

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
15 November 2001 (15.11.2001)

PCT

(10) International Publication Number
WO 01/86445 A1

(51) International Patent Classification⁷: **G06F 11/00**

(74) Agents: **GOTTS, Lawrence, J.** et al.; Shaw Pittman, 1650 Tysons Boulevard, McLean, VA 22102-4859 (US).

(21) International Application Number: PCT/US01/15128

(22) International Filing Date: 10 May 2001 (10.05.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/203,453 11 May 2000 (11.05.2000) US
09/657,230 7 September 2000 (07.09.2000) US

(71) Applicant: **PATMOS INTERNATIONAL CORPORATION** [US/US]; 4003 East Speedway Boulevard, Tucson, AZ 85712-4555 (US).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

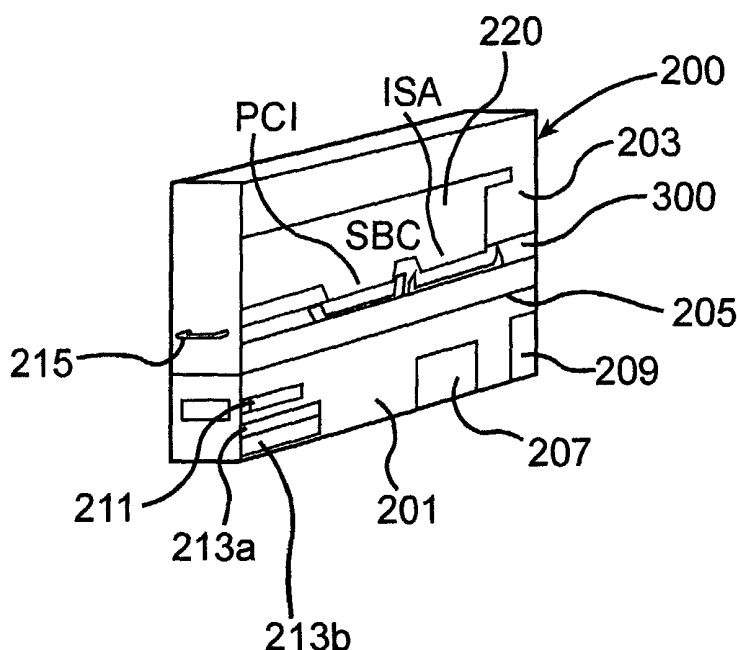
(72) Inventors: **GATZKA, James, A.**; 12827 Whisper Trace, Ocean City, MD 21842 (US). **FORSYTHE, Shane, A.**; 28197 Canterbury Drive, Salisbury, MD 21801 (US). **ANDRUSCAVAGE, Rowin, W.**; 4906 Reston Lane, Bowie, MD 20715 (US). **VAN DER SLICE, Chris, K.**; 282 Ocean Parkway, Berlin, MD 21811 (US).

Published:

— with international search report

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.

(54) Title: CONNECTIONIST TOPOLOGY COMPUTER/SERVER



(57) Abstract: A Multi-processor computer wherein each processor (220) is associated with a single independent computer (200) and each computer is connected to all other computers in a connectionist topology. Preferably each independent computer is self-contained, hot-swappable and rack mounted. A front end computer system distributes processes to each of the independent computers (200) wherein the independent computers function synchronously or asynchronously, as is required by the particular distributed process. A high-capacity secondary storage system, preferably a RAID system, is implemented to replicate the information stored on the several independent computers so that in the event one of the independent computers is swapped, the replacement independent computer can be updated from the high-capacity secondary storage system.



WO 01/86445 A1

CONNECTIONIST TOPOLOGY COMPUTER/SERVER

This application claims the benefit of Provisional Application Serial. No.
5 60/203,453 filed May 11, 2000 and U.S. Serial No. 09/657,230 filed September 7,
2000.

FIELD OF THE INVENTION

The present invention relates to the field of super computers and servers. More
particularly, the present invention is directed to a multi-processor computer wherein
10 each processor is associated with a single computer and each computer is connected to
all other computers in a connectionist topology.

BACKGROUND OF THE INVENTION

The last three decades have seen an exponential increase in the installed base of
personal computer (PC) equipment in residential, business, and educational settings of
15 every size. This is undoubtedly the result of a precipitous drop in the cost of
computing. As PCs expanded, they did so based on price reductions in their
constituent components and, primarily, in memory and storage, along with an increase
in microprocessor speed. This made PC technology accessible even to individuals and
marginally capitalized companies.

20 Three decades ago, a concurrent approach to computer technology existed.
Specifically, very large-scale main frames using an operating system generally referred
to as UNIX were implemented. These computers had a high learning curve, making it
difficult for individuals to operate. Nevertheless, UNIX dominated the mainframe and
mini mainframe marketplace, in which computing power was distributed through
25 "dumb" terminals that had no inherent processing capacity, thus reducing the
complexity of networking these devices.

