application.

[0030] Figure 3 illustrates an exemplary branching store architecture 150, in accordance with an embodiment of the present invention. The exemplary branching store architecture 150 may be used in the system depicted in Figure 1 or Figure 2, although systems 100 and 200 are not limited to this architecture 150. The branching store architecture 150 has a root file system 104 (also referred to herein as a base image) and a branch modification log 205. The branch modification log 205 is one embodiment of a leaf image (Fig. 1, 105). Typically, there will be many branch modification logs 205, such that one is available for each compute node in the system.

[0031] The root file system 104 has a root metadata structure 205 and a root data structure 210. Entries in the root metadata structure 205 point to associated entries in the root data structure 210. The root data structure

210 comprises the base instance of the distributed application. Neither the root data structure 210 nor the root metadata structure 205 are modified by compute nodes. However, a system management node or the like may modify the root file system 104, such that changes to the distributed application are easily and rapidly deployed to compute nodes.

[0032] The branch modification log 205 has a branch data structure 310 comprising modified data reflecting changes made to an instance of the root data structure by a compute node. For example, when a compute node makes changes to its instance of the distributed application data is added or modified in the branch data structure 310 for that compute node.

[0033] The branch modification log 205 also comprises a branch metadata structure 305 comprising unmodified entries 320 and modified entries 315. The modified entries 315 reflect changes made to the instance of the distributed application by the compute node. Thus, a modified entry 135 may be an entry that has a modification attribute attached to the entry. A modified entry 315 points to an associated modified data block 325. For example, if an entry has an attribute indicating that it is a modified entry, then inodes are followed to the correct block 325 in the branch data structure 310.

[0034] The unmodified entries 325 point to associated entries in the root data structure 210. For example, if an entry does not have an attribute indicating that it is a modified entry, then the file lookup is re-directed such that inodes are followed to the root data structure 210.

[0035] The instance of the distributed application is thus defined by the branch metadata structure 305, wherein an unmodified portion of the distributed application resides in the root data structure 210 and a modified portion of the distributed application resides in the branch data structure 310.

[0036] Figure 4 is a diagram of branching store system 400 in accordance with an embodiment of the present invention in which a copy on write to change volumes 405 are used to record changes to the base instance of the distributed application. Changes to a compute node's instance of the distributed application, may also be kept as a database referred to as a change volume 405.

[0037] The file system 102 comprises a base image 104 of a distributed application 104 (e.g., a base OS) and a number of change volumes 405a-n. When a compute node 114 performs a write operation on a file, the entire original file is copied from the root image 104 into the change volume 405, if

the file does not already exist in the change volume 405. The write operation is then performed on the copy in the change volume 405.

[0038] When a compute node 114 mounts the distributed application, the compute node 114 mounts at least a portion of the base image 104 and the change volume 405. Access to files in a compute node's instance of the distributed application may proceed by first determining if the file exists in the compute node's 114 change volume 405, then determining if the file is in the base image 104.

[0039] There is a filter 110 at each compute node 114, in this embodiment. Alternatively, there could be a single filter 110 at the file system 102. The implementation of the filter 110 in this embodiment may be different than the implementation of the filter 110 of system 200 of Figure 2.

Application Groups

[0040] An embodiment of the present invention provides for application groups (AG). An AG is a way to provide a group of compute nodes their own instantiation of an application, while allowing versions of the application to be frozen from time to time. Thus, an archive of the application is created.

[0041] At creation time, an AG comprises the original root file system. A number of leaf nodes or images are created to match the desired capacity

demand of the AG. The leaf images can be created very quickly and require very little storage because the leaf image reflect changes to the root instance. In one embodiment, the leaf images comprise BMLs. In another embodiment, the leaf images comprise change volumes.

[0042] Figure 5 is a tree structure 500 that illustrates a branching file system, in accordance with an embodiment of the present invention. Some nodes are read-only, others are read-write-modify. A node may change from read-write-modify to read-only. For example, creating a new edition of the AG may be accomplished by freezing a current leaf image structure and creating a new set of leaf images. The frozen node is then made read-only. Referring now to Figure 5, the root node 104 (also referred to herein as a base image) is the base distributed application. As previously discussed, the root node 104 is not modified by compute nodes for which the application is distributed.

[0043] The first level of the exemplary tree structure 500 comprises first branch node 604 and branch node 606, which represent two different application service groups. Compute nodes have read only access to the service group applications represented by branches 604 and 606. Branch node 604 has three leaf nodes 105a, 105b, 105c. The leaf nodes 105 may be for configuration files, for example. The compute nodes can boot off from the configuration files. The leaf nodes are also referred to herein as leaf images.

In one embodiment, the leaf nodes 105 are change volumes. In another embodiment, the leaf nodes 105 are branch modification logs.

[0044] The branch node 606 also has three leaf nodes 605d, 605e, 605f, which are analogous to the leaf nodes 605a-c associated with branch node 604. However, branch node 606 also has a service group 620, which reflects a change to the application. The service group 620 has leaf nodes 605g, 605h, and 605i.

[0045] Figure 6 is diagram illustrating components of a file system 600, in accordance with an embodiment of the present invention. The file system 600 may be used in a copy-on-write embodiment, such as the system 400 depicted in Figure 4. However, it will be understood that the system 400 if Figure 4 is not limited to file system 600. The filter 602 presents the view to the compute nodes of a regular file system. In one embodiment, the filter presents the view of a UNIX file system.

[0046] The file system 600 also has a table of locations 606, which indicates the locations of branch nodes. The file system has an XML document that indicates the locations. The XML document can be sent to compute nodes.

[0047] Also stored are base addresses 608, application addresses 610, and change addresses 612. The file system has a networking layer 614 for interfacing on the network (Fig. 4, 112).

[0048] Figure 7 is flowchart illustrating a process 700 of branching file system access, in accordance with an embodiment of the present invention. Steps of process 700 may stored as instructions on a computer readable medium and executed on a computer processor. Step 710 is creating a root file system comprising a root image of an application.

[0049] Step 720 is creating leaf images that are configured to record changes to the root image of the application. The leaf images comprise branch modification logs in one embodiment. In another embodiment, the leaf images comprise change volumes.

[0050] Step 730 is creating a directory configured to access instances of the application for respective compute nodes. A respective compute node's application instance comprises at least a portion of the root image and the respective compute node's leaf image.

[0051] Step 740 is providing the compute nodes with respective instances of the application. Step 740 may comprise a filter determining

which files or data blocks are needed for a given compute node's instance of the application.

[0052] Step 750 is modifying one of the leaf images in response to a corresponding compute node's access to its instance of the application. The modifying of the leaf image may include copying a file from the root image to the first leaf image and then modifying the file in the first leaf image. However, this step is not so limited. The modifying may also be performed by copying one or more data blocks from the root image to a leaf image, and modifying the data block in the leaf image. It will be appreciated that the file or data block being modified may already exists in the leaf image, in which case it does not need to be copied from the base image before modification.

[0053] The preferred embodiment of the present invention, branching store file system, is thus described. While the present invention has been described in particular embodiments, it should be appreciated that the present invention should not be construed as limited by such embodiments, but rather construed according to the below claims.

CLAIMS

What is Claimed, is:

1. A branching store file system, comprising:

a root file system comprising a base image of a distributed application, wherein said base image is read-only accessible by compute nodes;

leaf images comprising modified portions of said distributed application for respective ones of said compute nodes, wherein an instance of said distributed application for a first of said compute nodes is based on said base image and at least one of said leaf images; and

a filter that controls access of instances of said distributed application for said compute nodes.

2. The system of Claim 1, wherein said leaf images comprise branch modification logs for said respective ones of said compute nodes.

3. The system of Claim 1, wherein said leaf images comprise change volumes for said respective ones of said compute nodes.

4. The system of Claim 1, wherein said filter is implemented at said compute nodes.

5. The system of Claim 1, wherein said filter is implemented at said root file system.

6. The system of Claim 1, further comprising intermediate images between said base image and said leaf images, wherein said intermediate images comprise application service groups.

7. The system of Claim 6, wherein access of said compute nodes to said intermediate images is read-only.

8. The system of Claim 1, wherein said distributed application comprises an operating system.

9. The system of Claim 1, wherein said base image is concurrently accessible to a plurality of said compute nodes.

10. The system of Claim 1, wherein said leaf images are configured to be modified on a block-by-block basis.

11. The system of Claim 1, wherein said leaf images are configured to be modified on a file-by-file basis.

12. A branching store file system, comprising:

a root file system having a root metadata structure and a root data structure, said entries in said root metadata structure pointing to associated unmodified data in said root data structure; and

a branch modification log for respective ones of compute nodes, comprising:

a branch data structure comprising modified data reflecting changes made to an instance of said root data structure by said branch modification log's respective compute node; and

a branch metadata structure comprising unmodified and modified entries, said unmodified entries pointing to associated portions of said unmodified data in said root data structure, and said modified entries pointing to associated portions of said modified data in said branch data structure, wherein an instance of a distributed application for a first of said compute nodes is defined by said branch metadata structure.

13. The system of Claim 12, further comprising a filter that directs accesses of said distributed application from compute nodes to a respective compute node's instantiation of the distributed application.

14. The system of Claim 12, wherein said distributed application comprises an operating system.

15. The system of Claim 13, wherein said filter is configured to traverse inodes to said root data structure for accesses to said un-modified data.

16. The system of Claim 13, wherein said filter is configured to traverse inodes to said branch data structure for accesses to said modified data.

17. The system of Claim 12, wherein access to said root data structure by said compute nodes is read-only and access to said branch data structure is read-write-modify by said branch modification log's respective compute node.

18. The system of Claim 12, wherein ones of said modified entries have a modification attribute attached.

19. The system of Claim 12, wherein said branch modification log is configured to receive updates to said modified data on a block-by-block basis.

20. A branching store file system, comprising:

a root file system comprising a base instance of a distributed application;

change volumes for respective ones of compute nodes and comprising modified versions of files of said distributed application; and

a filter program code that directs accesses of said compute nodes' respective instances of said distributed application, wherein said compute nodes' respective instances of said distributed application are based on said base instance and respective ones of said change volumes.

21. The system of Claim 20, further comprising a table of locations indicating locations of said change volumes.

22. The system of Claim 20, wherein said base instance is concurrently accessible to a plurality of said compute nodes.

23. The system of Claim 20, wherein said compute nodes have read-only access to files in said base instance.

24. The system of Claim 21, wherein said distributed application comprises an operating system.

25. The system of Claim 20, wherein said filter performs a copy-on-write to one of the change volumes when a compute node writes to said compute node's instantiation of said distributed application.

26. The system of Claim 20, wherein said change volumes are modified on a file-by-file basis.

27. A computer-implemented method of distributed application deployment, said method comprising:

creating a root file system comprising a root image of an application;

creating a plurality of leaf images that are configured to record changes to said root image of the application for respective compute nodes;

creating a directory configured to access instances of said application for said respective compute nodes, wherein a respective one of the compute node's instance of said application comprises at least a portion of said root image and the respective compute node's leaf image; and

providing said compute nodes with said respective instances of said application.

28. The method of Claim 27, wherein said leaf images comprise branch modification logs.

29. The method of Claim 28, wherein said providing said compute nodes with respective instances of said application comprises traversing inodes as indicated by said branch modification logs.

30. The method of Claim 27, wherein said leaf images comprise change volumes.

31. The method of Claim 30, wherein said providing said compute nodes with respective instances of said application comprises consulting a table of locations.

32. The method of Claim 27, wherein said application comprises an operating system.

33. The method of Claim 27, wherein access to said root image by said compute nodes is read-only.

34. The method of Claim 27, further comprising freezing a version of said distributed application.

35. The method of Claim 34, further comprising creating new leaf nodes that are children of said frozen version of said distributed application.

36. The method of Claim 27, further comprising modifying a first of said leaf images in response to a corresponding compute node's access to its instance of the application.

37. The method of Claim 36, wherein said modifying comprises a copying a file from said root image to said first leaf image and then modifying said file in said first leaf image.

38. The method of Claim 36, wherein said modifying comprises a copying a data block from said root image to said first leaf image and then modifying said data block data block in said first leaf image.

BRANCHING STORE FILE SYSTEM

ABSTRACT

A branch and store file system. The file system includes a root file system having a base image of a distributed application. The base image is read-only accessible by compute nodes. The file system also has leaf images of the distributed application for respective ones of the compute nodes. The leaf images include changes made to instances of the base image by respective compute nodes. The file system also has a filter that controls accesses of the distributed application by the compute nodes. The compute nodes' respective instances of the distributed application are based on the base image and the compute node's leaf image.

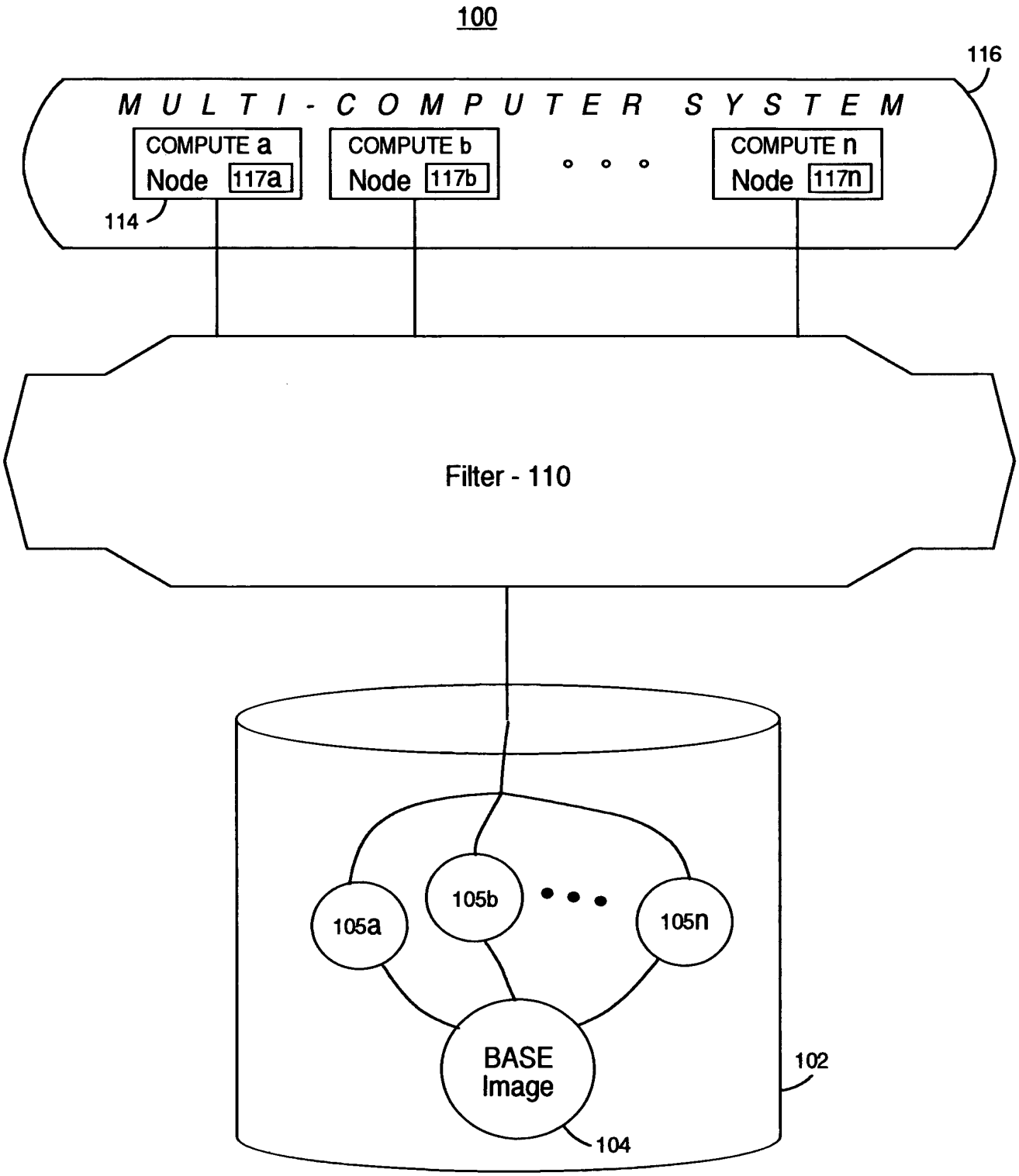

100
116
MULTI-COMPUTER SYSTEM
COMPUTE a Node 117a
COMPUTE b Node 117b
COMPUTE n Node 117n
114
Filter - 110
105a
105b
105n
BASE Image
104
102

Figure 1

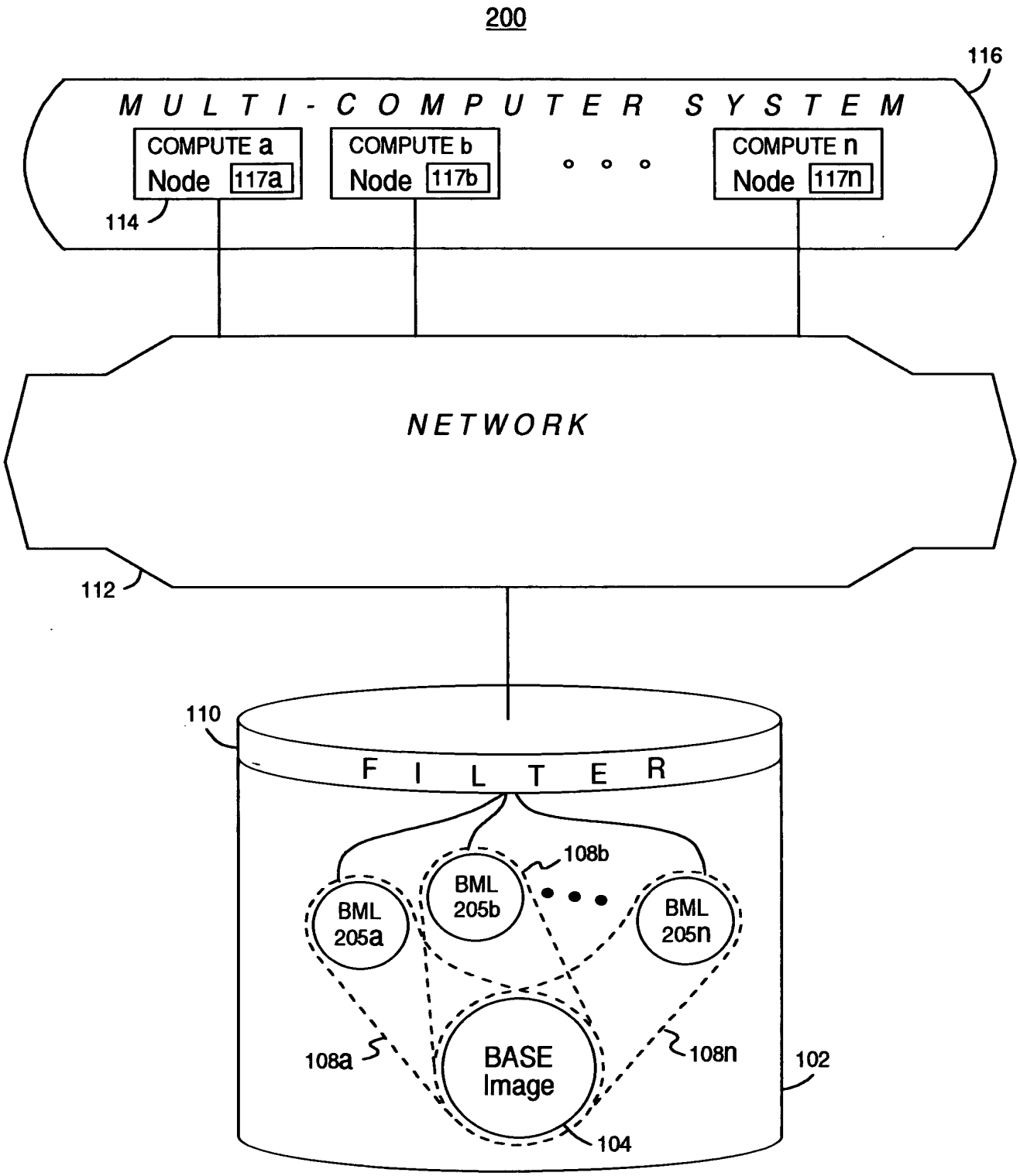

200
116
MULTI-COMPUTER SYSTEM
COMPUTE a Node 117a
COMPUTE b Node 117b
COMPUTE n Node 117n
114
NETWORK
112
110
FILTER
BML 205a
BML 205b
BML 205n
108b
108a
108n
BASE Image
104
102

Figure 2

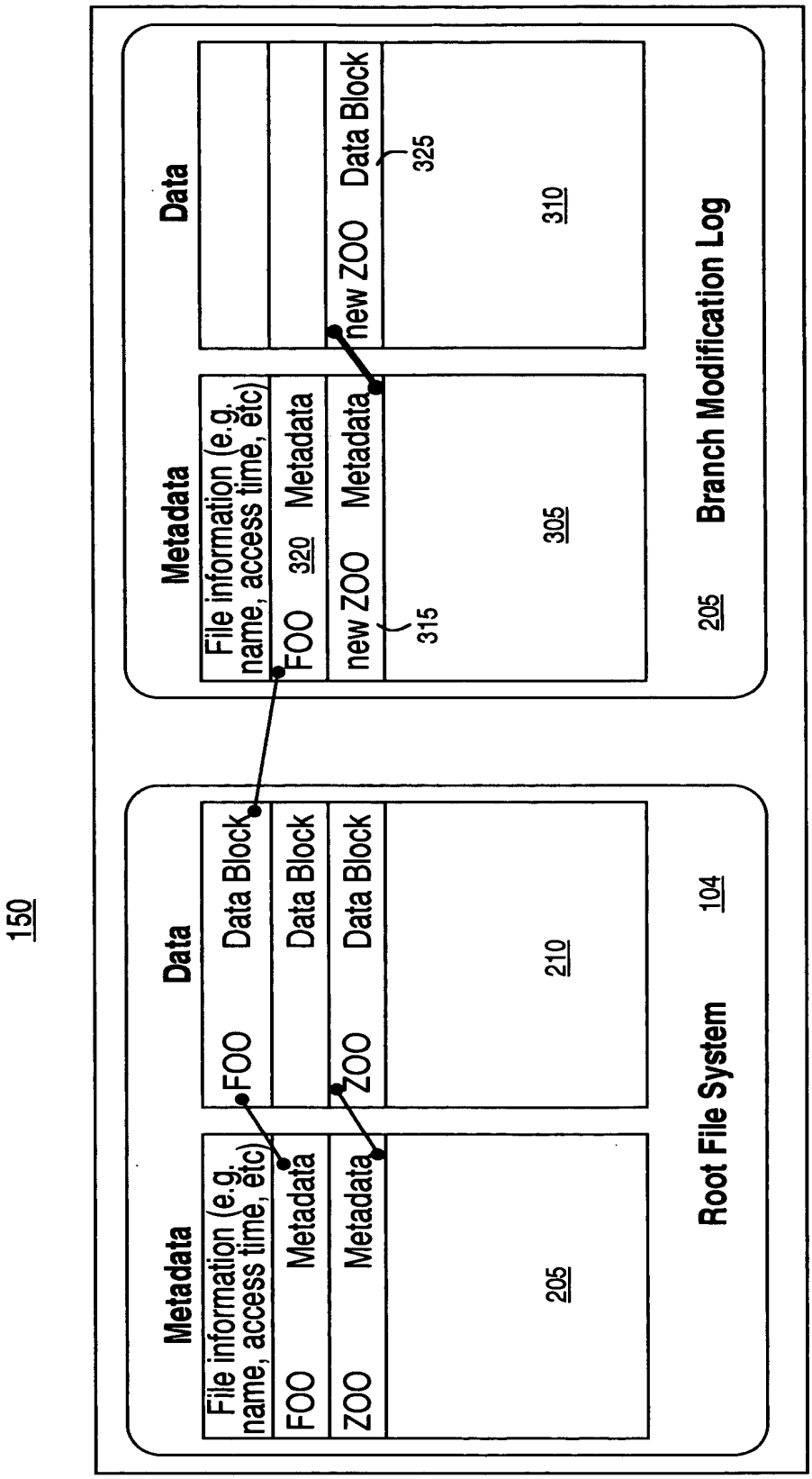

150
Metadata
File information (e.g. name, access time, etc)
FOO Metadata
ZOO Metadata
Data
FOO Data Block
Data Block
ZOO Data Block
205
210
Root File System
104
Metadata
File information (e.g. name, access time, etc)
FOO 320 Metadata
new ZOO Metadata
Data
new ZOO Data Block
325
315
305
310
205
Branch Modification Log

Figure 3

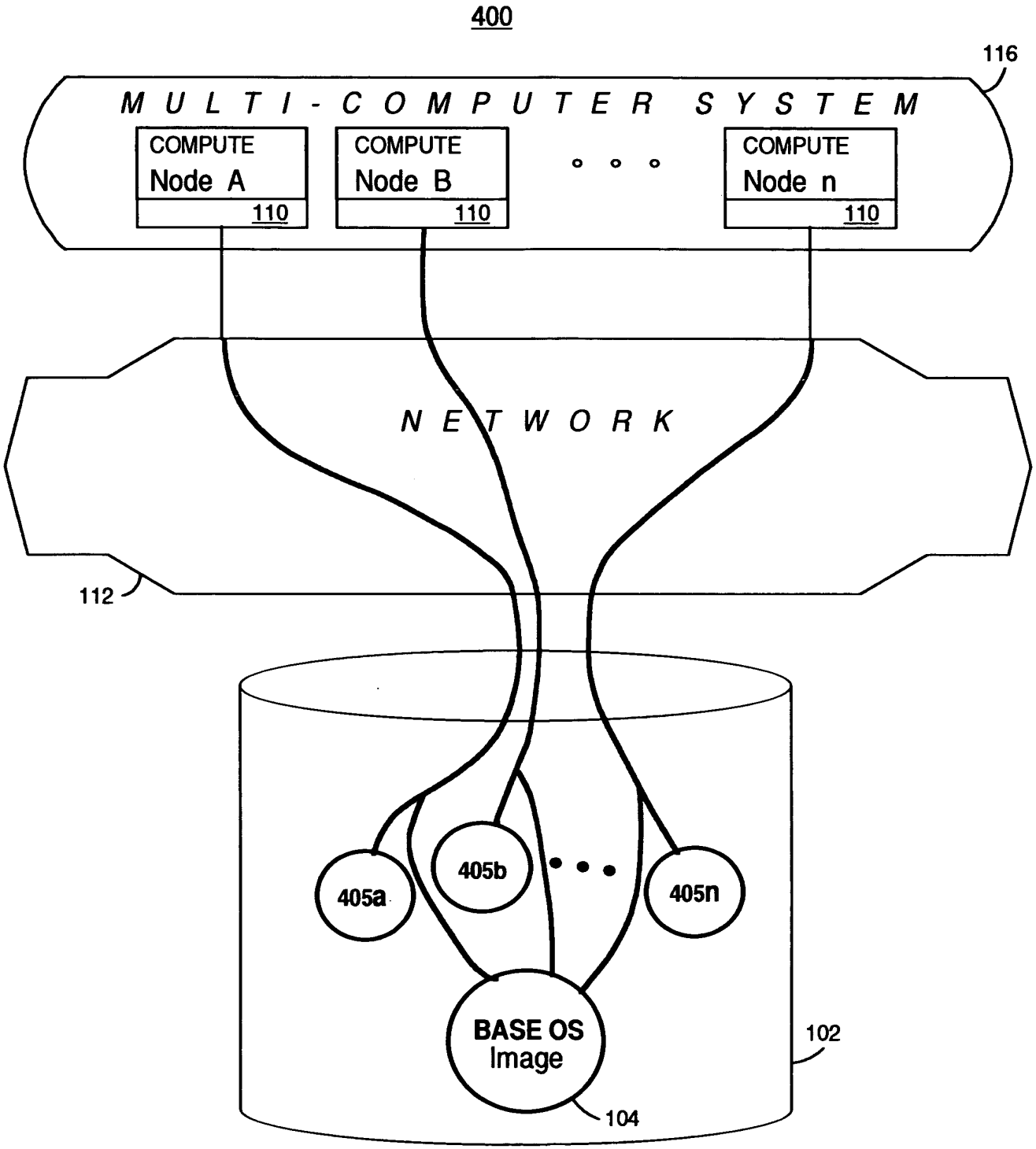

400
116
MULTI-COMPUTER SYSTEM
COMPUTE
Node A
110
COMPUTE
Node B
110
COMPUTE
Node n
110
NETWORK
112
405a
405b
405n
BASE OS
Image
104
102

Figure 4

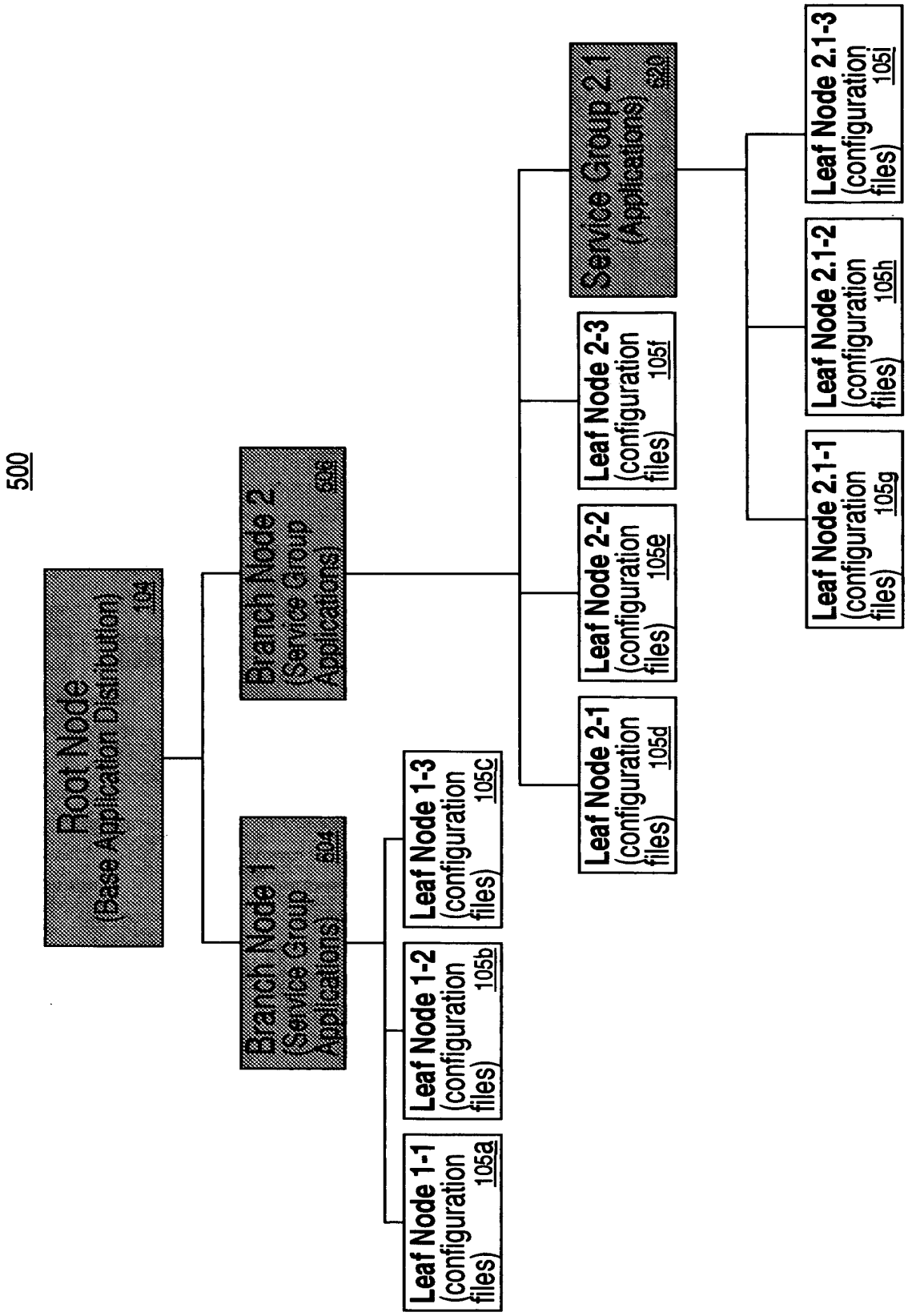

500
Root Node
(Base Application Distribution)
104
Branch Node 1
(Service Group
Applications)
604
Branch Node 2
(Service Group
Applications)
606
Leaf Node 1-1
(configuration
files)
105a
Leaf Node 1-2
(configuration
files)
105b
Leaf Node 1-3
(configuration
files)
105c
Leaf Node 2-1
(configuration
files)
105d
Leaf Node 2-2
(configuration
files)
105e
Leaf Node 2-3
(configuration
files)
105f
Service Group 2.1
(Applications)
620
Leaf Node 2.1-1
(configuration
files)
105g
Leaf Node 2.1-2
(configuration
files)
105h
Leaf Node 2.1-3
(configuration
files)
105i

