

16	Non Patent Literature	Exhibit_C-4.pdf	1292065	no	102
			729e9599521cce5ea4824b4bed7dfb9383b8e40f		
Warnings:					
Information:					
17	Non Patent Literature	Exhibit_C-5.pdf	1358835	no	43
			a9d83f3fe1a4c1311be599d3aaace5e6f21b5d1a		
Warnings:					
Information:					
18	Non Patent Literature	Exhibit_C-6.pdf	3334841	no	81
			038bbce8358acc7485895cf22b06fea415be76b6		
Warnings:					
Information:					
19	Non Patent Literature	Exhibit_C-7.pdf	7356789	no	64
			06ce6c80de05afbdf1cb31708a187bf8e99ce180		
Warnings:					
Information:					
20	Non Patent Literature	Exhibit_C-9.pdf	10438274	no	83
			ce1e31838a9ca2789eba92bf677ac078bdc515f		
Warnings:					
Information:					
21	Non Patent Literature	Exhibit_C-10.pdf	1968918	no	54
			a8a3c617feb423497ba0f0fae427cd28b10fc404		
Warnings:					
Information:					
22	Non Patent Literature	Exhibit_C-11.pdf	10470064	no	122
			f3988e4b2b4c243b473d69111a829aa250e49fd7		
Warnings:					
Information:					

23	Non Patent Literature	Exhibit_C-12.pdf	1363034	no	102
			7baa1dba9d70a0782db2660e83e9799c040a4adc		
Warnings:					
Information:					
24	Non Patent Literature	Exhibit_C-13.pdf	1360725	no	43
			797cb69688029be865f0029f24a36e901f94202d		
Warnings:					
Information:					
25	Non Patent Literature	Exhibit_C-14.pdf	3338251	no	81
			3bde7881ded9f346ab1550b28d62dcc63fee72cd		
Warnings:					
Information:					
26	Non Patent Literature	Exhibit_C-15.pdf	3753001	no	64
			6e176426e8ce6220cce6dceaddfe5e2add772384		
Warnings:					
Information:					
27	Non Patent Literature	Exhibit_D-1.pdf	9890885	no	75
			629dda3fdd93829f16298601e6f47b9eaba7d6f9		
Warnings:					
Information:					
28	Non Patent Literature	Exhibit_D-2.pdf	1727230	no	61
			c3d1e65247ffebb076bdeb3320cddb098da4ede2		
Warnings:					
Information:					
29	Non Patent Literature	Exhibit_D-3.pdf	11379078	no	129
			7130cb14c90343ed19fc8f90d8d3626c437f13e0		
Warnings:					
Information:					

30	Non Patent Literature	Exhibit_D-4.pdf	2004540	no	155
			554e164536f13d31d40c7b5b38ad61c645e dbb73		
Warnings:					
Information:					
31	Non Patent Literature	Exhibit_D-5.pdf	790185	no	36
			a7e317b0a6f30f683de807b397f0fb02f57fc 406		
Warnings:					
Information:					
32	Non Patent Literature	Exhibit_D-6.pdf	4573875	no	97
			80b16f05cbc31b36b527709c1b67921d434 d464e		
Warnings:					
Information:					
33	Non Patent Literature	Exhibit_D-7.pdf	2765712	no	73
			644bc582115beea908fb4b97c8d57406d7d e15e3		
Warnings:					
Information:					
34	Non Patent Literature	Exhibit_D-9.pdf	10140785	no	75
			dccd920b6489df921c0963a59c4a27d6269 318a4		
Warnings:					
Information:					
35	Non Patent Literature	Exhibit_D-10.pdf	1742163	no	63
			006458769971ea9e7b23d0426e67d81460 7dadc5		
Warnings:					
Information:					
36	Non Patent Literature	Exhibit_D-11.pdf	11381762	no	129
			414dc0689ae4712f4b3a2beac5237b2e219 a6e08		
Warnings:					
Information:					

37	Non Patent Literature	Exhibit_D-12.pdf	1940669	no	155
			d5db8ef1271fcd09f023fc7f69fb693f3e37a61a		
Warnings:					
Information:					
38	Non Patent Literature	Exhibit_D-13.pdf	791319	no	36
			bccb39937372c0bdadab49bba1adff3b8485708f		
Warnings:					
Information:					
39	Non Patent Literature	Exhibit_D-14.pdf	4578138	no	97
			d3684b7724bfbcb4808bab15012ac31e85cc36c16		
Warnings:					
Information:					
40	Non Patent Literature	Exhibit_D-15.pdf	2769299	no	73
			b3adac0067b5aeaf47b426b32b97f74b5c700478		
Warnings:					
Information:					
41	Non Patent Literature	PnPParallelComputing_NVIDIA_ACS_005654.pdf	1519211	no	8
			5104c7c8d40e5a2e91267a73b4c0c06217de9c4b		
Warnings:					
Information:					
42	Non Patent Literature	DRApplHPCSeminar041202_NVIDIA_ACS_001238.pdf	4510108	no	66
			ffe87b9beb2b59dcdcee8a24f7a121dfb18e9838c		
Warnings:					
Information:					
43	Non Patent Literature	_PnPParallelComputing_NVIDIA_ACS_005654.pdf	1519292	no	8
			f48d501e34e99c08b38dc50bd9e7f66ede8624ca8		
Warnings:					
Information:					
Total Files Size (in bytes):			223578289		

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.

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
11 May 2006 (11.05.2006)

PCT

(10) International Publication Number
WO 2006/050404 A1

(51) International Patent Classification:
G06F 9/50 (2006.01)

(21) International Application Number:
PCT/US2005/039580

(22) International Filing Date: 28 October 2005 (28.10.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/623,682 29 October 2004 (29.10.2004) US
11/262,475 28 October 2005 (28.10.2005) US

(71) Applicant (for all designated States except US): **INTERACTIVE SUPERCOMPUTING, INC.** [US/US]; 135 Beaver Street, Floor 2, Waltham, MA 02452 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **HUSBANDS, Parry** [LC/US]; 82 Kains Avenue, No. 305, Albany, CA 94706 (US). **CHOY, Long, Yin** [US/CN]; 45/F Two Exchange Square, Central Hong Kong (CN). **EDELMAN, Alan** [US/US]; 20 Garland Road, Newton, MA 02459 (US).

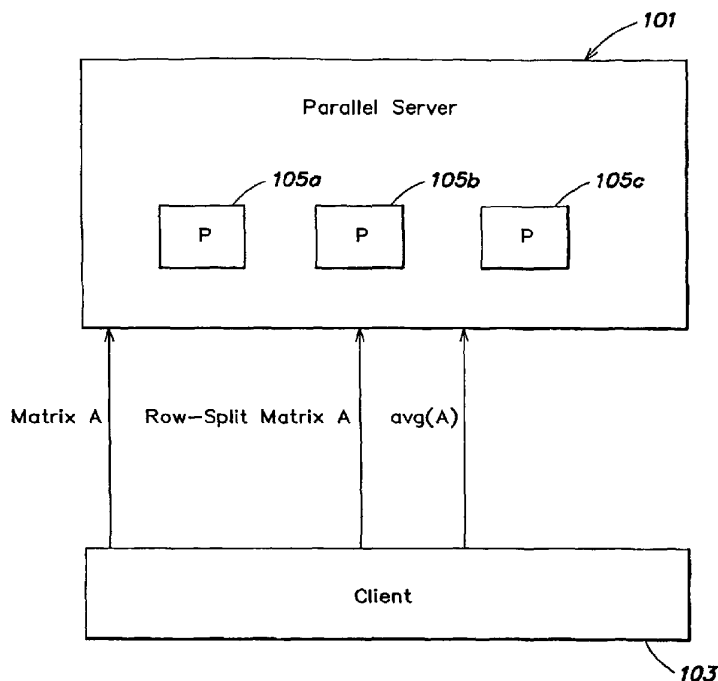
(74) Agent: **HENRY, Steven, J.**; Wolf, Greenfield & Sacks, P.C., 600 Atlantic Avenue, Boston, MA 02210 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHODS AND APPARATUS FOR PARALLEL EXECUTION OF A PROCESS



(57) Abstract: One embodiment of the invention is directed to executing a computer program that manipulates data stored in a distributed manner on a parallel server. At least one portion of the computer program treats the data from a global perspective, wherein commands to the parallel server are not specific to the manner in which the data is distributed on the parallel server. In addition, at least one portion of the computer program treats the data from a local perspective, wherein commands to the parallel server are specific to the manner in which the data is distributed on the parallel server.

WO 2006/050404 A1



Published:

- with international search report
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

- 1 -

METHODS AND APPARATUS FOR PARALLEL EXECUTION OF A PROCESSCross Reference to Related Applications

This application claims the benefit under 35 U.S.C. §119(e) of co-pending U.S. Provisional Application Serial No. 60/623,682, filed October 29, 2004, entitled "A Method And System For An Interactive Parallel Programming Environment" by Long Yin Choy, et. al., the contents of which are incorporated herein by reference.

Field of the Invention

The present invention relates to providing parallel processing capability.

Description of the Related Art

Parallel computing is the concurrent use of multiple processors to solve a computational problem. Large problems may take significant time to solve or may be impossible to solve using only a single processor. Thus, such problems may be divided among multiple processors, each of which solves a portion of the problem.

However, writing program code to solve a computational problem in parallel may present challenges. For example, a programmer may have to devise a complex algorithm to determine an efficient way in which to divide the problem up among the multiple processors, how memory should be allocated and shared among these processors, and how messages should be passed between processors.

As used herein, the term "computer program" refers to a related set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result. Different portions of the same computer program may be stored in different files and/or on different computers.

The programmer may also wish to employ pre-existing parallel computing software packages, such as, for example, parallel virtual machine (PVM) or message passing interface (MPI) packages, so that routines for sharing data between processors, spawning additional processes, and other general parallel computing tasks need not be coded from scratch. Thus, the programmer may have to be familiar with such software packages.

- 2 -

Summary

One aspect of the invention is directed to a method of processing a computer program that manipulates a data structure stored on a parallel server comprising a plurality of processors, wherein a portion of the data structure is stored on each of the plurality of processors. The method comprises acts of: (a) executing a first routine based on code in the computer program that causes a first request to perform a first operation on the data structure to be sent to the parallel server, wherein the first routine is a routine that overloads a serial routine that, when executed, causes the first operation to be performed serially; and (b) executing a second routine based on code in the computer program that causes a second request to be sent to the parallel server, wherein the second request includes an instruction to at least one specific processor of the plurality of processors to perform a second operation on the portion of the data structure stored on the at least one specific processor. Another aspect is directed to at least one computer readable medium encoded with instructions that, when executed on a computer system, perform the above-described method.

A further aspect is directed to a computer that executes a computer program for manipulating a data structure stored on a parallel server comprising a plurality of processors, wherein a portion of the data structure is stored on each of the plurality of processors. The computer comprises: an output for sending data to the parallel server; and at least one controller, coupled to the output that: (a) executes a first routine based on code in the computer program that causes a first request to perform a first operation on the data structure to be sent to the parallel server, via the output, wherein the first routine is a routine that overloads a serial routine that, when executed, causes the first operation to be performed serially; and (b) executes a second routine based on code in the computer program that causes a second request to be sent to the parallel server, via the output, wherein the second request includes an instruction to at least one specific processor of the plurality of processors to perform a second operation on the portion of the data structure stored on the at least one specific processor.

Another aspect is directed to a method of processing a computer program written in an interpreted programming language that manipulates a data structure stored on a parallel server comprising a plurality of processors, wherein a portion of the data structure is stored on each of the plurality of processors. The method comprises acts of:

- 3 -

(a) executing a first routine based on code in the computer program that causes a first request to perform a first operation on the data structure to be sent to the parallel server; and (b) executing a second routine based on code in the computer program that causes a second request to be sent to the parallel server, wherein the second request includes an instruction to at least one specific processor of the plurality of processors to perform a second operation on the portion of the data structure stored on the at least one specific processor. A further aspect is directed to at least one computer readable medium encoded with instructions that, when executed on a computer system, perform the above-described method.

Another aspect is directed to a computer that executes a computer program written in an interpreted programming language for manipulating a data structure stored on a parallel server comprising a plurality of processors, wherein a portion of the data structure is stored on each of the plurality of processors. The computer comprises: an output for sending data to the parallel server; and at least one controller, coupled to the output that: (a) executes a first routine based on code in the computer program that causes a first request to perform a first operation on the data structure to be sent to the parallel server, via the output; and (b) executes a second routine based on code in the computer program that causes a second request to be sent to the parallel server, via the output, wherein the second request includes an instruction to at least one specific processor of the plurality of processors to perform a second operation on the portion of the data structure stored on the at least one specific processor.

A further aspect is directed to a method of processing a computer program that manipulates a data structure stored on a parallel server comprising a plurality of processors, wherein a portion of the data structure is physically stored on each of the plurality of processors. The method comprises acts of: (a) executing a first routine based on code in the computer program that causes a first request to perform a first operation on the data structure to be sent to the parallel server; and (b) executing a second routine based on code in the computer program that causes a second request to be sent to the parallel server, wherein the second request defines an abstract splitting of the data structure that defines a plurality of abstract portions of the data structure, and wherein the second request includes an instruction to perform a second operation on at least one of the plurality of abstract portions.

- 4 -

Brief Description of the Drawings

Figure 1A is a block diagram of a system in which global operations on data may be performed;

Figure 1B is a block diagram of a system in which local operations on data may be performed;

Figure 2 is a block diagram of a system adaptable to perform a process of a scientific computing software application in parallel, in accordance with some embodiments for practicing the teachings herein;

Figure 3 is a block diagram of an example of parallel server that executes the server software of Figure 2, in accordance with the methods and systems as taught herein;

Figure 4 is a block diagram of an example of parallel server that executes the server software of Figure 2, in accordance with the methods and systems as taught herein;

Figure 5 is a block diagram of an object that reference local data in accordance with the methods and systems as taught herein;

Figure 6 is a block diagram of a matrix row-distributed on a parallel server.

Figure 7A is a diagram of a stencil-splitting of a matrix in accordance with the methods and systems taught herein; and

Figure 7B is a diagram of a multi-grid splitting of a matrix in accordance with the methods and systems taught herein.

Detailed Description

Many programmers have existing computer programs or computer code that was originally designed to be run serially (i.e., on a single processor). Applicants have appreciated that it often may be desirable to use such serial computer program code in the context of a parallel computing platform, to take advantage of parallel processing capabilities. Achieving this goal, however, is non-trivial.

More specifically, Applicants have appreciated that, in a parallel computing environment, a user may wish to treat data in different ways within the same parallel program. Thus, a user may wish to take a “global” view of the data, in which an object

- 5 -

stored on the parallel server is viewed by the user as a single object, even though the object may be distributed across multiple processors of the parallel computer. As used herein, the term "processor" refers to a processor core and its associated components. Thus, a single integrated circuit or chip that has multiple processor cores may be considered to be multiple processors. Thus, for example, an integrated circuit or chip with two processor cores may be considered to be two processors, an integrated circuit or chip with four processor cores may be considered to be four processors, and an integrated circuit or chip with n processor cores may be considered to be n processors.

Thus, the user need not be aware of how the object is distributed amongst the multiple processors of the parallel server or what action each individual processor takes to perform an operation on the object in parallel. An example of a series of global view operations is provided in Figure 1A.

The system of Figure 1A includes a parallel server 101 and a client 103, which sends commands to and receives results from parallel server 101. Parallel server 101 includes three processors, 105a, 105b, and 105c, though a user sending global view commands to parallel server 101 through client 103 need not be aware of the number of processors being used in parallel server 101. As shown in Figure 1A, client 103 may create a Matrix A on parallel server 101, which may be, for example, a 3x2 matrix of integer values. Parallel server 101 may distribute the data of Matrix A amongst processors 105a, 105b, and/or 105c in any suitable way and the user need not be aware of how this data is distributed or that it is distributed.

For example, parallel server 101 may row-distribute the data, such that each processor stores one row of Matrix A. That is, processor 105a may store element (0,0) and element (0,1), processor 105b may store element (1,0) and element (1,1) and processor 105c may store element (2,0) and element (2,1). Alternatively, the parallel server may column-distribute the data so that each column of Matrix A is stored on one of the processors 105 of parallel server 101. For example, element (0,0), element (1,0), and element (2,0) may be stored on processor 105a, while element (0,1), element (1,1), and element (2,1) may be stored on processor 105b. Any other suitable distribution of data may be used, as the invention is not limited in this respect.

It should be appreciated that when data is referred to herein as being stored on a particular processor or a particular processor is referred to as storing data, this does not

- 6 -

require that the data be physically stored on the processor. Rather, data being stored on a processor or a processor storing data means that the data is stored in memory that is allocated to or under control of the processor. Likewise, when data is referred to herein as being distributed amongst processors, it is not required that the data be physically stored on the processors, but rather means that the data is stored in one or more memories and each processor is allocated or controls memory that stores at least a portion of the data.

From the user's global view perspective, the manner in which Matrix A is distributed amongst the processors 105 is irrelevant. Client 103 may subsequently create a second matrix, Matrix B, on the parallel server. Matrix B may, for example, also be a 3x2 matrix of integer values and may also be distributed amongst processors 105 in any suitable way. Matrix A and Matrix B need not be distributed amongst the processors in the same way. For example, the parallel server may row-distribute Matrix A, but may column-distribute Matrix B. However, like Matrix A, the manner in which Matrix B is distributed amongst the processors 105 is irrelevant when taking a global view of the data.

Client 103 may next send a command to parallel server 101 to add Matrix A and Matrix B and store the result in a third matrix, Matrix C. Because the user is unaware of how Matrix A and Matrix B are distributed amongst processors 105, the user is unable to instruct each processor on what action to take to add Matrix A and Matrix B. However, parallel server 101 is aware of how these matrices are distributed. Thus, when parallel server 101 receives the command to add Matrix A and Matrix B, each processor 105 may be instructed to take the appropriate actions to add the two matrices. For example, if both Matrix A and Matrix B are row distributed, processor 105a may be instructed to add the content of cells Matrix A(0,0) and Matrix B(0,0) (hereinafter, simply referred to as adding the referenced cells) and to add Matrix A(0,1) and Matrix B(0,1). Likewise, processor 105b may be instructed to add Matrix A(1,0) and Matrix B(1,0) and to add Matrix A(1,1) and Matrix B(1,1). Similarly, processor 105c may be instructed to add Matrix A(2,0) and Matrix B(2,0) and to add Matrix A(2,1) and Matrix B(2,1). The result of the addition of Matrix A and Matrix B is a 3x2 row-distributed matrix (i.e., Matrix C). Parallel server 101 may return a pointer or handle that the client may use to refer to Matrix C. Client 103, using the pointer or handle returned from parallel server 101, may

- 7 -

subsequently request that parallel server return Matrix C to the client. In response, parallel server 103 may instruct each processor 105 to return the portion of Matrix C that it stores. Thus, for example, processor 105a may return Matrix C(0,0) and Matrix C(0,1), processor 105b may return Matrix C(1,0) and Matrix C(1,1), and processor 105c may return Matrix C(2,0) and Matrix C(2,1).

In addition to taking a global view of data, the user may, in some situations take a “local” view of the data. When treating data from a local perspective, a user may be aware of how an object is split amongst processors and may instruct specific processors to perform operations on the portions of the distributed object that they store. Thus, for example, if a user has previously-written serial code that operates on a certain type of data, the user may instruct each processor to execute that code on the portion of the distributed data it stores.

As used herein, a splitting of an object or data structure, such as, for example, a matrix, amongst processors may be an “abstract splitting” or a “physical distribution.” In a physical distribution, data is physically distributed among the processors in the manner specified in the splitting. For example, if a matrix is physically row-distributed, then each row of the matrix is stored on a different processor. In an abstract splitting, data is split among abstract processors and the abstract splitting may have little or nothing to do with the way in which data is physically distributed among the physical processors. The term “abstract processor” as used herein, refers to a non-physical virtual processor whose operations may be performed by one or more physical processors. For example, a 2x4 matrix may be physically row-distributed on a parallel server that has two processors, such that one row of the matrix is physically stored on each processor. However, the same matrix may be abstractly split amongst four abstract processors, such that each column of the matrix is abstractly stored on one of the four abstract processors. Thus, for the purposes of performing operations on the matrix, the matrix is treated by the parallel server and the user as if it were column-split among four processors, though the actual physical distribution of the data is a row-distribution among two physical processors.

An example of a series of local view operations is provided in Figure 1B. In Figure 1B, client 103 creates a Matrix A on parallel server 101. As in the example of Figure 1A, Matrix A may be a 3x2 matrix of integers. The user may have previously written a serial routine called, for example, ‘avg’ that takes as input a 1x2 matrix of

- 8 -

integers, computes the average of the two integers in the matrix, and outputs the average as a 1x1 matrix. Thus, Client 103 may subsequently send a command to parallel server 101 to row-split Matrix A across processors 105a, 105b, and 105c so that each processor has 1x2 matrix that is one row of Matrix A. It should be appreciated that, in some embodiments, processors 105a, 105b, and 105c may be abstract processors and the row-splitting may be an abstract splitting. That is, the data of Matrix A may be physically stored on the physical processors of parallel server 101 in a manner different from the row-splitting described above. Alternatively, in other embodiments, the row-splitting of Matrix A may be a physical distribution. That is, processors 105a, 105b, and 105c may be physical processors and one row of Matrix A may be physically stored on each of these processors.

Client 103 may instruct the parallel server to have each processor 105 execute the previously-written 'avg' routine, using the 1x2 matrix stored on the processor as input. The results of this operation is that there is a 1x1 matrix stored on each of processors 105, wherein the value in each of the 1x1 matrices is the average of the numbers in one of the rows of Matrix A. In embodiments in which Matrix A is abstractly row-split, it should be appreciated that is the abstract processors that each execute the 'avg' routine. At the physical level, depending on how the data is physically distributed among the processors, some or all of the physical processors may execute the 'avg' routine or some physical processors may execute the 'avg' routine multiple times using different portions of the matrix physically stored thereon as input. In some embodiments, the 'avg' routine is not executed at all, but rather the parallel server causes the physical processes to perform operations that would have the same result as executing the 'avg' routine on each processor that stores a row of the row-split matrix.

One embodiment of the invention is directed to providing a utility which allows a user to write a program to be executed on the parallel server that uses both a global view of data and local view of data. That is, a portion of the program may operate on the data globally, while another portion of same program operates on the data locally.

U.S. Patent Application serial no. 11/217,114, titled "METHODS AND APPARATUS FOR PARALLEL EXECUTION OF A PROCESS," filed on August 31, 2005 (hereinafter, "the '114 application"), which is hereby incorporated by reference in its entirety, describes a system in which a user may write parallel programs using a

- 9 -

global view of the data. Specifically, the application describes a system that employs a scientific computing software application to write programs which are then “parallelized” by software on the parallel server. As used herein, the term scientific computing software application refers to a software application capable of solving computational problems that are input by a user. Examples of commercially available scientific computing software applications include MATLAB™, produced and sold by The MathWorks, Natick, Massachusetts, Maple™, produced and sold by Maplesoft, Waterloo, Ontario, Canada, Mathematica™, produced and sold by Wolfram Research, Champaign, Illinois, Octave, produced and distributed by GNU, and Excel™, produced and sold by Microsoft Corporation, Redmond, WA.

The system of the ‘114 application that allows a user to operate globally on distributed data is described below and may employ a technique referred to as overloading. As used herein, the term overloading refers to having two or more objects or methods that have the same name but have different meanings or functionalities depending on the context in which the object or method is used. For example, which method is called may depend on the number of parameters with which the method is called or the data type of parameters with which the method is called. That is, a programmer may define two methods each named *Area*. The first *Area* method may take as a parameter a user-defined data type called *Triangle* and computes the area of a triangle. The second *Area* method may take as a parameter a user-defined data type called *Circle* and computes the area of a circle. Thus, simply from the line of code “B = *Area*(A),” it cannot be determined which *Area* method is to be called. That is, if the variable A is of the data type *Triangle*, the first *Area* method is called, while if the variable A is of the data type *Circle*, the second *Area* method is called.

As another example, from the programmer’s perspective, the method call for the two or more methods that share the same name may be identical. That is, the method name, the number of parameters and the type of parameters are the same for both (or all) of the methods and the decision as to which of the two or methods is called is not made by the programmer at the time of programming. For example, the line of code $x + y$, where both x and y are variables that represent matrices of integers may have different functionality depending on the context in which the line of code is written. Thus, for example, if both x and y are small matrices, an addition routine that performs a normal

- 10 -

matrix addition may be called. However, if either x or y represents a matrix larger than a certain size an addition routine that performs the matrix addition in parallel may be called. The decision as to which addition routine is to be called may be made in any suitable way as the invention is not limited in this respect. For example, the decision may be made either at compile time, at run time, or at any other suitable time. Further, the decision may be made by any suitable entity, such as the compiler or interpreter of the programming language, or an outside software program (though the compiler, interpreter, or outside software program may be manually configured, e.g., by a programmer, to make these decisions in a certain way).

As in the example above, operators may also be overloaded. For example, the function of the '+' operator in the line of code ' $C = A + B$ ' may be different depending on the data types of the operands A and B . That is, if the operands A and B are integers, the '+' operator may be defined to perform a integer addition. However, if the operands A and B are arrays or matrices, the '+' operator may be defined, by the programmer, to perform a matrix addition.

Figure 2 is an example of a system in which polymorphism (e.g., by overloading methods and/or operators) may be used to provide parallel computing capability to a scientific computing software application. The system of Figure 2 includes a scientific computing software application 201, communication software 203, and server software 205.

In one embodiment, operators and/or methods of a program of scientific computing software application 201 may be overloaded. The overloaded methods and/or operators may be defined to pass a message to communication software 203 requesting that a certain operation or process be performed in parallel. Communication software 203, in response to receiving the message, may then send a command to server software 205 to perform the process in parallel. This may be done in any suitable way, as the invention is not limited in this respect. For example, a new data type or class may be defined and operators and/or methods may be overloaded such that when the operators and/or methods are called with a parameter or operand of the newly defined data type, the overloaded method is called.

As an example, the *rand* function in the programming language of a scientific computing software application may be a built-in method of scientific computing

- 11 -

software application 201 that creates an n-by-n matrix of randomly-valued entries, where n is an integer value specified in the parameter of the method. Thus, the line of code in Table 1 creates a 100-by-100 matrix of random numbers and stores the result in the variable 'X.' The server software may then pass out tasks to a plurality of processors, to complete the task or tasks require by the method.

Table 1

$X = \text{rand}(100);$

However, the *rand* method may be overloaded so that if the parameter provided is of the new data type (i.e., as opposed to an integer or scalar), then the overloaded *rand* method is called rather than the built-in *rand* method. For example, in the line of code in Table 2, the parameter provided to the *rand* method is "100*p." The variable *p* may be an object of the new data type or class which, for the purposes of this example is called the *dlayout* class. The * operator may be the built-in multiplication operator of scientific computing software application 201, which takes two operands. However, the * operator may be overloaded, such that when one or both of its operands are objects of the *dlayout* class, the overloaded operator is called. The overloaded * operator may return an object of the *dlayout* class.

Table 2

$X = \text{rand}(100*p);$

Thus, the operation "100*p" in Table 2 may return a *dlayout* object. The *rand* method may also be overloaded so that when the parameter provided is a *dlayout* object (instead of an integer or scalar), the overloaded *rand* method is called. The overloaded *rand* method may call communication software 203 which sends a command to server software 205 to create a distributed 100-by-100 matrix. The overloaded *rand* method

- 12 -

may return an object of a user-defined data type or class that may be used as a name or handle to reference the matrix created on the parallel server. For example, the overloaded *rand* method may return an object of the user-defined *ddense* class, which is stored in the variable X. Thus, the distributed matrix on the parallel server may be accessed and manipulated using the variable X. For example, as shown in Table 3, a method for computing the eigenvalues of a matrix may be performed on the distributed matrix.

That is, the built-in method *eig* of scientific computing software application 201 may take a matrix or an array as its parameter. However, the *eig* function may be overloaded so that if the parameter provided to the method is an object of the *ddense* class, the overloaded method is called. Like the overloaded *rand* method, the overloaded *eig* method, when called, may call communication software 203 which may send a command to server software 205 to calculate the eigenvalues of the distributed matrix X. The overloaded *eig* method may also return an object of the *ddense* class, which is stored in the variable Y, and may be used as a handle to access the matrix of eigenvalues on the parallel server.

Table 3

$Y = \text{eig}(X)$

Because the overloaded methods and/or operators used to contact the parallel server as well as the parallel algorithms that execute on the parallel server may be provided for the user of the scientific computing software application (i.e., without the user having to code them), the user need not have detailed knowledge of parallel programming. Rather, much of the parallel processing capability is transparent to the user.

For example, continuing with the examples provided above, from the user's perspective, the only difference in the program code needed to create a 100-by-100 matrix of random numbers and compute its eigenvalues is the addition of the tag “*p” in the parameter of the *rand* method, as shown above in Table 2. However, the “*p” tag

- 13 -

causes a distributed matrix to be created on the parallel server and future operations on the matrix to be performed in parallel, without the user having to devise or code an parallel algorithms. In the examples above, the *rand* and *eig* methods and the *** operator were overloaded to perform certain operations in parallel, such as creating a distributed matrix on a parallel server and computing its eigenvalues in parallel. However, the invention is not limited in this respect, as any suitable built-in method or operator of the scientific computing software application may be overloaded (e.g., to cause its functionality to be performed in parallel).

In the examples described above in connection with Tables 2 and 3, the user is taking a global view of the data. That is, the objects *X* and *Y* are global objects and the user need not be aware of how these objects are distributed amongst the processors of the parallel server. Thus, when the user instructs the parallel server to computer the eigenvalues of the matrix referenced by the variable *X*, the user need not specify what specific operation each processor of the parallel server needs to take to computer these eigenvalues. Rather, server software 205 is aware of the manner in which the data is distributed and has a routine for computing, in parallel, the eigenvalues of a matrix that is distributed in this manner.

It should be appreciated that, although in the examples above certain data types, classes, methods, and/or operators are described as being user-defined, these data-types, classes, methods, and/or operators need not be defined by the ultimate end user of the scientific computing software application. Indeed, the term user-defined is merely used to distinguish between those classes, methods, and operators that are built-in to the scientific computing software application programming language by its publisher's specification and those that are added in by an external programmer. Thus, in one embodiment, the overloaded methods, operators, and user-defined classes may be provided in a software package that also includes communication software 203 and server software 205. The software package, when installed, may work in conjunction with scientific computing software application 201 to provide the above-described parallel processing functionality.

Further, in the examples above, the built-in methods and operators were only overloaded with a single additional method or operator. However, the invention is not limited in this respect, as a method or operator may be overloaded with any suitable

- 14 -

number of methods or operators. That is, in the example provided above, the * operator was overloaded with a * operator that takes an object of the *dlayout* class as one of its operands and returns a *dlayout* object. However, the * operator could also be overloaded with an operator that takes two *ddense* objects (each of which references a distributed matrix on the parallel server) as its operands, causes a matrix multiplication of the two distributed matrices to be performed in parallel, and returns a *ddense* object that serves as reference to the distributed matrix that results from the matrix multiplication. Thus, in the example above, the * operator may perform one of three different operations, depending on the data types of its operands.

Additionally, in the examples above, two data types or classes were defined (i.e., *dlayout* and *ddense*). It should be appreciated that these class names are provided as examples and any suitable class names or data type names may be used. Moreover, while in the examples above two user-defined data types or classes were used in connection with providing parallel computing capability in scientific computing software application 201, the invention is not limited in this respect, as any suitable number of user-defined data types or classes may be used. Further, any suitable data and methods may be encapsulated in these user defined data types, as the invention is not limited in this respect.

In one embodiment, scientific computing software application 201 and communication software 203 may execute on the processor of a client computer, while server software 205 may execute on multiple processors of one or more parallel servers. However, the invention is not limited in this respect, as scientific software application 201 and communication software 203 may execute on any suitable processor or processors. For example, scientific software application 201 may execute on one of the processors of the parallel server that executes server software 205. In such embodiments, the processor that executes scientific software application 201 and/or communication software 203 may be viewed as the client, while the remaining processors in parallel server 205 (individually and in collections of one or more of them) may be viewed as the parallel server. It should be appreciated that the processor that executes scientific software application 201 and/or communication software 203 may also execute a portion of server software 205. Thus, this processor may be viewed as the client as well as part of the parallel server.

- 15 -

The parallel server that executes server software 205 may be any suitable type of parallel server, as the invention is not limited in this respect. For example, the parallel server may be a symmetric multiprocessing (SMP) system, a massively parallel processor (MPP) system, or may be a Beowulf cluster. In this respect, it should be appreciated that parallel server may include a plurality of separate machines that are linked together so that they may operate on a single problem in parallel. A system having such an architecture may be viewed as a single parallel server, despite being made up of many individual computers that are also capable of operating independently of each other.

In the examples above, overloaded methods and operators are used in scientific computing software application 201 to call and/or pass messages to communication software 203 (which, in turn, sends a command to parallel server 205). These software entities may communicate in any suitable way, as the invention is not limited in this respect. For example, in one embodiment, subroutines of communication software 203 may be linked (either statically or dynamically) into scientific computing software application 201, so that the overloaded methods and/or operators of scientific computing software application may directly call these subroutines. In another embodiment, interprocess communication may be used to communicate between scientific computing software application 201 and communication software 203. Any suitable form of interprocess communication may be used, such as, for example, pipes, signals, message queues, semaphores, shared memory, sockets, or any other suitable form, as the invention is not limited in this respect.

Because scientific computing software applications are intended primarily for computing and modeling, the programming languages of these software applications typically do not provide the capability to perform and/or handle communications over a network. Thus, in one embodiment, communication software 203 may be written in a language (e.g., C or Fortran) that provides network communication capability. However, in embodiments of the invention in which the programming language of scientific computing software application 201 allows a programmer to perform network communications, communication software 203 need not be used, and the overloaded methods and operators of scientific computing software application 201 may

- 16 -

programmed to send commands to server software 205 (i.e., without using communication software 203 as an intermediary).

In embodiments wherein scientific computing software application 201 and communication software 203 reside on a separate machine from server software 205, communication software 203 or (in embodiments where communication software 203 is not used) scientific computing software application 201 may communicate with server software 205 over a network. Any suitable network protocol or protocols may be used, as the invention is not limited in this respect. For example, in some embodiments, the TCP/IP protocol may be used. In embodiments wherein scientific computing software application 201 and communication software reside on a processor of the parallel server that executes server software 205, any suitable method of communication between communication software 203 (or, in some embodiments, scientific computing software application 201) and parallel server 205 may be used. For example, network based communication may be used, even though the communications are not necessarily traveling over a network; interprocess communication may be used; or any other suitable form of communication may be used, as the invention is not limited in this respect.

Solving problems in parallel often involves operating on very large data sets. Consequently, passing results between the client and the parallel server each time they are computed may decrease performance. Thus, in one embodiment, data may remain on the parallel server, unless results are explicitly requested to be sent to the client. For example, as discussed in above, the example line of code in Table 2 causes a command to be sent to the parallel server to create a 100-by-100 matrix distributed matrix of random numbers. The parallel server may create the 100-by-100 matrix in response to the command. If the result of this computation (i.e., the 100-by-100 matrix) were to be sent back to the client rather than keeping the data on the parallel server, 10,000 numbers may have to be sent back to the client. Further, future operations on the matrix to be performed by the parallel server may involve sending the matrix data back to the parallel server from the client so that the parallel server may operate on the data. This sequence may increase network traffic, latency, and overall processing time.

However, if the data remains on the parallel server, the data need not be transferred unless the user of the scientific computing software application explicitly requests to view the data. For example, the 100-by-100 matrix created on the parallel

- 17 -

server using the line of code in Table 4 may remain on the parallel server and only a reference to that data may be returned to the scientific computing software application. In the example, in Table 4, this reference is stored in the *ddense* variable 'X.' Thus, if it is desired to perform future operations on the matrix, the matrix data can be identified and located on the parallel server using this reference.

If a user of a scientific software computing application desires to view a portion of the resulting data, the user may request that this data be returned from the parallel server and displayed. This may be done in any suitable way, as the invention is not limited in this respect. In some embodiments, method and/or operator overloading may be used to request that data stored on the parallel server be returned to the client. For example, the line of code shown in Table 4 may be used to retrieve and display the first element of the distributed matrix created by the line of code in Table 2. Scientific computing software application 201 may provide, for example, a method called *subsref* that is used to access an array or matrix element and invoked by the notation $A(R,C)$, where A is a variable that represents the array or matrix and R,C refers to the row and column of the requested matrix element. The *subsref* function may be overloaded so that if called with a *ddense* variable as its parameter instead of an array or matrix, a command may be sent to the parallel server to return the requested element of the matrix that is referenced by the *ddense* variable.

Table 4

X(1,1)

Server software 205 may operate in any suitable way to receive commands from the client and perform operations in parallel, as the invention is not limited in this respect. Figure 3 is but one example of a suitable configuration of server software 205 for at least some embodiments. In Figure 3, parallel server 301 includes a plurality of processors, 303a, 303b, ..., 303n, and a memory 313. Each of processors 303a-303n includes four software modules: client managers 305a, 305b, ..., 305n; server managers 307a, 307b, ..., 307n; library managers 309a, 309b, ..., 309n; and data managers 311a,

- 18 -

311b, ..., 311n. Memory 313 of parallel server 301 stores a plurality of software libraries 315a, 315b, 315c, ..., 315n that include routines for performing various operations.

Client managers 305 interface with clients, and provides routines for reading commands and arguments from a client and sending results and other data back to the client. Server managers 307 handle communications between server processes executing on different processors of parallel server 301. That is, server managers 307 manage the transmission of data between the processors of parallel server 301 and collect results and error codes from the processors. Library managers 309 are responsible for maintaining a list of available software libraries 315 and the routines provided by them. When instructed by a server manager 307, library manager may perform a call to a routine of one of the libraries 315. Data managers 311 include routines for creating, deleting, and changing, data stored on the parallel server. Data managers 311 maintain a mapping between the references and/or identifiers used by the client to identify the data and the actual storage location of the data.

In one embodiment, when server software 205 is initialized, one processor may be designated the head processor and all or some of the remaining processors may be designated slave processors that operate under the control of the head processor. The head processor may serve as the central processor and may receive the commands from the client. Thus, in one embodiment, only the client manager module 305 on the head processor is used. When the head processor receives a command from the client, the server manager 307 on the head processor may send messages to the "slave" processors to perform the desired operation. The library managers 309 on each of the processors may call the appropriate routine in libraries 315.

The server managers 307 on the processors 303 may communicate with each other in any suitable way, as the invention is not limited in this respect. In one embodiment, the message passage interface (MPI) application programming interface (API) may be used. Alternatively, the parallel virtual machine (PVM) API or any other suitable form of communication may be used.

In the example of Figure 2, parallel server 301 is shown having a memory 313 which stores software libraries 315. Memory 313 may be assembled from any suitable type of memory or memories and may include, for example, non-volatile storage media,

- 19 -

such as magnetic disk, optical disc, or tape, and/or volatile memory, such as random access memory (RAM). Further, in Figure 2, parallel server 301 is shown having a single memory that is shared by all processors. However, the invention is not limited in this respect, as each processor may have a separate memory or clusters of processors may share memory. For example, in embodiments in which parallel server 301 is implemented as a Beowulf cluster, each processor may have a separate memory and a separate copy of libraries 315 (or parts thereof). Further, in some embodiments in which processors 303 share a memory, portions of the memory may be allocated to each processor for exclusive use by that processor. Thus, each processor may maintain a separate copy of libraries 315 in the portion of the memory allocated to it.

In the example of Figure 3 parallel server 301 includes processors 303a-303n. It should be appreciated that the parallel server may have any suitable number of processors and the invention is not limited in this respect. Similarly, memory 313 may store any suitable number of software libraries. Examples of software libraries that may be used are ScaLAPACK and the Parallel Basic Linear Algebra Subprograms (PBLAS) library, both of which include parallel linear algebra routines.

In the example of Figure 3, server software 205 is implemented, in part, as four software modules on each processor. It should be appreciated that the invention is not limited in this respect, as server software 205 may be implemented in any suitable way. Further, it should be understood that the four modules in the above-described examples are intended to illustrate functionally how server software 205 may operate. However, the four modules need not be implemented as separate computer programs and may be implemented in any suitable way, as the invention is not limited in this respect.

In one embodiment, the system of Figure 2 may, in addition to enabling the programmer to perform parallel operations using global addressing of a distributed data (i.e., referring to and operating on the distributed data as a single object), enable the programmer to perform parallel operations using local addressing of the same distributed object (i.e., referring to and operating on portions of the distributed data on a processor-by-processor basis). That is, to multiply a Matrix A and a Matrix B that are both split amongst processors on a parallel server, a programmer may refer to the matrices globally with the line of code $C = A * B$. Alternatively, the programmer may take a local view of the two matrices by instructing the parallel server to distribute the matrices amongst the

- 20 -

processors in a particular manner, and operating on each portion of the distributed matrices separately.

An example of operating on distributed data in both a global mode and a local mode is shown in Table 5. As in the examples above, the line of code “A = rand(3*p)” creates a distributed 3x3 matrix of random numbers on the parallel server. Similarly, the line of code “B = rand(3*p)” also creates a distributed 3x3 matrix of random numbers on the parallel server. The line of code “C=A+B” is global operation that adds the matrix referenced by the variable A and the matrix referenced by the variable B. The operation is a global operation because the user does not instruct each processor which stores a portion of the two matrices what operation to perform. Rather, the user treats the two matrices as non-distributed matrices and simply instructs the parallel server to add them. The intelligence of what operation(s) each processor needs to perform in order to add the two matrices resides in the parallel server and the user need not be aware of how this is done.

However, in the subsequent line of code that reads “D=ppeval(‘avg’, split(C,2))” the user operates on the data locally on each processor. The function *ppeval* may be a function that takes two arguments. The first argument may be the name of a function to be executed on each processor and the second argument is the data to be input into the function and a specification of how that data should be split amongst the processors.

In the example of Table 5, the name of the function to be executed on each processor is ‘avg.’ This may be a previously-written routine that takes as input a 3x1 matrix, computes the average of the numbers in the matrix and returns a scalar that is the average of the numbers. This routine may be executed on each of the processors of the parallel server that stores a portion of the input data. The routine may be provided to each of these processors in any suitable way. In some embodiments, the routine may be stored on the client. In such embodiments, the routine may be provided from the client to the parallel server in a manner that is transparent to the user. For example, the routine may be stored as a file in a file system on the client. The *ppeval* function may access the file that includes the program code for the routine on the client and send it to the parallel server for execution on each processor. In some embodiments, the parallel server may cache the routine thereon for future use. Thus, for example, if the same routine is used again in the computer program (e.g., in a subsequent invocation of the *ppeval* function)

- 21 -

or in a different computer program, the *ppeval* function may determine if the routine has changed since the last time it was provided to the parallel server. If the routine has not changed, then the parallel server may use the cached version, without the client having to provide the routine to the parallel server again. If the routine has changed since the last time it was provided to the parallel server, then the new version of the routine may be provided to the parallel server. The determination as to whether a routine has changed since it was last provided to the parallel server may be made in any suitable way, as the invention is not limited in this respect. For example, in embodiments in which the routine is stored as a file in a file system, the file system timestamp for the file may be used in determining when the routine was last modified.

In some embodiments, the file that includes the program code for the routine may be stored in memory on the parallel server, in addition to or instead of on the client. In such embodiments, a copy of the file may be stored on each processor or a single copy of the file may be stored in a memory location accessible to all processors. This allows a user to execute previously-written serial routines that the user desires to re-use in the context of parallel program.

In the examples above, the *ppeval* function is a function that accepts two arguments (i.e., the name of a routine to be executed and the input data to the routine). However, the invention is not limited in this respect, as the function used to execute a routine in local mode may take any suitable form, may have any suitable name, and may take any suitable number of arguments. For example, the *ppeval* function may take any number of arguments, where the first argument is the name of a serial routine that takes multiple arguments and the subsequent arguments are the multiple inputs to the routine. It should further be appreciated that the arguments may be presented in any suitable order, as the invention is not limited to providing the arguments in the illustrative order shown in the examples above.

Figure 4 is one example of a suitable configuration of server software 205 that may be used in a system which enables both global and local operations. The server software is similar to the server software shown in Figure 3, but each processor 303 also includes a scientific computing software application interpreter 401. That is, processor 303a includes scientific computing software application interpreter 401a, processor 303b includes scientific computing software application interpreter 401b, and processor 303n

- 22 -

includes scientific computing software application interpreter 401n. Scientific computing software application interpreters 401 are provided on each processor to allow each processor to locally execute routines provided by the *ppeval* function. For example, the 'avg' routine that is the first argument in the *ppeval* function may be a routine written in the language of scientific computing software application 201. When this routine is provided to each processor 303, scientific computing software application interpreter 401 may interpret the routine allowing it to be executed on each processor.

In the example above, the 'avg' routine is a routine written in the language of the scientific computing software application 202. Thus, for example, the computer program excerpt shown in Table 5 may be written in the language of scientific computing software application 202 and interpreted by an instance of the scientific computing software application interpreter executing on the client. Further, the 'avg' routine may be written in the same programming language and interpreted by instances of scientific computing software application interpreter executing on each of the processors of the parallel server. However, the invention is not limited in this respect, as the client program and the routine to be executed on a particular processor of the parallel server may be written in different programming languages. For example, the client program (e.g., the program of Table 5) may be written in the language of the scientific computing software application, and the routine specified in the client program that is to be executed on a particular processor or processors of the parallel server (e.g., the 'avg' routine) may be written in a different program language. For example, this routine may be written in Perl and each processor of the parallel server may execute a Perl interpreter. Indeed, any suitable programming language may be used and each processor of the parallel server may have an interpreter for processing that particular programming language.

In some embodiments, the programming language in which the client program and/or the routine to be executed on each processor of the parallel server (e.g., the 'avg' routine) are written is not an interpreted language, but rather is a compiled language. As used herein, an interpreted language refers to a programming language in which the processor instructions (e.g., the machine level instructions) of a program written in the language are generated from the program at run time. A compiled language refers to a programming language in which the processor instructions of a program written in the language are generated from the program prior to run time. In some embodiments in

- 23 -

which the routine to be executed on the processors of the parallel server is written in a compiled language, each processor may be provided with a compiler for that application so that it may receive the routine and compile it, so that the routine may be executed on the processor. In other embodiment, the routine may be compiled and assembled before being provided to the processors of the parallel server, such that the object code of the routine is provide to the processors. Thus, the processors need not compile or interpret the routine, but may simply execute the object code. Further, in such embodiments, the processors need not be provided with a compiler or an interpreter to compile or interpret the routine.

Table 5

```

A = rand(3*p);
B= rand(3*p);
C = A+B;
D = ppeval('avg', split(C,2));
D.a = D.a+1;
D.b = D.b+1;
D.c = D.c+1;
E = ppjoin(D, 3, 1);
E(0,0)

```

The second argument in the *ppeval* function specifies the input data for the function specified in the first argument and how the input data should be split amongst the processors. In the example of Table 5, the second argument to the *ppeval* function is “split(C,2).” This argument indicates that the input data to the ‘avg’ routine is the matrix referenced by the variable C. The function *split* is a function that causes the data of the matrix C to be split amongst the processors of the parallel server. As discussed above, this split may be an abstract splitting or a physical distribution. The first argument of the *split* functions specifies the data to be distributed (as well as the data to be input into the ‘avg’ function) and the second argument specifies the manner in which the data is to be distributed. In the example of Table 5, the data to be distributed is, as mentioned above, the matrix referenced by the variable C. The second argument to the *split* function is

- 24 -

“2”. This indicates that the data is to be distributed along the second dimension of the n -dimensional matrix. Because the matrix referenced by the variable C is a two-dimensional matrix, the second dimension of the matrix is its width (i.e., the number of columns). Thus, the “2” indicates that the data of the matrix is to be column-split. In the example above, the matrix referenced by the variable C is 3×3 matrix, thus the data of the matrix will be split amongst three processors, with each processor storing one column of the matrix. If the arguments provided to the *split* function were *split*($C,1$) rather than *split*($C,2$), then the matrix would be split along the first dimension of the matrix (i.e., its height). Thus, *split*($C,1$) would result in the matrix being row-split amongst three processors, which each processor storing one column of the matrix.

In the examples discussed above in connection with Table 5, two possible splittings of the matrix are discussed (i.e., row-distribution and column-distribution) and syntax is provided for distributing a matrix along its n^{th} dimension.. It should be appreciated that the invention is not limited in this respect as the user may specify any suitable splitting to the parallel server using any suitable syntax. For example, the matrix could be split along its diagonal, could be split into $m \times n$ blocks that are each stored on one of processors, or could be split in any other suitable way. In one embodiment, rather than splitting the matrix and distributing it, the entire matrix may be provided to each processor. For example, as shown in the example of Table 6, rather than distribute the matrix referenced by object C amongst the processors of the parallel server, the matrix may be broadcast to each processor, such that each processor stores its own copy of the entire matrix. Thus, rather than specifying *split*($C, 2$) in the argument specifying the object to be distributed, *bcast*(C) is provided, indicating that the matrix referenced by the object C should be provided to each processor.

Table 6

$D = \text{ppeval}(\text{'avg'}, \text{bcast}(C));$

In some embodiments, a splitting may cause an overlap of data among processors. For example, a three-day-moving-average split may be used, wherein a 1×7 matrix is split such that a first processor stores elements 1, 2, and 3 of the matrix, a second processor stores elements 2, 3, and 4, a third processor stores elements 3, 4, and

- 25 -

5, a fourth processor stores elements 4, 5, and 6, and a fifth processor stores elements 5, 6, and 7.

Another example of a suitable splitting is a stencil splitting, whereby each element (i,j) in a matrix and its north, south, east, and west neighbors is stored on a different processors. For example, as shown in Figure 7A, element (2,3) of matrix 701 may be provided to a processor with its north, south, east, and west neighbors: element (1,3); element (2,2); element (2,4); and element (3,3). This may be done for each element (i,j) in the matrix. For elements along an edge of the matrix, such as element (4,1), the neighbors may wrap around or may be cut off and treated as zero. For example, the north neighbor of element (4,1) is element (3,1) and the west neighbor is element (4,2). In embodiments which employ a wrap-around, its east neighbor may be treated as element (4,4) and its south neighbor may be treated as element (1,1). In embodiments which employ a cut-off, its east and south neighbors may be treated as elements with a value of zero.

Another example of a suitable splitting is referred to as a multi-grid splitting. In a multi-grid splitting, each odd element and its subsequent even neighbor in a vector are averaged to generate a new element and the new elements are combined to create a new vector. This process is repeated on the new vector and subsequent new vectors until a vector of only a single element (i.e., a scalar) is generated. For example, as shown in Figure 7B, in vector 703, elements 1 and 2 are averaged, elements 3 and 4 are averaged, element 5 and 6 are averaged, and elements 7 and 8 are averaged. The result of each of these average operations is placed in a vector 705. In vector 705, elements 1 and 2 are averaged and elements 3 and 4 are averaged, and the result of each of these average operations is placed in a vector 707. In vector 707, elements 1 and 2 are averaged to generate vector 709 (a scalar). Each vector (i.e., 703, 705, 707, and 709) may be stored on a different processor.

Another example of splitting is a splitting of a spreadsheet, such as an ExcelTM, that is stored on the parallel server. That is, for example, if the spreadsheet has multiple sheets, each sheet may be stored on a processor of the parallel server. Alternatively, a single sheet in a spreadsheet may be row-split or column-split among the processors of the parallel server.

- 26 -

The illustrative splittings described above are only a few examples of splittings that may be used in various embodiments. The invention is not limited to these splittings, as any suitable splitting may be used.

As discussed in connection with the examples above, a user may specify a manner in which data is to be split amongst the processors of the parallel server and an operation for each processor to perform on its portion of the data. In some embodiments, the parallel server may not physically distribute the data in the manner specified by the user, but rather the may maintain an abstract splitting of the data. For example, as shown in Figure 6, parallel server 601 may have three physical processors: 603a, 603b, and 603c. Matrix 605 is a 2x3 matrix that is physically row-distributed on processors 603a and 603b of parallel server 601. That is, the first row of matrix 605 is stored on processor 603b and the second row of matrix 605 is stored on processor 603a. A user's program may include the line of code shown in Table 7 which when executed causes the data referenced by the object 'C' (i.e., matrix 605 in this example) to be column distributed and the 'avg' routine to be run on each processor using each respective processor's column of matrix 605 as input to the 'avg' routine. Rather than physically redistribute the data such that each of physical processors 603a, 603b, and 603c stores one column of matrix 605, matrix 605 may remain physically row-distributed on processors on 603a and 603b. However, both the user and the parallel server may treat and operate on matrix 605 as if it were column-distributed over three processors. As discussed above, this is referred to as an abstract splitting and the three processor over which the matrix is abstractly split may be viewed as abstract processors. Though the user need only be aware of the abstract splitting, the parallel server is aware of both the abstract splitting and the underlying physical distribution of the data and may perform operations on the physical data using the physical processors such that the user's desired result of the operation or operations using the abstract splitting among the abstract processors is achieved. This may be done in any suitable way, as the invention is not limited in this respect.

For example, referring to Figure 6, the parallel server may achieve the user's desired result of the operation in Table 7 (i.e., averaging the elements in each column of the Matrix C) as follows. Processor 603a may send the first element of its row (i.e., "9") to processor 603b, which may use the first element from row 2 received from processor

- 27 -

603b and the first element from row 1 (i.e., "5") stored thereon as input to the 'avg' routine which is executed on processor 603b. The output of the 'avg' routine may be stored on processor 603b. Processor 603a may then send the second element of its row to processor 603b, which may then execute the 'avg' routine again using the second element of row 2 received from processor 603a and the second element of row 1 stored thereon as input to the 'avg' routine and may store the result. Processor 603a may then send the third element of its row to processor 603b, which may then execute the 'avg' routine for a third time using the third element of row 2 received from processor 603a and the third element of row 1 stored thereon as input to the 'avg' routine and may store the result.

Thus, from the perspective of the user, matrix 605a is column-distributed across three processors, though the physical distribution of the data is a row distribution across two processors. In addition, the parallel server may return a pointer or a handle that the client may use to access the abstract matrices resulting from the performance of the operation indicated in Table 7. This pointer or handle may be stored in the object D, as shown in Table 7.

Table 7

D = ppeval('avg', split(C,2));

As discussed above, the *ppeval* function in Table 5 causes a command to be sent to the parallel server to column-distribute the data of the matrix referenced by the variable C and to execute the 'avg' routine on each processor that stores a portion of the matrix, using the portion of the matrix stored on that processor as input to the routine. In the example of Table 5, the result of running the 'avg' routine on a column of the column-distributed matrix is a scalar that is the average of the values in the column. That is, the result of the *ppeval* function is that each of the three processors that stores a column of the matrix referenced by the variable C also stores a scalar that is the average of the values in the column stored by the processor.

In some embodiments, the *ppeval* function returns an object that includes a reference to each of the local pieces of the output of the function specified in the first argument of the *ppeval* function. Thus, in the example of Table 5, the object D includes a reference to each of the outputs of the 'avg' function. That is, the object D includes a

- 28 -

reference to each scalar output by the 'avg' function that is stored on one of the three processors on which the 'avg' function was executed. For example, as shown in Figure 5, parallel server 501 may have three processors 503a, 503b, and 503c. Each of these processors may store a scalar that is the output of the 'avg' function. Object D 505, which is returned by the *ppeval* function, may include a reference to each of the scalars.

As shown in Table 5, the user may manipulate and operate on the local data (i.e., the scalars) using the reference to the data in the object D. For example, the line of code $D.a = D.a + 1$ causes processor 503a to add one to the value of the scalar stored thereon. Similarly, the line of code $D.b = D.b + 1$ causes processor 503b to add one to the value of its scalar and the line of code $D.c = D.c + 1$ causes processor 503c to add one to the value of its scalar.

The user may subsequently join the local pieces into a global object and operate on the global object using a global view. This may be done in any suitable way, as the invention is not limited in this respect. For example, as shown in Table 5, the line of code " $E = ppjoin(D, 3, 1)$ " joins the local pieces referenced by the object the D and creates a global object, E, that references the matrix resulting from joining the scalars referenced by the object D. The function *ppjoin* is a function that causes a command to be sent to the parallel server to join local pieces into a global object. In the example of Table 5, the *ppjoin* function takes three arguments. The first argument is the object that references the local pieces (in this example, object D). The second and third arguments are the dimensions of the global matrix into which the object will be reconstituted. Thus, in the example of Table 5, the data of referenced by the object D may be joined in a 3x1 matrix.

In the example in Table 5 a user selects whether an operation is to be performed globally or locally. However, the invention is not limited in this respect, as in some embodiments, the decision whether to perform an operation globally or locally may be made automatically. This may be done in any suitable way, as the invention is not limited in this respect. For example, a pre-processor may recognize serial portions of a user's program that are suited for parallel local operation and cause these portions to be executed locally in parallel, rather than serially. For example, as shown in Table 8, rather than execute the "while" loop five times serially on a single processor, the pre-processor may recognize that the five iterations of the "while" loop may be performed in

- 29 -

parallel and may cause five processors of the parallel server to each perform the operation $a[n] = b * n$ in parallel, each using a different value of n (i.e., where n is 0 through 4).

Table 8

<pre>n=1; while (n < 6){ a[n] = b*n; n = n + 1; }</pre>
--

The embodiments described above may incorporate any or all of the above-described aspects of the invention. Further, the above-described embodiments of the invention are not mutually exclusive of each other, such that any feature or aspect of one embodiment may be used in any other embodiment.

The above-described embodiments of the present invention can be implemented in any of numerous ways. For example, the embodiments may be implemented using hardware, software or a combination thereof. When implemented in software, the software code can be executed on any suitable processor or collection of processors, whether provided in a single computer or distributed among multiple computers. It should be appreciated that any component or collection of components that perform the functions described above can be generically considered as one or more controllers that control the above-discussed functions. The one or more controllers can be implemented in numerous ways, such as with dedicated hardware, or with general purpose hardware (e.g., one or more processors) that is programmed using microcode or software to perform the functions recited above. Further, when the embodiments are implemented in software, such software may be accessed via download over a network.