The MicrosoftTM disk operating system (DOS), and the graphic user interface
introduced by Apple Computer, Mac OS, emerged almost simultaneously. These two

innovations led to the ubiquitous product called a personal computer (PC), which unlike the UNIX based mainframe, did not require connection to a network of any kind. As the PC became important in business settings in which workers needed to act in a cooperative manner, the need for an entirely new hardware/software technology emerged to provide a common work place for many PCs to operate in. Specifically, a device referred to as a router was developed, which allowed one of the PCs to handle the traffic of the many PCs involved in the workplace. Cisco is presently a leading manufacturer of such devices. This hardware technology also led to software that enhances multi-user workplaces. Oracle is one corporation that has led in this market niche.

Even during the three-decade PC age, the earlier technology of large-scale mainframes continued to flourish. The need for massive computing accelerated the evolution of mainframe technology into the so-called super computer, most recently dominated by a company called Cray.

Meanwhile, in several institutes of higher education, interest continued in the development of machines and technologies capable of performing high complexity computational tasks, which could not possibly be accomplished on a conventional PC.

Norbert Wiener, a mathematician, had coined the word "cybernetics", which, as Issac Asimov fans know, patterns the design of machines after naturally-occurring biological systems. As the PC age emerged, the word cybernetics gradually fell into disuse. However, a number of institutions of higher education continued to study the emerging needs of complexity and technologies therefor as sought by the military industrial complex and medicine. While Cray supercomputers and similar products manufactured by IBM led the way, a second paradigm emerged at the Massachusetts Institute of Technology (MIT) and was referred to as artificial intelligence (AI), in which computers were developed to mimic the human brain.

Nearly immediately, in the 1970's, AI or "cognitive science" research, was divided into two branches, the first of which was symbolic processing which is deeply embedded in a computer language called "Lisp". This first branch led to the

development of several small companies which built machines that were operated with the Lisp language and hardware that enhanced this type of program. These companies were not able to survive the PC boom which tended to overpower the opportunity for alternate technologies.

5 The other branch of AI research tended to be directed to neural technology which does not require symbols to perform computational functions. The origins of neural technology lie in the thinking of Freidrich von Hayek, an economist and social scientist who developed the "connectionist" theory regarding the neural function of the human mind, in the late 1950's. Two decades later these ideas re-emerged and a
10 machine was constructed by a team led by Daniel Hillis, then a student at MIT, under the encouragement of Marvin Minsky, a professor at that university. This device was built with the idea of assembling many processors and connecting them to create a brain-like neural network in which algorithms could perform mathematical functions in much the same way as Hayek viewed the human brain.

15 The work of Hillis and Minsky was the beginning of the development of hardware based on the notion of parallel processing. It led to the manufacture of a computer referred to as a "thinking machine" which the military industrial complex believed could operate a space-based missile defense system known in the 1970s and 1980s as "Star Wars." A change in point of view regarding the military threat of
20 nuclear disaster, however, changed the strategy of relying on a connectionist machine, and in spite of the many peaceful uses this technology could have brought to the marketplace, it has not been converted into a commercially viable product.

 An implementation of the aforementioned "Thinking Machine" is the CM-5 located at the University of Tennessee. The machine consists of 32 processing nodes
25 and a control processor. Each processing node consists of a 32 MHz SPARC processor which, together with its four vector units, is capable of performing 64-bit floating point arithmetic at a rate of 128 megaflops. Thus the maximum aggregate performance of the CM-5 is 4 gigaflops. The control processor, which is also referred to as the partition manager, is a Sun SPARCstation. The CM-5 is a distributed memory machine
30 with each processing node having 32 megabytes of primary memory available locally

giving a total of 1024 megabytes (1 gigabyte) of memory to the CM-5. Of the 32 megabytes at each processing node, only a little over 27 megabytes is actually available to the user and the rest is used by the operating system.

Each of the processing nodes as well as the control processor connect to a data network, a control network, and a diagnostics network. Of the three networks, only the data and control networks are for interprocessor communication. While the control network is used for global operations such as synchronization and broadcasting, the data network is used for communication between individual processors. The data network is capable of delivering messages to nearby nodes at rates up to 20MBps. The diagnostic network is only for system administration and is invisible to the user.

The CM-5 operates on a timesharing system which allows several users to use the system simultaneously. Each user process is given access to all the 32 processing nodes during a predetermined time slice. The control processor of the CM-5 runs the CMOST operating system which is an enhanced version of UNIX. Each processing node runs a micro-kernel of CMOST. Significantly, once a program has been loaded on the CM-5, it remains in the primary memory of the processing nodes until completion, even if it is idle during another user's time slice. Therefore, the amount of memory available at any time depends on all the programs loaded on the processing node at that time. This can affect both the scheduling and the performance of programs.