Figure 5

600

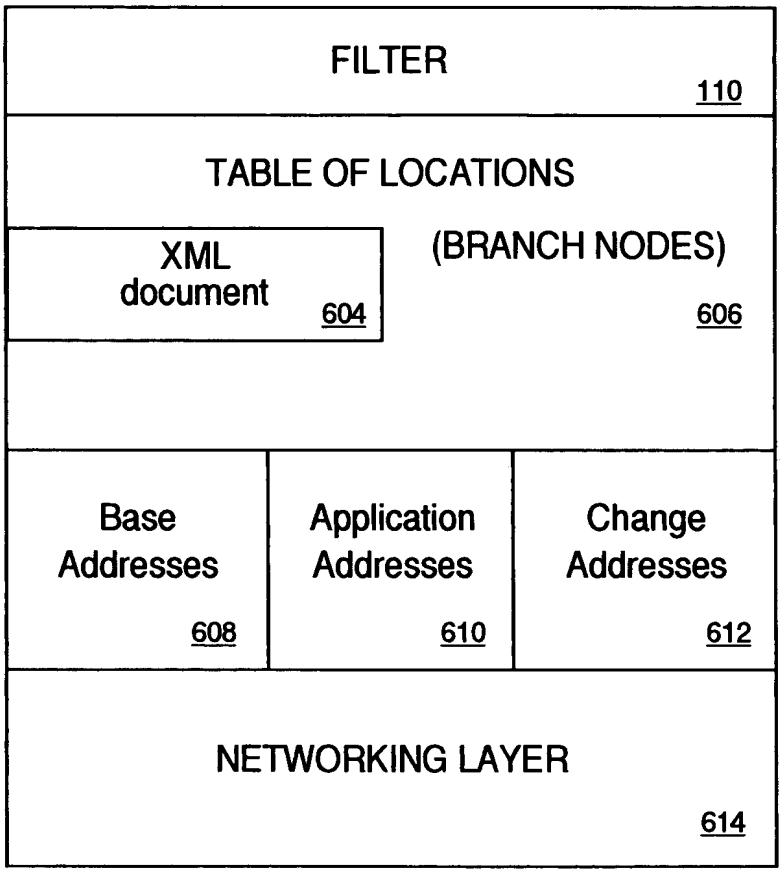

FILTER
110
TABLE OF LOCATIONS
XML document
604
(BRANCH NODES)
606
Base Addresses
608
Application Addresses
610
Change Addresses
612
NETWORKING LAYER
614

Figure 6

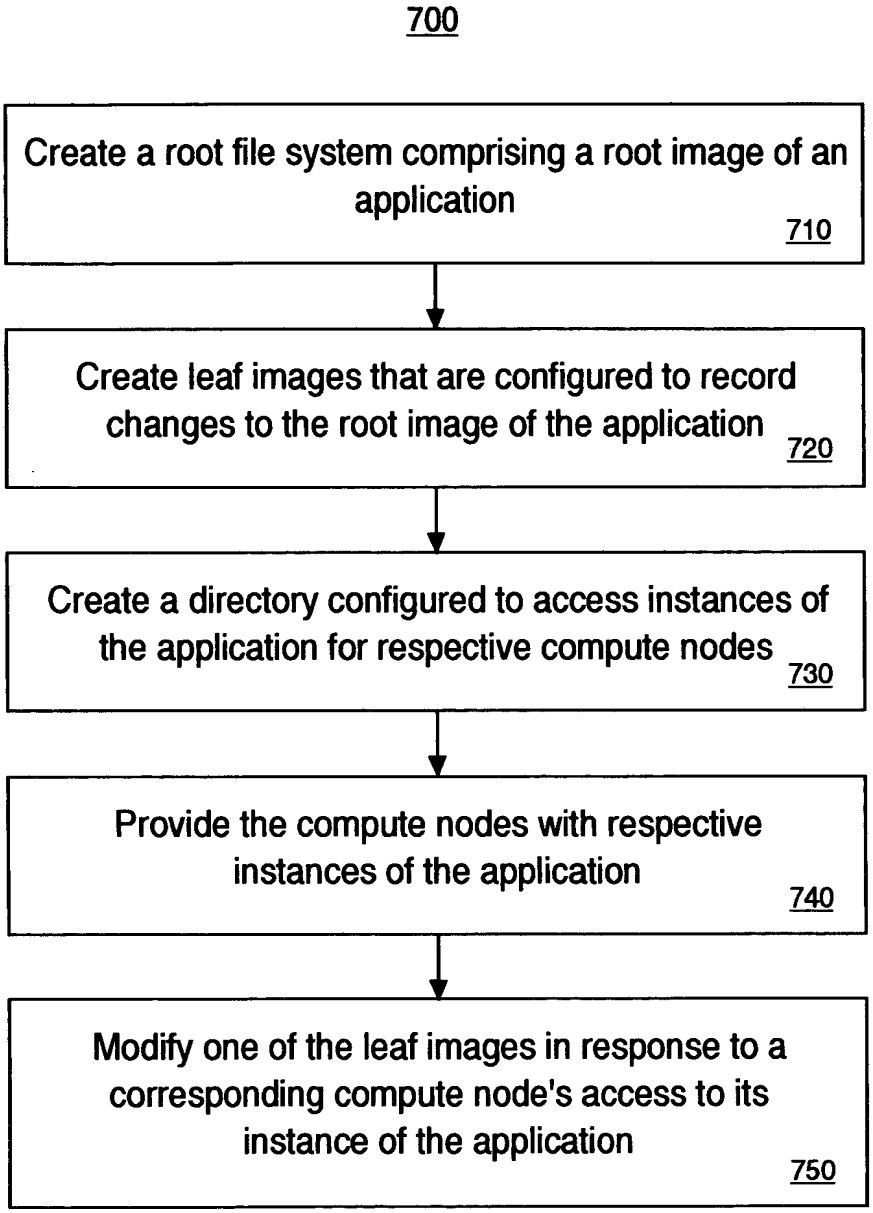

700
Create a root file system comprising a root image of an application
710
Create leaf images that are configured to record changes to the root image of the application
720
Create a directory configured to access instances of the application for respective compute nodes
730
Provide the compute nodes with respective instances of the application
740
Modify one of the leaf images in response to a corresponding compute node's access to its instance of the application
750

Figure 7

Attorney Docket No.:PNTA-P002

Declaration and Power of Attorney for a Patent Application

Declaration

As below named inventor, I hereby declare that my residence post office address, and citizenship are as stated below my name. Further, I hereby declare that 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:

BRANCHING STORE FILE SYSTEM

the specification of which:

__X__ is attached hereto, or
_____ 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; and

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56.

Foreign Priority Claim

I hereby claim foreign priority benefits under Title 35, United States Code Section 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Number	Country	Date Filed	Priority Claimed	
			_____ yes	_____ no
			_____ yes	_____ no

U.S. Priority Claim

I hereby claim the benefit under Title 35, United States Code, Section 120 and 119(e) of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Serial Number	Filing Date	Status (patented/pending/abandoned)

BEST AVAILABLE COPY

Attorney Docket No.:PNTA-P002

Power of Attorney

As a named inventor, I hereby appoint practitioners associated with this Customer Number:

41066

as attorney(s) or agent(s) to represent the undersigned before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned only to the undersigned according to the USPTO assignment records or assignment documents attached to this form in accordance with 37 CFR 3.73(b).

Send Correspondence to:

WAGNER, MURABITO & HAO LLP
Two North Market Street
Third Floor
San Jose, California 95113
(408) 938-9060

Signatures

I hereby declare that all statements made herein of my own 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/First Inventor: Tung M. NGUYEN

Inventor's Signature _____ Date 12/27/2004
Residence Cupertino, CA Citizenship USA
(City State)
P.O. Address 22346 Mc. Clellan Road, Cupertino, CA 95014

Full Name of Second Inventor: Richard AU

Inventor's Signature _____ Date 12/23/2004
Residence Woodside, CA Citizenship USA
(City State)
P.O. Address 389 Moore Road, Woodside, CA 94062

Full Name of Third Inventor: Eric POUYOUL

Inventor's Signature _____ Date 12-27-2004
Residence San Francisco, CA Citizenship France
(City State)
P.O. Address 350 Day Street, San Francisco, CA 94131

BEST AVAILABLE COPY

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

01/06/2005 RMEBRAHT 00000079 11026622

01 FC:1011	300.00 OP
02 FC:1111	500.00 OP
03 FC:1311	200.00 OP
04 FC:1201	200.00 OP
05 FC:1202	900.00 OP

PTO-1556
(5/87)

*U.S. Government Printing Office: 2002 — 489-267/69033

PATENT APPLICATION FEE DETERMINATION RECORD
Effective December 8, 2004

Application or Docket Number: 11026622

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	38	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	38 minus 20=	* 18
INDEPENDENT CLAIMS	4 minus 3 =	* 1
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	150.00	OR	BASIC FEE	300.00
X$ 25=		OR	X$50=	900
X100=		OR	X200=	200
+180=		OR	+360=	
TOTAL		OR	TOTAL	

CLAIMS AS AMENDED - PART II

AMENDMENT A

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

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDI-TIONAL FEE		RATE	ADDI-TIONAL FEE
X$ 25=		OR	X$50=	
X100=		OR	X200=	
+180=		OR	+360=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

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

RATE	ADDI-TIONAL FEE		RATE	ADDI-TIONAL FEE
X$ 25=		OR	X$50=	
X100=		OR	X200=	
+180=		OR	+360=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

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

RATE	ADDI-TIONAL FEE		RATE	ADDI-TIONAL FEE
X$ 25=		OR	X$50=	
X100=		OR	X200=	
+180=		OR	+360=	
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.

Attorney Docket No.: **PNTA-P002**

Inventor(s): Tung M. NGUYEN, Richard AU, Eric POUYOUL

Title: BRANCHING STORE FILE SYSTEM

REQUEST AND CERTIFICATION UNDER 35 U.S.C. 122(b)(2)(B)(i)

I hereby certify that the invention disclosed in the attached application **has not and will not be** the subject of an application filed in another country, or under a multilateral agreement, that requires publication at eighteen months after filing. I hereby request that the attached application not be published under 35 U.S.C. 122(b).

Date: December 30, 2004

By: Ronald M. Pomerenke
Reg. No. 43,009

This request must be signed in compliance with 37 CFR 1.33(b) and submitted with the application **upon filing**.

Applicant may rescind this nonpublication request at any time. If applicant rescinds a request that an application not be published under U.S.C. 122(b), the application will be scheduled for publication at eighteen months from the earliest claimed filing date for which a benefit is claimed.

If applicant subsequently files an application directed to the invention disclosed in the attached application in another country, or under a multilateral international agreement, that requires publication of applications eighteen months after filing, the applicant must notify the United States Patent and Trademark Office of such filing within forty-five (45) days after the date of the filing of such foreign or international application. **Failure to do so will result in abandonment of this application (35 U.S.C. 122(b)(2)(B)(iii)).**

OIPE IAP19
APR 03 2006
PATENT & TRADEMARK OFFICE

Attorney Docket No.: PNTA-P006

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

I hereby certify that this transmittal of the below described document is being deposited with the United States Postal Service in an envelope bearing First Class Postage and addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the below date of deposit.					
Date of Deposit:	03/30/06	Name of Person Making the Deposit:	Anthony Chou	Signature of the Person Making the Deposit:	Anthony Chou

In re Application

Inventor(s): NGUYEN, et al.

Application No.: 11/026,622

Filed: 12/30/2004

Title: NARROW WIDE SPACER

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

NOTIFICATION OF FILING OF CONTINUATION-IN-PART PATENT APPLICATION

[X] Notification is hereby being made of the filing of a continuation-in-part application for this case

[X] concurrently herewith.
[ ] on .....

Please direct all correspondence concerning the above-identified application to the following address:

WAGNER, MURABITO & HAO LLP
Two North Market Street, Third Floor
San Jose, California 95113
(408) 938-9060
Customer No: 000041066

Respectfully submitted,

Date: March 30, 2006

By: James P. Hao
Reg. No. 36,398

rev. 12/97 dbp

OIPE IAP19
APR 0 3 2006
PATENT & TRADEMARK OFFICE

Attorney Docket No.: PNTA-P006

IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

I hereby certify that this transmittal of the below described document is being deposited with the United States Postal Service in an envelope bearing First Class Postage and addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the below date of deposit.

Date of Deposit:	03/30/06	Name of Person Making the Deposit:	Anthony Chou	Signature of the Person Making the Deposit:	Anthony Chou

In re Application of:

Inventor(s): NGUYEN, et al.

Application No.: 11/026,622

Filed: 12/30/2004

Title: NARROW WIDE SPACER

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

CONDITIONAL PETITION FOR EXTENSION OF TIME

This conditional petition is being filed concurrently with the filing of a CONTINUATION-IN-PART PATENT APPLICATION

and provides for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Conditional petition for extension of time

If any extension of time for the accompanying response is required, applicant requests that this be considered a petition therefor.

Status

This application is on behalf of

- [X] other than a small entity
- [ ] a small entity
  - A verified statement:
    - [ ] is attached
    - [ ] is already filed

[X] The Commissioner is hereby authorized to charge any additional fees associated with this communication or credit any overpayment to Deposit Account No.: 23-0085.
A duplicate copy of this authorization is enclosed.

rev. 12/97 dbp

Please direct all correspondence concerning the above-identified application to the following address:

WAGNER, MURABITO & HAO LLP
Two North Market Street, Third Floor
San Jose, California 95113
(408) 938-9060
Customer No: 000041066

Respectfully submitted,

Date: March 30, 2006

By: _____
James P. Hao
Reg. No. 36,398

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
11/026,622		2141	2141

Correspondence Address / Fee Address Change

The following fields have been set to Customer Number 70848 on 06/04/2007

- Correspondence Address
- Maintenance Fee Address

The address of record for Customer Number 70848 is:
Panta c/o Murabito, Hao & Barnes LLP
Two Market Street
3rd Floor
San Jose,CA 95113

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.
11/026,622	12/30/2004	Tung M. Nguyen	PNTA-P002	8329

70848 7590 04/13/2009
PANTA c/o MURABITO HAO & BARNES LLP
Two Market Street
3rd Floor
San Jose, CA 95113

EXAMINER
SHAW, PETER C

ART UNIT	PAPER NUMBER
2458	

MAIL DATE	DELIVERY MODE
04/13/2009	PAPER

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

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	11/026,622	NGUYEN ET AL.	
	Examiner	Art Unit	
	PETER SHAW	2458	

-- 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 30 December 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) 1-38 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) 1-38 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 30 December 2004 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/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-38 are pending in this action.

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.

2. Claims 1-38 are rejected under 35 U.S.C. 102(b) as being anticipated by Chang (US PGPUB No. 2003/0126242).

As per claim 1, Chang teaches a branching store title system, comprising:

> a root file system comprising a base image of a distributed application ([0011], line 14, “base image”), wherein said base image is read-only accessible by compute nodes ([0026], line 2, “base image never written by the clients”) ([0022], line 15, base image is a building block for boot images, changes are made to copies to the base image);
>
> leaf images comprising modified portions of said distributed application for respective ones of said compute nodes ([0026], line 3, “client specific blocks”),

wherein an instance of said distributed application for a first of said compute nodes is based on said base image and at least one of said leaf images ([0033], line 17, OS and/or application based on common blocks from based image and client-specific blocks); and

a filter that controls access of instances of said distributed application for said compute nodes ([0017], lines 7-11, “external storage controller…controlling access to…boot images”).

As per claim 2, Chang teaches said leaf images comprising branch modification logs for said respective ones of said compute nodes ([0027], client image copies containing just the modifications to the base image, i.e. client-specific writes) ([0024], line 8, new block modifications logged and included in new image).

As per claim 3, Chang teaches said leaf images comprising change volumes for said respective ones of said compute nodes ([0025], lines 14-15, new blocks written to original volume to form new volumes).

As per claim 4, Chang teaches said filter implemented at said compute nodes ([0018], lines 1-3, “in one embodiment, the clients are configured...to provide similar or identical functions").

As per claim 5, Chang teaches said filter implemented at said root file system ([0017], line 1, "external storage controller residing in the network boot system").

As per claim 6, Chang teaches intermediate images between said base image and said leaf images ([0023], line 12, “client group images”), wherein said intermediate images comprise application service groups ([0023], line 12, “client group images”).

As per claim 7, Chang teaches access of said compute nodes to said intermediate images being read-only ([0026], line 2, “base image never written by the clients”).

As per claim 8, Chang teaches said distributed application comprising an operating system ([0016], line 10, “operating system”).

As per claim 9, Chang teaches said base image concurrently accessible to a plurality of said compute nodes ([0017], line 3, The boot system provides access to boot images for a “plurality of clients.”).

As per claim 10, Chang teaches said leaf images configured to be modified on a block-by-block basis ([0024], line 5, “data blocks”).

As per claim 11, Chang teaches said leaf images configured to be modified on a file-by-file basis ([0024], line 5, “files”).

As per claim 12, Chang teaches a branching store file system, comprising:

a root file system having a root metadata structure and a root data structure, said entries in said root metadata structure pointing to associated unmodified data in said root data structure ([0008], line 12, “Cloning…provide two collections of hard links to the same files”); and

a branch modification log for respective ones of compute nodes, comprising:

a branch data structure comprising modified data reflecting changes made to an instance of said root data structure by said branch modification log's respective compute node ([0027], line 10, The lazy clone technique does not contain an image, only provides instruction as to how to modify the base image to properly produce the client-specific image, i.e. metadata.); and

a branch metadata structure comprising unmodified and modified entries, said unmodified entries pointing to associated portions of said unmodified data in said root data structure ([0008], line 12, “Cloning…provide two collections of hard links to the same files”), and said modified entries pointing to associated portions of said modified data in said branch data

structure ([0027], line 10, The lazy clone technique does not contain an image, only provides instruction as to how to modify the base image to properly produce the client-specific image, i.e. metadata.), wherein an instance of a distributed application for a first of said compute nodes is defined by said branch metadata structure ([0027], line 8, It is inherent that the application is defined by the metadata contained in the client-specific writes.).

As per claim 13, Chang teaches a filter that directs accesses of said distributed application from compute nodes ([0017], lines 7-11, “external storage controller…controlling access to…boot images”) to a respective compute node's instantiation of the distributed application ([0030], lines 4-5, “access to client-specific image copy”).

As per claim 14, the substance of the claim language is identical to that of claim 8. Accordingly, this claim is rejected under the same rationale.

As per claim 15, Chang teaches said filter configured to traverse nodes to said root data structure for accesses to said unmodified data ([0033], lines 17-18, unmodified common data stored at separate location from updated client-specific blocks, Traversal to common data is inherent given the successful production of a copy image.).

As per claim 16, Chang teaches said filter configured to traverse nodes to said branch data structure for accesses to said modified data ([0033], lines 17-18, unmodified common data stored at separate location from updated client-specific blocks, Traversal to common data is inherent given the successful production of a copy image.) ([0023], line 5 and 12, Traversal to a varying base image or a client group image would be similarly inherent.).

As per claim 17, Chang teaches access to said root data structure by said compute nodes being read-only ([0026], line 2, “base image never written by the clients”) ([0022], line 15, base image is a building block for boot images, changes are made to copies to the base image) and access to said branch data structure being read-write-modify by said branch modification log's respective compute node ([0030], line 13, Given that the client boots from the image, it is inherent that it reads it.) ([0033], line 1, writes or changes received from clients).

As per claim 18, Chang teaches ones of said modified entries having a modification attribute attached ([0027], line 7, Lazy clone technique logs changes to the base image instead of writing an entire boot image. It is inherent that each of these entries would include both a pointer and what is being changed, i.e. modification attribute.).

As per claim 19, Chang teaches said branch modification log configured to receive updates to said modified data on a block-by-block basis ([0024], line 5, “data blocks”).

As per claim 20, Chang teaches a branching store file system, comprising:

a root file system comprising a base instance of a distributed application ([0011], line 14, “base image”);

change volumes for respective ones of compute nodes and comprising modified versions of files of said distributed application ([0025], lines 14-15, new blocks written to original volume to form new volumes); and

a filter program code that directs accesses of said compute nodes' respective instances of said distributed application ([0017], lines 7-11, “external storage controller…controlling access to…boot images”), wherein said compute nodes' respective instances of said distributed application are based on said base instance ([0011], line 14, “base image”) and respective ones of said change volumes ([0022], line 16, “client-specific image").

As per claim 21, Chang teaches a table of locations indicating locations of said change volumes ([0035], line 9, discovery of the pooled storage device, id of the specific-client image, i.e. change volumes).

As per claim 22, Chang teaches said base instance concurrently accessible to a plurality of said compute nodes ([0017], line 3, The boot system provides access to boot images for a “plurality of clients.”).

As per claim 23, Chang teaches said compute nodes having read-only access to files in said base instance ([0026], line 2, “base image never written by the clients”) ([0022], line 15, base image is a building block for boot images, changes are made to copies to the base image).

As per claim 24, the substance of the claim language is identical to that of claim 8. Accordingly, this claim is rejected under the same rationale.

As per claim 25, Chang teaches said filter performs a copy-on-write to one of the change volumes when a compute node writes to said compute node's instantiation of said distributed application ([0024], lines 1-3, “snapshot manager further functions to control copy-on-writes to the client-specific image copies from the clients”).

As per claim 26, Chang teaches said change volumes are modified on a file-by-file basis ([0024], line 5, “files”).

As per claim 27, Chang teaches a computer-implemented method of distributed application deployment, said method comprising:

creating a root file system comprising a root image of an application ([0011], line 14, “base image”);

creating a plurality of leaf images that are configured to record changes to said root image of the application for respective compute nodes ([0022], line 16, “client-specific image");

creating a directory configured to access instances of said application for said respective compute nodes ([0029], lines 20-21, pool storage device with paths to each client image), wherein a respective one of the compute node's instance of said application comprises at least a portion of said root image and the respective compute node's leaf image ([0024], line 8, New image is a combination of base image and new blocks.); and

providing said compute nodes with said respective instances of said application ([0017], lines 7-11, the external storage controller provides access).

As per claim 28, the substance of the claim language is identical to that of claim 2. Accordingly, this claim is rejected under the same rationale.

As per claim 29, Chang teaches traversing nodes as indicated by said branch modification logs ([0033], lines 17-18, unmodified common data stored at separate location from updated client-specific blocks, Traversal to common data is inherent given the successful production of a copy image.).

As per claim 30, the substance of the claim language is identical to that of claim 3. Accordingly, this claim is rejected under the same rationale.

As per claim 31, Chang teaches consulting a table of locations (The location of the storage controller and the copy image are implicitly stored on some form of a table.).

As per claim 32, the substance of the claim language is identical to that of claim 8. Accordingly, this claim is rejected under the same rationale.

As per claim 33, Chang teaches access to said root image by said compute nodes being read-only ([0026], line 2, “base image never written by the clients”) ([0022], line 15, base image is a building block for boot images, changes are made to copies to the base image).

As per claim 34, Chang teaches freezing a version of said distributed application ([0022], line 3 and 10, “snapshot of a boot image…deployed as a base image”).

As per claim 35, Chang teaches creating new leaf nodes that are children of said frozen version of said distributed application ([0022], line 6, “for generating new images”).

As per claim 36, Chang teaches modifying a first of said leaf images in response to a corresponding compute node's access to its instance of the application ([0027], line 7, In the lazy clone technique where only modifications are stored on “client images,” the image is modified by combining it with the common blocks from the base image.).

As per claim 37, Chang teaches copying a file from said root image to said first leaf image and then modifying said title in said first leaf image ([0023], line 12, When a “client image” is formed, it is implicit that its name will differ from the base image.).

As per claim 38, Chang teaches copying a data block from said root image to said first leaf image ([0024], lines 7, “copied block”) and then modifying said data block data block in said first leaf image ([0033], line 1, “writes or changes to image”).

Conclusion

3. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Pierre-Louis et al. (US Patent No. 6,421,777) describes multiple boot images for a distributed application taking into account the state of the client.

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to PETER SHAW whose telephone number is (571)270-7179. The examiner can normally be reached on Monday - Friday 7:30 A.M. to 5:00 P.M.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, GLENTON BURGESS can be reached on (571) 272-3949. 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.

/P. S./ April 1, 2009
Examiner, Art Unit 2458

/Joseph E. Avellino/

Primary Examiner, Art Unit 2446

Notice of References Cited	Application/Control No. 11/026,622	Applicant(s)/Patent Under Reexamination NGUYEN ET AL.	
	Examiner PETER SHAW	Art Unit 2458	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-2003/0126242	07-2003	Chang, Albert H.	709/222
*	B	US-6,421,777	07-2002	Pierre-Louis et al.	713/2
	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/Control No.	Applicant(s)/Patent Under Reexamination
	11026622	NGUYEN ET AL.
	Examiner	**Art Unit**
	PETER SHAW	2458

SEARCHED

Class	Subclass	Date	Examiner
709	220, 222, 203, 208	3/31/2009	PS
713	2	3/31/2009	PS
717	174,175	3/31/2009	PS

SEARCH NOTES

Search Notes	Date	Examiner
EAST Database Search (hist. incl.)	3/31/2009	PS
Inventor Search (no relevant art found)	3/31/2009	PS
Interpreted Fig. 3 of drawings to comprise hardware	4/8/2009	PS

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11026622	NGUYEN ET AL.
	Examiner	**Art Unit**
	PETER SHAW	2458

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

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

CLAIM		DATE								
Final	Original	03/31/2009								
	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	✓								

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11026622	NGUYEN ET AL.
	Examiner	**Art Unit**
	PETER SHAW	2458

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

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

CLAIM		DATE								
Final	Original	03/31/2009								
	37	✓								
	38	✓								

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

BIB DATA SHEET

CONFIRMATION NO. 8329

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
11/026,622	12/30/2004 **RULE**	~~707~~ 709	2458	PNTA-P002

APPLICANTS

Tung M. Nguyen, Cupertino, CA;
Richard Au, Woodside, CA;
Eric Pouyoul, San Francisco, CA;

**** CONTINUING DATA ***************************

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****

02/18/2005

Foreign Priority claimed ❑ Yes ❑ No 35 USC 119(a-d) conditions met ❑ Yes ❑ No Verified and Acknowledged _____ Examiner's Signature	❑ Met after Allowance _____ Initials	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
		CA	7	38	4

ADDRESS

PANTA c/o MURABITO HAO & BARNES LLP
Two Market Street
3rd Floor
San Jose, CA 95113
UNITED STATES

TITLE

Branching store file system

FILING FEE RECEIVED		
2100	FEES: Authority has been given in Paper No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	❑ All Fees ❑ 1.16 Fees (Filing) ❑ 1.17 Fees (Processing Ext. of time) ❑ 1.18 Fees (Issue) ❑ Other _____ ❑ Credit

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L8	0	(runtime OR dynamic).ti. AND "boot image".ti.	US-PGPUB; USPAT	OR	OFF	2009/04/01 13:25
L9	1	(runtime OR dynamic).ti. AND "boot image".ti.	US-PGPUB; USPAT	OR	ON	2009/04/01 13:25
L10	30	(runtime OR dynamic).ti. AND boot.ti.	US-PGPUB; USPAT	OR	ON	2009/04/01 13:26
S1	1	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	OFF	2009/03/31 14:02
S2	190	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:02
S3	548	"file system".ti. AND image	US-PGPUB; USPAT	OR	ON	2009/03/31 14:03
S4	4	"file system".ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:03
S5	120453	"operating system" AND image	US-PGPUB; USPAT	OR	ON	2009/03/31 14:07
S6	195	"distributed application".ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:19
S7	133	"distributed application".ti. AND "operating system"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:19
S8	15	"thin client" AND "operating system" AND (different OR version).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:25
S9	158308	"operating system" AND "thin client" (image OR configuration).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:27

S10	172	"operating system" AND "thin client" AND (image OR configuration).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:27
S11	1	"thin client".ti. AND (os OR "operating system").ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:50
S12	6533	(os OR "operating system").ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:51
S13	73	(os OR "operating system").ti. AND "thin client"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:51
S14	1	boot$.ti. AND "root image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:55
S15	5	boot$.ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:55
S16	2	"boot image".ti. AND (root OR custom OR various OR different OR base).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:57
S17	1	("6,421,777").PN.	US-PGPUB; USPAT	OR	OFF	2009/03/31 14:59
S18	0	"bootimage".ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:01
S19	0	"boot image".ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:01
S20	1	"boot image".ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 15:02
S21	31	boot$.ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 15:03
S22	19091	709/220.ccls. 709/203.ccls. 709/208.ccls. 709/222.ccls. 713/2.ccls. 717/174.ccls. 717/175.ccls.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:19
S23	820	S22 AND boot$.ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:19

S24	28	S22 AND ("distributed application" OR "boot image").ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:20

4/1/2009 4:07:09 PM
C:\ Documents and Settings\ pshaw1\ My Documents\ EAST\ Workspaces\ 11026622.wsp

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants	: Nguyen, et al.	Confirmation	: 8329
Serial No.	: 11/026,622	Examiner	: Shaw, Peter
Filing Date	: 12/30/2004	Art Unit	: 2458

Attorney Docket No. : PNTA-P002

For: BRANCHING STORE FILE SYSTEM

AMENDMENT AND RESPONSE TO OFFICE ACTION

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

Dear Sir:

In response to the Office Action mailed April 13, 2009, please consider the following amendments and remarks, in the above captioned patent application.

I hereby certify that this correspondence is being electronically transmitted to the USPTO via EFS on the date shown below:
June 18, 2009
/Shannon Warren/
Shannon Warren

AMENDMENTS TO THE CLAIMS

1. (Currently amended) A branching store file system, comprising:

a root file system comprising a first file collection ~~base image~~ of a distributed application, wherein said first file collection ~~base image~~ is ~~read-only~~ accessible by a plurality of compute nodes;

a plurality of second file collections ~~leaf images~~ comprising modified portions of said distributed application for respective ones of said plurality of compute nodes, wherein an instance of said distributed application for a first of said plurality of compute nodes is instantiated by accessing the first file collection and at least one of the plurality of second file collections, wherein the first file collection and the plurality of second file collections are on separate nodes ~~based on said base image and at least one of said leaf images~~; and

a filter that controls access of instances of said distributed application for said compute nodes.

2. (Currently amended) The system of Claim 1, wherein said plurality of second file collections ~~leaf images~~ comprise branch modification logs for said respective ones of said compute nodes.

3. (Currently amended) The system of Claim 1, wherein said plurality of second file collections ~~leaf images~~ comprise change volumes for said respective ones of said compute nodes.

4. (Original) The system of Claim 1, wherein said filter is implemented at said compute nodes.

5. (Original) The system of Claim 1, wherein said filter is implemented at said root file system.

6. (Currently amended) The system of Claim 1, further comprising intermediate file collections ~~images~~ between said first file collection ~~base image~~ and said plurality of second file collections ~~leaf images~~, wherein said intermediate file collections ~~images~~ comprise application service groups.