It should be appreciated that one implementation of the present invention comprises at least one computer-readable medium (e.g., a computer memory, a floppy disk, a compact disk, a tape, etc.) encoded with a computer program (i.e., a plurality of instructions), which, when executed on a processor, performs the above-discussed functions of the embodiments of the present invention. The computer-readable medium

- 30 -

can be transportable such that the program stored thereon can be loaded onto any computer environment resource to implement the aspects of the present invention discussed herein. In addition, it should be appreciated that the reference to a computer program which, when executed, performs the above-discussed functions, is not limited to an application program running on a host computer. Rather, the term computer program is used herein in a generic sense to reference any type of computer code (e.g., software or microcode) that can be employed to program a processor to implement the above-discussed aspects of the present invention.

It should be appreciated that in accordance with several embodiments of the present invention wherein processes are implemented in a computer readable medium, the computer implemented processes may, during the course of their execution, receive input manually (e.g., from a user and/or via external electrical connection).

The phraseology and terminology used herein is for the purpose of description and should not be regarded as limiting. The use of "including," "comprising," "having," "containing", "involving", and variations thereof, is meant to encompass the items listed thereafter and additional items.

Having described several embodiments of the invention in detail, various modifications and improvements will readily occur to those skilled in the art. Such modifications and improvements are intended to be within the spirit and scope of the invention. Accordingly, the foregoing description is by way of example only, and is not intended as limiting. The invention is limited only as defined by the following claims and the equivalents thereto.

What is claimed is:

- 31 -

CLAIMS

1. A method of processing a computer program that manipulates a data structure stored on a parallel server comprising a plurality of processors, wherein a portion of the data structure is stored on each of the plurality of processors, and wherein the method comprises acts of:

(a) executing a first routine based on code in the computer program that causes a first request to perform a first operation on the data structure to be sent to the parallel server, wherein the first routine is a routine that overloads a serial routine that, when executed, causes the first operation to be performed serially; and

(b) executing a second routine based on code in the computer program that causes a second request to be sent to the parallel server, wherein the second request includes an instruction to at least one specific processor of the plurality of processors to perform a second operation on the portion of the data structure stored on the at least one specific processor.

2. The method of claim 1, wherein the computer program is written in an interpreted programming language.

3. The method of claim 1, wherein the computer program is written in a compiled programming language.

4. The method of claim 1, wherein the second request specifies a splitting of the data structure on the parallel server.

5. The method of claim 4, wherein the splitting of the data structure is an abstract splitting.

6. The method of claim 4, wherein the splitting of the data structure is a physical distribution.

- 32 -

7. The method of claim 1, wherein the computer program is a first computer program and the second request includes a second computer program that, when executed on the at least one specific processor, performs the second operation on the portion of the data structure stored on the at least one specific processor.

8. The method of claim 7, wherein the first computer program and the second computer program are written in a same programming language.

9. The method of claim 7, wherein the first computer program and the second computer program are written in different programming languages.

10. The method of claim 1, wherein the parallel server includes at least one processor in addition to the plurality of processors.

11. The method of claim 1, wherein the first routine begins execution before the second routine.

12. The method of claim 1, wherein the second routine begins execution before the first routine.

13. The method of claim 1, wherein the plurality of processors are abstract processors.

14. The method of claim 1, wherein the plurality of processors are physical processors.

15. At least one computer readable medium encoded with instructions that, when executed on a computer system, perform a method of processing a computer program that manipulates a data structure stored on a parallel server comprising a plurality of processors, wherein a portion of the data structure is stored on each of the plurality of processors, and wherein the method comprises acts of:

- 33 -

(a) executing a first routine based on code in the computer program that causes a first request to perform a first operation on the data structure to be sent to the parallel server, wherein the first routine is a routine that overloads a serial routine that, when executed, causes the first operation to be performed serially; and

(b) executing a second routine based on code in the computer program that causes a second request to be sent to the parallel server, wherein the second request includes an instruction to at least one specific processor of the plurality of processors to perform a second operation on the portion of the data structure stored on the at least one specific processor.

16. The at least one computer readable medium of claim 15, wherein the computer program is written in an interpreted programming language.

17. The at least one computer readable medium of claim 15, wherein the computer program is written in a compiled programming language.

18. The at least one computer readable medium of claim 15, wherein the second request specifies a splitting of the data structure on the parallel server.

19. The at least one computer readable medium of claim 18, wherein the splitting of the data structure is an abstract splitting.

20. The at least one computer readable medium of claim 18, wherein the splitting of the data structure is a physical distribution.

21. The at least one computer readable medium of claim 15, wherein the computer program is a first computer program and the second request includes a second computer program that, when executed on the at least one specific processor, performs the second operation on the portion of the data structure stored on the at least one specific processor.

- 34 -

22. The at least one computer readable medium of claim 21, wherein the first computer program and the second computer program are written in a same programming language.

23. The at least one computer readable medium of claim 21, wherein the first computer program and the second computer program are written in different programming languages.

24. The at least one computer readable medium of claim 15, wherein the parallel server includes at least one processor in addition to the plurality of processors.

25. The at least one computer readable medium of claim 15, wherein the first routine begins execution before the second routine.

26. The at least one computer readable medium of claim 15, wherein the second routine begins execution before the first routine.

27. The at least one computer readable medium of claim 15, wherein the plurality of processors are abstract processors.

28. The at least one computer readable medium of claim 15, wherein the plurality of processors are physical processors.

29. A computer that executes a computer program for manipulating a data structure stored on a parallel server comprising a plurality of processors, wherein a portion of the data structure is stored on each of the plurality of processors, and wherein the computer comprises:

an output for sending data to the parallel server; and

at least one controller, coupled to the output that:

(a) executes a first routine based on code in the computer program that causes a first request to perform a first operation on the data structure to be sent to the

- 35 -

parallel server, via the output, wherein the first routine is a routine that overloads a serial routine that, when executed, causes the first operation to be performed serially; and

(b) executes a second routine based on code in the computer program that causes a second request to be sent to the parallel server, via the output, wherein the second request includes an instruction to at least one specific processor of the plurality of processors to perform a second operation on the portion of the data structure stored on the at least one specific processor.

30. The computer of claim 29, wherein the computer program is written in an interpreted programming language.

31. The computer of claim 29, wherein the computer program is written in a compiled programming language.

32. The computer of claim 29, wherein the second request specifies a splitting of the data structure on the parallel server.

33. The computer of claim 32, wherein the splitting of the data structure is an abstract splitting.

34. The computer of claim 32, wherein the splitting of the data structure is a physical distribution.

35. The computer of claim 29, wherein the computer program is a first computer program and the second request includes a second computer program that, when executed on the at least one specific processor, performs the second operation on the portion of the data structure stored on the at least one specific processor.

36. The computer of claim 35, wherein the first computer program and the second computer program are written in a same programming language.

- 36 -

37. The computer of claim 35, wherein the first computer program and the second computer program are written in different programming languages.

38. The computer of claim 29, wherein the parallel server includes at least one processor in addition to the plurality of processors.

39. The computer of claim 29, wherein the first routine begins execution before the second routine.

40. The computer of claim 29, wherein the second routine begins execution before the first routine.

41. The computer of claim 29, wherein the plurality of processors are abstract processors.

42. The computer of claim 29, wherein the plurality of processors are physical processors.

43. A method of processing a computer program written in an interpreted programming language that manipulates a data structure stored on a parallel server comprising a plurality of processors, wherein a portion of the data structure is stored on each of the plurality of processors, and wherein the method comprises acts of:

(a) executing a first routine based on code in the computer program that causes a first request to perform a first operation on the data structure to be sent to the parallel server; and

(b) executing a second routine based on code in the computer program that causes a second request to be sent to the parallel server, wherein the second request includes an instruction to at least one specific processor of the plurality of processors to perform a second operation on the portion of the data structure stored on the at least one specific processor.

- 37 -

44. The method of claim 43, wherein the second routine is written in an interpreted programming language.

45. The method of claim 43, wherein the computer program is written in a compiled programming language.

46. The method of claim 43, wherein the second request specifies a splitting of the data structure on the parallel server.

47. The method of claim 46, wherein the splitting of the data structure is an abstract splitting.

48. The method of claim 46, wherein the splitting of the data structure is a physical distribution.

49. The method of claim 43, wherein the computer program is a first computer program and the second request includes a second computer program that, when executed on the at least one specific processor, performs the second operation on the portion of the data structure stored on the at least one specific processor.

50. The method of claim 49, wherein the first computer program and the second computer program are written in a same programming language.

51. The method of claim 49, wherein the first computer program and the second computer program are written in different programming languages.

52. The method of claim 43, wherein the parallel server includes at least one processor in addition to the plurality of processors.

53. The method of claim 43, wherein the first routine begins execution before the second routine.

- 38 -

54. The method of claim 43, wherein the second routine begins execution before the first routine.

55. The method of claim 43, wherein the plurality of processors are abstract processors.

56. The method of claim 43, wherein the plurality of processors are physical processors.

57. At least one computer readable medium encoded with instructions that, when executed on a computer system, perform a method of processing a computer program written in an interpreted programming language that manipulates a data structure stored on a parallel server comprising a plurality of processors, wherein a portion of the data structure is stored on each of the plurality of processors, and wherein the method comprises acts of:

(a) executing a first routine based on code in the computer program that causes a first request to perform a first operation on the data structure to be sent to the parallel server; and

(b) executing a second routine based on code in the computer program that causes a second request to be sent to the parallel server, wherein the second request includes an instruction to at least one specific processor of the plurality of processors to perform a second operation on the portion of the data structure stored on the at least one specific processor.

58. The at least one computer readable medium of claim 57, wherein the second routine is written in an interpreted programming language.

59. The at least one computer readable medium of claim 57, wherein the computer program is written in a compiled programming language.

60. The at least one computer readable medium of claim 57, wherein the second request specifies a splitting of the data structure on the parallel server.

- 39 -

61. The at least one computer readable medium of claim 60, wherein the splitting of the data structure is an abstract splitting.

62. The at least one computer readable medium of claim 60, wherein the splitting of the data structure is a physical distribution.

63. The at least one computer readable medium of claim 57, wherein the computer program is a first computer program and the second request includes a second computer program that, when executed on the at least one specific processor, performs the second operation on the portion of the data structure stored on the at least one specific processor.

64. The at least one computer readable medium of claim 63, wherein the first computer program and the second computer program are written in a same programming language.

65. The at least one computer readable medium of claim 63, wherein the first computer program and the second computer program are written in different programming languages.

66. The at least one computer readable medium of claim 57, wherein the parallel server includes at least one processor in addition to the plurality of processors.

67. The at least one computer readable medium of claim 57, wherein the first routine begins execution before the second routine.

68. The at least one computer readable medium of claim 57, wherein the second routine begins execution before the first routine.

69. The at least one computer readable medium of claim 57, wherein the plurality of processors are abstract processors.

- 40 -

70. The at least one computer readable medium of claim 57, wherein the plurality of processors are physical processors.

71. A computer that executes a computer program written in a interpreted programming language for manipulating a data structure stored on a parallel server comprising a plurality of processors, wherein a portion of the data structure is stored on each of the plurality of processors, and wherein the computer comprises:

an output for sending data to the parallel server; and

at least one controller, coupled to the output that:

(a) executes a first routine based on code in the computer program that causes a first request to perform a first operation on the data structure to be sent to the parallel server, via the output; and

(b) executes a second routine based on code in the computer program that causes a second request to be sent to the parallel server, via the output, wherein the second request includes an instruction to at least one specific processor of the plurality of processors to perform a second operation on the portion of the data structure stored on the at least one specific processor.

72. The computer of claim 71, wherein the second routine is written in an interpreted programming language.

73. The computer of claim 71, wherein the computer program is written in a compiled programming language.

74. The computer of claim 71, wherein the second request specifies a splitting of the data structure on the parallel server.

75. The computer of claim 74, wherein the splitting of the data structure is an abstract splitting.

- 41 -

76. The computer of claim 74, wherein the splitting of the data structure is a physical distribution.

77. The computer of claim 71, wherein the computer program is a first computer program and the second request includes a second computer program that, when executed on the at least one specific processor, performs the second operation on the portion of the data structure stored on the at least one specific processor.

78. The computer of claim 77, wherein the first computer program and the second computer program are written in a same programming language.

79. The computer of claim 77, wherein the first computer program and the second computer program are written in different programming languages.

80. The computer of claim 71, wherein the parallel server includes at least one processor in addition to the plurality of processors.

81. The computer of claim 71, wherein the first routine is executed before the second routine.

82. The computer of claim 71, wherein the second routine is executed before the first routine.

83. The computer of claim 71, wherein the plurality of processors are abstract processors.

84. The computer of claim 71, wherein the plurality of processors are physical processors.

85. A method of processing a computer program that manipulates a data structure stored on a parallel server comprising a plurality of processors, wherein a

- 42 -

portion of the data structure is physically stored on each of the plurality of processors, and wherein the method comprises acts of:

(a) executing a first routine based on code in the computer program that causes a first request to perform a first operation on the data structure to be sent to the parallel server; and

(b) executing a second routine based on code in the computer program that causes a second request to be sent to the parallel server, wherein the second request defines an abstract splitting of the data structure that defines a plurality of abstract portions of the data structure, and wherein the second request includes an instruction to perform a second operation on at least one of the plurality of abstract portions.

86. The method of claim 85, wherein the second routine begins execution before the first routine.

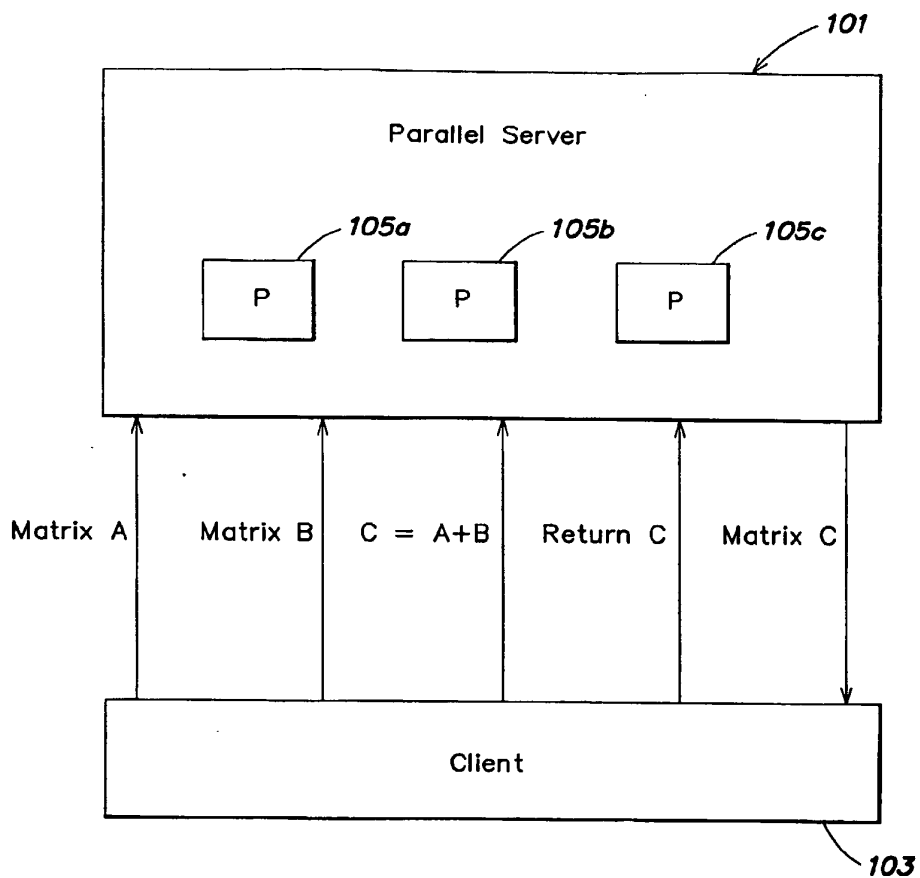
87. The method of claim 85, wherein the parallel server includes at least one processor in addition to the plurality of processors.

89. The method of claim 85, wherein the second operation takes as input the at least one of the plurality of abstract portions.

90. The method of claim 89, further comprising an act of:

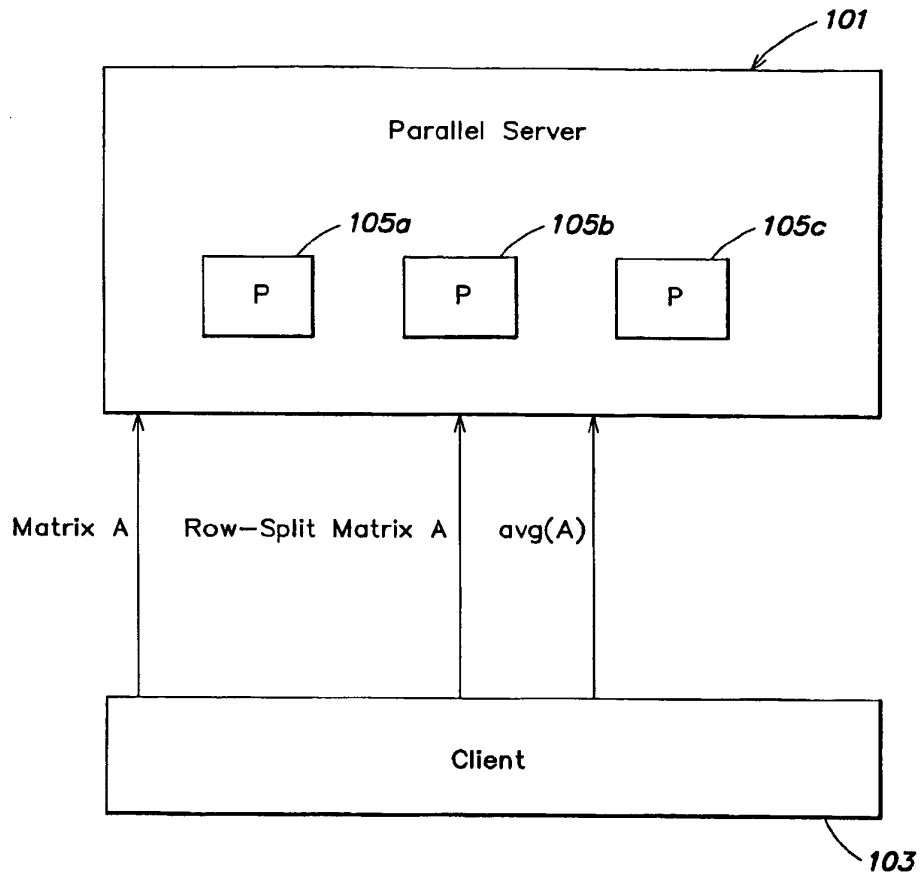
(c) in response to receiving the second request at the parallel server, performing the second operation on the at least one of the plurality of abstract portions without physically distributing the data structure on the parallel server in the manner of the abstract splitting.

1/8

**FIG. 1A**

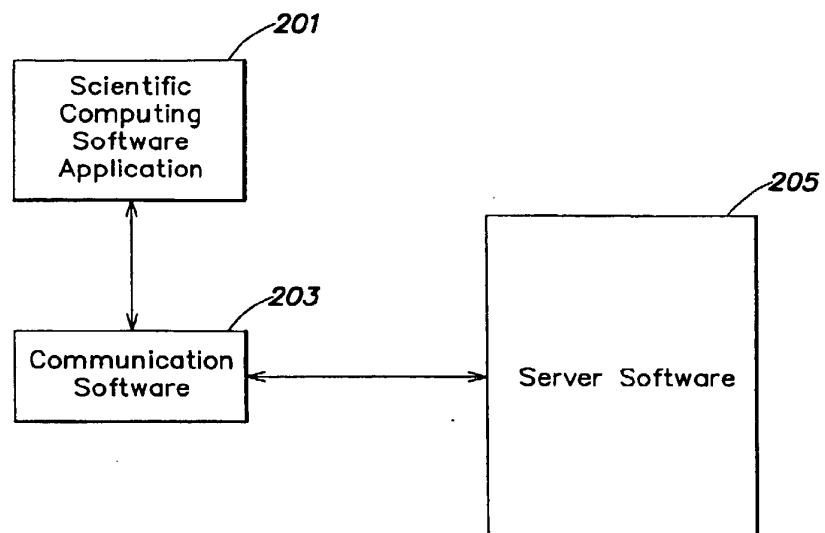
SUBSTITUTE SHEET (RULE 26)

2/8

**FIG. 1B**

SUBSTITUTE SHEET (RULE 26)

3/8

**FIG. 2**

SUBSTITUTE SHEET (RULE 26)

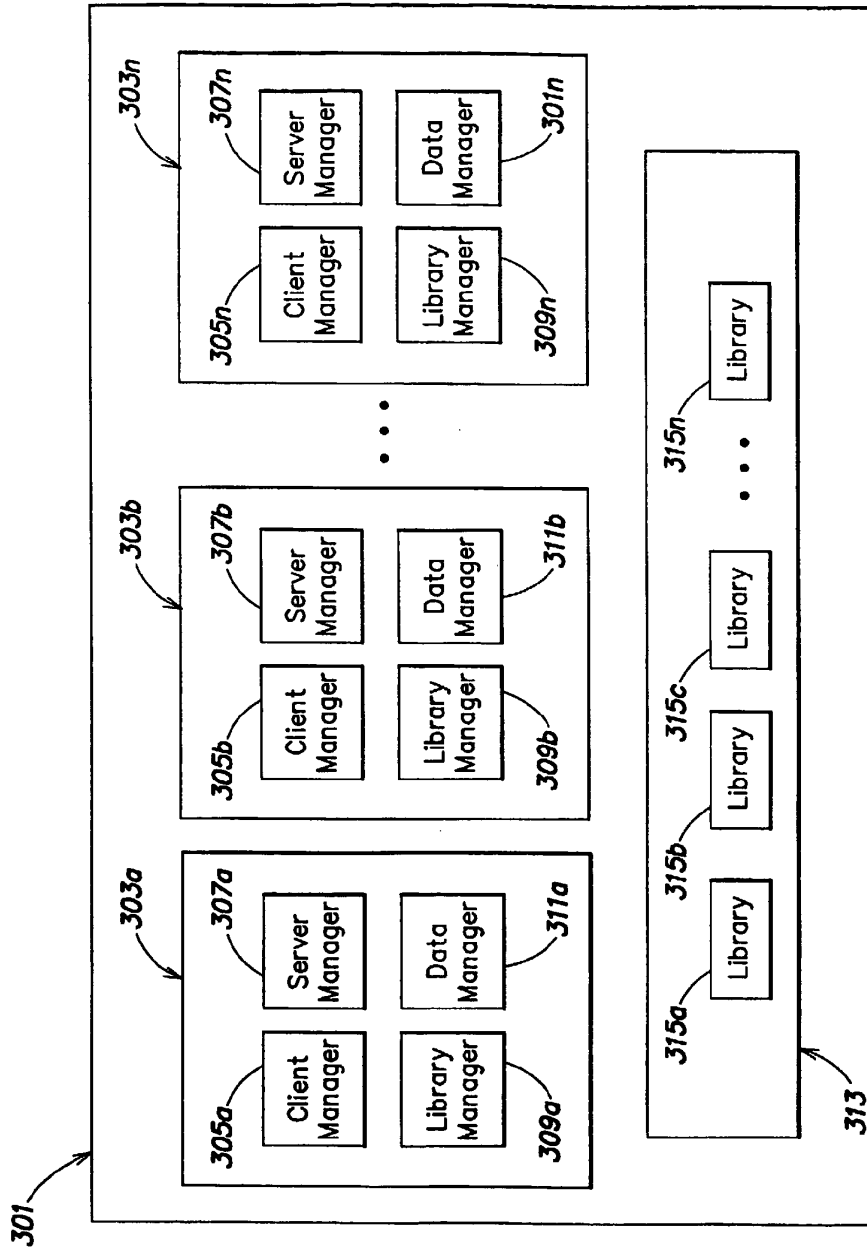


FIG. 3

SUBSTITUTE SHEET (RULE 26)

5/8

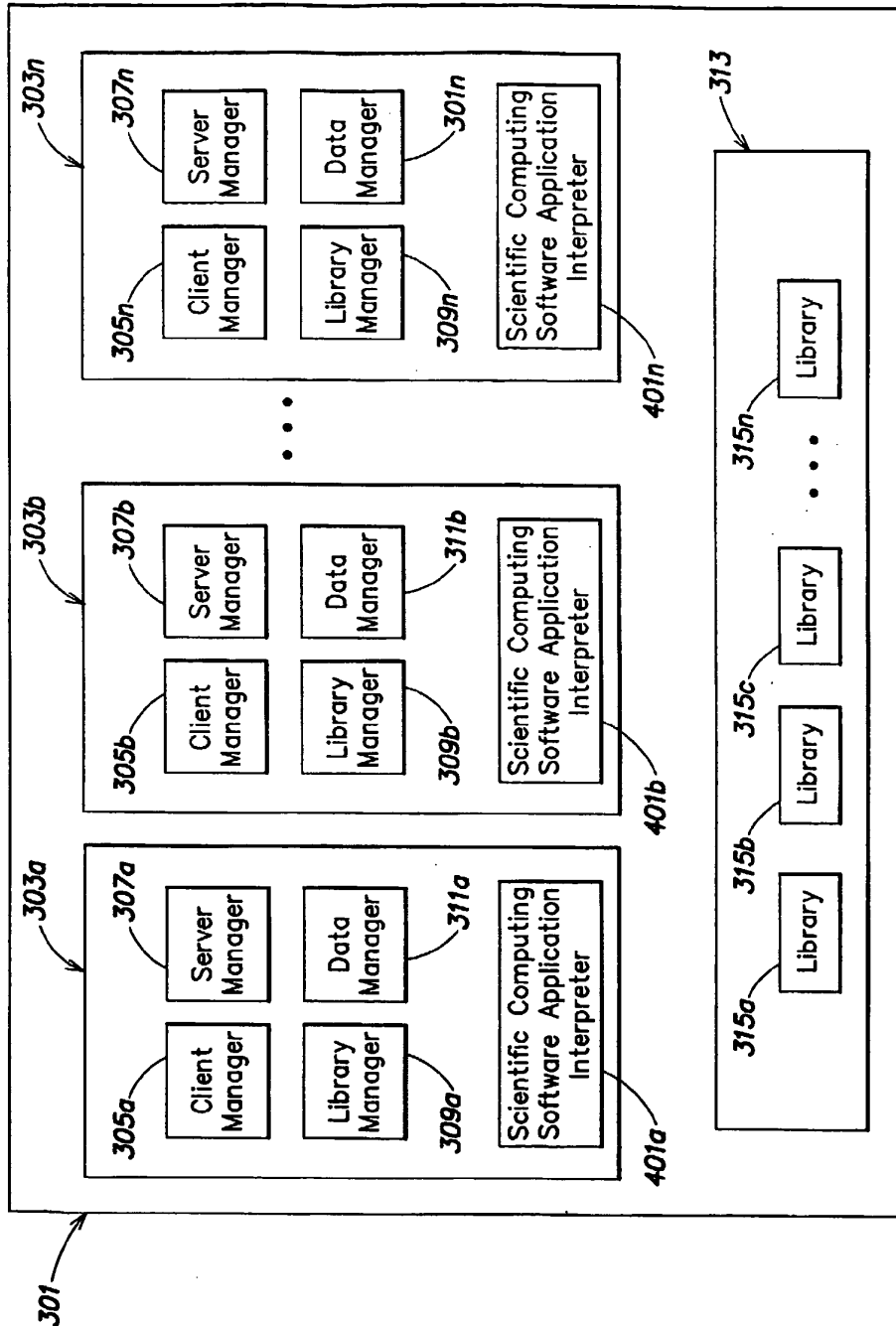
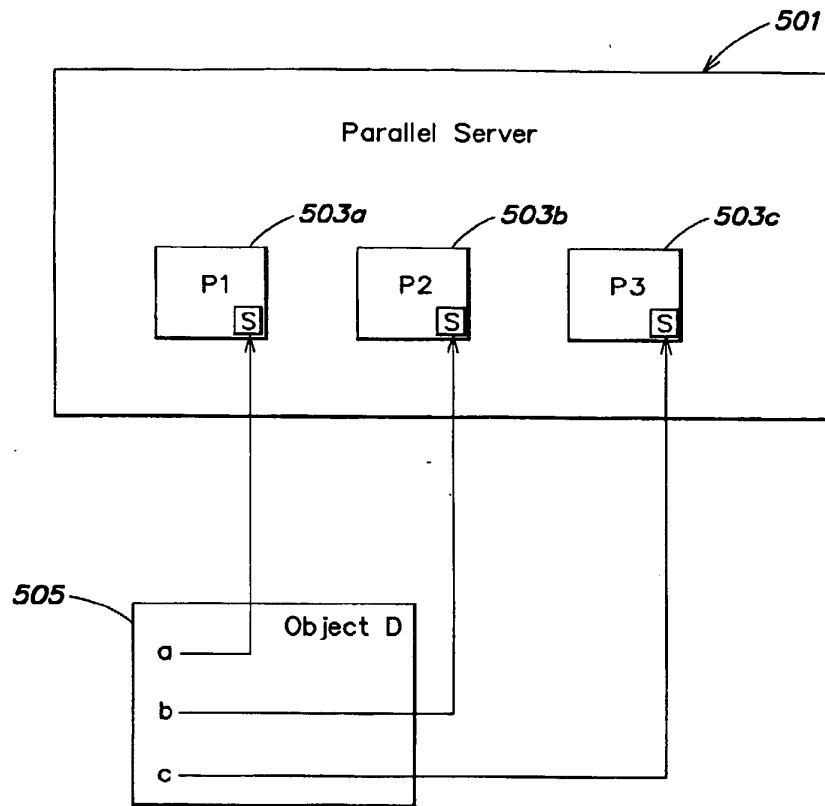


FIG. 4

SUBSTITUTE SHEET (RULE 26)

6/8

**FIG. 5**

SUBSTITUTE SHEET (RULE 26)

7/8

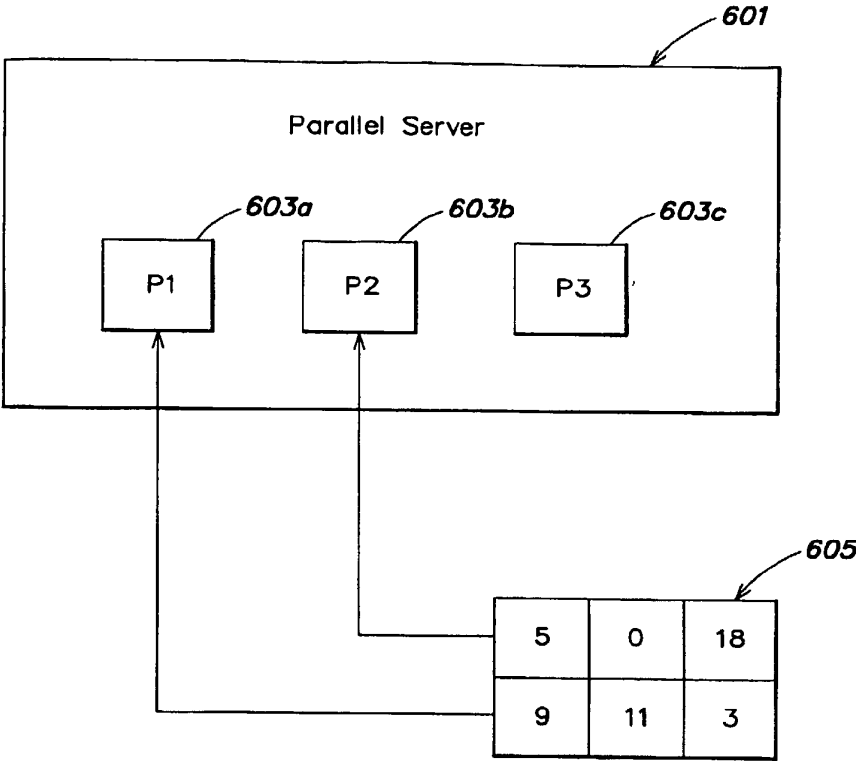


FIG. 6

SUBSTITUTE SHEET (RULE 26)

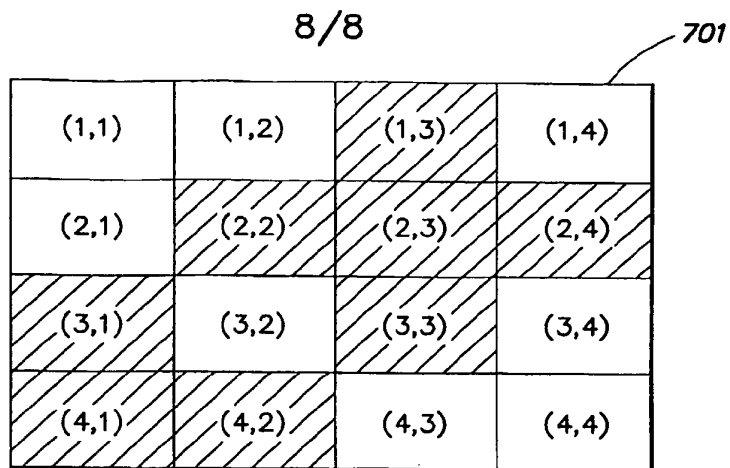


FIG. 7A

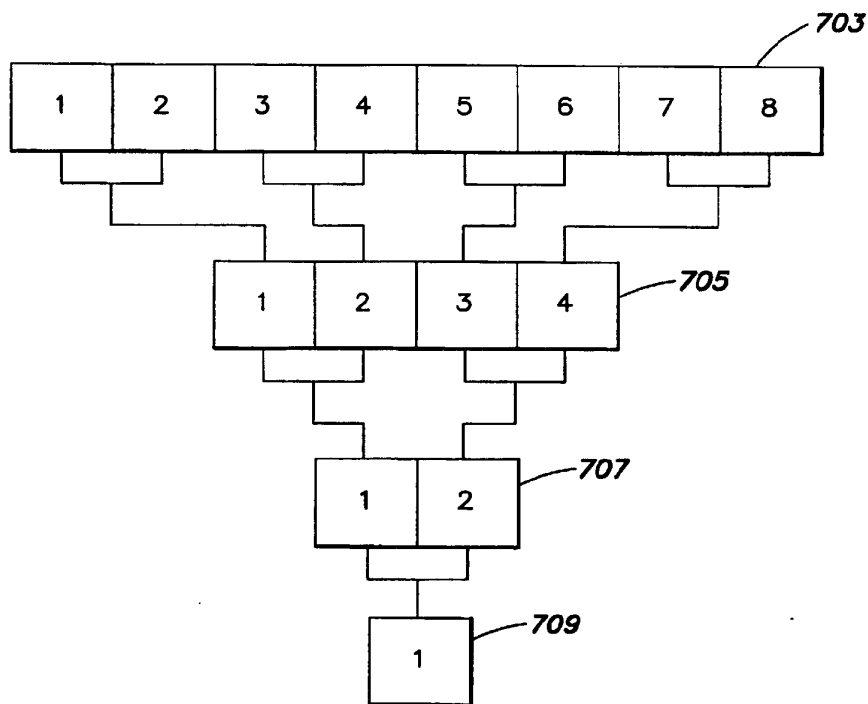


FIG. 7B

SUBSTITUTE SHEET (RULE 26)

INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER G06F9/50		International application No I 'US2005/039580
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used) EPO-Internal, IBM-TDB, INSPEC, COMPENDEX		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	MENON Y ET AL: "MultimATLAB: integrating MATLAB with high-performance parallel computing" PROCEEDINGS OF THE 1997 ACM/IEEE SC97 CONFERENCE ACM NEW YORK, NY, USA, 1997, page 18 pp., XP002373292 ISBN: 0-89791-985-8 the whole document <div style="text-align: center; margin-top: 10px;">----- -/--</div>	1-87,89, 90
☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.		
<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>* Special categories of cited documents:</p> <p>*A* document defining the general state of the art which is not considered to be of particular relevance</p> <p>*E* earlier document but published on or after the international filing date</p> <p>*L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>*O* document referring to an oral disclosure, use, exhibition or other means</p> <p>*P* document published prior to the international filing date but later than the priority date claimed</p> </div> <div style="width: 45%;"> <p>*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>*&* document member of the same patent family</p> </div> </div>		
Date of the actual completion of the international search <div style="text-align: center; font-weight: bold;">22 March 2006</div>		Date of mailing of the international search report <div style="text-align: center; font-weight: bold;">11/04/2006</div>
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016		Authorized officer <div style="text-align: center; font-weight: bold;">Carciofi, A</div>

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2005/039580

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>CHOY R., EDELMAN, A.: "Solving Multiple Classes of Problems in Parallel with MATLAB*P"</p> <p>SINGAPORE-MIT ALLIANCE (SMA) - COMPUTER SCIENCE (CS) COLLECTION, [Online]</p> <p>January 2004 (2004-01), XP002373293</p> <p>Retrieved from the Internet:</p> <p>URL:http://hdl.handle.net/1721.1/3874></p> <p>[retrieved on 2006-03-21]</p> <p>the whole document</p> <p>-----</p>	<p>1-87,89,90</p>
A	<p>HUSBANDS P ET AL: "The parallel problems server: a client-server model for interactive large scale scientific computation"</p> <p>VECTOR AND PARALLEL PROCESSING - VECPAR'98. THIRD INTERNATIONAL CONFERENCE. SELECTED PAPERS AND INVITED TALKS</p> <p>SPRINGER-VERLAG BERLIN, GERMANY, 1999, pages 156-169, XP002373294</p> <p>ISBN: 3-540-66228-6</p> <p>the whole document</p> <p>-----</p>	<p>1-87,89,90</p>

Form PCT/ISA/210 (continuation of second sheet) (April 2005)

NVIDIA-ACS-0233431

Ex.1002 / Page 379 of 616

Intel Corporation et al

Electronic Acknowledgement Receipt	
EFS ID:	45301674
Application Number:	17374864
International Application Number:	
Confirmation Number:	1083
Title of Invention:	CLUSTER COMPUTING
First Named Inventor/Applicant Name:	Zvi Tannenbaum
Customer Number:	20995
Filer:	Jordan Michael Cox/Aimee Kazan
Filer Authorized By:	Jordan Michael Cox
Attorney Docket Number:	ZTANN.002P1C6
Receipt Date:	24-MAR-2022
Filing Date:	13-JUL-2021
Time Stamp:	14:48:55
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	Non Patent Literature	Windows_Compute_Cluster_Server_2003_alld_computing_tools_-_NVIDIA-ACS-0238598.pdf	2184878 8f81a7d8f0e0917400bc0cc7935c1fc2b24391f7	no	6

Warnings:

Information:					
2	Non Patent Literature	Windows_Compute_Cluster_Server_Overview_Sept_2005_-_NVIDIA-ACS-0238610.pdf	654528	no	12
			c9f3366070c98ca06624147dceef627e3fda86d3		
Warnings:					
Information:					
3	Non Patent Literature	Windows_IB_Introduction_to_Windows_200_-_NVIDIA-ACS-0238622.pdf	7094505	no	28
			fb36e753d3435dfb69c31a58e6d348a4cee5a297		
Warnings:					
Information:					
4	Non Patent Literature	Wired_News_That_s_a_Whole_Lot_of_Power_Mac_-_NVIDIA-ACS-0239372.pdf	3311575	no	9
			5eb5090f3bf0e988619231d48600caa4b3e17604		
Warnings:					
Information:					
5	Foreign Reference	WO_2006-050404_A1_to_Husbands_et_al_-_NVIDIA-ACS-0233378.pdf	2210629	no	54
			81428079913db76ecd462c70e9060bb5aa14112		
Warnings:					
Information:					
6	Non Patent Literature	Wolfram_Announcements_Archive_-_Wolfram_-_NVIDIA-ACS-0238999.pdf	469668	no	1
			196507983cfbad7c14ca96f4b075d99d65e1375d		
Warnings:					
Information:					
7	Non Patent Literature	Wolfram_Library_Archive_-_NVIDIA-ACS-0239000.pdf	1223445	no	3
			6dac9b9cf0c09b42601e028e4c22d0323a356e55		
Warnings:					
Information:					
8	Non Patent Literature	Wolfram_Research_-_Mathematica_Parallel_Computing_-_NVIDIA-ACS-0239486.pdf	3896030	no	100
			cf4ad6afe239bc7889065d592be93797392a427e		
Warnings:					
Information:					

9	Non Patent Literature	Cotronis_- _Recent_Advances_in_Parallel _Virtual_Machine_2001_- _Chen_-_NVIDIA- ACS-0170449_Part1.pdf	15667586	no	252
			806cd8762bbb26a4dc67ca2ae25d871815056d30		
Warnings:					
Information:					
10	Non Patent Literature	Cotronis_- _Recent_Advances_in_Parallel _Virtual_Machine_2001_- _Chen_-_NVIDIA- ACS-0170449_Part2.pdf	12536953	no	204
			54a03148d9b2c0899393e544bedd19f2d3f46ab1		
Warnings:					
Information:					
11	Non Patent Literature	Microsoft_aims_to_enter_supe rcomputer_market_- _SC_05_11-15-2005_-_NVIDIA- ACS-0239623_Part1.pdf	24907019	no	63
			6a03b85814fab4b376c17344256cdd604e3d92b1		
Warnings:					
Information:					
12	Non Patent Literature	Microsoft_aims_to_enter_supe rcomputer_market_- _SC_05_11-15-2005_-_NVIDIA- ACS-0239623_Part2.pdf	25162872	no	114
			cc8ed8bad574fa268f2ff0ce7df0263ea9adc0efa		
Warnings:					
Information:					
13	Non Patent Literature	Microsoft_aims_to_enter_supe rcomputer_market_- _SC_05_11-15-2005_-_NVIDIA- ACS-0239623_Part3.pdf	24297311	no	168
			dc84f962243e1a7873c22d78c223d9b1cecd8f107		
Warnings:					
Information:					
14	Non Patent Literature	Microsoft_aims_to_enter_supe rcomputer_market_- _SC_05_11-15-2005_-_NVIDIA- ACS-0239623_Part4.pdf	6891357	no	19
			121dc3b8da5e5491fc81494a6b3aa50c0227183a		
Warnings:					
Information:					
15	Non Patent Literature	SC09_printed_program_- _NVIDIA-ACS-0238414_Part1. pdf	26211875	no	28
			e86b2c3111ea8adcb1cc22d69f3281558087ba54		
Warnings:					
Information:					

16	Non Patent Literature	SC09_printed_program_-_NVIDIA-ACS-0238414_Part2.pdf	26029388	no	45
			778fc78c41a89d77366b1343d14e1f310682031c		
Warnings:					
Information:					
17	Non Patent Literature	SC09_printed_program_-_NVIDIA-ACS-0238414_Part3.pdf	26085118	no	48
			28dcf76aa5631179eb82583a3c064fb305c67dca		
Warnings:					
Information:					
18	Non Patent Literature	SC09_printed_program_-_NVIDIA-ACS-0238414_Part4.pdf	22420286	no	37
			bfdcf0177dfb45d0ec58240abe50756fbad02e9f		
Warnings:					
Information:					
19	Non Patent Literature	Silberschatz_-_Applied_Operating_System_Concepts_1st_Ed_2000_-_NVIDIA-ACS-0172412_Part1.pdf	25150156	no	431
			eb16112f3bb7842d739bd4508af910d49dd59061		
Warnings:					
Information:					
20	Non Patent Literature	Silberschatz_-_Applied_Operating_System_Concepts_1st_Ed_2000_-_NVIDIA-ACS-0172412_Part2.pdf	22083721	no	410
			5fff78021d2445db4e2e95d8fe0ecf714f35de7e		
Warnings:					
Information:					
21	Non Patent Literature	The_Mathematica_Book_5e_-_NVIDIA-ACS-0235750_Part1.pdf	25143942	no	410
			44227625bfacc0f9aced724b72f149e9410982b8		
Warnings:					
Information:					
22	Non Patent Literature	The_Mathematica_Book_5e_-_NVIDIA-ACS-0235750_Part2.pdf	25124702	no	369
			00447aee5ff85057383828e0aad4afeca781dfb4		
Warnings:					
Information:					

23	Non Patent Literature	The_Mathematica_Book_5e_-_NVIDIA-ACS-0235750_Part3.pdf	24785386	no	343
			22a669812550bf0e272f6f62662457aecf87e5dd		
Warnings:					
Information:					
24	Non Patent Literature	The_Mathematica_Book_5e_-_NVIDIA-ACS-0235750_Part4.pdf	24851913	no	65
			3f3afb2c16437ff4c5b3ba4e8b5786698a9228d		
Warnings:					
Information:					
25	Non Patent Literature	The_Mathematica_Book_5e_-_NVIDIA-ACS-0235750_Part5.pdf	25140025	no	66
			7e6b82a639d4490787386727d5c54d49d6ff912d		
Warnings:					
Information:					
26	Non Patent Literature	The_Mathematica_Book_5e_-_NVIDIA-ACS-0235750_Part6.pdf	16443821	no	48
			07721e28169c0581d06bded7f28d72837b8fec35		
Warnings:					
Information:					
27	Non Patent Literature	_CCS_Presentation_-_NVIDIA-ACS-0237381_Part1.pdf	7335660	no	21
			4c049e0d4ef8be2b661f052f9f16c513e7e08d66		
Warnings:					
Information:					
28	Non Patent Literature	CCS_Presentation_-_NVIDIA-ACS-0237381_Part2.pdf	7200497	no	22
			b84aefeb1e52e92be123877c75ea642ee5b2fab		
Warnings:					
Information:					
29	Non Patent Literature	CCS_Presentation_-_NVIDIA-ACS-0237381_Part3.pdf	8427317	no	25
			86e273478f75353a2bbe5c204e9d2a626f94f1d7		
Warnings:					
Information:					

30	Non Patent Literature	CCS_Presentation_-_NVIDIA-ACS-0237381_Part4.pdf	7371554	no	23
			a60604011d36b54433f3c87b2c7a6e7a27cbf9d1		
Warnings:					
Information:					
31	Non Patent Literature	CCS_Presentation_-_NVIDIA-ACS-0237381_Part5.pdf	6131771	no	22
			2b2536d487f9f47820f8b8c1078e94bc11682d1f		
Warnings:					
Information:					
32	Non Patent Literature	CCS_Presentation_-_NVIDIA-ACS-0237381_Part6.pdf	6519554	no	19
			b62927d612fee21ef51d8740c31c0bd4f15bd027		
Warnings:					
Information:					
33	Non Patent Literature	CCS_Presentation_-_NVIDIA-ACS-0237381_Part7.pdf	5883593	no	20
			928efaed4f928d9484d3ab6f29067ac3a143dabd		
Warnings:					
Information:					
34	Non Patent Literature	CCS_Presentation_-_NVIDIA-ACS-0237381_Part8.pdf	2895630	no	9
			17f5b7b5d2f25400bc4f6c3cf91f15bf1f37d76		
Warnings:					
Information:					
35	Non Patent Literature	Wolfram_Technology_Conference_shows_date_of_conference_-_NVIDIA-ACS-0239586.pdf	2374173	no	27
			2988df5b48a2fb10cad178f870c56235a1d5417a		
Warnings:					
Information:					
36	Non Patent Literature	DRACSWolfram06_-_NVIDIA-ACS-0237217_Part1.pdf	2590361	no	14
			95bd88f9a419aa42ea3532d70ef2b1c3b829f990		
Warnings:					
Information:					

37	Non Patent Literature	DRACSWolfram06_-_NVIDIA-ACS-0237217_Part2.pdf	2970228	no	30
			87af73aa5478cfb7049dd507ec3c86328d66d099		
Warnings:					
Information:					
38	Non Patent Literature	DRACSWolfram06_-_NVIDIA-ACS-0237217_Part3.pdf	250962	no	1
			4fad269cbbcc3e48295027f3f231ae086646e2d		
Warnings:					
Information:					
39	Non Patent Literature	2020-10-05_NVIDIA_s_Invalidity_Contentions.pdf	6981672	no	215
			1c60fdd91d54b23d69d5775878ba09b6702bd527		
Warnings:					
Information:					
40	Non Patent Literature	Exhibit_A-1.pdf	11835266	no	211
			3f4027665e39472dfdcf11251662a8e3bcabc9a1		
Warnings:					
Information:					
41	Non Patent Literature	Exhibit_A-2.pdf	4688115	no	171
			9bbcc4879d4a5ae9ce8f6f85f1dd862606c5cb738		
Warnings:					
Information:					
42	Non Patent Literature	Exhibit_A-3.pdf	18758449	no	246
			65b3eaf5d17b81acc76c1f569b351031c162ebb		
Warnings:					
Information:					
43	Non Patent Literature	Exhibit_A-4.pdf	4019163	no	467
			fc2229803e7793f7718f72c98dbd9fe50670f3ad		
Warnings:					
Information:					

44	Non Patent Literature	Exhibit_A-5.pdf	1658326	no	137
			b9d62acc254f962cd783b64257c2772017101f7f		
Warnings:					
Information:					
45	Non Patent Literature	Exhibit_A-6.pdf	5731825	no	268
			0b11e62dd3268d82181c43e4bb15cbac1bee133d		
Warnings:					
Information:					
46	Non Patent Literature	Exhibit_A-7.pdf	8259021	no	127
			f5a685e8580dd3a91c563badf42136f5c990208a		
Warnings:					
Information:					
47	Non Patent Literature	Exhibit_A-9.pdf	17275687	no	211
			8eecd108389659c56a9bb441f2f6c6b07a552a9e		
Warnings:					
Information:					
48	Non Patent Literature	Exhibit_A-10.pdf	5063111	no	170
			5c6dd6a1a14614d60742a5b2401c946f6a2a10b4		
Warnings:					
Information:					
49	Non Patent Literature	Exhibit_A-11.pdf	18757836	no	242
			993678bdce4e0c6e83ed839312ebcb26c104eab1		
Warnings:					
Information:					
50	Non Patent Literature	Exhibit_A-12.pdf	4031208	no	467
			e045c7f81135b67c5396a2c58e8116ec06d2d37f		
Warnings:					
Information:					

51	Non Patent Literature	Exhibit_A-13.pdf	1663083	no	137
			a21dbec4f58c610cc1758b581a81932cb453e93f		
Warnings:					
Information:					
52	Non Patent Literature	Exhibit_A-14.pdf	5742433	no	268
			a568bfc16967ec3e4734404307db868ae256ff14		
Warnings:					
Information:					
53	Non Patent Literature	Exhibit_A-15.pdf	5170447	no	127
			c6aabecbb9e87990de440d05562e7a7c1f24f985		
Warnings:					
Information:					
54	Non Patent Literature	Exhibit_B-1.pdf	13549352	no	167
			9fbadcd4fba040e0c7cb38be5e717cfb3e9b4ff4		
Warnings:					
Information:					
55	Non Patent Literature	Exhibit_B-2.pdf	2929485	no	90
			e4d78d7880cb195fd260cf00d886ffab110ad1eb		
Warnings:					
Information:					
Total Files Size (in bytes):			616044468		

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.