Both symbolics and connectionism fell out of favor during the PC boom, and with it, artificial intelligence (AI). The "super" typewriter/calculator/arcade art world of the PC, on the other hand, continued along until the appearance of the Internet, which has now presented levels of complexity that, while allowing it to be distributed amongst the many, limits the technological richness that could evolve from the "dumb" connections it stands for.

It is a return to the principals of symbolic computing and connectionism which leads to an entirely new approach to computing. The lack of symbolic processing in software and the poverty of connections in hardware has brought the Internet to become

a very slow and sometimes tortuous series of waiting lines for users, and may worse, come to a halt, without the appearance of incredibly fast, powerful and stable hardware, and a return to the use of high level languages.

Numerous efforts have been made to increase the ability of computers to handle
5 increasingly complex tasks and increase speed and productivity by connecting clusters
of PCs. This effort has in most cases failed to meet the needs and requirements
implied by the name "supercomputer." This approach, generally known as a
"Beowulf," while driven by highly specialized software designed to balance the load
and expand productivity, has not produced a robust machine or an effective platform
10 for connectionism which would produce a machine capable of learning and artificial
intelligence. That is, high level programming which is capable of producing symbolic
processing is not feasible in the so-called "Beowulf" approach.

SUMMARY OF THE INVENTION

The present invention provides a computer system that achieves high
15 availability and redundancy, superior performance, increased storage capability,
scalability, and superior intelligence. This is accomplished through a unique
combination of multiple processors, intelligent software, and network technologies
combined in a single self-contained, modular, scalable enclosure. Both hardware and
software function together to achieve the high performance capabilities of the present
20 invention.

Significantly, the hardware platform of the present invention is capable of
supporting the demands created by the achievement of the aforementioned goals.

In a preferred embodiment of the present invention, the hardware platform
comprises several modular components that are connected by high performance
25 network and processing software technologies. There are 5 major segments to the
architecture of the hardware platform:

(i) "nBoxen" - Raw processing component of the supercomputer of the present
invention. Each nBoxen, or node of the machine, preferably comprises primary and

secondary memory, a single processor, two hard disks (one for system information and one for data), a fibre channel RAID (Redundant Array of Independent Devices) controller, and four Fibre Channel network interfaces cards (NICs). Each nBoxen preferably houses uniform hardware. The nBoxen are designed to be highly available,
5 to be rugged and to be hot swappable. An nBoxen can be regarded as an independent computer or node.

(ii) "nMime" - This component is a collection of secondary storage devices. The nMime provides redundant storage for the data contained on the nBoxen and makes the data available in the event of an nBoxen failure. This is achieved through
10 the fibre channel RAID controller installed on each nBoxen, which permits access to every drive in the nBoxen and nMime simultaneously. The RAID controller also handles synchronization of a replacement nBoxen to a drive in the nMime. Collectively, the nMime can be regarded as a secondary storage computer.

(iii) "nSwitch" - Hyper bandwidth, low latency, switching component. The
15 nSwitch provides the fibre channel data pathways between each device. An optical switch may also be implemented when fiber optics are used.

(iv) "Limbix" - This component is the front end to the supercomputer of the present invention and to which a user interface is connected. There are preferably two Limbix, each completely capable of running the supercomputer on its own. The job of
20 the Limbix is to route, load balance, vectorize, and migrate processes to and from one or more nBoxen.

(v) "Network" - A large, multi-path, high-speed fibre channel interconnect, providing gigabit per second performance on each of several networks. The network, or communication path, can also be implemented with fiber optics.

25 The Operating System (or software, generally) that allows the hardware devices described above to function as a unit is based on a derivative of UNIX called Linux. The present invention preferably uses a distribution of RedHatTM Linux that is modified at the kernel level to transparently load-balance processes among the nBoxen. Tuning of the load balancing kernel dictates the distribution of processes according to I/O or

CPU intensity, thereby maximizing processing performance. User applications may execute in parallel by simply forking off subprocesses, which occurs automatically. Redundancy preferably is achieved through the combination of layers of software, RAID configuration, and distributed file systems.

5 Preferably, the entire system is housed in a self-contained enclosure, designed to receive each of the individual modular components. More specifically, each of the individual components is removable and replaceable, with (in some cases) redundant (dual hot swappable) power supplies and/or quick connect mountings, whereby the system, i.e., each component thereof, can be maintained (repaired) or upgraded without
10 interruption of overall service. Therefore, just as an nBoxen can be replaced within the system's enclosure, entire systems can be added or removed within the confines of a room or building, transforming that room or building into the self-contained enclosure. In other words, a modular, scalable, self-contained computing environment is achieved by the present invention.