7. (Currently amended) The system of Claim 6, wherein access of said compute nodes to said first file collection and intermediate file collections ~~images~~ is read-only.

8. (Original) The system of Claim 1, wherein said distributed application comprises an operating system.

9. (Currently amended) The system of Claim 1, wherein said first file collection ~~base image~~ is concurrently accessible to a plurality of said compute nodes.

10. (Currently amended) The system of Claim 1, wherein said plurality of second file collections ~~leaf images~~ are configured to be modified on a block-by-block basis.

11. (Currently amended) The system of Claim 1, wherein said plurality of second file collections ~~leaf images~~ are configured to be modified on a file-by-file basis.

12. (Currently amended) A branching store file system, comprising:

a root file system having a root metadata structure and a root data structure, said entries in said root metadata structure pointing to associated unmodified data in said root data structure; and

a branch modification log for respective ones of compute nodes, comprising:

a branch data structure comprising modified data reflecting changes made to an instance of said root data structure by said branch modification log's respective compute node; and

a branch metadata structure comprising unmodified and modified entries, said unmodified entries pointing to associated portions of said unmodified data in said root data structure, and said modified entries pointing to associated portions of said modified data in said branch data structure, wherein an instance of a distributed application for a first of said compute nodes is instantiated by accessing ~~defined by~~ said branch metadata structure, wherein said root data structure and said branch data structure are on separate nodes.

13. (Original) The system of Claim 12, further comprising a filter that directs accesses of said distributed application from compute nodes to a respective compute node's instantiation of the distributed application.

14. (Original) The system of Claim 12, wherein said distributed application comprises an operating system.

15. (Original) The system of Claim 13, wherein said filter is configured to traverse inodes to said root data structure for accesses to said un-modified data.

16. (Original) The system of Claim 13, wherein said filter is configured to traverse inodes to said branch data structure for accesses to said modified data.

17. (Original) The system of Claim 12, wherein access to said root data structure by said compute nodes is read-only and access to said branch data structure is read-write-modify by said branch modification log's respective compute node.

18. (Original) The system of Claim 12, wherein ones of said modified entries have a modification attribute attached.

19. (Original) The system of Claim 12, wherein said branch modification log is configured to receive updates to said modified data on a block-by-block basis.

20. (Currently amended) A branching store file system, comprising:

a root file system comprising a base instance of a distributed application;

change volumes for respective ones of compute nodes and comprising modified versions of files of said distributed application; and

a filter program code that directs accesses of said compute nodes' respective instances of said distributed application, wherein said compute nodes' respective instances of said distributed application are instantiated by accessing ~~based on~~ said base instance and respective ones of said change volumes, wherein said base instance and said respective ones of said change volumes are on separate nodes.

21. (Original) The system of Claim 20, further comprising a table of locations indicating locations of said change volumes.

22. (Original) The system of Claim 20, wherein said base instance is concurrently accessible to a plurality of said compute nodes.

23. (Original) The system of Claim 20, wherein said compute nodes have read-only access to files in said base instance.

24. (Original) The system of Claim 21, wherein said distributed application comprises an operating system.

25. (Original) The system of Claim 20, wherein said filter performs a copy-on-write to one of the change volumes when a compute node writes to said compute node's instantiation of said distributed application.

26. (Original) The system of Claim 20, wherein said change volumes are modified on a file-by-file basis.

27. (Currently amended) A computer-implemented method of distributed application deployment, said method comprising:

creating a root file system comprising a first file collection ~~root image~~ of an application;

creating a plurality of second file collections ~~leaf images~~ that are configured to record changes to said first file collection ~~root image~~ of the application for respective compute nodes;

creating a directory configured to access instances of said application for said respective compute nodes, wherein a respective one of the compute node's instance of said application is instantiated by accessing at least a portion of said first file collection and the respective compute node's second file collection, wherein the first file collection and the plurality of second file collections are on separate nodes ~~comprises at least a portion of said root image and the respective compute node's leaf image~~; and

providing said compute nodes with said respective instances of said application.

28. (Currently amended) The method of Claim 27, wherein said second file collections ~~leaf images~~ comprise branch modification logs.

29. (Original) The method of Claim 28, wherein said providing said compute nodes with respective instances of said application comprises traversing modes as indicated by said branch modification logs.

30. (Currently amended) The method of Claim 27, wherein said second file collections ~~leaf images~~ comprise change volumes.

31. (Original) The method of Claim 30, wherein said providing said compute nodes with respective instances of said application comprises consulting a table of locations.

32. (Original) The method of Claim 27, wherein said application comprises an operating system.

33. (Currently amended) The method of Claim 27, wherein access to said first file collection ~~root image~~ by said compute nodes is read-only.

34. (Original) The method of Claim 27, further comprising freezing a version of said distributed application.

35. (Currently amended) The method of Claim 34, further comprising creating intermediate file collections ~~new leaf nodes~~ that are children of said frozen version of said distributed application.

36. (Currently amended) The method of Claim 27, further comprising modifying a respective one ~~first~~ of said second file collections ~~leaf images~~ in response to a corresponding compute node's access to its instance of the application.

37. (Currently amended) The method of Claim 36, wherein said modifying comprises a copying a file from said first file collection ~~root image~~ to said one of said second file collections ~~first of said leaf image~~ and then modifying said file in said one of said second file collections ~~first leaf image~~.

38. (Currently amended) The method of Claim 36, wherein said modifying comprises a copying a data block from said first file collection ~~root image~~ to said one of said second file collections ~~first leaf image~~ and then modifying said ~~data block~~ data block in said one of said second file collections ~~first leaf image~~.

REMARKS

Claims 1-38 remain pending. Claims 1-3, 6-7, 9-12, 20, 27-28, 30, 33, and 35-38 have been amended. No new matter has been added as a result of these amendments.

Claim Rejections – 35 U.S.C. §102

Claims 1-38 are rejected under 35 U.S.C. 102(b) as being anticipated by Chang (US 2003/0126242), hereinafter "Chang."

As is well established, anticipation requires the presence of a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. There must be no difference between the claimed invention and the reference disclosure, as viewed by a person of ordinary skill in the field of the invention. Scripps Clinic & Research Found. v. Genentech Inc., 18 USPQ 2d 1001, 1010 (Fed. Cir. 1991).

Applicants contend that Chang fails to teach or suggest all of the claimed elements of Independent Claims 1, 12, 20, and 27.

Claims 1 and 12:

Claim 1 recites in part:

> a root file system comprising a first file collection of a distributed application, wherein said first file collection is accessible by a plurality of compute nodes;
>
> a plurality of second file collections comprising modified portions of said distributed application for respective ones of said plurality of compute nodes, wherein an instance of said distributed application for a first of said plurality of compute nodes is instantiated by accessing the first file collection and at least one of the plurality of second file collections,

> wherein the first file collection and the plurality of second file collections are on separate nodes…(emphasis added).

Applicants respectfully contend that Chang does not teach "wherein an instance of said distributed application for a first of said plurality of compute nodes is instantiated by accessing the first file collection and at least one of the plurality of second file collections, wherein the first file collection and the plurality of second file collections are on separate nodes," as claimed in Claim 1.

As understood by Applicants, Chant discloses a network and system for network booting of clients linked to a network, wherein each client image copy is unique to a client device and is created using a snapshot of a base boot image with a virtual copy or reverse snapshot of the base image being stored for each client device, such that each client image copy includes the base boot image and information specific to the client (Abstract). Chang discloses a unique, private image of the operating system, applications, and other information for each client (Paragraphs 02, and 33), rather than each instance instantiated by accessing the first file collection and at least one of the plurality of second file collections, as claimed in Claim 1. In other words, where Applicants claim compute nodes each accessing the root file system itself (by accessing the first file collection) and at least one of the plurality of second file collections for each respective compute node, as claimed in Claim 1, Chang teaches the dedication of a boot image for each compute node that contains a unique copy of the common OS and application blocks for each client (Para 31-32, Figure 3).

For this reason, Applicants respectfully submit that Independent Claim 1 is in condition for allowance. Further, Applicants respectfully assert that dependent Claims 2-11 are patentable by virtue of their dependency, as well as for their additional patentable features.

Claim 12 recites in part:

> a root file system having a root metadata structure and a root data structure, said entries in said root metadata structure pointing to associated unmodified data in said root data structure; and
> a branch modification log for respective ones of compute nodes, comprising:
> a branch data structure comprising modified data reflecting changes made to an instance of said root data structure by said branch modification log's respective compute node; and
> a branch metadata structure comprising unmodified and modified entries, said unmodified entries pointing to associated portions of said unmodified data in said root data structure, and said modified entries pointing to associated portions of said modified data in said branch data structure, wherein an instance of a distributed application for a first of said compute nodes is instantiated by accessing said branch metadata structure, wherein said root data structure and said branch data structure are on separate nodes (emphasis added).

Applicants respectfully contend that Chang does not teach "a branch metadata structure comprising unmodified and modified entries, said unmodified entries pointing to associated portions of said unmodified data in said root data structure, and said modified entries pointing to associated portions of said modified data in said branch data structure, wherein an instance of a distributed application for a first of said compute nodes is instantiated by accessing said branch metadata structure, wherein said root data structure and said branch data structure are on separate nodes," as claimed in Claim 12. Chang teaches a unique, private boot image (whether created by the snapshot, reverse snapshot

or lazy clone methods) of the operating system, applications, and other information for each client, wherein each client's instance of the distributed application is instantiated by the private boot image (Paragraphs 02, 25-27,and 33), rather than each compute node accessing a branch modification log that includes a branch metadata structure comprising both unmodified and modified entries, with the unmodified entries pointing to associated portions of the unmodified data in the root data structure, wherein an instance of a distributed application for a first of said compute nodes is instantiated by accessing said branch metadata structure, wherein said root data structure and said branch data structure are on separate nodes, as claimed in Claim 12.

Whether through the use of pointers to the root data structure, as claimed in Claim 12 or directly reading data from a root file system comprising a first file collection, as claimed in Claim 1, Applicants claim compute nodes with respective instances of a distributed application instantiated by accessing the root data structure directly, rather than through a dedicated image for each compute node that contains a unique copy of the common OS and application blocks for each client, as taught by Chang (Para 31-32, Figure 3).

For this reason, Applicants respectfully submit that Independent Claim 12 is in condition for allowance. Further, Applicants respectfully assert that dependent Claims 13-19 are patentable by virtue of their dependency, as well as for their additional patentable features.

Claim 20:

Claim 20 recites in part:

> a root file system comprising a base instance of a distributed application;
> change volumes for respective ones of compute nodes and comprising modified versions of files of said distributed application; and
> a filter program code that directs accesses of said compute nodes' respective instances of said distributed application, wherein said compute nodes' respective instances of said distributed application are instantiated by accessing said base instance and respective ones of said change volumes, wherein said base instance and said respective ones of said change volumes are on separate nodes (emphasis added).

For at least the same or similar reasons as stated above, Applicants respectfully contend that Chang does not teach "a filter program code that directs accesses of said compute nodes' respective instances of said distributed application, wherein said compute nodes' respective instances of said distributed application are instantiated by accessing said base instance and respective ones of said change volumes, wherein said base instance and said respective ones of said change volumes are on separate nodes," as claimed in Claim 20.

As stated above, Chang teaches the network booting of clients linked to the network, wherein each client is booted from a respective boot image that includes a separate and unique image of the operating system, applications, and other information for each client (Paragraphs 02, and 33), rather than each respective compute nodes' instance being instantiated by accessing said base instance and respective ones of said change volumes, wherein said base instance and said respective ones of said change volumes are on separate nodes, as claimed in Claim 20.

For this reason, Applicants respectfully submit that Independent Claim 20 is in condition for allowance. Further, Applicants respectfully assert that dependent Claims

21-26 are patentable by virtue of their dependency, as well as for their additional patentable features.

Claim 27:

Claim 27 recites in part:

> creating a root file system comprising a first file collection of an application;
> creating a plurality of second file collections that are configured to record changes to said first file collection of the application for respective compute nodes;
> creating a directory configured to access instances of said application for said respective compute nodes, wherein a respective one of the compute node's instance of said application is instantiated by accessing at least a portion of said first file collection and the respective compute node's second file collection, wherein the first file collection and the plurality of second file collections are on separate nodes; and
> providing said compute nodes with said respective instances of said application (emphasis added).

For at least the same or similar reasons as stated above, Applicants respectfully contend that Chang does not teach "a root file system comprising a root image of an application; creating a plurality of leaf images that are configured to record changes to said root image of the application for respective compute nodes…wherein a respective one of the compute node's instance of said application comprises at least a portion of said root image and the respective compute node's leaf image," as claimed in Claim 27.

As stated above, Chang teaches the network booting of clients linked to the network, wherein each client is booted from a individual boot image that includes a separate and unique image of the operating system, applications, and other information

for each client (Paragraphs 02, and 33), rather than each respective compute nodes' instance being "instantiated by accessing at least a portion of said first file collection and the respective compute node's second file collection, wherein the first file collection and the plurality of second file collections are on separate nodes," as claimed in Claim 27.

For this reason, Applicants respectfully submit that Independent Claim 27 is in condition for allowance. Further, Applicants respectfully assert that dependent Claims 28-38 are patentable by virtue of their dependency, as well as for their additional patentable features.

For the above reasons, the Applicant requests reconsideration and withdrawal of the rejections under 35 U.S.C. §102.

CONCLUSION

In light of the above listed remarks, reconsideration of rejected Claims is requested. Based on the arguments presented above, it is respectfully submitted that Claims 1-38 overcome the rejections of record and, therefore, allowance of Claims 1-38 is earnestly solicited.

Please charge any additional fees that may be required to maintain pendency of the present application, or apply any credits to our PTO deposit account number: 50-4160.

Respectfully submitted,

MURABITO, HAO & BARNES LLP

Dated: June 18, 2009

/James P. Hao/

James P. Hao
Registration No. 36,398

MURABITO, HAO & BARNES LLP
Two North Market Street
Third Floor
San Jose, California 95113

(408) 938-9060 Voice
(408) 938-9069 Facsimile

Electronic Acknowledgement Receipt

EFS ID:	5544666
Application Number:	11026622
International Application Number:	
Confirmation Number:	8329
Title of Invention:	Branching store file system
First Named Inventor/Applicant Name:	Tung M. Nguyen
Customer Number:	70848
Filer:	James P. Hao/Shannon Warren
Filer Authorized By:	James P. Hao
Attorney Docket Number:	PNTA-P002
Receipt Date:	18-JUN-2009
Filing Date:	30-DEC-2004
Time Stamp:	16:25:37
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		PNTA-P002-Response.pdf	102685 a1296321b9b2b2f8803a72f1e3500a25e1f8052f	yes	17

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Amendment/Req. Reconsideration-After Non-Final Reject	1	1
	Claims	2	9
	Applicant Arguments/Remarks Made in an Amendment	10	17

Warnings:	
Information:	
Total Files Size (in bytes):	102685

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.

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. 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	Filing Date	☐ To be Mailed
11/026,622	12/30/2004	

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY ☐ RATE ($)	SMALL ENTITY FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	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).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT	06/18/2009	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 38	Minus	** 38	= 0	X $ =		OR	X $52=	0
	Independent (37 CFR 1.16(h))	* 4	Minus	***4	= 0	X $ =		OR	X $220=	0
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

AMENDMENT		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)	OR	RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	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.

Legal Instrument Examiner:
/Theresa Dawkins/

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.
11/026,622	12/30/2004	Tung M. Nguyen	PNTA-P002	8329

70848 7590 08/24/2009
PANTA c/o MURABITO HAO & BARNES LLP
Two Market Street
3rd Floor
San Jose, CA 95113

EXAMINER
SHAW, PETER C

ART UNIT	PAPER NUMBER
2458	

MAIL DATE	DELIVERY MODE
08/24/2009	PAPER

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

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	11/026,622	NGUYEN ET AL.	
	Examiner	Art Unit	
	PETER SHAW	2458	

-- 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 *18 June 2009*.

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) *1-38* 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) *1-38* 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 *30 December 2004* 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/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-38 are pending in this action.

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.

1. Claims 1-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chang (US PGPUB No. 2003/0126242) in view of Haun et al. (US Patent No. 6,751,658) [hereinafter "Haun"].

As per claim 1, Chang teaches a branching store file system, comprising:

> a root file system comprising a first file collection of a distributed application ([0011], line 14, "base image"), wherein said first file collection is read-only accessible by a plurality compute nodes ([0026], line 2, "base image never written by the clients") ([0022], line 15, first file collection is a building block for boot images, changes are made to copies to the base image);

a plurality of second file collections comprising modified portions of said distributed application for respective ones of said plurality of compute nodes ([0026], line 3, “client specific blocks”); and

a filter that controls access of instances of said distributed application for said compute nodes ([0017], lines 7-11, “external storage controller…controlling access to…boot images”).

Chang does not explicitly teach an instance of said distributed application for a first of said compute nodes is instantiated by accessing said first file collection and at least one of said plurality of second file collections, wherein the first file collection and the plurality of second file collections are on separate nodes. Haun teaches an instance of said distributed application for a first of said compute nodes is instantiated by accessing said first file collection (Col. 14, lines 10-11, core system volume) and at least one of said plurality of second file collections (Col. 14, line 28-29, user shadow volume), wherein the first file collection and the plurality of second file collections are on separate nodes (Col. 6, lines 45-47, volumes, i.e. file collections, application volumes and client data, i.e. second file collections, can be distributed across multiple server systems, i.e. on separate nodes).

At the time of invention, it would have been obvious to one of ordinary skill in the art to combine Chang, with the teachings of Haun, an instance of said distributed application for a first of said compute nodes is instantiated by accessing said first file

collection and at least one of said plurality of second file collections, wherein the first file collection and the plurality of second file collections are on separate nodes, to allow the distributed application to be modifiable as it is distributed away from the root file system.

As per claim 2, Chang teaches said plurality of second file collections comprising branch modification logs for said respective ones of said compute nodes ([0027], client image copies containing just the modifications to the base image, i.e. client-specific writes) ([0024], line 8, new block modifications logged and included in new image).

As per claim 3, Chang teaches said plurality of second file collections comprising change volumes for said respective ones of said compute nodes ([0025], lines 14-15, new blocks written to original volume to form new volumes).

As per claim 4, Chang teaches said filter implemented at said compute nodes ([0018], lines 1-3, “in one embodiment, the clients are configured...to provide similar or identical functions").

As per claim 5, Chang teaches said filter implemented at said root file system ([0017], line 1, "external storage controller residing in the network boot system").

As per claim 6, Chang teaches intermediate images between said first file collection and said plurality of second file collections ([0023], line 12, “client group images”), wherein

said intermediate images comprise application service groups ([0023], line 12, “client group images”).

As per claim 7, Chang teaches access of said compute nodes to said intermediate images being read-only ([0026], line 2, “base image never written by the clients”).

As per claim 8, Chang teaches said distributed application comprising an operating system ([0016], line 10, “operating system”).

As per claim 9, Chang teaches said first file collection concurrently accessible to a plurality of said compute nodes ([0017], line 3, The boot system provides access to boot images for a “plurality of clients.”).

As per claim 10, Chang teaches said plurality of second file collections configured to be modified on a block-by-block basis ([0024], line 5, “data blocks”).

As per claim 11, Chang teaches said plurality of second file collections configured to be modified on a file-by-file basis ([0024], line 5, “files”).

As per claim 12, Chang teaches a branching store file system, comprising:

a root file system having a root metadata structure and a root data structure, said entries in said root metadata structure pointing to associated unmodified data in said root data structure ([0008], line 12, “Cloning…provide two collections of hard links to the same files”); and

a branch modification log for respective ones of compute nodes, comprising:

a branch data structure comprising modified data reflecting changes made to an instance of said root data structure by said branch modification log's respective compute node ([0027], line 10, The lazy clone technique does not contain an image, only provides instruction as to how to modify the base image to properly produce the client-specific image, i.e. metadata.); and

a branch metadata structure comprising unmodified and modified entries, said unmodified entries pointing to associated portions of said unmodified data in said root data structure ([0008], line 12, “Cloning…provide two collections of hard links to the same files”), and said modified entries pointing to associated portions of said modified data in said branch data structure ([0027], line 10, The lazy clone technique does not contain an image, only provides instruction as to how to modify the first file collection to properly produce the client-specific image, i.e. metadata.), wherein an

instance of a distributed application for a first of said compute nodes is defined by said branch metadata structure ([0027], line 8, It is inherent that the application is defined by the metadata contained in the client-specific writes.).

Chang does not explicitly teach an instance of said distributed application for a first of said compute nodes is instantiated by accessing, said branch metadata structure, wherein the root data structure and said branch data structure are on separate nodes. Haun teaches an instance of said distributed application for a first of said compute nodes is instantiated by accessing, said branch metadata structure (Col. 14, line 28-29, user shadow volume), wherein the root data structure and said branch data structure are on separate nodes (Col. 6, lines 45-47, volumes, i.e. file collections, application volumes and client data, i.e. second file collections, can be distributed across multiple server systems, i.e. on separate nodes).

At the time of invention, it would have been obvious to one of ordinary skill in the art to combine Chang, with the teachings of Haun, an instance of said distributed application for a first of said compute nodes is instantiated by accessing, said branch metadata structure, wherein the root data structure and said branch data structure are on separate nodes, to allow the distributed application to be modifiable as it is distributed away from the root file system.

As per claim 13, Chang teaches a filter that directs accesses of said distributed application from compute nodes ([0017], lines 7-11, “external storage controller…controlling access to…boot images”) to a respective compute node's instantiation of the distributed application ([0030], lines 4-5, “access to client-specific image copy”).

As per claim 14, the substance of the claim language is identical to that of claim 8. Accordingly, this claim is rejected under the same rationale.

As per claim 15, Chang teaches said filter configured to traverse nodes to said root data structure for accesses to said unmodified data ([0033], lines 17-18, unmodified common data stored at separate location from updated client-specific blocks, Traversal to common data is inherent given the successful production of a copy image.).

As per claim 16, Chang teaches said filter configured to traverse nodes to said branch data structure for accesses to said modified data ([0033], lines 17-18, unmodified common data stored at separate location from updated client-specific blocks, Traversal to common data is inherent given the successful production of a copy image.) ([0023], line 5 and 12, Traversal to a varying first file collection or a client group image would be similarly inherent.).

As per claim 17, Chang teaches access to said root data structure by said compute nodes being read-only ([0026], line 2, “base image never written by the clients”) ([0022], line 15, base image is a building block for boot images, changes are made to copies to the base image) and access to said branch data structure being read-write-modify by said branch modification log's respective compute node ([0030], line 13, Given that the client boots from the image, it is inherent that it reads it.) ([0033], line 1, writes or changes received from clients).

As per claim 18, Chang teaches ones of said modified entries having a modification attribute attached ([0027], line 7, Lazy clone technique logs changes to the first file collection instead of writing an entire boot image. It is inherent that each of these entries would include both a pointer and what is being changed, i.e. modification attribute.).

As per claim 19, Chang teaches said branch modification log configured to receive updates to said modified data on a block-by-block basis ([0024], line 5, “data blocks”).

As per claim 20, Chang teaches a branching store file system, comprising:

> a root file system comprising a base instance of a distributed application ([0011], line 14, “base image”);

change volumes for respective ones of compute nodes and comprising modified versions of files of said distributed application ([0025], lines 14-15, new blocks written to original volume to form new volumes); and

a filter program code that directs accesses of said compute nodes' respective instances of said distributed application ([0017], lines 7-11, “external storage controller…controlling access to…boot images”), wherein said compute nodes' respective instances of said distributed application are based on said base instance ([0011], line 14, “base image”) and respective ones of said change volumes ([0022], line 16, “client-specific image").

Chang does not explicitly teach an instance of said distributed application for a first of said compute nodes is instantiated by accessing, said base instance and respective ones of said change volumes, wherein the base instance and said change volumes are on separate nodes. Haun teaches an instance of said distributed application for a first of said compute nodes is instantiated by accessing, said base instance (Col. 14, lines 10-11, core system volume) and respective ones of said change volumes (Col. 14, line 28-29, user shadow volume), wherein the base instance and said change volumes are on separate nodes (Col. 6, lines 45-47, volumes, i.e. file collections, application volumes and client data, i.e. second file collections, can be distributed across multiple server systems, i.e. on separate nodes).

At the time of invention, it would have been obvious to one of ordinary skill in the art to combine Chang, with the teachings of Haun, an instance of said distributed application for a first of said compute nodes is instantiated by accessing, said base instance and respective ones of said change volumes, wherein the base instance and said change volumes are on separate nodes, to allow the distributed application to be modifiable as it is distributed away from the root file system.

As per claim 21, Chang teaches a table of locations indicating locations of said change volumes ([0035], line 9, discovery of the pooled storage device, id of the specific-client image, i.e. change volumes).

As per claim 22, Chang teaches said base instance concurrently accessible to a plurality of said compute nodes ([0017], line 3, The boot system provides access to boot images for a “plurality of clients.”).

As per claim 23, Chang teaches said compute nodes having read-only access to files in said base instance ([0026], line 2, “base image never written by the clients”) ([0022], line 15, base image is a building block for boot images, changes are made to copies to the base image).

As per claim 24, the substance of the claim language is identical to that of claim 8. Accordingly, this claim is rejected under the same rationale.

As per claim 25, Chang teaches said filter performs a copy-on-write to one of the change volumes when a compute node writes to said compute node's instantiation of said distributed application ([0024], lines 1-3, “snapshot manager further functions to control copy-on-writes to the client-specific image copies from the clients”).

As per claim 26, Chang teaches said change volumes are modified on a file-by-file basis ([0024], line 5, “files”).

As per claim 27, Chang teaches a computer-implemented method of distributed application deployment, said method comprising:

> creating a root file system comprising a root image of an application ([0011], line 14, “base image”);
>
> creating a plurality of plurality of second file collections that are configured to record changes to said root image of the application for respective compute nodes ([0022], line 16, “client-specific image");
>
> creating a directory configured to access instances of said application for said respective compute nodes ([0029], lines 20-21, pool storage device with paths to each client image), wherein a respective one of the compute node's instance of

said application comprises at least a portion of said root image and the respective compute node's leaf image ([0024], line 8, New image is a combination of first file collection and new blocks.); and

providing said compute nodes with said respective instances of said application ([0017], lines 7-11, the external storage controller provides access).

Chang does not explicitly teach an instance of said distributed application for a first of said compute nodes is instantiated by accessing at least a portion of said first file collection and the respective node's second file collection, wherein the first file collection and the respective node's second file collection are on separate nodes. Haun teaches an instance of said distributed application for a first of said compute nodes is instantiated by accessing at least a portion said first file collection (Col. 14, lines 10-11, core system volume) and the respective node's second file collection (Col. 14, line 28-29, user shadow volume), wherein the first file collection and the respective node's second file collection are on separate nodes (Col. 6, lines 45-47, volumes, i.e. file collections, application volumes and client data, i.e. second file collections, can be distributed across multiple server systems, i.e. on separate nodes).

At the time of invention, it would have been obvious to one of ordinary skill in the art to combine Chang, with the teachings of Haun, an instance of said distributed application for a first of said compute nodes is instantiated by accessing at least a portion aid first file collection and the respective node's second file collection, wherein

the first file collection and the respective node's second file collection are on separate nodes, to allow the distributed application to be modifiable as it is distributed away from the root file system.

As per claim 28, the substance of the claim language is identical to that of claim 2. Accordingly, this claim is rejected under the same rationale.

As per claim 29, Chang teaches traversing nodes as indicated by said branch modification logs ([0033], lines 17-18, unmodified common data stored at separate location from updated client-specific blocks, Traversal to common data is inherent given the successful production of a copy image.).

As per claim 30, the substance of the claim language is identical to that of claim 3. Accordingly, this claim is rejected under the same rationale.

As per claim 31, Chang teaches consulting a table of locations (The location of the storage controller and the copy image are implicitly stored on some form of a table.).

As per claim 32, the substance of the claim language is identical to that of claim 8. Accordingly, this claim is rejected under the same rationale.

As per claim 33, Chang teaches access to said root image by said compute nodes being read-only ([0026], line 2, “base image never written by the clients”) ([0022], line 15, first file collection is a building block for boot images, changes are made to copies to the base image).

As per claim 34, Chang teaches freezing a version of said distributed application ([0022], line 3 and 10, “snapshot of a boot image…deployed as a base image”).

As per claim 35, Chang teaches creating intermediate file collections that are children of said frozen version of said distributed application ([0022], line 6, “for generating new images”).

As per claim 36, Chang teaches modifying a first of said plurality of second file collections in response to a corresponding compute node's access to its instance of the application ([0027], line 7, In the lazy clone technique where only modifications are stored on “client images,” the image is modified by combining it with the common blocks from the base image.).

As per claim 37, Chang teaches copying a file from said root image to said first leaf image and then modifying said title in said first leaf image ([0023], line 12, When a “client image” is formed, it is implicit that its name will differ from the base image.).

As per claim 38, Chang teaches copying a data block from said root image to said first leaf image ([0024], lines 7, “copied block”) and then modifying said data block data block in said first leaf image ([0033], line 1, “writes or changes to image”).

Response to Arguments

2. Applicant's arguments with respect to the prior art rejections of claims 1-38 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

3. **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.

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to PETER SHAW whose telephone number is (571)270-

7179. The examiner can normally be reached on Monday - Friday 7:30 A.M. to 5:00 P.M.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, JOSEPH AVELLINO can be reached on 571-272-3905. 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.

/P. S./ August 17, 2009
Examiner, Art Unit 2458

/Joseph E. Avellino/
Supervisory Patent Examiner, Art Unit 2458

Notice of References Cited	Application/Control No. 11/026,622	Applicant(s)/Patent Under Reexamination NGUYEN ET AL.	
	Examiner PETER SHAW	Art Unit 2458	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,751,658	06-2004	Haun et al.	709/222
	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/Control No.	Applicant(s)/Patent Under Reexamination
	11026622	NGUYEN ET AL.
	Examiner	**Art Unit**
	PETER SHAW	2458

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

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

CLAIM		DATE								
Final	Original	03/31/2009	08/17/2009							
	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	✓	✓							

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11026622	NGUYEN ET AL.
	Examiner	**Art Unit**
	PETER SHAW	2458

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

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

CLAIM		DATE								
Final	Original	03/31/2009	08/17/2009							
	37	✓	✓							
	38	✓	✓							

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11026622	NGUYEN ET AL.
	Examiner	**Art Unit**
	PETER SHAW	2458

SEARCHED

Class	Subclass	Date	Examiner
709	220, 222, 203, 208	3/31/2009	PS
713	2	3/31/2009	PS
717	174,175	3/31/2009	PS
709	220, 222, 203, 208	8/17/2009	PS
713	2	8/17/2009	PS
717	174,175	8/17/2009	PS

SEARCH NOTES

Search Notes	Date	Examiner
EAST Database Search (hist. incl.)	3/31/2009	PS
Inventor Search (no relevant art found)	3/31/2009	PS
Interpreted Fig. 3 of drawings to comprise hardware	4/8/2009	PS
EAST Database Search (hist. incl.)	8/17/2009	PS
Inventor Search (no relevant art found)	8/17/2009	PS

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L9	20293	709/220.ccls. 709/203.ccls. 709/208.ccls. 709/222.ccls. 713/2.ccls. 717/174.ccls. 717/175.ccls.	US-PGPUB; USPAT	OR	OFF	2009/08/17 14:31
L10	31	L9 AND ("distributed application" OR "boot image").ti.	US-PGPUB; USPAT	OR	OFF	2009/08/17 14:31
L11	199	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	ON	2009/08/17 14:32
L12	30	(runtime OR dynamic).ti. AND boot.ti.	US-PGPUB; USPAT	OR	ON	2009/08/17 14:32
L13	1	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	OFF	2009/08/17 14:32
S1	1	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	OFF	2009/03/31 14:02
S2	190	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:02
S3	548	"file system".ti. AND image	US-PGPUB; USPAT	OR	ON	2009/03/31 14:03
S4	4	"file system".ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:03
S5	120453	"operating system" AND image	US-PGPUB; USPAT	OR	ON	2009/03/31 14:07
S6	195	"distributed application".ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:19
S7	133	"distributed application".ti. AND "operating system"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:19

S8	15	"thin client" AND "operating system" AND (different OR version).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:25
S9	158308	"operating system" AND "thin client" (image OR configuration).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:27
S10	172	"operating system" AND "thin client" AND (image OR configuration).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:27
S11	1	"thin client".ti. AND (os OR "operating system").ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:50
S12	6533	(os OR "operating system").ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:51
S13	73	(os OR "operating system").ti. AND "thin client"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:51
S14	1	boot$.ti. AND "root image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:55
S15	5	boot$.ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:55
S16	2	"boot image".ti. AND (root OR custom OR various OR different OR base).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:57
S17	1	("6,421,777").PN.	US-PGPUB; USPAT	OR	OFF	2009/03/31 14:59
S18	0	"bootimage".ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:01
S19	0	"boot image".ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:01
S20	1	"boot image".ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 15:02
S21	31	boot$.ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 15:03

S22	19091	709/220.ccls. 709/203.ccls. 709/208.ccls. 709/222.ccls. 713/2.ccls. 717/174.ccls. 717/175.ccls.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:19
S23	820	S22 AND boot$.ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:19
S24	28	S22 AND ("distributed application" OR "boot image").ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:20
S25	0	(runtime OR dynamic).ti. AND "boot image".ti.	US-PGPUB; USPAT	OR	OFF	2009/04/01 13:25
S26	1	(runtime OR dynamic).ti. AND "boot image".ti.	US-PGPUB; USPAT	OR	ON	2009/04/01 13:25
S27	30	(runtime OR dynamic).ti. AND boot.ti.	US-PGPUB; USPAT	OR	ON	2009/04/01 13:26

8/17/2009 2:33:06 PM
H:\mybackup\mydocuments\EAST\Workspaces\11026622.wsp

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
(Submitted Only via EFS-Web)

Application Number	11026622	Filing Date	2004-12-30	Docket Number (if applicable)	PNTA-P002	Art Unit	2458
First Named Inventor	NGUYEN, et al.			Examiner Name	SHAW, Peter C.		