Electronic Acknowledgement Receipt	
EFS ID:	45301566
Application Number:	17374864
International Application Number:	
Confirmation Number:	1083
Title of Invention:	CLUSTER COMPUTING
First Named Inventor/Applicant Name:	Zvi Tannenbaum
Customer Number:	20995
Filer:	Jordan Michael Cox/Aimee Kazan
Filer Authorized By:	Jordan Michael Cox
Attorney Docket Number:	ZTANN.002P1C6
Receipt Date:	24-MAR-2022
Filing Date:	13-JUL-2021
Time Stamp:	14:48:02
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	Non Patent Literature	Declaration_of_Wolfgang_Schr einer_PhD_768_-_NVIDIA- ACS-0170912.pdf	3931226 6e3ca576cca94e6c35c906468af516d495b0 c71a	no	85

Warnings:

Information:					
2	Non Patent Literature	Decyk_- _Advanced_PIC_Technquest_ 03-2005_-_NVIDIA- ACS-0170997.pdf	195109	no	3
			7719521e2a1177105846eba025ba92aacce9e9b7		
Warnings:					
Information:					
3	Non Patent Literature	Decyk_- _Supercomputing_for_the_Ma sses_-_NVIDIA-ACS-0171000. pdf	930093	no	13
			498922ce2f9b2fe52fc783a49dbf6b522e2d1131		
Warnings:					
Information:					
4	Non Patent Literature	Deploying_and_Managing_Mic rosoft_Windows_Compute_Clu ster_Server_2003_-_NVIDIA- ACS-0238305.pdf	1669072	no	10
			184a2e9e7810a50c10bf75261b23ecf8bfafc458		
Warnings:					
Information:					
5	Non Patent Literature	Download_details_Windows_ Server_2003_D_-_NVIDIA- ACS-0238315.pdf	885791	no	3
			efbf2bc23b675cb7942c4e8ccdbf71f04ba05468		
Warnings:					
Information:					
6	Non Patent Literature	DR_s_Pooch_Cluster_Software _For_Mac_OS_X_v10_- _NVIDIA-ACS-0239021.pdf	1240077	no	3
			378947031f2a633b1a3f76f9ec33c6befa8709c3		
Warnings:					
Information:					
7	Non Patent Literature	Edelman_- _Interactive_Supercomputings _Star-P_-_NVIDIA- ACS-0239024.pdf	305452	no	2
			58b1ccaec6b3fd44e966eb19fb67191911b96dbe		
Warnings:					
Information:					
8	Non Patent Literature	Evers_- _Windows_Server_for_superco mputers_delayed_-_NVIDIA- ACS-0239410.pdf	1402874	no	2
			b588f4ab2bf78d7f2e6ebccce3ee8b6a22227906		
Warnings:					
Information:					

9	Non Patent Literature	Fried_0_Gates_touts_Microsoft_s_supercomputing_move_ZDNet_11-15-2005_-_NVIDIA-ACS-0239412.pdf	2075666	no	3
			82776ec65d8afe2e6e4c8b01709dc6a70ce0492c		
Warnings:					
Information:					
10	Non Patent Literature	Frigo_-_The_Fastest_Fourier_Transform_in_the_West_-_NVIDIA-ACS-0165490.pdf	1167889	no	20
			9f7e1d5b46079e312929a672b224db9ee73dc50c		
Warnings:					
Information:					
11	Non Patent Literature	Getting_Started_With_Compute_Cluster_2003_June_2006_-_NVIDIA-ACS-0238321.pdf	6408995	no	72
			dd651ba56bf1e7f2d6cc3c080cd6d9e84d1c69d6		
Warnings:					
Information:					
12	Non Patent Literature	GettingStartedWithStarP_20061031_e_-_NVIDIA-ACS-0233516.pdf	3310947	no	32
			e9a5e5f1dcb2175596681fd024e6e01f5867324c		
Warnings:					
Information:					
13	Non Patent Literature	gridMathematica_Extras_-_NVIDIA-ACS-0239622.pdf	533934	no	1
			d2229b5c5c1bab772754d940ea258b955013a91c		
Warnings:					
Information:					
14	Non Patent Literature	Hemsoth_Startup_Aims_to_Bring_Parallel_Applications_to_the_Masses_-_NVIDIA-ACS-0165918.pdf	287826	no	4
			3ffda22a742d2b9312c1c2da5e6721017b3a5bf0		
Warnings:					
Information:					
15	Non Patent Literature	High-Performance_Computing_with_Mathematica_-_NVIDIA-ACS-0237262.pdf	748502	no	2
			4673e8cba6350531a26892d2a6eb83b1531f55cc		
Warnings:					
Information:					

16	Non Patent Literature	High- Performance_Computing_with _Mathema_-_NVIDIA- ACS-0238393.pdf	808768	no	2
			90da631a94f047e39fb9337badb8a26170f6c8f0		
Warnings:					
Information:					
17	Non Patent Literature	HPCwire_-_Unveils_Software_tha_Parallelizes_Sequential_Apps_11-01-2011_-_NVIDIA-ACS-0239483.pdf	941012	no	1
			584ebce674f903db6b81b2b1e61ed58413733a71		
Warnings:					
Information:					
18	Non Patent Literature	Husbands_-_Interactive_Supercomputing_with_MITMatlab_-_NVIDIA-ACS-0239026.pdf	879311	no	10
			2e32a1ea0a6af44fc45a13ebb0a8947486126c7f		
Warnings:					
Information:					
19	Non Patent Literature	HusbandsThesis_-_NVIDIA-ACS-0233548.pdf	8658079	no	97
			727af9f177468652f8d775dbfc600f4fa6987064		
Warnings:					
Information:					
20	Non Patent Literature	Johnston_-_Gates_-_Microsoft_Ships_Beta_2_of_HPC_s_11-16-2005_-_NVIDIA-ACS-0239415.pdf	289066	no	2
			0351b741bf22ff8ee8ab0bdc2ac6c350d74ed92		
Warnings:					
Information:					
21	Non Patent Literature	Kepner_-_MATLABMPI_2003_-_NVIDIA-ACS-0239036.pdf	800717	no	12
			2d50917a9ac338b5cb9f339199d592ed106e4eae		
Warnings:					
Information:					
22	Non Patent Literature	Khot_-_A_Parallell_Data_Mining_Toolbox_-_NVIDIA-ACS-0239048.pdf	5343164	no	44
			3b882df6b77dd8df7236c3a27a241ad53d156763		
Warnings:					
Information:					

23	Non Patent Literature	Lai_- _Microsoft_offers_Beta_2_of_C ompute_Cluster_Server_11-15- 2005_-_NVIDIA-ACS-0239447. pdf	2295545	no	4
			6116b42096bcb6da3a3583399e70e2da21907acd9		
Warnings:					
Information:					
24	Non Patent Literature	Martens_- _Platform_steps_up_partnershi p_with_Microsoft_11-15-2005_- _NVIDIA-ACS-0239451.pdf	1940966	no	3
			62e0c09079a945e30500419ada591aff1fcbf08a		
Warnings:					
Information:					
25	Non Patent Literature	Microsoft_Keynote_Data_Shee t_-_NVIDIA-ACS-0238402.pdf	1345890	no	2
			7722d8ad49349e10e9d1f4819e1f1eae095cea8		
Warnings:					
Information:					
26	Non Patent Literature	Microsoft_Releases_Windows_ Compute_Clus_-_NVIDIA- ACS-0238395.pdf	1804166	no	6
			731c843f8659d46bcc2c38249d9f574ddbf0ba3		
Warnings:					
Information:					
27	Non Patent Literature	Microsoft_Windows_Compute_ _Cluster_Serve_-_NVIDIA- ACS-0238401.pdf	717924	no	1
			8b4ba438569a7a935b3e8dd20bf55e0ffa259f1		
Warnings:					
Information:					
28	Non Patent Literature	Microsofts_Leadership_in_Distr ibuted_Eng_Builds_NVIDIA- ACS-0238404.pdf	514506	no	2
			2d0eda767eb700adadc651f3ff55f937d6cfc457		
Warnings:					
Information:					
29	Non Patent Literature	Krishnan_- _Windows_Compute_Cluster_ Server_and_Desktop_Clusters _-_NVIDIA-ACS-0239417.pdf	2947715	no	30
			9d0225cfe13f3250e05a66b49e672430452498b5		
Warnings:					
Information:					

30	Non Patent Literature	Nayak_- _Design_and_Charaterization_ of_MATLAB_Functions_- _NVIDIA-ACS-0234790.pdf	5502614	no	98
			72b031bc7ad70c1cda2d74a898218cfa824e78db		
Warnings:					
Information:					
31	Non Patent Literature	Omneta_Technology_- _Pooch_- _NVIDIA- ACS-0239121.pdf	238690	no	1
			9d6d8bd79e2b4b451de51257c46f827c1fe5f700		
Warnings:					
Information:					
32	Non Patent Literature	Parallel_Programming_Software_- _NVIDIA-ACS-0238406.pdf	585627	no	4
			1baa85871b2ca339c7d31e3efc8e0866cd73db3f		
Warnings:					
Information:					
33	Non Patent Literature	Partial_MPI_library_based_on_ Sockets_in_Macintosh_OS_X_C ontinued_- _NVIDIA- ACS-0172347.pdf	1706309	no	65
			3979a4bc1aa8d42c30bff01e59933a07c26d0cc20		
Warnings:					
Information:					
34	Non Patent Literature	Pooch_2001-04-28_- _NVIDIA- ACS-0239122.pdf	148053	no	1
			f74a99e45c5b9bb8242f974ac1f124c660ff3f5a		
Warnings:					
Information:					
35	Non Patent Literature	Pooch_Application_Manual_v1 3_- _NVIDIA-ACS-0239123.pdf	3888303	no	65
			fd8d92501d2d082f74e6d81bc91e664355f27ec0		
Warnings:					
Information:					
36	Non Patent Literature	Pooch_Application_Manual_v1 4_- _NVIDIA-ACS-0239188.pdf	5401395	no	82
			1755b50951dc5cd6ce374977f9c933e7c9a4a63b		
Warnings:					
Information:					

37	Non Patent Literature	Pooch_Application_Manual_v15_-_NVIDIA-ACS-0239270.pdf	7256712	no	98
			1b7f3f2aacadeb4324d87a07a0fbd52c2808acef		
Warnings:					
Information:					
38	Non Patent Literature	quicksilver_software_data_sheet_-_NVIDIA-ACS-0238410.pdf	3140659	no	4
			20dd914e9ba17178611041a032d224a7fbfbaf1e		
Warnings:					
Information:					
39	Non Patent Literature	SC_05_-_Press_Releases_-_NVIDIA-ACS-0239987.pdf	1223336	no	2
			ae41c09ca5c1ffc0c50d3d412b9f8b8dac13be76		
Warnings:					
Information:					
40	Non Patent Literature	SC_05_-_Technical_Papers_-_NVIDIA-ACS-0239989.pdf	4802868	no	5
			2cb5e98d48b9e9c6d70d72e43a99033dc9c8a5b3		
Warnings:					
Information:					
41	Non Patent Literature	SC_05_-_Tutorials_-_NVIDIA-ACS-0239994.pdf	2587416	no	3
			2994e86e4dec567c6efde7e0d5e71861ce7655ab		
Warnings:					
Information:					
42	Non Patent Literature	Schreiner_4_-_NVIDIA-ACS-0237051.pdf	1226192	no	23
			4016292d20e8b7663bd555641adde53068dec50d		
Warnings:					
Information:					
43	Non Patent Literature	Schreiner_5_-_NVIDIA-ACS-0237074.pdf	1099872	no	15
			bc55ae0a6bd4a0c81b52c7bfaad8ccffb694cf60		
Warnings:					
Information:					

44	Non Patent Literature	SEIDEL_httpwwwservernetbseidelappleseedsoftwarehtml_-_NVIDIA-ACS-0240006.pdf	1514484	no	2
			c6e7fc9b59685ce78cc1d43c69132d725328e05e		
Warnings:					
Information:					
45	Non Patent Literature	SONNEVELD_starp_client_installation_-_NVIDIA-ACS-0233645.pdf	2431362	no	30
			7f76c866214ccd06cf05d5560fd1bda9bf45a2d7		
Warnings:					
Information:					
46	Non Patent Literature	SONNEVELD_starp_installation_20070209_-_NVIDIA-ACS-0233675.pdf	608252	no	8
			98b62ac459b8bc2668cdf6ba3966e98caa575ea5		
Warnings:					
Information:					
47	Non Patent Literature	SEM_Claim_Chart_IPR2021-00020_Ex_2031_-_NVIDIA-ACS-0237264.pdf	4826597	no	29
			19bf48bf324712ab616f8d61548733c667391581		
Warnings:					
Information:					
48	Non Patent Literature	Schreiner_6_Distributed_MapReduce_Clemson_2003_Presentation_-_NVIDIA-ACS-0237089.pdf	2611113	no	47
			59caf4a7115c955bfff29bad471230aea6078bd34		
Warnings:					
Information:					
49	Non Patent Literature	Supercomputing_2005_Move_it_to_a_Schedule_-_NVIDIA-ACS-0239998.pdf	1465856	no	2
			832b9309a276af98f74e819b38d5a701577bcd1d1		
Warnings:					
Information:					
50	Non Patent Literature	Supercomputing_Engine_for_Mathematica_-_NVIDIA-ACS-0237307.pdf	1128953	no	4
			e07952e2e08fe558bedb76d4fb7a930593f33972		
Warnings:					
Information:					

51	Non Patent Literature	Update_to_Pooch_Software_Enhances_the_Cluster_Experience_-_NVIDIA-ACS-0239368.pdf	1377235	no	4
			67140265126081b75ee135e981d36e108302ab3c		
Warnings:					
Information:					
52	Non Patent Literature	Using_Microsoft_Message_Passing_Interface_-_NVIDIA-ACS-0238572.pdf	1461965	no	18
			d9f7a437d3be08dbe13c1db572c2f513495157eb		
Warnings:					
Information:					
53	Non Patent Literature	Using_Windows_Compute_Cluster_Server_2003_Job_Scheduler_June_2006_-_NVIDIA-ACS-0238590.pdf	772020	no	8
			589782720afb86f82b4897b86828f620a44df56e		
Warnings:					
Information:					
54	Non Patent Literature	Supercomputing_2005_Floor_Map_-_NVIDIA-ACS-0239997.pdf	237848	no	1
			77c1e9cc656b6826e351ffd4b023b32b7458bf2a		
Warnings:					
Information:					
Total Files Size (in bytes):			111624013		
<p>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.</p> <p><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.</p> <p><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.</p> <p><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.</p>					

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2449
SHEET 1 OF 12	Examiner	Astex, Hemmon
	Attorney Docket No.	ZTANN.002PIC6

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	7,542,981	06-02-2009	Choy et al.	
	2	7,627,744	12-01-2009	Maher et al.	
	3	2005/0251644	11-10-2005	Maher et al.	

FOREIGN PATENT DOCUMENTS						
Examiner Initials	Cite No.	Foreign Patent Document Country Code-Number-Kind Code Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T ¹
	4	WO 2006/050404	05-11-2006	Interactive Supercomputing, Inc.		

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	5	"128-Xserve G5 CLUSTER ACHIEVES 1.21 TeraFlop", January 14, 2005, https://www.hpcwire.com/2005/01/14/128-xserve-g5-cluster-achieves-1-21-teraflop/ , pp. 2. [NVIDIA-ACS-0239003 - NVIDIA-ACS-0239004]	
	6	Advanced Cluster Systems, LLC, "Math Supercomputer-in-A-Box", 2007, pp. 1. [NVIDIA-ACS-0237306]	
	7	BEKKER, Scott, "HPC Edition of Windows Server 2003 on Tap for 2005", ENT News, June 23, 2004, pp. 3. [NVIDIA-ACS-0238318 - NVIDIA-ACS-0238320]	
	8	BEKKER, Scott, "Microsoft Enterprise Product Roadmap for 2005", Redmond Channel Partner, January 10, 2005, pp. 14. https://rcpmag.com/articles/2005/01/10/microsoft-enterprise-product-roadmap-for-2005.aspx [NVIDIA-ACS-0239391 - NVIDIA-ACS-0239404]	
	9	BEKKER, Scott, "Muglia Updates Windows Server Roadmap", MCP Mag, December 10, 2004, pp. 5. https://rcpmag.com/articles/2004/12/10/muglia-updates-windows-server-roadmap.aspx [NVIDIA-ACS-0239405 - NVIDIA-ACS-0239409]	
	10	BERNARDIN, Laurent, "Maple on the Intel Paragon", October 15, 1998, pp. 7. https://www.research-collection.ethz.ch/handle/20.500.11850/68829 [NVIDIA-ACS-0172328 - NVIDIA-ACS-0172334]	
	11	CCS Sample RIS Scripts, 2003, pp. 55. [NVIDIA-ACS-0237326 - NVIDIA-ACS-0237380]	
	12	CHOY, Long Yin, "MATLAB [®] P 2.0: Interactive Supercomputing Made Practical", Thesis, MIT, September 2002, pp. 67. [NVIDIA-ACS-0233448 - NVIDIA-ACS-0233514] and [NVIDIA-ACS-0233683 - NVIDIA-ACS-0233749]	
	13	CHOY, Long Yin, "Chapter 1", Draft, MIT, September 2002, pp. 16. [NVIDIA-ACS-0233432 - NVIDIA-ACS-0233447]	
	14	CHINCHILLA et al., "Parallel N-Body Simulation using GPUs", Technical Report TR04-032, December, 2004, pp. 8. [NVIDIA-ACS-0165482 - NVIDIA-ACS-0165489]	

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

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2449
SHEET 2 OF 12	Examiner	Astex, Hemmon
	Attorney Docket No.	ZTANN.002PIC6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	15	"Compute Cluster Job Aids", Windows Server 2003 Deployment Kit, 2003, pp. 80. [NVIDIA-ACS-0238212 - NVIDIA-ACS-0238291]	
	16	"Compute Cluster Server: Getting Started", 2006, pp. 1. [NVIDIA-ACS-0238300]	
	17	"Compute Cluster Server Glossary", December 5, 2005, pp. 6. [NVIDIA-ACS-0238292 - NVIDIA-ACS-0238297]	
	18	"Compute Cluster Server Setup Command-Line Interface Reference", June 6, 2006, pp. 2. [NVIDIA-ACS-0238298 - NVIDIA-ACS-0238299]	
	19	"Compute Cluster Server: Technical Reference", 2007, pp. 2. [NVIDIA-ACS-0238301 - NVIDIA-ACS-0238302]	
	20	COTRONIS et al., "Recent Advances in Parallel Virtual Machine and Message Passing Interface", 8th European PVM/MPI Users' Group Meeting, Santorini/Thera, Greece, September 2001, Proceedings, pp. 456. [NVIDIA-ACS-0170449 - NVIDIA-ACS-0170904] (In 2 parts)	
	21	COWNIE et al., "A Standard Interface for Debugger Access to Message Queue Information in MPI", Conference Paper, September 1999, pp. 9. [NVIDIA-ACS-0232908 - NVIDIA-ACS-0232916]	
	22	DAUGER, Dean, "DR's Pooch Cluster Software for MAC OS X v10.4 'Tiger'", https://www.prweb.com/releases/2005/04/prweb234831.htm , April 29, 2005, pp. 3. [NVIDIA-ACS-0239021 - NVIDIA-ACS-0239023]	
	23	DAUGER, Dean, "Pooch Application", Pooch Quick Start, Parallel Operation and Control Heuristic Application, Version 1.2.1, 2002, pp. 61. [DR_NVIDIA_000006 - DR_NVIDIA_000066]	
	24	DAUGER, Dean, "Pooch Application", Pooch Quick Start, Parallel Operation and Control Heuristic Application, Version 1.4, 2004, pp. 82. [DR_NVIDIA_000160 - DR_NVIDIA_000241]	
	25	DAUGER, Dean, "Pooch Application", Parallel Operation and Control Heuristic Application, Version 1.3.5, 2003, pp. 67. [DR_NVIDIA_000400 - DR_NVIDIA_000466]	
	26	DAUGER, Dean, "The Launch Den Mother and Launch Puppy", Advanced Information, 2000, pp. 10. [DR_NVIDIA_000713 - DR_NVIDIA_000722] & [NVIDIA-ACS-0239381 - NVIDIA-ACS-0239390]	
	27	DAUGER, Dean, "The Launch Den Mother and Launch Puppy", Read Me, 1999, pp. 10. [DR_NVIDIA_000703 - DR_NVIDIA_000712]	
	28	DAUGER et al., "Numerically-Intensive 'Plug-and-Play' Parallel Computing", Proceedings. 2001 IEEE Cluster, pp. 8. [ACS_NVIDIA_005621 - ACS_NVIDIA_005628]	
	29	DAUGER et al., "Numerically-Intensive 'Plug-and-Play' Parallel Computing", Proceedings. 2001 IEEE Cluster, pp. 8. [ACS_NVIDIA_005654 - ACS_NVIDIA_005661]	
	30	DAUGER, Dean E., "Plug-and-Play Cluster Computing Brings HPC to the Mainstream", 2007, pp. 1. [NVIDIA-ACS-0239454]	
	31	DAUGER et al., "Plug-and-Play Cluster Computing: High-Performance Computing for the Mainstream", Computing in Science & Engineering, January /February 2005, pp. 22-28. [ACS_NVIDIA_005607 - ACS_NVIDIA_005613]	
	32	DAUGER et al., "Plug-and-Play Cluster Computing: HPC Designed for the Mainstream Scientist", ICCS 2005, pp. 7. [ACS_NVIDIA_005614 - ACS_NVIDIA_005620] & [NVIDIA-ACS-0170905 - NVIDIA-ACS-0170911]	

Examiner Signature	Date Considered
<p>*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.</p>	

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2449
SHEET 3 OF 12	Examiner	Astex, Hemmon
	Attorney Docket No.	ZTANN.002PIC6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	33	Dauger Research, Computing in Science & Engineering, Excerpt, March/April 2005, pp. 2. [ACS_NVIDIA_005634 - ACS_NVIDIA_005635]	
	34	Dauger Research, "Keeping America Strong Award", 2005, pp. 3. [NVIDIA-ACS-0239455 - NVIDIA-ACS-0239457]	
	35	Dauger Research, "Parallel Zoology", Dauger Research Vault, 2006, pp. 7. [NVIDIA-ACS-0237202 - NVIDIA-ACS-0237208]	
	36	Dauger Research, "Parallelization", Dauger Research Vault, 2006, pp. 3. [NVIDIA-ACS-0237209 - NVIDIA-ACS-0237211]	
	37	Dauger Research, "Parallelization - Parallel Adder", Dauger Research Vault, 2006, pp. 7. [NVIDIA-ACS-0237154 - NVIDIA-ACS-0237160]	
	38	Dauger Research, "Parallelization - Parallel Circle Pi", Dauger Research Vault, 2006, pp. 13. [NVIDIA-ACS-0237161 - NVIDIA-ACS-0237173]	
	39	Dauger Research, "Parallelization - Parallel Knock", Dauger Research Vault, 2006, pp. 4. [NVIDIA-ACS-0237174 - NVIDIA-ACS-0237177]	
	40	Dauger Research, "Parallelization - Parallel Life", Dauger Research Vault, 2006, pp. 14. [NVIDIA-ACS-0237178 - NVIDIA-ACS-0237191]	
	41	Dauger Research, "Parallelization - Parallel Pascal's Triangle", Dauger Research Vault, 2006, pp. 10. [NVIDIA-ACS-0237192 - NVIDIA-ACS-0237201]	
	42	Dauger Research, "Pooch Application: Parallel Operation and Control Heuristic Application", Dauger Research Vault, April 28, 2001, pp. 1. [NVIDIA-ACS-0239122]	
	43	Dauger Research, "Pooch Quick Start", Pooch Application, Parallel Operation and Control Heuristic Application, Version 1.3, 2002, pp. 65. [NVIDIA-ACS-0239123 - NVIDIA-ACS-0239187]	
	44	Dauger Research, "Pooch Quick Start", Pooch Application, Parallel Operation and Control Heuristic Application, Version 1.4, 2004, pp. 62. [NVIDIA-ACS-0239188 - NVIDIA-ACS-0239269]	
	45	Dauger Research, "Pooch Quick Start", Pooch Application, Parallel Operation and Control Heuristic Application, Version 1.5, 2004, pp. 98. [NVIDIA-ACS-0239270 - NVIDIA-ACS-0239367]	
	46	Dauger Research, "Plug-and-Play" Clustering, Build your cluster <i>in minutes!</i> , Apple HPC Seminar Series, 2002, pp. 66. [ACS_NVIDIA_001238 - ACS_NVIDIA_001303]	
	47	Dauger Research, "Pooch Selected for Omneta Supercomputer", Paris, France, Apple Expo, September 21, 2005, pp. 1. [NVIDIA-ACS-0239020]	
	48	Dauger Research, "Pooch, Build Your Personal Computing Cluster <i>in minutes!</i> ", 2004, pp. 4. [DR_NVIDIA_000002 - DR_NVIDIA_000006]	
	49	Dauger Research, "Pooch", Supercomputer-Style Cluster Computing Support for Mathematica, 2006, pp. 4. [NVIDIA-ACS-0237307 - NVIDIA-ACS-0237310]	
	50	Dauger Research, "The PoochMPI Toolkit", High-Performance Computing with Mathematica, Wolfram Technology Conference 2006, Champaign, IL, pp. 45. [NVIDIA-ACS-0237217 - NVIDIA-ACS-0237261] (in 3 parts)	
	51	Dauger Research, "Visualizing Message Passing with MacMPI", Dauger Research Vault, 2006, pp. 5. [NVIDIA-ACS-0237212 - NVIDIA-ACS-0237216]	

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

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2449
	Examiner	Astex, Hemmon
SHEET 4 OF 12	Attorney Docket No.	ZTANN.002PIC6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	52	Dauger Research, "What's New", SET and SET/SciLab debut at Supercomputing 2012, 2012, pp. 25. [NVIDIA-ACS-0239458 - NVIDIA-ACS-0239482]	
	53	Declaration of Wolfgang Schreiner, Ph.D., for Inter Partes Review of U.S. Patent No. 10,333,768, dated October 1, 2020 in 85 pages. [NVIDIA-ACS-0170912 - NVIDIA-ACS-0170996]	
	54	DECYK, Viktor K., "Advanced PIC Techniques: Parallelization and Object-Oriented Programming", Proceedings of ISSS-7, March 26-31, 2005, pp. 3. [NVIDIA-ACS-0170997 - NVIDIA-ACS-0170999]	
	55	DECYK et al., "AppleSeed: A Parallel Macintosh Cluster for Numerically Intensive Computing", January 2001, pp. 8. [ACS_NVIDIA_005692 - ACS_NVIDIA_005699]	
	56	DECYK et al., "AppleSeed: A Parallel Macintosh Cluster for Numerically Intensive Computing", January 2001, pp. 6. [ACS_NVIDIA_005700 - ACS_NVIDIA_005705]	
	57	DECYK et al., "AppleSeed: A Parallel Macintosh Cluster for Scientific Computing", 2003, pp. 16. [ACS_NVIDIA_005638 - ACS_NVIDIA_005653]	
	58	DECYK et al., "How to Build an AppleSeed: A Parallel Macintosh Cluster for Numerically Intensive Computing", January 2001, pp. 11. [ACS_NVIDIA_005680 - ACS_NVIDIA_005690]	
	59	DECYK et al., "Supercomputing for the Masses: A Parallel Macintosh Cluster", 2002, pp. 10-22. [NVIDIA-ACS-0171000 - NVIDIA-ACS-0171012]	
	60	"Deploying and Managing Microsoft Windows Compute Cluster Server 2003", Microsoft, June 6, 2006, pp. 10. [NVIDIA-ACS-0238305 - NVIDIA-ACS-0238314]	
	61	EDELMAN et al., "Interactive Supercomputing's Star-P Platform: Parallel MATLAB and MPI Homework Classroom Study on High Level Language Productivity", 2006, pp. 2. [NVIDIA-ACS-0239024 - NVIDIA-ACS-0239025]	
	62	EVERS, Joris, "Windows Server for Supercomputers Delayed", Network World, April 6, 2005, pp. 2. https://www.computerworld.com/article/2556337/windows-server-for-supercomputers-delayed.html [NVIDIA-ACS-0239410 - NVIDIA-ACS-0239411]	
	63	FERRON, Jay, "Microsoft Compute Cluster Server Partner Training", Windows Compute Cluster Server 2003, 2006, pp. 161. [NVIDIA-ACS-0237381 - NVIDIA-ACS-0237541] (in 8 parts)	
	64	FRIED, Ina, "Gates Touts Microsoft's Supercomputing Move", ZD Net, November 15, 2005, pp. 3. [NVIDIA-ACS-0239412 - NVIDIA-ACS-0239414]	
	65	FRIGO et al., "The Fastest Fourier Transform in the West", Technical Report, MIT, September 11, 1997, pp. 20. [NVIDIA-ACS-0165490 - NVIDIA-ACS-0165509]	
	66	"Getting Started with Compute Cluster 2003", Chapter 2: Designing Unattended Installations, June 2006, pp. 72. https://download.microsoft.com/download/e/2/b/e2bfb017-8525-4991-bbd5-7d7081f3d228/04_CHAPTER_2_Designing_Unattended_Installations.doc [NVIDIA-ACS-0238321 - NVIDIA-ACS-0238392]	
	67	"Getting Started with Star-P Taking Your First Test-Drive", Interactive Supercomputing, Inc., 2006, pp. 32. [NVIDIA-ACS-0233516 - NVIDIA-ACS-0233547]	
	68	"gridMathematica 2: Grid Mathematica Extras", Wolfram Research, November 9, 2006, in 1 page. https://web.archive.org/web/20061109144357/http://www.wolfram.com/products/gridmathematica/extras.html [NVIDIA-ACS-0239622]	

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

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2449
	Examiner	Astex, Hemmon
SHEET 5 OF 12	Attorney Docket No.	ZTANN.002PIC6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	69	GRUBER et al., "Choosing an Automated Installation Method", Chapter 1, Microsoft Windows Server 2003 Deployment Kit, Automating and Customizing Installations, 2003, pp. 18. [NVIDIA-ACS-0238572 - NVIDIA-ACS-0238589]	
	70	GRUBER et al., "Introduction", Microsoft Windows Server 2003 Deployment Kit, Automating and Customizing Installations, 2003, pp. 8. [NVIDIA-ACS-0238590 - NVIDIA-ACS-0238597]	
	71	HEMSOOTH, Nicole, "Startup Aims to Bring Parallel Applications to the Masses", HPC Wire, November 16, 2011, pp. 4. [NVIDIA-ACS-0165918 - NVIDIA-ACS-0165921]	
	72	HPC Wire, "Advanced Cluster Systems Unveils Software that Parallelizes Sequential Apps", November 1, 2011, pp. 1. [NVIDIA-ACS-0239483]	
	73	HUSBANDS, Parry, "Interactive Supercomputing", Thesis, MIT, February 1999, pp. 97. http://dspace.mit.edu/bitstream/handle/1721.1/78973/42430246-MIT.pdf;sequence=2 [NVIDIA-ACS-0233648 - NVIDIA-ACS-0233644]	
	74	HUSBANDS, Parry, "Interactive Supercomputing with MITMatlab", August 2001, pp. 10. https://math.mit.edu/~edelman/publications/interactive_supercomputing.pdf [NVIDIA-ACS-0239026 - NVIDIA-ACS-0239035]	
	75	JOHNSTON, Stuart J., "Gates: Microsoft Ships Beta 2 of HPC Cluster Version of Windows", MCP Mag, November 16, 2005, pp. 2. [NVIDIA-ACS-0239415 - NVIDIA-ACS-00239416]	
	76	KAHNEY, Leander, "That's a Whole Lot of Power, Mac", Wired, January 29, 2002, pp. 9. [NVIDIA-ACS-0239372 - NVIDIA-ACS-0239380]	
	77	KEPNER et al., "MatlabMPI", January 3, 2003, pp. 12. https://arxiv.org/pdf/astro-ph/0305090.pdf [NVIDIA-ACS-0239036 - NVIDIA-ACS-0239047]	
	78	KHOT et al., "A Parallel Data Mining Toolbox Using MatlabMPI", August 2004, pp. 44. [NVIDIA-ACS-0239048 - NVIDIA-ACS-0239091]	
	79	KRISHNAN et al., "Windows Compute Cluster Server and Desktop Clusters: An Overview", Microsoft WinHEC 2006, pp. 30. [NVIDIA-ACS-0239417 - NVIDIA-ACS-0239446]	
	80	LAJ, Eric, "Microsoft Offers Beta 2 of Compute Cluster Server 2003", Computerworld, November 15, 2005, pp. 4. https://www.computerworld.com/article/2559328/microsoft-offers-beta-2-of-compute-cluster-server-2003.html [NVIDIA-ACS-0239447 - NVIDIA-ACS-0239450]	
	81	LANTZ, Eric, "Introduction to Windows 2003 Compute Cluster Edition", Datacenter Fabric Workshop Windows IB, August 22, 2005, pp. 28. [NVIDIA-ACS-0238622 - NVIDIA-ACS-0238649]	
	82	MARTENS, China, "Platform Steps Up Partnership with Microsoft", InfoWorld, November 15, 2005, pp. 3. https://www.infoworld.com/article/2673914/platform-steps-up-partnership-with-microsoft.html [NVIDIA-ACS-0239451 - NVIDIA-ACS-0239453]	
	83	Microsoft, "Accelerating Scientific Discovery with Microsoft Windows Compute Cluster Server 2003", 2003, pp. 10. http://download.microsoft.com/download/8/6/1/86149563-03ec-427e-80f1-d683bd715b26/Bio%20Team%20and%20Life%20Sciences%20Solution%20Brief.pdf [NVIDIA-ACS-0237316 - NVIDIA-ACS-0237325]	
	84	"Microsoft Aims to Enter Supercomputer Market", Reuters, November 15, 2005, pp. 364. [NVIDIA-ACS-0239623 - NVIDIA-ACS-0239986] (In 4 parts)	
	85	Microsoft, "Bill Gates Shares Microsoft's Vision for Technical Computing", November 15, 2005, pp. 5. https://news.microsoft.com/2005/11/15/bill-gates-shares-microsofts-vision-for-technical-computing/ [NVIDIA-ACS-0237311 - NVIDIA-ACS-0237315]	

Examiner Signature	Date Considered
<p>*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.</p>	

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2449
	Examiner	Astex, Hemmon
SHEET 6 OF 12	Attorney Docket No.	ZTANN.002PIC6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	86	Microsoft, "Cluster Components and Interconnects", Microsoft Windows 2000, January 5, 2001, pp. 3. [NVIDIA-ACS-0237553 - NVIDIA-ACS-0237555]	
	87	Microsoft, "Increasing the Value of Simulation While Shrinking the Costs", Microsoft Windows Compute Cluster Server 2003 Partner Solution Brief, March 2007, pp. 6. [NVIDIA-ACS-0237556 - NVIDIA-ACS-0237561]	
	88	Microsoft, "On-Demand Supercomputing Multiplies the Possibilities", Microsoft Windows Compute Cluster Server 2003 Partner Solution Brief, April 2007, pp. 7. [NVIDIA-ACS-0237542 - NVIDIA-ACS-0237548]	
	89	Microsoft, "Microsoft Releases Windows Compute Cluster Server 2003, Bringing High-Performance Computing to the Mainstream", Microsoft.com, June 9, 2006, pp. 6. [NVIDIA-ACS-0238395 - NVIDIA-ACS-0238400]	
	90	Microsoft, "Parallel Programming Software", Microsoft.com, January 26, 2001, pp. 4. [NVIDIA-ACS-0238406 - NVIDIA-ACS-0238409]	
	91	Microsoft, "Technical Computing Tools Combined with Cluster Computing Deliver High-Performance Solutions", Microsoft Windows Compute Cluster Server 2003 Partner Solution Brief, May 2007, pp. 6. [NVIDIA-ACS-0238598 - NVIDIA-ACS-0238603] (In 2 parts)	
	92	Microsoft, "Windows Compute Cluster Server 2003", Overview, September 2005, pp. 12. [NVIDIA-ACS-0238610 - NVIDIA-ACS-0238621]	
	93	Microsoft, "Windows Server 2003 Deployment Kit, Automating and Customizing Installations", October 19, 2005, pp. 3. [NVIDIA-ACS-0238315 - NVIDIA-ACS-0238317]	
	94	"Microsoft Windows Server 2003 Deployment Kit, Automating and Customizing Installations, A Resource Kit Publication", 2003, pp. 359. [NVIDIA-ACS-0237567 - NVIDIA-ACS-0237925]	
	95	"Microsoft Windows Server 2003 Deployment Kit, Automating and Customizing Installations, A Resource Kit Publication", 2003, pp. 286. [NVIDIA-ACS-0237926 - NVIDIA-ACS-0238211]	
	96	MOORE, Charles, "Apple Ends DIY Repairs, Does Mac mini Kill the Used Mac Market?, USB Drive Kit Matches Mac mini, and More, May 13, 2005, pp. 15. https://lowendmac.com/newsrev/05/0513.html [NVIDIA-ACS-0239005 - NVIDIA-ACS-0239019]	
	97	NAYAK, Anshuman, "Design and Characterization of MATLAB Functions on a Distributed and Heterogeneous System", Center for Parallel and Distributed Computing, Technical Report No. CPDC-TR-9912-021, December 1999, pp. 98. [NVIDIA-ACS-0234790 - NVIDIA-ACS-0234887]	
	98	"Omnet Technology: Pooch", Omnet PLC, October 30, 2005, pp. 1. [NVIDIA-ACS-0239121]	
	99	Partial MPI Library based on the program-to-Program Communications ToolBox in the Macintosh OS, November 8, 2000, pp. 65. [NVIDIA-ACS-0172347 - NVIDIA-ACS-0172411]	
	100	PELIKÁN, Josef, "Interconnect Efficiency of Tyan PSC T-630 with Microsoft Compute Cluster Server 2003", pp. 1-5 [NVIDIA-ACS-0237562 - NVIDIA-ACS-0237566]	
	101	"QuickSilver Software Architecture", Software, Data Sheet, November 2005, pp. 4. [NVIDIA-ACS-0238410 - NVIDIA-ACS-0238413]	
	102	SC 05, Seattle, WA, 2005, Press Releases, "Microsoft Chairman and Chief Software Architect Bill Gates to Deliver Keynote at SC05 Conference", July 25, 2005, pp. 2. [NVIDIA-ACS-0239987 - NVIDIA-ACS-0239988]	
	103	SC 05, Seattle, WA, 2005, Technical Papers, November 12-18, 2005, pp. 5. [NVIDIA-ACS-0239989 - NVIDIA-ACS-0239993]	

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

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2449
	Examiner	Astex, Hemmon
SHEET 7 OF 12	Attorney Docket No.	ZTANN.002PIC6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	104	SC 06, Seattle, WA, 2005, Tutorials, November 12-18, 2005, pp. 3. [NVIDIA-ACS-0239994 - NVIDIA-ACS-0239996]	
	105	SC 09, Portland, OR, 2009, Conference Program, November 14-20, 2009, pp. 158. [NVIDIA-ACS-0238414 - NVIDIA-ACS-0238571] (in 4 parts)	
	106	SCHREINER et al., "Distributed Maple: Lessons Learned on Parallel Computer Algebra in Distributed Environments", Presentation, 2003, pp. 47. [NVIDIA-ACS-0237089 - NVIDIA-ACS-0237135]	
	107	SCHREINER, Wolfgang, "Manager-Worker Parallelism versus Dataflow in a Distributed Computer Algebra System", Research Institute for Symbolic Computation (RISC-Linz), Johannes Kepler University, A-4040 Linz, Austria, 2001, pp. 15. [NVIDIA-ACS-0237074 - NVIDIA-ACS-0237088]	
	108	SCHREINER, Wolfgang, "Manager-Worker Parallelism versus Dataflow in Distributed Maple", Research Institute for Symbolic Computation (RISC-Linz), Johannes Kepler University, A-4040 Linz, Austria, December 13, 2000, pp. 23. [NVIDIA-ACS-0237051 - NVIDIA-ACS-0237073]	
	109	SEIDEL, Brent, "Sample Software", May 24, 2000, pp. 2. https://web.archive.org/web/20050220000225/http://www.serv.net/~bseidel/AppleSeed/Software.html [NVIDIA-ACS-0240006 - NVIDIA-ACS-0240007]	
	110	SILBERSCHATZ et al., "Applied Operating System Concepts", First Edition, Joh Wiley & Sons, Inc., 2000, pp. 641. [NVIDIA-ACS-0172412 - NVIDIA-ACS-0173252] (in 2 parts)	
	111	SONNEVELD, Nick, "Star-P / MATLAB Installation Notes", January 18, 2007, pp. 8. [NVIDIA-ACS-0233675 - NVIDIA-ACS-0233682]	
	112	SONNEVELD, Nick, "Star-P Client Installation", January 18, 2007, pp. 30. [NVIDIA-ACS-0233646 - NVIDIA-ACS-0233673]	
	113	Supercomputing 2005 - Floor Map, Level 4, Exhibit Halls, pp. 1. [NVIDIA-ACS-0239997]	
	114	Supercomputing 2005 - Exhibitor Freight Move in Schedule, Level 4, Exhibit Halls, pp. 2. [NVIDIA-ACS-0239998 - NVIDIA-ACS-0239999]	
	115	"The Power of Distributed Computing in Cancer Research", Supercomputing 2005 Keynote Demonstration, 2005, pp. 2. [NVIDIA-ACS-0238402 - NVIDIA-ACS-0238403]	
	116	"Topics", Cornell University, Cornell Theory Center, 2005, pp. 2. [NVIDIA-ACS-0238303 - NVIDIA-ACS-0238304]	
	117	"Update to Pooch Software Enhances the Cluster Experience", PRWeb, https://www.prweb.com/releases/2006/01prweb330475.htm , January 10, 2006, pp. 4. [NVIDIA-ACS-0239368 - NVIDIA-ACS-0239371]	
	118	Windows in Financial Services, "Microsoft's Leadership in Distributed Enterprise Computing Builds Case for HPC", as archived November 13, 2004, pp. 2. [NVIDIA-ACS-0238404 - NVIDIA-ACS-0238405]	
	119	"Windows Server 2003 Deployment Kit: Automating and Customizing Installations", 2003, pp. 349. [NVIDIA-ACS-0238652 - NVIDIA-ACS-0238998]	
	120	Wolfram Library Archive, "High-Performance Computing with Mathematica: The PoochMPI Toolkit", library.wolfram.com/infocenter/Conferences/6456, pp. 2. [NVIDIA-ACS-0237262 - NVIDIA-ACS-0237263] & [NVIDIA-ACS-0238393 - NVIDIA-ACS-0238394]	
	121	Wolfram Library Archive, "Microsoft Windows Compute Cluster Server 2003 Partner Solution Brief", library.wolfram.com/infocenter/TechNotes/6723, pp. 1. [NVIDIA-ACS-0238401]	

Examiner Signature	Date Considered
<p>*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.</p>	

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2449
	Examiner	Astex, Hemmon
SHEET 8 OF 12	Attorney Docket No.	ZTANN.002PIC6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	122	Wolfram Library Archive, Search Results as printed August 14, 2021 in 3 pages. [NVIDIA-ACS-0239000 - NVIDIA-ACS-0239002]	
	123	"Wolfram Research Offers Support for Microsoft Windows Compute Cluster Server 2003", June 9, 2006, pp. 1. [NVIDIA-ACS-0238999]	
	124	WOLFRAM, Stephen, "The Mathematica Book", 5th Edition, Wolfram Research, Inc., 2003, pp. 1301. [NVIDIA-ACS-0235750 - NVIDIA-ACS-0237950] (In 6 parts)	
	125	Wolfram Science Conference, "NKS 2006", June 16-18, 2006, pp. 2. [NVIDIA-ACS-0239485 - NVIDIA-ACS-0239486]	
	126	Wolfram Technology Conference, "Accelerating Mathematica: Vectors for All", Presentation by Simon McIntosh-Smith, ClearSpeed Technology, October 12, 2006, pp. 27. [NVIDIA-ACS-0239586 - NVIDIA-ACS-0239612]	
	127	Ex. 2031 - Nexus Claim Chart - SEM - 768 Patent, pp. 29. [NVIDIA-ACS-0237264 - NVIDIA-ACS-0237292]	
	128	Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 215 pages.	
	129	Exhibit A-1 - PYMaple System , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 211 pages.	
	130	Exhibit A-2 - Distributed Maple System , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 171 pages.	
	131	Exhibit A-3 - Matlab[®] P System , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 246 pages.	
	132	Exhibit A-4 - FFTW , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 467 pages.	
	133	Exhibit A-5 - Intel Paragon Maple System , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 137 pages.	
	134	Exhibit A-6 - Stony Brook , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 268 pages.	
	135	Exhibit A-7 - AppleSeed Pooch System , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 127 pages.	
	136	Exhibit A-9 - PYMaple References , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 211 pages.	
	137	Exhibit A-10 - Distributed Maple References , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 170 pages.	

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

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2449
	Examiner	Asres, Hermon
SHEET 9 OF 12	Attorney Docket No.	ZTANN.002PIC6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	138	Exhibit A-11 - Matlab *P References , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 242 pages.	
	139	Exhibit A-12 - FFTW References , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 467 pages.	
	140	Exhibit A-13 - Intel Paragon Maple References , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 137 pages.	
	141	Exhibit A-14 - Stony Brook Reference , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 268 pages.	
	142	Exhibit A-15 - AppleSeed Pooch References , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 127 pages.	
	143	Exhibit B-1 - PVMaple System , U.S. Patent No. 8,082,289, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 167 pages.	
	144	Exhibit B-2 - Distributed Maple System , U.S. Patent No. 8,082,289, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 90 pages.	
	145	Exhibit B-3 - Matlab*P System , U.S. Patent No. 8,082,289, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 296 pages.	
	146	Exhibit B-4 - FFTW , U.S. Patent No. 8,082,289, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 178 pages.	
	147	Exhibit B-5 - Intel Paragon Maple System , U.S. Patent No. 8,082,289, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 128 pages.	
	148	Exhibit B-6 - Stony Brook , U.S. Patent No. 8,082,289, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 120 pages.	
	149	Exhibit B-7 - AppleSeed Pooch System , U.S. Patent No. 8,082,289, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 194 pages.	
	150	Exhibit B-9 - PVMaple References , U.S. Patent No. 8,082,289, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 167 pages.	
	151	Exhibit B-10 - Distributed Maple References , U.S. Patent No. 8,082,289, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 91 pages.	
	152	Exhibit B-11 - Matlab *P References , U.S. Patent No. 8,082,289, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 296 pages.	

Examiner Signature	Date Considered
<p>*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.</p>	

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2449
	Examiner	Asres, Hermon
SHEET 10 OF 12	Attorney Docket No.	ZTANN.002PIC6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	153	Exhibit B-12 - FFTW References , U.S. Patent No. 8,082,289, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 178 pages.	
	154	Exhibit B-13 - Intel Paragon Maple References , U.S. Patent No. 8,082,289, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 128 pages.	
	155	Exhibit B-14 - Stony Brook Reference , U.S. Patent No. 8,082,289, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 120 pages.	
	156	Exhibit B-15 - AppleSeed Pooch References , U.S. Patent No. 8,082,289, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 189 pages.	
	157	Exhibit C-1 - PVM Maple System , U.S. Patent No. 8,140,612, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 83 pages.	
	158	Exhibit C-2 - Distributed Maple System , U.S. Patent No. 8,140,612, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 54 pages.	
	159	Exhibit C-3 - Matlab*P System , U.S. Patent No. 8,140,612, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 122 pages.	
	160	Exhibit C-4 - FFTW , U.S. Patent No. 8,140,612, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 102 pages.	
	161	Exhibit C-5 - Intel Paragon Maple System , U.S. Patent No. 8,140,612, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 43 pages.	
	162	Exhibit C-6 - Stony Brook , U.S. Patent No. 8,140,612, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 81 pages.	
	163	Exhibit C-7 - AppleSeed Pooch System , U.S. Patent No. 8,140,612, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 64 pages.	
	164	Exhibit C-9 - PVM Maple References , U.S. Patent No. 8,140,612, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 83 pages.	
	165	Exhibit C-10 - Distributed Maple References , U.S. Patent No. 8,140,612, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 54 pages.	
	166	Exhibit C-11 - Matlab *P References , U.S. Patent No. 8,140,612, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 122 pages.	
	167	Exhibit C-12 - FFTW References , U.S. Patent No. 8,140,612, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 102 pages.	

Examiner Signature	Date Considered
<p>*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.</p>	

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2449
	Examiner	Astex, Hemmon
SHEET 11 OF 12	Attorney Docket No.	ZTANN.002PIC6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	168	Exhibit C-13 - Intel Paragon Maple References , U.S. Patent No. 8,140,612, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 43 pages.	
	169	Exhibit C-14 - Stony Brook Reference , U.S. Patent No. 8,140,612, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 81 pages.	
	170	Exhibit C-15 - AppleSeed Pooch References , U.S. Patent No. 8,140,612, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 64 pages.	
	171	Exhibit D-1 - PVMaple System , U.S. Patent No. 8,676,877, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 75 pages.	
	172	Exhibit D-2 - Distributed Maple System , U.S. Patent No. 8,676,877, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 61 pages.	
	173	Exhibit D-3 - Matlab*P System , U.S. Patent No. 8,676,877, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 129 pages.	
	174	Exhibit D-4 - FFTW , U.S. Patent No. 8,676,877, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 155 pages.	
	175	Exhibit D-5 - Intel Paragon Maple System , U.S. Patent No. 8,676,877, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 36 pages.	
	176	Exhibit D-6 - Stony Brook , U.S. Patent No. 8,676,877, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 97 pages.	
	177	Exhibit D-7 - AppleSeed Pooch System , U.S. Patent No. 8,676,877, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 73 pages.	
	178	Exhibit D-9 - PVMaple References , U.S. Patent No. 8,676,877, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 75 pages.	
	179	Exhibit D-10 - Distributed Maple References , U.S. Patent No. 8,676,877, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 63 pages.	
	180	Exhibit D-11 - Matlab *P References , U.S. Patent No. 8,676,877, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 129 pages.	
	181	Exhibit D-12 - FFTW References , U.S. Patent No. 8,676,877, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 155 pages.	
	182	Exhibit D-13 - Intel Paragon Maple References , U.S. Patent No. 8,676,877, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 36 pages.	

Examiner Signature	Date Considered
<p>*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.</p>	

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannebaum, Zvi
	Art Unit	2449
	Examiner	Astex, Hemon
SHEET 12 OF 12	Attorney Docket No.	ZTANN.002PIC6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	183	Exhibit D-14 - Stony Brook Reference , U.S. Patent No. 8,676,877, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 97 pages.	
	184	Exhibit D-15 - AppleSeed Pooch References , U.S. Patent No. 8,676,877, as listed in Defendant NVIDIA Corporation's Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated October 5, 2019 in 73 pages. <i>[Incorrectly labeled in upper right hand corner of document as Ex. D-7]</i>	

55328828

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

T¹ - Place a check mark in this area when an English language Translation is attached.

Electronic Acknowledgement Receipt	
EFS ID:	45298434
Application Number:	17374864
International Application Number:	
Confirmation Number:	1083
Title of Invention:	CLUSTER COMPUTING
First Named Inventor/Applicant Name:	Zvi Tannenbaum
Customer Number:	20995
Filer:	Jordan Michael Cox/Aimee Kazan
Filer Authorized By:	Jordan Michael Cox
Attorney Docket Number:	ZTANN.002P1C6
Receipt Date:	24-MAR-2022
Filing Date:	13-JUL-2021
Time Stamp:	14:43:06
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	Non Patent Literature	appleseed_html-NVIDIA_ACS_005700.pdf	1725757 c70a729523465017df2927d7002947106b9402c4	no	6

Warnings:

Information:					
2	Non Patent Literature	Appleseed_report_html_NVIDIA_A_ACS_005692.pdf	2740091	no	8
			ffdea344de0b1f4705df3a4c566ae525a9f6275c		
Warnings:					
Information:					
3	Non Patent Literature	appleseed_report03_NVIDIA_ACS_005638.pdf	3747628	no	16
			878881c56fd46c9d5590d3e4ca1d9715fd1611e6		
Warnings:					
Information:					
4	Non Patent Literature	appleseed_report-NVIDIA_ACS_005680.pdf	3634311	no	11
			01d0e698cbaa40b8afcc9d833fe02ca4a7cae312		
Warnings:					
Information:					
5	Non Patent Literature	c2dau_NVIDIA_ACS_005607.pdf	2455366	no	7
			bb2be4dc8643685fbde1b19a28b85cf6a24f240e		
Warnings:					
Information:					
6	Non Patent Literature	Dauger_Research_Vault_CiSE_NVIDIA_ACS_005634.pdf	389135	no	2
			54100fc446f0006024ca00902a3eff5c88cd6347		
Warnings:					
Information:					
7	Non Patent Literature	iccs35150084_-_NVIDIA_ACS_005614.pdf	1377266	no	7
			5e52830402997abadb99317b873af6e33baef9b42		
Warnings:					
Information:					
8	Non Patent Literature	Launch_Den_Mother_Puppy_README_NVIDIA_DR_000703.pdf	1675139	no	10
			e17a17f73db2b8c710c3bc310861c5791b6f81d5		
Warnings:					
Information:					

9	Non Patent Literature	LDM_LP_Advanced_Information_NVIDIA_DR_000713.pdf	2057813	no	10
			5378babcf13db1ddc3e456750c24d175b324210		
Warnings:					
Information:					
10	Non Patent Literature	PnPParallelComputing_NVIDIA_ACS_005654.pdf	2664105	no	8
			d78e0020e069e05fb6818fef5f8784f117445b9b		
Warnings:					
Information:					
11	Non Patent Literature	Pooch4Flyer2_NVIDIA_DR_000002.pdf	862109	no	4
			1282fa07634ccd5bb3c681f465fc0d7996bc0ac1		
Warnings:					
Information:					
12	Non Patent Literature	PoochManual_NVIDIA_DR_000006.pdf	10857993	no	61
			5737d77d2e61c23b0af7b540351380ba9cd1bb8c		
Warnings:					
Information:					
13	Non Patent Literature	PoochManual_NVIDIA_DR_000160.pdf	15216708	no	82
			351b97bd9ce78751baf2ed809e7d35df9d1549f		
Warnings:					
Information:					
14	Non Patent Literature	PoochManual_NVIDIA_DR_000400.pdf	12186818	no	67
			0b58e93508a49fdbf27fac27fbc4f587ca9ed83a		
Warnings:					
Information:					
15	Non Patent Literature	128-Xserve_G5_CLUSTER_ACHIEVE_S_1_-_NVIDIA-ACS-0239003.pdf	185537	no	2
			4c11574c5f477692dba4d9dbadda79c1f97c615		
Warnings:					
Information:					

16	Non Patent Literature	ACS_SEM_productflyer_Oct_2007_-_NVIDIA-ACS-0237306.pdf	178116	no	1
			97e0f473a4086ca4244e0c31b9a306cf419410d5		
Warnings:					
Information:					
17	Non Patent Literature	Apple_Ends_DIY_Repairs_Does_Mac_mini_K_-_NVIDIA-ACS-0239005.pdf	2028169	no	15
			119ed7bbbf3dd67ad238af700ea6e0cec24a7515		
Warnings:					
Information:					
18	Non Patent Literature	Bekker_-_Muglia_Updates_Windows_Server_Roadmap_-_MPC_Mag_12-10-2004_-_NVIDIA-ACS-0239405.pdf	2357970	no	5
			700ec6c67a976c5ceaa53d7e4304ecea9d3ad197		
Warnings:					
Information:					
19	Non Patent Literature	Bekker_Microsoft_Enterprise_Product_Roadmap_for_2005_RealmChannel.pdf	9535263	no	14
			b47c1a74cef72514a61350ca1d6879e8b030f2e2		
Warnings:					
Information:					
20	Non Patent Literature	BEKKER_ENT_News_News_HP_C_Edition_of_Windows_Server_2003_on_Tap_for_2005_-_NVIDIA-ACS-0238318.pdf	2697732	no	3
			63af660a5b2ec982081ec154f3f0c72e3669aa9c		
Warnings:					
Information:					
21	Non Patent Literature	Bernardin_-_Maple_on_the_Intel_Paragon_1996_-_NVIDIA-ACS-0172328.pdf	274972	no	7
			9cce1dc3e1ea21c6ff05b1c0825016929a34351c		
Warnings:					
Information:					
22	Non Patent Literature	Bill_Gates_Shares_Microsofts_Vision_for_-_NVIDIA-ACS-0237311.pdf	1427477	no	5
			251b4694342f375809e5fb7ee535add3e0b35b0f		
Warnings:					
Information:					

23	Non Patent Literature	Bio_Team_and_Life_Sciences_Solution_Brief_-_NVIDIA-ACS-0237316.pdf	3424277	no	10
			d67d102ee56df0e404aa877b6667b475c1753de		
Warnings:					
Information:					
24	Non Patent Literature	CCS_Sample_RIS_Scripts_-_NVIDIA-ACS-0237326.pdf	1499670	no	55
			1ca2c409dcb7af60e3d131b3d590b6d92736c2e		
Warnings:					
Information:					
25	Non Patent Literature	CCS_SB_Wolfram_-_NVIDIA-ACS-0237542.pdf	2189558	no	7
			049ae045aefda17b75576c35e307ad914920aa78		
Warnings:					
Information:					
26	Non Patent Literature	Chinchilla_-_Parallel_N-Body_Simulation_using_GPUs_12-2004_-_NVIDIA-ACS-0165482.pdf	875535	no	8
			279b22779e23f94f8bd839d1cedea5a48e25939c		
Warnings:					
Information:					
27	Non Patent Literature	ChoyDraftPHd_-_NVIDIA-ACS-0233432.pdf	345748	no	16
			e556fdad1f60e371e91544de59a665d9f7ade528a		
Warnings:					
Information:					
28	Non Patent Literature	ChoyThesis_-_NVIDIA-ACS-0233448.pdf	4438692	no	67
			48eb65c65c58be2227cd347425a2c8d7d5751a07		
Warnings:					
Information:					
29	Non Patent Literature	Cluster_Components_and_Inte-rconnects_-_NVIDIA-ACS-0237553.pdf	440553	no	3
			1db7a533f70bf1c848a6dc156fb4d4b059b275b2		
Warnings:					
Information:					

30	Non Patent Literature	cluster_configuration_aided_by_simulation_-_NVIDIA-ACS-0237556.pdf	2400003	no	6
			928cc0c827b169d8535d3114dbb535369d56f4ef		
Warnings:					
Information:					
31	Non Patent Literature	clusterReport2007_-_NVIDIA-ACS-0237562.pdf	612153	no	5
			ad90b2c68040c32b299ddcc20bc6ee60a2a31257		
Warnings:					
Information:					
32	Non Patent Literature	Compute_Cluster_Automation_-_NVIDIA-ACS-0237567.pdf	25886740	no	359
			4df052af9b64479943b159bfeca7194f5d0f0e6e		
Warnings:					
Information:					
33	Non Patent Literature	Compute_Cluster_Installation_-_NVIDIA-ACS-0237926.pdf	20350936	no	286
			4db9af9154e7a09e99af6acdd4b9a99a7625d3ab		
Warnings:					
Information:					
34	Non Patent Literature	Compute_Cluster_Job_Aids_-_NVIDIA-ACS-0238212.pdf	13334243	no	80
			e9e98dcb6d6bf0a2e06b5e9d43c7bfcc772edeba		
Warnings:					
Information:					
35	Non Patent Literature	Compute_Cluster_Server_Getting_Started_-_NVIDIA-ACS-0238300.pdf	47872	no	1
			9b888f0f30477bc0fdffb366c0c86d31b4a7dff8a		
Warnings:					
Information:					
36	Non Patent Literature	Compute_Cluster_Server_Technical_Reference_-_NVIDIA-ACS-0238301.pdf	135337	no	2
			cdfd07d6653dc99f3d84f9784a85dd4888ec0d0		
Warnings:					
Information:					

37	Non Patent Literature	Compute_Cluster_Server_Glossary_-_NVIDIA-ACS-0238292.pdf	301942	no	6
			225a30e61657cc85dd1aebfd74b90c03a0c3a747		
Warnings:					
Information:					
38	Non Patent Literature	Cownie_-_NVIDIA-ACS-0232908.pdf	436640	no	9
			d5373600780ecf5a2a7540c08d7f740ece400221		
Warnings:					
Information:					
39	Non Patent Literature	CTC_Topics_-_NVIDIA-ACS-0238303.pdf	563908	no	2
			87a2b9515a798042dc70ec3cda0f0bad7d1add17		
Warnings:					
Information:					
40	Non Patent Literature	Dauger_Research_-_Keeping_America_Strong_Award_02-2005_-_NVIDIA-ACS-0239455.pdf	1374571	no	3
			8e3d897075b1383ac8d14469a5de6417a784ba90		
Warnings:					
Information:					
41	Non Patent Literature	Dauger_Research_-_Whats_New_-_NVIDIA-ACS-0239458.pdf	20289829	no	25
			0782736d0a66b749ccfb0cb6ba9ef8b9d3b7830		
Warnings:					
Information:					
42	Non Patent Literature	Dauger_Research_Inc_-_NVIDIA-ACS-0239020.pdf	235068	no	1
			c0e4f6f287591caf96c4f067a652dea93316a3a0		
Warnings:					
Information:					
43	Non Patent Literature	Dauger_Research_Vault_-_Parallel_Adder_-_NVIDIA-ACS-0237154.pdf	3113349	no	7
			f3ca165fb65497dee47cdd4a35bec579c93abde6		
Warnings:					
Information:					

44	Non Patent Literature	Dauger_Research_Vault_-_Parallel_Circle_Pi_-_NVIDIA-ACS-0237161.pdf	6532955	no	13
			f1281b8cfbd0f60bb2cbb6e6860b5a23fafae7858		
Warnings:					
Information:					
45	Non Patent Literature	Dauger_Research_Vault_-_Parallel_Knock_-_NVIDIA-ACS-0237174.pdf	1271414	no	4
			20e5258b67409a142bd09ab573b130a01f8fac65		
Warnings:					
Information:					
46	Non Patent Literature	Dauger_Research_Vault_-_Parallel_Life_-_NVIDIA-ACS-0237178.pdf	6528277	no	14
			5230ab3ff58944dd873c8377085e25249dfff3532		
Warnings:					
Information:					
47	Non Patent Literature	Dauger_Research_Vault_-_Parallel_Pascal_s_Triangle_-_NVIDIA-ACS-0237192.pdf	5270717	no	10
			aeb001ae843a9f559472241906173932851a3141		
Warnings:					
Information:					
48	Non Patent Literature	Dauger_Research_Vault_-_Parallel_Zoology_-_NVIDIA-ACS-0237202.pdf	2104508	no	7
			b2dedb070c04935528959dccc6a72ef5ce139c29		
Warnings:					
Information:					
49	Non Patent Literature	Compute_Cluster_Server_Setup_Command-Line_Interface_Reference_-_NVIDIA-ACS-0238298.pdf	208824	no	2
			36cc849953f8576d557ab5570a78a6a9b8a04fffd		
Warnings:					
Information:					
50	Non Patent Literature	Dauger_-_Plug-and-Play_Cluster_Computing_Bring_s_HPC_to_the_Mainstream_-_NVIDIA-ACS-0239454.pdf	164767	no	1
			cd633e17640d83ba8261f0c5f999f07c67beba4a		
Warnings:					
Information:					

51	Non Patent Literature	Dauger_Research_Vault_-_Parallelization_-_NVIDIA-ACS-0237209.pdf	912577 6a114c6fb748374626cb4448411b5fdb28175c017	no	3
Warnings:					
Information:					
52	Non Patent Literature	Dauger_Research_Vault_-_Visualizing_Message_Passing_with_MacMPI_-_NVIDIA-ACS-0237212.pdf	1440368 8e22079961ff8068eb1b948efcf35ec54566dba	no	5
Warnings:					
Information:					
53		IDS_Transmittal_and_listing.pdf	2031999 aaea34483a2e80b42ea22e9fd3b16b88206153d8	yes	14
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Transmittal Letter		1	2	
	Information Disclosure Statement (IDS) Form (SB08)		3	14	
Warnings:					
Information:					
Total Files Size (in bytes):			209038505		
<p>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.</p> <p><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.</p> <p><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.</p> <p><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.</p>					

INFORMATION DISCLOSURE STATEMENT

First Inventor: Tannenbaum, Zvi	Conf. No.: 1083
App. No.: 17/374864	Filed: July 13, 2021
Examiner: Asres, Hermon	Art Unit: 2449
Title: CLUSTER COMPUTING	

Mail Stop RCE

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

References and Listing

Pursuant to 37 CFR 1.56, an Information Disclosure Statement listing references is provided herewith. Copies of any listed foreign and non-patent literature references are being submitted.