15 The present invention leverages the modular, scalable base to provide what is known as a "connectionist" machine. The philosophy of the connectionist machine is to allow, simultaneously, both synchronous and asynchronous computation without the strict confines of time. The connectionist machine philosophy is borne of models of the human mind's operation as stated by Hayekian psychology. Also at work are principles
20 termed "Janus", to further extend the duality required to reproduce the complex operations of the human mind. Janus principles dictate that segments of the mind, or the connectionist machine, must be "two faced". In other words, the segments must be capable of action on their own (asynchronous), and as a group (synchronous).

 The connectionist machine of the present invention, however, is merely the
25 foundation upon which complex computational tasks can be created. Thus the present invention also provides intelligent software algorithms and advanced programming paradigms. Preferably, high-level symbolic programming languages are implemented, although conventional programming languages may be used, albeit without the added benefit of the other high level languages.

In a preferred embodiment, a program running on the computer of the present invention is a program that, instead of spawning one process and following strict instructions to complete a task, preferably has degrees of freedom, spawning many processes, allowing it to cast aside undefined variables not necessary for completion of the immediate task. These variables are then calculated on-demand when their definition is required to complete an immediate task. This type of processing is sometimes referred to as "suspended construction." This provides the type of complexity believed to be involved in the human thought process. Conventional programming languages do not provide the capability of this complexity.

To achieve the functionality described above, high-level languages such as Lisp and Lisp-like languages are preferably implemented. These are languages left mainly unused outside the gaming world because conventional platforms do not allow for the language's full capabilities. However, the connectionist machine platform of the present invention can take full advantage of Lisp and Lisp-like languages. Specifically, through the use of LISP and LISP-like languages, Message Passing Interface layers (MPI), Data Space for the Interpreter (DSI) protocol implementation, true distributed processing, distributed memory usage, and complexity is achieved. In the context of the present invention, complexity allows for "symbolic" processing to occur within a machine, just as "symbolic" thought processing occurs in the human mind.

The culmination of the aforementioned architecture and software is cognitive intelligence occurring through symbolic processing on a connectionist machine, held in a self-contained, modular, scalable enclosure.

In view of the foregoing, it is an object of the present invention to provide a computer system that is self-contained, modular, scalable and fully redundant.

It is a further object of the present invention to provide a computer system that has a connectionist topology.

It is still further an object of the present invention to provide a computer system that includes multiple processors, each processor operating as a node, among multiple similar nodes.

It is also an object of the present invention to provide a computer system in which individual components, including processors, are hot swappable.

It is an object of the present invention to provide a computer system that is able to distribute processes and load balance.

5 It is also an object of the present invention to provide a computer system that is a platform that is conducive to running Lisp and Lisp-like languages.

It is also an object of the present invention to provide a computer system that can function as a redundant server as well as a symbolics processor.

10 It is another object of the present invention to provide a computer environment in which both synchronous and asynchronous processes can be simultaneously implemented.

It is still further an object of the present invention to provide a computer system having health monitoring capabilities.

15 These and other objects of the present invention will become apparent upon a reading of the detailed description in conjunction with the associated drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a schematic diagram of an nBoxen in accordance with the preferred embodiment of the present invention.

20 Figure 2 is a schematic diagram of a single board computer for the nBoxen, or independent computer, in accordance with the preferred embodiment of the present invention.

Figure 3 is a top plan view of a passive backplane in accordance with the preferred embodiment of the present invention.

25 Figure 4 is a schematic diagram of a quick connect feature in accordance with the preferred embodiment of the present invention.

Figure 5 is a schematic diagram of a Limbix, or Front end computer, in accordance with a preferred embodiment of the present invention.

Figure 6 is a schematic diagram of an nMime, or secondary storage computer, in accordance with the preferred embodiment of the present invention.

5 Figure 7 is a schematic diagram of an nSwitch in accordance with the preferred embodiment of the present invention.

Figure 8 illustrates an enclosure and a preferred arrangement of various components in accordance with the preferred embodiment of the present invention.

10 Figure 9 depicts a preferable interconnectivity among the components of the present invention.

Figure 10 is a schematic diagram of the component connection topology in accordance with the preferred embodiment of the present invention.

Figure 11 is yet another schematic diagram of the connection topology in accordance with the preferred embodiment of the present invention.

15 DETAILED DESCRIPTION OF THE INVENTION

Figure 1 is a schematic diagram of an nBoxen 200 in accordance with the preferred embodiment of the present invention. Each nBoxen 200 is divided into two compartments 2014 and 203 via plate 205. The lower compartment 201 houses a power supply 207 and a cooling fan 209 and space for a floppy drive 211 and hard
20 drives 213a, 213b. It has a pull away handle 215 on an outer front face thereof. There are preferably an on/off switch, reset buttons and hard drive and power indicator lights (not shown) on the front face thereof. Additionally, there may also be provided another cooling fan adjacent the front face which provides additional cool air to a mainboard, which preferably is a single board computer (SBC) 220.