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. The Instruction Sheet for this form is located at WWW.USPTO.GOV

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).

☐ 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.

- ☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on _____
- ☐ Other _____

☒ Enclosed

- ☒ Amendment/Reply
- ☐ Information Disclosure Statement (IDS)
- ☐ Affidavit(s)/ Declaration(s)
- ☐ Other _____

MISCELLANEOUS

☐ 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)

☐ Other _____

FEES

The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.
☒ The Director is hereby authorized to charge any underpayment of fees, or credit any overpayments, to Deposit Account No 504160

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

☒ Patent Practitioner Signature
☐ Applicant Signature

Signature of Registered U.S. Patent Practitioner			
Signature	/James P. Hao/	Date (YYYY-MM-DD)	2009-11-17
Name	James P. Hao	Registration Number	36398

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.

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 the Freedom of Information Act requires disclosure of these record s.

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 inspections 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.

Electronic Patent Application Fee Transmittal

Application Number:	11026622
Filing Date:	30-Dec-2004
Title of Invention:	Branching store file system
First Named Inventor/Applicant Name:	Tung M. Nguyen
Filer:	James P. Hao/Shannon Warren
Attorney Docket Number:	PNTA-P002

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Request for continued examination	1801	1	810	810
		Total in USD ($)		**810**

Electronic Acknowledgement Receipt

EFS ID:	6473197
Application Number:	11026622
International Application Number:	
Confirmation Number:	8329
Title of Invention:	Branching store file system
First Named Inventor/Applicant Name:	Tung M. Nguyen
Customer Number:	70848
Filer:	James P. Hao/Shannon Warren
Filer Authorized By:	James P. Hao
Attorney Docket Number:	PNTA-P002
Receipt Date:	17-NOV-2009
Filing Date:	30-DEC-2004
Time Stamp:	19:05:17
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Electronic Funds Transfer
Payment was successfully received in RAM	$810
RAM confirmation Number	5997
Deposit Account	
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if app.)

1	Amendment Submitted/Entered with Filing of CPA/RCE	PNTA-P002-RCEResponse.pdf	127329 4c2d4b5327c3d032bf88eca1df85c50f50cde9b8	no	19
Warnings:					
Information:					
2	Request for Continued Examination (RCE)	PNTA-P002-RCETran.pdf	698433 4fba33c4e72ceb6dbc3066d95d0462ac54fa2ffe	no	3
Warnings:					
Information:					
3	Fee Worksheet (PTO-875)	fee-info.pdf	30056 fef71bdb87cb3c8ef6fa67b7d07cb0bfbf41fb9d	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	855818		

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.

<u>New Applications Under 35 U.S.C. 111</u>
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.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
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.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: Nguyen, et al.	Confirmation	: 8329
Serial No.	: 11/026,622	Examiner	: Shaw, Peter C.
Filing Date	: 12/30/2004	Art Unit	: 2458

Attorney Docket No. : PNTA-P002

For: BRANCHING STORE FILE SYSTEM

AMENDMENT AND RESPONSE TO OFFICE ACTION WITH REQUEST FOR CONTINUED EXAMINATION (RCE)

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

Dear Sir:

In response to the Office Action mailed August 24, 2009, please consider the following amendments and remarks in the above captioned patent application.

I hereby certify that this correspondence is being electronically transmitted to the USPTO via EFS on the date shown below:
November 17, 2009
/Shannon Warren/
Shannon Warren

Amendments to the Claims:

1. (Currently amended) A branching store file system, comprising:

a root file system comprising a single first file collection of a distributed application, wherein said first file collection is accessible by a plurality of compute nodes, and wherein said single first file collection is a complete, unmodified, read-only application shared by the plurality of compute nodes;

a plurality of second file collections comprising modified portions of said distributed application for respective ones of said plurality of compute nodes, wherein an instance of said distributed application for a first of said plurality of compute nodes is instantiated by accessing the first file collection and at least one of the plurality of second file collections, wherein the modified portions of the application for a particular compute node are accessed from the at least one second file collection and the unmodified portions of the application for a particular compute node are accessed from the first file collection, such that each compute node appears to have access to their own version of the application, and wherein the first file collection and the plurality of second file collections are on separate nodes; and

a filter that controls access of instances of said distributed application for said compute nodes, wherein the filter selects at least one of the plurality of second file collections and selects portions of the first file collection to instantiate an instance of the distributed application for a particular compute node, and wherein the filter adds to the particular second file collection of the compute node when the compute node makes changes to its instantiation of the distributed application.

2. (Previously presented) The system of Claim 1, wherein said plurality of second file collections comprise branch modification logs for said respective ones of said compute nodes.

3. (Previously presented) The system of Claim 1, wherein said plurality of second file collections comprise change volumes for said respective ones of said compute nodes.

4. (Original) The system of Claim 1, wherein said filter is implemented at said compute nodes.

5. (Original) The system of Claim 1, wherein said filter is implemented at said root file system.

6. (Previously presented) The system of Claim 1, further comprising intermediate file collections between said first file collection and said plurality of second file collections, wherein said intermediate file collections comprise application service groups.

7. (Previously presented) The system of Claim 6, wherein access of said compute nodes to said first file collection and intermediate file collections is read-only.

8. (Original) The system of Claim 1, wherein said distributed application comprises an operating system.

9. (Previously presented) The system of Claim 1, wherein said first file collection is concurrently accessible to a plurality of said compute nodes.

10. (Previously presented) The system of Claim 1, wherein said plurality of second file collections are configured to be modified on a block-by-block basis.

11. (Previously presented) The system of Claim 1, wherein said plurality of second file collections are configured to be modified on a file-by-file basis.

12. (Currently amended) A branching store file system, comprising:

a root file system having a root metadata structure and a single root data structure, said entries in said root metadata structure pointing to associated unmodified data in said root data structure, wherein the root data structure is an unmodified, read-only application shared by a plurality of compute nodes; and

a branch modification log for respective ones of the plurality of compute nodes, comprising:

a branch data structure comprising modified data reflecting changes made to an instance of said root data structure by said branch modification log's respective compute node; and

a branch metadata structure comprising unmodified and modified entries, said unmodified entries pointing to associated portions of said unmodified data in said root data structure, and said modified entries pointing to associated portions

of said modified data in said branch data structure, wherein an instance of a distributed application for a first of said compute nodes is instantiated by accessing said branch metadata structure, wherein said root data structure and said branch data structure are on separate nodes, wherein the modified portions of the application for a particular compute node are accessed from the branch data structure and the unmodified portions of the application for a particular compute node are accessed from the root data structure, such that portions of modified data and portions of unmodified data are collected to instantiate an instance of the distributed application for a particular compute node, and wherein the branch data structure is added to when the compute node makes changes to its instantiation of the distributed application, such that each compute node appears to have access to their own version of the application.

13. (Original) The system of Claim 12, further comprising a filter that directs accesses of said distributed application from compute nodes to a respective compute node's instantiation of the distributed application.

14. (Original) The system of Claim 12, wherein said distributed application comprises an operating system.

15. (Original) The system of Claim 13, wherein said filter is configured to traverse inodes to said root data structure for accesses to said un-modified data.

16. (Original) The system of Claim 13, wherein said filter is configured to traverse inodes to said branch data structure for accesses to said modified data.

17. (Original) The system of Claim 12, wherein access to said root data structure by said compute nodes is read-only and access to said branch data structure is read-write-modify by said branch modification log's respective compute node.

18. (Original) The system of Claim 12, wherein ones of said modified entries have a modification attribute attached.

19. (Original) The system of Claim 12, wherein said branch modification log is configured to receive updates to said modified data on a block-by-block basis.

20. (Currently amended) A branching store file system, comprising:

a root file system comprising a single base instance of a distributed application, wherein the single base instance is an unmodified, read-only application shared by the plurality of compute nodes;

change volumes for respective ones of a plurality of compute nodes and comprising only modified versions of files of said distributed application; and

a filter program code that directs accesses of said compute nodes' respective instances of said distributed application, wherein said compute nodes' respective instances of said distributed application are instantiated by accessing said base instance and respective ones of said change volumes, wherein said base instance and said

respective ones of said change volumes are on separate nodes, wherein the modified portions of the application for a particular compute node are accessed from the respective ones of said change volumes and the unmodified portions of the application for a particular compute node are accessed from the base instance, such that each compute node appears to have access to their own version of the application, and wherein the filter program adds to the respective ones of said change volumes when the compute node makes changes to its instantiation of the distributed application.

21. (Original) The system of Claim 20, further comprising a table of locations indicating locations of said change volumes.

22. (Original) The system of Claim 20, wherein said base instance is concurrently accessible to a plurality of said compute nodes.

23. (Original) The system of Claim 20, wherein said compute nodes have read-only access to files in said base instance.

24. (Original) The system of Claim 21, wherein said distributed application comprises an operating system.

25. (Original) The system of Claim 20, wherein said filter performs a copy-on-write to one of the change volumes when a compute node writes to said compute node's instantiation of said distributed application.

26. (Original) The system of Claim 20, wherein said change volumes are modified on a file-by-file basis.

27. (Currently amended) A computer-implemented method of distributed application deployment, said method comprising:

creating a root file system comprising a single first file collection of an application;

creating a plurality of second file collections that are configured to record changes to said first file collection of the application for respective compute nodes;

creating a directory configured to access instances of said application for said respective compute nodes, wherein a respective one of the compute node's instance of said application is instantiated by accessing at least a portion of said first file collection and the respective compute node's second file collection, wherein modified portions of the application for a particular compute node are accessed from the respective second file collection and the unmodified portions of the application for a particular compute node are accessed from the first file collection, such that each compute node appears to have access to their own version of the application, wherein the first file collection and the plurality of second file collections are on separate nodes; and

providing said compute nodes with said respective instances of said application.

28. (Previously presented) The method of Claim 27, wherein said second file collections comprise branch modification logs.

29. (Original) The method of Claim 28, wherein said providing said compute nodes with respective instances of said application comprises traversing modes as indicated by said branch modification logs.

30. (Previously presented) The method of Claim 27, wherein said second file collections comprise change volumes.

31. (Original) The method of Claim 30, wherein said providing said compute nodes with respective instances of said application comprises consulting a table of locations.

32. (Original) The method of Claim 27, wherein said application comprises an operating system.

33. (Previously presented) The method of Claim 27, wherein access to said first file collection by said compute nodes is read-only.

34. (Original) The method of Claim 27, further comprising freezing a version of said distributed application.

35. (Previously presented) The method of Claim 34, further comprising creating intermediate file collections that are children of said frozen version of said distributed application.

36. (Previously presented) The method of Claim 27, further comprising modifying a respective one of said second file collections in response to a corresponding compute node's access to its instance of the application.

37. (Previously presented) The method of Claim 36, wherein said modifying comprises a copying a file from said first file collection to said one of said second file collections and then modifying said file in said one of said second file collections.

38. (Previously presented) The method of Claim 36, wherein said modifying comprises a copying a data block from said first file collection to said one of said second file collections and then modifying said data block in said one of said second file collections.

REMARKS

Claims 1-38 are pending. Claims 1, 12, 20, and 27 are amended. No new matter has been added as a result of these amendments.

Claim Rejections – 35 U.S.C. §103

Claims 1-38 are rejected under 35 U.S.C. 103(a) as being allegedly unpatentable over Chang (US 2003/0126242), hereinafter "Chang," in view of Haun, et al. (US 6,751,658), hereinafter "Huan." Applicants respectfully assert that Claims 1-38 are patentable over the cited combination in view of the following.

Applicants respectfully point out that the Examiner has the burden of establishing a prima case of obviousness. To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim features. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on Applicant's disclosure. In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). See MPEP 2100-126. Specifically, "all words in a claim must be considered when judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d., 1382 (CCPA 1970).

Moreover, in response to the recent U.S. Supreme Court decision in *KSR Int'l Co. v. Teleflex, Inc.* (U.S. 2007), new guidelines were set forth for examining obviousness under 35 U.S.C. 103. The U.S. Supreme Court reaffirmed the *Graham* factors and, while not totally rejecting the "teachings, suggestion, or motivation" test, the Court appears to now require higher scrutiny on the part of the U.S. Patent & Trademark Office. In accordance with the recently submitted guidelines, it is "now necessary to identify the reason" why a person of ordinary skill in the art would have combined the elements of cited references, or at least describe the pertinence of the elements set forth in the cited disclosure, in the manner presently claimed.

Applicants contend that the combination of Chang and Haun fails to teach or suggest all the claimed elements of Claims 1-38 in view of the following rationale.

Claim 1:

Amended Claim 1 recites:

> 1. (Currently amended) A branching store file system, comprising:
> a root file system comprising a single first file collection of a distributed application, wherein said first file collection is accessible by a plurality of compute nodes, and wherein said single first file collection is a complete, unmodified, read-only application shared by the plurality of compute nodes;
> a plurality of second file collections comprising modified portions of said distributed application for respective ones of said plurality of compute nodes, wherein an instance of said distributed application for a first of said plurality of compute nodes is instantiated by accessing the first file collection and at least one of the plurality of second file collections, wherein the modified portions of the application for a particular compute node are accessed from the at least one second file collection and the unmodified portions of the application for a particular compute node are accessed from the first file collection, such that each compute node appears to have access to their own version of the application, and wherein

the first file collection and the plurality of second file collections are on separate nodes; and

a filter that controls access of instances of said distributed application for said compute nodes, wherein the filter selects at least one of the plurality of second file collections and selects portions of the first file collection to instantiate an instance of the distributed application for a particular compute node, and wherein the filter adds to the particular second file collection of the compute node when the compute node makes changes to its instantiation of the distributed application.

Applicants respectfully assert that the combination of Chang and Haun does not teach or suggest "a root file system comprising a single first file collection [that]…is a complete, unmodified, read-only application shared by the plurality of compute nodes…wherein an instance of said distributed application for a first of said plurality of compute nodes is instantiated by accessing the first file collection and at least one of the plurality of second file collections, wherein the modified portions of the application for a particular compute node are accessed from the at least one second file collection and the unmodified portions of the application for a particular compute node are accessed from the first file collection, such that each compute node appears to have access to their own version of the application, and wherein the first file collection and the plurality of second file collections are on separate nodes…and a filter [that]…selects at least one of the plurality of second file collections and selects portions of the first file collection to instantiate an instance of the distributed application for a particular compute node, and wherein the filter adds to the plurality of second file collections when the compute node makes changes to its instantiation of the distributed application," as claimed in Claim 1.

The rejection agrees that Chang does not explicitly teach an instance of said distributed application for a first of said compute nodes is instantiated by accessing said

first file collection and at least one of said plurality of second file collections, wherein the first file collection and the plurality of second file collections are on separate nodes, as claimed in Claim 1. Applicants respectfully assert that rather than instantiating an instance for a particular compute node by accessing respective second file collections and portions of the first file collection for the particular compute node, as claimed in Claim 1, Chang discloses a unique, private image of the operating system, applications, and other information for each client (Paragraph 02, and 33). The rejection contends that Haun cures these deficiencies. Applicants respectfully disagree.

As understood by Applicants, Haun is directed towards providing an operating system in a net-booted environment, wherein the network computer client boots from a boot image provided by a server (Abstract). Haun further discloses that in response to attempts by the client to modify the operating system files, the changes are recorded on the server separate from the system volumes storing the operating system files and into a storage area associated with the client (Abstract). As understood by Applicants, the client has access to a shared split OS that includes read-only files in a core system volume as well as a user system volume that contains user-configurable system components, including preferences and all system additions installed by application software (Figure 9; Column 14, lines 1-25). Haun further discloses the creation of a shadow system volume for each connected client (Figure 9; Column 14, lines 26-58). As understood by Applicants, the shadow system volume "shadows" the user system volume by storing modifications that are made to the user system volume, such that whenever a changes are made to the user system volume, they are copied to the shadow system

volume (Figure 9; Column 14, lines 26-58). Haun further notes that data may only be written to the shadow system volume (Column 14, lines 45-52).

To the extent that Haun discloses a read-only volume and a read/write volume that is also copied to a shadow system volume, Haun does not teach or suggest "a root file system comprising a single first file collection…[that] is a complete, unmodified, read-only application shared by the plurality of compute nodes…wherein an instance of said distributed application for a first of said plurality of compute nodes is instantiated by accessing the first file collection and at least one of the plurality of second file collections, wherein the modified portions of the application for a particular compute node are accessed from the at least one second file collection and the unmodified portions of the application for a particular compute node are accessed from the first file collection, such that each compute node appears to have access to their own version of the application, and wherein the first file collection and the plurality of second file collections are on separate nodes…and a filter…[that] selects at least one of the plurality of second file collections and selects portions of the first file collection to instantiate an instance of the distributed application for a particular compute node, and wherein the filter adds to the plurality of second file collections when the compute node makes changes to its instantiation of the distributed application," as claimed in Claim 1. Applicants respectfully assert that the shadow system volume of Haun is not one of a plurality of second file collections, as claimed in Claim 1. The shadow system volume of Haun is not used to instantiate an instance of a distributed application for a compute node, as claimed

in Claim 1; rather it is a copy of the changes made to the user system volume and not used for creating an instance.

Applicants respectfully assert that Haun teaches away from a root file system that contains a single, complete, unmodified, read-only application shared by the plurality of compute nodes, and a plurality of second file collections, at least one of the plurality of second file collections for each compute node, wherein the respective compute node's instantiation of an instance of the distributed application is accomplished by a filter selecting at least one of the plurality of second file collections and selecting portions of the first file collection, as claimed in Claim 1. Rather than a root file system that contains an unmodified copy of the distributed application, Haun discloses that the core system volume preferably contains those parts of the system that do not need to be written back to during system operation, thereby providing the system components that are mandatory for system operation (Column 14, lines 8-12). Haun discloses that these "essential components" are separated from the user system volume and protected as read-only, while the user system volume contains all the user-configurable system components, including preferences and all the system additions installed by application software (Column 14, lines 12-32). Haun further notes that the shadow system volume is a copy of the user system volume.

Therefore, Applicants respectfully assert that embodiments as recited by Claim 1 are not rendered obvious by the combination of Chang and Haun. Accordingly, Applicants respectfully assert that dependent Claims 2-11 are patentable by virtue of their

dependency on an allowable base claim, as well as for their additional recited patentable features.

Claim 12:

Independent Claim 12 recites features similar to that of independent Claim 1, and is therefore patentable for at least the same or similar reasons as recited above. Accordingly, Applicants respectfully assert that dependent Claims 13-19 are patentable by virtue of their dependency on an allowable base claim, as well as for their additional recited patentable features.

Claim 20:

Independent Claim 20 recites features similar to that of independent Claim 1, and is therefore patentable for at least the same or similar reasons as recited above. Accordingly, Applicants respectfully assert that dependent Claims 21-26 are patentable by virtue of their dependency on an allowable base claim, as well as for their additional recited patentable features.

Claim 27:

Independent Claim 27 recites features similar to that of independent Claims 1, 12, and 20, and is therefore patentable for at least the same or similar reasons as recited above. Accordingly, Applicants respectfully assert that dependent Claims 28-38 are patentable by virtue of their dependency on an allowable base claim, as well as for their additional recited patentable features.

For the above reasons, Applicants request reconsideration and withdrawal of the rejections under 35 U.S.C. §103.

CONCLUSION

In light of the above listed remarks, reconsideration of rejected Claims is requested. Based on the arguments presented above, it is respectfully submitted that Claims 1-38 overcome the rejections of record and, therefore, allowance of Claims 1-38 is earnestly solicited.

Please charge any additional fees that may be required to maintain pendency of the present application, or apply any credits to our PTO deposit account number: 50-4160.

Respectfully submitted,

MURABITO, HAO & BARNES LLP

Dated: November 17, 2009

/James P. Hao/

James P. Hao
Registration No. 36,398

MURABITO, HAO & BARNES LLP
Two North Market Street
Third Floor
San Jose, California 95113

(408) 938-9060 Voice
(408) 938-9069 Facsimile

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. 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	Filing Date	☐ To be Mailed
11/026,622	12/30/2004	

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	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).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
	11/17/2009									
	Total (37 CFR 1.16(i))	* 38	Minus	** 38	= 0	X $ =		OR	X $52=	0
	Independent (37 CFR 1.16(h))	* 4	Minus	***4	= 0	X $ =		OR	X $220=	0
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

AMENDMENT		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	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.

Legal Instrument Examiner:
/DONNA D. SMALLS LOGAN/

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.
11/026,622	12/30/2004	Tung M. Nguyen	PNTA-P002	8329

70848 7590 03/15/2010
PANTA c/o MURABITO HAO & BARNES LLP
Two Market Street
3rd Floor
San Jose, CA 95113

EXAMINER
SHAW, PETER C

ART UNIT	PAPER NUMBER
2458	

MAIL DATE	DELIVERY MODE
03/15/2010	PAPER

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

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	11/026,622	NGUYEN ET AL.	
	Examiner	Art Unit	
	PETER SHAW	2458	

-- 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 *17 November 2009*.

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) *1-38* 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) *1-38* 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 *30 December 2004* 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/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-38 are pending in this action.

Continued Examination Under 37 CFR 1.114

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 11/17/2009 has been entered.

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 1-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chang (US PGPUB No. 2003/0126242) in view of Haun et al. (US Patent No. 6,751,658) [hereinafter “Haun”].

As per claim 1, Chang teaches a branching store file system, comprising:

a root file system comprising a first file collection of a distributed application ([0011], line 14, “base image”), wherein said first file collection is read-only accessible by a plurality compute nodes ([0026], line 2, “base image never written by the clients”) ([0022], line 15, first file collection is a building block for boot images, changes are made to copies to the base image);

a plurality of second file collections comprising modified portions of said distributed application for respective ones of said plurality of compute nodes ([0026], line 3, “client specific blocks”);

a filter that controls access of instances of said distributed application for said compute nodes ([0017], lines 7-11, “external storage controller…controlling access to…boot images”).

Chang does not explicitly teach an instance of said distributed application for a first of said compute nodes is instantiated by accessing said first file collection and at least one of said plurality of second file collections, wherein the first file collection and the plurality of second file collections are on separate nodes; wherein said single first file collection is a complete, unmodified, read-only application shared by the plurality of compute nodes; wherein the modified portions of the application for a particular compute node are accessed from the at least one second file collection and the

unmodified portions of the application for a particular compute node are accessed from the first file collection such that each compute node appears to have access to their own version of the application; and wherein the filter selects at least one of the plurality of second file collections and selects portions of the first file collection to instantiate an instance of the distributed application for a particular compute node, and wherein the filter adds to the particular second file collection of the compute node when the compute node makes changes to its instantiation of the distributed application. Haun teaches an instance of said distributed application for a first of said compute nodes is instantiated by accessing said first file collection (Col. 14, lines 10-11, core system volume) and at least one of said plurality of second file collections (Col. 14, line 28-29, user shadow volume), wherein the first file collection and the plurality of second file collections are on separate nodes (Col. 6, lines 45-47, volumes, i.e. file collections, application volumes and client data, i.e. second file collections, can be distributed across multiple server systems, i.e. on separate nodes); wherein said single first file collection is a complete, unmodified, read-only application shared by the plurality of compute nodes; wherein the modified portions of the application for a particular compute node are accessed from the at least one second file collection (Col. 14, lines 18-25, user system volume contains all of the user preferences) and the unmodified portions of the application for a particular compute node are accessed from the first file collection (Col. 14, lines 8-10, core system volume contains all unmodified data) such that each compute node appears to have access to their own version of the application (Col. 14, lines 31-32, user system volume for each client); and wherein the filter selects at least one of the plurality of

second file collections and selects portions of the first file collection to instantiate an instance of the distributed application for a particular compute node (Col. 14, line 56-58, system volume with user volume, i.e. user preferences), and wherein the filter adds to the particular second file collection of the compute node when the compute node makes changes to its instantiation of the distributed application (Col. 14, lines 40-44, making changes to the user volume/shadow volume based on changes made by client).

At the time of invention, it would have been obvious to one of ordinary skill in the art to combine Chang, with the teachings of Haun, an instance of said distributed application for a first of said compute nodes is instantiated by accessing said first file collection and at least one of said plurality of second file collections, wherein the first file collection and the plurality of second file collections are on separate nodes; wherein said single first file collection is a complete, unmodified, read-only application shared by the plurality of compute nodes; wherein the modified portions of the application for a particular compute node are accessed from the at least one second file collection and the unmodified portions of the application for a particular compute node are accessed from the first file collection such that each compute node appears to have access to their own version of the application; and wherein the filter selects at least one of the plurality of second file collections and selects portions of the first file collection to instantiate an instance of the distributed application for a particular compute node, and wherein the filter adds to the particular second file collection of the compute node when the compute node makes changes to its instantiation of the distributed application, to

allow the distributed application to be modifiable as it is distributed away from the root file system.

As per claim 2, the combination of Chang and Haun teaches said plurality of second file collections comprising branch modification logs for said respective ones of said compute nodes (Chang; [0027], client image copies containing just the modifications to the base image, i.e. client-specific writes) (Chang; [0024], line 8, new block modifications logged and included in new image).

As per claim 3, the combination of Chang and Haun teaches said plurality of second file collections comprising change volumes for said respective ones of said compute nodes (Chang; [0025], lines 14-15, new blocks written to original volume to form new volumes).

As per claim 4, the combination of Chang and Haun teaches said filter implemented at said compute nodes (Chang; [0018], lines 1-3, “in one embodiment, the clients are configured...to provide similar or identical functions").

As per claim 5, the combination of Chang and Haun teaches said filter implemented at said root file system (Chang; [0017], line 1, "external storage controller residing in the network boot system").

As per claim 6, the combination of Chang and Haun teaches intermediate images between said first file collection and said plurality of second file collections (Chang; [0023], line 12, “client group images”), wherein said intermediate images comprise application service groups (Chang; [0023], line 12, “client group images”).

As per claim 7, the combination of Chang and Haun teaches access of said compute nodes to said intermediate images being read-only (Chang; [0026], line 2, “base image never written by the clients”).

As per claim 8, the combination of Chang and Haun teaches said distributed application comprising an operating system (Chang; [0016], line 10, “operating system”).

As per claim 9, the combination of Chang and Haun teaches said first file collection concurrently accessible to a plurality of said compute nodes (Chang; [0017], line 3, The boot system provides access to boot images for a “plurality of clients.”).

As per claim 10, the combination of Chang and Haun teaches said plurality of second file collections configured to be modified on a block-by-block basis (Chang; [0024], line 5, “data blocks”).

As per claim 11, the combination of Chang and Haun teaches said plurality of second file collections configured to be modified on a file-by-file basis (Chang; [0024], line 5, “files”).

As per claim 12, Chang teaches a branching store file system, comprising:

> a root file system having a root metadata structure and a root data structure, said entries in said root metadata structure pointing to associated unmodified data in said root data structure (Chang; [0008], line 12, “Cloning…provide two collections of hard links to the same files”); and
>
> a branch modification log for respective ones of the plurality of compute nodes, comprising:
>
> > a branch data structure comprising modified data reflecting changes made to an instance of said root data structure by said branch modification log's respective compute node (Chang; [0027], line 10, The lazy clone technique does not contain an image, only provides instruction as to how to modify the base image to properly produce the client-specific image, i.e. metadata.); and

a branch metadata structure comprising unmodified and modified entries, said unmodified entries pointing to associated portions of said unmodified data in said root data structure (Chang; [0008], line 12, "Cloning…provide two collections of hard links to the same files"), and said modified entries pointing to associated portions of said modified data in said branch data structure (Chang; [0027], line 10, The lazy clone technique does not contain an image, only provides instruction as to how to modify the first file collection to properly produce the client-specific image, i.e. metadata.), wherein an instance of a distributed application for a first of said compute nodes is defined by said branch metadata structure (Chang; [0027], line 8, It is inherent that the application is defined by the metadata contained in the client-specific writes.).

Chang does not explicitly teach an instance of said distributed application for a first of said compute nodes is instantiated by accessing, said branch metadata structure, wherein the root data structure and said branch data structure are on separate nodes; wherein the root data structure is an unmodified, read-only application shared by a plurality of compute nodes; wherein the modified portions of the application for a particular compute node are accessed from the branch data structure and the unmodified portions of the application for a particular compute node are accessed from the root data structure, such that portions of modified data and portions of unmodified data are collected to instantiate an instance of the distributed application for a particular

compute node, and wherein the branch data structure is added to when the compute node makes changes to its instantiation of the distributed application, such that each compute node appears to have access to their own version of the application. Haun teaches an instance of said distributed application for a first of said compute nodes is instantiated by accessing, said branch metadata structure (Col. 14, line 28-29, user shadow volume), wherein the root data structure and said branch data structure are on separate nodes (Col. 6, lines 45-47, volumes, i.e. file collections, application volumes and client data, i.e. second file collections, can be distributed across multiple server systems, i.e. on separate nodes); wherein the root data structure is an unmodified, read-only application shared by a plurality of compute nodes (Col. 14, line 14, core system volume contains read-only data); wherein the modified portions of the application for a particular compute node are accessed from the branch data structure (Col. 14, lines 18-25, user system volume contains all of the user preferences) and the unmodified portions of the application for a particular compute node are accessed from the root data structure (Col. 14, lines 8-10, core system volume contains all unmodified data), such that portions of modified data and portions of unmodified data are collected to instantiate an instance of the distributed application for a particular compute node (Col. 14, line 56-58, system volume with user volume, i.e. user preferences), and wherein the branch data structure is added to when the compute node makes changes to its instantiation of the distributed application (Col. 14, lines 40-44, making changes to the user volume/shadow volume based on changes made by client), such that each

compute node appears to have access to their own version of the application (Col. 14, lines 31-32, user system volume for each client).