As previously noted, Advanced Cluster Systems, Inc. ("ACS") filed suit against NVIDIA Corporation ("NVIDIA") in the U.S. District Court for the District of Delaware (Case No. 1:19-cv-02032-UNA) on October 28, 2019, alleging infringement of various issued patents, including U.S. Patent Nos. 8,082,289, 8,140,612, 8,676,877, and 10,333,768. Recently, certain documents were received by ACS from NVIDIA during discovery. These documents, including invalidity contentions, are hereby disclosed, adding to other documents previously disclosed from the litigation. The litigation is ongoing.

Pursuant to 37 CFR 1.97(g) and (h), Applicant makes no representation that the information is considered to be material to patentability. Additionally, inclusion on this list is not an admission that any of the cited documents are prior art in this application. Further, Applicant makes no representation regarding the completeness of this list, or that better art does not exist.

No Disclaimers

To the extent that anything in the Information Disclosure Statement or the listed references could be construed as a disclaimer of any subject matter supported by the present application, Applicant hereby rescinds and retracts such disclaimer.

Timing of Disclosure

This Information Disclosure Statement is being filed within three months of the filing date or date of national phase entry, with an RCE or before the receipt of a First Office Action after an RCE, and no fee is believed to be required.

The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment, to Account No. 11-1410.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: March 23, 2022

By: /Jordan M. Cox/
Jordan M. Cox
Registration No. 70,875
Registered Practitioner
(949) 760-0404

55323982

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2449
	Examiner	Asres, Hermon
SHEET 1 OF 2	Attorney Docket No.	ZTANN.002PIC6

U.S. PATENT DOCUMENTS						
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	

FOREIGN PATENT DOCUMENTS						
Examiner Initials	Cite No.	Foreign Patent Document Country Code-Number-Kind Code Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T ¹

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	1	Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 151 pages.	
	2	Exhibit A-1 - PVMaPle System , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 270 pages.	
	3	Exhibit A-2 - Distributed Maple System , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 261 pages.	
	4	Exhibit A-3 - Matlab*P System , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 254 pages.	
	5	Exhibit A-4 - FFTW , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 467 pages.	
	6	Exhibit A-5 - Intel Paragon Maple System , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 137 pages.	
	7	Exhibit A-6 - Stony Brook Bluster , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 289 pages.	
	8	Exhibit A-7 - AppleSeed Pooch System , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 160 pages.	
	9	Exhibit A-9 - PVMaPle References , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 270 pages.	
	10	Exhibit A-10 - Distributed Maple References , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 262 pages.	
	11	Exhibit A-11 - Matlab *P References , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 246 pages.	

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

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2449
	Examiner	Asres, Hermon
SHEET 2 OF 2	Attorney Docket No.	ZTANN.002P1C6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	12	Exhibit A-12 - FFTW References , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 467 pages.	
	13	Exhibit A-13 - Intel Paragon Maple References , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 137 pages.	
	14	Exhibit A-14 - Stony Brook Cluster References , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 289 pages.	
	15	Exhibit A-15 - AppleSeed Pooch References , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 163 pages.	
	16	Exhibit A-17 - MultiMatlab System , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 249 pages.	
	17	Exhibit A-18 - MultiMatlab References , U.S. Patent No. 10,333,768, as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 254 pages.	
	18	Exhibit A-19 - U.S. Patent No. 10,333,768 ("768") v. Microsoft Windows Compute Cluster Server 2003 ("CCS") , as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 41 pages.	
	19	Exhibit A-20 - U.S. Patent No. 10,333,768 ("768") v. Supercomputing Engine for Mathematica ("SEM") , as listed in Defendant NVIDIA Corporation's Supplemental Invalidity Contentions in Case No. 1:19-cv-2032-CFC, Advanced Cluster Systems, Inc., v. NVIDIA Corporation as dated August 17, 2021 in 126 pages.	

55129870

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

T¹ - Place a check mark in this area when an English language Translation is attached.

Electronic Patent Application Fee Transmittal				
Application Number:		17374864		
Filing Date:		13-Jul-2021		
Title of Invention:		CLUSTER COMPUTING		
First Named Inventor/Applicant Name:		Zvi Tannenbaum		
Filer:		Jordan Michael Cox/Janet Teeters		
Attorney Docket Number:		ZTANN.002P1C6		
Filed as Small Entity				
Filing Fees for Utility under 35 USC 111(a)				
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:				
RCE- 1ST REQUEST	2801	1	680	680
Total in USD (\$)				680

Electronic Acknowledgement Receipt	
EFS ID:	45033230
Application Number:	17374864
International Application Number:	
Confirmation Number:	1083
Title of Invention:	CLUSTER COMPUTING
First Named Inventor/Applicant Name:	Zvi Tannenbaum
Customer Number:	20995
Filer:	Jordan Michael Cox/Jennifer Neat
Filer Authorized By:	Jordan Michael Cox
Attorney Docket Number:	ZTANN.002P1C6
Receipt Date:	18-FEB-2022
Filing Date:	13-JUL-2021
Time Stamp:	18:35:39
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$ 680
RAM confirmation Number	E20222HI36412362
Deposit Account	111410
Authorized User	Jennifer Neat
<p>The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:</p> <p>37 CFR 1.16 (National application filing, search, and examination fees)</p> <p>37 CFR 1.17 (Patent application and reexamination processing fees)</p>	

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Request for Continued Examination (RCE)	ZTANN-002P1C6_RCE.pdf	1349938	no	3
			8b6a285431cf33599a80f181e3d8a654566e7e076		
Warnings:					
Information:					
2	Non Patent Literature	2021-08-17_NVIDIA_Supplemental_Invalidity_Contentions.pdf	12127450	no	151
			071d1d19cd14e213851071314b2215d2bf25b540		
Warnings:					
Information:					
3	Non Patent Literature	A01_PVMaple_System.pdf	16980608	no	270
			9f42284107d9508cc14a63931cd656a0c9098133		
Warnings:					
Information:					
4	Non Patent Literature	A02_Distributed_Maple_System.pdf	11857045	no	261
			a0137331a27a29d3146bd060fda60951cfa6f332		
Warnings:					
Information:					
5	Non Patent Literature	A03_Matlab_P.pdf	21355123	no	254
			bd3aa72288272bb872ffdd239bd314936c46ad884		
Warnings:					
Information:					
6	Non Patent Literature	A04_FFTW.pdf	4398659	no	467
			be0c587e4b2ef816a0c975e4606e46034b65735d		
Warnings:					
Information:					

7	Non Patent Literature	A05_Intel_Paragon_Maple.pdf	1861221	no	137
			ba6ddcd3ddd2c9a9224a0dac98a549d828a064c8		
Warnings:					
Information:					
8	Non Patent Literature	A06_Stony_Brook_GPU.pdf	7140455	no	289
			650b4aef374e11e8f99f5815c693e09e2cc9c2b2		
Warnings:					
Information:					
9	Non Patent Literature	A07_AppleSeed.pdf	11837044	no	160
			6444d063ae5c3299b06541e31f78f464e36a8545		
Warnings:					
Information:					
10	Non Patent Literature	A09_PV_Maple.pdf	16987857	no	270
			f01cb767dfcd4e80d788a323d23e833966e8cb11		
Warnings:					
Information:					
11	Non Patent Literature	A10_Distributed_Maple.pdf	11868380	no	262
			bff060c619c9bade0ff8fb551f596d4a0e3a2eb739		
Warnings:					
Information:					
12	Non Patent Literature	A11_Matlab_P_References.pdf	20232454	no	246
			e5e6f600128caa7ba41fedc5779b074cbdfdc407		
Warnings:					
Information:					
13	Non Patent Literature	A12-768_Patent_FFTW_References.pdf	4416100	no	467
			9073d9db692833b4f4f352d9cf06ffa4d1d2878d		
Warnings:					
Information:					

14	Non Patent Literature	A13_Intel_Paragon_Maple.pdf	1867818	no	137
			0fe65de6f9f32e8e7062030ec70f133d5f975928		
Warnings:					
Information:					
15	Non Patent Literature	A14_Stony_Brook_GPU.pdf	7150693	no	289
			d8049dc913c542ff3c0ac87bce2bb005d5745519		
Warnings:					
Information:					
16	Non Patent Literature	A15_AppleSeed_References.pdf	7978651	no	163
			04209148cbc6f6414562dbcebc5f395a3fa74a		
Warnings:					
Information:					
17	Non Patent Literature	A17_MultiMatlab.pdf	21683734	no	249
			82f869843862464e7ee8e8dbbd29e65ff610b52f		
Warnings:					
Information:					
18	Non Patent Literature	A18_MultiMatlab_References.pdf	21713335	no	254
			75e79d30cc728e4b53e9b5b745574d0d70fa0836		
Warnings:					
Information:					
19	Non Patent Literature	A19_Microsoft_CCS.pdf	8789563	no	41
			c6bb4093e53fce225b1c5467eb90f68258216afe		
Warnings:					
Information:					
20	Non Patent Literature	A20_SEM.pdf	23348401	no	126
			2f2180b60de7cad26667bdf3a6bf3f19481dc642		
Warnings:					
Information:					

21		ZTANN-002P1C6_IDS.pdf	55123	yes	4
			8f817a14402a67297dc5f002cba36fd6a58d01e8		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Transmittal Letter		1	2	
	Information Disclosure Statement (IDS) Form (SB08)		3	4	
Warnings:					
Information:					
22	Fee Worksheet (SB06)	fee-info.pdf	37934	no	2
			ce69e407aff8e258d3eabacbb96883cfc1d8ca1		
Warnings:					
Information:					
Total Files Size (in bytes):			235037586		
<p>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.</p> <p><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.</p> <p><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.</p> <p><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.</p>					

Doc code: RCEX

Doc description: Request for Continued Examination (RCE)

PTO/SB/30EFS (02-18)

Approved for use through 11/30/2020. 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	17374864	Filing Date	2021-07-13	Docket Number (if applicable)	ZTANN.002P1C6	Art Unit	2449
First Named Inventor	Zvi Tannenbaum			Examiner Name	Hermon Asres		

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 111410

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

☒ Patent Practitioner Signature

Applicant Signature

Doc code: RCEX

Doc description: Request for Continued Examination (RCE)

PTO/SB/30EFS (02-18)

Approved for use through 11/30/2020. 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	Jordan M. Cox/	Date (YYYY-MM-DD)	2022-02-17
Name	Jordan M. Cox	Registration Number	70875

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.

INFORMATION DISCLOSURE STATEMENT

First Inventor: Tannenbaum, Zvi	Conf. No.: 1083
App. No.: 17/374864	Filed: July 13, 2021
Examiner: Asres, Hermon	Art Unit: 2449
Title: CLUSTER COMPUTING	

Mail Stop RCE

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

References and Listing

Pursuant to 37 CFR 1.56, an Information Disclosure Statement listing references is provided herewith. Copies of any listed foreign and non-patent literature references are being submitted.

Advanced Cluster Systems, Inc. (“ACS”) filed suit against NVIDIA Corporation (“NVIDIA”) in the U.S. District Court for the District of Delaware (Case No. 1:19-cv-02032-UNA) on October 28, 2019, alleging infringement of various issued patents, including U.S. Patent Nos. 8,082,289, 8,140,612, 8,676,877, and 10,333,768. Recently, certain documents were received by ACS from NVIDIA during discovery. These documents, including invalidity contentions, are hereby disclosed, adding to other documents previously disclosed from the litigation. The litigation is ongoing.

No Disclaimers

To the extent that anything in the Information Disclosure Statement or the listed references could be construed as a disclaimer of any subject matter supported by the present application, Applicant hereby rescinds and retracts such disclaimer.

Timing of Disclosure

This Information Disclosure Statement is being filed within three months of the filing date or date of national phase entry, with an RCE or before receipt of a First Office Action after an RCE, and no fee is believed to be required.

The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment, to Account No. 11-1410.

Page 2 of 2

By: /Jordan M. Cox/
Jordan M. Cox
Registration No. 70,875
Registered Practitioner
(949) 760-0404

Ex.1002 / Page 435 of 616
Intel Corporation et al



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.
17/374,864	07/13/2021	Zvi Tannenbaum	ZTANN.002P1C6	1083
20995	7590	12/08/2021	EXAMINER	
KNOBBE MARTENS OLSON & BEAR LLP			ASRES, HERMON	
2040 MAIN STREET			ART UNIT	PAPER NUMBER
FOURTEENTH FLOOR			2449	
IRVINE, CA 92614			NOTIFICATION DATE	DELIVERY MODE
			12/08/2021	ELECTRONIC

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 the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

efiling@knobbe.com
jayna.cartee@knobbe.com

Notice of Allowability	Application No. 17/374,864	Applicant(s) Tannenbaum et al.	
	Examiner HERMON ASRES	Art Unit 2449	AIA (FITF) Status No

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

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

1. ☒ This communication is responsive to 11/19/2021.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

3. ☒ The allowed claim(s) is/are 2-31. As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

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

Certified copies:

a) ☐ All b) ☐ Some* c) ☐ None of the:

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

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

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

* Certified copies not received: _____.

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

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

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

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

Attachment(s)

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

/HERMON ASRES/ Primary Examiner, Art Unit 2449	
---	--

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	17/374864	
	Filing Date	July 13, 2021	
	First Named Inventor	Tannenbaum, Zvi	
	Art Unit	2449	
(Multiple sheets used when necessary)		Examiner	Asres, Hermon
SHEET 1 OF 1		Attorney Docket No.	ZTANN.002P1C6

U.S. PATENT DOCUMENTS

Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	11,128,519 (Attorney Docket No. ZTANN.002P1C5) U.S. Application No. 16/449,084 and its entire prosecution history**	09-21-2021	Tannenbaum et al.	
	2	2021/0344551 (Attorney Docket No. ZTANN.002P1C7) U.S. Application No. 17/374,789 and its entire prosecution history**	11-04-2021	Tannenbaum et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials	Cite No.	Foreign Patent Document Country Code-Number-Kind Code Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T ¹
	3	JP 2001-229145	08-24-2001	Oki Systek Tokai Co Ltd.		X

NON PATENT LITERATURE DOCUMENTS

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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹

54607160

** Signature indicates consideration of publication and file history. The Examiner has access to these materials through the PTO computer systems. If additional copies are desired, please notify Applicants through their attorneys.

Examiner Signature	/HERMON ASRES/	Date Considered	12/03/2021
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.			

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.A./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2449
	Examiner	Asres, Hermon
SHEET 1 OF 1	Attorney Docket No.	ZTANN.002P1C6

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	11,128,519 (Attorney Docket No. ZTANN.002P1C5) U.S. Application No. 16/449,084 and its entire prosecution history**	09-21-2021	Tannenbaum et al.	
	2	2021/0344551 (Attorney Docket No. ZTANN.002P1C7) U.S. Application No. 17/374,789 and its entire prosecution history**	11-04-2021	Tannenbaum et al.	

FOREIGN PATENT DOCUMENTS						
Examiner Initials	Cite No.	Foreign Patent Document Country Code-Number-Kind Code Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T ¹
	3	JP 2001-229145	08-24-2001	Oki Systek Tokai Co Ltd.		X

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹

54607160

** Signature indicates consideration of publication and file history. The Examiner has access to these materials through the PTO computer systems. If additional copies are desired, please notify Applicants through their attorneys.

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

T¹ - Place a check mark in this area when an English language Translation is attached.



Method and device for data communication between processors

Abstract

PROBLEM TO BE SOLVED: To improve the reliability of the communications of an inter-processor data communication device. **SOLUTION:** The inter-processor data communication device which sends and/or receives unit data by using a communication function having a non-exclusive transfer instruction by which an instruction other than a transfer instruction is executed during the execution of the transfer instruction is equipped with a time-out monitor means which monitors a time-out state as to the transmission and/or reception of the unit data.

JP2001229145A

Japan

Find Prior Art

Similar

Other languages: Japanese

Inventor: Hideki Ashikawa, 阿部 秀樹, Shigeru Miemi, 米見 茂, Akihiko Sugisawa, 杉沢 彰彦

Worldwide applications

2000 - JP

Application JP2000037878A events ⓘ

2000-02-16 - Application filed by Oki System Tokai Co Ltd, 株式会社沖システック東海, Oki Electric Ind Co Ltd, 沖電気工業株式会社

2000-02-16 - Priority to JP2000037878A

2001-08-24 - Publication of JP2001229145A

Status - Pending

Info: Legal events, Similar documents, Priority and Related Applications

External links: Espacenet, Global Dossier, Discuss

Claims (4)

Hide Dependent ^

[Claims]

1. An inter-processor data communication method for transmitting and / or receiving a unit message using a communication function having a non-exclusive transfer instruction capable of executing an instruction other than the transfer instruction during execution of the transfer instruction. The method according to claim 1, wherein a timeout is monitored for transmission and / or reception of the unit message.
2. The data communication method between processors according to claim 1, wherein identification information for identifying the difference between the information contents contained in said unit message at the time of reception is added to each unit message. When a timeout is detected by the timeout monitoring unit, a retransmission process of retransmitting the data. And / or upon receiving, based on the identification information, check for differences in information contents contained in each of the received unit messages, and when a plurality of unit messages containing the same information contents are received, finally, A data communication method between processors, comprising executing a received data selection process for processing only a received unit message as a genuine unit message.
3. An inter-processor data communication device for transmitting and / or receiving a unit message using a communication function having a non-exclusive transfer instruction capable of executing an instruction other than the transfer instruction during execution of the transfer instruction. 2. The inter-processor data communication device according to claim 1, further comprising a timeout monitoring unit that monitors a timeout with respect to transmission and / or reception of the unit message.
4. The interprocessor data communication apparatus according to claim 3, wherein identification information for identifying a difference in information content contained in said unit message at the time of reception is added to each unit message. Retransmission means for retransmitting the data when a timeout is detected by the timeout monitoring means. And / or upon receiving, based on the identification information, check for differences in information contents contained in each of the received unit messages, and when a plurality of unit messages containing the same information contents are received, finally, An inter-processor data communication device, comprising: a received data selection unit that processes only a received unit message as a genuine unit message.

Description

DETAILED DESCRIPTION OF THE INVENTION

[0001]

BACKGROUND OF THE INVENTION 1. Field of the Invention The present invention relates to an inter-processor data communication method, which can be used, for example, when data is transferred between a plurality of processors using a non-blocking transfer instruction of an MPI (Message Passing Interface) function. It is,

[0002] The present invention also relates to an inter-processor data communication apparatus using such an inter-processor data communication method.

[0003]

2. Description of the Related Art MPI is a message passing library whose specifications are being examined by the MPI Forum.

MPI is a program written in a language or C language

It is called as a function of C or a subroutine. MPI

MPI includes one-to-one communication function, collective communication (collection / distribution / integration operation) function, derived data type function, communicator (Communicator)

locator) function, process topology (Process Topology)

y) There is an advantage that it can support a function, a dynamic process generation / management function, a one-side communication function, an extended collective communication (collective distribution / integrated operation) function, and a parallel input / output function.

[0004] Among them, the communicator function can realize a multiplex communication space which has been difficult to realize with a conventional message passing library, and the process topology function allows a programmer to define a process in an easy-to-understand form. .

Such an MPI function has a non-blocking transfer instruction as another feature.

[0006] The non-blocking transfer instruction is, for example, an arithmetic processing requiring an enormous amount of arithmetic processing is shared by a plurality of processors in a network, and distributed processing (or parallel processing) is performed.

This is advantageous in cases such as processing.

[0007] Due to the non-blocking nature of the transfer instruction,

Each of the processors involved in the distributed processing can execute other processing even before the started transfer instruction is completely completed. By applying the calculation shared by the processors among them, each processor can execute the message transfer processing even while executing the calculation, and can execute the calculation even while executing the message transfer processing.

FIGS. 2A and 2B show a flowchart of only the communication portion for executing the non-blocking transfer instruction. In this transfer instruction, as shown in FIG. 3, message transfer between the processor 10 and the processor 11 in the network 8 is executed.

FIG. 2A shows the operation in the processor 10, and FIG. 2B shows the operation in the processor 11. The flowchart of FIG. 2A includes steps S1 to S5, and the flowchart of FIG.

To S9.

In order to transmit (or receive) a message in the processor 10, as shown in FIG. 2A, for example, function names MPI_Isend, MPI_Irecv is executed (S1), the processor 11 also receives (or transmits) the message, so that the function names MPI_Irecv, MPI_Isend (S6).

Next, the processor 10 executes another process different from the non-blocking transfer instruction, for example, the above-mentioned shared operation (S2). Similarly, the processor 11 performs another process different from the non-blocking transfer instruction, for example, the assigned calculation is executed (S7).

Then, step S3 following step S2.

Then, the processor 10 executes, for example, the function name MPI_Test.

In step S8 following step S7, the processor 11 checks, for example, the function name MPI

Wait for transfer completion corresponding to MPI_Test.

In the transfer confirmation processing in step S3, the MPI

When the function of MPI_Test is called, a status indicating whether or not the transfer is completed is set, and the function immediately returns to the function call side. If the function of MPI_Test is called here, as soon as the transfer is completed, the function returns to the function calling side.

MPI is an application interface and MPI for applications that use MPI.

Merely specifies the function of each function in the function library that should be prepared, so various contents may be considered as the specific contents of each process such as the transfer confirmation process as long as they do not violate the specifications. But not mentioned here.

When there is no problem in the operation of the network and each processor, and there is no failure, the processor 10 performs a status check following a one-shot transfer confirmation (S3) (S4), and completes the communication part normally. This is performed (S5). Even after the normal termination, the processing such as the above-mentioned shared operation can be continuously executed.

The status check (S4) performed after the transfer confirmation (S3) is performed because the transfer confirmation (S3) simply outputs whether or not the transfer has been completed.

To know if the termination was successful,

This is because it is necessary to confirm the contents of the status by the status check.

Similarly, in the normal state, the processor 11

However, after a short transfer completion wait (S8), normal termination of the communication part is performed (S9).

Subsequently, when a message other than the above message is transferred, the same processing as that in the flowcharts of FIGS. 2A and 2B is repeated in each of the processors 10 and 11.

If it is assumed that a transfer instruction or the like has a blocking property, each processor completely occupies the hardware resources for the operation until the operation of the shared operation is completed. Then, no other processing can be executed. That is, the transfer instruction cannot be executed until the operation to be shared is completed, and the operation cannot be executed until the transfer instruction is completely completed.

From this, the usefulness of the non-blocking transfer instruction can be understood.

[0021]

When there is a problem in the operation of the network or each processor and a failure occurs, the step S
The transfer confirmation of No. 3 may be sorry only for one shot, that is, once, and the transfer completion wait (S8) may be sorry in a short time.

In the non-blocking transfer of the MPI function, there is no concept of a timeout, and transfer time management and time monitoring cannot be performed. In the worst case, the transfer confirmation in the step S3 is repeated endlessly. A situation in which the waiting for the transfer completion in step S8 is unlimitedly long may occur.

In this case, the next transfer is kept waiting until the transfer is completed, and communication is stagnated. Also,
Regarding the transfer completion wait (S8), when the end status cannot be obtained due to some failure, the program is in a stopped state, and the system is stopped.

After all, the non-blocking transfer command having such a problem has the above-mentioned great advantage, but it can be said that the communication efficiency is low as a whole, the usability is low, and the reliability is low.

[0025]

According to a first aspect of the present invention, there is provided a communication function having a non-exclusive transfer instruction capable of executing an instruction other than the transfer instruction during execution of the transfer instruction. A method for transmitting and / or receiving a unit message by using an inter-processor data communication method, characterized in that a timeout is monitored for the transmission and / or reception of the unit message.

In the second invention, a unit message is transmitted and / or received by using a communication function having a non-exclusive transfer instruction capable of executing an instruction other than the transfer instruction during execution of the transfer instruction. An inter-processor data communication apparatus according to claim 1, further comprising timeout monitoring means for monitoring a timeout with respect to transmission and / or reception of said unit message.

[0027]

DESCRIPTION OF THE PREFERRED EMBODIMENTS (A) Embodiments First and second embodiments of the inter-processor data communication method and apparatus of the present invention will be described below.

(A-1) Configuration of the First Embodiment FIG. 5 shows the internal configuration of the processor 30 of the present embodiment.
The processor 30 is the same as the processor 10 of FIG.
Is the processor corresponding to
Communicate with A network including the processors 30 and 31 is referred to as a network 9.

In FIG. 5, the processor 30 includes a transfer processing unit 21, a timeout monitoring unit 22, and a transfer termination unit 23.

Of these, the transfer processing unit 21 is the same as that shown in FIG.

This is a portion having a function of executing a non-blocking transfer operation on the transmission side similar to the flowchart of FIG.

Note that the transmitting side performs the processing of FIG.

Performing the processing in FIG. 2B on the receiving side is an example of a system in which distributed processing is performed and each processing operates in an input-driven manner, and is not necessarily required in the MPI standard.

When a non-blocking transfer instruction of MPI is performed, activation (MPI_Isend, MPI_Irecv) and completion of the transfer (MPI_Test or MPI_Wait), the MPI_Isend transfer confirmation includes the MPI_Test may be used, and MPI_Wait may be used.

Similarly, for MPI_Irecv,

Not only wait for transfer completion by MPI_Wait,

Confirmation of completion by MPI_Test may be performed. Where M

When MPI_Test is used, it is equivalent to simply inquiring about the result of the transfer, so that an inquiry (that is, a call to MPI_Test) must be made until the transfer is completed.

If the completion of the transfer cannot be confirmed at the transfer confirmation in step S22 (see FIG. 1) (this process is the same as the transfer confirmation in step S3), the timeout monitor 22 is the part that executes

In the timeout check process, a difference between the transfer start time and the current time is obtained, and when the difference is equal to or longer than a preset timeout time, it is determined that a transfer timeout has occurred. If it is determined that a transfer timeout has occurred, the timeout monitoring unit 22
Outputs the timeout detection signal TD to the transfer termination unit 23.

The transfer start time required for the timeout monitoring unit 22 to obtain the difference can be obtained by checking the time data 25 included in the data structure 26 shown in FIG.

The data structure 26 shown in FIG. 4 is a handshake structure (here, a handshake structure passed between the data transfer activation process and the data transfer confirmation process in the same processor for associating the transfer activation and the confirmation. "handshake"
Indicates a handshake between programs in the same processor), the request unit 24 and the time data unit 2
And 5.

The request section 24 includes the MPI_

It is a portion including a data buffer address, a transfer word number (communication word number), a data type, a transfer destination address (MPI rank), a communication request handle, and the like used as arguments of MPI functions such as send and MPI_Irecv.

In the communication using the MPI function, it is necessary to specify the communication partner and the number of communication words for each communication. Therefore, each of the processors 30 and 31 recognizes the address of the communication partner and the number of communication words prior to the communication. However, this recognition is not conscious of the application using the MPI, but should be realized by a method other than the message transfer by the MPI function.

For example, from the processor 30 to the processor 3

For example, when a message is transmitted to the transmission source 1, the transmission function of the transmission source processor 30 performs the handshaking between the processors prior to the transmission of the message, and the address of the transmission destination processor 31 and the address of the transmission destination of the message are transmitted. It is necessary to specify the number of communication words, and to specify the address of the processor 30 as the

transmission source and the number of communication words of the message in the reception function of the processor 31 as the transmission destination. It is considered necessary to perform the method by a method other than the message transfer by the PI function.

If the MPI communication is performed up to the inter-processor handshake, there is a contradiction that further preceding MPI communication is required to transmit the address of the communication partner and the number of communication words which are the premise of the inter-processor handshake. Because.

Accordingly, a specific example of control communication such as handshaking between processors is contrary to the spirit of the present embodiment, which intends to cover the entire range of the MPI standard as it is, and will not be described in detail here.

The transfer processing section 21 of the components of the processor 30 shown in FIG. 5 is a section for taking out the argument from the request section 24, setting the argument in the MPI function, and activating the transfer.

Further, the timeout monitoring unit 22 adds the time data 25 indicating the communication start time corresponding to the argument to the request unit 24 from which the transfer processing unit 21 has extracted the argument, thereby creating the data structure 26. It has the function of generating and storing.

The transfer termination unit 23 outputs a transfer termination signal CL upon receiving the timeout detection signal TD from the timeout monitoring unit 22, and causes the processor 30 to terminate the transfer process (message transmission process). As a result, it is possible to prevent communication from stagnating due to endless repetition of transfer confirmation as in step S3.

In the case of the present embodiment, it is assumed that the functions of the processor 31 on the receiving side are the same as those of the processor 30 on the transmitting side shown in FIG. Therefore, the internal configuration of the processor 31 is the same as the configuration shown in FIG.

The only difference is that the contents of the transfer processing performed by the transfer processing unit 21 are transmission processing in the transmission processor 30 and reception processing in the reception processor 31.

Hereinafter, the operation of the present embodiment having the above configuration will be described.

(A-2) Operation of the First Embodiment FIG. 1 shows the processing of the communication part of the processor 30 for transmitting a message. FIG. 1 also shows the processing of the communication part of the processor 31 that receives the message. The flowchart of FIG. 1 includes steps S20 to S27.

The content of the processing in step S20 may be the same as that in step S1.

S21 may be the same as step S2, step S22 may be step S3, step S23 may be step S4, and normal termination S25 may be the same as step S5.

The network 9 and the processors 30 and 31

in the normal case where there is no problem in the operation such as the operation and no failure occurs, in an ideal case, this flowchart is usually composed of steps S20, S21, S22, S23 and S2.

4. The process proceeds in the order of S25 and ends normally.

At this time, in step S22, the transfer confirmation processing is performed only once (the transmission confirmation processing

The reception-side processor 31 performs a reception confirmation process), and confirms the completion of the transfer in that process.

It will branch to the (Yes) side.

However, in step S24 arranged between steps S23 and S25, transfer confirmation (S22)

if transfer completion cannot be confirmed at the time of

The process branches to n) and the process proceeds to step S26.

In the process of step S 26, the timeout monitoring unit 22 sends the time data 25 of the data structure 26

To determine the transfer start time, take the difference between the transfer start time and the current time, and determine whether the difference is equal to or greater than a preset timeout time.

If it is determined that the difference is equal to or longer than the timeout period, step S26 branches to the Y side and the process proceeds to step S27, if it is determined that the difference is less than the timeout period, the process proceeds again. The process proceeds to step S22. This is a procedure in anticipation of the possibility that transfer completion may be seen within a normal range within the timeout period even if transfer completion cannot be confirmed in the first transfer confirmation (S22).

In step S27, the transfer termination unit 23

Cancels the transfer request, that is, aborts the transfer process.

Although it depends on how long the timeout time is set, usually, even if the transfer completion cannot be confirmed by one transfer confirmation (S22), the transfer is not immediately terminated (S27). Steps S22, S23, S2

4. The loop composed of S26 is repeatedly processed,

If transfer completion is confirmed (S22) within the timeout period, the process ends with normal termination (S25), and if not, it ends with transfer termination (S27).

By terminating the transfer, the processor 3

The processing of the communication part by 0 and 31 may start the transfer processing of the next new message (the destination of this message may be a processor other than the processor 31) irrespective of the transfer completion of the message forever. It can.

Note that M shown in step S27 of FIG.

PI_Cancel is a function name indicating cancellation of a transfer request.

In the receiving processor 31, similarly to the transmitting processor 30, transfer confirmation (reception confirmation, reception completion wait) is repeated until the time is less than the timeout time, and transfer (reception) is terminated when the time exceeds the timeout time. Can receive a message other than the message (the source of the message may be a processor other than the processor 30).

(A-3) Effects of the First Embodiment According to the present embodiment, a non-blocking transfer instruction of an MPI function, which conventionally could not perform the next transfer processing until a certain transfer processing was completed, is executed. Even if it is used, the next transfer process can be performed after the timeout period, so that communication stagnation is prevented, smooth data (message) transfer is performed, overall communication efficiency is improved, and usability is improved.

The reliability of communication can be improved.

Further, in this embodiment, by performing timeout monitoring, it is possible to determine the normality of a transfer path such as a network or a processor of a communication partner.

(B) Second Embodiment (8-2) Configuration and Operation of Second Embodiment FIG. 10 shows the internal configuration of the transmitting processor 40 of the present embodiment, and FIG. 11 shows the internal configuration of the receiving processor 50 of the present embodiment.

FIG. 12 shows the internal configuration of FIG. 10. The processors 40 and 50 are included in the network 60 of the present embodiment shown in FIG. 10.

In FIG. 10, the processor 40 includes a transfer processing unit 41, a timeout monitoring unit 42, a transfer termination unit 43, and a retransmission processing unit 44.

The transfer processing section 41 corresponds to the transfer processing section 21 of the first embodiment. Similarly, the timeout monitoring unit 42 corresponds to the timeout monitoring unit 22, and the transfer termination unit 43 corresponds to the transfer termination unit 23.

Therefore, the substantial difference in operation between the processor 40 of the present embodiment and the processor 30 of the first embodiment is limited to the portion related to the retransmission processing unit 44.

The operation of the processor 40 is shown in the flowchart of FIG. 13. The flowchart of FIG. 13 is similar to the flowchart of FIG. 12.

Step S30 indicating transfer start (transmission start) corresponds to step S20 in the transmitting processor 30 of the first embodiment.

Similarly, step S31 is the same as step S31.

Step S32 corresponds to Step S22, Step S33 corresponds to Step S23, Step S34 corresponds to Step S24, Step S35 corresponds to Step S25, and Step S36 corresponds to step S26, and step S37 corresponds to step S27.

Therefore, in FIG. 13, only step S38 is a step unique to this embodiment.

After all, the processor 40 of the present embodiment and the first

in the configuration of the processor 30 of this embodiment, the substantial difference in operation is limited to the portion related to the retransmission processing section 44 and the retransmission process S38.

On the other hand, in FIG. 11,

has a transfer processing unit 51, a timeout monitoring unit 52, a transfer termination unit 53, a synchronous data check unit 54, and a discard processing unit 55.

The functions of the portions denoted by the same reference numerals as those in FIG. 10 are the same as those in FIG. 10. This is part 41, 42, and 43.

The operation of the processor 50 is shown in the flowchart of FIG. 14. The flowchart of FIG. 14 is similar to the flowchart of FIG. 13.

The step S40 indicating the transfer start (reception start) corresponds to step S20 in the reception-side processor 31 of the first embodiment.

Similarly, step S41 is performed in step S41.

Step S42 corresponds to step S22, step S43 corresponds to step S23, step S44 corresponds to step S24, step S45 corresponds to step S25, and step S46 corresponds to step S26. Step S47 corresponds to S26.

Corresponds to the step S27.

Therefore, in FIG. 14, only step S48 is a step unique to this embodiment.

After all, the processor 50 of the present embodiment and the first

in the configuration of the processor 31 of the third embodiment, the substantial difference in operation is limited to the synchronous data check unit 54, the discard processing unit 55, or the part related to the synchronous data check processing S48.

In the following, only differences between the present embodiment and the first embodiment will be described.

In the transmission-side processor 40, the retransmission processing section 44 operates to perform the retransmission processing in step S38 after the transfer termination processing in step S37 is performed.

That is, the difference between the transfer start (transmission start) time and the current time is calculated, and the difference is equal to or longer than a preset timeout time, and it is determined that a transfer timeout has occurred. This is the case.

When occurrence of a transfer timeout is recognized in the transmitting processor 40, it is highly likely that a normal message has not been received by the receiving processor 50 in the previous transfer, so that a message having the same content is re-transmitted by the retransmission processing S38. Transmit and ensure communication reliability.

On the other hand, in the receiving processor 50, the synchronous data check unit 54 operates to perform the synchronous data check processing in step S48.

This is to prepare for the reception of the message transmitted by the transmission-side processor 40 by S37.

In the present embodiment, a message (transfer text) ME transmitted from the transmitting processor 40 and received by the receiving processor 50 is, as shown in FIG. 15,

The value of the synchronous data SD having the structure with the added SD is changed based on a predetermined regularity every time the transfer processing unit 41 of the transmitting processor 40 transmits the message ME having the new transmission data DC. In the present embodiment, this change operation is assumed to be an increment (+1).

Therefore, when normal transmission and reception are performed, the synchronization data SD of the time-series message ME transmitted and received is sequentially incremented. However, when retransmission and reception thereof are performed, the synchronization data SD having the same value is obtained. The message ME having the SD is subsequently transmitted and received.

In the receiving processor 50 that receives these messages ME, the synchronous data check unit 54 stores at least the value of the synchronization data SD relating to the message ME received last time.

The value of the synchronous data SD of the message ME received this time is compared, and if they are the same, a coincidence signal indicating the coincidence of the synchronous data SD is transmitted to the discard processing unit 55.

The discard processing unit 55 that has received the coincidence signal discards the contents (data DC) of the previously received message ME stored in the transfer processing unit 41. Performing such a discarding process is likely to have a problem in the transfer route of the message ME when a timeout occurs.

This is because the possibility that a transmission error has occurred in the data DC is considered to be higher than when no timeout occurs.

in order to explain the operation of the present embodiment, a case where the transfer processing as shown in FIG. 7 is performed between the processors 40 and 50 will be described as an example.

In FIG. 7, in the transfer of the sequence SE1, the value of the synchronous data SD is set to "1" and the message ME is set. No timeout is detected by either the processor 40 on the transmission side or the processor 50 on the reception side.

In this case, the flowchart of FIG. 8 ends with the normal end S35, and the flowchart of FIG. 9 ends with the normal end S35. After 45, the synchronous data check processing S48 is performed and the processing ends. In the current synchronization data check process S48, a value different from the "1", for example, "0" is detected as the value of the synchronization data SD of the message ME received last time. The unit 52 does not operate.

Therefore, the transmission-side processor 40 starts transmission activation processing S30 for transmitting the next message ME. is performed, and the next message M
A reception start process S40 for receiving E is performed.

Next, in the message transfer of the sequence SE2-1, a timeout is detected on the transmission side for the transfer of the message ME in which the value of the synchronous data SD is "2", and the transfer termination processing S37 is performed.

At this time, since no timeout has been detected in the receiving processor 50, the synchronous data check processing S48 is performed following the normal end processing S45 as in the previous case, and the message ME with the value of the synchronous data SD of "2" is executed. Is received, and reception start processing S40 for receiving the next message ME is performed.

In the sequence SE2-2 following the sequence SE2-1, the retransmission processing unit 44 of the transmitting processor 40 functions to perform the retransmission processing S38 of the message ME whose synchronous data SD is "2".

The retransmission processing S38 itself has the same logical structure as the entire flow chart of FIG. 8, and if necessary, the retransmission processing may be repeatedly performed two or more times.

Assuming that the message ME retransmitted in step S38 is normally received on both the transmitting side and the receiving side without detecting a timeout, the receiving processor 50 checks the synchronous data after the normal end processing S45. In process S48, the message ME received this time Of the synchronous data SD is "2", which is the same as the previously received message ME, and the transmitting side performs retransmission processing S3. Recognizing that step 8 has been performed, the discard processing unit 52 is operated.

The discard processing unit 52 discards the message ME whose previously received synchronous data SD is "2", and treats the message ME whose received synchronous data SD is "2" as a genuine received message.

In the next sequence SE3-1, the timeout of the transfer of the message ME whose synchronous data SD is "3" has been detected by the receiving processor 50, so that the transmitting processor 40 has also successfully completed the transmission in step S33. Cannot be confirmed, and a timeout is detected on both the transmitting side and the receiving side.

At this time, the processing of the transmitting processor 40 ends with the transfer abort processing S37, and the processing of the receiving processor 50 also ends with the transfer abort processing S47.

In the sequence S3-2, the transmission-side processor 40 that has completed the previous message transmission in the transfer termination processing S37 executes the same retransmission processing S47 of the message ME whose synchronous data SD is "3" as the previous time.

This message transfer is normally completed on both the transmitting side and the receiving side without detecting a timeout (S3 B, S45), the synchronous data check processing S48 is executed on the receiving side, but the timeout has been detected on the receiving side the previous time, so that the synchronous data SD of "3" is received by the receiving side processor 50. Message M
This is the first reception of E.

Thereafter, the same operation as each sequence in FIG. 7 can be repeated.

(B-2) Effects of the Second Embodiment According to the present embodiment, the same effects as those of the first embodiment can be obtained.

In addition, in the present embodiment, by performing retransmission at the time of transfer timeout, it is possible to increase the reliability of data transfer in the sense that the frequency of occurrence of data (message) that cannot be transferred can be reduced. It is.

(C) Other Embodiments In the second embodiment, the retransmission processing S38 itself has the same logical structure as the entire flow chart of FIG. 8, and if necessary, repeats the retransmission processing at least twice. Is executed, but it is not necessary to do so.

For example, the message transmission is performed uniformly or once a predetermined number of times without performing the transfer confirmation S32 or the status check S33.
38.

Depending on the type of failure that occurs in the network, the processor, or the like, there may be a case where a timeout is detected even if retransmission is repeated many times. In such a case, repeating the retransmission itself will solve the invention. This is because there is a possibility that substantially the same problem as the problem such as communication stagnation or system stoppage described in the section of the problem will be caused.

In the first and second embodiments, the number of processors included in the network is two. However, it is natural that the number of processors can be increased to three or more. .

Further, the present invention can be applied to a standard in which the main part is the same as the MPI, but partially different.

That is, the present invention relates to a method for transmitting and / or receiving unit data using a communication function having a non-exclusive transfer instruction capable of executing an instruction other than the transfer instruction during execution of the transfer instruction. The data communication method and apparatus can be widely applied.

[0109]

As described above, according to the present invention, the communication efficiency can be improved, the usability can be improved, and the communication reliability can be improved while enjoying the advantages of the non-exclusive transfer instruction. .

[Brief description of the drawings]

FIG. 1 is a flowchart showing an operation of the first embodiment.

FIG. 2 is a flowchart showing the operation of a conventional processor on the transmitting or receiving side in a network.

FIG. 3 is a schematic diagram showing a configuration of a network according to the first and second embodiments.

FIG. 4 is a schematic diagram showing a configuration of a data structure used in the first and second embodiments.

FIG. 5 is a schematic diagram illustrating an internal configuration of a processor according to the first embodiment.

FIG. 6 is a schematic diagram illustrating a structure of a message in the second embodiment.

FIG. 7 is an operation explanatory diagram illustrating an inter-processor communication operation according to the second embodiment.

FIG. 8 is a flowchart illustrating an operation of a transmitting processor according to the second embodiment.

FIG. 9 is a flowchart illustrating an operation of a receiving processor according to the second embodiment.

FIG. 10 is a schematic diagram illustrating an internal configuration of a transmission-side processor according to the second embodiment.

FIG. 11 is a schematic diagram illustrating an internal configuration of a receiving processor according to the second embodiment.

[Explanation of symbols]

8, 9, 60: Network, 10, 11, 30, 3

1, 40, 50: Processor, 21, 41: Transfer processing unit, 22, 42: Timeout monitoring unit, 23, 43: Transfer termination unit, 44: Retransmission processing unit, 51:

Synchronous data check unit, 52: Discard processing unit, SD: Synchronous data, DC

(transmission) data, ME ... message.

Similar Documents

Continuing from the front page (72) Inventor Shigeru Minami 1-7-12 Toranomon, Minato-ku, Tokyo Oki Electric Industry Co., Ltd. (72) Inventor Akihiko Sugisawa 1-7-12

Toransawa, Minato-ku, Tokyo Oki Electric Industry Co., Ltd. F term in reference (reference) SB045 BB31 BB42 BB44 JJ05 JJ14 JJ45

US5907547B2	2005-06-14	Test tool and methods for testing a computer function employing a multi-system testcase
US6393497B1	2002-05-21	Downloadable smart proxies for performing processing associated with a remote procedure call in a distributed system
US7003777B2	2006-02-21	Coordination-centric framework for software design in a distributed environment
US8659492B2	2014-01-28	Accelerated execution for emulated environments
JP3055970B2	2000-06-26	Method and apparatus for implementing interface between object-oriented languages
JP200515278A	2000-11-14	Method and apparatus for invoking and processing objects asynchronously
WO1999044125A2	1999-09-02	Methods and apparatus for remote method invocation
KR20100066071A	2010-06-17	System and method for inter connection between component in using software bus
US8288938B2	2012-08-28	Method and system for reading data, related network and computer program; product therefor
Kraemer et al.	2006	Aligning UML 2.0 state machines and temporal logic for the efficient execution of services
JPH09218754A	1997-08-19	Method and device for dynamically boarding method calling exception code
JP2001229146A	2001-08-24	Method and device for data communication between processors
JP40630161B	1994-10-28	Remote procedure accessing method
CN111045797A	2020-04-21	Task scheduling execution method, related device and medium
KR20010041226A	2001-05-15	Downloadable smart proxies for performing processing associated with a remote procedure call in a distributed system
Akassar et al.	2009	Proving the correctness of client/server software
JPH06301685A	1994-10-28	Distributed processing system
JP4691982B2	2005-09-07	Agent system and information processing method
Li	2009	DynaQoS@-RDF: a best effort for QoS-assurance of dynamic reconfiguration of dataflow systems
Williamson et al.	1984	Concurrent communication and synchronization mechanisms
JPH0628322A	1994-02-04	Information processor
US10379918B2	2019-08-13	System and method for MPI implementation in an embedded operating system
CN110825455A	2020-02-21	Application program running method, device and system
KR100668376B1	2010-07-09	Device and method for processing application between different processor, and application processor(ap) communication system comprising the same device

KR20143742B1 1998-08-17 A method of communicating data for multiprocessor system

Priority And Related Applications

Priority Applications (1)

Application	Priority date	Filing date	Title
JP2000037878A	2000-02-16	2000-02-16	Method and device for data communication between processors

Applications Claiming Priority (1)

Application	Filing date	Title
JP2000037878A	2000-02-16	Method and device for data communication between processors

Legal Events

Date	Code	Title	Description
2006-02-01	A621	Written request for application examination	Free format text: JAPANESE INTERMEDIATE CODE: A621 Effective date: 20060131
2007-07-27	A977	Report on retrieval	Free format text: JAPANESE INTERMEDIATE CODE: A971007 Effective date: 20070727
2007-08-08	A131	Notification of reasons for refusal	Free format text: JAPANESE INTERMEDIATE CODE: A131 Effective date: 20070807
2007-12-05	A02	Decision of refusal	Free format text: JAPANESE INTERMEDIATE CODE: A02 Effective date: 20071204

Concepts

machine-extracted				Download	Filter table 
Name	Image	Sections	Count	Query match	
 biological transmission		claims,abstract,description	30	0.000	
 method		claims,description	35	0.000	
Show all concepts from the description section					

Data provided by IFI CLAIMS Patent Services

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-229145

(P2001-229145A)

(43) 公開日 平成13年8月24日 (2001.8.24)

(51) Int.Cl.⁷

G 0 6 F 15/177

識別記号

6 7 6

F I

G 0 6 F 15/177

テーマコード* (参考)

6 7 6 F 5 B 0 4 5

審査請求 未請求 請求項の数 4 O L (全 10 頁)

(21) 出願番号 特願2000-37878(P2000-37878)

(22) 出願日 平成12年2月16日 (2000.2.16)

(71) 出願人 591063084

株式会社沖システック東海

静岡県清水市庵原町1210

(71) 出願人 000000295

沖電気工業株式会社

東京都港区虎ノ門1丁目7番12号

(72) 発明者 芦川 秀樹

静岡県清水市庵原町1210 株式会社沖シス

テック東海内

(74) 代理人 100090620

弁理士 工藤 宣幸

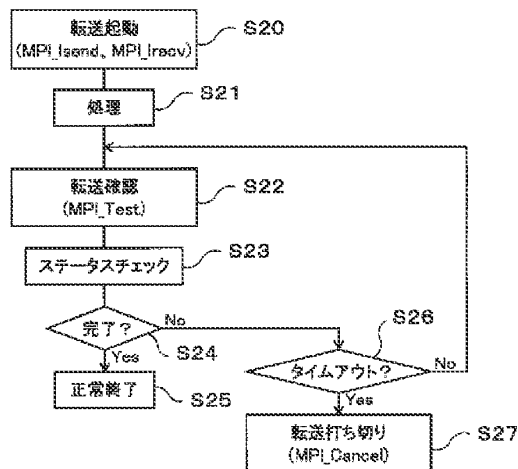
最終頁に続く

(54) 【発明の名称】 プロセッサ間データ通信方法及び装置

(57) 【要約】

【課題】 プロセッサ間データ通信装置の通信の信頼性を向上する。

【解決手段】 転送命令の実行中に当該転送命令以外の命令を実行し得る非排他的転送命令を備えた通信関数を使用して、単位データを送信及び／又は受信するプロセッサ間データ通信装置において、前記単位データの送信及び／又は受信に関してタイムアウトを監視するタイムアウト監視手段を備える。



【特許請求の範囲】

【請求項1】 転送命令の実行中に当該転送命令以外の命令を実行し得る非排他的転送命令を備えた通信関数を使用して、単位メッセージを送信及び／又は受信するプロセッサ間データ通信方法において、前記単位メッセージの送信及び／又は受信に関してタイムアウトを監視することを特徴とするプロセッサ間データ通信方法。

【請求項2】 前記単位メッセージが収容している情報内容に関する異同を受信時に識別させるための識別情報を、各単位メッセージに付加した請求項1のプロセッサ間データ通信方法であって、

送信時に前記タイムアウト監視手段によってタイムアウトが検出された場合、前記データを再送する再送処理、及び／又は、

受信時に、前記識別情報をもとに、受信した各単位メッセージの収容する情報内容の異同を調べ、同一の情報内容を収容した単位メッセージが複数受信された場合には、最後に受信された単位メッセージだけを真正な単位メッセージとして処理する受信データ選別処理を実行することを特徴とするプロセッサ間データ通信方法。

【請求項3】 転送命令の実行中に当該転送命令以外の命令を実行し得る非排他的転送命令を備えた通信関数を使用して、単位メッセージを送信及び／又は受信するプロセッサ間データ通信装置において、前記単位メッセージの送信及び／又は受信に関してタイムアウトを監視するタイムアウト監視手段を備えたことを特徴とするプロセッサ間データ通信装置。

【請求項4】 前記単位メッセージが収容している情報内容に関する異同を受信時に識別させるための識別情報を、各単位メッセージに付加した請求項3のプロセッサ間データ通信装置であって、

送信時に前記タイムアウト監視手段によってタイムアウトが検出された場合、前記データを再送する再送手段、及び／又は、

受信時に、前記識別情報をもとに、受信した各単位メッセージの収容する情報内容の異同を調べ、同一の情報内容を収容した単位メッセージが複数受信された場合には、最後に受信された単位メッセージだけを真正な単位メッセージとして処理する受信データ選別手段を備えたことを特徴とするプロセッサ間データ通信装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明はプロセッサ間データ通信方法に関し、例えば、MPI（メッセージパッシングインタフェース）関数のノンブロッキング転送命令を使用して複数のプロセッサ間でデータ転送を行なう場合などに使用し得るものである。

【0002】 また、本発明は、かかるプロセッサ間データ通信方法を使用するプロセッサ間データ通信装置に関

するものである。

【0003】

【従来の技術】 MPIは、MPIフォーラムで仕様検討されているメッセージパッシングライブラリで、Fortran言語または、C言語で記述されたプログラムからMPIの関数または、サブルーチンとして呼び出される。MPIには、1対1通信機能、集団通信（集配信／集積演算）機能、派生データ型機能、コミュニケータ（Communicator）機能、プロセストポロジ（Process Topology）機能、動的プロセス生成／管理機能、片側通信機能、拡張集団通信（集配信／集積演算）機能、並列入出力機能をサポートできる利点がある。

【0004】 このうちコミュニケータ機能では、これまでのメッセージパッシングライブラリでは実現困難であった多重通信空間を実現することができ、プロセストポロジ機能では、プログラマがプロセスを分かりやすい形で定義することができる。

【0005】 このようなMPI関数は、もう1つの特徴として、ノンブロッキング転送命令を装備している。

【0006】 ノンブロッキング転送命令は、例えば、膨大な演算量を要する演算処理を、ネットワーク中の複数のプロセッサで分担し、分散処理（あるいは並列処理）で、処理するケースなどに有利である。

【0007】 転送命令のノンブロッキング性によって、分散処理に関与する各プロセッサは、開始された転送命令が完全に終了する前であっても他の処理を実行し得るから、この他の処理に、上述した膨大な演算量を要する演算処理のなかで各プロセッサが分担する分の演算を当てはめると、各プロセッサは演算の実行中でもメッセージの転送処理を実行でき、メッセージの転送処理の実行中でも演算を実行することが可能になる。

【0008】 ノンブロッキング転送命令を実行する通信部分のみのフローチャートを図2(A)および(B)に示す。この転送命令では、図3に示すように、ネットワーク8中のプロセッサ10とプロセッサ11とのあいだのメッセージ転送が実行される。

【0009】 図2(A)はプロセッサ10における動作を示し、同図(B)はプロセッサ11における動作を示す。図2(A)のフローチャートはS1～S5の各ステップから構成され、同図(B)のフローチャートはS6～S9の各ステップから構成されている。

【0010】 プロセッサ10で、あるメッセージを送信（または受信）するため、図2(A)に示すように、例えば関数名MPI_Isend、MPI_Irecvに対応する転送起動が行われるとき（S1）、プロセッサ11でもそのメッセージを受信（または送信）するため、関数名MPI_Irecv、MPI_Isendに対応する転送起動が行われる（S6）。

【0011】 次に、プロセッサ10は当該ノンブロッキング転送命令とは別の処理、例えば上述した分担する演

算を実行し（S2）、同様にプロセッサ11も、当該ノンブロッキング転送命令とは別の処理、例えば上述した分担する演算を実行する（S7）。

【0012】そしてステップS2につづくステップS3では、プロセッサ10が例えば関数名MPI_Testに対応する転送確認を行い、ステップS7につづくステップS8ではプロセッサ11が例えば関数名MPI_Waitに対応する転送完了待ちを行う。

【0013】ステップS3の転送確認処理では、MPI_Testの関数をコールすると転送が完了したか否かのステータスがセットされ、関数呼び出し側に直ちにリターンする。もしもここで、MPI_Waitの関数をコールするものとする、転送が完了し次第、関数呼び出し側にリターンしてくる。

【0014】MPIは、MPIを使用するアプリケーションのためにアプリケーションインタフェースとMPIが用意すべき関数ライブラリのそれぞれの関数の機能を規定しているだけであるから、その規定に反しないかぎりにおいて、当該転送確認処理などの各処理の具体的な中身としてはさまざまなものが考えられるが、ここでは言及しない。

【0015】ネットワークや各プロセッサなどの動作に問題が無く、障害が発生しない正常時には、プロセッサ10はワンショットの転送確認（S3）につづいてステータスチェックを行い（S4）、通信部分に関して正常終了が行われる（S5）。この正常終了のあとでも、前記分担する演算などの処理は、引き続き実行され得る。

【0016】なお、転送確認（S3）のあとにステータスチェック（S4）を行うのは、転送確認（S3）では転送が単に終了したかどうか出力されるだけなので、その終了が正常に行われたかどうかを認識するために、当該ステータスチェックによってステータスの中身を確認する必要があるためである。

【0017】同様に、前記正常時には、プロセッサ11でも、短時間の転送完了待ち（S8）につづいて通信部分に関する正常終了が行われる（S9）。

【0018】引き続き、前記メッセージとは別なメッセージを転送する場合には、図2（A）、（B）のフローチャートと同様な処理が、各プロセッサ10、11で繰り返されることになる。

【0019】もしも、転送命令などがブロッキング性であった場合を想定すると、各プロセッサは分担している分の演算が終了するまで、当該演算のために完全にハードウェア資源を占有されてしまうために、その他の処理をいっさい実行できなくなる。すなわち分担する分の演算が完全に終了するまで、転送命令を実行することができず、転送命令が完全に終了するまで、演算を実行することができない。

【0020】このことから、ノンブロッキング転送命令の有用性が見える。

【0021】

【発明が解決しようとする課題】ところで、ネットワークや各プロセッサなどの動作に問題があり、障害が発生する異常時には、前記プロセッサ10の行うステップS3の転送確認はワンショット、すなわち1回だけではすまなくなり、前記転送完了待ち（S8）も短時間ではすまなくなることがある。

【0022】MPI関数のノンブロッキング転送にはタイムアウトの概念が無く、転送の時間管理および時間監視を行なうことが出来ないため、最悪の状況では、前記ステップS3の転送確認が際限なく繰り返されるとともに、前記ステップS8の転送完了待ちが無制限に長時間化する事態も起こり得る。

【0023】この場合、転送が完了するまで次の転送は待たされた状態になって通信が停滞してしまう。また、転送完了待ち（S8）については、何らかの障害により終了ステータスを得ることができなかった時はプログラムが停止した状態になり、システムが停止してしまう。

【0024】結局、このような問題のあるノンブロッキング転送命令は、上述した大きな利点を持っている反面、全体として通信効率が低く、使い勝手が悪く、信頼性が低いといえることができる。

【0025】

【課題を解決するための手段】かかる課題を解決するために、第1の発明では、転送命令の実行中に当該転送命令以外の命令を実行し得る非排他的転送命令を備えた通信関数を使用して、単位メッセージを送信及び／又は受信するプロセッサ間データ通信方法において、前記単位メッセージの送信及び／又は受信に関してタイムアウトを監視することを特徴とする。

【0026】また、第2の発明では、転送命令の実行中に当該転送命令以外の命令を実行し得る非排他的転送命令を備えた通信関数を使用して、単位メッセージを送信及び／又は受信するプロセッサ間データ通信装置において、前記単位メッセージの送信及び／又は受信に関してタイムアウトを監視するタイムアウト監視手段を備えたことを特徴とする。

【0027】

【発明の実施の形態】（A）実施形態

以下、本発明のプロセッサ間データ通信方法および装置に関する第1および第2の実施形態について説明する。

【0028】（A-1）第1の実施形態の構成

本実施形態のプロセッサ30の内部構成を図5に示す。このプロセッサ30は、上述した図3のプロセッサ10に対応するプロセッサであり、対向するプロセッサ31と通信する。プロセッサ30と31とを含むネットワークをネットワーク9とする。