25 On SBC 220 (Figure 2) are typically an IDE hard drive controller (HDC), but more preferably a fibre channel SCSI hard drive controller (HDC) 250. Associated

with the HDC are two connectors 267, 268 that provide connection to the hard drives 213a, 213b respectively. SBC 220 further preferably includes a super socket7 CPU socket configuration to mount a central processing unit (CPU) 251. While an AMD K6-2 processor may be used for CPU 251, the Athlon™ chip is preferred, since it can
5 run at gigahertz speeds. Of course, any suitable processor can be used as will be appreciated by those skilled in the art. SBC 220 further comprises an on board AGP video chip 253 and keyboard/mouse connector(s) 255. SBC 220 still further includes - 7nsPC100 SDRAM memory 257, floppy drive controller (FDC) 259 with associated connector 261, and a serial port 263 and its associated connector 265. Also on board
10 are a health monitoring chip 269 and a hardware watchdog chip 271. An Ethernet port 273 is provided for connection to one embodiment of a network which is described later herein. The bottom of SBC 220 includes protrusions for attachment to connectors for conventional 32-bit PCI architecture and 16 bit ISA architecture.

Figure 3 is a schematic diagram of a passive backplane in accordance with the
15 preferred embodiment of the present invention. Backplane 300 preferably is located in upper compartment 203 of nBoxen 200 and is secured to the nBoxen housing or plate 205. Backplane 300 preferably includes connectors 301 and 303 for receiving the protrusions at the bottom of SBC 220. Also preferably included are four 64-bit PCI connectors 305 that accept 64-bit type network interface cards (NICs) (not shown).
20 Such cards are used to implement I/O functionality to SBC 220 and are most preferably used to implement the connectionist topology of the present invention using fibre channel connectivity.

Figure 4 is a schematic diagram of a quick connect feature for the nBoxen. The quick connect feature permits nBoxen 200 to be easily removed from the system
25 without having to disconnect individual wires. Quick connect 400 includes connectors for connecting the input/output (I/O) of SBC 220 to the wiring of the enclosure (Figure 8) which implements the connectionist network of the present invention. Specifically, quick connect 400 includes connectors for a mouse 401, keyboard 403, Ethernet 405, and has a parallel port 407, serial port 409, serial port 411, video port 413, and four
30 DB9 connectors 415a, 415b, 415c, 415d. These latter connectors are used to connect

the respective 64-bit NIC cards that are plugged into backplane 300. A multi-pin connector 417 preferably has sufficient pins to pass all of the signal paths for all of the aforesaid connectors and ports to a multi-pin receptacle 420 securely attached to enclosure 800 in which the components of the present invention are arranged.

- 5 Preferably, enclosure 800 includes the desired wiring interconnections between the various components of the system thereby allowing simple, quick and unfettered swapping of components.

Although quick connect 400 includes places for connecting keyboards etc., the way the system is preferably configured it is not necessary to connect any of the components (e.g., nBoxen) directly with a monitor, mouse or keyboard. What is
10 important is that the network connection between the components is established. Should there be a need to individually connect the monitor, keyboard, mouse to one of the computers (nBoxen), there is provided an outlet through quick connect 400 for such connection. Access to each nBoxen is achieved via Limbix 500 as is described below.

15 With respect to the hot-swappable nature of nBoxen 200, SBC 220 does not automatically run and shut itself down when it is disconnected by virtue of it being pulled from enclosure 800. Rather, when an nBoxen is removed, its supply of power is disconnected and thus it is as if the power button were pushed off, such that when another computer (e.g., another nBoxen) is returned to its location in the system, one of
20 the hard drives comes up and goes through its file system checks, etc. (assuming the nBoxen runs off an internal operating system). Thus, the component is hot swappable from the point of view of the system; replacing a component does not take the system down nor is it necessary to turn the entire machine off if a bad processor or piece of memory or network card in the node is detected.

25 Preferably, several nBoxen 200 are mounted side by side in a carriage (not shown) that holds 4 nBoxen in a standard 19" rack enclosure. (See Figure 8). Thus, each nBoxen 200 is easily replaceable via the quick-connect and a sliding rack mounting apparatus (not shown, but well known by those skilled in the art). The collection of nBoxen is the "muscle" in the computer system of the present invention
30 due to its ability to pull together as a unit and share the processing load(s) amongst

themselves and the Limbix (described later herein). Each nBoxen's two hard drives allows for numerous storage and server configurations. More specifically, there is preferably a smaller drive which holds a local Operating System (OS) and a second much larger drive for storage. With such a configuration, and with recent software
5 advancements, it is possible to mount relatively large distributed RAID partitions and/or distributed file systems across the nBoxen.