At the time of invention, it would have been obvious to one of ordinary skill in the art to combine Chang, with the teachings of Haun, an instance of said distributed application for a first of said compute nodes is instantiated by accessing, said branch metadata structure, wherein the root data structure and said branch data structure are on separate nodes; wherein the root data structure is an unmodified, read-only application shared by a plurality of compute nodes; wherein the modified portions of the application for a particular compute node are accessed from the branch data structure and the unmodified portions of the application for a particular compute node are accessed from the root data structure, such that portions of modified data and portions of unmodified data are collected to instantiate an instance of the distributed application for a particular compute node, and wherein the branch data structure is added to when the compute node makes changes to its instantiation of the distributed application, such that each compute node appears to have access to their own version of the application, to allow the distributed application to be modifiable as it is distributed away from the root file system.

As per claim 13, the combination of Chang and Haun teaches a filter that directs accesses of said distributed application from compute nodes (Chang; [0017], lines 7-11, “external storage controller…controlling access to…boot images”) to a respective

compute node's instantiation of the distributed application ([0030], lines 4-5, “access to client-specific image copy”).

As per claim 14, the substance of the claim language is identical to that of claim 8. Accordingly, this claim is rejected under the same rationale.

As per claim 15, the combination of Chang and Haun teaches said filter configured to traverse nodes to said root data structure for accesses to said unmodified data (Chang; [0033], lines 17-18, unmodified common data stored at separate location from updated client-specific blocks, Traversal to common data is inherent given the successful production of a copy image.).

As per claim 16, the combination of Chang and Haun teaches said filter configured to traverse nodes to said branch data structure for accesses to said modified data (Chang; [0033], lines 17-18, unmodified common data stored at separate location from updated client-specific blocks, Traversal to common data is inherent given the successful production of a copy image.) (Chang; [0023], line 5 and 12, Traversal to a varying first file collection or a client group image would be similarly inherent.).

As per claim 17, the combination of Chang and Haun teaches access to said root data structure by said compute nodes being read-only (Chang; [0026], line 2, “base image never written by the clients”) (Chang; [0022], line 15, base image is a building block for

boot images, changes are made to copies to the base image) and access to said branch data structure being read-write-modify by said branch modification log's respective compute node (Chang; [0030], line 13, Given that the client boots from the image, it is inherent that it reads it.) (Chang; [0033], line 1, writes or changes received from clients).

As per claim 18, the combination of Chang and Haun teaches ones of said modified entries having a modification attribute attached (Chang; [0027], line 7, Lazy clone technique logs changes to the first file collection instead of writing an entire boot image. It is inherent that each of these entries would include both a pointer and what is being changed, i.e. modification attribute.).

As per claim 19, the combination of Chang and Haun teaches said branch modification log configured to receive updates to said modified data on a block-by-block basis (Chang; [0024], line 5, “data blocks”).

As per claim 20, Chang teaches a branching store file system, comprising:

> a root file system comprising a single base instance of a distributed application ([0011], line 14, “base image”);

change volumes for respective ones of a plurality of compute nodes and comprising modified versions of files of said distributed application ([0025], lines 14-15, new blocks written to original volume to form new volumes); and

a filter program code that directs accesses of said compute nodes' respective instances of said distributed application ([0017], lines 7-11, “external storage controller…controlling access to…boot images”), wherein said compute nodes' respective instances of said distributed application are based on said base instance ([0011], line 14, “base image”) and respective ones of said change volumes ([0022], line 16, “client-specific image").

Chang does not explicitly teach an instance of said distributed application for a first of said compute nodes is instantiated by accessing, said base instance and respective ones of said change volumes, wherein the base instance and said change volumes are on separate nodes; wherein the single base instance is an unmodified, read-only application shared by the plurality of compute nodes; wherein the modified ones of said change volumes and the unmodified portions of the application for a particular compute node are accessed from the base instance such that each compute node appears to have access to their own version of the application, and wherein the filter program adds to the respective ones of said change volumes when the compute node makes changes to its instantiation of the distributed application. Haun teaches an instance of said distributed application for a first of said compute nodes is instantiated

by accessing, said base instance (Col. 14, lines 10-11, core system volume) and respective ones of said change volumes (Col. 14, line 28-29, user shadow volume), wherein the base instance and said change volumes are on separate nodes (Col. 6, lines 45-47, volumes, i.e. file collections, application volumes and client data, i.e. second file collections, can be distributed across multiple server systems, i.e. on separate nodes); wherein the single base instance is an unmodified, read-only application shared by the plurality of compute nodes (Col. 14, line 14, core system volume contains read-only data); wherein the modified ones of said change volumes and the unmodified portions of the application for a particular compute node are accessed from the base instance such that each compute node appears to have access to their own version of the application (Col. 14, lines 31-32, user system volume for each client), and wherein the filter program adds to the respective ones of said change volumes when the compute node makes changes to its instantiation of the distributed application (Col. 14, lines 40-44, making changes to the user volume/shadow volume based on changes made by client).

At the time of invention, it would have been obvious to one of ordinary skill in the art to combine Chang, with the teachings of Haun, an instance of said distributed application for a first of said compute nodes is instantiated by accessing, said base instance and respective ones of said change volumes, wherein the base instance and said change volumes are on separate nodes; wherein the single base instance is an unmodified, read-only application shared by the plurality of compute nodes; wherein the modified ones of said change volumes and the unmodified portions of the application for

a particular compute node are accessed from the base instance such that each compute node appears to have access to their own version of the application, and wherein the filter program adds to the respective ones of said change volumes when the compute node makes changes to its instantiation of the distributed application, to allow the distributed application to be modifiable as it is distributed away from the root file system.

As per claim 21, the combination of Chang and Haun teaches a table of locations indicating locations of said change volumes (Chang; [0035], line 9, discovery of the pooled storage device, id of the specific-client image, i.e. change volumes).

As per claim 22, the combination of Chang and Haun teaches said base instance concurrently accessible to a plurality of said compute nodes (Chang; [0017], line 3, The boot system provides access to boot images for a “plurality of clients.”).

As per claim 23, the combination of Chang and Haun teaches said compute nodes having read-only access to files in said base instance (Chang; [0026], line 2, “base image never written by the clients”) (Chang; [0022], line 15, base image is a building block for boot images, changes are made to copies to the base image).

As per claim 24, the substance of the claim language is identical to that of claim 8. Accordingly, this claim is rejected under the same rationale.

As per claim 25, the combination of Chang and Haun teaches said filter performs a copy-on-write to one of the change volumes when a compute node writes to said compute node's instantiation of said distributed application (Chang; [0024], lines 1-3, “snapshot manager further functions to control copy-on-writes to the client-specific image copies from the clients”).

As per claim 26, the combination of Chang and Haun teaches said change volumes are modified on a file-by-file basis (Chang; [0024], line 5, “files”).

As per claim 27, Chang teaches a computer-implemented method of distributed application deployment, said method comprising:

> creating a root file system comprising a single first file collection of an application ([0011], line 14, “base image”);
>
> creating a plurality of plurality of second file collections that are configured to record changes to said first file collection of the application for respective compute nodes ([0022], line 16, “client-specific image");
>
> creating a directory configured to access instances of said application for said respective compute nodes ([0029], lines 20-21, pool storage device with paths to

each client image), wherein a respective one of the compute node's instance of said application comprises at least a portion of said first file collection and the respective compute node's leaf image ([0024], line 8, New image is a combination of first file collection and new blocks.); and

providing said compute nodes with said respective instances of said application ([0017], lines 7-11, the external storage controller provides access).

Chang does not explicitly teach an instance of said distributed application for a first of said compute nodes is instantiated by accessing at least a portion said first file collection and the respective node's second file collection, wherein the first file collection and the respective node's second file collection are on separate nodes; and wherein the modified portion of the application for a particular compute node are accessed from the respective second file collection and the unmodified portions of the application for a particular compute node are accessed from the first file collection, such that each compute node appears to have access to their own version of the application. Haun teaches an instance of said distributed application for a first of said compute nodes is instantiated by accessing at least a portion said first file collection (Col. 14, lines 10-11, core system volume) and the respective node's second file collection (Col. 14, line 28-29, user shadow volume), wherein the first file collection and the respective node's second file collection are on separate nodes (Col. 6, lines 45-47, volumes, i.e. file collections, application volumes and client data, i.e. second file collections, can be

distributed across multiple server systems, i.e. on separate nodes); and wherein the modified portion of the application for a particular compute node are accessed from the respective second file collection (Col. 14, lines 18-25, user system volume contains all of the user preferences) and the unmodified portions of the application for a particular compute node are accessed from the first file collection (Col. 14, lines 8-10, core system volume contains all unmodified data), such that each compute node appears to have access to their own version of the application (Col. 14, lines 31-32, user system volume for each client).

At the time of invention, it would have been obvious to one of ordinary skill in the art to combine Chang, with the teachings of Haun, an instance of said distributed application for a first of said compute nodes is instantiated by accessing at least a portion said first file collection and the respective node's second file collection, wherein the first file collection and the respective node's second file collection are on separate nodes; and wherein the modified portion of the application for a particular compute node are accessed from the respective second file collection and the unmodified portions of the application for a particular compute node are accessed from the first file collection, such that each compute node appears to have access to their own version of the application, to allow the distributed application to be modifiable as it is distributed away from the root file system.

As per claim 28, the substance of the claim language is identical to that of claim 2. Accordingly, this claim is rejected under the same rationale.

As per claim 29, the combination of Chang and Haun teaches traversing nodes as indicated by said branch modification logs (Chang; [0033], lines 17-18, unmodified common data stored at separate location from updated client-specific blocks, Traversal to common data is inherent given the successful production of a copy image.).

As per claim 30, the substance of the claim language is identical to that of claim 3. Accordingly, this claim is rejected under the same rationale.

As per claim 31, the combination of Chang and Haun teaches consulting a table of locations (Haun; Col. 5, lines 6-8, database of addresses).

As per claim 32, the substance of the claim language is identical to that of claim 8. Accordingly, this claim is rejected under the same rationale.

As per claim 33, the combination of Chang and Haun teaches access to said root image by said compute nodes being read-only (Chang; [0026], line 2, “base image never written by the clients”) (Chang; [0022], line 15, first file collection is a building block for boot images, changes are made to copies to the base image).

As per claim 34, Chang teaches freezing a version of said distributed application (Chang; [0022], line 3 and 10, “snapshot of a boot image…deployed as a base image”).

As per claim 35, the combination of Chang and Haun teaches creating intermediate file collections that are children of said frozen version of said distributed application (Chang; [0022], line 6, “for generating new images”).

As per claim 36, the combination of Chang and Haun teaches modifying a first of said plurality of second file collections in response to a corresponding compute node's access to its instance of the application (Chang; [0027], line 7, In the lazy clone technique where only modifications are stored on “client images,” the image is modified by combining it with the common blocks from the base image.).

As per claim 37, the combination of Chang and Haun teaches copying a file from said root image to said first leaf image and then modifying said title in said first leaf image (Chang; [0023], line 12, When a “client image” is formed, it is implicit that its name will differ from the base image.).

As per claim 38, the combination of Chang and Haun teaches copying a data block from said root image to said first leaf image (Chang; [0024], lines 7, “copied block”) and then modifying said data block data block in said first leaf image (Chang; [0033], line 1, “writes or changes to image”).

Response to Arguments

4. Applicant's arguments with respect to the prior art rejections of claims 1-38 have been considered but are not persuasive.

As per claim 1, Applicant argues that the cited prior arts, Chang and Haun, do not in combination teach claim 1.

Specifically, Applicant argues that the shadow system volume in Haun is not a second file collection and is not used to instantiate an instance of a distributed application but is merely a copy of the changes made by a user. Examiner interprets the shadow system volume to be used in conjunction with core system volume to instantiate a user-specific version of the OS application. Examiner would suggest expanding on the specifics of the "distributed application" as this would likely overcome the current rejection.

Applicant further argues that Haun does not teach a root file system and teaches only a core system volume. The Examiner interprets the combination of the root file system in Chang along with the core system volume to teach the root file system described in the current form of the claims.

Accordingly, this rejection is maintained.

As per all remaining claims, the above arguments are echoed. Accordingly, these rejections are maintained.

Conclusion

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to PETER SHAW whose telephone number is (571)270-7179. The examiner can normally be reached on Monday - Friday 7:30 A.M. to 5:00 P.M.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, JOSEPH AVELLINO can be reached on 571-272-3905. 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.

/P. S./ March 9, 2010
Examiner, Art Unit 2458

/Joseph E. Avellino/
Supervisory Patent Examiner, Art Unit 2458

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11026622	NGUYEN ET AL.
	Examiner	**Art Unit**
	PETER SHAW	2458

SEARCHED

Class	Subclass	Date	Examiner
709	220, 222, 203, 208	3/31/2009	PS
713	2	3/31/2009	PS
717	174,175	3/31/2009	PS
709	220, 222, 203, 208	8/17/2009	PS
713	2	8/17/2009	PS
717	174,175	8/17/2009	PS
717	174,175	3/9/2010	PS
713	2	3/9/2010	PS
709	220, 222, 203, 208	3/9/2010	PS

SEARCH NOTES

Search Notes	Date	Examiner
EAST Database Search (hist. incl.)	3/31/2009	PS
Inventor Search (no relevant art found)	3/31/2009	PS
Interpreted Fig. 3 of drawings to comprise hardware	4/8/2009	PS
EAST Database Search (hist. incl.)	8/17/2009	PS
Inventor Search (no relevant art found)	8/17/2009	PS
EAST Database Search (hist. incl.)	3/9/2010	PS
Inventor Search (no relevant art found)	3/9/2010	PS

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11026622	NGUYEN ET AL.
	Examiner	**Art Unit**
	PETER SHAW	2458

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

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

CLAIM		DATE								
Final	Original	03/31/2009	08/17/2009	03/09/2010						
	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	✓	✓	✓						

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11026622	NGUYEN ET AL.
	Examiner	**Art Unit**
	PETER SHAW	2458

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

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

CLAIM		DATE								
Final	Original	03/31/2009	08/17/2009	03/09/2010						
	37	✓	✓	✓						
	38	✓	✓	✓						

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	22410	709/220.ccls. 709/203.ccls. 709/208.ccls. 709/222.ccls. 713/2.ccls. 717/174.ccls. 717/175.ccls.	US-PGPUB; USPAT	OR	OFF	2010/03/09 20:14
L2	1	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	OFF	2010/03/09 20:14
L3	30	(runtime OR dynamic).ti. AND boot.ti.	US-PGPUB; USPAT	OR	ON	2010/03/09 20:14
L4	218	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	ON	2010/03/09 20:14
L5	6	boot$.ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2010/03/09 20:14
S1	1	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	OFF	2009/03/31 14:02
S2	190	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:02
S3	548	"file system".ti. AND image	US-PGPUB; USPAT	OR	ON	2009/03/31 14:03
S4	4	"file system".ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:03
S5	120453	"operating system" AND image	US-PGPUB; USPAT	OR	ON	2009/03/31 14:07
S6	195	"distributed application".ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:19
S7	133	"distributed application".ti. AND "operating system"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:19

S8	15	"thin client" AND "operating system" AND (different OR version).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:25
S9	158308	"operating system" AND "thin client" (image OR configuration).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:27
S10	172	"operating system" AND "thin client" AND (image OR configuration).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:27
S11	1	"thin client".ti. AND (os OR "operating system").ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:50
S12	6533	(os OR "operating system").ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:51
S13	73	(os OR "operating system").ti. AND "thin client"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:51
S14	1	boot$.ti. AND "root image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:55
S15	5	boot$.ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:55
S16	2	"boot image".ti. AND (root OR custom OR various OR different OR base).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:57
S17	1	("6,421,777").PN.	US-PGPUB; USPAT	OR	OFF	2009/03/31 14:59
S18	0	"bootimage".ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:01
S19	0	"boot image".ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:01
S20	1	"boot image".ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 15:02
S21	31	boot$.ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 15:03

S22	19091	709/220.ccls. 709/203.ccls. 709/208.ccls. 709/222.ccls. 713/2.ccls. 717/174.ccls. 717/175.ccls.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:19
S23	820	S22 AND boot$.ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:19
S24	28	S22 AND ("distributed application" OR "boot image").ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:20
S25	0	(runtime OR dynamic).ti. AND "boot image".ti.	US-PGPUB; USPAT	OR	OFF	2009/04/01 13:25
S26	1	(runtime OR dynamic).ti. AND "boot image".ti.	US-PGPUB; USPAT	OR	ON	2009/04/01 13:25
S27	30	(runtime OR dynamic).ti. AND boot.ti.	US-PGPUB; USPAT	OR	ON	2009/04/01 13:26
S28	20293	709/220.ccls. 709/203.ccls. 709/208.ccls. 709/222.ccls. 713/2.ccls. 717/174.ccls. 717/175.ccls.	US-PGPUB; USPAT	OR	OFF	2009/08/17 14:31
S29	31	S28 AND ("distributed application" OR "boot image").ti.	US-PGPUB; USPAT	OR	OFF	2009/08/17 14:31
S30	199	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	ON	2009/08/17 14:32
S31	30	(runtime OR dynamic).ti. AND boot.ti.	US-PGPUB; USPAT	OR	ON	2009/08/17 14:32
S32	1	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	OFF	2009/08/17 14:32

EAST Search History (Interference)

<This search history is empty>

3/9/2010 8:15:24 PM

H:\ mybackup\ mydocuments\ EAST\ Workspaces\ 11026622.wsp

PTO/SB/08a (01-10)
Approved for use through 07/31/2012. 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.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	11026622
Filing Date	2004-12-30
First Named Inventor	NGUYEN
Art Unit	2458
Examiner Name	
Attorney Docket Number	PNTA-P002

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	7269664		2007-09-11	Hutsch, et al.	
	2	5175852		1992-12-29	Johnson, et al.	
	3	6502238		2002-12-31	Pavan, et al.	
	4	7475274		2009-01-09	Davidson	
	5	6101576		2000-08-08	Kobayashi, et al.	

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20050131962	A1	2005-06-16	Deshpande	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number	11026622
	Filing Date	2004-12-30
	First Named Inventor	NGUYEN
	Art Unit	2458
	Examiner Name	
	Attorney Docket Number	PNTA-P002

	2	20020083120	A1	2002-06-27	Soltis	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] ¡	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1		☐

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number	11026622
	Filing Date	2004-12-30
	First Named Inventor	NGUYEN
	Art Unit	2458
	Examiner Name	
	Attorney Docket Number	PNTA-P002

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☒ Fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☐ None

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Jeffrey A. Lehman/	Date (YYYY-MM-DD)	2010-04-22
Name/Print	Jeffrey A. Lehman	Registration Number	65494

This collection of information is required by 37 CFR 1.97 and 1.98. 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 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: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

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 the Freedom of Information Act requires disclosure of these record s.

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 inspections 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.

Electronic Patent Application Fee Transmittal

Application Number:	11026622
Filing Date:	30-Dec-2004
Title of Invention:	Branching store file system
First Named Inventor/Applicant Name:	Tung M. Nguyen
Filer:	Jeffrey A. Lehman/Shannon Warren
Attorney Docket Number:	PNTA-P002

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Submission- Information Disclosure Stmt	1806	1	180	180
		Total in USD ($)		**180**

Electronic Acknowledgement Receipt

EFS ID:	7466962
Application Number:	11026622
International Application Number:	
Confirmation Number:	8329
Title of Invention:	Branching store file system
First Named Inventor/Applicant Name:	Tung M. Nguyen
Customer Number:	70848
Filer:	Jeffrey A. Lehman/Shannon Warren
Filer Authorized By:	Jeffrey A. Lehman
Attorney Docket Number:	PNTA-P002
Receipt Date:	22-APR-2010
Filing Date:	30-DEC-2004
Time Stamp:	14:26:31
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Electronic Funds Transfer
Payment was successfully received in RAM	$180
RAM confirmation Number	597
Deposit Account	
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if app.)

1	Information Disclosure Statement (IDS) Filed (SB/08)	PNTA-P002-IDS.pdf	878657 39c06d92ca908fc8b02130fa916e2d388ca4539f	no	4
Warnings:					
Information:					
2	Fee Worksheet (PTO-875)	fee-info.pdf	29860 6e0dc76474da9264b6ad8c427584d8a025171198	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	908517		

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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: Nguyen, et al.	Confirmation	: 8329
Serial No.	: 11/026,622	Examiner	: Shaw, Peter C.
Filing Date	: 12/30/2004	Art Unit	: 2458

Attorney Docket No. : PNTA-P002

For: BRANCHING STORE FILE SYSTEM

AMENDMENT AND RESPONSE TO OFFICE ACTION

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

Dear Sir:

In response to the Office Action mailed March 15, 2010, please consider the following amendments and remarks in the above captioned patent application.

I hereby certify that this correspondence is being electronically transmitted to the USPTO via EFS on the date shown below:
June 15, 2010
/Shannon Warren/
Shannon Warren

Amendments to the Claims:

1. (Previously presented) A branching store file system, comprising:

a root file system comprising a ~~single~~ first file collection of a distributed application, wherein said first file collection is read-only accessible by a plurality of compute nodes, and wherein said ~~single~~ first file collection is a complete, unmodified copy of the distributed application ~~complete, unmodified, read-only application shared by the plurality of compute nodes~~;

a plurality of second file collections separate from the root file system, each comprising only modified portions of said distributed application for respective ones of said plurality of compute nodes, wherein an instance of said distributed application for a first ~~of said plurality of~~ compute node[[s]] is instantiated by accessing the unmodified portions of the application for the first computer node from the first file collection and the modified portions of the application for the first compute node from at least one of the plurality of second file collections~~, wherein the modified portions of the application for a particular compute node are accessed from the at least one second file collection and the unmodified portions of the application for a particular compute node are accessed from the first file collection, such that each compute node appears to have access to their own version of the application~~, and wherein the first file collection and the plurality of second file collections are on separate nodes; and

a filter that controls each instantiation ~~access of instances~~ of said distributed application for said compute nodes, wherein the filter selects at least one of the plurality of second file collections and selects portions of the first file collection to instantiate an instance of the distributed application for the first ~~a particular~~ compute node, and wherein the filter adds to the ~~particular~~ second file collection of the first compute node when the first compute node makes changes to its instantiation of the distributed application.

2. (Previously presented) The system of Claim 1, wherein said plurality of second file collections comprise branch modification logs for said respective ones of said compute nodes.

3. (Previously presented) The system of Claim 1, wherein said plurality of second file collections comprise change volumes for said respective ones of said compute nodes.

4. (Original) The system of Claim 1, wherein said filter is implemented at said compute nodes.

5. (Original) The system of Claim 1, wherein said filter is implemented at said root file system.

6. (Previously presented) The system of Claim 1, further comprising intermediate file collections between said first file collection and said plurality of second file collections, wherein said intermediate file collections comprise application service groups.

7. (Previously presented) The system of Claim 6, wherein access of said compute nodes to said first file collection and intermediate file collections is read-only.

8. (Original) The system of Claim 1, wherein said distributed application comprises an operating system.

9. (Previously presented) The system of Claim 1, wherein said first file collection is concurrently accessible to a plurality of said compute nodes.

10. (Previously presented) The system of Claim 1, wherein said plurality of second file collections are configured to be modified on a block-by-block basis.

11. (Previously presented) The system of Claim 1, wherein said plurality of second file collections are configured to be modified on a file-by-file basis.

12. (Currently amended) A branching store file system, comprising:

a root file system having a root metadata structure and a ~~single~~ root data structure, ~~said~~ entries in said root metadata structure pointing to ~~associated unmodified~~ data in said

root data structure, wherein the root data structure is an unmodified, read-only application shared by a plurality of compute nodes; and

a plurality of branch modification logs for respective ones of the plurality of compute nodes, each comprising:

a branch data structure comprising only modified portions of the application for a respective compute node ~~data reflecting changes made to an instance of said root data structure by said branch modification log's respective compute node~~; and

a branch metadata structure comprising unmodified and modified entries, said unmodified entries pointing to ~~associated~~ portions of said unmodified data in said root data structure, and said modified entries pointing to ~~associated~~ portions of said modified data in said branch data structure, wherein an instance of a distributed application for a first ~~of said~~ compute node[[s]] is instantiated by accessing said modified portions of said application for said first compute node from the branch data structure and said unmodified portions of the application for said first compute node from said root data structure, said unmodified and modified portions of said application instantiating an instance of said application for said first compute node ~~branch metadata structure~~, wherein said root data structure and said branch data structure are on separate nodes, ~~wherein the modified portions of the application for a particular compute node are accessed from the branch data structure and the unmodified portions of the application for a particular compute node are accessed from the root data structure, such that portions of modified data and portions of unmodified data are collected to instantiate an instance of the distributed application for a particular compute node,~~ and wherein the branch data structure is added to when the first compute node makes changes to its instantiation of the distributed application~~, such that each compute node appears to have access to their own version of the application~~.

13. (Original) The system of Claim 12, further comprising a filter that directs accesses of said distributed application from compute nodes to a respective compute node's instantiation of the distributed application.

14. (Original) The system of Claim 12, wherein said distributed application comprises an operating system.

15. (Original) The system of Claim 13, wherein said filter is configured to traverse inodes to said root data structure for accesses to said un-modified data.

16. (Original) The system of Claim 13, wherein said filter is configured to traverse inodes to said branch data structure for accesses to said modified data.

17. (Original) The system of Claim 12, wherein access to said root data structure by said compute nodes is read-only and access to said branch data structure is read-write-modify by said branch modification log's respective compute node.

18. (Original) The system of Claim 12, wherein ones of said modified entries have a modification attribute attached.

19. (Original) The system of Claim 12, wherein said branch modification log is configured to receive updates to said modified data on a block-by-block basis.

20. (Currently amended) A branching store file system, comprising:

a root file system comprising a ~~single~~ base instance of a distributed application, wherein the ~~single~~ base instance is an unmodified, read-only application shared by a ~~the~~ plurality of compute nodes;

change volumes for respective ones of [[a]] the plurality of compute nodes, ~~and~~ each comprising only modified portions ~~versions of files~~ of said distributed application; and

a filter program ~~code that~~ controlling each instantiation of the distributed application for respective ones of the plurality of compute nodes ~~directs accesses of said compute nodes' respective instances of said distributed application~~, wherein an instance of the distributed application for a first compute node is instantiated by the filter program ~~said compute nodes' respective instances of said distributed application are instantiated~~

by accessing unmodified portions of the distributed application for the first compute node from said base instance and modified portions of the distributed application for the first compute node from at least one change volume ~~respective ones of said change volumes~~, wherein said base instance and said at least one ~~respective ones of said~~ change volume[[s]] are on separate nodes, ~~wherein the modified portions of the application for a particular compute node are accessed from the respective ones of said change volumes and the unmodified portions of the application for a particular compute node are accessed from the base instance, such that each compute node appears to have access to their own version of the application,~~ and wherein the filter program adds to the at least one change volume of the first compute node when the first compute node ~~the respective ones of said change volumes when the compute node~~ makes changes to its instantiation of the distributed application.

21. (Original) The system of Claim 20, further comprising a table of locations indicating locations of said change volumes.

22. (Original) The system of Claim 20, wherein said base instance is concurrently accessible to a plurality of said compute nodes.

23. (Original) The system of Claim 20, wherein said compute nodes have read-only access to files in said base instance.

24. (Original) The system of Claim 21, wherein said distributed application comprises an operating system.

25. (Original) The system of Claim 20, wherein said filter performs a copy-on-write to one of the change volumes when a compute node writes to said compute node's instantiation of said distributed application.

26. (Original) The system of Claim 20, wherein said change volumes are modified on a file-by-file basis.

27. (Currently amended) A computer-implemented method of distributed application deployment, said method comprising:

creating a root file system comprising a ~~single~~ first file collection of an application, read-only accessible by a plurality of compute nodes;

creating a plurality of second file collections that are configured to only record changes to said first file collection of the application for said respective compute nodes;

creating a directory configured to instantiate ~~access~~ instances of said application for said respective compute nodes, wherein an instance of said application for a first compute node ~~a respective one of the compute node's instance of said application~~ is instantiated by accessing unmodified portions of said application for said first compute node from ~~at least a portion of~~ said first file collection and modified portions of said application for said first compute node from said respective ~~the respective compute node's~~ second file collection, ~~wherein modified portions of the application for a particular compute node are accessed from the respective second file collection and the unmodified portions of the application for a particular compute node are accessed from the first file collection, such that each compute node appears to have access to their own version of the application~~, wherein the first file collection and the plurality of second file collections are on separate nodes; and

providing said compute nodes with said respective instances of said application.

28. (Previously presented) The method of Claim 27, wherein said second file collections comprise branch modification logs.

29. (Original) The method of Claim 28, wherein said providing said compute nodes with respective instances of said application comprises traversing modes as indicated by said branch modification logs.

30. (Previously presented) The method of Claim 27, wherein said second file collections comprise change volumes.

31. (Original) The method of Claim 30, wherein said providing said compute nodes with respective instances of said application comprises consulting a table of locations.

32. (Original) The method of Claim 27, wherein said application comprises an operating system.

33. (Previously presented) The method of Claim 27, wherein access to said first file collection by said compute nodes is read-only.

34. (Original) The method of Claim 27, further comprising freezing a version of said distributed application.

35. (Previously presented) The method of Claim 34, further comprising creating intermediate file collections that are children of said frozen version of said distributed application.

36. (Previously presented) The method of Claim 27, further comprising modifying a respective one of said second file collections in response to a corresponding compute node's access to its instance of the application.

37. (Previously presented) The method of Claim 36, wherein said modifying comprises a copying a file from said first file collection to said one of said second file collections and then modifying said file in said one of said second file collections.

38. (Previously presented) The method of Claim 36, wherein said modifying comprises a copying a data block from said first file collection to said one of said second file collections and then modifying said data block in said one of said second file collections.

REMARKS

Claims 1-38 are pending. Claims 1, 12, 20, and 27 are amended. No new matter has been added as a result of these amendments.

Claim Rejections – 35 U.S.C. §103

Claims 1-38 are rejected under 35 U.S.C. 103(a) as being allegedly unpatentable over Chang (US 2003/0126242), hereinafter "Chang," in view of Haun, et al. (US 6,751,658), hereinafter "Huan." Applicants respectfully assert that Claims 1-38 are patentable over the cited combination in view of the following.

Applicants respectfully point out that the Examiner has the burden of establishing a prima facie case of obviousness. To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim features. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on Applicant's disclosure. In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). See MPEP 2100-126. Specifically, "all words in a claim must be considered when judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d., 1382 (CCPA 1970).

Moreover, in response to the recent U.S. Supreme Court decision in *KSR Int'l Co. v. Teleflex, Inc.* (U.S. 2007), new guidelines were set forth for examining obviousness under 35 U.S.C. 103. The U.S. Supreme Court reaffirmed the *Graham* factors and, while not totally rejecting the "teachings, suggestion, or motivation" test, the Court appears to now require higher scrutiny on the part of the U.S. Patent & Trademark Office. In accordance with the recently submitted guidelines, it is "now necessary to identify the reason" why a person of ordinary skill in the art would have combined the elements of cited references, or at least describe the pertinence of the elements set forth in the cited disclosure, in the manner presently claimed.

Applicants respectfully assert the combination of Chang and Haun fails to teach or suggest all the claimed elements of Claims 1-38 in view of the following rationale.

<u>Claim 1:</u>

Amended Claim 1 recites in part:

> a root file system comprising a first file collection of a distributed application, wherein said first file collection is read-only accessible by a plurality of compute nodes, and wherein said first file collection is a complete, unmodified copy of the distributed application;
>
> <u>a plurality of second file collections separate from the root file system, each comprising only modified portions of said distributed application for respective ones of said plurality of compute nodes</u>, wherein an instance of said distributed application for a first compute node is instantiated by accessing the <u>unmodified portions</u> of the application for the first computer node <u>from the first file collection</u> and the <u>modified portions</u> of the application for the first compute node <u>from at least one of the plurality of second file collections</u>, and wherein the first file collection and the plurality of second file collections are on separate nodes; and
>
> a filter that controls each instantiation of said distributed application for said compute nodes, <u>wherein the filter selects at least one of the plurality of second file collections and selects portions of the first</u>

> file collection to instantiate an instance of the distributed application for the first compute node, and wherein the filter adds to the second file collection of the first compute node when the first compute node makes changes to its instantiation of the distributed application (emphasis added).