【0029】図5において、プロセッサ30は、転送処理部21と、タイムアウト監視部22と、転送打ち切り部23とを備えている。

【0030】このうち転送処理部21は、上述した図2（A）のフローチャートと同様な送信側のノンブロッキング転送動作を実行する機能を備えた部分である。

【0031】なお、送信側で図2（A）の処理を行い、受信側で図2（B）の処理を行うのは、分散処理を行い各処理が入力ドリブンで動作するシステムの一例であって、MPI規格上は、必ずしもその必要はない。

【0032】MPIのノンブロッキング転送命令を行う場合、起動（MPI_Isend、MPI_Irecv）とその転送の完了（MPI_TestまたはMPI_Waitのどちらか）を対にして行う必要があるだけで、MPI_Isendの転送確認には、MPI_Testを用いてもよく、MPI_Waitを用いてもよい。

【0033】同様に、MPI_Irecvに対しては、MPI_Waitによる転送完了待ちだけでなく、MPI_Testによる完了確認を行ってもよい。ただしMPI_Testを使う場合は、単純に転送の結果を問い合わせているのに等しいため、転送が完了するまで問い合わせ（すなわちMPI_Testのコール）を行わなければならない。

【0034】また、タイムアウト監視部22は、ステップS22（図1参照）の転送確認（この処理は、前記ステップS3の転送確認と同じ）時に転送の完了が確認できなかった場合に、タイムアウトチェック処理を実行する部分である。

【0035】タイムアウトチェック処理では、転送開始時刻と現在時刻との差分を取り、この差分があらかじめ設定されているタイムアウト時間以上となったときには、転送タイムアウト発生と判定する。転送タイムアウト発生と判定した場合、このタイムアウト監視部22は、タイムアウト検出信号TDを転送打ち切り部23に出力する。

【0036】なお、タイムアウト監視部22が前記差分を取るために必要な転送開始時刻は、図4に示すデータ構造体26に含まれる時刻データ25を調べることによって得られる。

【0037】図4のデータ構造体26は、同一プロセッサ内のデータ転送起動処理とデータ転送確認処理の間で、転送起動、確認の対応付けを行うために受け渡されるハンドシェイク構造体（ここで、「ハンドシェイク」は、同一プロセッサ内のプログラム間のハンドシェイクを示す）で、当該リクエスト部24と、時刻データ部25とから構成されている。

【0038】このうちリクエスト部24は、MPI_IsendやMPI_IrecvなどのMPI関数の引数として用いられるデータバッファアドレス、転送語数（通信語数）、データ型、転送先アドレス（MPIランク）、通信要求ハンドルなどを含んだ部分である。

【0039】MPI関数を使用した通信においては、各

通信ごとに通信相手および通信語数を指定する必要があるため、各プロセッサ30、31は通信相手のアドレスや通信語数を当該通信に先立って認識していなければならないが、この認識は、MPIを使用するアプリケーションが意識するものではなく、MPI関数によるメッセージ転送以外の方法によって実現されるべきものである。

【0040】例えば、プロセッサ30からプロセッサ31へメッセージを送信する場合などには、当該メッセージ送信に先立って行われるプロセッサ間のハンドシェイクにより、送信元であるプロセッサ30側の送信関数では、送信先であるプロセッサ31のアドレスと、当該メッセージの通信語数を指定し、送信先となるプロセッサ31側の受信関数では送信元であるプロセッサ30のアドレスと当該メッセージの通信語数を指定しておく必要があるが、このプロセッサ間ハンドシェイクは通常のMPI関数によるメッセージ転送以外の方法で行うことが必要であると考えられる。

【0041】なぜなら、このプロセッサ間ハンドシェイクまでMPI通信で行うとすると、当該プロセッサ間ハンドシェイクの前提となる通信相手のアドレスや通信語数を伝えるために、さらに先行するMPI通信が必要になるという矛盾が生じるからである。

【0042】したがって、プロセッサ間のハンドシェイク等の制御用通信の具体例については、MPI規格の全範囲をそのままカバーしようとする本実施形態の趣旨に反するので、ここでは詳述しない。

【0043】図5のプロセッサ30の構成要素のうち転送処理部21は、前記リクエスト部24から前記引数を取り出してMPI関数にセットし、転送を起動する部分である。

【0044】また、タイムアウト監視部22は、転送処理部21が引数を取り出したリクエスト部24に、当該引数に対応した通信の開始時刻を示す時刻データ25を付加することにより、データ構造体26を生成し記憶しておく機能を備えている。

【0045】前記転送打ち切り部23は、タイムアウト監視部22からタイムアウト検出信号TDを受け取ることによって、転送打ち切り信号CLを出力し、プロセッサ30に転送処理（メッセージ送信処理）の進行を打ち切らせる部分である。これにより前記ステップS3のように転送確認が際限なく繰り返されて通信が停滞することを防止できる。

【0046】なお、本実施形態の場合、受信側のプロセッサ31が装備している機能も、図5に示した送信側のプロセッサ30と同じであるものとする。したがってプロセッサ31の内部構成も図5に示した構成と同じである。

【0047】ただ、転送処理部21が行う転送処理の内容が、送信側プロセッサ30においては送信処理である

のに対し、受信側プロセッサ31では受信処理である点が相違するだけである。

【0048】以下、上記のような構成を有する本実施形態の動作について説明する。

【0049】(A-2)第1の実施形態の動作
メッセージを送信するプロセッサ30の通信部分の処理を、図1に示す。図1はまた、当該メッセージを受信するプロセッサ31の通信部分の処理をも示している。図1のフローチャートは、S20～S27の各ステップから構成されている。

【0050】このうちステップS20の処理内容は前記ステップS1と同じであってよく、同様に、ステップS21は前記ステップS2と、ステップS22は前記ステップS3と、ステップS23は前記ステップS4と、正常終了S25は前記ステップS5とそれぞれ同じ処理内容であってよい。

【0051】ネットワーク9や各プロセッサ30、31などの動作に問題が無く、障害が発生しない正常時のうち、理想的なケースでは、通常、このフローチャートは、ステップS20、S21、S22、S23、S24、S25の順番に進行して正常に終了する。

【0052】このときステップS22では、1回だけ転送確認処理(送信側プロセッサ30では送信確認処理、受信側プロセッサ31では受信確認処理)が行われ、その処理で転送の完了が確認されて、ステップS24はY(Yes)側に分岐することになる。

【0053】ただし、ステップS23とS25のあいだに配置されたステップS24では、転送確認(S22)の時点で、転送完了の確認が行えなかった場合、N(No)側に分岐して処理をステップS26に進める。

【0054】ステップS26の処理では、タイムアウト監視部22が、前記データ構造体26の時刻データ25を調べて転送開始時刻を認識し、その転送開始時刻と現在時刻との差分を取って、この差分があらかじめ設定されているタイムアウト時間以上であるかどうか判定される。

【0055】当該差分が、タイムアウト時間以上であると判定された場合には、ステップS26はY側に分岐してステップS27に処理を進め、タイムアウト時間未満であると判定された場合には再度、ステップS22に処理を進める。これは、1回目の転送確認(S22)で転送完了の確認が行えなくても、タイムアウト時間内であれば、正常範囲内に転送の完了をみる可能性があることを見越した手順である。

【0056】ステップS27では、転送打ち切り部23が、転送要求のキャンセル、すなわち転送処理の打ち切りを行う。

【0057】前記タイムアウト時間をどの程度に設定するかにもよるが、通常は、1回の転送確認(S22)で転送の完了が確認できなくても、ただちに転送打ち切り

(S27)とはせず、ステップS22、S23、S24、S26から構成されるループを繰り返して処理して、タイムアウト時間内に転送完了が確認(S22)できれば正常終了(S25)で終わり、できなければ転送打ち切り(S27)で終わることになる。

【0058】転送打ち切りとすることで、プロセッサ30、31による通信部分の処理は、いつまでも当該メッセージの転送完了に拘泥せず、次の新たなメッセージ(このメッセージの送信先はプロセッサ31以外のプロセッサであってよい)の転送処理に取り掛かることができる。

【0059】なお、図1のステップS27中に示したMPI_Cancelは、転送要求のキャンセルを示す関数名である。

【0060】受信側プロセッサ31でもこの送信側プロセッサ30と同様に、タイムアウト時間未満は転送確認(受信確認、受信完了待ち)が繰り返されて、タイムアウト時間以上になると転送(受信)打ち切りとされ、以降は、当該メッセージ以外のメッセージ(このメッセージの送信元はプロセッサ30以外のプロセッサであってよい)の受信を行うことができる。

【0061】(A-3)第1の実施形態の効果
本実施形態によれば、従来は、ある転送処理が完了するまで次の転送処理を行うことができなかったMPI関数のノンブロッキング転送命令を用いても、タイムアウト時間以降は次の転送処理を行うことができるため、通信の停滞を防止して円滑なデータ(メッセージ)転送を行ない、全体として通信効率を高め、使い勝手を良くし、通信の信頼性を向上することができる。

【0062】また、本実施形態では、タイムアウト監視を行なうことにより、ネットワークや通信相手のプロセッサなどの転送経路について、正常性の判断を行なうことができる。

【0063】(B)第2の実施形態

(B-2)第2の実施形態の構成および動作

図10に、本実施形態の送信側プロセッサ40の内部構成を示し、図11に本実施形態の受信側プロセッサ50の内部構成を示す。プロセッサ40と50は、図3に示す本実施形態のネットワーク60に含まれている。

【0064】図10において、プロセッサ40は、転送処理部41と、タイムアウト監視部42と、転送打ち切り部43と、再送処理部44とを備えている。

【0065】このうち、転送処理部41は第1の実施形態の転送処理部21に対応する。同様に、タイムアウト監視部42は前記タイムアウト監視部22に対応し、転送打ち切り部43は前記転送打ち切り部23に対応する。

【0066】したがって本実施形態のプロセッサ40と第1の実施形態のプロセッサ30との構成上、動作上の実質的な相違点は、再送処理部44に関連する部分に限

られる。

【0067】このプロセッサ40の動作は、図8のフローチャートに示す。図8のフローチャートはS30～S38の各ステップから構成されているが、転送起動（送信起動）を示すステップS30は第1の実施形態の送信側プロセッサ30におけるステップS20に対応する。

【0068】同様に、ステップS31は前記ステップS21に対応し、ステップS32は前記ステップS22に対応し、ステップS33は前記ステップS23に対応し、ステップS34は前記ステップS24に対応し、ステップS35は前記ステップS25に対応し、ステップS36は前記ステップS26に対応し、ステップS37は前記ステップS27に対応する。

【0069】したがって図8中では、ステップS38だけが本実施形態に特有なステップである。

【0070】結局、本実施形態のプロセッサ40と第1の実施形態のプロセッサ30との構成上、動作上の実質的な相違点は、再送処理部44、再送処理S38に関連する部分に限られる。

【0071】一方、図11において、プロセッサ50は、転送処理部41と、タイムアウト監視部42と、転送打ち切り部43と、同期データチェック部51と、廃棄処理部52とを備えている。

【0072】このうち、図10と同じ符号を付した部分の機能は図10の各部と同じである。これは、部分41、42、43の各部である。

【0073】このプロセッサ50の動作は、図9のフローチャートに示す。図9のフローチャートはS40～S48の各ステップから構成されているが、転送起動（受信起動）を示すステップS40は第1の実施形態の受信側プロセッサ31におけるステップS20に対応する。

【0074】同様に、ステップS41は前記ステップS21に対応し、ステップS42は前記ステップS22に対応し、ステップS43は前記ステップS23に対応し、ステップS44は前記ステップS24に対応し、ステップS45は前記ステップS25に対応し、ステップS46は前記ステップS26に対応し、ステップS47は前記ステップS27に対応する。

【0075】したがって図9中では、ステップS48だけが本実施形態に特有なステップである。

【0076】結局、本実施形態のプロセッサ50と第1の実施形態のプロセッサ31との構成上、動作上の実質的な相違点は、同期データチェック部51と、廃棄処理部52、あるいは、同期データチェック処理S48に関連する部分に限られる。

【0077】以下では、本実施形態が第1の実施形態と相違する点についてのみ説明する。

【0078】送信側プロセッサ40において、再送処理部44が動作してステップS38の再送処理が行われるのは、ステップS37の転送打ち切り処理が行われたあ

とである。

【0079】すなわち、転送開始（送信起動開始）時刻と現在時刻との差分を取り、この差分があらかじめ設定されているタイムアウト時間以上となり、転送タイムアウト発生と判定されステップS37でY側の分岐が行われた場合である。

【0080】送信側プロセッサ40において転送タイムアウトの発生が認められる場合、前回の転送では正常なメッセージが受信側プロセッサ50に受信されていない可能性が高いため、再送処理S38によって同じ内容のメッセージを再度送信し、通信の信頼性を確保する。

【0081】一方、受信側プロセッサ50において、同期データチェック部51が動作してステップS48の同期データチェック処理が行われるのは、この再送処理S37によって送信側プロセッサ40が送信したメッセージの受信に備えるためである。

【0082】本実施形態で、送信側プロセッサ40から送信され、受信側プロセッサ50で受信されるメッセージ（転送テキスト）MEは、図6に示すように、転送目的の送信データDCに対し、同期データSDを付加した構造を持っている同期データSDの値は、送信側プロセッサ40の転送処理部41が新たな送信データDCを持つメッセージMEを送信するたびに所定の規則性に基づいて変更操作する。本実施形態ではこの変更操作は、インクリメント（+1）であるものとする。

【0083】したがって通常の送信、受信が行われるときは送受信される時系列なメッセージMEの同期データSDは、逐次、インクリメントされて行くが、再送およびその受信が行われると、同じ値の同期データSDを持つメッセージMEがつづけて送信され、受信されることになる。

【0084】これらのメッセージMEを受信する受信側プロセッサ50では、同期データチェック部51が、少なくとも前回に受信したメッセージMEに関する同期データSDの値を記憶していて、当該前回のSDの値と、今回受信したメッセージMEの同期データSDの値とを比較し、同じであれば同期データSDの一致を示す一致信号を、廃棄処理部52に送信する。

【0085】一致信号を受信した廃棄処理部52は、転送処理部41が記憶している前回に受信したメッセージMEの内容（データDC）を廃棄処理する。このような廃棄処理を行うのは、タイムアウトが生じた場合、当該メッセージMEの転送経路に問題がある可能性が高く、データDCに伝送誤りが生じている可能性も、タイムアウトが生じない場合よりも高いと考えられるからである。

【0086】本実施形態の動作を説明するために、プロセッサ40と50のあいだで図7に示すような転送処理が行われる場合を例に取る。

【0087】図7において、シーケンスSE1の転送で

は、同期データSDの値を「1」としてメッセージMEの転送を行い送信側のプロセッサ40でも受信側のプロセッサ50でもタイムアウトを検出ししない。

【0088】この場合、図8のフローチャートは正常終了S35で終わり、図9のフローチャートは正常終了S45のあとで同期データチェック処理S48を行って終わる。今回の同期データチェック処理S48では、前回に受信したメッセージMEの同期データSDの値としては、当該「1」と異なる値、例えば「0」が検出されるので、受信側プロセッサ50内で廃棄処理部52が動作

【0089】したがって、送信側プロセッサ40では次のメッセージMEを送信するための送信起動処理S30が行われ、受信側プロセッサ50では次のメッセージMEを受信するための受信起動処理S40が行われる。

【0090】次いで、シーケンスSE2-1のメッセージ転送では、同期データSDの値を「2」としたメッセージMEの転送につき、送信側でタイムアウトが検出されて、転送打ち切り処理S37が行われる。

【0091】このとき受信側プロセッサ50では、タイムアウトを検出していないため、前回同様、正常終了処理S45につづいて同期データチェック処理S48が行われ、同期データSDの値が「2」のメッセージMEが受信され、次のメッセージMEを受信するための受信起動処理S40が行われる。

【0092】当該シーケンスSE2-1につづくシーケンスSE2-2では、送信側プロセッサ40の再送処理部44が機能して、同期データSDの値が「2」のメッセージMEの再送処理S38を行う。

【0093】再送処理S38そのものも、図8のフローチャート全体と同様な論理構造を持っていて、必要ならば2回以上繰り返して再送処理が実行されることもある。

【0094】このステップS38で再送されたメッセージMEは、送信側でも受信側でもタイムアウトを検出することなく正常に受信されるものとする、受信側プロセッサ50では、正常終了処理S45につづく同期データチェック処理S48で、今回受信したメッセージMEの同期データSDの値が前回受信したメッセージMEと同じ「2」であることを検出し、送信側で再送処理S38が行われたことを認識して、廃棄処理部52を動作させる。

【0095】廃棄処理部52は、前回受信した同期データSDが「2」のメッセージMEを廃棄し、今回受信した同期データSDが「2」のメッセージMEを真正な受信メッセージとして取り扱う。

【0096】次のシーケンスSE3-1では、同期データSDが「3」のメッセージMEの転送について、受信側プロセッサ50でタイムアウトを検出したため、送信側プロセッサ40でも、ステップS33で送信が正常に

終了したことを確認することができず、送信側と受信側の双方でタイムアウトを検出する。

【0097】このとき、送信側プロセッサ40の処理は、転送打ち切り処理S37によって終わり、受信側プロセッサ50の処理も、転送打ち切り処理S47によって終わる。

【0098】シーケンスS3-2では、転送打ち切り処理S37で前回のメッセージ送信を終えた送信側プロセッサ40が、前回と同じ同期データSDが「3」のメッセージMEの再送処理S47を実行する。

【0099】このメッセージ転送は、送信側でも受信側でもタイムアウトを検出することなく正常に終了（S35、S45）するものとする、受信側では同期データチェック処理S48が実行されるが、前回は受信側でもタイムアウトが検出されているため、受信側プロセッサ50にとって、同期データSDが「3」のメッセージMEの受信はこれが最初となる。

【0100】以降も、図7の各シーケンスと同様な動作が繰り返され得る。

【0101】（B-2）第2の実施形態の効果
本実施形態によれば、第1の実施形態と同様な効果を得ることができる。

【0102】加えて、本実施形態では、転送のタイムアウト時に再送を行なうことにより、転送することができないデータ（メッセージ）の発生頻度を低下できるという意味で、データ転送の信頼性を高めることが可能である。

【0103】（C）他の実施形態

なお、第2の実施形態では、再送処理S38自体が、図8のフローチャート全体と同様な論理構造を持っていて、必要ならば2回以上繰り返して再送処理が実行されるものとしたが、必ずしもそのようにする必要はない。

【0104】例えば、転送確認S32やステータスチェックS33などを行わず、画一的に、1回または所定回数だけ、メッセージ送信を行うことを、当該再送処理S38の内容としてもよい。

【0105】ネットワークやプロセッサなどに発生する障害の形態によっては、何度再送を繰り返してもタイムアウトを検出してしまうケースも考えられ、その場合には当該再送を繰り返すこと自体が、発明が解決しようとする課題の欄で述べた通信の停滞やシステムの停止などの問題と実質的に同じ問題をもたらす可能性があるからである。

【0106】また、第1および第2の実施形態では、ネットワーク中に含まれているプロセッサの数は2つであったが、3つ以上のプロセッサ数にも対応することができることは当然である。

【0107】さらに、本発明は、MPIと主要部が同じで、部分的に相違するような規格にも適用することが可能である。

【0108】すなわち本発明は、転送命令の実行中に当該転送命令以外の命令を実行し得る非排他的転送命令を備えた通信関数を使用して、単位データを送信及び／又は受信するプロセッサ間データ通信方法および装置について、広く適用することができる。

【0109】

【発明の効果】以上に説明したように、本発明によれば、非排他的転送命令の利点を享有しつつ、通信効率を高め、使い勝手を良くし、通信の信頼性を向上することができる。

【図面の簡単な説明】

【図1】第1の実施形態の動作を示すフローチャートである。

【図2】従来のネットワーク中の送信または受信側のプロセッサの動作を示すフローチャートである。

【図3】従来、第1および第2の実施形態のネットワークの構成を示す概略図である。

【図4】第1および第2の実施形態で使用するデータ構造体の構成を示す概略図である。

【図5】第1の実施形態のプロセッサの内部構成を示す

* 概略図である。

【図6】第2の実施形態でメッセージの構造を示す概略図である。

【図7】第2の実施形態のプロセッサ間通信動作を示す動作説明図である。

【図8】第2の実施形態の送信側プロセッサの動作を示すフローチャートである。

【図9】第2の実施形態の受信側プロセッサの動作を示すフローチャートである。

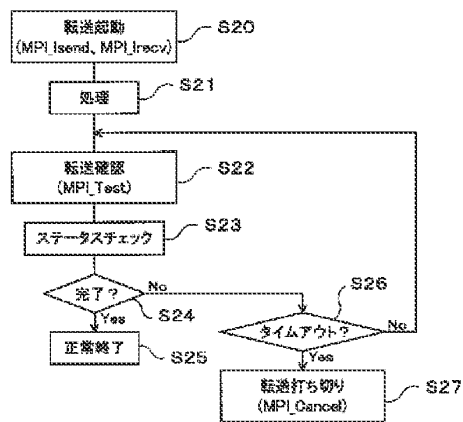
10 【図10】第2の実施形態の送信側プロセッサの内部構成を示す概略図である。

【図11】第2の実施形態の受信側プロセッサの内部構成を示す概略図である。

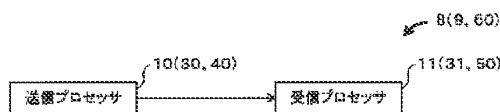
【符号の説明】

8, 9, 60…ネットワーク、10, 11, 30, 31, 40, 50…プロセッサ、21, 41…転送処理部、22, 42…タイムアウト監視部、23, 43…転送打ち切り部、44…再送処理部、51…同期データチェック部、52…廃棄処理部、SD…同期データ、DC…(送信)データ、ME…メッセージ。

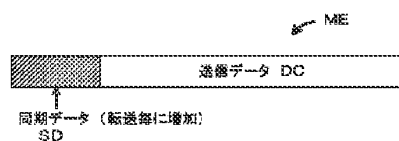
【図1】



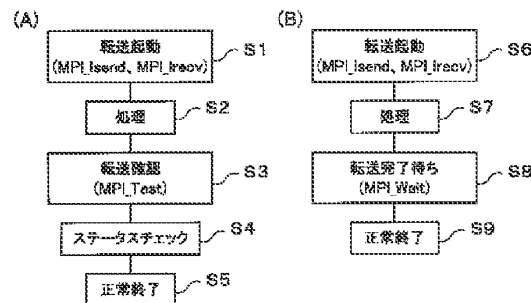
【図3】



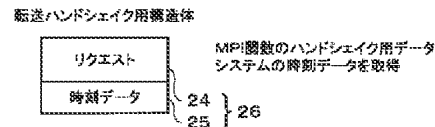
【図6】



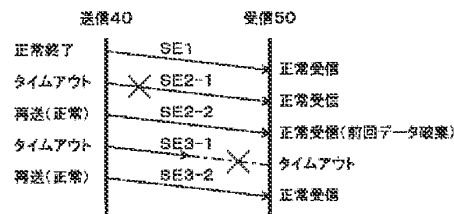
【図2】



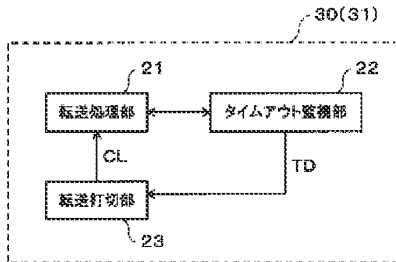
【図4】



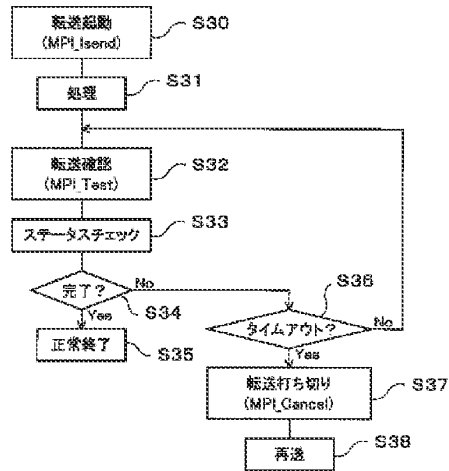
【図7】



【図5】

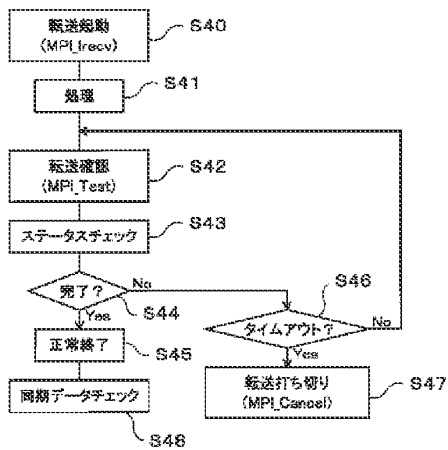
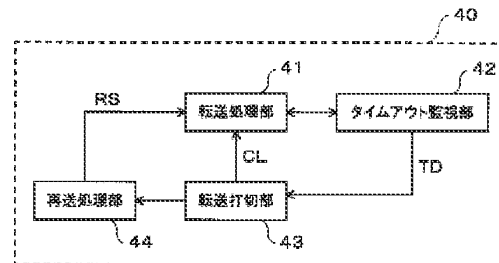


【図8】

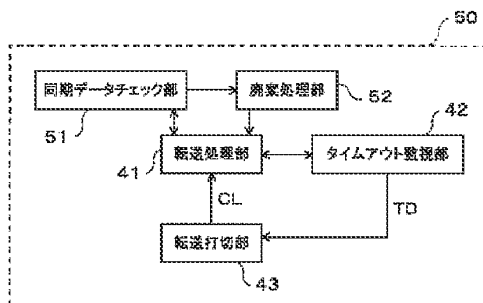


【図9】

【図10】



【図11】



フロントページの続き

(72)発明者 民實 茂
東京都港区虎ノ門1丁目7番12号 沖電気
工業株式会社内

(72)発明者 杉沢 彰彦
東京都港区虎ノ門1丁目7番12号 沖電気
工業株式会社内
Fターム(参考) 5B045 BB31 BB42 BB44 JJ05 JJ14
JJ45

Electronic Patent Application Fee Transmittal				
Application Number:		17374864		
Filing Date:		13-Jul-2021		
Title of Invention:		CLUSTER COMPUTING		
First Named Inventor/Applicant Name:		Zvi Tannenbaum		
Filer:		Jordan Michael Cox/Janet Teeters		
Attorney Docket Number:		ZTANN.002P1C6		
Filed as Small Entity				
Filing Fees for Utility under 35 USC 111(a)				
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	2806	1	130	130
Total in USD (\$)				130

Electronic Acknowledgement Receipt	
EFS ID:	44321921
Application Number:	17374864
International Application Number:	
Confirmation Number:	1083
Title of Invention:	CLUSTER COMPUTING
First Named Inventor/Applicant Name:	Zvi Tannenbaum
Customer Number:	20995
Filer:	Jordan Michael Cox/Fabiola Esmerio
Filer Authorized By:	Jordan Michael Cox
Attorney Docket Number:	ZTANN.002P1C6
Receipt Date:	19-NOV-2021
Filing Date:	13-JUL-2021
Time Stamp:	18:44:19
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$130
RAM confirmation Number	E2021AII44513120
Deposit Account	111410
Authorized User	Fabiola Esmerio
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: 37 CFR 1.16 (National application filing, search, and examination fees) 37 CFR 1.17 (Patent application and reexamination processing fees)	

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		ZTANN-002P1C6_IDS.pdf	46254	yes	3
			a4a80eebedff2125ac2fae19f7df0554d41218d3		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Transmittal Letter		1	2	
	Information Disclosure Statement (IDS) Form (SB08)		3	3	
Warnings:					
Information:					
2	Foreign Reference	JP_2001-229145_wMT.pdf	749751	no	18
			8de501e21234c3fce8fa19e491999604c51e40f		
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	37812	no	2
			d0df9d8bf447f3e86e8db8882eb142f70cf7afa4		
Warnings:					
Information:					
Total Files Size (in bytes):			833817		

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.

INFORMATION DISCLOSURE STATEMENT

First Inventor: Tannenbaum, Zvi	Conf. No.: 1083
App. No.: 17/374864	Filed: July 13, 2021
Examiner: Asres, Hermon	Art Unit: 2449
Title: CLUSTER COMPUTING	

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

References and Listing

Pursuant to 37 CFR 1.56, an Information Disclosure Statement listing references is provided herewith. Copies of any listed foreign and non-patent literature references are being submitted. Any foreign references may also include English abstract(s) and/or machine translation(s), but no representation is made as to their accuracy.

Pursuant to 37 C.F.R § 1.97(g) and (h), Applicant makes no representation that the information is considered to be material to patentability. Additionally, inclusion on this list is not an admission that any of the cited documents are prior art in this application. Further, Applicant makes no representation that better art does not exist.

If the Examiner would like additional information regarding these references or if anything is unclear, the Examiner is invited to contact the undersigned for assistance.

No Disclaimers

To the extent that anything in the Information Disclosure Statement or the listed references could be construed as a disclaimer of any subject matter supported by the present application, Applicant hereby rescinds and retracts such disclaimer.

Timing of Disclosure

This Information Disclosure Statement is being filed after the mailing date of a final action, Notice of Allowance, or other action that closes prosecution.

CERTIFICATION UNDER 37 CFR 1.97(e)(2)

I hereby certify that no item of information contained in this Statement was cited in a communication from a foreign Patent Office in a counterpart foreign application, and, to my knowledge after making reasonable inquiry, was known to any individual designated in 37 CFR 1.56(c) more than three (3) months prior to the filing of this Information Disclosure Statement.

Docket No.: ZTANN.002P1C6
App. No.: 17/374864

Page 2 of 2

This Statement is accompanied by the fee set forth in 37 CFR 1.17(p). The Commissioner is authorized to charge any additional fees which may be required or to credit any overpayment to Account No. 11-1410.

Respectfully submitted,
KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: November 18, 2021

By: /Jordan M. Cox/
Jordan M. Cox
Registration No. 70,875
Registered Practitioner
(949) 760-0404

54613371



UNITED STATES PATENT AND TRADEMARK OFFICE

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

NOTICE OF ALLOWANCE AND FEE(S) DUE

20995 7590 11/18/2021
KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

EXAMINER

ASRES, HERMON

ART UNIT

PAPER NUMBER

2449

DATE MAILED: 11/18/2021

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/374,864	07/13/2021	Zvi Tannenbaum	ZTANN.002PIC6	1083

TITLE OF INVENTION: CLUSTER COMPUTING

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	\$600	\$0.00	\$0.00	\$600	02/18/2022

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

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

HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.

If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.

If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".

For purposes of this notice, small entity fees are 1/2 the amount of undiscounted fees, and micro entity fees are 1/2 the amount of small entity fees.

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

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

IMPORTANT REMINDER: Maintenance fees are due in utility patents issuing on applications filed on or after Dec. 12, 1980. It is patentee's responsibility to ensure timely payment of maintenance fees when due. More information is available at www.uspto.gov/PatentMaintenanceFees.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), by mail or fax, or via EFS-Web.

By mail, send to: Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

By fax, send to: (571)-273-2885

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

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

20995 7590 11/18/2021
KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

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

Certificate of Mailing or Transmission

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

(Typed or printed name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/374,864	07/13/2021	Zvi Tannenbaum	ZTANN.002P1C6	1083

TITLE OF INVENTION: CLUSTER COMPUTING

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	\$600	\$0.00	\$0.00	\$600	02/18/2022

EXAMINER	ART UNIT	CLASS-SUBCLASS
ASRES, HERMON	2449	709-223000

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

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

☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-09 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

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

1 _____

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

2 _____

3 _____

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

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document must have been previously recorded, or filed for recordation, as set forth in 37 CFR 3.11 and 37 CFR 3.81(a). Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

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

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

4a. Fees submitted: ☐ Issue Fee ☐ Publication Fee (if required) ☐ Advance Order - # of Copies _____

4b. Method of Payment: (Please first reapply any previously paid fee shown above)

☐ Electronic Payment via EFS-Web ☐ Enclosed check ☐ Non-electronic payment by credit card (Attach form PTO-2038)

☐ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment to Deposit Account No. _____

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

☐ Applicant certifying micro entity status. See 37 CFR 1.29

☐ Applicant asserting small entity status. See 37 CFR 1.27

☐ Applicant changing to regular undiscounted fee status.

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____



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.
17/374,864	07/13/2021	Zvi Tannenbaum	ZTANN.002PIC6	1083
20995	7590	11/18/2021	EXAMINER	
KNOBBE MARTENS OLSON & BEAR LLP			ASRES, HERMON	
2040 MAIN STREET			ART UNIT	
FOURTEENTH FLOOR			PAPER NUMBER	
IRVINE, CA 92614			2449	

DATE MAILED: 11/18/2021

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

The Office has discontinued providing a Patent Term Adjustment (PTA) calculation with the Notice of Allowance.

Section 1(h)(2) of the AIA Technical Corrections Act amended 35 U.S.C. 154(b)(3)(B)(i) to eliminate the requirement that the Office provide a patent term adjustment determination with the notice of allowance. See Revisions to Patent Term Adjustment, 78 Fed. Reg. 19416, 19417 (Apr. 1, 2013). Therefore, the Office is no longer providing an initial patent term adjustment determination with the notice of allowance. The Office will continue to provide a patent term adjustment determination with the Issue Notification Letter that is mailed to applicant approximately three weeks prior to the issue date of the patent, and will include the patent term adjustment on the patent. Any request for reconsideration of the patent term adjustment determination (or reinstatement of patent term adjustment) should follow the process outlined in 37 CFR 1.705.

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

OMB Clearance and PRA Burden Statement for PTOL-85 Part B

The Paperwork Reduction Act (PRA) of 1995 requires Federal agencies to obtain Office of Management and Budget approval before requesting most types of information from the public. When OMB approves an agency request to collect information from the public, OMB (i) provides a valid OMB Control Number and expiration date for the agency to display on the instrument that will be used to collect the information and (ii) requires the agency to inform the public about the OMB Control Number's legal significance in accordance with 5 CFR 1320.5(b).

The information collected by PTOL-85 Part B is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450. Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Privacy Act Statement

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

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

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

Notice of Allowability	Application No. 17/374,864	Applicant(s) Tannenbaum et al.	
	Examiner HERMON ASRES	Art Unit 2449	AIA (FITF) Status No

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

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

1. ☒ This communication is responsive to 08/04/2021.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

3. ☒ The allowed claim(s) is/are 2-31. As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

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

Certified copies:

a) ☐ All b) ☐ Some* c) ☐ None of the:

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

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

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

* Certified copies not received: _____.

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

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

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

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

Attachment(s)

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

/HERMON ASRES/ Primary Examiner, Art Unit 2449	
---	--

DETAILED ACTION

REASONS FOR ALLOWANCE

Claims 2-31 are allowed over the prior art of record.

The following is an examiner's statement of reasons for allowance:

In interpreting the claims, in light of the specification the Examiner finds the claimed invention to be patentably distinct from the prior art of record.

Block et al. (2005/0021751) teaches an apparatus, program product and method utilize cluster data port services within a cluster infrastructure to provide reliable and efficient communications between nodes in a clustered computer system. The cluster data port services present an abstracted transport service that encapsulates and manages the establishment of multiple connection paths between a source node, a target node and one or more backup nodes in such a manner that a cluster data port is effectively utilized as single data port from the perspective of a user program. **(See Abstract).**

Fought et al. (6,074,427) teaches a system and method for simulating multiple clusters of independent computer nodes in a single machine. A cluster contains one or more computer nodes interconnected by a communications link. A user can simulate a cluster of n nodes by generating n user-level procedures where each user-level procedure represents the

kernel of a respective node. An additional mechanism is provided which allows a user to exercise the operation of any intended function in any of the nodes in any of the clusters. **(See Abstract).**

Singh (8,601,101) teaches a system and method for each node in a cluster including a peer-to-peer communication channel compiling and maintaining its own cluster membership are disclosed. In one embodiment, each cluster node may be coupled to every other node and commonly-accessible storage through a network. When a node starts up and attempts to join the cluster, it may access cluster information data stored in the commonly-accessible storage. This information may include a list of all nodes that are potential cluster members. Based on the cluster membership information, the joining node may request a peer-to-peer connection with each potential cluster member. When such a request results in establishment of a connection, the joining node may add the name of the target node to its cluster membership list. Likewise, when a member node grants a connection to a node, it may add the joining node to its cluster membership list. **(See Abstract).**

However, the prior art of records fail to teach or suggest individually or in combination as set forth in independent claims 2 filed on 08/04/2021. Claims 3-31 are also allowed for the same reasons.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably

accompany the issue fee. Such submission should clearly labeled "Comments on statement of Reason for Allowance".

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to HERMON ASRES whose telephone number is (571)272-4257. The examiner can normally be reached Monday to Friday 9AM to 5PM.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Vivek Srivastava can be reached on (571)272-7304. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of published or unpublished applications may be obtained from Patent Center. Unpublished application information in Patent Center is available to registered users. To file and manage patent submissions in Patent Center, visit: <https://patentcenter.uspto.gov>. Visit <https://www.uspto.gov/patents/apply/patent-center> for more information about Patent Center and <https://www.uspto.gov/patents/docx> for information about filing in DOCX format. For additional questions, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Application/Control Number: 17/374,864
Art Unit: 2449

Page 5

/HERMON ASRES/
Primary Examiner, Art Unit 2449

<i>Examiner-Initiated Interview Summary</i>	Application No. 17/374,864	Applicant(s) Tannenbaum et al.		
	Examiner HERMON ASRES	Art Unit 2449	AIA (First Inventor to File) Status No	Page 1 of 1

All Participants (applicant, applicants representative, PTO personnel)	Title	Type
HERMON ASRES	Primary Examiner	Telephonic
Jordan Cox (REG NO 70,875)	Attorney	

Date of Interview: 06 November 2021

Issues Discussed:

Other

Examiner contacted applicant to file a Terminal Disclaimer in order to put the application in condition for allowance. Terminal disclaimer was filed and approved on November 02, 2021.

/HERMON ASRES/ Primary Examiner, Art Unit 2449	
<p>Applicant is reminded that a complete written statement as to the substance of the interview must be made of record in the application file. It is the applicants responsibility to provide the written statement, unless the interview was initiated by the Examiner and the Examiner has indicated that a written summary will be provided. See MPEP 713.04</p> <p>Please further see: MPEP 713.04 Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews, paragraph (b) 37 CFR § 1.2 Business to be transacted in writing</p>	

Applicant recordation instructions: It is not necessary for applicant to provide a separate record of the substance of interview.

Examiner recordation instructions: Examiners must summarize the substance of any interview of record. A complete and proper recordation of the substance of an interview should include the items listed in MPEP 713.04 for complete and proper recordation including the identification of the general thrust of each argument or issue discussed, a general indication of any other pertinent matters discussed regarding patentability and the general results or outcome of the interview, to include an indication as to whether or not agreement was reached on the issues raised.

<i>Notice of References Cited</i>	Application/Control No. 17/374,864		Applicant(s)/Patent Under Reexamination Tannenbaum et al.	
	Examiner HERMON ASRES		Art Unit 2449	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-20050021751-A1	01-2005	Block, Timothy Roy	G06F9/54	709/225
*	B	US-8601101-B1	12-2013	Singh; Servesch Pratap	H04L69/40	709/223
*	C	US-6074427-A	06-2000	Fought; Steven K.	G06F9/45558	703/21
*	D	US-6968359-B1	11-2005	Miller; Robert	G06F9/5061	709/205
*	E	US-20070094532-A1	04-2007	SENGUPTA; Sabyasachi	G06F11/3632	714/5.1
*	F	US-20030195938-A1	10-2003	Howard, Kevin David	G06F9/5044	709/208
*	G	US-20070073705-A1	03-2007	Gray; Theodore W.	G06F9/451	1/1
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC 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.

U.S. Patent and Trademark Office
PTO-892 (Rev. 01-2001)

Notice of References Cited

Part of Paper No. 20211106

<i>Search Notes</i> 	Application/Control No. 17/374,864	Applicant(s)/Patent Under Reexamination Tannenbaum et al.
	Examiner HERMON ASRES	Art Unit 2449

CPC - Searched*		
Symbol	Date	Examiner

CPC Combination Sets - Searched*		
Symbol	Date	Examiner
H04L41/0213, H04L29/08072, H04L29/06, H04L41/22, H04L41/12	11/06/2021	HA

US Classification - Searched*			
Class	Subclass	Date	Examiner
709	223	11/06/2021	HA

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

--	--

<i>Search Notes</i> 	Application/Control No. 17/374,864	Applicant(s)/Patent Under Reexamination Tannenbaum et al.
	Examiner HERMON ASRES	Art Unit 2449

Search Notes		
Search Notes	Date	Examiner
Searched East (US Patents, USPGPubs, EPO, JPO, Derwent, IBM)	11/06/2021	HA
See East search attached	11/06/2021	HA
Inventor search performed	11/06/2021	HA
Key word search performed in NPL (google scholar and ip.com)	11/06/2021	HA
Interference search performed	11/06/2021	HA
IPR2020-01608 Reviewed for Petition for Inter Parties Review of U.S. Patent No. 8,082,289	11/06/2021	HA
IPR2021-00019 Reviewed for Petition for Inter Parties Review of U.S. Patent No. 10,333,768	11/06/2021	HA
IPR2021-00020 Reviewed for Petition for Inter Parties Review of U.S. Patent No. 10,333,768	11/06/2021	HA
IPR2021-00075 Reviewed for Petition for Inter Parties Review of U.S. Patent No. 8,140,612	11/06/2021	HA
IPR2021-00108 Reviewed for Petition for Inter Parties Review of U.S. Patent No. 8,676,877	11/06/2021	HA

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner
709	223	11/06/2021	HA

--	--

Issue Classification 	Application/Control No. 17/374,864	Applicant(s)/Patent Under Reexamination Tannenbaum et al.
	Examiner HERMON ASRES	Art Unit 2449

CPC						
Symbol					Type	Version
H04L	/	41	/	04	F	2013-01-01
G06F	/	9	/	5072	I	2013-01-01
G06F	/	9	/	54	I	2013-01-01
G06F	/	15	/	76	I	2013-01-01

CPC Combination Sets				
Symbol	Type	Set	Ranking	Version
/		/		

NONE (Assistant Examiner) (Date)		Total Claims Allowed: 30	
/HERMON ASRES/ Primary Examiner, Art Unit 2449 (Primary Examiner) (Date)		06 November 2021 (Date)	O.G. Print Claim(s) 1
			O.G. Print Figure 1

Issue Classification 	Application/Control No. 17/374,864	Applicant(s)/Patent Under Reexamination Tannenbaum et al.
	Examiner HERMON ASRES	Art Unit 2449

INTERNATIONAL CLASSIFICATION			
CLAIMED			
H04L12/24	/	12	/ 24
G06F9/50	/	9	/ 50
G06F9/54	/	9	/ 54
G06F15/76	/	15	/ 76
NON-CLAIMED			
	/		/

US ORIGINAL CLASSIFICATION	
CLASS	SUBCLASS
709	223

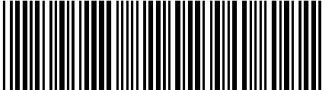
CROSS REFERENCES(S)						
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)					

NONE (Assistant Examiner) (Date)		Total Claims Allowed: 30	
/HERMON ASRES/ Primary Examiner, Art Unit 2449 (Primary Examiner)		06 November 2021 (Date)	O.G. Print Claim(s) 1
			O.G. Print Figure 1

Issue Classification 	Application/Control No. 17/374,864	Applicant(s)/Patent Under Reexamination Tannenbaum et al.
	Examiner HERMON ASRES	Art Unit 2449

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47															
CLAIMS															
Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	2	10	11	19	20	28	29								
2	3	11	12	20	21	29	30								
3	4	12	13	21	22	30	31								
4	5	13	14	22	23										
5	6	14	15	23	24										
6	7	15	16	24	25										
7	8	16	17	25	26										
8	9	17	18	26	27										
9	10	18	19	27	28										

NONE (Assistant Examiner) _____ (Date) _____		Total Claims Allowed: 30	
/HERMON ASRES/ Primary Examiner, Art Unit 2449 (Primary Examiner) _____ (Date) 06 November 2021		O.G. Print Claim(s) 1	O.G. Print Figure 1

<i>Index of Claims</i> 	Application/Control No. 17/374,864	Applicant(s)/Patent Under Reexamination Tannenbaum et al.
	Examiner HERMON ASRES	Art Unit 2449

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

CLAIMS										
☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47										
CLAIM		DATE								
Final	Original	11/06/2021								
	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	=								

Bibliographic Data

Application No: 17/374,864

Foreign Priority claimed: ☐ Yes ☒ No

35 USC 119 (a-d) conditions met: ☐ Yes ☒ No ☐ Met After Allowance

Verified and Acknowledged: /HERMON ASRES/

Examiner's Signature

Initials

Title: CLUSTER COMPUTING

FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
07/13/2021	709	2449	ZTANN.002P1C6
RULE			

APPLICANTS

Advanced Cluster Systems, Inc., Newport Beach, CA,

INVENTORS

Zvi Tannenbaum, Palo Alto, CA, UNITED STATES

Dean E. Dauger, Huntington Beach, CA, UNITED STATES

CONTINUING DATA

This application is a CON of 16449084 06/21/2019 PAT 11128519

16449084 is a CON of 14181112 02/14/2014 PAT 10333768

14181112 is a CON of 13423063 03/16/2012 PAT 8676877

13423063 is a CON of 12040519 02/29/2008 PAT 8140612

12040519 is a CIP of 11744461 05/04/2007 PAT 8082289

11744461 has PRO of 60850908 10/11/2006

11744461 has PRO of 60813738 06/13/2006

FOREIGN APPLICATIONS

IF REQUIRED, FOREIGN LICENSE GRANTED**

07/27/2021

** SMALL ENTITY **

STATE OR COUNTRY

UNITED STATES

ADDRESS

KNOBBE MARTENS OLSON & BEAR LLP

2040 MAIN STREET

FOURTEENTH FLOOR

IRVINE, CA 92614

UNITED STATES
FILING FEE RECEIVED
\$3,000

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
SHEET 1 OF 12		Attorney Docket No. ZTANN.002P1C6

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
/H.A./	1	7,533,168	05-12-2009	Pabla et al.	
/H.A./	2	2020/0153677 (Attorney Docket No. ZTANN.002P1C5) U.S. Application No. 16/449,084 and its entire prosecution history**	05-14-2020	Tannenbaum et al.	
/H.A./	3	5,423,046	06-06-1995	Nunnelley et al.	
/H.A./	4	5,881,315	03-09-1999	Cohen	
/H.A./	5	5,930,768	07-27-1999	Hooban	
/H.A./	6	6,006,259	11-20-1998	Adelman et al.	
/H.A./	7	6,067,609	05-23-2000	Meeker et al.	
/H.A./	8	6,074,427	06-13-2000	Fought et al.	
/H.A./	9	6,108,699	08-22-2000	Moiin	
/H.A./	10	6,195,680	02-27-2001	Goldszmidt et al.	
/H.A./	11	6,202,080	03-13-2001	Lu et al.	
/H.A./	12	6,546,403	04-08-2003	Carlson, Jr. et al.	
/H.A./	13	6,578,068	06-10-2003	Bowman-Amuah	
/H.A./	14	6,604,134	08-05-2003	Haury	
/H.A./	15	6,691,216	02-10-2004	Kelly et al.	
/H.A./	16	6,751,698	06-15-2004	Deneroff et al.	
/H.A./	17	6,782,537	08-24-2004	Blackmore et al.	
/H.A./	18	6,968,335	11-22-2005	Bayliss et al.	
/H.A./	19	6,968,359	11-22-2005	Miller	
/H.A./	20	7,031,944	04-18-2006	Tanioka	
/H.A./	21	7,093,004	08-15-2006	Bernardin et al.	
/H.A./	22	7,136,924	11-14-2006	Dauger	
/H.A./	23	7,139,882	11-21-2006	Suzuoki et al.	
/H.A./	24	7,174,381	02-06-2007	Gulko et al.	

Examiner Signature /HERMON ASRES/	Date Considered 11/06/2021
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
	Examiner	Unassigned
SHEET 2 OF 12	Attorney Docket No.	ZTANN.002P1C6

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
/H.A./	25	7,177,874	02-13-2007	Jardin	
/H.A./	26	7,249,357	07-24-2007	Landman et al.	
/H.A./	27	7,315,877	01-01-2008	Bhanot et al.	
/H.A./	28	7,334,232	02-19-2008	Jacobs et al.	
/H.A./	29	7,437,460	10-14-2008	Chidambaran et al.	
/H.A./	30	7,472,193	12-30-2008	Dauger	
/H.A./	31	7,502,915	03-10-2009	Jacob et al.	
/H.A./	32	7,519,652	04-14-2009	Page et al.	
/H.A./	33	7,533,141	05-12-2009	Nadgir et al.	
/H.A./	34	7,533,389	05-12-2009	Verbeke et al.	
/H.A./	35	7,554,949	06-30-2009	Chen	
/H.A./	36	7,568,131	07-28-2009	Vertes	
/H.A./	37	7,634,530	12-15-2009	Dean et al.	
/H.A./	38	7,644,130	01-05-2010	Magro et al.	
/H.A./	39	7,698,390	04-13-2010	Harkness et al.	
/H.A./	40	7,716,323	05-11-2010	Gole et al.	
/H.A./	41	7,757,236	07-13-2010	Singh	
/H.A./	42	7,788,314	08-31-2010	Holt	
/H.A./	43	7,835,022	11-16-2010	Matsumoto	
/H.A./	44	7,937,455	05-03-2011	Saha et al.	
/H.A./	45	8,082,289 (Attorney Docket No. ZTANN.002A) U.S. Application No. 11/744,461 and its entire prosecution history**	12-20-2011	Tannenbaum et al.	
/H.A./	46	8,117,288	02-14-2012	Bhanot et al.	

Examiner Signature /HERMON ASRES/	Date Considered 11/06/2021
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	17/374864	
	Filing Date	July 13, 2021	
	First Named Inventor	Tannenbaum, Zvi	
	Art Unit	2447	
(Multiple sheets used when necessary)		Examiner	Unassigned
SHEET 3 OF 12		Attorney Docket No.	ZTANN.002P1C6

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
/H.A./	47	8,140,612 (Attorney Docket No. ZTANN.002CP1) U.S. Application No. 12/040,519 and its entire prosecution history**	03-20-2012	Tannenbaum et al.	
/H.A./	48	8,402,080 (Attorney Docket No. ZTANN.001C1) U.S. Application No. 11/749,554 and its entire prosecution history**	03-19-2013	Tannenbaum et al.	
/H.A./	49	8,402,083 (Attorney Docket No. ZTANN.002CP1C1) U.S. Application No. 12/465,519 and its entire prosecution history**	03-19-2013	Tannenbaum et al.	
/H.A./	50	8,601,101	12-03-2013	Singh	
/H.A./	51	8,676,877 (Attorney Docket No. ZTANN.002P1C2) U.S. Application No. 13/423,063 and its entire prosecution history**	03-18-2014	Tannenbaum et al.	
/H.A./	52	8,849,889 (Attorney Docket No. ZTANN.001C2) U.S. Application No. 13/846,567 and its entire prosecution history**	09-30-2014	Tannenbaum et al.	
/H.A./	53	9,405,564	08-02-2016	Muellers et al.	
/H.A./	54	10,333,768 (Attorney Docket No. ZTANN.002P1C3) U.S. Application No. 14/181,112 and its entire prosecution history**	06-25-2019	Tannenbaum et al.	

Examiner Signature /HERMON ASRES/	Date Considered 11/06/2021
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
(Multiple sheets used when necessary)	Examiner	Unassigned
SHEET 4 OF 12	Attorney Docket No.	ZTANN.002P1C6

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
/H.A./	55	2002/0049859	04-25-2002	Bruckert et al.	
/H.A./	56	2003/0005068	01-02-2003	Nickel et al.	
/H.A./	57	2003/0005266	01-02-2003	Akkary et al.	
/H.A./	58	2003/0051062	03-13-2003	Circenis	
/H.A./	59	2003/0135621	07-17-2003	Romagnoli	
/H.A./	60	2003/0195931	10-16-2003	Dauger	
/H.A./	61	2003/0195938	10-16-2003	Howard et al.	
/H.A./	62	2004/0015968	01-22-2004	Neiman et al.	
/H.A./	63	2004/0098359	05-20-2004	Bayliss et al.	
/H.A./	64	2004/0098447	05-20-2004	Verbeke et al.	
/H.A./	65	2004/0110209	06-10-2004	Yokota et al.	
/H.A./	66	2004/0157203	08-12-2004	Dunk	
/H.A./	67	2004/0195572	10-07-2004	Kato et al.	
/H.A./	68	2004/0252710	12-16-2004	Jeter, Jr. et al.	
/H.A./	69	2004/0254984	12-16-2004	Dinker	
/H.A./	70	2005/0015460	01-20-2005	Goyle et al.	
/H.A./	71	2005/0021751	01-27-2005	Block et al.	
/H.A./	72	2005/0038852	02-17-2005	Howard	
/H.A./	73	2005/0060237	03-17-2005	Barsness et al.	
/H.A./	74	2005/0076105	04-07-2005	Keohane et al.	
/H.A./	75	2005/0097300	05-05-2005	Gildea et al.	
/H.A./	76	2005/0108394	05-19-2005	Braun et al.	
/H.A./	77	2005/0154789	07-14-2005	Fellenstein et al.	
/H.A./	78	2005/0180095	08-18-2005	Ellis	
/H.A./	79	2005/0188088	08-25-2005	Fellenstein et al.	
/H.A./	80	2005/0235055	10-20-2005	Davidson	
/H.A./	81	2006/0026601	02-02-2006	Solt	

Examiner Signature /HERMON ASRES/	Date Considered 11/06/2021
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
SHEET 5 OF 12		Attorney Docket No. ZTANN.002P1C6

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
/H.A./	82	2006/0053216	03-09-2006	Deokar et al.	
/H.A./	83	2006/0059473	03-16-2006	Moler	
/H.A./	84	2006/0106931	05-18-2006	Richou	
/H.A./	85	2007/0044099	02-22-2007	Rajput	
/H.A./	86	2007/0073705	03-29-2007	Gray	
/H.A./	87	2007/0094532	04-26-2007	Sengupta	
/H.A./	88	2007/0124363	05-31-2007	Lurie et al.	
/H.A./	89	2008/0250347	10-09-2008	Gray et al.	
/H.A./	90	2008/0281997	11-13-2008	Archer et al.	

FOREIGN PATENT DOCUMENTS						
Examiner Initials	Cite No.	Foreign Patent Document Country Code-Number-Kind Code Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T ¹
/H.A./	91	EP 1 368 948	12-10-2003	Prismmedia Networks, Inc.		PCT Eq.
/H.A./	92	JP 08-087473	02-04-1996	Toshiba Corp		X
/H.A./	93	JP 11-126196	05-11-1999	Hitachi Ltd.		X-MT
/H.A./	94	JP 11-328134	11-30-1999	Hitachi Ltd.		X-Abs
/H.A./	95	JP 2002-117010	04-19-2002	International Business Machine Corp.		X
/H.A./	96	JP 2004-061359	02-26-2004	Ricoh Co., LTd.		X-Abs
/H.A./	97	JP 2004-247405	09-02-2004	Semiconductor Energy Lab		X-MT
/H.A./	98	JP 2005-063033	10-03-2005	Toshiba Corp.		X-Abs
/H.A./	99	JP 2005-071333	03-17-2005	Network Appliance Inc.		X-MT
/H.A./	100	WO 2004/086270	10-07-2004	ECI Telecom Ltd.		

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹

Examiner Signature /HERMON ASRES/	Date Considered 11/06/2021
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
(Multiple sheets used when necessary)	Examiner	Unassigned
SHEET 6 OF 12	Attorney Docket No.	ZTANN.002P1C6

NON PATENT LITERATURE DOCUMENTS

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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
/H.A./	101	Affidavit of Elizabeth Rosenberg, dated September 1, 2020 in 7 pages.	
/H.A./	102	AGARWAL et al., "A High Performance Parallel Algorithm for 1-D FFT", Proceedings, Supercomputing '94, Washington, D.C., November 14-18, 1994, pp. 34-40.	
/H.A./	103	ANNARATONE et al., "The Warp Computer: Architecture, Implementation, and Performance", Dept. of Computer Science, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 1987, pp. 31.	
/H.A./	104	BÓSA et al., "Task Logging, Rescheduling and Peer Checking in Distributed Maple", RISC-Linz, Johannes Kepler University, Linz, Austria. Technical Report No. 02-10, March 18, 2002, pp. 24.	
/H.A./	105	CARNS et al., "An Evaluation of Message Passing Implementations on Beowulf Workstations", Aerospace Conference, March 6-13, 1999, IEEE 1999, Vol. 5, pp. 41-54.	
/H.A./	106	CHU et al., "FFT Algorithms and their Adaptation to Parallel Processing", Linear Algebra and its Applications, Vol. 284, 1998, pp. 95-124.	
/H.A./	107	CRAMER, et al., "The Development and Integration of a Distributed 3D FFT for a Cluster of Workstations", Proceedings of the 4th Annual Liux Showcase & Conference, Atlanta, GA, October 10-14, 2000, pp. 1-9.	
/H.A./	108	DALLY et al., "Merrimac: Supercomputing with Streams", SC'03, November 15-21, 2003, Phoenix, AZ, pp. 8.	
/H.A./	109	DAUGER et al., "'Plug-and-Play' Cluster Computing using Mac OS X", Proceedings. 2003 IEEE International Conference on Cluster Computing, December 1-4, 2003, published January 2004, pp. 6.	
/H.A./	110	DAUGER et al., "Plug-and-Play Cluster Computing: High-Performance Computing for the Mainstream", IEEE Computing in Science and Engineering, March/April 2005, pp. 27-33.	
/H.A./	111	Declaration of Henry Tufo, Ph.D., under 37 C.F.R. § 1.68 in Support of Petition for Inter Partes Review of U.S. Patent No. 8,082,289, dated September 18, 2020 in 160 pages.	
/H.A./	112	Declaration of Sylvia Hall-Ellis Ph.D., for Inter Partes Review of U.S. Patent No. 8,082,289, dated September 18, 2020 in 138 pages.	
/H.A./	113	Declaration of Wolfgang Schreiner, Ph.D., for Inter Partes Review of U.S. Patent No. 8,082,289, dated September 7, 2020 in 84 pages.	
/H.A./	114	"Distributed Maple", https://web.archive.org/web/20040216100904/http://www.risc.uni-linz.ac.at/software/distmaple/ as archived February 16, 2004 in 1 page.	
/H.A./	115	"Distributed Maple Papers", https://web.archive.org/web/20040304035455/http://www.risc.uni-linz.ac.at/software/distmaple/index_1.html as archived March 4, 2004 in 3 pages.	
/H.A./	116	EREZ, Mattan, "MERRIMAC – High-Performance and Highly-Efficient Scientific Computing with Streams", A Dissertation Submitted to the Department of Electrical Engineering and the Committee on Graduate Studies of Stanford University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy, May 2007, pp. 229	
/H.A./	117	FAN et al., "GPU Cluster for High Performance Computing", SC '04: Proceedings of the 2004 ACM/IEEE Conference on Supercomputing : 6-12 November 2004, Pittsburgh, PA, USA, pp. 12.	
/H.A./	118	FRIGO et al., "FFTW User's Manual", Version 2.1.5, March 16, 2003, pp. 74.	