Practically speaking, the primary purpose of each nBoxen 200 is to take the processing burden off of the Limbix whenever possible. Each is relatively small (in a physical sense) and thus easily maneuverable and thus easily hot-swappable. With the
10 two hard drive configuration, the nBoxen can have local storage to use as swap space, temporary data storage and/or data caches. Additionally, an individual nBoxen may share some of its respective storage capabilities as part of a distributed network file system.

Figure 5 is a schematic diagram of a Limbix 500 in accordance with a preferred
15 embodiment of the present invention. The configuration of Limbix 500 can vary greatly depending on the type of inter-connect used to interface a user's network, and the specific use of the machine by the user. Generally, however, included in Limbix 500 are dual hot swappable power supplies 501, 503 and an ATX main board 505. ATX main board 505 preferably includes an ISA connector 507, four 32-bit PCI
20 connectors 509, four 64-bit PCI connectors 511, an AGP graphics video adapter 513, RAM 515, CPU 517 (preferably an AMD Athlon) and a fibre channel SCSI Hard Drive controller 519. Also included in Limbix 500 is a local operating system hard drive 521, indicator lights 523, a reset button 525 and a power button 527. A CD ROM drive 529 and a floppy drive 531 are preferably also included.

25 Limbix 500 serves as the user's front end of the computer system of the present invention. Where two Limbix are implemented in a single overall system, one Limbix preferably keeps track of the other through a "heartbeat function" and immediately takes over in the event of a failure. This is accomplished by changing its network address to that of the failed Limbix and assuming its current jobs. It may also act as a
30 firewall to connect the nodes of the present invention in an internal fibre channel

network to the Internet. To that effect, Limbix 500 is preferably highly-secured. Limbix 500 is responsible for monitoring, analyzing, and reporting on the health of the other nodes. For example, it may notify a system administrator through an alphanumeric paging device when the overall system or a component thereof
5 encounters a severe error or requires any type of maintenance.

Programs and applications are typically launched on Limbix 500 before being migrated off to one or several nBoxen. For workstation machines, Limbix 500 serves as the attachment point for peripherals and high-end video cards. Within the system, the Limbix performs a variety of tasks. These tasks include, but are not limited to,
10 system health, management, configuration, routing, firewalling, fail-over recovery, and task assignment for distributed file systems, memory, and programs. In a preferred implementation, Limbix 800 acts as an interface between the system of the present invention and the Internet whereby the present invention operates as a highly stable Internet server.

15 Figure 6 is a schematic diagram of nMime 600 in accordance with the preferred embodiment of the present invention. nMime 600 preferably comprises rack mountable RAID case 601 that holds eight or more hot swappable fibre channel hard drives 603. This component preferably further comprises an alpha numeric menu display 605 for RAID configuration, and a fibre channel controller card 607. Fibre channel controller
20 card 607 is, in itself, a small SBC capable of managing the RAID and its traffic, but nMime 600 also is a computer in its own right, with the ability to serve configuration information to new or replaced nodes (nBoxen) when they are added to the system.

nMime 600 also preferably comprises four PCI fibre channel NIC cards (now shown), and runs an AMD Athlon with -7nsPC100 SDRAM on a full-sized ATX
25 mainboard 609, with its own local Operating System running on a fibre channel SCSI Hard Drive 611. Dual hot-swappable power supplies 615 are also preferably provided to increase redundancy.

Through known hardware RAID techniques, nMime 600 adds highly available redundancy and access, as well as improved read response by mimicking the data stored

in each nBoxen 200. Thus, nMime 600 may provide a mirror of the hard disks that are in the respective nBoxen so that when an nBoxen is replaced, the replacement nBoxen will automatically be updated from nMime 600.

5 In the above-described configuration, nMime 600 acts as a shared resource and can be used for general storage like home directories and configuration files. nMime 600 is unique because in one sense it is a traditional multiple drive RAID. However, in the present invention, each nBoxen also is capable of sharing its drives with all others, creating a true distributed file server that eliminates the bottlenecks otherwise encountered if all the servers were attempting to access the nMime simultaneously.

10 Figure 7 is a schematic diagram of nSwitch 700 in accordance with the preferred embodiment of the present invention. Any switched fabric capable of handling internet protocol (IP) over fibre channel connections is suitable and within the scope of the present invention. The presently preferred embodiment implements the Gadzoox Capellix 3000 fibre channel switch. The several nSwitch 700 in the present
15 invention can be thought of as the communications backbone of the computer system by providing a high speed interconnect between the modular components of the system as shown in the Figure. Preferably there are at least two independent fiber channel nSwitches (nSwitch 700) in a four node machine (Figure 10) and at least three independent fibre channel switches (nSwitch 700) in a 16 node machine (Figure 11).