Applicants respectfully assert that Chang and Haun fail to teach or suggest "a plurality of second file collections separate from the root file system, each comprising only modified portions of said distributed application for respective ones of said plurality of compute nodes, wherein an instance of said distributed application for a first compute node is instantiated by accessing the unmodified portions of the application for the first compute node from the first file collection and the modified portions of the application for the first compute node from at least one of the plurality of second file collections…and a filter that controls each instantiation of said distributed application [by] select[ing] at least one of the plurality of second file collections and select[ing] portions of the first file collection to instantiate an instance of the distributed application for the first compute node," as claimed in Claim 1.

As understood by Applicants, the rejection is reading the client specific blocks of Chang (Paragraph 26) on Applicants' claimed "plurality of second file collections separate from the root file system, each comprising only modified portions of said distributed application for respective ones of said plurality of compute nodes," as claimed in Claim 1. Applicants respectfully disagree. As understood by Applicants, Chang discloses a pooled storage comprising a base boot image and a plurality of client image copies created by a snapshot manager (Figure 1; Paragraphs 0022-0025). In particular, Chang discloses that each client image comprises both unmodified common OS and

application blocks, as well as modified, client-specific blocks that can be added as needed to the particular client image copies (Figure 1; Paragraphs 0022-0025). To the extent that Chang discloses modified, client-specific blocks in a client image copy, Chang does not teach or suggest a "plurality of second file collections separate from the root file system, each comprising only modified portions of said distributed application for respective ones of said plurality of compute nodes," as claimed in Claim 1. While Applicants claim a second file collection comprising only modified portions of a distributed application for respective compute nodes, as claimed in Claim 1, Chang discloses a client image copy containing both unmodified and modified portions of the application.

Further, as understood by Applicants, the rejection agrees that Chang does not teach or suggest "an instance of said distributed application for a first compute node is instantiated by accessing the unmodified portions of the application for the first compute node from the first file collection and the modified portions of the application for the first compute node from at least one of the plurality of second file collections…and a filter that controls each instantiation of said distributed application [by] select[ing] at least one of the plurality of second file collections and select[ing] portions of the first file collection to instantiate an instance of the distributed application for the first compute node," as claimed in Claim 1. The rejection contends that Haun cures these deficiencies. Applicants respectfully disagree.

As understood by Applicants, Haun is directed towards providing an operating system in a net-booted environment, wherein the network computer client boots from a

boot image provided by a server (Abstract). Haun further discloses that in response to attempts by the client to modify the operating system files, the changes are recorded on the server separate from the system volumes storing the operating system files and into a storage area associated with the client (Abstract). As understood by Applicants, the client has access to a shared split OS that includes read-only files in a core system volume as well as a user system volume that contains user-configurable system components, including preferences and all system additions installed by application software (Figure 9; Column 14, lines 1-25). Haun further discloses the creation of a shadow system volume for each connected client (Figure 9; Column 14, lines 26-58). As understood by Applicants, the shadow system volume "shadows" the user system volume by storing modifications that are made to the user system volume, such that whenever a changes are made to the user system volume, they are copied to the shadow system volume (Figure 9; Column 14, lines 26-58). Haun further notes that data may only be written to the shadow system volume (Column 14, lines 45-52).

To the extent that Haun discloses a read-only volume and a read/write volume that is also copied to a shadow system volume, Haun does not teach or suggest "an instance of said distributed application for a first compute node is instantiated by accessing the unmodified portions of the application for the first compute node from the first file collection and the modified portions of the application for the first compute node from at least one of the plurality of second file collections…and a filter that controls each instantiation of said distributed application [by] select[ing] at least one of the plurality of second file collections and select[ing] portions of the first file collection to instantiate an

instance of the distributed application for the first compute node," as claimed in Claim 1. Applicants respectfully assert that the shadow system volume of Haun is not one of a plurality of second file collections, as claimed in Claim 1. The shadow system volume of Haun is not used to instantiate an instance of a distributed application for a compute node, as claimed in Claim 1; rather it is a copy of the changes made to the user system volume and not used for creating an instance. Applicants respectfully assert that Haun is silent with regards to using the shadow volume to instantiate an instance of an application, as claimed in Claim 1.

Further, Applicants respectfully assert there is no motivation to combine Chang with Haun to realize Applicants' claimed root file system comprising a first file collection of a distributed application read-only accessible by a plurality of compute nodes, the first file collection a complete, unmodified copy of the distributed application, and a plurality of second file collections, wherein an instantiation of the distributed application for a first compute node is accomplished by a filter selecting at least one of the plurality of second file collections and selecting portions of the first file collection to instantiate an instance of the distributed application for the first compute node, as claimed in Claim 1. Rather than a root file system that contains an unmodified copy of the distributed application, Haun discloses that the core system volume preferably contains those parts of the system that do not need to be written back to during system operation, thereby providing the system components that are mandatory for system operation (Column 14, lines 8-12). Haun discloses that these "essential components" are separated from the user system volume and protected as read-only, while the user system volume contains all the user-

configurable system components, including preferences and all the system additions installed by application software (Column 14, lines 12-32). Haun further notes that the shadow system volume is a copy of the user system volume. Further, Chang discloses a plurality of client image copies that contain both unmodified and modified portions of the application, rather than comprising only modified portions of said distributed application for respective ones of said plurality of compute nodes, as claimed in Claim 1.

Therefore, Applicants respectfully assert that Claim 1 is not rendered obvious by the combination of Chang and Haun. Accordingly, Applicants respectfully assert that dependent Claims 2-11 are patentable by virtue of their dependency on patentable independent Claim 1, as well as for their additional recited patentable features.

Claim 12:

Independent Claim 12 recites features similar to that of independent Claim 1, and is therefore patentable for at least the same or similar reasons as recited above. Accordingly, Applicants respectfully assert that dependent Claims 13-19 are patentable by virtue of their dependency on an allowable base claim, as well as for their additional recited patentable features.

Claim 20:

Independent Claim 20 recites features similar to that of independent Claim 1, and is therefore patentable for at least the same or similar reasons as recited above. Accordingly, Applicants respectfully assert that dependent Claims 21-26 are patentable

by virtue of their dependency on an allowable base claim, as well as for their additional recited patentable features.

Claim 27:

Independent Claim 27 recites features similar to that of independent Claims 1, 12, and 20, and is therefore patentable for at least the same or similar reasons as recited above. Accordingly, Applicants respectfully assert that dependent Claims 28-38 are patentable by virtue of their dependency on an allowable base claim, as well as for their additional recited patentable features.

For the above reasons, Applicants request reconsideration and withdrawal of the rejections under 35 U.S.C. §103.

CONCLUSION

In light of the above listed remarks, reconsideration of rejected Claims is requested. Based on the arguments presented above, it is respectfully submitted that Claims 1-38 overcome the rejections of record and, therefore, allowance of Claims 1-38 is earnestly solicited.

Please charge any additional fees that may be required to maintain pendency of the present application, or apply any credits to our PTO deposit account number: 50-4160.

Respectfully submitted,

MURABITO, HAO & BARNES LLP

Dated: June 15, 2010

/Jeffrey A. Lehman/

Jeffrey A. Lehman
Registration No. 65,494

MURABITO, HAO & BARNES LLP
Two North Market Street
Third Floor
San Jose, California 95113

(408) 938-9060 Voice
(408) 938-9069 Facsimile

Electronic Acknowledgement Receipt

EFS ID:	7819323
Application Number:	11026622
International Application Number:	
Confirmation Number:	8329
Title of Invention:	Branching store file system
First Named Inventor/Applicant Name:	Tung M. Nguyen
Customer Number:	70848
Filer:	Jeffrey A. Lehman/Shannon Warren
Filer Authorized By:	Jeffrey A. Lehman
Attorney Docket Number:	PNTA-P002
Receipt Date:	15-JUN-2010
Filing Date:	30-DEC-2004
Time Stamp:	17:07:46
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		PNTA-P002-Response.pdf	126911 6fcf089f39b6078a9fbab52df6d11e9cfaecad15	yes	17

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Amendment/Req. Reconsideration-After Non-Final Reject	1	1
	Claims	2	8
	Applicant Arguments/Remarks Made in an Amendment	9	17

Warnings:	
Information:	
Total Files Size (in bytes):	126911

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.

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. 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	Filing Date	☐ To be Mailed
11/026,622	12/30/2004	

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY ☐ RATE ($)	SMALL ENTITY FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	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).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT 06/15/2010	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 38	Minus	** 38	= 0	X $ =		OR	X $52=	0
Independent (37 CFR 1.16(h))	* 4	Minus	***4	= 0	X $ =		OR	X $220=	0
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

AMENDMENT	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	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.

Legal Instrument Examiner:
/LASHAWN MARKS/

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.
11/026,622	12/30/2004	Tung M. Nguyen	PNTA-P002	8329

70848 7590 08/13/2010
PANTA c/o MURABITO HAO & BARNES LLP
Two Market Street
3rd Floor
San Jose, CA 95113

EXAMINER
SHAW, PETER C

ART UNIT	PAPER NUMBER
2458	

MAIL DATE	DELIVERY MODE
08/13/2010	PAPER

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

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	11/026,622	NGUYEN ET AL.	
	Examiner	Art Unit	
	PETER SHAW	2458	

-- 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 *15 June 2010*.

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) *1-38* 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) *1-38* 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 *30 December 2004* 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/SB/08) Paper No(s)/Mail Date *4/22/2010*.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-38 are pending in this action.

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.

2. Claims 1-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chang (US PGPUB No. 2003/0126242) in view of Haun et al. (US Patent No. 6,751,658) [hereinafter "Haun"].

As per claim 1, Chang teaches a branching store file system, comprising:

> a root file system comprising a first file collection of a distributed application ([0011], line 14, "base image"), wherein said first file collection is read-only accessible by a plurality compute nodes ([0026], line 2, "base image never written by the clients") ([0022], line 15, first file collection is a building block for boot images, changes are made to copies to the base image);

a plurality of second file collections separate from the root file system, each comprising only modified portions of said distributed application for respective ones of said plurality of compute nodes ([0026], line 3, “client specific blocks”);

a filter that controls each instantiation of said distributed application for said compute nodes ([0017], lines 7-11, “external storage controller…controlling access to…boot images”).

Chang does not explicitly teach an instance of said distributed application for a first compute node is instantiated by accessing the unmodified portions of the application for the first computer node from the first file collection and the modified portions of the application for the first compute node from at least one of said plurality of second file collections, wherein the first file collection and the plurality of second file collections are on separate nodes; wherein said first file collection is a complete, unmodified, read-only copy of the distributed application; and wherein the filter selects at least one of the plurality of second file collections and selects portions of the first file collection to instantiate an instance of the distributed application for the first compute node, and wherein the filter adds to the second file collection of the first compute node when the compute node makes changes to its instantiation of the distributed application.

Haun teaches an instance of said distributed application for a first compute node is instantiated by accessing the unmodified portions of the application for the first computer node from the first file collection (Col. 14, lines 10-11, core system volume)

and the modified portions of the application for the first compute node from at least one of said plurality of second file collections (Col. 14, line 28-29, user shadow volume), wherein the first file collection and the plurality of second file collections are on separate nodes (Col. 6, lines 45-47, volumes, i.e. file collections, application volumes and client data, i.e. second file collections, can be distributed across multiple server systems, i.e. on separate nodes); wherein said first file collection is a complete, unmodified, read-only copy of the distributed application (Col. 14, lines 8-10, core system volume contains all unmodified data); and wherein the filter selects at least one of the plurality of second file collections (Col. 14, lines 18-25, user system volume contains all of the user preferences) and selects portions of the first file collection to instantiate an instance of the distributed application for the first compute node (Col. 14, line 56-58, system volume with user volume, i.e. user preferences), and wherein the filter adds to the second file collection of the first compute node when the compute node makes changes to its instantiation of the distributed application (Col. 14, lines 40-44, making changes to the user volume/shadow volume based on changes made by client).

At the time of invention, it would have been obvious to one of ordinary skill in the art to combine Chang, with the teachings of Haun, an instance of said distributed application for a first compute node is instantiated by accessing the unmodified portions of the application for the first computer node from the first file collection and the modified portions of the application for the first compute node from at least one of said plurality of second file collections, wherein the first file collection and the plurality of second file collections are on separate nodes; wherein said first file collection is a

complete, unmodified, read-only copy of the distributed application; and wherein the filter selects at least one of the plurality of second file collections and selects portions of the first file collection to instantiate an instance of the distributed application for the first compute node, and wherein the filter adds to the second file collection of the first compute node when the compute node makes changes to its instantiation of the distributed application, to allow the distributed application to be modifiable as it is distributed away from the root file system.

As per claim 2, the combination of Chang and Haun teaches said plurality of second file collections comprising branch modification logs for said respective ones of said compute nodes (Chang; [0027], client image copies containing just the modifications to the base image, i.e. client-specific writes) (Chang; [0024], line 8, new block modifications logged and included in new image).

As per claim 3, the combination of Chang and Haun teaches said plurality of second file collections comprising change volumes for said respective ones of said compute nodes (Chang; [0025], lines 14-15, new blocks written to original volume to form new volumes).

As per claim 4, the combination of Chang and Haun teaches said filter implemented at said compute nodes (Chang; [0018], lines 1-3, “in one embodiment, the clients are configured...to provide similar or identical functions").

As per claim 5, the combination of Chang and Haun teaches said filter implemented at said root file system (Chang; [0017], line 1, "external storage controller residing in the network boot system").

As per claim 6, the combination of Chang and Haun teaches intermediate images between said first file collection and said plurality of second file collections (Chang; [0023], line 12, “client group images”), wherein said intermediate images comprise application service groups (Chang; [0023], line 12, “client group images”).

As per claim 7, the combination of Chang and Haun teaches access of said compute nodes to said intermediate images being read-only (Chang; [0026], line 2, “base image never written by the clients”).

As per claim 8, the combination of Chang and Haun teaches said distributed application comprising an operating system (Chang; [0016], line 10, “operating system”).

As per claim 9, the combination of Chang and Haun teaches said first file collection concurrently accessible to a plurality of said compute nodes (Chang; [0017], line 3, The boot system provides access to boot images for a “plurality of clients.”).

As per claim 10, the combination of Chang and Haun teaches said plurality of second file collections configured to be modified on a block-by-block basis (Chang; [0024], line 5, “data blocks”).

As per claim 11, the combination of Chang and Haun teaches said plurality of second file collections configured to be modified on a file-by-file basis (Chang; [0024], line 5, “files”).

As per claim 12, Chang teaches a branching store file system, comprising:

> a root file system having a root metadata structure and a root data structure, said in said root metadata structure pointing data in said root data structure (Chang; [0008], line 12, “Cloning…provide two collections of hard links to the same files”); and
>
> a plurality of branch modification log for respective ones of the plurality of compute nodes, comprising:
>
> > a branch data structure comprising only modified portions of the application for a respective compute node (Chang; [0027], line 10, The lazy clone technique does not contain an image, only provides instruction

as to how to modify the base image to properly produce the client-specific image, i.e. metadata.); and

a branch metadata structure comprising unmodified and modified entries, said unmodified entries pointing to portions of said unmodified data in said root data structure (Chang; [0008], line 12, "Cloning…provide two collections of hard links to the same files"), and said modified entries pointing to portions of said modified data in said branch data structure (Chang; [0027], line 10, The lazy clone technique does not contain an image, only provides instruction as to how to modify the first file collection to properly produce the client-specific image, i.e. metadata).

Chang does not explicitly teach an instance of said distributed application for a first compute nodes is instantiated by accessing said modified portions of said application for said first compute node from the branch data structure and said unmodified portions of the application for said first compute node from said root data structure, said unmodified and modified portions of said application instantiating an instance of said application for said first compute node, wherein said root data structure and said branch data structure are on separate nodes; wherein the root data structure is an unmodified, read-only application shared by a plurality of compute nodes, and wherein the branch data structure is added to when the first compute node makes changes to its instantiation of the distributed application. Haun teaches an instance of

said distributed application for a first compute nodes is instantiated by accessing said modified portions of said application for said first compute node from the branch data structure (Col. 14, lines 18-25, user system volume contains all of the user preferences) and said unmodified portions of the application for said first compute node from said root data structure (Col. 14, lines 8-10, core system volume contains all unmodified data), said unmodified and modified portions of said application instantiating an instance of said application for said first compute node (Col. 14, line 28-29, user shadow volume) (Col. 14, lines 40-44, making changes to the user volume/shadow volume based on changes made by client), wherein said root data structure and said branch data structure are on separate nodes (Col. 6, lines 45-47, volumes, i.e. file collections, application volumes and client data, i.e. second file collections, can be distributed across multiple server systems, i.e. on separate nodes); wherein the root data structure is an unmodified, read-only application shared by a plurality of compute nodes (Col. 14, line 14, core system volume contains read-only data), and wherein the branch data structure is added to when the first compute node makes changes to its instantiation of the distributed application (Col. 14, lines 40-44, making changes to the user volume/shadow volume based on changes made by client).

At the time of invention, it would have been obvious to one of ordinary skill in the art to combine Chang, with the teachings of Haunan instance of said distributed application for a first compute nodes is instantiated by accessing said modified portions of said application for said first compute node from the branch data structure and said unmodified portions of the application for said first compute node from said root data

structure, said unmodified and modified portions of said application instantiating an instance of said application for said first compute node, wherein said root data structure and said branch data structure are on separate nodes; wherein the root data structure is an unmodified, read-only application shared by a plurality of compute nodes, and wherein the branch data structure is added to when the first compute node makes changes to its instantiation of the distributed application, to allow the distributed application to be modifiable as it is distributed away from the root file system.

As per claim 13, the combination of Chang and Haun teaches a filter that directs accesses of said distributed application from compute nodes (Chang; [0017], lines 7-11, “external storage controller…controlling access to…boot images”) to a respective compute node's instantiation of the distributed application ([0030], lines 4-5, “access to client-specific image copy”).

As per claim 14, the substance of the claim language is identical to that of claim 8. Accordingly, this claim is rejected under the same rationale.

As per claim 15, the combination of Chang and Haun teaches said filter configured to traverse nodes to said root data structure for accesses to said unmodified data (Chang; [0033], lines 17-18, unmodified common data stored at separate location from updated client-specific blocks, Traversal to common data is inherent given the successful production of a copy image.).

As per claim 16, the combination of Chang and Haun teaches said filter configured to traverse nodes to said branch data structure for accesses to said modified data (Chang; [0033], lines 17-18, unmodified common data stored at separate location from updated client-specific blocks, Traversal to common data is inherent given the successful production of a copy image.) (Chang; [0023], line 5 and 12, Traversal to a varying first file collection or a client group image would be similarly inherent.).

As per claim 17, the combination of Chang and Haun teaches access to said root data structure by said compute nodes being read-only (Chang; [0026], line 2, “base image never written by the clients”) (Chang; [0022], line 15, base image is a building block for boot images, changes are made to copies to the base image) and access to said branch data structure being read-write-modify by said branch modification log's respective compute node (Chang; [0030], line 13, Given that the client boots from the image, it is inherent that it reads it.) (Chang; [0033], line 1, writes or changes received from clients).

As per claim 18, the combination of Chang and Haun teaches ones of said modified entries having a modification attribute attached (Chang; [0027], line 7, Lazy clone technique logs changes to the first file collection instead of writing an entire boot image. It is inherent that each of these entries would include both a pointer and what is being changed, i.e. modification attribute.).

As per claim 19, the combination of Chang and Haun teaches said branch modification log configured to receive updates to said modified data on a block-by-block basis (Chang; [0024], line 5, “data blocks”).

As per claim 20, Chang teaches a branching store file system, comprising:

a root file system comprising a base instance of a distributed application ([0011], line 14, “base image”);

change volumes for respective ones of the plurality of compute nodes and each comprising only modified portions of said distributed application ([0025], lines 14-15, new blocks written to original volume to form new volumes); and

a filter program controlling each instantiation of the distributed application for respective ones of the plurality of compute nodes ([0017], lines 7-11, “external storage controller…controlling access to…boot images”), wherein an instance of the distributed application for a first compute node is instantiated by the filter program ([0011], line 14, “base image”) (volumes ([0022], line 16, “client-specific image").

Chang does not explicitly teach an instance of said distributed application for a first of said compute nodes is instantiated by accessing unmodified portions of the distributed application for the first compute node from said base instance and modified portions of the distributed application for the first compute node from at least one change volume, wherein the base instance and said change volumes are on separate nodes; wherein the single base instance is an unmodified, read-only application shared by the plurality of compute nodes; and wherein the filter program adds to the the at least one change volume of the first compute node when the first compute node makes changes to its instantiation of the distributed application. Haun teaches an instance of said distributed application for a first of said compute nodes is instantiated by accessing unmodified portions of the distributed application for the first compute node from said base instance (Col. 14, lines 10-11, core system volume) and modified portions of the distributed application for the first compute node from at least one change volume (Col. 14, line 28-29, user shadow volume), wherein the base instance and said change volumes are on separate nodes (Col. 6, lines 45-47, volumes, i.e. file collections, application volumes and client data, i.e. second file collections, can be distributed across multiple server systems, i.e. on separate nodes); wherein the single base instance is an unmodified, read-only application shared by the plurality of compute nodes nodes (Col. 14, line 14, core system volume contains read-only data); and wherein the filter program adds to the at least one change volume of the first compute node when the first compute node makes changes to its instantiation of the distributed

application (Col. 14, lines 40-44, making changes to the user volume/shadow volume based on changes made by client).

At the time of invention, it would have been obvious to one of ordinary skill in the art to combine Chang, with the teachings of Haun, an instance of said distributed application for a first of said compute nodes is instantiated by accessing unmodified portions of the distributed application for the first compute node from said base instance and modified portions of the distributed application for the first compute node from at least one change volume, wherein the base instance and said change volumes are on separate nodes; wherein the single base instance is an unmodified, read-only application shared by the plurality of compute nodes; and wherein the filter program adds to the the at least one change volume of the first compute node when the first compute node makes changes to its instantiation of the distributed application, to allow the distributed application to be modifiable as it is distributed away from the root file system.

As per claim 21, the combination of Chang and Haun teaches a table of locations indicating locations of said change volumes (Chang; [0035], line 9, discovery of the pooled storage device, id of the specific-client image, i.e. change volumes).

As per claim 22, the combination of Chang and Haun teaches said base instance concurrently accessible to a plurality of said compute nodes (Chang; [0017], line 3, The boot system provides access to boot images for a “plurality of clients.”).

As per claim 23, the combination of Chang and Haun teaches said compute nodes having read-only access to files in said base instance (Chang; [0026], line 2, “base image never written by the clients”) (Chang; [0022], line 15, base image is a building block for boot images, changes are made to copies to the base image).

As per claim 24, the substance of the claim language is identical to that of claim 8. Accordingly, this claim is rejected under the same rationale.

As per claim 25, the combination of Chang and Haun teaches said filter performs a copy-on-write to one of the change volumes when a compute node writes to said compute node's instantiation of said distributed application (Chang; [0024], lines 1-3, “snapshot manager further functions to control copy-on-writes to the client-specific image copies from the clients”).

As per claim 26, the combination of Chang and Haun teaches said change volumes are modified on a file-by-file basis (Chang; [0024], line 5, “files”).

As per claim 27, Chang teaches a computer-implemented method of distributed application deployment, said method comprising:

creating a root file system comprising a first file collection of an application ([0011], line 14, “base image”), read-only Accessible by a plurality of compute nodes ([0026], line 2, “base image never written by the clients”) ([0022], line 15, first file collection is a building block for boot images, changes are made to copies to the base image);;

creating a plurality of second file collections that are configured to only record changes to said first file collection of the application for said respective compute nodes ([0022], line 16, “client-specific image");

creating a directory configured to instantiate instances of said application for said respective compute nodes ([0029], lines 20-21, pool storage device with paths to each client image) ([0024], line 8, New image is a combination of first file collection and new blocks.); and

providing said compute nodes with said respective instances of said application ([0017], lines 7-11, the external storage controller provides access).

Chang does not explicitly teach an instance of said application for a compute node instantiated by accessing unmodified portions of said application for said first compute node from said first file collection and modified portions of said application for said first compute node from said respective second file collection, wherein the first file

collection and the respective node's second file collection are on separate nodes. Haun teaches an instance of said application for a compute node instantiated by accessing unmodified portions of said application for said first compute node from said first file collection (Col. 14, lines 10-11, core system volume) and modified portions of said application for said first compute node from said respective second file collection (Col. 14, line 28-29, user shadow volume), wherein the first file collection and the respective node's second file collection are on separate nodes (Col. 6, lines 45-47, volumes, i.e. file collections, application volumes and client data, i.e. second file collections, can be distributed across multiple server systems, i.e. on separate nodes).

At the time of invention, it would have been obvious to one of ordinary skill in the art to combine Chang, with the teachings of Haun, an instance of said application for a compute node instantiated by accessing unmodified portions of said application for said first compute node from said first file collection and modified portions of said application for said first compute node from said respective second file collection, wherein the first file collection and the respective node's second file collection are on separate nodes, to allow the distributed application to be modifiable as it is distributed away from the root file system.

As per claim 28, the substance of the claim language is identical to that of claim 2. Accordingly, this claim is rejected under the same rationale.

As per claim 29, the combination of Chang and Haun teaches traversing nodes as indicated by said branch modification logs (Chang; [0033], lines 17-18, unmodified common data stored at separate location from updated client-specific blocks, Traversal to common data is inherent given the successful production of a copy image.).

As per claim 30, the substance of the claim language is identical to that of claim 3. Accordingly, this claim is rejected under the same rationale.

As per claim 31, the combination of Chang and Haun teaches consulting a table of locations (Haun; Col. 5, lines 6-8, database of addresses).

As per claim 32, the substance of the claim language is identical to that of claim 8. Accordingly, this claim is rejected under the same rationale.

As per claim 33, the combination of Chang and Haun teaches access to said root image by said compute nodes being read-only (Chang; [0026], line 2, “base image never written by the clients”) (Chang; [0022], line 15, first file collection is a building block for boot images, changes are made to copies to the base image).

As per claim 34, Chang teaches freezing a version of said distributed application (Chang; [0022], line 3 and 10, “snapshot of a boot image…deployed as a base image”).

As per claim 35, the combination of Chang and Haun teaches creating intermediate file collections that are children of said frozen version of said distributed application (Chang; [0022], line 6, “for generating new images”).

As per claim 36, the combination of Chang and Haun teaches modifying a first of said plurality of second file collections in response to a corresponding compute node's access to its instance of the application (Chang; [0027], line 7, In the lazy clone technique where only modifications are stored on “client images,” the image is modified by combining it with the common blocks from the base image.).

As per claim 37, the combination of Chang and Haun teaches copying a file from said root image to said first leaf image and then modifying said title in said first leaf image (Chang; [0023], line 12, When a “client image” is formed, it is implicit that its name will differ from the base image.).

As per claim 38, the combination of Chang and Haun teaches copying a data block from said root image to said first leaf image (Chang; [0024], lines 7, “copied block”) and then modifying said data block data block in said first leaf image (Chang; [0033], line 1, “writes or changes to image”).

Response to Arguments

3. Applicant's arguments with respect to the prior art rejections of claims 1-38 have been considered but are not persuasive.

As per claim 1, Applicant argues that the cited prior arts, Chang and Haun, do not in combination teach claim 1.

Specifically, Applicant argues that the shadow system volume in Haun is not a second file collection and is not used to instantiate an instance of a distributed application but is merely a copy of the changes made by a user. Examiner interprets the shadow system volume to be used in conjunction with core system volume to instantiate a user-specific version of the OS application. Examiner would suggest expanding on the specifics of the "distributed application" as this would likely overcome the current rejection.

Applicant states that there is no motivation to combine Chang and Haun and the proceeds to present arguments why each prior art does not teach various limitations of the claimed invention. The Examiner interprets the combination of the root file system in Chang along with the core system volume to teach the root file system described in the current form of the claims. The motivation for combining the prior arts is to allow the distributed application to be modifiable as it is distributed away from the root file system.

Accordingly, this rejection is maintained.

As per all remaining claims, the above arguments are echoed. Accordingly, these rejections are maintained.

Conclusion

4. **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.

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to PETER SHAW whose telephone number is (571)270-7179. The examiner can normally be reached on Monday - Friday 7:30 A.M. to 5:00 P.M.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, JOSEPH AVELLINO can be reached on 571-272-3905. 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.

/P. S./ August 11, 2010
Examiner, Art Unit 2458

/Joseph E. Avellino/

Supervisory Patent Examiner, Art Unit 2458

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11026622	NGUYEN ET AL.
	Examiner	**Art Unit**
	PETER SHAW	2458

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

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

CLAIM		DATE								
Final	Original	03/31/2009	08/17/2009	03/09/2010	08/11/2010					
	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	✓	✓	✓	✓					

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11026622	NGUYEN ET AL.
	Examiner	**Art Unit**
	PETER SHAW	2458

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

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

CLAIM		DATE								
Final	Original	03/31/2009	08/17/2009	03/09/2010	08/11/2010					
	37	✓	✓	✓	✓					
	38	✓	✓	✓	✓					

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11026622	NGUYEN ET AL.
	Examiner	**Art Unit**
	PETER SHAW	2458

SEARCHED

Class	Subclass	Date	Examiner
709	220, 222, 203, 208	3/31/2009	PS
713	2	3/31/2009	PS
717	174,175	3/31/2009	PS
709	220, 222, 203, 208	8/17/2009	PS
713	2	8/17/2009	PS
717	174,175	8/17/2009	PS
717	174,175	3/9/2010	PS
713	2	3/9/2010	PS
709	220, 222, 203, 208	3/9/2010	PS
717	174,175	8/11/2010	PS
713	2	8/11/2010	PS
709	220, 222, 203, 208	8/11/2010	PS

SEARCH NOTES

Search Notes	Date	Examiner
EAST Database Search (hist. incl.)	3/31/2009	PS
Inventor Search (no relevant art found)	3/31/2009	PS
Interpreted Fig. 3 of drawings to comprise hardware	4/8/2009	PS
EAST Database Search (hist. incl.)	8/17/2009	PS
Inventor Search (no relevant art found)	8/17/2009	PS
EAST Database Search (hist. incl.)	3/9/2010	PS
Inventor Search (no relevant art found)	3/9/2010	PS
EAST Database Search (hist. incl.)	8/11/2010	PS
Inventor Search (no relevant art found)	8/11/2010	PS

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	2	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	OFF	2010/08/11 16:32
L2	5	"file system".ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2010/08/11 16:32
L3	24272	709/220.ccls. 709/203.ccls. 709/208.ccls. 709/222.ccls. 713/2.ccls. 717/174.ccls. 717/175.ccls.	US-PGPUB; USPAT	OR	OFF	2010/08/11 16:32
L4	31	(runtime OR dynamic).ti. AND boot.ti.	US-PGPUB; USPAT	OR	ON	2010/08/11 16:32
L5	6	boot$.ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2010/08/11 16:32
S1	1	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	OFF	2009/03/31 14:02
S2	190	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:02
S3	548	"file system".ti. AND image	US-PGPUB; USPAT	OR	ON	2009/03/31 14:03
S4	4	"file system".ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:03
S5	120453	"operating system" AND image	US-PGPUB; USPAT	OR	ON	2009/03/31 14:07
S6	195	"distributed application".ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:19
S7	133	"distributed application".ti. AND "operating system"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:19

S8	15	"thin client" AND "operating system" AND (different OR version).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:25
S9	158308	"operating system" AND "thin client" (image OR configuration).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:27
S10	172	"operating system" AND "thin client" AND (image OR configuration).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:27
S11	1	"thin client".ti. AND (os OR "operating system").ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:50
S12	6533	(os OR "operating system").ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:51
S13	73	(os OR "operating system").ti. AND "thin client"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:51
S14	1	boot$.ti. AND "root image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:55
S15	5	boot$.ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:55
S16	2	"boot image".ti. AND (root OR custom OR various OR different OR base).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:57
S17	1	("6,421,777").PN.	US-PGPUB; USPAT	OR	OFF	2009/03/31 14:59
S18	0	"bootimage".ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:01
S19	0	"boot image".ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:01
S20	1	"boot image".ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 15:02
S21	31	boot$.ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 15:03

S22	19091	709/220.ccls. 709/203.ccls. 709/208.ccls. 709/222.ccls. 713/2.ccls. 717/174.ccls. 717/175.ccls.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:19
S23	820	S22 AND boot$.ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:19
S24	28	S22 AND ("distributed application" OR "boot image").ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:20
S25	0	(runtime OR dynamic).ti. AND "boot image".ti.	US-PGPUB; USPAT	OR	OFF	2009/04/01 13:25
S26	1	(runtime OR dynamic).ti. AND "boot image".ti.	US-PGPUB; USPAT	OR	ON	2009/04/01 13:25
S27	30	(runtime OR dynamic).ti. AND boot.ti.	US-PGPUB; USPAT	OR	ON	2009/04/01 13:26
S28	20293	709/220.ccls. 709/203.ccls. 709/208.ccls. 709/222.ccls. 713/2.ccls. 717/174.ccls. 717/175.ccls.	US-PGPUB; USPAT	OR	OFF	2009/08/17 14:31
S29	31	S28 AND ("distributed application" OR "boot image").ti.	US-PGPUB; USPAT	OR	OFF	2009/08/17 14:31
S30	199	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	ON	2009/08/17 14:32
S31	30	(runtime OR dynamic).ti. AND boot.ti.	US-PGPUB; USPAT	OR	ON	2009/08/17 14:32
S32	1	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	OFF	2009/08/17 14:32
S33	22410	709/220.ccls. 709/203.ccls. 709/208.ccls. 709/222.ccls. 713/2.ccls. 717/174.ccls. 717/175.ccls.	US-PGPUB; USPAT	OR	OFF	2010/03/09 20:14

S34	1	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	OFF	2010/03/09 20:14
S35	30	(runtime OR dynamic).ti. AND boot.ti.	US-PGPUB; USPAT	OR	ON	2010/03/09 20:14
S36	218	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	ON	2010/03/09 20:14
S37	6	boot$.ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2010/03/09 20:14

EAST Search History (Interference)

<This search history is empty>

8/11/2010 4:45:06 PM
H:\mybackup\mydocuments\EAST\Workspaces\11026622.wsp

PTO/SB/08a (01-10)
Approved for use through 07/31/2012. 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.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	11026622
Filing Date	2004-12-30
First Named Inventor	NGUYEN
Art Unit	2458
Examiner Name	
Attorney Docket Number	PNTA-P002

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	7269664		2007-09-11	Hutsch, et al.	
	2	5175852		1992-12-29	Johnson, et al.	
	3	6502238		2002-12-31	Pavan, et al.	
	4	7475274		2009-01-09	Davidson	
	5	6101576		2000-08-08	Kobayashi, et al.	