Examiner Signature /HERMON ASRES/	Date Considered 11/06/2021
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
(Multiple sheets used when necessary)	Examiner	Unassigned
SHEET 7 OF 12	Attorney Docket No.	ZTANN.002P1C6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
/H.A./	119	"gridMathematica 1.1: Grid Computing Gets a Speed Boost from Mathematica 5", The Mathematica Journal, Vol. 9, No. 2, August 25, 2003, in 1 page. < https://web.archive.org/web/20040618053429/https://www.mathematica-journal.com/issue/v9i2/newproducts/gridmathematica.html >	
/H.A./	120	HAMSCHER et al., "Evaluation of Job-Scheduling Strategies for Grid Computing", GRID 2000, 2000, pp. 191-202.	
/H.A./	121	HAYNOS, Matt, "Perspectives on Grid: Grid Computing -- Next-Generation Distributed Computing", IBM Developer Works, January 27, 2004 in 7 pages, http://www-106.ibm.com/developerworks/library/gr-heritage	
/H.A./	122	HEAL et al., "Maple V: Learning Guide", Springer, Waterloo Maple Inc., 1996, 1998, pp. 300.	
/H.A./	123	HILLIS, W. Daniel, "Richard Feynman and The Connection Machine", for Physics Today, originally published 1989, http://longnow.org/essays/richard-feynman-connection-machine/ , printed October 1, 2019, pp. 8.	
/H.A./	124	"INSTALL", File, Distributed Maple Application Software, June 16, 2000, pp. 5.	
/H.A./	125	"INSTALL", File, Distributed Maple, March 28, 2003, pp. 4.	
/H.A./	126	Source Code, "Berlekamp Distinct Degree Factorization", Distributed Maple Application Software from the "distsoft" directory, v. 1.1, July 26, 2001, pp. 58.	
/H.A./	127	Source Code, "CASA NeighbGraph", Distributed Maple Application Software from the "distsoft" directory, v. 1.6, January 9, 2001, pp. 47.	
/H.A./	128	Source Code, "Maple Interface to Java Scheduler", Distributed Maple, dist.maple5, v 1.63, March 28, 2003, pp. 28.	
/H.A./	129	JAIN et al., "Data Clustering: A Review", ACM Computing Surveys, September 1999, Vol. 31, No. 3, pp. 264-323.	
/H.A./	130	KEPNER et al., "Parallel Matlab: The Next Generation", MIT Lincoln Laboratory, Lexington, MA, August 20, 2004, pp. 2.	
/H.A./	131	KIM et al., "Introduction to Parallel Programming and pMatlab v2.0", MIT Lincoln Laboratory, Lexington, MA, 2011, pp. 30.	
/H.A./	132	KOFLER, Michael, "Maple: An Introduction and Reference", Addison-Wesley, 1997, pp. 545.	
/H.A./	133	KONISHI, et al., "Performance Evaluation of Parallel Computing Tool for MATLAB on PC Grid Environment," IPSJ SIG Technical Reports, August 5, 2005, Vol. 2005, No. 84, pp. 9.	X-Abs
/H.A./	134	MAEDER, Roman, "Mathematica: Parallel Computing Toolkit - Unleash the Power of Parallel Computing", Wolfram Research, January 2005, pp. 95.	
/H.A./	135	MATSUMURA, et al. "Construction of Distributed Computing System for Large-Scale Matrices Intended or Reduction of Communication Blocks", August 5, 1998, Vol. 98, No. 18, pp. 9.	X-MT
/H.A./	136	"MPI: A Message-Passing Interface Standard", Message Passing Interface Forum, June 12, 1995, pp. 239.	

Examiner Signature /HERMON ASRES/	Date Considered 11/06/2021
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
(Multiple sheets used when necessary)	Examiner	Unassigned
SHEET 8 OF 12	Attorney Docket No.	ZTANN.002P1C6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
/H.A./	137	PAU et al., "Distributed Mathematica - User and Reference Manual", July 5, 2000, pp. 18.	
/H.A./	138	"Pingpong MPI Benchmark - SEM vs 'grid'", Dauger Research, Inc., January 2009, 1 page, http://daugerresearch.com/pooch/mathmatica/	
/H.A./	139	"README File", Distributed Maple Application Software, July 26, 2001, pp. 3.	
/H.A./	140	"README File", Distributed Maple 1.1.15, March 28, 2003, pp. 5.	
/H.A./	141	Reference Manual, TRANSPUTER, Inmos, IMS, January 1987, pp. 162.	
/H.A./	142	SCHREINER et al., "Distributed Maple: Parallel Computer Algebra in Networked Environments", Journal of Symbolic Computation, March 2003, Vol. 35, No. 3, pp. 305-347.	
/H.A./	143	SCHREINER, Wolfgang, "Distributed Maple - User and Reference Manual (V 1.1.12)", Research Institute for Symbolic Computation (RISC-Linz), Johannes Kepler University, A-4040 Linz, Austria, July 6, 2001, pp. 1-37.	
/H.A./	144	SHERWANI et al., "Libra: A Computational Economy-Based Job Scheduling System for Clusters", Software-Practice and Experience, 2004, Vol. 34, pp. 573-590.	
/H.A./	145	TATEBE et al., "Efficient Implementation of MPI Using Remote Memory Write," Transactions of Information Processing Society of Japan, May 1999, Vol. 40, No. 5, pp. 14.	X-Abs
/H.A./	146	TEPENEU et al., "MathGridLink - A Bridge Between Mathematica and the Grid", Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba, 2003, pp. 4.	
/H.A./	147	U.S. Provisional Application No. 60/799,474, filed May 10, 2006, titled "Graphical Interface for Monitoring Status of a Concurrent Computing Process", in 45 pages.	
/H.A./	148	U.S. Provisional Application No. 60/813,738, filed June 13, 2006, titled "Pooch/Mathmatica Interface", in 6 pages.	
/H.A./	149	U.S. Provisional Application No. 60/850,908, filed October 11, 2006, titled "Poochmpi Toolkit for Mathematica", in 25 pages.	
/H.A./	150	"Wolfram gridMathematica™", < http://www.wolfram.com/products/gridmathematica/ > as printed October 3, 2007 in 2 pages.	
/H.A./	151	WOLFRAM, Stephen, "The Mathematica Book", 5th Edition, Wolfram Research, Inc., 2003, pp. 1486.	
/H.A./	152	NVIDIA Power of Attorney, October 5, 2020, pp. 2.	
/H.A./	153	Ex. 2001 - Singh Declaration, February 9, 2021, pp. 42.	
/H.A./	154	Ex. 2002 - Singh CV, pp. 4.	

Examiner Signature /HERMON ASRES/	Date Considered 11/06/2021
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
	Examiner	Unassigned
SHEET 9 OF 12	Attorney Docket No.	ZTANN.002P1C6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
/H.A./	155	Ex. 2007 - Bhansali Declaration, February 9, 2021, pp. 5.	
/H.A./	156	Ex. 2008 - Bancroft Declaration, February 9, 2021, pp. 13.	
/H.A./	157	Ex. 2009 - SEM Poster, 2009, pp. 1.	
/H.A./	158	Ex. 2011 - SEM Manual, pp. 20.	
/H.A./	159	Ex. 2012 - SEM White Paper, 2007, pp. 20.	
/H.A./	160	Ex. 2013 - SET White Paper, pp. 7.	
/H.A./	161	Ex. 2014 - SET Manual, pp. 67.	
/H.A./	162	Ex. 2015 - SET Presentation, December 2011, pp. 54.	
/H.A./	163	Ex. 2016 - SET Datasheet, pp. 4.	
/H.A./	164	Ex. 2017 - Parallel Bars the Economist, June 4, 2011, pp. 9.	
/H.A./	165	Ex. 2018 - SET Presentation, HPC for the Missing Middle!, December 2011, pp. 4.	
/H.A./	166	Ex. 2019 - DOE Review of Plasma Visualization Project, 2015, pp. 5.	
/H.A./	167	Ex. 2020 - SET Data Sheet given to Jones, pp. 4.	
/H.A./	168	Ex. 2021 - DOE Review of Expanded Validation Project, 2015, pp. 5	
/H.A./	169	Ex. 2022 - DOE Review of SET Windows Implementation, 2015, pp. 5.	
/H.A./	170	Ex. 2023 - SEM an Independent Report, 2008, pp. 2.	
/H.A./	171	Ex. 2024 - Multicore Parallelism presentation, February 2011, pp. 2.	
/H.A./	172	Ex. 2025 - DOE SET Windows Project Narrative, pp. 2.	
/H.A./	173	Ex. 2026 - DOE Topics, pp. 11.	

Examiner Signature /HERMON ASRES/	Date Considered 11/06/2021
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
	Examiner	Unassigned
SHEET 10 OF 12	Attorney Docket No.	ZTANN.002P1C6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
/H.A./	174	Ex. 2027 - Dager Declaration, February 9, 2021, pp. 30.	
/H.A./	175	Ex. 2028 - Nexus Claim Chart - SEM - 768 Patent, pp. 13.	
/H.A./	176	Ex. 2029 - Nexus Claim Chart - SET - 768 Patent, pp. 17.	
/H.A./	177	Exhibit 1002 - File History of USP 10,333,768 B2, pp. 627.	
/H.A./	178	Exhibit 1005 - Declaration of Henry Tufo, October 2, 2020, pp. 132.	
/H.A./	179	Exhibit 1006 - Declaration of Wolfgang Schreiner, October 1, 2020, pp. 85.	
/H.A./	180	Exhibit 1007 - Declaration of Sylvia Hall-Ellis, October 5, 2020, pp. 109.	
/H.A./	181	Exhibit 1019 - SUN Debuts UltraSPARC IV (2003), pp. 2.	
/H.A./	182	Exhibit 1020 - AMD to Unveil Dual-Core PC Chips (2005), May 31, 2005, pp. 1.	
/H.A./	183	Ex. 2001 - Singh Declaration, February 9, 2021, pp. 42.	
/H.A./	184	Ex. 2030 - Dager Declaration, February 9, 2021, pp. 31.	
/H.A./	185	Ex. 2031 - Nexus Claim Chart - SEM - 768 Patent, pp. 13.	
/H.A./	186	Ex. 2032 - Nexus Claim Chart - SET - 768 Patent, pp. 40.	
/H.A./	187	Exhibit 1105 - Declaration of Henry Tufo, October 9, 2020, pp. 147.	
/H.A./	188	Exhibit 1106 - Declaration of Wolfgang Schreiner, October 1, 2020, pp. 85.	
/H.A./	189	Exhibit 1107 - Declaration of Sylvia Hall-Ellis, October 9, 2020, pp. 255.	
/H.A./	190	Exhibit 1134 - Maple 5 - PC Magazine (1992)	
/H.A./	191	Exhibit 1135 - Maple V Looks Better with Improved Graphics - PC Magazine	
/H.A./	192	Exhibit 1136 - Nayak, A Library based compiler to execute MATLAB Programs	

Examiner Signature /HERMON ASRES/	Date Considered 11/06/2021
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
(Multiple sheets used when necessary)	Examiner	Unassigned
SHEET 11 OF 12	Attorney Docket No.	ZTANN.002P1C6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
/H.A./	193	Ex. 2001 - Singh Declaration, February 9, 2021, pp. 41.	
/H.A./	194	Ex. 2033 - Dager Declaration, February 10, 2021, pp. 30.	
/H.A./	195	Ex. 2034 - Nexus Claim Chart - SEM - 612 Patent, pp. 11.	
/H.A./	196	Ex. 2035 - Nexus Claim Chart - SET - 612 Patent, pp. 14.	
/H.A./	197	Exhibit 1002 - File History of USP 8,140,612 B2	
/H.A./	198	Exhibit 1005 - Declaration of Henry Tufo, October 16, 2020, pp. 127.	
/H.A./	199	Exhibit 1006 - Declaration of Wolfgang Schreiner, October 1, 2020, pp. 86.	
/H.A./	200	Exhibit 1007 - Declaration of Sylvia Hall-Ellis, October 16, 2020, pp. 178.	
/H.A./	201	Exhibit 1019 - Deitel - Operating Systems, 2004, pp. 1270 (2parts).	
/H.A./	202	Exhibit 1020 - Chen et al., TH-MPI_ OS Kernel Integrated Fault Tolerant MPI (2001), pp. 75-82.	
/H.A./	203	Exhibit 1026 - Jin et al., LiMIC_ Support for High-Performance MPI Intra-Node Communication on Linux, 2005, pp. 184-191.	
/H.A./	204	Ex. 2001 - Singh Declaration, February 17, 2021, pp. 40.	
/H.A./	205	Ex. 2036 - Dager Declaration, February 15, 2021, pp. 30.	
/H.A./	206	Ex. 2037 - Nexus Claim Chart - SEM - 877 Patent, pp. 33.	
/H.A./	207	Ex. 2038 - Nexus Claim Chart - SET - 877 Patent, pp. 50.	
/H.A./	208	Exhibit 1002 - File History of USP 8,676,877 B2	
/H.A./	209	Exhibit 1005 - Declaration of Henry Tufo, October 23, 2020, pp. 130.	
/H.A./	210	Exhibit 1006 - Declaration of Wolfgang Schreiner, October 1, 2020, pp. 86.	
/H.A./	211	Exhibit 1007 - Declaration of Sylvia Hall-Ellis, October 23, 2020, pp. 186.	

Examiner Signature /HERMON ASRES/	Date Considered 11/06/2021
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
(Multiple sheets used when necessary)	Examiner	Unassigned
SHEET 12 OF 12	Attorney Docket No.	ZTANN.002P1C6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
/H.A./	212	Extended European Search Report received in European Application No. 17207443.7 (ZTANN.002EPD1), dated August 8, 2018, in 9 pages.	
/H.A./	213	International Search Report and Written Opinion received in PCT Application No. PCT/US2006/037275 (ZTANN.001VPC), as mailed September 24, 2007 in 5 pages.	
/H.A./	214	International Search Report and Written Opinion received in PCT Application No. PCT/US2007/070585 (ZTANN.002VPC), as mailed September 11, 2008 in 5 pages.	

53865465

** Signature indicates consideration of publication and file history. The Examiner has access to these materials through the PTO computer systems. If additional copies are desired, please notify Applicants through their attorneys.

Examiner Signature /HERMON ASRES/	Date Considered 11/06/2021
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.



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
17/374,864	07/13/2021	Zvi Tannenbaum	ZTANN.002P1C6

CONFIRMATION NO. 1083

20995
KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

PUBLICATION NOTICE



OC000000129599167

Title:CLUSTER COMPUTING

Publication No.US-2021-0344552-A1

Publication Date:11/04/2021

NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO's publically available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently <http://www.uspto.gov/patft/>.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Public Records Division. The Public Records Division can be reached by telephone at (571) 272-3150 or (800) 972-6382, by facsimile at (571) 273-3250, by mail addressed to the United States Patent and Trademark Office, Public Records Division, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently <https://portal.uspto.gov/pair/PublicPair>. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

Doc Code: DIST.E.FILE Document Description: Electronic Terminal Disclaimer - Filed		PTO/SB/26 U.S. Patent and Trademark Office Department of Commerce
Electronic Petition Request	TERMINAL DISCLAIMER TO OBVIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT	
Application Number	17374864	
Filing Date	13-Jul-2021	
First Named Inventor	Zvi Tannenbaum	
Attorney Docket Number	ZTANN.002P1C6	
Title of Invention	CLUSTER COMPUTING	
☒ Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action ☒ This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.		
Owner	Percent Interest	
Advanced Cluster Systems, Inc.	100%	
<p>The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)</p> <p>11128519</p> <p>as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.</p> <p>In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:</p> <ul style="list-style-type: none"> - expires for failure to pay a maintenance fee; - is held unenforceable; - is found invalid by a court of competent jurisdiction; - is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321; - has all claims canceled by a reexamination certificate; - is reissued; or - is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer. <p>☒ Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.</p>		

☐ I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.	
Applicant claims the following fee status: ☒ Small Entity ☐ Micro Entity ☐ Regular Undiscounted	
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.	
THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES I certify, in accordance with 37 CFR 1.4(d)(4) that I am: ☒ An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application Registration Number <u>70875</u> ☐ A sole inventor ☐ A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application ☐ A joint inventor; all of whom are signing this request	
Signature	/Jordan M. Cox/
Name	Jordan M. Cox

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
 Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

Electronic Patent Application Fee Transmittal				
Application Number:		17374864		
Filing Date:		13-Jul-2021		
Title of Invention:		CLUSTER COMPUTING		
First Named Inventor/Applicant Name:		Zvi Tannenbaum		
Filer:		Jordan Michael Cox/Jennifer Neat		
Attorney Docket Number:		ZTANN.002P1C6		
Filed as Small Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
STATUTORY OR TERMINAL DISCLAIMER	2814	1	170	170
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				170

Doc Code: DISQ.E.FILE

Document Description: Electronic Terminal Disclaimer – Approved

Application No.: 17374864

Filing Date: 13-Jul-2021

Applicant/Patent under Reexamination: Tannenbaum

Electronic Terminal Disclaimer filed on November 2, 2021

☒ APPROVED

This patent is subject to a terminal disclaimer

☐ DISAPPROVED

Approved/Disapproved by: Electronic Terminal Disclaimer automatically approved by EFS-Web

U.S. Patent and Trademark Office

Electronic Acknowledgement Receipt	
EFS ID:	44187286
Application Number:	17374864
International Application Number:	
Confirmation Number:	1083
Title of Invention:	CLUSTER COMPUTING
First Named Inventor/Applicant Name:	Zvi Tannenbaum
Customer Number:	20995
Filer:	Jordan Michael Cox/Jennifer Neat
Filer Authorized By:	Jordan Michael Cox
Attorney Docket Number:	ZTANN.002P1C6
Receipt Date:	02-NOV-2021
Filing Date:	13-JUL-2021
Time Stamp:	19:17:18
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$170
RAM confirmation Number	E2021A2J17133767
Deposit Account	111410
Authorized User	Jennifer Neat
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: 37 CFR 1.16 (National application filing, search, and examination fees) 37 CFR 1.17 (Patent application and reexamination processing fees)	

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Terminal Disclaimer-Filed (Electronic)	eTerminal-Disclaimer.pdf	41160	no	2
			949874cdd514edc9e5b63d89ae400595b360c64a		
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	37866	no	2
			1d66544f11fb2a51ba7ec4ec89af03fa0610f174		
Warnings:					
Information:					
Total Files Size (in bytes):			79026		
<p>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.</p> <p><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.</p> <p><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.</p> <p><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.</p>					



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.
17/374,864	07/13/2021	Zvi Tannenbaum	ZTANN.002P1C6	1083
20995 7590 10/13/2021 KNOBBE MARTENS OLSON & BEAR LLP 2040 MAIN STREET FOURTEENTH FLOOR IRVINE, CA 92614			EXAMINER CENTRAL, DOCKET	
			ART UNIT OPAP	PAPER NUMBER
			NOTIFICATION DATE 10/13/2021	DELIVERY MODE ELECTRONIC

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 the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

efiling@knobbe.com
jayna.carttee@knobbe.com

<i>Decision Granting Request for Prioritized Examination (Track I)</i>	Application No. 17/374,864	Applicant(s) Tannenbaum et al.	
	Examiner KIMBERLY A INABINET	Art Unit OPET	AIA (FITF) Status No
<p>1. THE REQUEST FILED <u>13 July 2021</u> IS GRANTED .</p> <p>The above-identified application has met the requirements for prioritized examination</p> <p>A. ☒ for an original nonprovisional application (Track I).</p> <p>B. ☐ for an application undergoing continued examination (RCE).</p> <p>2. The above-identified application will undergo prioritized examination. The application will be accorded special status throughout its entire course of prosecution until one of the following occurs:</p> <p>A. filing a <u>petition for extension of time</u> to extend the time period for filing a reply;</p> <p>B. filing an <u>amendment to amend the application to contain more than four independent claims, more than thirty total claims</u>, or a multiple dependent claim;</p> <p>C. filing a <u>request for continued examination</u> ;</p> <p>D. filing a notice of appeal;</p> <p>E. filing a request for suspension of action;</p> <p>F. mailing of a notice of allowance;</p> <p>G. mailing of a final Office action;</p> <p>H. completion of examination as defined in 37 CFR 41.102; or</p> <p>I. abandonment of the application.</p> <p>Telephone inquiries with regard to this decision should be directed to KIMBERLY INABINET at (571) 272-4618. In his/her absence, calls may be directed to Petition Help Desk at (571) 272-3282.</p>			
/KIMBERLY A INABINET/ Paralegal Specialist, OPET			

RESCISSION OF ANY PRIOR DISCLAIMERS AND REQUEST TO REVISIT ART

First Inventor: Zvi Tannenbaum	Conf. No.: 1083
App. No.: 17/374864	Filed: July 13, 2021
Examiner: Unassigned	Art Unit: 2449
Title: CLUSTER COMPUTING	

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

Dear Commissioner:

The claims of the present application are different and possibly broader in scope than the claims pursued in the parent application(s). To the extent any prior amendments or characterizations of the scope of any claim or referenced art could be construed as a disclaimer of any subject matter supported by the present disclosure, Applicant hereby rescinds and retracts such disclaimer. Accordingly, the references previously considered in the parent application(s) may need to be re-visited.

Please charge any additional fees, including any fees for additional extension of time, or credit overpayment to Deposit Account No. 11-1410.

Respectfully submitted,
Knobbe, Martens, Olson & Bear, LLP

Dated: August 11, 2021

/Jordan M. Cox/
Jordan M. Cox
Registration No. 70,875
Registered Practitioner
(949) 760-0404

53956410

Electronic Acknowledgement Receipt	
EFS ID:	43480527
Application Number:	17374864
International Application Number:	
Confirmation Number:	1083
Title of Invention:	CLUSTER COMPUTING
First Named Inventor/Applicant Name:	Zvi Tannenbaum
Customer Number:	20995
Filer:	Jordan Michael Cox/Chelsea Veinot
Filer Authorized By:	Jordan Michael Cox
Attorney Docket Number:	ZTANN.002P1C6
Receipt Date:	11-AUG-2021
Filing Date:	13-JUL-2021
Time Stamp:	13:32:21
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	Miscellaneous Incoming Letter	ZTANN-002P1C6_Request_revisit_art.pdf	17600 3b2e7484dce361e28dbc6b731788e7e7b4f1c97f	no	1

Warnings:

Information:	
Total Files Size (in bytes):	17600
<p>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.</p> <p><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.</p> <p><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.</p> <p><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.</p>	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
	Examiner	Unassigned
SHEET 1 OF 12	Attorney Docket No.	ZTANN.002P1C6

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	7,533,168	05-12-2009	Pabla et al.	
	2	2020/0153677 (Attorney Docket No. ZTANN.002P1C5) U.S. Application No. 16/449,084 and its entire prosecution history**	05-14-2020	Tannenbaum et al.	
	3	5,423,046	06-06-1995	Nunnelley et al.	
	4	5,881,315	03-09-1999	Cohen	
	5	5,930,768	07-27-1999	Hooban	
	6	6,006,259	11-20-1998	Adelman et al.	
	7	6,067,609	05-23-2000	Meeker et al.	
	8	6,074,427	06-13-2000	Fought et al.	
	9	6,108,699	08-22-2000	Moiin	
	10	6,195,680	02-27-2001	Goldszmidt et al.	
	11	6,202,080	03-13-2001	Lu et al.	
	12	6,546,403	04-08-2003	Carlson, Jr. et al.	
	13	6,578,068	06-10-2003	Bowman-Amuah	
	14	6,604,134	08-05-2003	Haury	
	15	6,691,216	02-10-2004	Kelly et al.	
	16	6,751,698	06-15-2004	Deneroff et al.	
	17	6,782,537	08-24-2004	Blackmore et al.	
	18	6,968,335	11-22-2005	Bayliss et al.	
	19	6,968,359	11-22-2005	Miller	
	20	7,031,944	04-18-2006	Tanioka	
	21	7,093,004	08-15-2006	Bernardin et al.	
	22	7,136,924	11-14-2006	Dauger	
	23	7,139,882	11-21-2006	Suzuoki et al.	
	24	7,174,381	02-06-2007	Gulko et al.	

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

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
	Examiner	Unassigned
SHEET 2 OF 12	Attorney Docket No.	ZTANN.002P1C6

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	25	7,177,874	02-13-2007	Jardin	
	26	7,249,357	07-24-2007	Landman et al.	
	27	7,315,877	01-01-2008	Bhanot et al.	
	28	7,334,232	02-19-2008	Jacobs et al.	
	29	7,437,460	10-14-2008	Chidambaran et al.	
	30	7,472,193	12-30-2008	Dauger	
	31	7,502,915	03-10-2009	Jacob et al.	
	32	7,519,652	04-14-2009	Page et al.	
	33	7,533,141	05-12-2009	Nadgir et al.	
	34	7,533,389	05-12-2009	Verbeke et al.	
	35	7,554,949	06-30-2009	Chen	
	36	7,568,131	07-28-2009	Vertes	
	37	7,634,530	12-15-2009	Dean et al.	
	38	7,644,130	01-05-2010	Magro et al.	
	39	7,698,390	04-13-2010	Harkness et al.	
	40	7,716,323	05-11-2010	Gole et al.	
	41	7,757,236	07-13-2010	Singh	
	42	7,788,314	08-31-2010	Holt	
	43	7,835,022	11-16-2010	Matsumoto	
	44	7,937,455	05-03-2011	Saha et al.	
	45	8,082,289 (Attorney Docket No. ZTANN.002A) U.S. Application No. 11/744,461 and its entire prosecution history**	12-20-2011	Tannenbaum et al.	
	46	8,117,288	02-14-2012	Bhanot et al.	

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

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
	Examiner	Unassigned
SHEET 3 OF 12	Attorney Docket No.	ZTANN.002P1C6

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	47	8,140,612 (Attorney Docket No. ZTANN.002CP1) U.S. Application No. 12/040,519 and its entire prosecution history**	03-20-2012	Tannenbaum et al.	
	48	8,402,080 (Attorney Docket No. ZTANN.001C1) U.S. Application No. 11/749,554 and its entire prosecution history**	03-19-2013	Tannenbaum et al.	
	49	8,402,083 (Attorney Docket No. ZTANN.002CP1C1) U.S. Application No. 12/465,519 and its entire prosecution history**	03-19-2013	Tannenbaum et al.	
	50	8,601,101	12-03-2013	Singh	
	51	8,676,877 (Attorney Docket No. ZTANN.002P1C2) U.S. Application No. 13/423,063 and its entire prosecution history**	03-18-2014	Tannenbaum et al.	
	52	8,849,889 (Attorney Docket No. ZTANN.001C2) U.S. Application No. 13/846,567 and its entire prosecution history**	09-30-2014	Tannenbaum et al.	
	53	9,405,564	08-02-2016	Muellers et al.	
	54	10,333,768 (Attorney Docket No. ZTANN.002P1C3) U.S. Application No. 14/181,112 and its entire prosecution history**	06-25-2019	Tannenbaum et al.	

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

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
	Examiner	Unassigned
SHEET 4 OF 12	Attorney Docket No.	ZTANN.002P1C6

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	55	2002/0049859	04-25-2002	Bruckert et al.	
	56	2003/0005068	01-02-2003	Nickel et al.	
	57	2003/0005266	01-02-2003	Akkary et al.	
	58	2003/0051062	03-13-2003	Circenis	
	59	2003/0135621	07-17-2003	Romagnoli	
	60	2003/0195931	10-16-2003	Dauger	
	61	2003/0195938	10-16-2003	Howard et al.	
	62	2004/0015968	01-22-2004	Neiman et al.	
	63	2004/0098359	05-20-2004	Bayliss et al.	
	64	2004/0098447	05-20-2004	Verbeke et al.	
	65	2004/0110209	06-10-2004	Yokota et al.	
	66	2004/0157203	08-12-2004	Dunk	
	67	2004/0195572	10-07-2004	Kato et al.	
	68	2004/0252710	12-16-2004	Jeter, Jr. et al.	
	69	2004/0254984	12-16-2004	Dinker	
	70	2005/0015460	01-20-2005	Goyle et al.	
	71	2005/0021751	01-27-2005	Block et al.	
	72	2005/0038852	02-17-2005	Howard	
	73	2005/0060237	03-17-2005	Barsness et al.	
	74	2005/0076105	04-07-2005	Keohane et al.	
	75	2005/0097300	05-05-2005	Gildea et al.	
	76	2005/0108394	05-19-2005	Braun et al.	
	77	2005/0154789	07-14-2005	Fellenstein et al.	
	78	2005/0180095	08-18-2005	Ellis	
	79	2005/0188088	08-25-2005	Fellenstein et al.	
	80	2005/0235055	10-20-2005	Davidson	
	81	2006/0026601	02-02-2006	Solt	

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

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
SHEET 5 OF 12		Attorney Docket No. ZTANN.002P1C6

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	82	2006/0053216	03-09-2006	Deokar et al.	
	83	2006/0059473	03-16-2006	Moler	
	84	2006/0106931	05-18-2006	Richou	
	85	2007/0044099	02-22-2007	Rajput	
	86	2007/0073705	03-29-2007	Gray	
	87	2007/0094532	04-26-2007	Sengupta	
	88	2007/0124363	05-31-2007	Lurie et al.	
	89	2008/0250347	10-09-2008	Gray et al.	
	90	2008/0281997	11-13-2008	Archer et al.	

FOREIGN PATENT DOCUMENTS						
Examiner Initials	Cite No.	Foreign Patent Document Country Code-Number-Kind Code Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T ¹
	91	EP 1 368 948	12-10-2003	Prismmedia Networks, Inc.		PCT Eq.
	92	JP 08-087473	02-04-1996	Toshiba Corp		X
	93	JP 11-126196	05-11-1999	Hitachi Ltd.		X-MT
	94	JP 11-328134	11-30-1999	Hitachi Ltd.		X-Abs
	95	JP 2002-117010	04-19-2002	International Business Machine Corp.		X
	96	JP 2004-061359	02-26-2004	Ricoh Co., LTd.		X-Abs
	97	JP 2004-247405	09-02-2004	Semiconductor Energy Lab		X-MT
	98	JP 2005-063033	10-03-2005	Toshiba Corp.		X-Abs
	99	JP 2005-071333	03-17-2005	Network Appliance Inc.		X-MT
	100	WO 2004/086270	10-07-2004	ECI Telecom Ltd.		

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹

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

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
(Multiple sheets used when necessary)	Examiner	Unassigned
SHEET 6 OF 12	Attorney Docket No.	ZTANN.002P1C6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	101	Affidavit of Elizabeth Rosenberg, dated September 1, 2020 in 7 pages.	
	102	AGARWAL et al., "A High Performance Parallel Algorithm for 1-D FFT", Proceedings, Supercomputing '94, Washington, D.C., November 14-18, 1994, pp. 34-40.	
	103	ANNARATONE et al., "The Warp Computer: Architecture, Implementation, and Performance", Dept. of Computer Science, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 1987, pp. 31.	
	104	BÓSA et al., "Task Logging, Rescheduling and Peer Checking in Distributed Maple", RISC-Linz, Johannes Kepler University, Linz, Austria. Technical Report No. 02-10, March 18, 2002, pp. 24.	
	105	CARNS et al., "An Evaluation of Message Passing Implementations on Beowulf Workstations", Aerospace Conference, March 6-13, 1999, IEEE 1999, Vol. 5, pp. 41-54.	
	106	CHU et al., "FFT Algorithms and their Adaptation to Parallel Processing", Linear Algebra and its Applications, Vol. 284, 1998, pp. 95-124.	
	107	CRAMER, et al., "The Development and Integration of a Distributed 3D FFT for a Cluster of Workstations", Proceedings of the 4th Annual Liux Showcase & Conference, Atlanta, GA, October 10-14, 2000, pp. 1-9.	
	108	DALLY et al., "Merrimac: Supercomputing with Streams", SC'03, November 15-21, 2003, Phoenix, AZ, pp. 8.	
	109	DAUGER et al., "'Plug-and-Play' Cluster Computing using Mac OS X", Proceedings. 2003 IEEE International Conference on Cluster Computing, December 1-4, 2003, published January 2004, pp. 6.	
	110	DAUGER et al., "Plug-and-Play Cluster Computing: High-Performance Computing for the Mainstream", IEEE Computing in Science and Engineering, March/April 2005, pp. 27-33.	
	111	Declaration of Henry Tufo, Ph.D., under 37 C.F.R. § 1.68 in Support of Petition for Inter Partes Review of U.S. Patent No. 8,082,289, dated September 18, 2020 in 160 pages.	
	112	Declaration of Sylvia Hall-Ellis Ph.D., for Inter Partes Review of U.S. Patent No. 8,082,289, dated September 18, 2020 in 138 pages.	
	113	Declaration of Wolfgang Schreiner, Ph.D., for Inter Partes Review of U.S. Patent No. 8,082,289, dated September 7, 2020 in 84 pages.	
	114	"Distributed Maple", https://web.archive.org/web/20040216100904/http://www.risc.uni-linz.ac.at/software/distmaple/ as archived February 16, 2004 in 1 page.	
	115	"Distributed Maple Papers", https://web.archive.org/web/20040304035455/http://www.risc.uni-linz.ac.at/software/distmaple/index_1.html as archived March 4, 2004 in 3 pages.	
	116	EREZ, Mattan, "MERRIMAC – High-Performance and Highly-Efficient Scientific Computing with Streams", A Dissertation Submitted to the Department of Electrical Engineering and the Committee on Graduate Studies of Stanford University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy, May 2007, pp. 229	
	117	FAN et al., "GPU Cluster for High Performance Computing", SC '04: Proceedings of the 2004 ACM/IEEE Conference on Supercomputing : 6-12 November 2004, Pittsburgh, PA, USA, pp. 12.	
	118	FRIGO et al., "FFTW User's Manual", Version 2.1.5, March 16, 2003, pp. 74.	

Examiner Signature	Date Considered
<p>*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.</p>	

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
(Multiple sheets used when necessary)	Examiner	Unassigned
SHEET 7 OF 12	Attorney Docket No.	ZTANN.002P1C6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	119	"gridMathematica 1.1: Grid Computing Gets a Speed Boost from Mathematica 5", The Mathematica Journal, Vol. 9, No. 2, August 25, 2003, in 1 page. < https://web.archive.org/web/20040618053429/https://www.mathematica-journal.com/issue/v9i2/newproducts/gridmathematica.html >	
	120	HAMSCHER et al., "Evaluation of Job-Scheduling Strategies for Grid Computing", GRID 2000, 2000, pp. 191-202.	
	121	HAYNOS, Matt, "Perspectives on Grid: Grid Computing -- Next-Generation Distributed Computing", IBM Developer Works, January 27, 2004 in 7 pages, http://www-106.ibm.com/developerworks/library/gr-heritage	
	122	HEAL et al., "Maple V: Learning Guide", Springer, Waterloo Maple Inc., 1996, 1998, pp. 300.	
	123	HILLIS, W. Daniel, "Richard Feynman and The Connection Machine", for Physics Today, originally published 1989, http://longnow.org/essays/richard-feynman-connection-machine/ , printed October 1, 2019, pp. 8.	
	124	"INSTALL", File, Distributed Maple Application Software, June 16, 2000, pp. 5.	
	125	"INSTALL", File, Distributed Maple, March 28, 2003, pp. 4.	
	126	Source Code, "Berlekamp Distinct Degree Factorization", Distributed Maple Application Software from the "distsoft" directory, v. 1.1, July 26, 2001, pp. 58.	
	127	Source Code, "CASA NeighbGraph", Distributed Maple Application Software from the "distsoft" directory, v. 1.6, January 9, 2001, pp. 47.	
	128	Source Code, "Maple Interface to Java Scheduler", Distributed Maple, dist.maple5, v 1.63, March 28, 2003, pp. 28.	
	129	JAIN et al., "Data Clustering: A Review", ACM Computing Surveys, September 1999, Vol. 31, No. 3, pp. 264-323.	
	130	KEPNER et al., "Parallel Matlab: The Next Generation", MIT Lincoln Laboratory, Lexington, MA, August 20, 2004, pp. 2.	
	131	KIM et al., "Introduction to Parallel Programming and pMatlab v2.0", MIT Lincoln Laboratory, Lexington, MA, 2011, pp. 30.	
	132	KOFLER, Michael, "Maple: An Introduction and Reference", Addison-Wesley, 1997, pp. 545.	
	133	KONISHI, et al., "Performance Evaluation of Parallel Computing Tool for MATLAB on PC Grid Environment," IPSJ SIG Technical Reports, August 5, 2005, Vol. 2005, No. 84, pp. 9.	X-Abs
	134	MAEDER, Roman, "Mathematica: Parallel Computing Toolkit - Unleash the Power of Parallel Computing", Wolfram Research, January 2005, pp. 95.	
	135	MATSUMURA, et al. "Construction of Distributed Computing System for Large-Scale Matrices Intended or Reduction of Communication Blocks", August 5, 1998, Vol. 98, No. 18, pp. 9.	X-MT
	136	"MPI: A Message-Passing Interface Standard", Message Passing Interface Forum, June 12, 1995, pp. 239.	

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

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
	Examiner	Unassigned
SHEET 8 OF 12	Attorney Docket No.	ZTANN.002P1C6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	137	PAU et al., "Distributed Mathematica - User and Reference Manual", July 5, 2000, pp. 18.	
	138	"Pingpong MPI Benchmark - SEM vs 'grid'", Dauger Research, Inc., January 2009, 1 page, http://daugerresearch.com/pooch/mathmatica/	
	139	"README File", Distributed Maple Application Software, July 26, 2001, pp. 3.	
	140	"README File", Distributed Maple 1.1.15, March 28, 2003, pp. 5.	
	141	Reference Manual, TRANSPUTER, Inmos, IMS, January 1987, pp. 162.	
	142	SCHREINER et al., "Distributed Maple: Parallel Computer Algebra in Networked Environments", Journal of Symbolic Computation, March 2003, Vol. 35, No. 3, pp. 305-347.	
	143	SCHREINER, Wolfgang, "Distributed Maple - User and Reference Manual (V 1.1.12)", Research Institute for Symbolic Computation (RISC-Linz), Johannes Kepler University, A-4040 Linz, Austria, July 6, 2001, pp. 1-37.	
	144	SHERWANI et al., "Libra: A Computational Economy-Based Job Scheduling System for Clusters", Software-Practice and Experience, 2004, Vol. 34, pp. 573-590.	
	145	TATEBE et al., "Efficient Implementation of MPI Using Remote Memory Write," Transactions of Information Processing Society of Japan, May 1999, Vol. 40, No. 5, pp. 14.	X-Abs
	146	TEPENEU et al., "MathGridLink - A Bridge Between Mathematica and 'the Grid'", Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba, 2003, pp. 4.	
	147	U.S. Provisional Application No. 60/799,474, filed May 10, 2006, titled "Graphical Interface for Monitoring Status of a Concurrent Computing Process", in 45 pages.	
	148	U.S. Provisional Application No. 60/813,738, filed June 13, 2006, titled "Pooch/Mathmatica Interface", in 6 pages.	
	149	U.S. Provisional Application No. 60/850,908, filed October 11, 2006, titled "Poochmpi Toolkit for Mathematica", in 25 pages.	
	150	"Wolfram gridMathematica™", < http://www.wolfram.com/products/gridmathematica/ > as printed October 3, 2007 in 2 pages.	
	151	WOLFRAM, Stephen, "The Mathematica Book", 5th Edition, Wolfram Research, Inc., 2003, pp. 1486.	
	152	NVIDIA Power of Attorney, October 5, 2020, pp. 2.	
	153	Ex. 2001 - Singh Declaration, February 9, 2021, pp. 42.	
	154	Ex. 2002 - Singh CV, pp. 4.	

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

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
	Examiner	Unassigned
SHEET 9 OF 12	Attorney Docket No.	ZTANN.002P1C6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	155	Ex. 2007 - Bhansali Declaration, February 9, 2021, pp. 5.	
	156	Ex. 2008 - Bancroft Declaration, February 9, 2021, pp. 13.	
	157	Ex. 2009 - SEM Poster, 2009, pp. 1.	
	158	Ex. 2011 - SEM Manual, pp. 20.	
	159	Ex. 2012 - SEM White Paper, 2007, pp. 20.	
	160	Ex. 2013 - SET White Paper, pp. 7.	
	161	Ex. 2014 - SET Manual, pp. 67.	
	162	Ex. 2015 - SET Presentation, December 2011, pp. 54.	
	163	Ex. 2016 - SET Datasheet, pp. 4.	
	164	Ex. 2017 - Parallel Bars the Economist, June 4, 2011, pp. 9.	
	165	Ex. 2018 - SET Presentation, HPC for the Missing Middle!, December 2011, pp. 4.	
	166	Ex. 2019 - DOE Review of Plasma Visualization Project, 2015, pp. 5.	
	167	Ex. 2020 - SET Data Sheet given to Jones, pp. 4.	
	168	Ex. 2021 - DOE Review of Expanded Validation Project, 2015, pp. 5	
	169	Ex. 2022 - DOE Review of SET Windows Implementation, 2015, pp. 5.	
	170	Ex. 2023 - SEM an Independent Report, 2008, pp. 2.	
	171	Ex. 2024 - Multicore Parallelism presentation, February 2011, pp. 2.	
	172	Ex. 2025 - DOE SET Windows Project Narrative, pp. 2.	
	173	Ex. 2026 - DOE Topics, pp. 11.	

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

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
	Examiner	Unassigned
SHEET 10 OF 12	Attorney Docket No.	ZTANN.002P1C6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	174	Ex. 2027 - Dager Declaration, February 9, 2021, pp. 30.	
	175	Ex. 2028 - Nexus Claim Chart - SEM - 768 Patent, pp. 13.	
	176	Ex. 2029 - Nexus Claim Chart - SET - 768 Patent, pp. 17.	
	177	Exhibit 1002 - File History of USP 10,333,768 B2, pp. 627.	
	178	Exhibit 1005 - Declaration of Henry Tufo, October 2, 2020, pp. 132.	
	179	Exhibit 1006 - Declaration of Wolfgang Schreiner, October 1, 2020, pp. 85.	
	180	Exhibit 1007 - Declaration of Sylvia Hall-Ellis, October 5, 2020, pp. 109.	
	181	Exhibit 1019 - SUN Debuts UltraSPARC IV (2003), pp. 2.	
	182	Exhibit 1020 - AMD to Unveil Dual-Core PC Chips (2005), May 31, 2005, pp. 1.	
	183	Ex. 2001 - Singh Declaration, February 9, 2021, pp. 42.	
	184	Ex. 2030 - Dager Declaration, February 9, 2021, pp. 31.	
	185	Ex. 2031 - Nexus Claim Chart - SEM - 768 Patent, pp. 13.	
	186	Ex. 2032 - Nexus Claim Chart - SET - 768 Patent, pp. 40.	
	187	Exhibit 1105 - Declaration of Henry Tufo, October 9, 2020, pp. 147.	
	188	Exhibit 1106 - Declaration of Wolfgang Schreiner, October 1, 2020, pp. 85.	
	189	Exhibit 1107 - Declaration of Sylvia Hall-Ellis, October 9, 2020, pp. 255.	
	190	Exhibit 1134 - Maple 5 - PC Magazine (1992)	
	191	Exhibit 1135 - Maple V Looks Better with Improved Graphics - PC Magazine	
	192	Exhibit 1136 - Nayak, A Library based compiler to execute MATLAB Programs	

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

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
	Examiner	Unassigned
SHEET 11 OF 12	Attorney Docket No.	ZTANN.002P1C6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	193	Ex. 2001 - Singh Declaration, February 9, 2021, pp. 41.	
	194	Ex. 2033 - Dager Declaration, February 10, 2021, pp. 30.	
	195	Ex. 2034 - Nexus Claim Chart - SEM - 612 Patent, pp. 11.	
	196	Ex. 2035 - Nexus Claim Chart - SET - 612 Patent, pp. 14.	
	197	Exhibit 1002 - File History of USP 8,140,612 B2	
	198	Exhibit 1005 - Declaration of Henry Tufo, October 16, 2020, pp. 127.	
	199	Exhibit 1006 - Declaration of Wolfgang Schreiner, October 1, 2020, pp. 86.	
	200	Exhibit 1007 - Declaration of Sylvia Hall-Ellis, October 16, 2020, pp. 178.	
	201	Exhibit 1019 - Deitel - Operating Systems, 2004, pp. 1270 (2parts).	
	202	Exhibit 1020 - Chen et al., TH-MPI_ OS Kernel Integrated Fault Tolerant MPI (2001), pp. 75-82.	
	203	Exhibit 1026 - Jin et al., LiMIC_ Support for High-Performance MPI Intra-Node Communication on Linux, 2005, pp. 184-191.	
	204	Ex. 2001 - Singh Declaration, February 17, 2021, pp. 40.	
	205	Ex. 2036 - Dager Declaration, February 15, 2021, pp. 30.	
	206	Ex. 2037 - Nexus Claim Chart - SEM - 877 Patent, pp. 33.	
	207	Ex. 2038 - Nexus Claim Chart - SET - 877 Patent, pp. 50.	
	208	Exhibit 1002 - File History of USP 8,676,877 B2	
	209	Exhibit 1005 - Declaration of Henry Tufo, October 23, 2020, pp. 130.	
	210	Exhibit 1006 - Declaration of Wolfgang Schreiner, October 1, 2020, pp. 86.	
	211	Exhibit 1007 - Declaration of Sylvia Hall-Ellis, October 23, 2020, pp. 186.	

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

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Multiple sheets used when necessary)</i>	Application No.	17/374864
	Filing Date	July 13, 2021
	First Named Inventor	Tannenbaum, Zvi
	Art Unit	2447
	Examiner	Unassigned
SHEET 12 OF 12	Attorney Docket No.	ZTANN.002P1C6

NON PATENT LITERATURE DOCUMENTS			
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, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	212	Extended European Search Report received in European Application No. 17207443.7 (ZTANN.002EPD1), dated August 8, 2018, in 9 pages.	
	213	International Search Report and Written Opinion received in PCT Application No. PCT/US2006/037275 (ZTANN.001VPC), as mailed September 24, 2007 in 5 pages.	
	214	International Search Report and Written Opinion received in PCT Application No. PCT/US2007/070585 (ZTANN.002VPC), as mailed September 11, 2008 in 5 pages.	

53865465

** Signature indicates consideration of publication and file history. The Examiner has access to these materials through the PTO computer systems. If additional copies are desired, please notify Applicants through their attorneys.

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

T¹ - Place a check mark in this area when an English language Translation is attached.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Inventor : Zvi Tannenbaum
App. No. : 17/374,864
Filed : July 13, 2021
For : CLUSTER COMPUTING
Examiner : Unassigned
Art Unit : 2449
Conf. No. : 1083

PRELIMINARY AMENDMENT

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

Dear Commissioner:

Please enter the following amendment prior to the examination of this application.

Amendments to the Claims begin on page 2.

Remarks begin on page 9.

AMENDMENTS TO THE CLAIMS

1. (Canceled)
2. (New) A computer cluster comprising:
 - a first processor;
 - a second processor;
 - a third processor;
 - at least one computer-readable medium in communication at least one of the first processor, the second processor, or the third processor;
 - a first kernel residing in the at least one computer-readable medium, said first kernel configured to translate commands into code for execution on the first processor;
 - a first cluster node module residing in the at least one computer-readable medium, said first cluster node module configured to:
 - receive first commands from the user interface without the first commands first passing through the first kernel, and
 - after receiving the first commands from the user interface, send second commands to the first kernel;
 - a second kernel residing in the at least one computer-readable medium, said second kernel configured to translate commands into code for execution on the second processor;
 - a second cluster node module residing in the at least one computer-readable medium, said second cluster node module configured to send commands to the second kernel and communicates with the first cluster node module;
 - a third kernel residing in the at least one computer-readable medium, said third kernel configured to translate commands into code for execution on the third processor;
 - and
 - a third cluster node module residing in the at least one computer-readable medium, said third cluster node module configured to send commands to the third kernel and configured to communicate with the first cluster node module and the second cluster node module;

wherein each of the first, second, and third cluster node modules can transmit tasks to any other of the first, second, and third cluster node modules without transmitting the tasks through a master node, and wherein the first cluster node module comprises a data structure in which messages originating from the second and third cluster node modules are stored.

3. (New) The computer cluster of claim 2, wherein the first cluster node module is logically positioned between a user interface and the first kernel.

4. (New) The computer cluster of claim 2, wherein the at least one computer-readable medium comprises a first computer-readable medium coupled to at least the first processor and a second computer-readable medium coupled to at least the second processor, wherein the first kernel and the first cluster node module reside in the first computer-readable medium, and wherein the second kernel and the second cluster node module reside in the second computer-readable medium.

5. (New) The computer cluster of claim 2, further comprising a cluster configuration module that initializes at least one of the first cluster node module, the second cluster node module, or the third cluster node module.

6. (New) The computer cluster of claim 2, wherein the second cluster node module communicates with the first cluster node module using commands that implement at least a portion of a Message-Passing Interface.

7. (New) The computer cluster of claim 2, wherein the data structure of the first cluster node module comprises a received message queue.

8. (New) The computer cluster of claim 7, wherein the first cluster node module further comprises a message receiving queue, in which data specifying the location to which the first cluster node module can expect to receive an expression and an identifier of a processor from which the expression is expected to be sent are stored.

9. (New) The computer cluster of claim 2, wherein the second cluster node module further comprises a received message queue in which messages originating from the first and third cluster node modules are stored.

10. (New) The computer cluster of claim 2, wherein the first cluster node module comprises an advanced functions module, and wherein the advanced functions module comprises a call that evaluates an expression across the computer cluster in parallel.

11. (New) The computer cluster of claim 10, wherein the advanced functions module comprises a call that calculates a Fourier transform across the computer cluster in parallel.

12. (New) The computer cluster of claim 10, wherein the advanced functions module comprises one or more calls that evaluate matrix operations across the computer cluster in parallel.

13. (New) The computer cluster of claim 2, wherein the second commands comprise at least one of the first commands.

14. (New) A computer cluster comprising:

- a first processor;

- a second processor;

- a third processor;

- at least one computer-readable medium in communication at least one of the first processor, the second processor, or the third processor;

- a first kernel residing in the at least one computer-readable medium, said first kernel configured to translate commands into code for execution on the first processor;

- a first cluster node module residing in the at least one computer-readable medium, said first cluster node module configured to:

- first receive first commands from the user interface, and

- after receiving the first commands from the user interface, send second commands to the first kernel;

- a second kernel residing in the at least one computer-readable medium, said second kernel configured to translate commands into code for execution on the second processor;

- a second cluster node module residing in the at least one computer-readable medium, said second cluster node module configured to send commands to the second kernel and communicates with the first cluster node module;

- a third kernel residing in the at least one computer-readable medium, said third kernel configured to translate commands into code for execution on the third processor;

and

- a third cluster node module residing in the at least one computer-readable medium, said third cluster node module configured to send commands to the third kernel

and configured to communicate with the first cluster node module and the second cluster node module;

wherein each of the first, second, and third cluster node modules can transmit tasks to any other of the first, second, and third cluster node modules without transmitting the tasks through a master node, and wherein the first cluster node module comprises a data structure in which messages originating from the second and third cluster node modules are stored.

15. (New) The computer cluster of claim 14, wherein the first cluster node module is logically positioned between a user interface and the first kernel.

16. (New) The computer cluster of claim 14, further comprising a communications network for connecting at least two of the first processor, the second processor, or the third processor to one another.

17. (New) The computer cluster of claim 14, wherein the at least one computer-readable medium comprises a first computer-readable medium coupled to at least the first processor and a second computer-readable medium coupled to at least the second processor, wherein the first kernel and the first cluster node module reside in the first computer-readable medium, and wherein the second kernel and the second cluster node module reside in the second computer-readable medium.

18. (New) The computer cluster of claim 14, further comprising a cluster configuration module that initializes at least one of the first cluster node module, the second cluster node module, or the third cluster node module.

19. (New) The computer cluster of claim 14, wherein the data structure of the first cluster node module comprises a received message queue.

20. (New) The computer cluster of claim 19, wherein the first cluster node module further comprises a message receiving queue, in which data specifying the location to which the first cluster node module can expect to receive an expression and an identifier of a processor from which the expression is expected to be sent are stored.

21. (New) The computer cluster of claim 14, wherein the first processor and the second processor reside on the same die.

22. (New) The computer cluster of claim 14, wherein the first cluster node module comprises an advanced functions module, and wherein the advanced functions module comprises a call that evaluates an expression across the computer cluster in parallel.

23. (New) The computer cluster of claim 21, wherein the advanced functions module comprises a call that calculates a Fourier transform across the computer cluster in parallel.

24. (New) The computer cluster of claim 21, wherein the advanced functions module comprises one or more calls that evaluate matrix operations across the computer cluster in parallel.

25. (New) A computer cluster comprising:

a first processor;

a second processor;

a third processor;

at least one computer-readable medium in communication at least one of the first processor, the second processor, or the third processor;

a first kernel residing in the at least one computer-readable medium, said first kernel configured to translate commands into code for execution on the first processor;

a first cluster node module residing in the at least one computer-readable medium, said first cluster node module configured to send the commands to the first kernel, wherein the first cluster node module is logically positioned between a user interface and the first kernel;

a second kernel residing in the at least one computer-readable medium, said second kernel configured to translate commands into code for execution on the second processor;

a second cluster node module residing in the at least one computer-readable medium, said second cluster node module configured to send commands to the second kernel and communicates with the first cluster node module;

a third kernel residing in the at least one computer-readable medium, said third kernel configured to translate commands into code for execution on the third processor; and

a third cluster node module residing in the at least one computer-readable medium, said third cluster node module configured to send commands to the third kernel

and configured to communicate with the first cluster node module and the second cluster node module;

wherein each of the first, second, and third cluster node modules can transmit tasks to any other of the first, second, and third cluster node modules without transmitting the tasks through a master node, and wherein the first cluster node module comprises a data structure in which messages originating from the second and third cluster node modules are stored.

26. (New) The computer cluster of claim 25, wherein the first processor and the second processor reside on the same die.

27. (New) The computer cluster of claim 25, wherein the first cluster node module comprises an advanced functions module, and wherein the advanced functions module comprises a call that evaluates an expression across the computer cluster in parallel, wherein the advanced functions module comprises one or more calls that evaluate matrix operations across the computer cluster in parallel.

28. (New) A computer cluster comprising:

a first processor;

a second processor;

a third processor;

at least one computer-readable medium in communication at least one of the first processor, the second processor, or the third processor;

a first kernel residing in the at least one computer-readable medium, said first kernel configured to translate commands into code for execution on the first processor;

a first cluster node module residing in the at least one computer-readable medium, said first cluster node module configured to send the commands to the first kernel, wherein the first cluster node module is logically positioned between a user interface and the first kernel;

a second kernel residing in the at least one computer-readable medium, said second kernel configured to translate commands into code for execution on the second processor;

a second cluster node module residing in the at least one computer-readable medium, said second cluster node module configured to send commands to the second kernel and communicates with the first cluster node module;

a third kernel residing in the at least one computer-readable medium, said third kernel configured to translate commands into code for execution on the third processor; and

a third cluster node module residing in the at least one computer-readable medium, said third cluster node module configured to send commands to the third kernel and configured to communicate with the first cluster node module and the second cluster node module;

wherein each of the first, second, and third cluster node modules can directly transmit tasks to any other of the first, second, and third cluster node modules, and wherein the first cluster node module comprises a data structure in which messages originating from the second and third cluster node modules are stored.

29. (New) The computer cluster of claim 28, wherein the first cluster node module comprises an advanced functions module, and wherein the advanced functions module comprises a call that evaluates an expression across the computer cluster in parallel, wherein the advanced functions module comprises one or more calls that evaluate matrix operations across the computer cluster in parallel.

30. (New) The computer cluster of claim 28, further comprising a communications network for connecting at least two of the first processor, the second processor, or the third processor to one another.

31. (New) The computer cluster of claim 28, wherein at least one of the first, second, or third cluster node modules transmits tasks directly to another of the first, second, and third cluster node modules via one or more network components.

Application No.: 17/374,864
Filing Date: July 13, 2021

REMARKS

Applicant cancels claim 1 without prejudice or disclaimer. Claims 2-31 are new. Applicant respectfully requests consideration of the merits following entry of the amendments set forth above.