20 nSwitch 700 preferably is a fibre channel fabric switch that contains removable modular blades that have eight DB9 fibre channel connections each. Each nSwitch preferably holds up to four of these blades for a total of 32 connections per nSwitch. Each nSwitch also preferably has both DB9 serial and 10/100 Ethernet ports for remote management of the fabric layers of network & storage switching. Of course, optical
25 switching may be employed where the network is a Fiber Optic network.

Figure 8 depicts an enclosure 800 for a 16 node machine in accordance with the present invention. Enclosure 800 preferably includes tandem standard 19-inch racks 810a, 810b and doors 812, 814. The several components of the present invention are mounted on carriages (not shown) that enable simple and quick swapping of the

components. Preferably, multiple nBoxen 200 are mounted in a single rack, four abreast. Thus, for a 16-node machine, four rows of four nBoxen are mounted as shown in Figure 8. Preferably, in the other rack are mounted the nSwitch, two Limbix 500 and two nMime 600. All of these components are connected in a connectionist network topology described below. The actual hard wire connections are provided on the back sides of racks 810a, 810b, and this wiring, once established, need not be modified. Desired connections among the components is accomplished via one or more Switch 700.

Figure 9 is a schematic diagram of a component connection topology in accordance with the preferred embodiment of the present invention. As shown, 16 separate nBoxen 200 each have four connections to other components, either other nBoxen or one of the two Limbix 500 or nMime 600. Though not shown, each Limbix 500 and nMime 600 also includes four connections; for clarity only three connections are shown in the figure. In a preferred embodiment, opposite Limbix/nMime pairs can be connected to each other to complete the 4-connection topology for each component.

To achieve the connectivity described above, the present invention preferably implements a multi-path, high-speed fibre channel interconnect or communication pathway 910, providing gigabit per second performance on each network. As described previously, there preferably are four PCI Fibre Channel NIC cards or HBAs (Host Bus Adapters) in each Limbix, nBoxen, and nMime. Each of these network cards is in turn connected to a separate nSwitch to form several separate networks. In a 16 node machine, the network is built with enough nSwitches to provide no less than three separate networks for each nBoxen, nMime, and Limbix. Additional nSwitches can be used as interconnects between, for example, two (or more) enclosures 800 to provide modular scaling whereby the Limbix of one enclosure is/are seen as nodes by a Limbix of other enclosures. This scalability facilitates the construction of super computers consisting of thousands of nodes for un-paralleled computation, rendering, data-mining, and modeling for a wide variety of scientific, business, or government applications. This type of networking also facilitates the development of free-form genetic algorithms that may be active within the network.

Operation of the network is controlled through a combination of routing control within the Limbix, and remote manipulation of the network fabric within the nSwitches. Specifically, a network IP number, such as "12.4.220.2" is one 32 bit number. This "dotted quad" notation is what is typically considered to be an "IP
 5 number" and is broken into 4 8-bit numbers, displayed in decimal form and separated by dots. Since 8 bits can represent 0-255, the maximum address range is 0.0.0.0 - 255.255.255.255. Another way to display the information is to represent each 8 bit number by hexadecimal numbers which gives a range of 0.0.0.0 - FF.FF.FF.FF

In the following description of an IP scheme for routing in accordance with the
 10 present invention, numbers (1) -(7) indicate positions for the standard dotted quad IP address.

10.	0	0	.	0	0	.	0	0
(1)	(2)	(3)		(4)	(5)		(6)	(7)

15

In the scheme, each hexadecimal number indicates unique information:

(1) 10.00.00.00 is the Class A network reserved for internal private networks.

(2) designates which network (interface card) is being used. In a presently preferred embodiment, these number range from 1 to 4, but, of course, it is quite
 20 possible to add as many network cards as desired.

(6)(7) Is used internally for a single 'janus faced' entity (a base 16 node machine) to distinguish between nBoxen, Limbix, and establish failover, as follows:

01 = Limbix1

02 = Limbix2

25

10 = 'Virtual IP' of Limbix. Either Limbix1 or Limbix2 is delegated to this IP number on the basis of failover. This is used by the nBoxen to generically respond to the Limbix.

5 11 = nBoxen1

 12 = nBoxen2

 13 = nBoxen3

 .

 .

10 1E = nBoxen15

 1F = nBoxen16

Thus, for a single 16 node machine, the unique addressing is as follows:

	Limbix1:	Limbix2:	nBoxen1	nBoxen2
15	10.10.00.01	10.10.00.02	10.10.00.11	10.10.00.12	
	10.20.00.01	10.20.00.02	10.20.00.11	10.20.00.12	
	10.30.00.01	10.30.00.02	10.30.00.11	10.30.00.12	
	10.40.00.01	10.40.00.02	10.40.00.11	10.40.00.12	

20 The three common zeros in the above (3)(4)(5) positions are used for growth as p16 node machines are linked together into a larger hierarchy.