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20050131962	A1	2005-06-16	Deshpande	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /P.S./

Receipt date: 04/22/2010	Application Number	11026622	11026622 - GAU: 2458
INFORMATION DISCLOSURE STATEMENT BY APPLICANT (**Not for submission under 37 CFR 1.99)**	Filing Date	2004-12-30	
	First Named Inventor	NGUYEN	
	Art Unit	2458	
	Examiner Name		
	Attorney Docket Number	PNTA-P002	

	2	20020083120	A1	2002-06-27	Soltis	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] ¡	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1		☐

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature	/Peter Shaw/	Date Considered	08/11/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

PTO/SB/30EFS (07-09)
Approved for use through 07/31/2012. 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
(Submitted Only via EFS-Web)

Application Number	11026622	Filing Date	2004-12-30	Docket Number (if applicable)	PNTA-P002	Art Unit	2458
First Named Inventor	NGUYEN, et al.			Examiner Name	SHAW, Peter C.		

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. The Instruction Sheet for this form is located at WWW.USPTO.GOV

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).

☐ 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.

- ☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on _____
- ☐ Other _____

☒ Enclosed

- ☒ Amendment/Reply
- ☐ Information Disclosure Statement (IDS)
- ☐ Affidavit(s)/ Declaration(s)
- ☐ Other _____

MISCELLANEOUS

☐ 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)

☐ Other _____

FEES

The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.
☒ The Director is hereby authorized to charge any underpayment of fees, or credit any overpayments, to Deposit Account No 504160

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

☒ Patent Practitioner Signature
☐ Applicant Signature

PTO/SB/30EFS (07-09)
Approved for use through 07/31/2012. 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.

Signature of Registered U.S. Patent Practitioner			
Signature	/Jeffrey A. Lehman/	Date (YYYY-MM-DD)	2010-11-15
Name	Jeffrey A. Lehman	Registration Number	65494

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.

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 the Freedom of Information Act requires disclosure of these records.

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 inspections 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.

Electronic Patent Application Fee Transmittal

Application Number:	11026622
Filing Date:	30-Dec-2004
Title of Invention:	Branching store file system
First Named Inventor/Applicant Name:	Tung M. Nguyen
Filer:	Jeffrey A. Lehman/Shannon Warren
Attorney Docket Number:	PNTA-P002

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Request for continued examination	1801	1	810	810
		Total in USD ($)		**810**

Electronic Acknowledgement Receipt

EFS ID:	8838753
Application Number:	11026622
International Application Number:	
Confirmation Number:	8329
Title of Invention:	Branching store file system
First Named Inventor/Applicant Name:	Tung M. Nguyen
Customer Number:	70848
Filer:	Jeffrey A. Lehman/Shannon Warren
Filer Authorized By:	Jeffrey A. Lehman
Attorney Docket Number:	PNTA-P002
Receipt Date:	15-NOV-2010
Filing Date:	30-DEC-2004
Time Stamp:	16:13:14
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Electronic Funds Transfer
Payment was successfully received in RAM	$810
RAM confirmation Number	3013
Deposit Account	
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if app.)

1	Amendment Submitted/Entered with Filing of CPA/RCE	PNTA-P002-RCEResponse.pdf	226825 e403b1f5217dadbf22259ca0228f5575171ecb82	no	16
Warnings:					
Information:					
2	Request for Continued Examination (RCE)	PNTA-P002-RCETran.pdf	768322 27f6df9e0dedadcba9d686e5efadec0dcc44975f	no	3
Warnings:					
Information:					
3	Fee Worksheet (PTO-875)	fee-info.pdf	30006 ba656cbb58b817a27762d9d09a2bf3fe14d4f55c	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	1025153		

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.

<u>New Applications Under 35 U.S.C. 111</u>
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.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
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.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: Nguyen, et al.	Confirmation	: 8329
Serial No.	: 11/026,622	Examiner	: Shaw, Peter C.
Filing Date	: 12/30/2004	Art Unit	: 2458

Attorney Docket No. : PNTA-P002

For: BRANCHING STORE FILE SYSTEM

AMENDMENT AND RESPONSE TO OFFICE ACTION AND REQUEST FOR CONTINUED EXAMINATION (RCE)

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

Dear Sir:

In response to the FINAL Office Action mailed August 13, 2010, please consider the following amendments and remarks in the above captioned patent application.

I hereby certify that this correspondence is being electronically transmitted to the USPTO via EFS on the date shown below:
November 15, 2010
/Shannon Warren/
Shannon Warren

Amendments to the Claims:

1. (Currently amended) A branching store file system, comprising:

a root file system comprising a first file collection of a distributed application, wherein said first file collection is read-only accessible by a plurality of compute nodes, and wherein said first file collection is a complete, unmodified copy of the entire distributed application;

a plurality of second file collections separate from the root file system, each comprising only modified portions of said distributed application for respective ones of said plurality of compute nodes, wherein said modified portions of said distributed application may be any portion of said distributed application, wherein an instance of said distributed application for a first compute node is instantiated by accessing the unmodified portions of the application for the first computer node from only the first file collection and the modified portions of the application for the first compute node from only at least one of the plurality of second file collections, and wherein the first file collection and the plurality of second file collections are on separate nodes; and

a filter that controls each instantiation of said distributed application for said compute nodes, wherein the filter selects at least one of the plurality of second file collections and selects portions of the first file collection to instantiate an instance of the distributed application for the first compute node, and wherein the filter adds to the second file collection of the first compute node when the first compute node makes changes to its instantiation of the distributed application.

2. (Previously presented) The system of Claim 1, wherein said plurality of second file collections comprise branch modification logs for said respective ones of said compute nodes.

3. (Previously presented) The system of Claim 1, wherein said plurality of second file collections comprise change volumes for said respective ones of said compute nodes.

4. (Original) The system of Claim 1, wherein said filter is implemented at said compute nodes.

5. (Original) The system of Claim 1, wherein said filter is implemented at said root file system.

6. (Previously presented) The system of Claim 1, further comprising intermediate file collections between said first file collection and said plurality of second file collections, wherein said intermediate file collections comprise application service groups.

7. (Previously presented) The system of Claim 6, wherein access of said compute nodes to said first file collection and intermediate file collections is read-only.

8. (Original) The system of Claim 1, wherein said distributed application comprises an operating system.

9. (Previously presented) The system of Claim 1, wherein said first file collection is concurrently accessible to a plurality of said compute nodes.

10. (Previously presented) The system of Claim 1, wherein said plurality of second file collections are configured to be modified on a block-by-block basis.

11. (Previously presented) The system of Claim 1, wherein said plurality of second file collections are configured to be modified on a file-by-file basis.

12. (Currently amended) A branching store file system, comprising:

a root file system having a root metadata structure and a root data structure, entries in said root metadata structure pointing to data in said root data structure, wherein the root data structure is an unmodified, read-only application shared by a plurality of compute nodes; and

a plurality of branch modification logs for respective ones of the plurality of compute nodes, each comprising:

a branch data structure comprising only modified portions of the application for a respective compute node, wherein said modified portions of said application may be any portion of said application; and

a branch metadata structure comprising unmodified and modified entries, said unmodified entries pointing to portions of said unmodified data in said root data structure, and said modified entries pointing to portions of said modified data in said branch data structure, wherein an instance of a distributed application for a first compute node is instantiated by accessing said modified portions of said application for said first compute node only from the branch data structure and said unmodified portions of the application for said first compute node only from said root data structure, said unmodified and modified portions of said application instantiating an instance of said application for said first compute node, wherein said root data structure and said branch data structure are on separate nodes, and wherein the branch data structure is added to when the first compute node makes changes to its instantiation of the distributed application.

13. (Original) The system of Claim 12, further comprising a filter that directs accesses of said distributed application from compute nodes to a respective compute node's instantiation of the distributed application.

14. (Original) The system of Claim 12, wherein said distributed application comprises an operating system.

15. (Original) The system of Claim 13, wherein said filter is configured to traverse inodes to said root data structure for accesses to said un-modified data.

16. (Original) The system of Claim 13, wherein said filter is configured to traverse inodes to said branch data structure for accesses to said modified data.

17. (Original) The system of Claim 12, wherein access to said root data structure by said compute nodes is read-only and access to said branch data structure is read-write-modify by said branch modification log's respective compute node.

18. (Original) The system of Claim 12, wherein ones of said modified entries have a modification attribute attached.

19. (Original) The system of Claim 12, wherein said branch modification log is configured to receive updates to said modified data on a block-by-block basis.

20. (Currently amended) A branching store file system, comprising:

a root file system comprising a base instance of a distributed application, wherein the base instance is an unmodified, read-only application shared by a plurality of compute nodes;

change volumes for respective ones of the plurality of compute nodes, each comprising only modified portions of said distributed application, wherein said modified portions of said distributed application may be any portion of said distributed application; and

a filter program controlling each instantiation of the distributed application for respective ones of the plurality of compute nodes, wherein an instance of the distributed application for a first compute node is instantiated by the filter program by accessing unmodified portions of the distributed application for the first compute node only from said base instance and modified portions of the distributed application for the first compute node only from at least one change volume, wherein said base instance and said at least one change volume are on separate nodes, and wherein the filter program adds to the at least one change volume of the first compute node when the first compute node makes changes to its instantiation of the distributed application.

21. (Original) The system of Claim 20, further comprising a table of locations indicating locations of said change volumes.

22. (Original) The system of Claim 20, wherein said base instance is concurrently accessible to a plurality of said compute nodes.

23. (Original) The system of Claim 20, wherein said compute nodes have read-only access to files in said base instance.

24. (Original) The system of Claim 21, wherein said distributed application comprises an operating system.

25. (Original) The system of Claim 20, wherein said filter performs a copy-on-write to one of the change volumes when a compute node writes to said compute node's instantiation of said distributed application.

26. (Original) The system of Claim 20, wherein said change volumes are modified on a file-by-file basis.

27. (Currently amended) A computer-implemented method of distributed application deployment, said method comprising:

creating a root file system comprising a first file collection of an application, read-only accessible by a plurality of compute nodes;

creating a plurality of second file collections that are configured to only record changes to said first file collection of the application for said respective compute nodes, wherein said second file collections may each record changes to any portion of said first file collection;

creating a directory configured to instantiate instances of said application for said respective compute nodes, wherein an instance of said application for a first compute node is instantiated by accessing unmodified portions of said application for said first compute node only from said first file collection and modified portions of said application for said first compute node only from said respective second file collection, wherein the first file collection and the plurality of second file collections are on separate nodes; and

providing said compute nodes with said respective instances of said application.

28. (Previously presented) The method of Claim 27, wherein said second file collections comprise branch modification logs.

29. (Original) The method of Claim 28, wherein said providing said compute nodes with respective instances of said application comprises traversing modes as indicated by said branch modification logs.

30. (Previously presented) The method of Claim 27, wherein said second file collections comprise change volumes.

31. (Original) The method of Claim 30, wherein said providing said compute nodes with respective instances of said application comprises consulting a table of locations.

32. (Original) The method of Claim 27, wherein said application comprises an operating system.

33. (Previously presented) The method of Claim 27, wherein access to said first file collection by said compute nodes is read-only.

34. (Original) The method of Claim 27, further comprising freezing a version of said distributed application.

35. (Previously presented) The method of Claim 34, further comprising creating intermediate file collections that are children of said frozen version of said distributed application.

36. (Previously presented) The method of Claim 27, further comprising modifying a respective one of said second file collections in response to a corresponding compute node's access to its instance of the application.

37. (Previously presented) The method of Claim 36, wherein said modifying comprises a copying a file from said first file collection to said one of said second file collections and then modifying said file in said one of said second file collections.

38. (Previously presented) The method of Claim 36, wherein said modifying comprises a copying a data block from said first file collection to said one of said second file collections and then modifying said data block in said one of said second file collections.

REMARKS

Claims 1-38 are pending. Claims 1, 12, 20, and 27 are amended. No new matter has been added as a result of these amendments.

Claim Rejections – 35 U.S.C. §103

Claims 1-38 are rejected under 35 U.S.C. 103(a) as being allegedly unpatentable over Chang (US 2003/0126242), hereinafter "Chang," in view of Haun, et al. (US 6,751,658), hereinafter "Huan." Applicants respectfully assert that Claims 1-38 are patentable over the cited combination in view of the following.

Applicants respectfully point out that the Examiner has the burden of establishing a prima facie case of obviousness. To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim features. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on Applicant's disclosure. In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). See MPEP 2100-126. Specifically, "all words in a claim must be considered when judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d., 1382 (CCPA 1970).

Moreover, in response to the recent U.S. Supreme Court decision in *KSR Int'l Co. v. Teleflex, Inc.* (U.S. 2007), new guidelines were set forth for examining obviousness under 35 U.S.C. 103. The U.S. Supreme Court reaffirmed the *Graham* factors and, while not totally rejecting the "teachings, suggestion, or motivation" test, the Court appears to now require higher scrutiny on the part of the U.S. Patent & Trademark Office. In accordance with the recently submitted guidelines, it is "now necessary to identify the reason" why a person of ordinary skill in the art would have combined the elements of cited references, or at least describe the pertinence of the elements set forth in the cited disclosure, in the manner presently claimed.

Applicants respectfully assert the combination of Chang and Haun fails to teach or suggest all the claimed elements of Claims 1-38 in view of the following rationale.

Claim 1:

Amended Claim 1 recites:

> A branching store file system, comprising:
> a root file system comprising a first file collection of a distributed application, wherein said first file collection is read-only accessible by a plurality of compute nodes, and wherein said first file collection is a complete, unmodified copy of the entire distributed application;
> a plurality of second file collections separate from the root file system, each comprising only modified portions of said distributed application for respective ones of said plurality of compute nodes, wherein said modified portions of said distributed application may be any portion of said distributed application, wherein an instance of said distributed application for a first compute node is instantiated by accessing the unmodified portions of the application for the first computer node from only the first file collection and the modified portions of the application for the first compute node from only at least one of the plurality of second file collections, and wherein the first file collection and the plurality of second file collections are on separate nodes; and

> a filter that controls each instantiation of said distributed application for said compute nodes, wherein the filter selects at least one of the plurality of second file collections and selects portions of the first file collection to instantiate an instance of the distributed application for the first compute node, and wherein the filter adds to the second file collection of the first compute node when the first compute node makes changes to its instantiation of the distributed application (emphasis added).

Applicants respectfully assert that the combination of Chang and Haun does not teach or suggest a "first file collection is a complete, unmodified copy of the entire distributed application…wherein said modified portions of said distributed application may be any portion of said distributed application, wherein an instance of said distributed application…is instantiated by accessing the unmodified portions of the application…from only the first file collection and the modified portions of the application…from only at least one of the plurality of second file collections," as claimed in amended Claim 1.

As understood by Applicants, the rejection is reading the client specific blocks of Chang (Paragraph 26) on Applicants' claimed "plurality of second file collections separate from the root file system, each comprising only modified portions of said distributed application for respective ones of said plurality of compute nodes," as claimed in Claim 1. Applicants have amended Claim 1 to clarify that the modified portions of said distributed application may be any portion of said distributed application, as claimed in amended Claim 1.

As understood by Applicants, Chang discloses a pooled storage comprising a base boot image and a plurality of client image copies created by a snapshot manager (Figure 1; Paragraphs 0022-0025). In particular, Chang discloses that each client image comprises both unmodified common OS and application blocks, as well as modified, client-specific blocks that can be added as needed to the particular client image copies (Figure 1; Paragraphs 0022-0025). To the extent that Chang discloses modified, client-specific blocks in a client image copy, Chang does not teach or suggest that the modified portions of the distributed application may be any portion of said distributed application, as claimed in Claim 1. Rather than the client-specific blocks being any portion of the client image copy, Chang teaches that the client-specific blocks do not include certain common OS and application files. In other words, there are portions of the OS and applications that are not meant to be found in a client-specific block.

Further, as understood by Applicants, the rejection agrees that Chang does not teach or suggest that an instance of said distributed application for a first compute node is instantiated by accessing the unmodified portions of the application for the first compute node from the first file collection and the modified portions of the application for the first compute node from at least one of said plurality of second file collections, wherein the first file collection and the plurality of second file collections are on separate nodes; wherein said first file collection is a complete, unmodified, read-only copy of the distributed application; and wherein the filter selects at least one of the plurality of second file collections and selects portions of the first file collection to instantiate an instance of the distributed application for the first compute node, and wherein the filter

adds to the second file collection of the first compute node when the compute node makes changes to its instantiation of the distributed application. The rejection contends that Haun cures these deficiencies. Applicants respectfully disagree.

As understood by Applicants, Haun is directed towards providing an operating system in a net-booted environment, wherein the network computer client boots from a boot image provided by a server (Abstract). Haun further discloses that in response to attempts by the client to modify the operating system files, the changes are recorded on the server separate from the system volumes storing the operating system files and into a storage area associated with the client (Abstract). As understood by Applicants, the client has access to a shared split OS that includes read-only files in a core system volume as well as a user system volume that contains user-configurable system components, including preferences and all system additions installed by application software (Figure 9; Column 14, lines 1-25). Haun teaches that the SplitOS divides components up into the core system volume (containing essential components) and a user system volume (containing user-configurable system components). In other words, the core system volume of Haun is not a first file collection containing a complete copy of an application, as claimed in Claim 1. Further, as understood by applications, the user system volume of Haun will not contain any of the portions of the application contained in the core system volume. Therefore, Haun does not teach or suggest "said modified portions of said distributed application may be any portion of said distributed application," as claimed in Claim 1.

Haun further discloses the creation of a shadow system volume for each connected client (Figure 9; Column 14, lines 26-58). As understood by Applicants, the shadow system volume "shadows" the user system volume by storing modifications that are made to the user system volume, such that whenever a changes are made to the user system volume, they are copied to the shadow system volume (Figure 9; Column 14, lines 26-58). Haun further notes that data may only be written to the shadow system volume (Column 14, lines 45-52). Applicants respectfully assert that the shadow system volume of Haun is not one of a plurality of second file collections, as claimed in Claim 1. The shadow system volume of Haun is not used to instantiate an instance of a distributed application for a compute node, as claimed in Claim 1; rather it is a copy of the changes made to the user system volume and not used for creating an instance. Applicants respectfully assert that Haun is silent with regards to using the shadow volume to instantiate an instance of an application, as claimed in Claim 1.

Therefore, Applicants respectfully assert that Claim 1 is not rendered obvious by the combination of Chang and Haun. Accordingly, Applicants respectfully assert that dependent Claims 2-11 are patentable by virtue of their dependency on patentable independent Claim 1, as well as for their additional recited patentable features.

Claim 12:

Independent Claim 12 recites features similar to that of independent Claim 1, and is therefore patentable for at least the same or similar reasons as recited above. Accordingly, Applicants respectfully assert that dependent Claims 13-19 are patentable

by virtue of their dependency on an allowable base claim, as well as for their additional recited patentable features.

Claim 20:

Independent Claim 20 recites features similar to that of independent Claim 1, and is therefore patentable for at least the same or similar reasons as recited above. Accordingly, Applicants respectfully assert that dependent Claims 21-26 are patentable by virtue of their dependency on an allowable base claim, as well as for their additional recited patentable features.

Claim 27:

Independent Claim 27 recites features similar to that of independent Claims 1, 12, and 20, and is therefore patentable for at least the same or similar reasons as recited above. Accordingly, Applicants respectfully assert that dependent Claims 28-38 are patentable by virtue of their dependency on an allowable base claim, as well as for their additional recited patentable features.

For the above reasons, Applicants request reconsideration and withdrawal of the rejections under 35 U.S.C. §103.

CONCLUSION

In light of the above listed remarks, reconsideration of rejected Claims is requested. Based on the arguments presented above, it is respectfully submitted that Claims 1-38 overcome the rejections of record and, therefore, allowance of Claims 1-38 is earnestly solicited.

Please charge any additional fees that may be required to maintain pendency of the present application, or apply any credits to our PTO deposit account number: 50-4160.

Respectfully submitted,

MURABITO, HAO & BARNES LLP

Dated: November 15, 2010

/Jeffrey A. Lehman/

Jeffrey A. Lehman
Registration No. 65,494

MURABITO, HAO & BARNES LLP
Two North Market Street
Third Floor
San Jose, California 95113

(408) 938-9060 Voice
(408) 938-9069 Facsimile

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. 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	Filing Date	☐ To be Mailed
11/026,622	12/30/2004	

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	38 minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	4 minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	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).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT 11/15/2010	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 38	Minus	** 38	=	X $ =		OR	X $ =	
Independent (37 CFR 1.16(h))	* 4	Minus	***4	=	X $ =		OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

AMENDMENT	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	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.

Legal Instrument Examiner:
/PEGGY YARBOROUGH/

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.
11/026,622	12/30/2004	Tung M. Nguyen	PNTA-P002	8329

70848 7590 12/06/2010
PANTA c/o MURABITO HAO & BARNES LLP
Two Market Street
3rd Floor
San Jose, CA 95113

EXAMINER
SHAW, PETER C

ART UNIT	PAPER NUMBER
2493	

MAIL DATE	DELIVERY MODE
12/06/2010	PAPER

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

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	11/026,622	NGUYEN ET AL.	
	Examiner	Art Unit	
	PETER SHAW	2493	

-- 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 *15 November 2010*.

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) *1-38* 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) *1-38* 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/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-38 are pending in this action.

Continued Examination Under 37 CFR 1.114

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 11/15/2010 has been entered.

Claim Rejections - 35 USC § 112

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

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

3. Claims 1, 12, 20 and 27 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. The limitation “wherein said modified portions of said distributed application may be any portion of said distributed

application" does not have support in the specification. *See response to arguments below.*

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.

4. Claim 27 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. It is unclear how a file collection can record changes to another file collection, *see* line 7.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. Claims 1, 12, 20 and 27 are rejected under 35 U.S.C. 101 because the claims are directed to nonstatutory subject matter.

As per claims 1, 12 and 20, the components described: "a root file system," "file collections," "filter," "branch modification logs," "change volumes," and "filter program," do not provide requisite structure to qualify the claimed invention as statutory since they

all can be implemented by software. Examiner suggests adding a hardware feature to each of these claims.

As per claim 27, a statutory process under 35 U.S.C. 101 must be (1) tied to a particular machine or (2) transform the underlying subject matter to a different state or thing. *See In re Bilski* 88 USPQ2d 1385 at p. 10. The “creating” and "providing" are not performed by a particular machine nor do they transform the underlying subject matter and thus the claim does not qualify as a statutory process. Examiner suggests tying these limitations to a particular machine.

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.

6. Claims 1-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chang (US PGPUB No. 2003/0126242) in view of Haun et al. (US Patent No. 6,751,658) [hereinafter “Haun”].

As per claim 1, Chang teaches a branching store file system, comprising:

a root file system comprising a first file collection of a distributed application ([0011], line 14, “base image”), wherein said first file collection is read-only accessible by a plurality compute nodes ([0026], line 2, “base image never written by the clients”) ([0022], line 15, first file collection is a building block for boot images, changes are made to copies to the base image);

a plurality of second file collections separate from the root file system, each comprising only modified portions of said distributed application for respective ones of said plurality of compute nodes ([0026], line 3, “client specific blocks”);

wherein said modified portions of said distributed application may be any portion of said distributed application ([0011], lines 11-15, client specific copies with modified portions of base image) (Spec; [0006], lines 3-5; [0026], lines 5-9; image is of a distributed application not a simple data file, where changes made result in a version of the distributed application thus Examiner interprets this limitation to only include portions that would not change some common character of the distributed application) *see 35 U.S.C. 112 rejection above*,

a filter that controls each instantiation of said distributed application for said compute nodes ([0017], lines 7-11, “external storage controller…controlling access to…boot images”).

Chang does not explicitly teach an instance of said distributed application for a first compute node is instantiated by accessing the unmodified portions of the application for the first computer node from only the first file collection and the modified portions of the application for the first compute node from only at least one of said plurality of second file collections, wherein the first file collection and the plurality of second file collections are on separate nodes; wherein said first file collection is a complete, unmodified, read-only copy of the entire distributed application; and wherein the filter selects at least one of the plurality of second file collections and selects portions of the first file collection to instantiate an instance of the distributed application for the first compute node, and wherein the filter adds to the second file collection of the first compute node when the compute node makes changes to its instantiation of the distributed application. Haun teaches an instance of said distributed application for a first compute node is instantiated by accessing the unmodified portions of the application for the first computer node from only the first file collection (Col. 14, lines 10-11, core system volume) and the modified portions of the application for the first compute node from only at least one of said plurality of second file collections (Col. 14, lines 28-29, user shadow volume), wherein the first file collection and the plurality of second file collections are on separate nodes (Col. 6, lines 45-47, volumes, i.e. file collections, application volumes and client data, i.e. second file collections, can be distributed across multiple server systems, i.e. on separate nodes); wherein said first file collection is a complete, unmodified, read-only copy of the entire distributed application (Col. 14, lines 8-10, core system volume contains all unmodified data) (Col. 14, lines 50-

52, data is only written to shadow volume thus providing a structure where changes would be reflected in the shadow volume not the user system volume which be maintained along with the core volume); and wherein the filter selects at least one of the plurality of second file collections (Col. 14, lines 18-25, user system volume contains all of the user preferences) and selects portions of the first file collection to instantiate an instance of the distributed application for the first compute node (Col. 14, line 56-58, system volume with user volume, i.e. user preferences), and wherein the filter adds to the second file collection of the first compute node when the compute node makes changes to its instantiation of the distributed application (Col. 14, lines 40-44, making changes to the user volume/shadow volume based on changes made by client).

At the time of invention, it would have been obvious to one of ordinary skill in the art to combine Chang, with the teachings of Haun, an instance of said distributed application for a first compute node is instantiated by accessing the unmodified portions of the application for the first computer node from only the first file collection and the modified portions of the application for the first compute node from only at least one of said plurality of second file collections, wherein the first file collection and the plurality of second file collections are on separate nodes; wherein said first file collection is a complete, unmodified, read-only copy of the entire distributed application; and wherein the filter selects at least one of the plurality of second file collections and selects portions of the first file collection to instantiate an instance of the distributed application for the first compute node, and wherein the filter adds to the second file collection of the first compute node when the compute node makes changes to its instantiation of the

distributed application, to allow the distributed application to be modifiable as it is distributed away from the root file system to service clients with varying requirements and demands.

As per claim 2, the combination of Chang and Haun teaches said plurality of second file collections comprising branch modification logs for said respective ones of said compute nodes (Chang; [0027], client image copies containing just the modifications to the base image, i.e. client-specific writes) (Chang; [0024], line 8, new block modifications logged and included in new image).

As per claim 3, the combination of Chang and Haun teaches said plurality of second file collections comprising change volumes for said respective ones of said compute nodes (Chang; [0025], lines 14-15, new blocks written to original volume to form new volumes).

As per claim 4, the combination of Chang and Haun teaches said filter implemented at said compute nodes (Chang; [0018], lines 1-3, “in one embodiment, the clients are configured...to provide similar or identical functions").

As per claim 5, the combination of Chang and Haun teaches said filter implemented at said root file system (Chang; [0017], line 1, "external storage controller residing in the network boot system").

As per claim 6, the combination of Chang and Haun teaches intermediate images between said first file collection and said plurality of second file collections (Chang; [0023], line 12, “client group images”), wherein said intermediate images comprise application service groups (Chang; [0023], line 12, “client group images”).

As per claim 7, the combination of Chang and Haun teaches access of said compute nodes to said intermediate images being read-only (Chang; [0026], line 2, “base image never written by the clients”).

As per claim 8, the combination of Chang and Haun teaches said distributed application comprising an operating system (Chang; [0016], line 10, “operating system”).

As per claim 9, the combination of Chang and Haun teaches said first file collection concurrently accessible to a plurality of said compute nodes (Chang; [0017], line 3, The boot system provides access to boot images for a “plurality of clients.”).

As per claim 10, the combination of Chang and Haun teaches said plurality of second file collections configured to be modified on a block-by-block basis (Chang; [0024], line 5, “data blocks”).

As per claim 11, the combination of Chang and Haun teaches said plurality of second file collections configured to be modified on a file-by-file basis (Chang; [0024], line 5, “files”).

As per claim 12, Chang teaches a branching store file system, comprising:

> a root file system having a root metadata structure and a root data structure, said in said root metadata structure pointing data in said root data structure (Chang; [0008], line 12, “Cloning…provide two collections of hard links to the same files”); and
>
> a plurality of branch modification log for respective ones of the plurality of compute nodes, comprising:
>
> > a branch data structure comprising only modified portions of the application for a respective compute node (Chang; [0027], line 10, The lazy clone technique does not contain an image, only provides instruction as to how to modify the base image to properly produce the client-specific image, i.e. metadata.); and
> >
> > wherein said modified portions of said application may be any portion of said application ([0011], lines 11-15, client specific copies with modified

portions of base image) (Spec; [0006], lines 3-5; [0026], lines 5-9; image is of a distributed application not a simple data file, where changes made result in a version of the distributed application thus Examiner interprets this limitation to only include portions that would not change some common character of the distributed application) *see see 35 U.S.C. 112 rejection above*,

a branch metadata structure comprising unmodified and modified entries, said unmodified entries pointing to portions of said unmodified data in said root data structure (Chang; [0008], line 12, "Cloning…provide two collections of hard links to the same files"), and said modified entries pointing to portions of said modified data in said branch data structure (Chang; [0027], line 10, The lazy clone technique does not contain an image, only provides instruction as to how to modify the first file collection to properly produce the client-specific image, i.e. metadata).