Other Applications of Assignee

Applicant wishes to draw the Examiner's attention to the following other applications of the present application's assignee.

Docket No.	Serial No.	Title	Filed
ZTANN.002A	11/744461	CLUSTER COMPUTING SUPPORT FOR APPLICATION PROGRAMS	05/04/07
ZTANN.002CP1	12/040519	CLUSTER COMPUTING SUPPORT FOR APPLICATION PROGRAMS	02/29/08
ZTANN.002CP1C1	12/465519	AUTOMATIC CLUSTER NODE DISCOVERY AND CONFIGURATION	05/13/09
ZTANN.002P1C2	13/423063	CLUSTER COMPUTING USING SPECIAL PURPOSE MICROPROCESSORS	03/16/12
ZTANN.002P1C3	14/181112	CLUSTER COMPUTING	02/14/14
ZTANN.002P1C5	16/449084	CLUSTER COMPUTING	06/21/19
ZTANN.002P1C7	17/374789	CLUSTER COMPUTING	07/13/21

Please charge any additional fees, including any fees for additional extension of time, or credit overpayment to Deposit Account No. 11-1410.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: August 4, 2021

By: /Jordan M. Cox/

Jordan M. Cox
Registration No. 70,875
Registered Practitioner
(949) 760-0404

32132313

Electronic Patent Application Fee Transmittal				
Application Number:		17374864		
Filing Date:		13-Jul-2021		
Title of Invention:		CLUSTER COMPUTING		
First Named Inventor/Applicant Name:		Zvi Tannenbaum		
Filer:		Jordan Michael Cox/Janet Teeters		
Attorney Docket Number:		ZTANN.002P1C6		
Filed as Small Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
CLAIMS IN EXCESS OF 20	2202	10	50	500
INDEPENDENT CLAIMS IN EXCESS OF 3	2201	1	240	240
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				740

Electronic Acknowledgement Receipt	
EFS ID:	43424847
Application Number:	17374864
International Application Number:	
Confirmation Number:	1083
Title of Invention:	CLUSTER COMPUTING
First Named Inventor/Applicant Name:	Zvi Tannenbaum
Customer Number:	20995
Filer:	Jordan Michael Cox/Melanie Terrazas
Filer Authorized By:	Jordan Michael Cox
Attorney Docket Number:	ZTANN.002P1C6
Receipt Date:	04-AUG-2021
Filing Date:	13-JUL-2021
Time Stamp:	15:36:23
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$740
RAM confirmation Number	E202184F36533441
Deposit Account	111410
Authorized User	Melanie Terrazas
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: 37 CFR 1.16 (National application filing, search, and examination fees) 37 CFR 1.17 (Patent application and reexamination processing fees)	

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		ZTANN-002P1C6_IDS.pdf	222488	yes	13
			15eeb624949b7b86ddec58275d28fae489893764		
	Multipart Description/PDF files in .zip description				
	Document Description		Start		End
	Transmittal Letter		1		1
	Information Disclosure Statement (IDS) Form (SB08)		2		13
Warnings:					
Information:					
2		ZTANN-002P1C6_Prelim_Amen d.pdf	52364	yes	9
			5e3d82341607eb3a8b1d8171b537297c2190104e		
	Multipart Description/PDF files in .zip description				
	Document Description		Start		End
	Preliminary Amendment		1		1
	Claims		2		8
	Applicant Arguments/Remarks Made in an Amendment		9		9
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	40103	no	2
			6bab136dad1ce8a50ad3e99f673c68f543e8d706		
Warnings:					
Information:					
Total Files Size (in bytes):			314955		

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.

INFORMATION DISCLOSURE STATEMENT

First Inventor: Zvi Tannenbaum	Conf. No.: 1083
App. No.: 17/374864	Filed: July 13, 2021
Examiner: Unassigned	Art Unit: 2447
Title: CLUSTER COMPUTING	

Mail Stop Amendment

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

References and Listing

Pursuant to 37 CFR 1.56, an Information Disclosure Statement listing references is provided herewith. References numbered 3 - 214 are of record in U.S. patent application No. 16/449084, filed June 21, 2019, which is relied upon for an earlier filing date under 35 USC 120. Accordingly, copies of references numbered 3 - 214 are not submitted pursuant to 37 CFR 1.98(d).

No Disclaimers

To the extent that anything in the Information Disclosure Statement or the listed references could be construed as a disclaimer of any subject matter supported by the present application, Applicant hereby rescinds and retracts such disclaimer.

Timing of Disclosure

This Information Disclosure Statement is being filed before the receipt of a First Office Action on the merits, and presumably no fee is required. If a First Office Action on the merits was mailed before the mailing date of this Statement, the Commissioner is authorized to charge the fee set forth in 37 CFR 1.17(p) to Deposit Account No. 11-1410.

Respectfully submitted,
KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: August 4, 2021

By: /Jordan M. Cox/
Jordan M. Cox
Registration No. 70,875
Registered Practitioner
(949) 760-0404

53912813

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875				Application or Docket Number 17/374,864		Filing Date 07/13/2021		☐ To be Mailed	
ENTITY: ☐ LARGE ☒ SMALL ☐ MICRO									
APPLICATION AS FILED - PART I									
	(Column 1)	(Column 2)							
FOR	NUMBER FILED	NUMBER EXTRA			RATE (\$)			FEE (\$)	
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A			N/A				
☐ SEARCH FEE (37 CFR 1.16(k), (l), or (m))	N/A	N/A			N/A				
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A			N/A				
TOTAL CLAIMS (37 CFR 1.16(j))	minus 20 =	*			x \$50 =				
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*			x \$240 =				
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 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			
APPLICATION AS AMENDED - PART II									
	(Column 1)		(Column 2)	(Column 3)					
AMENDMENT	08/04/2021	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA			RATE (\$)	ADDITIONAL FEE (\$)
	Total (37 CFR 1.16(j))	* 30	Minus	** 20	= 10			x \$ 50 =	500
	Independent (37 CFR 1.16(h))	* 4	Minus	*** 3	= 1			x \$ 240 =	240
	☐ Application Size Fee (37 CFR 1.16(s))								
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))								
						TOTAL ADD'L FEE		740	
AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA			RATE (\$)	ADDITIONAL FEE (\$)
	Total (37 CFR 1.16(j))	*	Minus	**	=			x \$ 0 =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=			x \$ 0 =	
	☐ Application Size Fee (37 CFR 1.16(s))								
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))								
						TOTAL ADD'L FEE			
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.						SLIE			
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".						/SHARON M WEST/			
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".									
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.									

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

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

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875						Application or Docket Number 17/374,864	
APPLICATION AS FILED - PART I							
(Column 1)		(Column 2)		SMALL ENTITY		OTHER THAN SMALL ENTITY	
FOR	NUMBER FILED	NUMBER EXTRA	RATE(\$)	FEE(\$)		RATE(\$)	FEE(\$)
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	80		N/A	
SEARCH FEE (37 CFR 1.16(k), (l), or (m))	N/A	N/A	N/A	350		N/A	
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	400		N/A	
TOTAL CLAIMS (37 CFR 1.16(j))	1	minus 20 = *	x 50 =	0.00	OR		
INDEPENDENT CLAIMS (37 CFR 1.16(h))	1	minus 3 = *	x 240 =	0.00			
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 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).			0.00			
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				0.00			
			TOTAL	830		TOTAL	
* If the difference in column 1 is less than zero, enter "0" in column 2.							
APPLICATION AS AMENDED - PART II							
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY	
AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)		RATE(\$)
	Total (37 CFR 1.16(i))	*	Minus **	=	x	=	OR
	Independent (37 CFR 1.16(h))	*	Minus ***	=	x	=	OR
	Application Size Fee (37 CFR 1.16(s))					OR	
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					OR	
			TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY	
AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)		RATE(\$)
	Total (37 CFR 1.16(i))	*	Minus **	=	x	=	OR
	Independent (37 CFR 1.16(h))	*	Minus ***	=	x	=	OR
	Application Size Fee (37 CFR 1.16(s))					OR	
	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 found in the appropriate box in column 1.							



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	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO.	TOT CLAIMS	IND CLAIMS
17/374,864	07/13/2021	2447	830	ZTANN.002PIC6	1	1

CONFIRMATION NO. 1083

FILING RECEIPT



0000000127288587

20995
KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

Date Mailed: 07/28/2021

Receipt is acknowledged of this non-provisional utility patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF FIRST INVENTOR, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection.

Please verify the accuracy of the data presented on this receipt. If an error is noted on this Filing Receipt, please submit a written request for a corrected Filing Receipt, including a properly marked-up ADS showing the changes with strike-through for deletions and underlining for additions. If you received a "Notice to File Missing Parts" or other Notice requiring a response for this application, please submit any request for correction to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections provided that the request is grantable.

Inventor(s)

Zvi Tannenbaum, Palo Alto, CA;
Dean E. Dager, Huntington Beach, CA;

Applicant(s)

Advanced Cluster Systems, Inc., Newport Beach, CA;

Power of Attorney: The patent practitioners associated with Customer Number 20995

Domestic Priority data as claimed by applicant

This application is a CON of 16/449,084 06/21/2019
which is a CON of 14/181,112 02/14/2014 PAT 10333768
which is a CON of 13/423,063 03/16/2012 PAT 8676877
which is a CON of 12/040,519 02/29/2008 PAT 8140612
which is a CIP of 11/744,461 05/04/2007 PAT 8082289
which claims benefit of 60/813,738 06/13/2006
and claims benefit of 60/850,908 10/11/2006

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see <http://www.uspto.gov> for more information.) - None.

Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

Permission to Access Application via Priority Document Exchange: Yes

Permission to Access Search Results: Yes

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

If Required, Foreign Filing License Granted: 07/27/2021

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 17/374,864**

Projected Publication Date: 11/04/2021

Non-Publication Request: No

Early Publication Request: No

**** SMALL ENTITY ****

Title

CLUSTER COMPUTING

Preliminary Class

709

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative,

page 2 of 4

this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop

technology, manufacture products, deliver services, and grow your business, visit <http://www.SelectUSA.gov> or call +1-202-482-6800.

SCORE Placeholder Sheet for IFW Content

Application Number: 17374864

Document Date: 07/13/2021

The presence of this form in the IFW record indicates that the following document type was received in electronic format on the date identified above. This content is stored in the SCORE database.

Since this was an electronic submission, there is no physical artifact folder, no artifact folder is recorded in PALM, and no paper documents or physical media exist. The TIFF images in the IFW record were created from the original documents that are stored in SCORE.

- Drawing

At the time of document entry (noted above):

- USPTO employees may access SCORE content via DAV or via the SCORE web page.
- External customers may access SCORE content via PAIR using the Supplemental Content tab.

TRANSMITTAL FOR POWER OF ATTORNEY TO ONE OR MORE REGISTERED PRACTITIONERS

NOTE: This form is to be submitted with the Power of Attorney by Applicant form (PTO/AIA/82B) to identify the application to which the Power of Attorney is directed, in accordance with 37 CFR 1.5, unless the application number and filing date are identified in the Power of Attorney by Applicant form. If neither form PTO/AIA/82A nor form PTO/AIA/82B identifies the application to which the Power of Attorney is directed, the Power of Attorney will not be recognized in the application.

Application Number	Unassigned
Filing Date	Herewith
First Named Inventor	Zvi Tannenbaum
Title	CLUSTER COMPUTING
Art Unit	Unassigned
Examiner Name	Unassigned
Attorney Docket Number	ZTANN.002P1C6

SIGNATURE of Applicant or Patent Practitioner

Signature	/Jordan M. Cox/	Date (Optional)	6/21/2021
Name	Jordan M. Cox	Registration Number	70,875
Title (if Applicant is a juristic entity)			
Applicant Name (if Applicant is a juristic entity)			

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4(d) for signature requirements and certifications. If more than one applicant, use multiple forms.



*Total of 1 forms are submitted.

This collection of information is required by 37 CFR 1.131, 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.

Doc Code: PA..

Document Description: Power of Attorney

PTO/AIA/82B (07-13)

Approved for use through 11/30/2014. OMB 0851-0051

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 BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number

Filing Date

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)



I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above:

20995

OR



I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:



The address associated with the above-mentioned Customer Number

OR



The address associated with Customer Number:

OR

Firm or
Individual Name

Address

City

State

Zip

Country

Telephone

Email

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box):

Advanced Cluster Systems, Inc.



Inventor or Joint Inventor (title not required below)



Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)



Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)



Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature

Date (Optional)

Name

Zvi Tannenbaum

Title

Chief Executive Officer

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.



Total of 1 forms are submitted.

This collection of information is required by 37 CFR 1.131, 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.

Electronic Patent Application Fee Transmittal				
Application Number:				
Filing Date:				
Title of Invention:		CLUSTER COMPUTING		
First Named Inventor/Applicant Name:		Zvi Tannenbaum		
Filer:		Jordan Michael Cox/Janet Teeters		
Attorney Docket Number:		ZTANN.002P1C6		
Filed as Small Entity				
Filing Fees for Track I Prioritized Examination - Nonprovisional Application under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
UTILITY FILING FEE (ELECTRONIC FILING)	4011	1	80	80
UTILITY SEARCH FEE	2111	1	350	350
UTILITY EXAMINATION FEE	2311	1	400	400
REQUEST FOR PRIORITIZED EXAMINATION	2817	1	2100	2100
Pages:				
Claims:				
Miscellaneous-Filing:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
PUBL. FEE- EARLY, VOLUNTARY, OR NORMAL	1504	1	0	0
PROCESSING FEE, EXCEPT PROV. APPLS.	2830	1	70	70
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				3000

Electronic Acknowledgement Receipt	
EFS ID:	43239526
Application Number:	17374864
International Application Number:	
Confirmation Number:	1083
Title of Invention:	CLUSTER COMPUTING
First Named Inventor/Applicant Name:	Zvi Tannenbaum
Customer Number:	20995
Filer:	Jordan Michael Cox/Sandra Autry
Filer Authorized By:	Jordan Michael Cox
Attorney Docket Number:	ZTANN.002P1C6
Receipt Date:	13-JUL-2021
Filing Date:	
Time Stamp:	19:51:51
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$ 3000
RAM confirmation Number	E20217CJ55312936
Deposit Account	111410
Authorized User	Sandra Autry
<p>The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:</p> <p>37 CFR 1.16 (National application filing, search, and examination fees)</p> <p>37 CFR 1.17 (Patent application and reexamination processing fees)</p>	

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	TrackOne Request	ZTANN-002P1C6_Track_1_Request.pdf	132513	no	2
			6f3c5e1b527bc83c0ee41a23d4628f8245b5bace		
Warnings:					
Information:					
2	Application Data Sheet	ZTANN-002P1C6_ADS.pdf	1256623	no	9
			60ac84ff4d4cd59eff397faab1457024d75b3854		
Warnings:					
Information:					
3		ZTANN-002P1C6_Specification.pdf	266340	yes	47
			87ea64da1ad09505e29c2a00c2c63ca543cbc3e8		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Specification		1	45	
	Claims		46	46	
	Abstract		47	47	
Warnings:					
Information:					
4	Drawings-other than black and white line drawings	ZTANN-002P1C6_Drawings.pdf	194697	no	5
			897b2d4695815a42f77fb51eab8fca23c524fa0f		
Warnings:					
Information:					
5	Oath or Declaration filed	ZTANN-002P1C6_Declarations.pdf	250287	no	2
			e39d32f0a5cb54238302d0cf29f329d0086c79bd		

Warnings:					
Information:					
6	Power of Attorney	ZTANN-002P1C6_POA.pdf	1277154	no	2
			3c538dcf8bfbd0cfb7a6568681b72002dabbc538		
Warnings:					
Information:					
7	Fee Worksheet (SB06)	fee-info.pdf	40133	no	2
			b9b9a5907e8b5a28424d9690609facfa2841eb4a		
Warnings:					
Information:					
Total Files Size (in bytes):			3417747		
<p>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.</p> <p><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.</p> <p><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.</p> <p><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.</p>					

**CERTIFICATION AND REQUEST FOR PRIORITIZED EXAMINATION
UNDER 37 CFR 1.102(e)** (Page 1 of 1)

First Named Inventor:	Zvi Tannenbaum	Nonprovisional Application Number (if known):	Unassigned
Title of Invention:	CLUSTER COMPUTING		

APPLICANT HEREBY CERTIFIES THE FOLLOWING AND REQUESTS PRIORITIZED EXAMINATION FOR THE ABOVE-IDENTIFIED APPLICATION.

1. The processing fee set forth in 37 CFR 1.17(i)(1) and the prioritized examination fee set forth in 37 CFR 1.17(c) have been filed with the request. The publication fee requirement is met because that fee, set forth in 37 CFR 1.18(d), is currently \$0. The basic filing fee, search fee, and examination fee are filed with the request or have been already been paid. I understand that any required excess claims fees or application size fee must be paid for the application.
2. I understand that the application may not contain, or be amended to contain, more than four independent claims, more than thirty total claims, or any multiple dependent claims, and that any request for an extension of time will cause an outstanding Track I request to be dismissed.
3. The applicable box is checked below:
 - I. ☒ **Original Application (Track One) - Prioritized Examination under § 1.102(e)(1)**
 - i. (a) The application is an original nonprovisional utility application filed under 35 U.S.C. 111(a). This certification and request is being filed with the utility application via EFS-Web.
---OR---
(b) The application is an original nonprovisional plant application filed under 35 U.S.C. 111(a). This certification and request is being filed with the plant application in paper.
 - ii. An executed inventor's oath or declaration under 37 CFR 1.63 or 37 CFR 1.64 for each inventor, or the application data sheet meeting the conditions specified in 37 CFR 1.53(f)(3)(i) is filed with the application.
 - II. ☐ **Request for Continued Examination - Prioritized Examination under § 1.102(e)(2)**
 - i. A request for continued examination has been filed with, or prior to, this form.
 - ii. If the application is a utility application, this certification and request is being filed via EFS-Web.
 - iii. The application is an original nonprovisional utility application filed under 35 U.S.C. 111(a), or is a national stage entry under 35 U.S.C. 371.
 - iv. This certification and request is being filed prior to the mailing of a first Office action responsive to the request for continued examination.
 - v. No prior request for continued examination has been granted prioritized examination status under 37 CFR 1.102(e)(2).

Signature <u>/Jordan M. Cox/</u>	Date <u>6/21/2021</u>
Name (Print/Typed) <u>Jordan M. Cox</u>	Practitioner Registration Number <u>70,875</u>
Note: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4(d) for signature requirements and certifications. Submit multiple forms if more than one signature is required.*	
☒ *Total of <u>1</u> forms are submitted.	

Privacy Act Statement

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

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

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

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.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	ZTANN.002P1C6
		Application Number	
Title of Invention	CLUSTER COMPUTING		
<p>The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.</p> <p>This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.</p>			

Secrecy Order 37 CFR 5.2:

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Inventor Information:

Inventor	1				Remove	
Legal Name						
Prefix	Given Name	Middle Name	Family Name	Suffix		
	Zvi		Tannenbaum			
Residence Information (Select One) ☒ US Residency ☐ Non US Residency ☐ Active US Military Service						
City	Palo Alto	State/Province	CA	Country of Residence	US	
Mailing Address of Inventor:						
Address 1	3513 Waverley St.					
Address 2						
City	Palo Alto	State/Province	CA			
Postal Code	94306-3537	Country	US			
Inventor	2				Remove	
Legal Name						
Prefix	Given Name	Middle Name	Family Name	Suffix		
	Dean	E.	Dauger			
Residence Information (Select One) ☒ US Residency ☐ Non US Residency ☐ Active US Military Service						
City	Huntington Beach	State/Province	CA	Country of Residence	US	
Mailing Address of Inventor:						
Address 1	P.O. Box 3074					
Address 2						
City	Huntington Beach	State/Province	CA			
Postal Code	92605	Country	US			
All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the Add button.						

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	ZTANN.002P1C6
		Application Number	
Title of Invention	CLUSTER COMPUTING		

☐ An Address is being provided for the correspondence information of this application.

Customer Number	20995		
Email Address	efiling@knobbe.com	<input type="button" value="Add Email"/>	<input type="button" value="Remove Email"/>

Application Information:

Title of the Invention	CLUSTER COMPUTING		
Attorney Docket Number	ZTANN.002P1C6	Small Entity Status Claimed	☒
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	5	Suggested Figure for Publication (if any)	2

Filing By Reference:

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)

☐ **Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and 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 international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32). Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.

Please Select One:	☒ Customer Number	☐ US Patent Practitioner	☐ Limited Recognition (37 CFR 11.9)
Customer Number	20995		

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	ZTANN.002P1C6
		Application Number	
Title of Invention	CLUSTER COMPUTING		

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, 365(c), or 386(c) or indicate National Stage entry from a PCT application. Providing benefit claim information in the Application Data Sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.

When referring to the current application, please leave the "Application Number" field blank.

Prior Application Status	Pending	Remove			
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)		
	Continuation of	16/449084	2019-06-21		
Prior Application Status	Patented	Remove			
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
16/449084	Continuation of	14/181112	2014-02-14	10333768	2019-06-25
Prior Application Status	Patented	Remove			
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
14/181112	Continuation of	13/423063	2012-03-16	8676877	2014-03-18
Prior Application Status	Patented	Remove			
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
13/423063	Continuation of	12/040519	2008-02-29	8140612	2012-03-20
Prior Application Status	Patented	Remove			
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
12/040519	Continuation in part of	11/744461	2007-05-04	8082289	2011-12-20
Prior Application Status	Expired	Remove			
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)		
11/744461	Claims benefit of provisional	60/813738	2006-06-13		
Prior Application Status	Expired	Remove			
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)		
11/744461	Claims benefit of provisional	60/850908	2006-10-11		
Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the Add button. Add					

Foreign Priority Information:

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	ZTANN.002P1C6
		Application Number	
Title of Invention	CLUSTER COMPUTING		

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55. When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX)ⁱ the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(i)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

Application Number	Country ⁱ	Filing Date (YYYY-MM-DD)	Access Code ⁱ (if applicable)	Remove

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button.

Add

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013. NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	ZTANN.002P1C6
		Application Number	
Title of Invention	CLUSTER COMPUTING		

Authorization or Opt-Out of Authorization to Permit Access:

When this Application Data Sheet is properly signed and filed with the application, applicant has provided written authority to permit a participating foreign intellectual property (IP) office access to the instant application-as-filed (see paragraph A in subsection 1 below) and the European Patent Office (EPO) access to any search results from the instant application (see paragraph B in subsection 1 below).

Should applicant choose not to provide an authorization identified in subsection 1 below, applicant **must opt-out** of the authorization by checking the corresponding box A or B or both in subsection 2 below.

NOTE: This section of the Application Data Sheet is **ONLY** reviewed and processed with the **INITIAL** filing of an application. After the initial filing of an application, an Application Data Sheet cannot be used to provide or rescind authorization for access by a foreign IP office(s). Instead, Form PTO/SB/39 or PTO/SB/69 must be used as appropriate.

1. Authorization to Permit Access by a Foreign Intellectual Property Office(s)

A. Priority Document Exchange (PDX) - Unless box A in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the State Intellectual Property Office of the People's Republic of China (SIPO), the World Intellectual Property Organization (WIPO), and any other foreign intellectual property office participating with the USPTO in a bilateral or multilateral priority document exchange agreement in which a foreign application claiming priority to the instant patent application is filed, access to: (1) the instant patent application-as-filed and its related bibliographic data, (2) any foreign or domestic application to which priority or benefit is claimed by the instant application and its related bibliographic data, and (3) the date of filing of this Authorization. See 37 CFR 1.14(h)(1).

B. Search Results from U.S. Application to EPO - Unless box B in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the EPO access to the bibliographic data and search results from the instant patent application when a European patent application claiming priority to the instant patent application is filed. See 37 CFR 1.14(h)(2).

The applicant is reminded that the EPO's Rule 141(1) EPC (European Patent Convention) requires applicants to submit a copy of search results from the instant application without delay in a European patent application that claims priority to the instant application.

2. Opt-Out of Authorizations to Permit Access by a Foreign Intellectual Property Office(s)

☐ A. Applicant **DOES NOT** authorize the USPTO to permit a participating foreign IP office access to the instant application-as-filed. If this box is checked, the USPTO will not be providing a participating foreign IP office with any documents and information identified in subsection 1A above.

☐ B. Applicant **DOES NOT** authorize the USPTO to transmit to the EPO any search results from the instant patent application. If this box is checked, the USPTO will not be providing the EPO with search results from the instant application.

NOTE: Once the application has published or is otherwise publicly available, the USPTO may provide access to the application in accordance with 37 CFR 1.14.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	ZTANN.002P1C6
		Application Number	
Title of Invention	CLUSTER COMPUTING		

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.			
Applicant	1		<input type="button" value="Remove"/>
<p>If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.</p>			
		<input type="button" value="Clear"/>	
☐ Assignee	Legal Representative under 35 U.S.C. 117		Joint Inventor
Person to whom the inventor is obligated to assign.		Person who shows sufficient proprietary interest	
If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:			
<div style="border: 1px solid black; height: 20px; width: 100%;"></div>			
Name of the Deceased or Legally Incapacitated Inventor: <div style="border: 1px solid black; width: 400px; height: 20px;"></div>			
If the Applicant is an Organization check here. ☒			
Organization Name	Advanced Cluster Systems, Inc.		
Mailing Address Information For Applicant:			
Address 1	220 Newport Center Drive, #11-202		
Address 2			
City	Newport Beach	State/Province	CA
Country	US	Postal Code	92660
Phone Number		Fax Number	
Email Address			
Additional Applicant Data may be generated within this form by selecting the Add button. <input type="button" value="Add"/>			

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

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.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	ZTANN.002P1C6
		Application Number	
Title of Invention	CLUSTER COMPUTING		

Assignee 1				
Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication. An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.				
				Remove
If the Assignee or Non-Applicant Assignee is an Organization check here.				☐
Prefix	Given Name	Middle Name	Family Name	Suffix
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Mailing Address Information For Assignee including Non-Applicant Assignee:				
Address 1		<input type="text"/>		
Address 2		<input type="text"/>		
City	<input type="text"/>	State/Province	<input type="text"/>	
Country i	<input type="text"/>	Postal Code	<input type="text"/>	
Phone Number	<input type="text"/>	Fax Number	<input type="text"/>	
Email Address	<input type="text"/>			
Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button.				Add

Signature:[Remove](#)

NOTE: This Application Data Sheet must be signed in accordance with 37 CFR 1.33(b). However, if this Application Data Sheet is submitted with the INITIAL filing of the application and either box A or B is not checked in subsection 2 of the "Authorization or Opt-Out of Authorization to Permit Access" section, then this form must also be signed in accordance with 37 CFR 1.14(c).

This Application Data Sheet **must** be signed by a patent practitioner if one or more of the applicants is a **juristic entity** (e.g., corporation or association). If the applicant is two or more joint inventors, this form must be signed by a patent practitioner, **all** joint inventors who are the applicant, or one or more joint inventor-applicants who have been given power of attorney (e.g., see USPTO Form PTO/AIA/81) on behalf of **all** joint inventor-applicants.

See 37 CFR 1.4(d) for the manner of making signatures and certifications.

Signature	/Jordan M. Cox/		Date (YYYY-MM-DD)	2021-06-21
First Name	Jordan M.	Last Name	Cox	Registration Number
				70,875
Additional Signature may be generated within this form by selecting the Add button.				Add

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.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	ZTANN.002P1C6
		Application Number	
Title of Invention	CLUSTER COMPUTING		

This collection of information is required by 37 CFR 1.76. 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 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet 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 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.

CLUSTER COMPUTING

INCORPORATION BY REFERENCE TO PRIORITY APPLICATIONS

[0001] All applications for which a foreign or domestic priority claim is identified in the Application Data Sheet as filed with the present application are incorporated by reference under 37 CFR 1.57 and made a part of this specification.

BACKGROUND

Field

[0002] The present disclosure relates to the field of cluster computing generally and to systems and methods for adding cluster computing functionality to a computer program, in particular.

Description of Related Art

[0003] Computer clusters include a group of two or more computers, microprocessors, and/or processor cores (“nodes”) that intercommunicate so that the nodes can accomplish a task as though they were a single computer. Many computer application programs are not currently designed to benefit from advantages that computer clusters can offer, even though they may be running on a group of nodes that could act as a cluster. Some computer programs can run on only a single node because, for example, they are coded to perform tasks serially or because they are designed to recognize or send instructions to only a single node.

[0004] Some application programs include an interpreter that executes instructions provided to the program by a user, a script, or another source. Such an interpreter is sometimes called a “kernel” because, for example, the interpreter can manage at least some hardware resources of a computer system and/or can manage communications between those resources and software (for example, the provided instructions, which can include a high-level programming language). Some software programs include a kernel that is designed to communicate with a single node. An example of a software package that includes a kernel that is designed to communicate with a single node is Mathematica® from Wolfram Research, Inc. (“Mathematica”). Mathematics software packages from other vendors and other types of software can also include such a kernel.

[0005] A product known as gridMathematica, also from Wolfram Research, Inc., gives Mathematica the capability to perform a form of grid computing known as “distributed computing.” Grid computers include a plurality of nodes that generally do not communicate with one another as peers. Distributed computing can be optimized for workloads that consist of many independent jobs or packets of work, which do not need to share data between the jobs during the computational process. Grid computers include at least one node known as a master node that manages a plurality of slave nodes or computational nodes. In gridMathematica, each of a plurality of kernels runs on a single node. One kernel is designated the master kernel, which handles all input, output, and scheduling of the other kernels (the computational kernels or slave kernels). Computational kernels receive commands and data only from the node running the master kernel. Each computational kernel performs its work independently of the other computational kernels and intermediate results of one job do not affect other jobs in progress on other nodes.

SUMMARY

[0006] Embodiments described herein have several features, no single one of which is solely responsible for their desirable attributes. Without limiting the scope of the invention as expressed by the claims, some of the advantageous features will now be discussed briefly.

[0007] Some embodiments described herein provide techniques for conveniently adding cluster computing functionality to a computer application. In one embodiment, a user of a software package may be able to achieve higher performance and/or higher availability from the software package by enabling the software to benefit from a plurality of nodes in a cluster. One embodiment allows a user to create applications, using a high-level language such as Mathematica, that are able to run on a computer cluster having supercomputer-like performance. One embodiment provides access to such high-performance computing through a Mathematica Front End, a command line interface, one or more high-level commands, or a programming language such as C or FORTRAN.

[0008] One embodiment adapts a software module designed to run on a single node, such as, for example, the Mathematica kernel, to support cluster computing, even when the software module is not designed to provide such support. One embodiment provides parallelization for an application program, even if no access to the program’s source code is

available. One embodiment adds and supports Message Passing Interface (“MPI”) calls directly from within a user interface, such as, for example, the Mathematica programming environment. In one embodiment, MPI calls are added to or made available from an interactive programming environment, such as the Mathematica Front End.

[0009] One embodiment provides a computer cluster including a first processor, a second processor, and a third processor. The cluster includes at least one computer-readable medium in communication at least one of the first processor, the second processor, or the third processor. A first kernel resides in the at least one computer-readable medium and is configured to translate commands into code for execution on the first processor. A first cluster node module resides in the at least one computer-readable medium. The first cluster node module is configured to send commands to the first kernel and receives commands from a user interface. A second kernel resides in the at least one computer-readable medium. The second kernel is configured to translate commands into code for execution on the second processor. A second cluster node module resides in the at least one computer-readable medium. The second cluster node module is configured to send commands to the second kernel and communicates with the first cluster node module. A third kernel resides in the at least one computer-readable medium. The third kernel is configured to translate commands into code for execution on the third processor. A third cluster node module resides in the at least one computer-readable medium. The third cluster node module is configured to send commands to the third kernel and configured to communicate with the first cluster node module and the second cluster node module. The first cluster node module comprises a data structure in which messages originating from the second and third cluster node modules are stored.

[0010] Another embodiment provides a computer cluster that includes a plurality of nodes and a software package including a user interface and a single-node kernel for interpreting program code instructions. A cluster node module is configured to communicate with the single-node kernel and other cluster node modules. The cluster node module accepts instructions from the user interface and interprets at least some of the instructions such that several cluster node modules in communication with one another act as a cluster. The cluster node module appears as a single-node kernel to the user interface. In one embodiment, the single-node kernel includes a Mathematical kernel. In some embodiments, the user interface

can include at least one of a Mathematica front end or a command line. In some embodiments, the cluster node module includes a toolkit including library calls that implement at least a portion of MPI calls. In some embodiments, the cluster node module includes a toolkit including high-level cluster computing commands. In one embodiment, the cluster system can include a plurality of Macintosh® computers (“Macs”), Windows®-based personal computers (“PCs”), and/or Unix/Linux-based workstations.

[0011] A further embodiment provides a computer cluster including a plurality of nodes. Each node is configured to access a computer-readable medium comprising program code for a user interface and program code for a single-node kernel module configured to interpret user instructions. The cluster includes a plurality of cluster node modules. Each cluster node module is configured to communicate with a single-node kernel and with one or more other cluster node modules, to accept instructions from the user interface, and to interpret at least some of the user instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster. A communications network connects the nodes. One of the plurality of cluster node modules returns a result to the user interface.

[0012] Another embodiment provides a method of evaluating a command on a computer cluster. A command from at least one of a user interface or a script is communicated to one or more cluster node modules within the computer cluster. Each of the one or more cluster node modules communicates a message based on the command to a respective kernel module associated with the cluster node module. Each of the one or more cluster node modules receives a result from the respective kernel module associated with the cluster node module. At least one of the one or more cluster node modules responds to messages from other cluster node modules.

[0013] Another embodiment provides a computing system for executing Mathematica code on multiple nodes. The computing system includes a first node module in communication with a first Mathematica kernel executing on a first node, a second node module in communication with a second Mathematica kernel executing on a second node, and a third node module in communication with a third Mathematica kernel executing on a third node. The first node module, the second node module, and the third node module are configured to communicate with one another using a peer-to-peer architecture. In some

embodiments, each of the first node module, the second node module, and third node module includes a data structure for maintaining messages originating from other node modules and a data structure for maintaining data specifying a location to which an message is expected to be received and an identifier for a node from which the message is expected to be sent.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] A general architecture that implements the various features are described with reference to the drawings. The drawings and the associated descriptions are provided to illustrate embodiments and not to limit the scope of the disclosure. Throughout the drawings, reference numbers are re-used to indicate correspondence between referenced elements.

[0015] FIG. 1 is a block diagram of one embodiment of a computer cluster.

[0016] FIG. 2 is a block diagram showing relationships between software modules running on one embodiment of a computer cluster.

[0017] FIG. 3 is a block diagram of one embodiment of a cluster node module.

[0018] FIG. 4 is a flow chart showing one embodiment of a cluster initialization process.

[0019] FIG. 5 is a flow chart showing one embodiment of the operation of a cluster node module.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0020] For purposes of illustration, some embodiments are described herein in the context of cluster computing with Mathematica software. The present disclosure is not limited to a single software program; the systems and methods can be used with other application software such as, for example, Maple®, MATLAB®, MathCAD®, Apple Shake®, Apple® Compressor, IDL®, other applications employing an interpreter or a kernel, Microsoft Excel®, Adobe After Effects®, Adobe Premiere®, Adobe Photoshop®, Apple Final Cut Pro®, and Apple iMovie®. Some figures and/or descriptions, however, relate to embodiments of computer clusters running Mathematica. The system can include a variety of uses, including but not limited to students, educators, scientists, engineers, mathematicians, researchers, and technicians. It is also recognized that in other embodiments, the systems and methods can be implemented as a single module and/or implemented in conjunction with a variety of other modules. Moreover, the specific implementations described herein are set forth in order to illustrate, and not to limit, the disclosure.

I. Overview

[0021] The cluster computing system described herein generally includes one or more computer systems connected to one another via a communications network or networks. The communications network can include one or more of a local area network (“LAN”), a wide area network (“WAN”), an intranet, the Internet, etc. In one embodiment, a computer system comprises one or more processors such as, for example, a microprocessor that can include one or more processing cores (“nodes”). The term “node” refers to a processing unit or subunit that is capable of single-threaded execution of code. The processors can be connected to one or more memory devices such as, for example, random access memory (“RAM”), and/or one or more optional storage devices such as, for example, a hard disk. Communications among the processors and such other devices may occur, for example, via one or more local buses of a computer system or via a LAN, a WAN, a storage area network (“SAN”), and/or any other communications network capable of carrying signals among computer system components. In one embodiment, one or more software modules such as kernels, run on nodes within the interconnected computer systems. In one embodiment, the kernels are designed to run on only a single node. In one embodiment, cluster node modules communicate with the kernels and with each other in order to implement cluster computing functionality.

[0022] FIG. 1 is a block diagram of one embodiment of a computer cluster 100 wherein computer systems 110, 120, 130 communicate with one another via a communications network 102. Network 102 includes one or more of a LAN, a WAN, a wireless network, an intranet, or the Internet. In one embodiment of the computer cluster, computer system 110 includes processors 112a, 112b, memory 114, and optional storage 116. Other computer systems 120, 130 can include similar devices, which generally communicate with one another within a computer system over a local communications architecture such as a local bus (not shown). A computer system can include one or more processors, and each processor can contain one or more processor cores that are capable of single-threaded execution. Processor cores are generally independent microprocessors, but more than one can be included in a single chip package. Software code designed for single-threaded execution can generally run on one processor core at a time. For example, single-threaded software code typically does not benefit from multiple processor cores in a computer system.

[0023] FIG. 2 is a block diagram showing relationships among software modules running on one embodiment of a computer cluster 100. In the embodiment shown in FIG. 2, the kernel modules 206a-e are designed for single-threaded execution. For example, if each of the processors 112a, 112b, 122a, 122b, 132 shown in FIG. 1 includes only one processor core, two kernel modules (for example, kernel modules 206a, 206b) loaded into the memory 114 of computer system 110 could exploit at least some of the processing bandwidth of the two processors 112a, 112b. Similarly, two kernel modules 206c, 206d loaded into the memory 124 of computer system 120 could exploit at least some of the processing bandwidth of the two processors 122a, 122b. Likewise, the bandwidth of processor 132 of computer system 130 could be utilized by a single instance of a cluster node module 204e loaded into the computer system's memory 134.

[0024] In the embodiment shown in FIG. 2, each of the kernel modules 206a-e is in communication with a single cluster node module 204a-e, respectively. For example, the kernel module 206a is in communication with the cluster node module 204a, the kernel module 206b is in communication with the cluster node module 206b, and so forth. In one embodiment, one instance of a cluster node module 204a-e is loaded into a computer system's memory 114, 124, 134 for every instance of a kernel module 206a-e running on the system. As shown in FIG. 2, each of the cluster node modules 204a-e is in communication with each of the other cluster node modules 204a-e. For example, one cluster node module 204a is in communication with all of the other cluster node modules 204b-e. A cluster node module 204a may communicate with another cluster node module 204b via a local bus (not shown) when, for example, both cluster node modules 204a-b execute on processors 112a, 112b within the same computer system 110. A cluster node module 204a may also communicate with another cluster node module 204c over a communications network 102 when, for example, the cluster node modules 204a, c execute on processors 112a, 122a within different computer systems 110, 120.

[0025] As shown in FIG. 2, an optional user interface module 202 such as, for example, a Mathematica front end and/or a command line interface, can connect to a cluster node module 204a. The user interface module can run on the same computer system 110 and/or the same microprocessor 112a on which the cluster node module 204a runs. The cluster node modules 204a-e provide MPI calls and/or advanced cluster functions that

implement cluster computing capability for the single-threaded kernel modules. The cluster node modules 204a-e are configured to look and behave like a kernel module 206a from the perspective of the user interface module 202. Similarly, the cluster node modules 204a-e are configured to look and behave like a user interface module 202 from the perspective of a kernel module 206a. The first cluster node module 204a is in communication with one or more other cluster node modules 204b, 204c, and so forth, each of which provides a set of MPI calls and/or advanced cluster commands. In one embodiment, MPI may be used to send messages between nodes in a computer cluster.

[0026] Communications can occur between any two or more cluster node modules (for example, between a cluster node module 204a and another cluster node module 204c) and not just between “adjacent” kernels. Each of the cluster node modules 204a-e is in communication with respective kernel modules 206a-e. Thus, the cluster node module 204a communicates with the kernel module 206a. MPI calls and advanced cluster commands are used to parallelize program code received from an optional user interface module 208 and distribute tasks among the kernel modules 206a-e. The cluster node modules 204a-e provide communications among kernel modules 206a-e while the tasks are executing. Results of evaluations performed by kernel modules 206a-e are communicated back to the first cluster node module 204a via the cluster node modules 204a-e, which communicates them to the user interface module 208.

[0027] Intercommunication among kernel modules 206a-e during thread execution, which is made possible by cluster node modules 204a-e, provides advantages for addressing various types of mathematic and scientific problems, for example. Intercommunication provided by cluster computing permits exchange of information between nodes during the course of a parallel computation. Embodiments of the present disclosure provide such intercommunication for software programs such as Mathematica, while grid computing solutions can implement communication between only one master node and many slave nodes. Grid computing does not provide for communication between slave nodes during thread execution.

[0028] For purposes of providing an overview of some embodiments, certain aspects, advantages, benefits, and novel features of the invention are described herein. It is to be understood that not necessarily all such advantages or benefits can be achieved in

accordance with any particular embodiment of the invention. Thus, for example, those skilled in the art will recognize that the invention can be embodied or carried out in a manner that achieves one advantage or group of advantages as taught herein without necessarily achieving other advantages or benefits as can be taught or suggested herein.

II. Computer Cluster 100

[0029] As shown in FIG. 1, one embodiment of a cluster system 100 includes computer systems 110, 120, 130 in communication with one another via a communications network 102. A first computer system 110 can include one or more processors 112a-b, a memory device 114, and an optional storage device 116. Similarly, a second computer system 120 can include one or more processors 122a-b, a memory device 124, and an optional storage device 126. Likewise, a third computer system 130 can include one or more processors 132, a memory device 134, and an optional storage device 136. Each of the computer systems 110, 120, 130 includes a network interface (not shown) for connecting to a communications network 102, which can include one or more of a LAN, a WAN, an intranet, a wireless network, and/or the Internet.

A. Computer System 110

[0030] In one embodiment, a first computer system 110 communicates with other computer systems 120, 130 via a network 102 as part of a computer cluster 100. In one embodiment, the computer system 110 is a personal computer, a workstation, a server, or a blade including one or more processors 112a-b, a memory device 114, an optional storage device 116, as well as a network interface module (not shown) for communications with the network 102.

1. Processors 112a-b

[0031] In one embodiment, the computer system 110 includes one or more processors 112a-b. The processors 112a-b can be one or more general purpose single-core or multi-core microprocessors such as, for example, a Pentium® processor, a Pentium® II processor, a Pentium® Pro processor, a Pentium® III processor, Pentium® 4 processor, a Core Duo® processor, a Core 2 Duo® processor, a Xeon® processor, an Itanium® processor, a Pentium® M processor, an x86 processor, an Athlon® processor, an 8051 processor, a MIPS® processor, a PowerPC® processor, an ALPHA® processor, etc. In addition, one or more of the processors 112a-b can be a special purpose microprocessor such

as a digital signal processor. The total number of processing cores (for example, processing units capable of single-threaded execution) within all processors 112a-b in the computer system 110 corresponds to the number of nodes available in the computer system 110. For example, if the processors 112a-b were each Core 2 Duo® processors having two processing cores, computer system 110 would have four nodes in all. Each node can run one or more instances of a program module, such as a single-threaded kernel module.

2. Network Interface Module

[0032] The computer system 110 can also include a network interface module (not shown) that facilitates communication between the computer system 110 and other computer systems 120, 130 via the communications network 102.

[0033] The network interface module can use a variety of network protocols. In one embodiment, the network interface module includes TCP/IP. However, it is to be appreciated that other types of network communication protocols such as, for example, Point-to-Point Protocol (“PPP”), Server Message Block (“SMB”), Serial Line Internet Protocol (“SLIP”), tunneling PPP, AppleTalk, etc., may also be used.

3. Memory 114 and Storage 116

[0034] The computer system 110 can include memory 114. Memory 114 can include, for example, processor cache memory (such as processor core-specific or cache memory shared by multiple processor cores), dynamic random-access memory (“DRAM”), static random-access memory (“SRAM”), or any other type of memory device capable of storing computer data, instructions, or program code. The computer system 110 can also include optional storage 116. Storage 116 can include, for example, one or more hard disk drives, floppy disks, flash memory, magnetic storage media, CD-ROMs, DVDs, optical storage media, or any other type of storage device capable of storing computer data, instructions, and program code.

4. Computer System 110 Information

[0035] The computer system 110 may be used in connection with various operating systems such as: Microsoft® Windows® 3.X, Windows 95®, Windows 98®, Windows NT®, Windows 2000®, Windows XP®, Windows CE®, Palm Pilot OS, OS/2, Apple® MacOS®, MacOS X®, MacOS X Server®, Disk Operating System (DOS), UNIX, Linux®, VxWorks, or IBM® OS/2®, Sun OS, Solaris OS, IRIX OS operating systems, etc.

[0036] In one embodiment, the computer system 110 is a personal computer, a laptop computer, a Blackberry® device, a portable computing device, a server, a computer workstation, a local area network of individual computers, an interactive kiosk, a personal digital assistant, an interactive wireless communications device, a handheld computer, an embedded computing device, or the like.

[0037] As can be appreciated by one of ordinary skill in the art, the computer system 110 may include various sub-routines, procedures, definitional statements, and macros. Each of the foregoing modules are typically separately compiled and linked into a single executable program. However, it is to be appreciated by one of ordinary skill in the art that the processes that are performed by selected ones of the modules may be arbitrarily redistributed to one of the other modules, combined together in a single module, made available in a shareable dynamic link library, or partitioned in any other logical way.

B. Computer System 120

[0038] In one embodiment, a second computer system 120 communicates with other computer systems 110, 130 via a network 102 as part of a computer cluster 100. In one embodiment, the computer system 120 is a personal computer, a workstation, a server, or a blade including one or more processors 122a-b, a memory device 124, an optional storage device 126, as well as a network interface module (not shown) for communications with the network 102.

1. Processors 122a-b

[0039] In one embodiment, the computer system 120 includes one or more processors 122a-b. The processors 122a-b can be one or more general purpose single-core or multi-core microprocessors such as a Pentium® processor, a Pentium® II processor, a Pentium® Pro processor, a Pentium® III processor, Pentium® 4 processor, a Core Duo® processor, a Core 2 Duo® processor, a Xeon® processor, an Itanium® processor, a Pentium® M processor, an x86 processor, an Athlon® processor, an 8051 processor, a MIPS® processor, a PowerPC® processor, an ALPHA® processor, etc. In addition, the processors 122a-b can be any special purpose microprocessors such as a digital signal processor. The total number of processing cores (for example, processing units capable of single-threaded execution) within all processors 122a-b in the computer system 120 corresponds to the number of nodes available in the computer system 120. For example, if

the processors 122a-b were each Core 2 Duo® processors having two processing cores, computer system 120 would have four nodes in all. Each node can run one or more instances of a program module, such as a single-threaded kernel module.

2. Network Interface Module

[0040] The computer system 120 can also include a network interface module (not shown) that facilitates communication between the computer system 120 and other computer systems 110, 130 via the communications network 102.

[0041] The network interface module can use a variety of network protocols. In one embodiment, the network interface module includes TCP/IP. However, it is to be appreciated that other types of network communication protocols such as, for example, Point-to-Point Protocol (“PPP”), Server Message Block (“SMB”), Serial Line Internet Protocol (“SLIP”), tunneling PPP, AppleTalk, etc., may also be used.

3. Memory 124 and Storage 126

[0042] The computer system 120 can include memory 124. Memory 124 can include, for example, processor cache memory (such as processor core-specific or cache memory shared by multiple processor cores), dynamic random-access memory (“DRAM”), static random-access memory (“SRAM”), or any other type of memory device capable of storing computer data, instructions, or program code. The computer system 120 can also include optional storage 126. Storage 126 can include, for example, one or more hard disk drives, floppy disks, flash memory, magnetic storage media, CD-ROMs, DVDs, optical storage media, or any other type of storage device capable of storing computer data, instructions, and program code.

4. Computer System 120 Information

[0043] The computer system 120 may be used in connection with various operating systems such as: Microsoft® Windows® 3.X, Windows 95®, Windows 98®, Windows NT®, Windows 2000®, Windows XP®, Windows CE®, Palm Pilot OS, OS/2, Apple® MacOS®, MacOS X®, MacOS X Server®, Disk Operating System (DOS), UNIX, Linux®, VxWorks, or IBM® OS/2®, Sun OS, Solaris OS, IRIX OS operating systems, etc.

[0044] In one embodiment, the computer system 120 is a personal computer, a laptop computer, a Blackberry® device, a portable computing device, a server, a computer workstation, a local area network of individual computers, an interactive kiosk, a personal

digital assistant, an interactive wireless communications device, a handheld computer, an embedded computing device, or the like.

[0045] As can be appreciated by one of ordinary skill in the art, the computer system 120 may include various sub-routines, procedures, definitional statements, and macros. Each of the foregoing modules are typically separately compiled and linked into a single executable program. However, it is to be appreciated by one of ordinary skill in the art that the processes that are performed by selected ones of the modules may be arbitrarily redistributed to one of the other modules, combined together in a single module, made available in a shareable dynamic link library, or partitioned in any other logical way.

C. Computer System 130

[0046] In one embodiment, a third computer system 130 communicates with other computer systems 110, 120 via a network 102 as part of a computer cluster 100. In one embodiment, the computer system 130 is a personal computer, a workstation, a server, or a blade including one or more processors 132, a memory device 134, an optional storage device 136, as well as a network interface module (not shown) for communications with the network 102.

1. Processors 112a-b

[0047] In one embodiment, the computer system 130 includes a processor 132. The processor 132 can be a general purpose single-core or multi-core microprocessors such as a Pentium® processor, a Pentium® II processor, a Pentium® Pro processor, a Pentium® III processor, Pentium® 4 processor, a Core Duo® processor, a Core 2 Duo® processor, a Xeon® processor, an Itanium® processor, a Pentium® M processor, an x86 processor, an Athlon® processor, an 8051 processor, a MIPS® processor, a PowerPC® processor, or an ALPHA® processor. In addition, the processor 132 can be any special purpose microprocessor such as a digital signal processor. The total number of processing cores (for example, processing units capable of single-threaded execution) within processor 132 in the computer system 130 corresponds to the number of nodes available in the computer system 130. For example, if the processor 132 was a Core 2 Duo® processor having two processing cores, the computer system 130 would have two nodes. Each node can run one or more instances of a program module, such as a single-threaded kernel module.

2. Network Interface Module

[0048] The computer system 130 can also include a network interface module (not shown) that facilitates communication between the computer system 130 and other computer systems 110, 120 via the communications network 102.

[0049] The network interface module can use a variety of network protocols. In one embodiment, the network interface module includes TCP/IP. However, it is to be appreciated that other types of network communication protocols such as, for example, Point-to-Point Protocol (“PPP”), Server Message Block (“SMB”), Serial Line Internet Protocol (“SLIP”), tunneling PPP, AppleTalk, etc., may also be used.

3. Memory 134 and Storage 136

[0050] The computer system 130 can include memory 134. Memory 134 can include, for example, processor cache memory (such as processor core-specific or cache memory shared by multiple processor cores), dynamic random-access memory (“DRAM”), static random-access memory (“SRAM”), or any other type of memory device capable of storing computer data, instructions, or program code. The computer system 130 can also include optional storage 136. Storage 136 can include, for example, one or more hard disk drives, floppy disks, flash memory, magnetic storage media, CD-ROMs, DVDs, optical storage media, or any other type of storage device capable of storing computer data, instructions, and program code.

4. Computer System 130 Information

[0051] The computer system 130 may be used in connection with various operating systems such as: Microsoft® Windows® 3.X, Windows 95®, Windows 98®, Windows NT®, Windows 2000®, Windows XP®, Windows CE®, Palm Pilot OS, OS/2, Apple® MacOS®, MacOS X®, MacOS X Server®, Disk Operating System (DOS), UNIX, Linux®, VxWorks, or IBM® OS/2®, Sun OS, Solaris OS, IRIX OS operating systems, etc.

[0052] In one embodiment, the computer system 130 is a personal computer, a laptop computer, a Blackberry® device, a portable computing device, a server, a computer workstation, a local area network of individual computers, an interactive kiosk, a personal digital assistant, an interactive wireless communications device, a handheld computer, an embedded computing device, or the like.

[0053] As can be appreciated by one of ordinary skill in the art, the computer system 130 may include various sub-routines, procedures, definitional statements, and macros. Each of the foregoing modules are typically separately compiled and linked into a single executable program. However, it is to be appreciated by one of ordinary skill in the art that the processes that are performed by selected ones of the modules may be arbitrarily redistributed to one of the other modules, combined together in a single module, made available in a shareable dynamic link library, or partitioned in any other logical way.

E. Communications Network 102

[0054] In one embodiment, computer systems 110, 120, 130 are in communication with one another via a communications network 102.

[0055] The communications network 102 may include one or more of any type of electronically connected group of computers including, for instance, the following networks: a virtual private network, a public Internet, a private Internet, a secure Internet, a private network, a public network, a value-added network, a wired network, a wireless network, an intranet, etc. In addition, the connectivity to the network can be, for example, a modem, Ethernet (IEEE 802.3), Gigabit Ethernet, 10-Gigabit Ethernet, Token Ring (IEEE 802.5), Fiber Distributed Datalink Interface (FDDI), Frame Relay, InfiniBand, Myrinet, Asynchronous Transfer Mode (ATM), or another interface. The communications network 102 may connect to the computer systems 110, 120, 130, for example, by use of a modem or by use of a network interface card that resides in each of the systems.

[0056] In addition, the same or different communications networks 102 may be used to facilitate communication between the first computer system 110 and the second computer system 120, between the first computer system 110 and the third computer system 130, and between the second computer system 120 and the third computer system 130.

III. Software Modules

[0057] As shown in FIG.s 1 and 2, one embodiment of a cluster system 100 includes a user interface module 202 that is able to access a plurality of kernel modules 206a-e by communicating with a first cluster node module 204a. User interface module can be stored in a memory 114, 124, 134 while running, for example, and/or can be stored in a storage device 116, 126, 136. The first cluster node module 204a is in communication with each of the other cluster node modules 204b-e. The kernel modules 206a-e can reside in the

memory of one or more computer systems on which they run. For example, the memory 114 of the first computer system 110 can store instances of kernel modules 206a-b, the memory 124 of the second computer system 120 can store instances of kernel modules 206c-d, and the memory 134 of the third computer system 130 can store an instance of kernel module 206e. The kernel modules 206a-e, which include single-threaded program code, are each associated with one of the processors 112a, 112b, 122a, 122b, 132. A cluster configuration module stored on one or more of the computer systems 110, 120, 130 or on a remote computer system, for example, can establish communication with the cluster node modules 204a-e. In one embodiment, communication between the cluster configuration module 208 and the cluster node modules 204a-e initializes the cluster node modules 204a-e to provide cluster computing support for the computer cluster 100.

A. Cluster Node Module 204

[0058] In one embodiment, the cluster node modules 204a-e provide a way for many kernel modules 206a-e such as, for example, Mathematica kernels, running on a computer cluster 100 to communicate with one another. A cluster node module 204 can include at least a portion of an application programming interface (“API”) known as the Message-Passing Interface (“MPI”), which is used in several supercomputer and cluster installations. A network of connections (for example, the arrows shown in FIG. 2) between the cluster node modules 204a-e can be implemented using a communications network 102, such as, for example, TCP/IP over Ethernet, but the connections could also occur over any other type of network or local computer bus.

[0059] A cluster node module 204 can use an application-specific toolkit or interface such as, for example, Mathematica’s MathLink, Add-Ons, or packets, to interact with an application. Normally used to connect a Mathematica kernel to a user interface known as the Mathematica Front End or other Mathematica kernels, MathLink is a bidirectional protocol to send “packets” containing messages, commands, or data between any of these entities. MathLink does not allow direct cluster computing-like simultaneous communication between Mathematica kernels during execution of a command or thread. MathLink is also not designed to perform multiple simultaneous network connections. In some embodiments, a cluster node module 204 can use an application-specific toolkit such as, for example, MathLink, for connections between entities on the same computer.

[0060] When speaking about procedures or actions on a cluster or other parallel computer, not all actions happen in sequential order, nor are they required to. For example, a parallel code, as opposed to a single-processor code of the classic “Turing machine” model, has multiple copies of the parallel code running across the cluster, typically one for each processor (or “processing element” or “core”). Such parallel code is written in such a way that different instances of the same code can communicate, collaborate, and coordinate work with each other. Multiple instances of these codes can run at the same time in parallel.

[0061] If the count of the code instances is an integer N , each instance of code execution can be labeled 0 through $N-1$. For example, a computer cluster can include N connected computers, each containing a processor. The first has cluster node module 0 connected with kernel module 0 running on processor 0. The next is cluster node module 1 and kernel module 1, on processor 1, and so forth for each of the N connected computers. Some steps of their procedure are collaborative, and some steps are independent. Even though these entities are not necessarily in lock-step, they do follow a pattern of initialization, main loop behavior (for example, cluster node module operation), and shut down.

[0062] In contrast, a parallel computing toolkit (PCT) that is provided as part of the gridMathematica software package does not provide a means for instances of the same code running on different nodes to communicate, collaborate, or coordinate work among the instances. The PCT provides commands that connect Mathematica kernels in a master-slave relationship rather than a peer-to-peer relationship as enabled by some embodiments disclosed herein. A computer cluster having peer-to-peer node architecture performs computations that can be more efficient, easier to design, and/or more reliable than similar computations performed on grid computers having master-slave node architecture. Moreover, the nature of some computations may not allow a programmer to harness multi-node processing power on systems that employ master-slave node architecture.

[0063] FIG. 3 shows one embodiment of a cluster node module 204 implementing MPI calls and advanced MPI functions. In the embodiment shown in FIG. 3, cluster node module 204 includes MPI module 302, advanced functions module 304, received message queue 306, and message receiving queue 308.