Chang does not explicitly teach an instance of said distributed application for a first compute nodes is instantiated by accessing said modified portions of said application for said first compute node only from the branch data structure and said unmodified portions of the application for said first compute node only from said root data structure, said unmodified and modified portions of said application instantiating an instance of said application for said first compute node, wherein said root data structure

and said branch data structure are on separate nodes; wherein the root data structure is an unmodified, read-only application shared by a plurality of compute nodes, and wherein the branch data structure is added to when the first compute node makes changes to its instantiation of the distributed application. Haun teaches an instance of said distributed application for a first compute nodes is instantiated by accessing said modified portions of said application for said first compute node only from the branch data structure (Col. 14, lines 28-29, shadow volume contains all of the user preferences) and said unmodified portions of the application for said first compute node only from said root data structure (Col. 14, lines 8-10, core system volume contains all unmodified data), said unmodified and modified portions of said application instantiating an instance of said application for said first compute node (Col. 14, line 28-29, user shadow volume) (Col. 14, lines 40-44, making changes to the user volume/shadow volume based on changes made by client), wherein said root data structure and said branch data structure are on separate nodes (Col. 6, lines 45-47, volumes, i.e. file collections, application volumes and client data, i.e. second file collections, can be distributed across multiple server systems, i.e. on separate nodes); wherein the root data structure is an unmodified, read-only application shared by a plurality of compute nodes (Col. 14, line 14, core system volume contains read-only data), and wherein the branch data structure is added to when the first compute node makes changes to its instantiation of the distributed application (Col. 14, lines 40-44, making changes to the user volume/shadow volume based on changes made by client).

At the time of invention, it would have been obvious to one of ordinary skill in the art to combine Chang, with the teachings of Haun an instance of said distributed application for a first compute nodes is instantiated by accessing said modified portions of said application for said first compute node only from the branch data structure and said unmodified portions of the application for said first compute node only from said root data structure, said unmodified and modified portions of said application instantiating an instance of said application for said first compute node, wherein said root data structure and said branch data structure are on separate nodes; wherein the root data structure is an unmodified, read-only application shared by a plurality of compute nodes, and wherein the branch data structure is added to when the first compute node makes changes to its instantiation of the distributed application, to allow the distributed application to be modifiable as it is distributed away from the root file system.

As per claim 13, the combination of Chang and Haun teaches a filter that directs accesses of said distributed application from compute nodes (Chang; [0017], lines 7-11, “external storage controller…controlling access to…boot images”) to a respective compute node's instantiation of the distributed application ([0030], lines 4-5, “access to client-specific image copy”).

As per claim 14, the substance of the claim language is identical to that of claim 8. Accordingly, this claim is rejected under the same rationale.

As per claim 15, the combination of Chang and Haun teaches said filter configured to traverse nodes to said root data structure for accesses to said unmodified data (Chang; [0033], lines 17-18, unmodified common data stored at separate location from updated client-specific blocks, Traversal to common data is inherent given the successful production of a copy image.).

As per claim 16, the combination of Chang and Haun teaches said filter configured to traverse nodes to said branch data structure for accesses to said modified data (Chang; [0033], lines 17-18, unmodified common data stored at separate location from updated client-specific blocks, Traversal to common data is inherent given the successful production of a copy image.) (Chang; [0023], line 5 and 12, Traversal to a varying first file collection or a client group image would be similarly inherent.).

As per claim 17, the combination of Chang and Haun teaches access to said root data structure by said compute nodes being read-only (Chang; [0026], line 2, “base image never written by the clients”) (Chang; [0022], line 15, base image is a building block for boot images, changes are made to copies to the base image) and access to said branch data structure being read-write-modify by said branch modification log's respective compute node (Chang; [0030], line 13, Given that the client boots from the image, it is inherent that it reads it.) (Chang; [0033], line 1, writes or changes received from clients).

As per claim 18, the combination of Chang and Haun teaches ones of said modified entries having a modification attribute attached (Chang; [0027], line 7, Lazy clone technique logs changes to the first file collection instead of writing an entire boot image. It is inherent that each of these entries would include both a pointer and what is being changed, i.e. modification attribute.).

As per claim 19, the combination of Chang and Haun teaches said branch modification log configured to receive updates to said modified data on a block-by-block basis (Chang; [0024], line 5, “data blocks”).

As per claim 20, Chang teaches a branching store file system, comprising:

> a root file system comprising a base instance of a distributed application ([0011], line 14, “base image”);
>
> change volumes for respective ones of the plurality of compute nodes and each comprising only modified portions of said distributed application ([0025], lines 14-15, new blocks written to original volume to form new volumes); and
>
> wherein said modified portions of said distributed application may be any portion of said distributed application ([0011], lines 11-15, client specific copies with

modified portions of base image) (Spec; [0006], lines 3-5; [0026], lines 5-9; image is of a distributed application not a simple data file, where changes made result in a version of the distributed application thus Examiner interprets this limitation to only include portions that would not change some common character of the distributed application) *see 35 U.S.C. 112 rejection above*,

a filter program controlling each instantiation of the distributed application for respective ones of the plurality of compute nodes ([0017], lines 7-11, “external storage controller…controlling access to…boot images”), wherein an instance of the distributed application for a first compute node is instantiated by the filter program ([0011], line 14, “base image”) (volumes ([0022], line 16, “client-specific image").

Chang does not explicitly teach an instance of said distributed application for a first of said compute nodes is instantiated by accessing unmodified portions of the distributed application for the first compute node only from said base instance and modified portions of the distributed application for the first compute node only from at least one change volume, wherein the base instance and said change volumes are on separate nodes; wherein the single base instance is an unmodified, read-only application shared by the plurality of compute nodes; and wherein the filter program adds to the the at least one change volume of the first compute node when the first compute node makes changes to its instantiation of the distributed application. Haun

teaches an instance of said distributed application for a first of said compute nodes is instantiated by accessing unmodified portions of the distributed application for the first compute node only from said base instance (Col. 14, lines 10-11, core system volume) and modified portions of the distributed application for the first compute node only from at least one change volume (Col. 14, line 28-29, user shadow volume), wherein the base instance and said change volumes are on separate nodes (Col. 6, lines 45-47, volumes, i.e. file collections, application volumes and client data, i.e. second file collections, can be distributed across multiple server systems, i.e. on separate nodes); wherein the single base instance is an unmodified, read-only application shared by the plurality of compute nodes nodes (Col. 14, line 14, core system volume contains read-only data); and wherein the filter program adds to the at least one change volume of the first compute node when the first compute node makes changes to its instantiation of the distributed application (Col. 14, lines 40-44, making changes to the user volume/shadow volume based on changes made by client).

At the time of invention, it would have been obvious to one of ordinary skill in the art to combine Chang, with the teachings of Haun, an instance of said distributed application for a first of said compute nodes is instantiated by accessing unmodified portions of the distributed application for the first compute node only from said base instance and modified portions of the distributed application for the first compute node only from at least one change volume, wherein the base instance and said change volumes are on separate nodes; wherein the single base instance is an unmodified, read-only application shared by the plurality of compute nodes; and wherein the filter

program adds to the the at least one change volume of the first compute node when the first compute node makes changes to its instantiation of the distributed application, to allow the distributed application to be modifiable as it is distributed away from the root file system.

As per claim 21, the combination of Chang and Haun teaches a table of locations indicating locations of said change volumes (Chang; [0035], line 9, discovery of the pooled storage device, id of the specific-client image, i.e. change volumes).

As per claim 22, the combination of Chang and Haun teaches said base instance concurrently accessible to a plurality of said compute nodes (Chang; [0017], line 3, The boot system provides access to boot images for a “plurality of clients.”).

As per claim 23, the combination of Chang and Haun teaches said compute nodes having read-only access to files in said base instance (Chang; [0026], line 2, “base image never written by the clients”) (Chang; [0022], line 15, base image is a building block for boot images, changes are made to copies to the base image).

As per claim 24, the substance of the claim language is identical to that of claim 8. Accordingly, this claim is rejected under the same rationale.

As per claim 25, the combination of Chang and Haun teaches said filter performs a copy-on-write to one of the change volumes when a compute node writes to said compute node's instantiation of said distributed application (Chang; [0024], lines 1-3, “snapshot manager further functions to control copy-on-writes to the client-specific image copies from the clients”).

As per claim 26, the combination of Chang and Haun teaches said change volumes are modified on a file-by-file basis (Chang; [0024], line 5, “files”).

As per claim 27, Chang teaches a computer-implemented method of distributed application deployment, said method comprising:

> creating a root file system comprising a first file collection of an application ([0011], line 14, “base image”), read-only Accessible by a plurality of compute nodes ([0026], line 2, “base image never written by the clients”) ([0022], line 15, first file collection is a building block for boot images, changes are made to copies to the base image);;
>
> creating a plurality of second file collections that are configured to only record changes to said first file collection of the application for said respective compute nodes ([0022], line 16, “client-specific image");

wherein said second file collections may each record changes to any portion of said first file collection ([0011], lines 11-15, client specific copies with modified portions of base image) (Spec; [0006], lines 3-5; [0026], lines 5-9; image is of a distributed application not a simple data file, where changes made result in a version of the distributed application thus Examiner interprets this limitation to only include portions that would not change some common character of the distributed application) *see 35 U.S.C. 112 rejection above*,

creating a directory configured to instantiate instances of said application for said respective compute nodes ([0029], lines 20-21, pool storage device with paths to each client image) ([0024], line 8, New image is a combination of first file collection and new blocks.); and

providing said compute nodes with said respective instances of said application ([0017], lines 7-11, the external storage controller provides access).

Chang does not explicitly teach an instance of said application for a compute node instantiated by accessing unmodified portions of said application for said first compute node only from said first file collection and modified portions of said application for said first compute node only from said respective second file collection, wherein the first file collection and the respective node's second file collection are on separate nodes. Haun teaches an instance of said application for a compute node instantiated

by accessing unmodified portions of said application for said first compute node only from said first file collection (Col. 14, lines 10-11, core system volume) and modified portions of said application for said first compute node only from said respective second file collection (Col. 14, line 28-29, user shadow volume), wherein the first file collection and the respective node's second file collection are on separate nodes (Col. 6, lines 45-47, volumes, i.e. file collections, application volumes and client data, i.e. second file collections, can be distributed across multiple server systems, i.e. on separate nodes).

At the time of invention, it would have been obvious to one of ordinary skill in the art to combine Chang, with the teachings of Haun, an instance of said application for a compute node instantiated by accessing unmodified portions of said application for said first compute node only from said first file collection and modified portions of said application for said first compute node only from said respective second file collection, wherein the first file collection and the respective node's second file collection are on separate nodes, to allow the distributed application to be modifiable as it is distributed away from the root file system.

As per claim 28, the substance of the claim language is identical to that of claim 2. Accordingly, this claim is rejected under the same rationale.

As per claim 29, the combination of Chang and Haun teaches traversing nodes as indicated by said branch modification logs (Chang; [0033], lines 17-18, unmodified

common data stored at separate location from updated client-specific blocks, Traversal to common data is inherent given the successful production of a copy image.).

As per claim 30, the substance of the claim language is identical to that of claim 3. Accordingly, this claim is rejected under the same rationale.

As per claim 31, the combination of Chang and Haun teaches consulting a table of locations (Haun; Col. 5, lines 6-8, database of addresses).

As per claim 32, the substance of the claim language is identical to that of claim 8. Accordingly, this claim is rejected under the same rationale.

As per claim 33, the combination of Chang and Haun teaches access to said root image by said compute nodes being read-only (Chang; [0026], line 2, “base image never written by the clients”) (Chang; [0022], line 15, first file collection is a building block for boot images, changes are made to copies to the base image).

As per claim 34, Chang teaches freezing a version of said distributed application (Chang; [0022], line 3 and 10, “snapshot of a boot image…deployed as a base image”).

As per claim 35, the combination of Chang and Haun teaches creating intermediate file collections that are children of said frozen version of said distributed application (Chang; [0022], line 6, “for generating new images”).

As per claim 36, the combination of Chang and Haun teaches modifying a first of said plurality of second file collections in response to a corresponding compute node's access to its instance of the application (Chang; [0027], line 7, In the lazy clone technique where only modifications are stored on “client images,” the image is modified by combining it with the common blocks from the base image.).

As per claim 37, the combination of Chang and Haun teaches copying a file from said root image to said first leaf image and then modifying said title in said first leaf image (Chang; [0023], line 12, When a “client image” is formed, it is implicit that its name will differ from the base image.).

As per claim 38, the combination of Chang and Haun teaches copying a data block from said root image to said first leaf image (Chang; [0024], lines 7, “copied block”) and then modifying said data block data block in said first leaf image (Chang; [0033], line 1, “writes or changes to image”).

Response to Arguments

7. Applicant's arguments with respect to the prior art rejections of claims 1-38 have been fully considered but are not persuasive.

As per claim 1, Applicant argues two points: (1) that the cited prior arts Chang and Haun do not teach “wherein said modified portions of said distributed application may be any portion of said distributed application” and (2) that the shadow volumes in Haun are not used to instantiate an instance of the distributed application.

As to the first point, Examiner submits that the images and file collections disclosed in the specification are that of a distributed application not a simple data file. *See* (Spec; [0006], lines 3-5). Furthermore, modifications made to the images and file collections are to result in a version of the distributed application. *See* (Spec; [0026], lines 5-9). Examiner has not found support in the specification for modification of any portion of the distributed application which would include common/core files that maintain some characteristic of the distributed application where the various images could still be called versions of the distributed application. A rejection has been made under 35 U.S.C. 112 first paragraph and the limitation has been interpreted in light of the specification to be only portions which would not change the image from being called a distributed application.

As to the second point, Examiner submits that the shadow volume is read from by the client in conjunction with the system volumes. Taking the broadest reasonable interpretation of "an instance of a distributed application” the Examiner submits that the

shadow volume is a second file collection or a modified portion of the distributed application.

Accordingly, this rejection is maintained.

As per all remaining claims, the above arguments are repeated. Accordingly, these rejections are maintained as well.

Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to PETER SHAW whose telephone number is (571)270-7179. The examiner can normally be reached on Monday - Friday 7:30 A.M. to 5:00 P.M.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, JOSEPH THOMAS can be reached on 571-272-6776. 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.

/P. S./
Examiner, Art Unit 2493

/Carl Colin/
Acting SPE of Art Unit 2493

December 1, 2010

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11026622	NGUYEN ET AL.
	Examiner	**Art Unit**
	PETER SHAW	2493

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

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

CLAIM		DATE								
Final	Original	03/31/2009	08/17/2009	03/09/2010	08/11/2010	12/01/2010				
	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	✓	✓	✓	✓	✓				

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11026622	NGUYEN ET AL.
	Examiner	**Art Unit**
	PETER SHAW	2493

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

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

CLAIM		DATE								
Final	Original	03/31/2009	08/17/2009	03/09/2010	08/11/2010	12/01/2010				
	37	✓	✓	✓	✓	✓				
	38	✓	✓	✓	✓	✓				

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11026622	NGUYEN ET AL.
	Examiner	**Art Unit**
	PETER SHAW	2493

SEARCHED

Class	Subclass	Date	Examiner
709	220, 222, 203, 208	3/31/2009	PS
713	2	3/31/2009	PS
717	174,175	3/31/2009	PS
709	220, 222, 203, 208	8/17/2009	PS
713	2	8/17/2009	PS
717	174,175	8/17/2009	PS
717	174,175	3/9/2010	PS
713	2	3/9/2010	PS
709	220, 222, 203, 208	3/9/2010	PS
717	174,175	8/11/2010	PS
713	2	8/11/2010	PS
709	220, 222, 203, 208	8/11/2010	PS
713	2	12/1/2010	PS
709	220, 222, 203, 208	12/1/2010	PS

SEARCH NOTES

Search Notes	Date	Examiner
EAST Database Search (hist. incl.)	3/31/2009	PS
Inventor Search (no relevant art found)	3/31/2009	PS
Interpreted Fig. 3 of drawings to comprise hardware	4/8/2009	PS
EAST Database Search (hist. incl.)	8/17/2009	PS
Inventor Search (no relevant art found)	8/17/2009	PS
EAST Database Search (hist. incl.)	3/9/2010	PS
Inventor Search (no relevant art found)	3/9/2010	PS
EAST Database Search (hist. incl.)	8/11/2010	PS
Inventor Search (no relevant art found)	8/11/2010	PS
EAST Database Search (hist. incl.)	12/1/2010	PS
Inventor Search (no relevant art found)	12/1/2010	PS
Class/subclass Search "OS image"	12/1/2010	PS
NPL Search with Google Scholar "OS image" AND deployment AND configuration AND version	12/1/2010	PS

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L8	5	image.ab. AND (multiple ADJ config $.ab.)	US-PGPUB; USPAT	OR	OFF	2010/12/01 11:38
L9	5	image.ab. AND (multiple ADJ config $).ab.	US-PGPUB; USPAT	OR	OFF	2010/12/01 11:38
L10	41	application.ab. AND (multiple ADJ config $).ab.	US-PGPUB; USPAT	OR	OFF	2010/12/01 11:38
L11	2127	application.ab. AND version.ab.	US-PGPUB; USPAT	OR	ON	2010/12/01 11:41
L12	2566	"OS image"	US-PGPUB; USPAT	OR	ON	2010/12/01 11:41
L13	82	"OS image".ab.	US-PGPUB; USPAT	OR	ON	2010/12/01 11:42
L14	25694	709/220.ccls. 709/203.ccls. 709/208.ccls. 709/222.ccls. 713/2.ccls. 717/174.ccls. 717/175.ccls.	US-PGPUB; USPAT	OR	OFF	2010/12/01 12:12
L15	213	"OS image" AND L14	US-PGPUB; USPAT	OR	ON	2010/12/01 12:12
S1	1	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	OFF	2009/03/31 14:02
S2	190	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:02
S3	548	"file system".ti. AND image	US-PGPUB; USPAT	OR	ON	2009/03/31 14:03
S4	4	"file system".ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:03

S5	120453	"operating system" AND image	US-PGPUB; USPAT	OR	ON	2009/03/31 14:07
S6	195	"distributed application".ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:19
S7	133	"distributed application".ti. AND "operating system"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:19
S8	15	"thin client" AND "operating system" AND (different OR version).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:25
S9	158308	"operating system" AND "thin client" (image OR configuration).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:27
S10	172	"operating system" AND "thin client" AND (image OR configuration).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:27
S11	1	"thin client".ti. AND (os OR "operating system").ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:50
S12	6533	(os OR "operating system").ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:51
S13	73	(os OR "operating system").ti. AND "thin client"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:51
S14	1	boot$.ti. AND "root image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:55
S15	5	boot$.ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2009/03/31 14:55
S16	2	"boot image".ti. AND (root OR custom OR various OR different OR base).ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 14:57
S17	1	("6,421,777").PN.	US-PGPUB; USPAT	OR	OFF	2009/03/31 14:59
S18	0	"bootimage".ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:01

S19	0	"boot image".ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:01
S20	1	"boot image".ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 15:02
S21	31	boot$.ti. AND distribut$.ti.	US-PGPUB; USPAT	OR	ON	2009/03/31 15:03
S22	19091	709/220.ccls. 709/203.ccls. 709/208.ccls. 709/222.ccls. 713/2.ccls. 717/174.ccls. 717/175.ccls.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:19
S23	820	S22 AND boot$.ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:19
S24	28	S22 AND ("distributed application" OR "boot image").ti.	US-PGPUB; USPAT	OR	OFF	2009/03/31 15:20
S25	0	(runtime OR dynamic).ti. AND "boot image".ti.	US-PGPUB; USPAT	OR	OFF	2009/04/01 13:25
S26	1	(runtime OR dynamic).ti. AND "boot image".ti.	US-PGPUB; USPAT	OR	ON	2009/04/01 13:25
S27	30	(runtime OR dynamic).ti. AND boot.ti.	US-PGPUB; USPAT	OR	ON	2009/04/01 13:26
S28	20293	709/220.ccls. 709/203.ccls. 709/208.ccls. 709/222.ccls. 713/2.ccls. 717/174.ccls. 717/175.ccls.	US-PGPUB; USPAT	OR	OFF	2009/08/17 14:31
S29	31	S28 AND ("distributed application" OR "boot image").ti.	US-PGPUB; USPAT	OR	OFF	2009/08/17 14:31
S30	199	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	ON	2009/08/17 14:32
S31	30	(runtime OR dynamic).ti. AND boot.ti.	US-PGPUB; USPAT	OR	ON	2009/08/17 14:32

S32	1	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	OFF	2009/08/17 14:32
S33	22410	709/220.ccls. 709/203.ccls. 709/208.ccls. 709/222.ccls. 713/2.ccls. 717/174.ccls. 717/175.ccls.	US-PGPUB; USPAT	OR	OFF	2010/03/09 20:14
S34	1	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	OFF	2010/03/09 20:14
S35	30	(runtime OR dynamic).ti. AND boot.ti.	US-PGPUB; USPAT	OR	ON	2010/03/09 20:14
S36	218	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	ON	2010/03/09 20:14
S37	6	boot$.ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2010/03/09 20:14
S38	2	"file system" AND "leaf image"	US-PGPUB; USPAT	OR	OFF	2010/08/11 16:32
S39	5	"file system".ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2010/08/11 16:32
S40	24272	709/220.ccls. 709/203.ccls. 709/208.ccls. 709/222.ccls. 713/2.ccls. 717/174.ccls. 717/175.ccls.	US-PGPUB; USPAT	OR	OFF	2010/08/11 16:32
S41	31	(runtime OR dynamic).ti. AND boot.ti.	US-PGPUB; USPAT	OR	ON	2010/08/11 16:32
S42	6	boot$.ti. AND "base image"	US-PGPUB; USPAT	OR	ON	2010/08/11 16:32

EAST Search History (Interference)

<This search history is empty>

12/1/2010 12:17:41 PM
H:\mybackup\mydocuments\EAST\Workspaces\11026622.wsp

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.
11/026,622	12/30/2004	Tung M. Nguyen	PNTA-P002	8329

70848 7590 06/22/2011
PANTA c/o MURABITO HAO & BARNES LLP
Two Market Street
3rd Floor
San Jose, CA 95113

EXAMINER
SHAW, PETER C

ART UNIT	PAPER NUMBER
2493	

MAIL DATE	DELIVERY MODE
06/22/2011	PAPER

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

The time period for reply, if any, is set in the attached communication.

Notice of Abandonment	Application No.	Applicant(s)	
	11/026,622	NGUYEN ET AL.	
	Examiner	Art Unit	
	PETER SHAW	2493	

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

This application is abandoned in view of:

1. ☒ Applicant's failure to timely file a proper reply to the Office letter mailed on *06 December 2010*.
   (a) ☐ A reply was received on _____ (with a Certificate of Mailing or Transmission dated _____), which is after the expiration of the period for reply (including a total extension of time of _____ month(s)) which expired on _____.
   (b) ☐ A proposed reply was received on _____, but it does not constitute a proper reply under 37 CFR 1.113 (a) to the final rejection.
   (A proper reply under 37 CFR 1.113 to a final rejection consists only of: (1) a timely filed amendment which places the application in condition for allowance; (2) a timely filed Notice of Appeal (with appeal fee); or (3) a timely filed Request for Continued Examination (RCE) in compliance with 37 CFR 1.114).
   (c) ☐ A reply was received on _____ but it does not constitute a proper reply, or a bona fide attempt at a proper reply, to the non-final rejection. See 37 CFR 1.85(a) and 1.111. (See explanation in box 7 below).
   (d) ☒ No reply has been received.

2. ☐ Applicant's failure to timely pay the required issue fee and publication fee, if applicable, within the statutory period of three months from the mailing date of the Notice of Allowance (PTOL-85).
   (a) ☐ The issue fee and publication fee, if applicable, was received on _____ (with a Certificate of Mailing or Transmission dated _____), which is after the expiration of the statutory period for payment of the issue fee (and publication fee) set in the Notice of Allowance (PTOL-85).
   (b) ☐ The submitted fee of $_____ is insufficient. A balance of $_____ is due.
   The issue fee required by 37 CFR 1.18 is $_____. The publication fee, if required by 37 CFR 1.18(d), is $_____.
   (c) ☐ The issue fee and publication fee, if applicable, has not been received.

3. ☐ Applicant's failure to timely file corrected drawings as required by, and within the three-month period set in, the Notice of Allowability (PTO-37).
   (a) ☐ Proposed corrected drawings were received on _____ (with a Certificate of Mailing or Transmission dated _____), which is after the expiration of the period for reply.
   (b) ☐ No corrected drawings have been received.

4. ☐ The letter of express abandonment which is signed by the attorney or agent of record, the assignee of the entire interest, or all of the applicants.

5. ☐ The letter of express abandonment which is signed by an attorney or agent (acting in a representative capacity under 37 CFR 1.34(a)) upon the filing of a continuing application.

6. ☐ The decision by the Board of Patent Appeals and Interference rendered on _____ and because the period for seeking court review of the decision has expired and there are no allowed claims.

7. ☒ The reason(s) below:

An attempt was made on June 13, 2011 to notify applicant's representative Jeffrey Lehman Reg. No. 65,494 about the abandonment status of this application.

/Carl Colin/
Supervisory Patent Examiner, Art Unit 2493

/P. S./
Examiner, Art Unit 2493

Petitions to revive under 37 CFR 1.137(a) or (b), or requests to withdraw the holding of abandonment under 37 CFR 1.181, should be promptly filed to minimize any negative effects on patent term.

RECEIVED
CENTRAL FAX CENTER

JUN 1 2 2012

PTO/SB/80 (11-08)
Approved for use through 11/30/2011. OMB 0651-0035
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.

POWER OF ATTORNEY TO PROSECUTE APPLICATIONS BEFORE THE USPTO

I hereby revoke all previous powers of attorney given in the application identified in the attached statement under 37 CFR 3.73(b).

I hereby appoint:

[x] Practitioners associated with the Customer Number: **52531**

OR

[ ] Practitioner(s) named below (if more than ten patent practitioners are to be named, then a customer number must be used):

Name	Registration Number	Name	Registration Number

as attorney(s) or agent(s) to represent the undersigned before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned only to the undersigned according to the USPTO assignment records or assignment documents attached to this form in accordance with 37 CFR 3.73(b).

Please change the correspondence address for the application identified in the attached statement under 37 CFR 3.73(b) to:

[x] The address associated with Customer Number: **52531**

OR

[ ] Firm or Individual Name					
Address					
City		State		Zip	
Country		Telephone		Email	

Assignee Name and Address:

Emendable Assets Limited Liability Company
2711 Centerville Rd
Suite 400
Wilmington, Delaware U.S.A.
19808

A copy of this form, together with a statement under 37 CFR 3.73(b) (Form PTO/SB/96 or equivalent) is required to be filed in each application in which this form is used. The statement under 37 CFR 3.73(b) may be completed by one of the practitioners appointed in this form if the appointed practitioner is authorized to act on behalf of the assignee, and must identify the application in which this Power of Attorney is to be filed.

SIGNATURE of Assignee of Record
The individual whose signature and title is supplied below is authorized to act on behalf of the assignee

Signature		Date	5/31/2012
Name	Peter Kwon	Telephone	
Title	Authorized Person for Emendable Assets Limited Liability Company		

This collection of information is required by 37 CFR 1.31, 1.32 and 1.33. 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 3 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.

DECLARATION REGARDING AUTHORITY TO SIGN
ON BEHALF OF A LEGAL ENTITY
37 C.F.R. 3.73(b)(2)(i)

I, Peter Kwon (whose title is supplied below), hereby declare that I am authorized to sign documents on behalf of Emendable Assets Limited Liability Company

Peter Kwon
Authorized Person for Emendable Assets Limited Liability Company

5/31/2012
Date

RECEIVED
CENTRAL FAX CENTER

JUN 1 2 2012

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventors:	T.M. Nguyen et al.	OLD Attorney Docket No.:	PNTA-P002
		NEW Attorney Docket No.:	139534
Application No.:	11/026,622	Confirmation No.:	8329
Filed:	December 30, 2004		

Title: BRANCHING STORE FILE SYSTEM

STATEMENT OF OWNERSHIP UNDER 37 C.F.R. § 3.73(b)

June 11, 2012

TO THE COMMISSIONER FOR PATENTS:

Emendable Assets Limited Liability Company, a limited liability company, certifies that it is the assignee of the entire right, title, and interest in the above-identified patent application by a chain of title from the inventors of the above-identified patent application to the current assignee as shown below:

1. From: Tung M. Nguyen, Richard Au, and Eric Pouyoul
   To: Panta Systems, Inc.

   The document was recorded in the U.S. Patent and Trademark Office at Reel 016149, Frame 0089.

2. From: Panta Systems, Inc.
   To: Venture Lending & Leasing IV, Inc.

   The document was recorded in the U.S. Patent and Trademark Office at Reel 028324, Frame 0627.

3. From: Venture Lending & Leasing IV, Inc.
   To: Emendable Assets Limited Liability Company

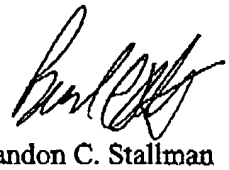
   The document was recorded in the U.S. Patent and Trademark Office at Reel 028321, Frame 0563.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

PAGE 4/5 * RCVD AT 6/12/2012 1:20:08 PM [Eastern Daylight Time] * SVR:W-PTOFAX-003/10 * DNIS:2738300 * CSID: * DURATION (mm-ss):02-07

The undersigned is an attorney of record and is authorized to act on behalf of the assignee.

Respectfully submitted,



Brandon C. Stallman
Attorney of Record
Registration No. 46,468

BCS:cae

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

39534184.DOC

PAGE 5/5 * RCVD AT 6/12/2012 1:20:08 PM [Eastern Daylight Time] * SVR:W-PTOFAX-003/10 * DNIS:2738300 * CSID: * DURATION (mm-ss):02-07

RECEIVED
CENTRAL FAX CENTER
JUN 1 2 2012

CHRISTENSEN | O'CONNOR
JOHNSON | KINDNESS

FACSIMILE COVER SHEET

DATE: June 11, 2012

TO:	Director of the U.S. Patent and Trademark Office
FACSIMILE NO:	571.273.8300
RE:	**Revocation/Power of Attorney** U.S. Patent Application No. 11/026,622
OUR REFERENCE:	OLD: PNTA-P002 **NEW: 139534**
YOUR REFERENCE:	U.S. Patent Application No. 11/026,622 Confirmation No. 8329
FROM:	Brandon C. Stallman

(Facsimile No. 206.224.0779)

MESSAGE:

Please see the attached.
Thank you.

*** The information contained in this facsimile message is privileged and confidential information intended only for the use of the recipient named above. If the reader of this message is not the intended recipient, or the employee or agent responsible to deliver it to the intended recipient, any distribution or copying of this communication is strictly prohibited. If you have received this communication in error, please notify us immediately by telephone and return the original message to us at the above address by mail. ***

We have 5 pages to send, including this sheet.
If any pages need to be retransmitted, please call 206.682.8100, Ext. 1610.

cac

T 206.682.8100
F 206.224.0779

COJK.com

1420 Fifth Avenue
Suite 2800
Seattle, WA 98101-2347

[Your Innovation Partners]

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 OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
11/026,622	12/30/2004	Tung M. Nguyen	139534

CONFIRMATION NO. 8329

POA ACCEPTANCE LETTER

52531
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 FIFTH AVENUE
SUITE 2800
SEATTLE, WA 98101-2347

OC000000055080333

Date Mailed: 07/02/2012

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 06/12/2012.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/snguyen/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

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 OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
11/026,622	12/30/2004	Tung M. Nguyen	PNTA-P002

CONFIRMATION NO. 8329

POWER OF ATTORNEY NOTICE

70848
PANTA c/o MURABITO HAO & BARNES LLP
Two Market Street
3rd Floor
San Jose, CA 95113

OC000000055080325

Date Mailed: 07/02/2012

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 06/12/2012.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

/snguyen/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101