1. MPI module 302

[0064] In one embodiment, the cluster node module 204 includes an MPI module 302. The MPI module 302 can include program code for one or more of at least five kinds of MPI instructions or calls. Selected constants, instructions, and/or calls that can be implemented by the MPI module 302 are as follows:

MPI Constants

[0065] Node identifiers are used to send messages to nodes or receive messages from them. In MPI, this is accomplished by assigning each node a unique integer (\$IdProc) starting with 0. This data, with a knowledge of the total count (\$NProc), makes it possible to programmatically divide any measurable entity.

Constant	Description
\$IdProc	The identification number of the current processor
\$NProc	The number of processors in the current cluster
\$mpiCommWorld	The communicator world of the entire cluster (see MPI Communicator routines, below)
mpiCommWorld	The default communicator world for the high-level routines.

Table A

Basic MPI Calls

[0066] In one embodiment, the MPI module 302 can include basic MPI calls such as, for example, relatively low-level routines that map MPI calls that are commonly used in other languages (such as C and Fortran), so that such calls can be available directly from the Mathematica user interface 204. In some embodiments, basic MPI calls include calls that send data, equations, formulas, and/or other expressions.

[0067] Simply sending expressions from one node to another is possible with these most basic MPI calls. One node can call to send an expression while the other calls a corresponding routine to receive the sent expression. Because it is possible that the receiver has not yet called mpiRecv even if the message has left the sending node, completion of mpiSend is not a confirmation that it has been received.

Call	Description
mpiSend[expr, target, comm, tag]	Sends an expression expr to a node with the ID target in the communicator world comm, waiting until that expression has left this kernel
mpiRecv [expr, target, comm, tag]	Receives an expression into expr from a node with the ID target in the communicator world comm, waiting until the expression has arrived
mpiSendRecv[sendexpr, dest, recvexpr, source, comm]	Simultaneously sends the expression sendexpr to the node with the ID target and receives an expression into recvexpr from the node with the ID source in the communicator world comm, waiting until both operations have returned.

Table B

Asynchronous MPI Calls

[0068] Asynchronous calls make it possible for the kernel to do work while communications are proceeding simultaneously. It is also possible that another node may not be able to send or receive data yet, allowing one kernel to continue working while waiting.

Call	Description
mpiISend[expr, target, comm, tag, req]	Sends an expression expr to a processor with the ID target in the communicator world comm, returning immediately. It can be balanced with calls to mpiTest[req] until mpiTest[req] returns True.
mpiIRecv[expr, target, comm, tag, req]	Receives an expression expr from a processor with the ID target in the communicator world comm, returning immediately. It can be balanced with calls to mpiTest[req] until mpiTest[req] returns True. The expr is not safe to access until mpiTest[req] returns True.
mpiTest[req]	Completes asynchronous behavior of mpiISend and mpiIRecv
mpiWait[req]	Calls mpiTest until it returns True.
mpiWaitall[reqlist]	Calls mpiWait all on every element of reqlist
mpiWaitany[reqlist]	Calls mpiTest on each element of reqlist until one of them returns True

Table C

[0069] The mpiISend[] command can be called from within a kernel module 206 (for example, a Mathematica kernel). It creates a packet containing the Mathematica expression to be sent as payload and where the expression should be sent. The packet itself is destined only for its local cluster node module. Once received by its local cluster node module, this packet is decoded and its payload is forwarded on to the cluster node module specified in the packet.

[0070] The mpiIRecv[] command can also be called from within a kernel module 206. It creates a packet specifying where it expects to receive an expression and from which processor this expression is expected. Once received by its local cluster node module, this packet is decoded and its contents are stored in a message receiving queue (MRQ) 308 (FIG. 3).

[0071] The mpiTest[] command can be called from within a kernel module 206. It creates a packet specifying which message to test for completion, then waits for a reply expression to evaluate. Once received by the kernel module's associated cluster node module 204, this packet is decoded and its message specifier is used to search for any matching

expressions listed as completed in its received message queue (RMQ) 306. If such completed expressions are found, it is sent to its local kernel module as part of the reply in mpiTest[]. The kernel module receives this reply expression and evaluates it, which updates the kernel module's variables as needed.

[0072] Other MPI calls are built on the fundamental calls mpiISend, mpiIRecv, and mpiTest. For example, mpiBcast, a broadcast, creates instructions to send information from the broadcast processor to all the others, while the other processors perform a Recv. Similarly, high-level calls of the toolkit can be built on top of the collection of MPI calls.

Collective MPI Calls

[0073] In one embodiment, the MPI module 302 can include program code for implementing collective MPI calls (for example, calls that provide basic multi-node data movement across nodes). Collective MPI calls can include broadcasts, gathers, transpose, and other vector and matrix operations, for example. Collective calls can also provide commonly used mechanisms to send expressions between groups of nodes.

Call	Description
mpiBcast[expr, root, comm]	Performs a broadcast of expr from the root processor to all the others in the communicator world comm. An expression is expected to be supplied by the root processor, while all the others expect expr to be overwritten by the incoming expression.
mpiGather[sendexpr, recvexpr, root, comm]	All processors (including root) in the communicator comm send their expression in sendexpr to the root processor, which produces a list of these expressions, in the order according to comm, in recvexpr. On the processors that are not root, recvexpr is ignored.
mpiAllgather[sendexpr, recvexpr, comm]	All processors in the communicator comm send their expression in sendexpr, which are organized into a list of these expressions, in the order according to comm, in recvexpr on all processors in comm.
mpiScatter[sendexpr, recvexpr, root, comm]	Processor root partitions the list in sendexpr into equal parts (if possible) and places each piece in recvexpr on all the processors (including root) in the communicator world comm, according the order and size of comm.
mpiAlltoall[sendexpr, recvexpr, comm]	Each processor sends equal parts of the list in sendexpr to all other processors in the communicator world comm, which each collects from all other processors are organizes into the order according to comm.

Table D

In one embodiment, the MPI module 302 includes program code for implementing parallel sums and other reduction operations on data stored across many nodes. MPI module 302 can also include program code for implementing simple parallel input/output calls (for example, calls that allow cluster system 200 to load and store objects that are located on a plurality of nodes).

Call	Description
mpiReduce[sendexpr, recvexpr, operation, root, comm]	Performs a collective reduction operation between expressions on all processors in the communicator world comm for every element in the list in sendexpr returning the resulting list in recvexpr on the processor with the ID root.
mpiAllreduce[sendexpr, recvexpr, operation, comm]	Performs a collective reduction operation between expressions on all processors in the communicator world comm for every element in the list in sendexpr returning the resulting list in recvexpr on every processor.
mpiReduceScatter[sendexpr, recvexpr, operation, comm]	Performs a collective reduction operation between expressions on all processors in the communicator world comm for every element in the list in sendexpr, partitioning the resulting list into pieces for each processor's recvexpr.

Table E

[0074] These additional collective calls perform operations that reduce the data in parallel. The operation argument can be one of the constants below.

Constant	Description
mpiSum	Specifies that all the elements on different processors be added together in a reduction call
mpiMax	Specifies that the maximum of all the elements on different processors be chosen in a reduction call
mpiMin	Specifies that the minimum of all the elements on different processors be chosen in a reduction call

Table F

MPI Communicator Calls

[0075] In one embodiment, the MPI module 302 includes program code for implementing communicator world calls (for example, calls that would allow subsets of nodes to operate as if they were a sub-cluster). Communicators organize groups of nodes into

user-defined subsets. The communicator values returned by `mpiCommSplit[]` can be used in other MPI calls instead of `mpiCommWorld`.

Call	Description
<code>mpiCommSize[comm]</code>	Returns the number of processors within the communicator <code>comm</code>
<code>mpiCommRank[comm]</code>	Returns the rank of this processor in the communicator <code>comm</code>
<code>mpiCommDup[comm]</code>	Returns a duplicate communicator of the communicator <code>comm</code>
<code>mpiCommSplit[comm, color, key]</code>	Creates a new communicator into several disjoint subsets each identified by color. The sort order within each subset is first by key, second according to the ordering in the previous communicator. Processors not meant to participate in any new communicator indicates this by passing the constant <code>mpiUndefined</code> . The corresponding communicator is returned to each calling processor.
<code>mpiCommMap[comm]</code> <code>mpiCommMap[comm, target]</code>	Returns the mapping of the communicator <code>comm</code> to the processor indexed according to <code>\$mpiCommWorld</code> . Adding a second argument returns just the ID of the processor with the ID <code>target</code> in the communicator <code>comm</code> .
<code>mpiCommFree[comm]</code>	Frees the communicator <code>comm</code>

Table G

Other MPI Support Calls

[0076] Other calls that provide common functions include:

Call	Description
mpiWtime[]	Provides wall-dock time since some fixed time in the past. There is no guarantee that this time will read the same on all processors.
mpiWtick[]	Returns the time resolution of mpiWtime[]
MaxByElement[in]	For every nth element of each list of the list in, chooses the maximum according to Max[], and returns the result as one list. Used in the mpiMax reduction operation.
MinByElement[in]	For every nth element of each list of the list in, chooses the minimum according to Min[], and returns the result as one list. Used in the mpiMin reduction operation.

Table H

2. Advanced Functions Module 304

[0077] In one embodiment, the cluster node module 204 includes an advanced functions module 304. The advanced functions module 304 can include program code that provides a toolkit of functions inconvenient or impractical to do with MPI instructions and calls implemented by the MPI module 302. The advanced functions module 304 can rely at least partially on calls and instructions implemented by the MPI module 302 in the implementation of advanced functions. In one embodiment, the advanced functions module 304 includes a custom set of directives or functions. In an alternative embodiment, the advanced functions module 304 intercepts normal Mathematica language and converts it to one or more functions optimized for cluster execution. Such an embodiment can be easier for users familiar with Mathematica functions to use but can also complicate a program debugging process. Some functions implemented by the advanced functions module 304 can simplify operations difficult or complex to set up using parallel computing. Several examples of such functions that can be implemented by the advanced functions module 304 are shown below.

[0078] Built on the MPI calls, the calls that are described below provide commonly used communication patterns or parallel versions of Mathematica features. Unless

otherwise specified, these are executed in the communicator `mpiCommWorld`, whose default is `$mpiCommWorld`, but can be changed to a valid communicator at run time.

Common Divide-and-Conquer Parallel Evaluation

[0079] In one embodiment, the advanced functions module 304 includes functions providing for basic parallelization such as, for example, routines that would perform the same operations on many data elements or inputs, stored on many nodes. These functions can be compared to parallelized for-loops and the like. The following calls address simple parallelization of common tasks. In the call descriptions, “*expr*” refers to an expression, and “*loopspec*” refers to a set of rules that determine how the expression is evaluated. In some embodiments, the advanced functions module 304 supports at least three forms of *loopspec*, including {*var*, *count*}, where the call iterates the variable *var* from 1 to the integer *count*; {*var*, *start*, *stop*}, where the call iterates the variable *var* every integer from *start* to *stop*; and {*var*, *start*, *stop*, *increment*}, where the call iterates the variable *var* from *start* adding *increment* for each iteration until *var* exceeds *stop*, allowing *var* to be a non-integer.

Call	Description
ParallelDo[expr, loopspec]	Like Do[] except that it evaluates expr across the cluster, rather than on just one processor. The rules for how expr is evaluated is specified in loopspec, like in Do[].
ParallelFunctionToList[f, count] ParallelFunctionToList[f, count, root]	Evaluates the function f[i] from 1 to count, but across the cluster, and returns these results in a list. The third argument has it gather this list into the processor whose ID is root.
ParallelTable[expr, loopspec] ParallelTable[expr, loopspec, root]	Like Table[] except that it evaluates expr across the cluster, rather than on just one processor, returning the locally evaluated portion. The third argument has it gather this table in to the processor whose ID is root.
ParallelFunction[f, inputs, root]	Like f[inputs] except that it evaluates f on a subset of inputs scattered across the cluster from processor root and gathered back to root.
ParallelNIntegrate[expr, loopspec] ParallelNIntegrate[expr, loopspec,digits]	Like NIntegrate[] except that it evaluates a numerical integration of expr over domains partitioned into the number of processors in the cluster, then returns the sum. The third argument has each numerical integration execute with at least that many digits of precision.

Table I

Guard-Cell Management

[0080] In one embodiment, the advanced functions module 304 includes functions providing for guard-cell operations such as, for example, routines that perform nearest-neighbor communications to maintain edges of local arrays in any number of dimensions (optimized for 1-, 2-, and/or 3-D). Typically the space of a problem is divided into partitions. Often, however, neighboring edges of each partition can interact, so a “guard cell” is inserted on both edges as a substitute for the neighboring data. Thus the space a processor sees is two elements wider than the actual space for which the processor is responsible. EdgeCell helps maintain these guard cells.

Call	Description
EdgeCell[list]	Copies the second element of list to the last element of the left processor and the second-to-last element of list to the first element of the right processor while simultaneously receiving the same from its neighbors.

Table J

Matrix and Vector Manipulation

[0081] The advanced functions module 304 can also include functions providing for linear algebra operations such as, for example, parallelized versions of basic linear algebra on structures partitioned on many nodes. Such linear algebra operations can reorganize data as needed to perform matrix and vector multiplication or other operations such as determinants, trace, and the like. Matrices are partitioned and stored in processors across the cluster. These calls manipulate these matrices in common ways.

Call	Description
ParallelTranspose[matrix]	Like Transpose[] except that it transposes matrix that is in fact represented across the cluster, rather than on just one processor. It returns the portion of the transposed matrix meant for that processor.
ParallelProduct[matrix, vector]	Evaluates the product of matrix and vector, as it would on one processor, except that matrix is represented across the cluster.
ParallelDimensions[matrix]	Like Dimensions[] except that matrix is represented across the cluster, rather than on just one processor. It returns a list of each dimension.
ParallelTr[matrix]	Like Tr[] except that the matrix is represented across the cluster, rather than on just one processor. It returns the trace of this matrix.
ParallelIdentity[rank]	Like Identity[], it generates a new identity matrix, except that the matrix is represented across the cluster, rather than on just one processor. It returns the portion of the new matrix for this processor.
ParallelOuter[f, vector1, vector2]	Like Outer[f, vector1, vector2] except that the answer becomes a matrix represented across the cluster, rather than on just one processor. It returns the portion of the new matrix for this processor.
ParallelInverse[matrix]	Like Inverse[] except that the matrix is represented across the cluster, rather than on just one processor. It returns the inverse of the matrix.

Table K

Element Management

[0082] In one embodiment, the advanced functions module 304 includes element management operations. For example, a large bin of elements or particles cut up in space across the nodes may need to migrate from node to node based on rules or criteria (such as their spatial coordinate). Such operations would migrate the data from one node to another.

Besides the divide-and-conquer approach, a list of elements can also be partitioned in arbitrary ways. This is useful if elements need to be organized or sorted onto multiple processors. For example, particles of a system may drift out of the space of one processor into another, so their data would need to be redistributed periodically.

Call	Description
ElementManage[list, switch]	Selects which elements of list will be sent to which processors according to the function switch[] is evaluated on each element of list. If switch is a function, switch[] should return the ID of the processor that element should be sent. If switch is an integer, the call assumes that each elements is itself a list, whose first element is a number ranging from 0 to the passed argument. This call returns a list of the elements, from any processor, that is switch selected for this processor.
ElementManage[list]	Each element of list can be a list of two elements, the first being the ID of the processor where the element should be sent, while the second is arbitrary data to send. This call returns those list elements, from any and all processors, whose first element is this processors ID in a list. This call is used internally by the two-argument version of ElementManage[].

Table L

Fourier Transform

[0083] In one embodiment, the advanced functions module 304 includes program code for implementing large-scale parallel fast Fourier transforms (“FFTs”). For example, such functions can perform FFTs in one, two, and/or three dimensions on large amounts of data that are not stored on one node and that are instead stored on many nodes. Fourier transforms of very large arrays can be difficult to manage, not the least of which is the memory requirements. Parallelizing the Fourier transform makes it possible to make use of

all the memory available on the entire cluster, making it possible to manipulate problem sizes that no one processor could possibly do alone.

Call	Description
ParallelFourier[list]	Like Fourier[] except that list is a two- or three-dimensional list represented across the cluster, like for matrices, above. It returns the portion of the Fourier-transformed array meant for that processor.

Table M

Parallel Disk I/O

[0084] In one embodiment, the advanced functions module 304 includes parallel disk input and output calls. For example, data may need to be read in and out of the cluster in such a way that the data is distributed across the cluster evenly. The calls in the following table enable the saving data from one or more processors to storage and the retrieval data from storage.

Call	Description
ParallelPut[expr, filename] ParallelPut[expr, filename root] ParallelPut[expr, filename, root, comm]	Puts expr into the file with the name filename in order on processor 0. The third argument specifies that the file be written on the processor whose ID is root. The fourth uses the communicator world comm.
ParallelGet[filename] ParallelGet[filename, root] ParallelGet[filename, root,comm]	Reads and returns data from the file with the name filename on processor 0 partitioned into each processor on the cluster. The second argument specifies that the file is to be read on the processor whose ID is root. The third uses the communicator world comm.

Call	Description
ParallelBinaryPut[expr, type,filename] ParallelBinaryPut[expr, filename, root] ParallelBinaryPut[expr, filename, root, comm]	Puts expr into the file with the binary format type with the name filename in order on processor 0. The fourth argument specifies that the file be written on the processor whose ID is root. The fifth uses the communicator world comm.
ParallelBinaryGet[type, filename] ParallelBinaryGet[type, filename, root] ParallelBinaryGet[type, filename, root, comm]	Reads and returns data in the binary format type from the file with the name filename on processor 0 partitioned into each processor on the cluster. The third argument specifies that the file is to be read on the processor whose ID is root. The fourth uses the communicator world comm.
ParallelGetPerProcessor [expr, filename] ParallelGetPerProcessor [filename, root] ParallelGetPerProcessor [filename, root, comm]	Puts expr into the file with the name filename in order on processor 0, one line per processor. The third argument specifies that the file be written on the processor whose ID is root. The fourth uses the communicator world comm.
ParallelGetPerProcessor [filename] ParallelGetPerProcessor [filename, root] ParallelGetPerProcessor [filename, root, comm]	Reads and returns data from the file with the name filename on processor 0, one line for each processor. The second argument specifies that the file is to be read on the processor whose ID is root. The third uses the communicator world comm.

Table N

Automatic Load Balancing

[0085] Some function calls can take an inconsistent amount of processing time to complete. For example, in Mathematica, the call f[20] could in general take much longer to evaluate than f[19]. Moreover, if one or more processors within the cluster are of different

speeds (for example, if some operate at a core frequency of 2.6 GHz while other operate at less than one 1 GHz), one processor may finish a task sooner than another processor.

[0086] In some embodiments, the advanced functions module 304 includes a call that can improve the operation of the computer cluster 100 in such situations. In some embodiments, the root processor assigns a small subset of the possible calls for a function to each processor on the cluster 100. Whichever processor returns its results first is assigned a second small subset of the possible calls. The root processor will continue to assign small subsets of the possible calls as results are received until an evaluation is complete. The order in which the processors finish can vary every time an expression is evaluated, but the root processor will continue assigning additional work to processors as they become available.

[0087] In one illustrative example, there are 4 processors and f[1] to f[100] to evaluate. One could implement this by assigning f[1], f[2], f[3], f[4] to each of processors 0 (the root can assign to oneself) through 3. If the f[2] result came back first, then processor 1 would be assigned f[5]. If the f[4] result is returned next, f[6] would be assigned to processor 3. The assignments continue until all results are calculated. The results are organized for output back to the user.

[0088] In alternative embodiments, the subsets of possible calls can be assigned in any order, rather than sequentially, or in batches (for example, f[1], f[5], f[9] assigned to processor 1, etc.). Also, the subsets could be organized by delegation. For example, one processor node may not necessarily be in direct control of the other processors. Instead, a large subset could be assigned to a processor, which would in turn assign subsets of its work to other processors. The result would create a hierarchy of assignments like a vast army.

Call	Description
LoadBalanceFunctionToList[f, count]	Evaluates the function f[i] from 1 to count, but across the cluster using load-balancing techniques, and returns these results in a list. The third argument has it gather this list into the processor whose ID is root.
LoadBalanceFunctionToList[f, count, root]	

Table O

3. Received Message Queue 306

[0089] In one embodiment, the cluster node module 204 includes a received message queue 306. The received message queue 306 includes a data structure for storing

messages received from other cluster node modules. Related data pertaining to the messages received, such as whether an expression has been completed, may also be stored in the received message queue 306. The received message queue 306 may include a queue and/or another type of data structure such as, for example, a stack, a linked list, an array, a tree, etc.

4. Message Receiving Queue 308

[0090] In one embodiment, the cluster node module 204 includes a message receiving queue 308. The message receiving queue 308 includes a data structure for storing information about the location to which an expression is expected to be sent and the processor from which the expression is expected. The message receiving queue 308 may include a queue and/or another type of data structure such as, for example, a stack, a linked list, an array, a tree, etc.

B. Cluster Configuration Module 208

[0091] Cluster configuration module 208 includes program code for initializing a plurality of cluster node modules to add cluster computing support to computer systems 110, 120, 130. United States Patent No. 7,136,924, issued to Dager (the “’924 patent”), the entirety of which is hereby incorporated by reference and made a part of this specification, discloses a method and system for parallel operation and control of computer clusters. One method generally includes obtaining one or more personal computers having an operating system with discoverable network services. In some embodiments, the method includes obtaining one or more processors or processor cores on which a kernel module can run. As described in the ‘924 patent, a cluster node control and interface (CNCI) group of software applications is copied to each node. When the CNCI applications are running on a node, the cluster configuration module 208 can permit a cluster node module 204, in combination with a kernel module 206, to use the node’s processing resources to perform a parallel computation task as part of a computer cluster. The cluster configuration module 208 allows extensive automation of the cluster creation process in connection with the present disclosure.

C. User Interface Module 202

[0092] In some embodiments, computer cluster 100 includes a user interface module 202, such as, for example a Mathematica Front End or a command line interface, that includes program code for a kernel module 206 to provide graphical output, accept graphical

input, and provide other methods of user communication that a graphical user interface or a command-line interface provides. To support a user interface module 202, the behavior of a cluster node module 204a is altered in some embodiments. Rather than sending output to and accepting input from the user directly, the user interface module 202 activates the cluster node module 204a to which it is connected and specifies parameters to form a connection, such as a MathLink connection, between the cluster node module 204a and the user interface module 202. The user interface module's activation of the cluster node module 204a can initiate the execution of instructions to activate the remaining cluster node modules 204b-e on the cluster and to complete the sequence to start all kernel modules 206a-e on the cluster. Packets from the user interface module 202, normally intended for a kernel module 206a, are accepted by the cluster node module 204a as a user command. Output from the kernel module 206a associated with the cluster node module 204a can be forwarded back to the user interface module 202 for display to a user. Any of the cluster node modules 204a-e can be configured to communicate with a user interface module 202.

D. Kernel Module 206

[0093] A kernel module 206 typically includes program code for interpreting high-level code, commands, and/or instructions supplied by a user or a script into low-level code, such as, for example, machine language or assembly language. In one embodiment, each cluster node module 204a-e is connected to all other cluster node modules, while each kernel module 206a-e is allocated and connected only to one cluster node module 204. In one embodiment, there is one cluster node module-kernel module pair per processor. For example, in an embodiment of a computer cluster 100 including single-processor computer systems, each cluster node module-kernel module pair could reside on a single-processor computer. If a computer contains multiple processors or processing cores, it may contain multiple cluster node module-kernel module pairs, but the pairs can still communicate over the cluster node module's network connections.

IV. Cluster Computing Methods

[0094] In one embodiment, the computer cluster 100 includes a cluster initialization process, a method of cluster node module operation, and a cluster shut down process.

A. Cluster Initialization Process

[0095] In one embodiment, a cluster configuration module 202 initializes one or more cluster node modules 204 in order to provide cluster computing support to one or more kernel modules 206, as shown in FIG. 4.

[0096] At 402, cluster node modules are launched on the computer cluster 100. In one embodiment, the cluster node module 204a running on a first processor 112a (for example, where the user is located) accesses the other processors 112b, 122a-b, 132 on the computer cluster 100 via the cluster configuration module 208 to launch cluster node modules 204b-e onto the entire cluster. In an alternative embodiment, the cluster configuration module 208 searches for processors 112a-b, 122a-b, 132 connected to one another via communications network 102 and launches cluster node modules 204a-e on each of the processors 112a-b, 122a-b, 132.

[0097] The cluster node modules 204a-e establish communication with one another at 404. In one embodiment, each of the cluster node modules 204a-e establish direct connections using the MPI_Init command with other cluster node modules 204a-e launched on the computer cluster 100 by the cluster configuration module 208.

[0098] At 406, each cluster node module 204 attempts to connect to a kernel module 206. In one embodiment, each instance of the cluster node modules 204a-e locates, launches, and connects with a local kernel module via MathLink connections and/or similar connection tools, for example, built into the kernel module 206.

[0099] At 408, the cluster node modules 204 that are unconnected to a kernel module 206 are shut down. In one embodiment, each cluster node module 204 determines whether the local kernel module cannot be found or connected to. In one embodiment, each cluster node module 204 reports the failure to connect to a kernel module 206 to the other cluster node modules on computer cluster 100 and quits.

[0100] Processor identification numbers are assigned to the remaining cluster node modules 204 at 410. In one embodiment, each remaining cluster node module 204 calculates the total number of active processors (N) and determines identification numbers describing the remaining subset of active cluster node modules 204a-e and kernel modules 206a-e. This new set of cluster node module-kernel module pairs may be numbered 0 through N-1, for example.

[0101] Message passing support is initialized on the kernel modules 206a-e at 412. In one embodiment, each cluster node module 204 supplies initialization code (for example, Mathematica initialization code) to the local kernel module 206 to support message passing.

[0102] Finally, at 414, the cluster node modules 204a-e enter a loop to accept user entry. In one embodiment, a main loop (for example, a cluster operation loop) begins execution after the cluster node module 204a on the first processor 112a returns to user control while each of the other cluster node modules 204 waits for messages from all other cluster node modules 204a-e connected to the network 102.

[0103] The initialization process creates a structure enabling a way for the kernel modules 206a-e to send messages to one another. In some embodiments, any kernel module can send data to and receive data from any other kernel module within the cluster when initialization is complete. The cluster node module creates an illusion that a kernel module is communicating directly with the other kernel modules. The initialization process can create a relationship among kernel modules on a computer cluster 100 such as the one shown by way of example in FIG. 2.

B. Cluster Node Module Operation

[0104] In one embodiment, a cluster node module 204 implements cluster computing support for a kernel module 206 during a main loop, as shown in FIG. 5.

[0105] At 502, cluster node modules 204 wait for user commands or messages from other cluster node modules. In one embodiment, the cluster node module 204a connected to the user interface module 202 waits for a user command, while the other cluster node modules 204b-e continue checking for messages.

[0106] Once a command or message is received, the method proceeds to 504. At 504, the cluster node module 204a determines whether the message received is a quit command. If a quit command is received, the cluster node module 204a exits the loop and proceeds to a cluster node module shut down process at 505. If the message received is not a quit command, the process continues to 506.

[0107] At 506, received commands are communicated to all cluster node modules 204a-e on the computer cluster 100. In one embodiment, when a user enters a command in the user interface module 202, the cluster node module 204a connected to the user interface

module 202 submits the user command to all other cluster node modules 204b-e in the computer cluster 100. The user commands can be simple (for example, “1+1”), but can also be entire subroutines and sequences of code (such as, for example, Mathematica code), including calls to MPI from within the user interface module 202 (for example, the Mathematica Front End) to perform message passing between kernel modules 206a-e (for example, Mathematica kernels). These include the fundamental MPI calls, which are implemented using specially identified messages between a cluster node module 204 and its local kernel module 206.

[0108] The message (or user command) is communicated to the kernel modules 206a-e at 508. In one embodiment, the cluster node module 204a connected to the user interface module 202 submits the user command to the kernel module 206a to which it is connected. Each of the other cluster node modules 204b-e, after receiving the message, submits the command to the respective kernel module 206b-e to which it is connected.

[0109] At 510, a cluster node module 204 receives a result from a kernel module 206. In one embodiment, once the kernel module 206 completes its evaluation, it returns the kernel module’s output to the cluster node module 204 to which it is connected. Depending on the nature of the result from the kernel module, the cluster node module 204 can report the result to a local computer system or pass the result as a message to another cluster node module 204. For example, the cluster node module 204a running on the first processor 112a reports the output on its local computer system 110. For example, on the first processor 112a, cluster node module 204a only directly reports the output of kernel module 206a.

[0110] Messages from other cluster node modules 204 are responded to at 512. In one embodiment, each cluster node module (for example, the cluster node module 204a) checks for and responds to messages from other cluster node modules 204b-e and from the kernel module 206a repeatedly until those are exhausted. In one embodiment, output messages from the kernel module 206 are forwarded to output on the local computer system. Messages from other cluster node modules 204 are forwarded to a received message queue 306 (“RMQ”). Data from each entry in the message receiving queue 308 (“MRQ”) is matched with entries in the RMQ 306 (see, for example, description of the `mpiIRecv[]` call, above). If found, data from the MRQ 308 are combined into those in the RMQ 306 and marked as “completed” (see, for example, description of the `mpiTest[]` call, above). This

process provides the peer-to-peer behavior of the cluster node modules 204a-e. Via this mechanism, code running within multiple, simultaneously running kernel modules (for example, Mathematica kernels) can interact on a pair-wise or collective basis, performing calculations, processing, or other work on a scale larger and/or faster than one kernel could have done alone. In this manner, user-entered instructions and data specifying what work will be done via user commands can be executed more quickly and/or reliably. Once responding to messages has completed, the process returns to 502.

[0111] In some embodiments, a computer system includes software, such as an operating system, that divides memory and/or other system resources into a user space, a kernel space, an application space (for example, a portion of the user space allocated to an application program), and/or an operating system space (for example, a portion of the user space allocated to an operating system). In some embodiments, some or all of the cluster node modules 204a-e are implemented in the application space of a computer system. In further embodiments, at least some of the cluster node modules 204a-e are implemented in the operating system space of a computer system. For example, some cluster node modules in a computer cluster may operate in the application space while others operate in the operating system space.

[0112] In some embodiments, some or all of the functionality of the cluster node modules 204a-e is incorporated into or integrated with the operating system. The operating system can add cluster computing functionality to application programs, for example, by implementing at least some of the methods, modules, data structures, commands, functions, and processes discussed herein. Other suitable variations of the techniques described herein can be employed, as would be recognized by one skilled in the art.

[0113] In some embodiments, the operating system or components of the operating system can identify and launch the front end 202 and the kernels 206. The operating system or its components can connect the front end 202 and kernels 206 to one another in the same manner as a cluster node module 204 would or by a variation of one of the techniques described previously. The operating system can also be responsible for maintaining the communications network 102 that connects the modules to one another. In some embodiments, the operating system implements at least some MPI-style calls, such as, for example, collective MPI-style calls. In some embodiments, the operating system includes

an application programming interface (API) library of cluster subroutine calls that is exposed to application programs. Applications programs can use the API library to assist with launching and operating the computer cluster.

C. Cluster Shut Down Process

[0114] In one embodiment, a computer cluster 100 includes a procedure to shut down the system. If the operation process (or main loop) on the cluster node module 204a connected to the user interface module 202 detects a “Quit” or “Exit” command or otherwise receives a message from the user indicating a shut down, the sequence to shut down the cluster node modules 204a-e and the kernel modules 206a-e is activated. In one embodiment, the cluster node module 204a connected to the user interface module 202 sends a quit message to all other cluster node modules 204b-e. Each cluster node module 204 forwards the quit command to its local kernel module 206. Once its Mathematica kernel has quit, each cluster node module 204 proceeds to tear down its communication network with other cluster node modules (for example, see description of the MPI_Finalize command, above). At the conclusion of the process, each cluster node module 204 exits execution.

V. Example Operation

[0115] For purposes of illustration, sample scenarios are discussed in which the computer cluster system is used in operation. In these sample scenarios, examples of Mathematica code are given, and descriptions of how the code would be executed by a cluster system are provided.

Basic MPI

[0116] Fundamental data available to each node includes the node’s identification number and total processor count.

```
In[1]:= {$IdProc, $NProc}  
Out[1]:= {0, 2}
```

[0117] The first element should be unique for each processor, while the second is generally the same for all. Processor 0 can see what other values are using a collective (see below) communications call such as mpiGather[].

```
In[2]:= mpiGather[{$IdProc, $NProc},list,0]; list  
Out[2]:= {{0, 2}, {1, 2}}
```

Peer-to-Peer MPI

[0118] The `mpiSend` and `mpiRecv` commands make possible basic message passing, but one needs to define which processor to target. The following defines a new variable, `targetProc`, so that each pair of processors will point to each other.

```
In[3] := targetProc = If[1 == Mod[$IdProc, 2], $IdProc - 1, $IdProc + 1]
Out[3] := 1
```

[0119] In this example, the even processors target its “right” processor, while the odd ones point its “left.” For example, if the processors were lined up in a row and numbered in order, every even-numbered processor would pair with the processor following it in the line, and every odd-numbered processor would pair with the processor preceding it. Then a message can be sent:

```
In[4] := If[1 == Mod[$IdProc, 2], mpiSend[N[Pi, 22], targetProc,
      mpiCommWorld, d], mpiRecv[a, targetProc, mpiCommWorld, d]]
```

[0120] The `If []` statement causes the processors to evaluate different code: the odd processor sends 22 digits of `Pi`, while the even receives that message. Note that these MPI calls return nothing. The received message is in the variable `a`:

```
In[5] := a
Out[5] := 3.1415926535897932384626
In[6] := Clear[a]
```

[0121] The variable `a` on the odd processors would have no definition. Moreover, if `$NProc` is 8, processor 3 sent `Pi` to processor 2, processor 5 sent `Pi` to processor 4, and so on. These messages were not sent through processor 0, but they communicated on their own.

[0122] The `mpiISend` and `mpiIRecv` commands have a letter “I” to indicate asynchronous behavior, making it possible to do other work while messages are being sent and received, or if the other processor is busy. So, the above example could be done asynchronously:

```
In[7] := If[1 == Mod[$IdProc, 2], mpiISend[N[Pi, 22], targetProc,
      mpiCommWorld, d, e],
      mpiIRecv[a, targetProc, mpiCommWorld, d, e]]
```

[0123] The variable `e` has important data identifying the message, and `mpiTest[e]` can return `True` before the expressions are to be accessed. At this point, many other evaluations can be performed. Then, one can check using `mpiTest` when the data is needed:

```

In[29]:= mpiTest[e]
Out[29]:= True
In[30]:= a
Out[30]:= 3.1415926535897932384626
In[31]:= Clear[a,e]

```

[0124] The `mpiWait[e]` command could have also have been used, which does not return until `mpiTest[e]` returns True. The power of using these peer-to-peer calls is that it becomes possible to construct any message-passing pattern for any problem.

Collective MPI

[0125] In some cases, such explicit control is not required and a commonly used communication pattern is sufficient. Suppose processor 0 has an expression in *b* that all processors are meant to have. A broadcast MPI call would do:

```
In[8]:= mpiBcast[b, 0, mpiCommWorld]
```

[0126] The second argument specifies which processor is the “root” of this broadcast; all others have their *b* overwritten. To collect values from all processors, use `mpiGatherD`:

```
In[9]:= mpiGather[b, c, 0, mpiCommWorld]
```

[0127] The variable *c* of processor 0 is written with a list of all the *b* of all the processors in `mpiCommWorld`. The temporal opposite is `mpiScatter`:

```

In[10]:= Clear[b]; a = {2, 4, 5, 6}; mpiScatter[a, b, 0,
      mpiCommWorld]; b
Out[10]:= {2, 4}

```

[0128] The `mpiScatter` command cuts up the variable *a* into even pieces (when possible) and scatters them to the processors. This is the result if $\$NProc = 2$, but if $\$NProc = 4$, *b* would only have {2}.

[0129] MPI provides reduction operations to perform simple computations mixed with messaging. Consider the following:

```

In[11]:= a = {{2 + $IdProc, 45[]}, 3, {1 + $IdProc, $NProc[] []};
      mpiReduce [a,d,mpiSum, 0,mpiCommWorld ]
In[12]:= d
Out[12]:= {{5, 90}, 6, {3, 4}}

```

[0130] The `mpiSum` constant indicates that variable a of every processor will be summed. In this case, `$NProc` is 2, so those elements that were not identical result in odd sums, while those that were the same are even.

[0131] Most of these calls have default values if not all are specified. For example each of the following calls will have the equivalent effect as the above `mpiGather[]` call:

```
mpiGather[b, c, 0]
mpiGather[b, c]
c = mpiGather[b]
```

High-Level Calls

[0132] High-level calls can include convenient parallel versions of commonly used application program calls (for example, Mathematica calls). For example, `ParallelTable[]` is like `Table[]`, except that the evaluations are automatically performed in a distributed manner:

```
In[13]:= ParallelTable[i, {i, 100}, 0]
Out[13]:= {1, 2, 3, 4, 5, ..., 99, 100}
```

[0133] The third argument specifies that the answers are collated back to processor 0. This is a useful, simple way to parallelize many calls to a complex function. One could define a complicated function and evaluate it over a large range of inputs:

```
In[14]:= g[x_] := Gamma[2 + 0.5*(x-1)];
ParallelTable[g[i], {i, 100}, 0]
Out[14]:= {1, 1.32934, 2., 3.32335, 6., 11.6317, 24., 52.3428,
120., 287.885, 720}
```

[0134] `ParallelFunctionToList[]` also provides a simplified way to perform this form of parallelism.

Operations with Non-Trivial Communication

Matrix Operations

[0135] In some embodiments, one or more functions can help solve matrix calculations in parallel:

```
In[15]:= a = Table[i + 3* $IdProc + 2 j, {i, 2}, {j, 4}]
Out[15]:= {{3, 5, 7, 9}, {4, 6, 8, 10}}
In[16]:= t = ParallelTranspose[a]
Out[16]:= {{3, 4, 6, 7}, {5, 6, 8, 9}}
```

Fourier Transforms

[0136] A Fourier transform of a large array can be solved faster in parallel, or made possible on a cluster because it can all be held in memory. A two-dimensional Fourier transform of the above example follows:

```
In[17]:= f = ParallelFourier[a]
Out[17]:= {{32. + 0. I, -4. - 4. I, -4., -4. + 4. I}, {-3. -
          3. I, 0. + 0. I, 0., 0. + 0. I}}
```

Edge Cell Management

[0137] Many problems require interactions between partitions, but only on the edge elements. Maintaining these edges can be performed using `EdgeCell[]`.

```
In[18]:= a = {2, 4, 5, 6, 7}+8*$IdProc
Out[18]:= {2, 4, 5, 6, 7}
In[19]:= EdgeCell[a]; a
Out[19]:= {14, 4, 5, 6, 12}
```

Element Management

[0138] In particle-based problems, items can drift through space, sometimes outside the partition of a particular processor. This can be solved with `ElementManage[]`:

```
In[20]:= list={{0,4},{1,3},{1,4},{0,5}}; fcn[x_]:=x[[1]]
In[21]:= ElementManage[list, fcn]
Out[21]:= {{0, 4}, {0, 5}, {0, 4}, {0, 5}}
In[21]:= ElementManage[list, 2]
Out[21]:= {{0, 4}, {0, 5}, {0, 4}, {0, 5}}
```

[0139] The second argument of `ElementManage` describes how to test elements of a list. The *fcn* identifier returns which processor is the “home” of that element. Passing an integer assumes that each element is itself a list, whose first element is a number ranging from 0 to the passed argument.

[0140] While the examples above involve Mathematica software and specific embodiments of MPI calls and cluster commands, it is recognized that these embodiments are used only to illustrate features of various embodiments of the systems and methods.

VI. Additional Embodiments

[0141] Although cluster computing techniques, modules, calls, and functions are disclosed with reference to certain embodiments, the disclosure is not intended to be limited

thereby. Rather, a skilled artisan will recognize from the disclosure herein a wide number of alternatives for the exact selection of cluster calls, functions, and management systems. For example, single-node kernels can be managed using a variety of management tools and/or can be managed manually by a user, as described herein. As another example, a cluster node module can contain additional calls and procedures, including calls and procedures unrelated to cluster computing, that are not disclosed herein.

[0142] Other embodiments will be apparent to those of ordinary skill in the art from the disclosure herein. Moreover, the described embodiments have been presented by way of example only, and are not intended to limit the scope of the disclosure. Indeed, the novel methods and systems described herein can be embodied in a variety of other forms without departing from the spirit thereof. Accordingly, other combinations, omissions, substitutions and modifications will be apparent to the skilled artisan in view of the disclosure herein. Thus, the present disclosure is not intended to be limited by the disclosed embodiments, but is to be defined by reference to the appended claims. The accompanying claims and their equivalents are intended to cover forms or modifications as would fall within the scope and spirit of the inventions.

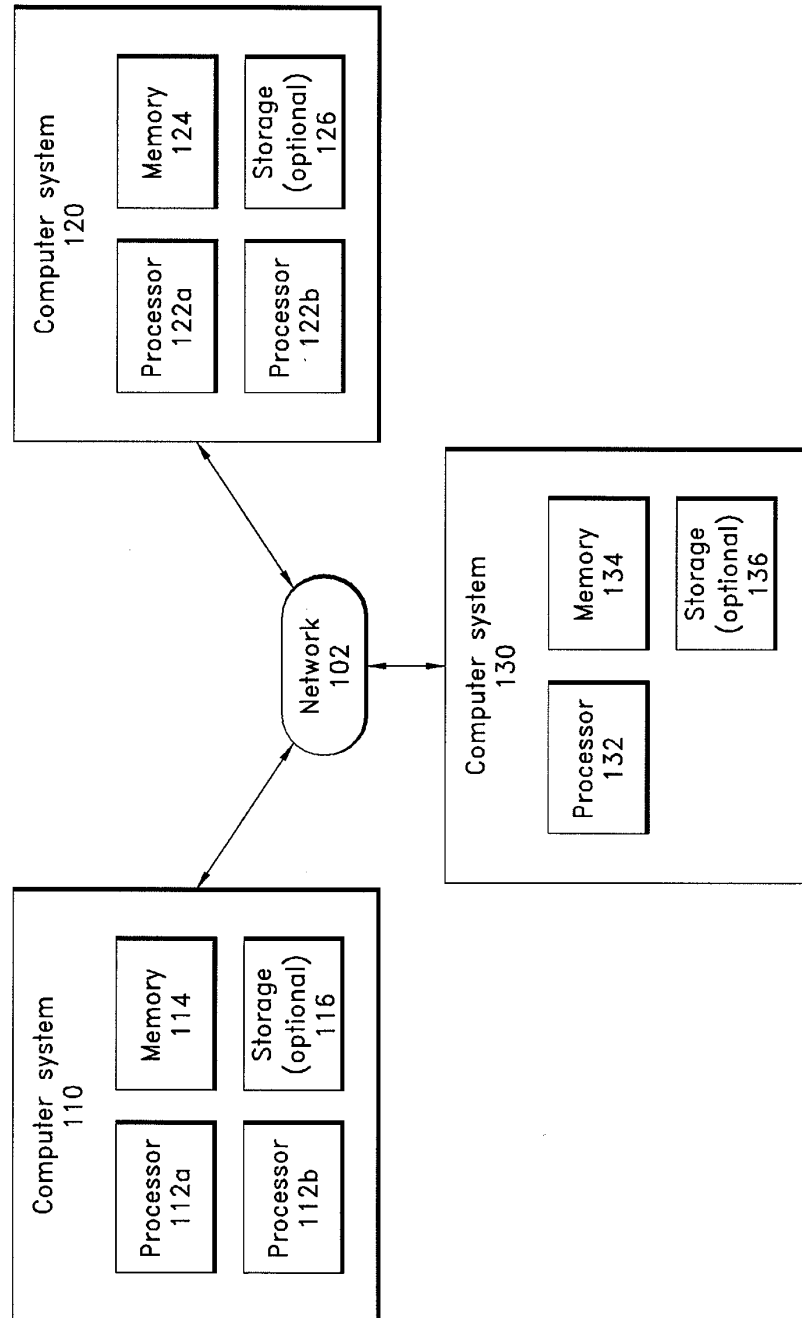
WHAT IS CLAIMED IS:

1. A computer cluster comprising:
 - a first processor;
 - a second processor;
 - a third processor;
 - at least one computer-readable medium in communication at least one of the first processor, the second processor, or the third processor;
 - a first kernel residing in the at least one computer-readable medium, said first kernel configured to translate commands into code for execution on the first processor;
 - a first cluster node module residing in the at least one computer-readable medium, said first cluster node module configured to send commands to the first kernel and receives commands from a user interface;
 - a second kernel residing in the at least one computer-readable medium, said second kernel configured to translate commands into code for execution on the second processor;
 - a second cluster node module residing in the at least one computer-readable medium, said second cluster node module configured to send commands to the second kernel and communicates with the first cluster node module;
 - a third kernel residing in the at least one computer-readable medium, said third kernel configured to translate commands into code for execution on the third processor; and
 - a third cluster node module residing in the at least one computer-readable medium, said third cluster node module configured to send commands to the third kernel and configured to communicate with the first cluster node module and the second cluster node module;
- wherein the first cluster node module comprises a data structure in which messages originating from the second and third cluster node modules are stored.

ABSTRACT

In some embodiments, a computer cluster system comprises a plurality of nodes and a software package comprising a user interface and a kernel for interpreting program code instructions. In certain embodiments, a cluster node module is configured to communicate with the kernel and other cluster node modules. The cluster node module can accept instructions from the user interface and can interpret at least some of the instructions such that several cluster node modules in communication with one another and with a kernel can act as a computer cluster.

35166502

*FIG. 1*

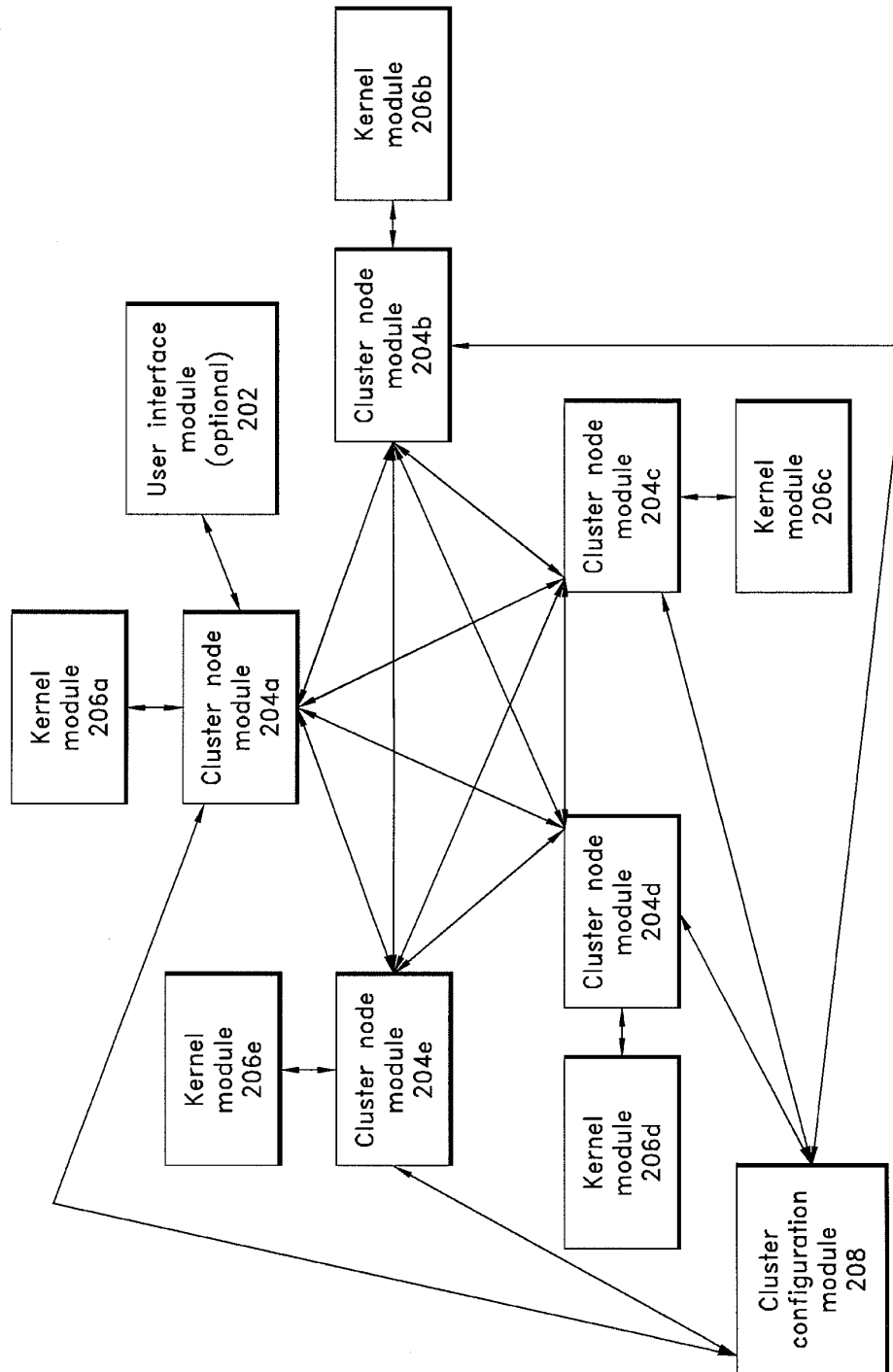


FIG. 2

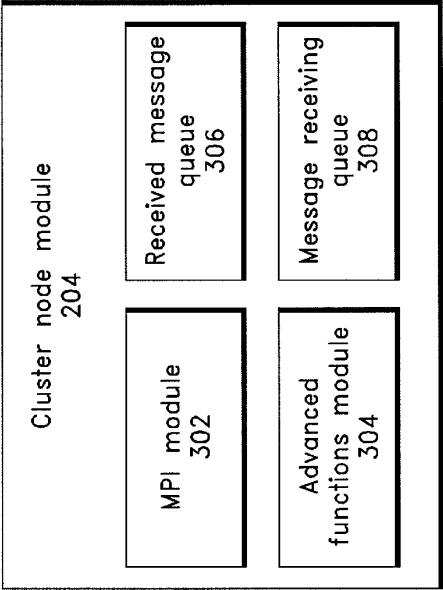
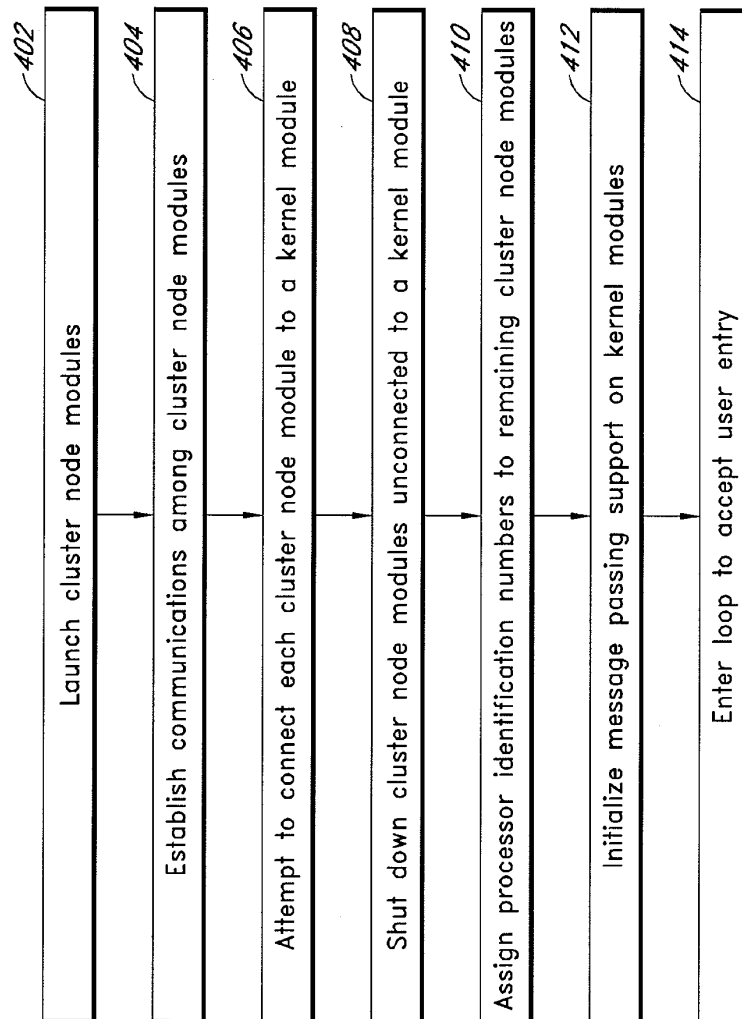


FIG. 3

*FIG. 4*

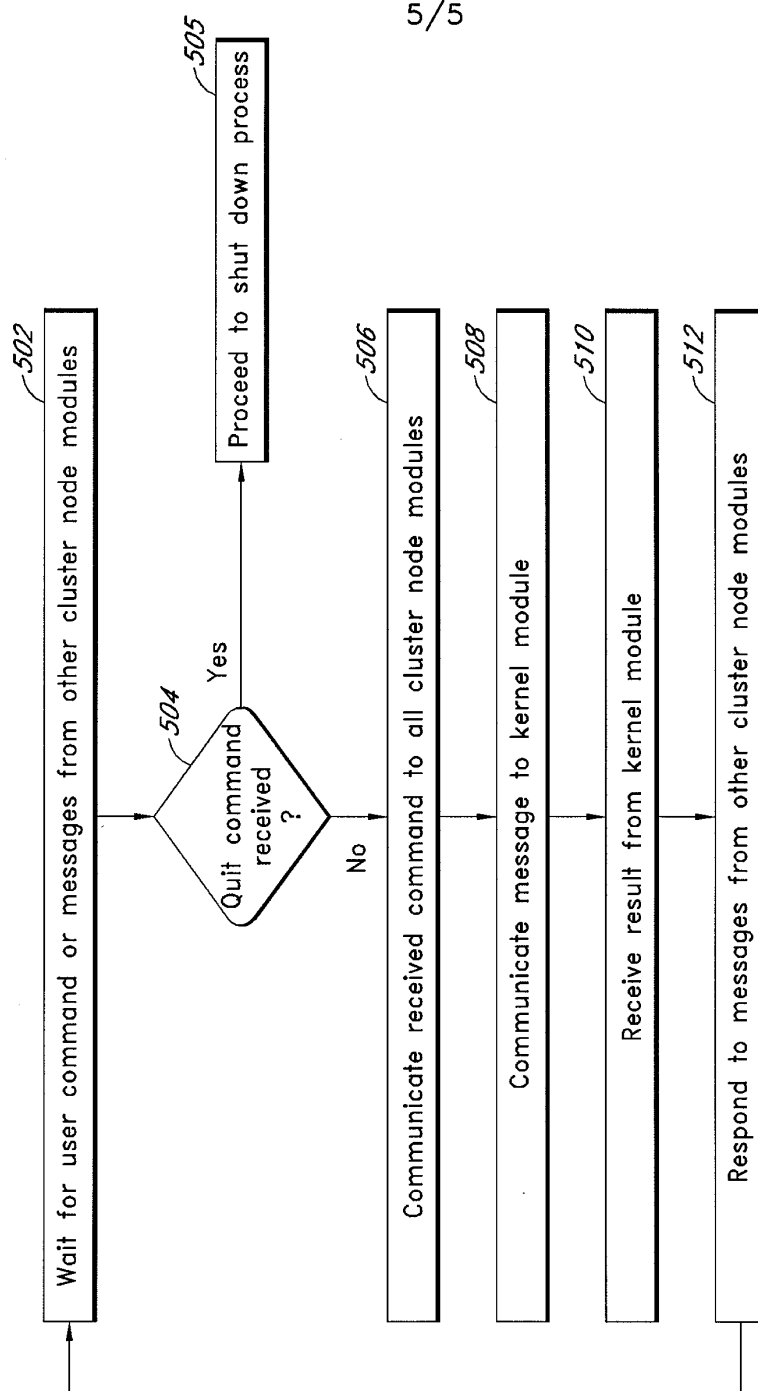


FIG. 5

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.

**DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN
APPLICATION DATA SHEET (37 CFR 1.76)****Title of
Invention****CLUSTER COMPUTING**

As the below named inventor, I hereby declare that:

This declaration
is directed to:☐

The attached application, or

☒United States application or PCT international application number 14/181,112filed on February 14, 2014

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001
by fine or imprisonment of not more than five (5) years, or both.**WARNING:**

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identify theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTORInventor: Zvi Tannenbaum

Date (Optional):

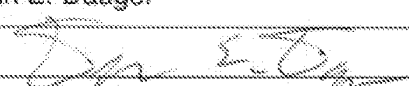
3/17/14Signature: 

Note: An application data sheet (PTO/SB/14 or equivalent), including naming the entire inventive entity, must accompany this form or must have been previously filed. Use an additional PTO/AIA/01 form for each additional inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. 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 1 minute 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 (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	CLUSTER COMPUTING
<p>As the below named inventor, I hereby declare that:</p> <p>This declaration is directed to: ☐ The attached application, or</p> <p>☒ United States application or PCT international application number <u>14/181,112</u></p> <p><u>February 14, 2014</u></p> <p>filed on _____</p> <p>The above-identified application was made or authorized to be made by me.</p> <p>I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.</p> <p>I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.</p> <p style="text-align: center;">WARNING:</p> <p>Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.</p>	
LEGAL NAME OF INVENTOR	
Inventor: <u>Dean E. Dager</u> Date (Optional): <u>3/17/14</u>	
Signature: <u></u>	
<p>Note: An application data sheet (PTO/SB/14 or equivalent), including naming the entire inventive entity, must accompany this form or must have been previously filed. Use an additional PTO/AIA/01 form for each additional inventor.</p>	

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. 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 1 minute 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.