The potential growth of multiple 16 node machines then flows as follows:

16 nBoxen connect together to make 1 P16;

16 P16 connect together to make a P256 node machine;

16 P256 connect together to make a P4096 node machine;

5 16 P4096 connect together to make a P65356 node machine;

Thus,

(3) indicates which 4096 node machine the unit is a member of;

(4) indicates which 256 node machine the unit is a member of; and

(5) indicates which 16 node machine the unit is a member of

10 The single unit (16 node computer with two Limbix which control access to the outer world) can be considered to 'janus faced'. Internally it is aware of smaller subdivisions, *e.g.*, the nboxen, but presents a single unified whole to any outside connections.

15 If a unit is connected to other P16s to form a larger unit, it will be directly informed of, or through re-entrant negotiations with the other P16s it is joining, which 'node' it has now become, i.e., the first P16, or the second, or the third, etc.

Once 'realizing' it is now part of a larger whole it recalculates its IP address and all sub-units correspondingly adjust.

20 Below is the IP list for 3 P16 which are part of a 256 node machine, i.e., the Janus IP of each IP as it acts as an nBoxen to the larger collection of 16 machines.

```

a)10.10.10.11 10.10.10.12 10.10.10.13
-----
b)10.10.11.01 10.10.12.01 10.10.13.01 .....
c)10.10.11.02 10.10.12.02 10.10.13.01

```

```

-----
10.10.11.10
10.10.11.11
10.10.11.12
5  10.10.11.13
...

```

Looking at line (a) above , positions(3)(4)(5) = 0.10. , and positions (6)(7) indicate the units are 'nBoxen'.

10 To understand this, the Janus idea of “relativistic perspective” is involved wherein each P16 is considered a single unit at this level, and its IP address determines that it is part of a 256 node machine (ID=1). Position 5 (which 16 node machine you are a part of), does not have relevance in this setting and its meaning is subsumed by positions (6)(7) which determine the unique identity of this unit.

15 On the opposing side of the Janus boundary, the Limbix calculates unique IP numbers based on the unique IP given to it as an nBoxen. It copies the network part of its nBoxen IP (10.10.10) and fills position (5) with the designated nBoxen number to create a new unique network number. (10.10.11 , 10.10.12 , 10.10.13).

20 Using that base network number, the sub-units calculate their own IP address by tacking on their own unique unit designator (6)(7).

Based on the foregoing, if 16 P256 node machines were connected, the growth would look like

```

(2 P256 node machine , part of a 4096 node machine)
10.11.00.11          10.11.00.12          nBoxen to a P4096
25 -----          -----
10.11.10.10          10.11.20.10          Limbix to a P256
.                    .
.                    .
10.11.10.11          10.11.20.11          nBoxen to a P256

```

-----	-----	
10.11.11.10	10.11.21.10	Limbix to a P16
.	.	
.	.	
5		
10.11.11.11	10.11.21.11	nBoxen to a P16

The scheme described above provides both network redundancy (high availability), increased performance through multiple high bandwidth networks, and
 10 extra bandwidth for status polling of system components.

The present invention preferably also incorporates DSI (Data Space for the Interpreter) protocols allowing distributed memory and programming to occur within the network.

As shown in Figure 9, the basic network or system topology is a tetrahedron.
 15 Accordingly, a "base" system preferably comprises four nodes. A single tetrahedron can be considered the basic building block of the system's topology. For sixteen nodes, four of the basic nodes are strung together to make a larger cluster that is self-similar to the basic cluster such that each sub-tetrahedron still has four interconnects whereby each of the clusters of tetrahedrons can communicate to every other one.

20 The connectivity among four nBoxen 200, two Limbix 500, two nMime 600 and two nSwitch 700 is shown in more detail in Figure 10. This connectivity is easily scalable to achieve a sixteen node machine, like that shown in Figure 11. (It is noted that Figure 11 depicts only three connections per component for purposes of clarity.) Machines with an even greater number of nodes is possible due to the scalability of the
 25 present invention as described above.

The logic that is applied to a four node machine can also be expanded. Specifically, to create a bigger module, the construct is expanded such that even though each node is actually four nodes, the logic remains the same. In accordance with the present invention, each node "talks" to the next one, whereby the nodes branch out like

a growing force. However, no matter what configuration is ultimately arrived at, there are still preferably four ways in and out of each cluster. Significantly, while, these four "ways" could be considered inputs and outputs, the topology is more akin to a four dimensional neural network which provides any number of different pathways to a
5 certain destination.

In certain configurations, e.g., a four-node machine, it is possible that there might be open connections. However, such connections are preferably looped back to themselves, thereby providing additional redundancy, i.e., another loop is provided inside the main unit. Accordingly, yet another internal route is provided. Thus, if there
10 are no intended external connections, additional internal connections can be realized.

The present invention implements hardware technology and topology that emulates views of how the mind works - the connectionist view. Specifically, each node of the machine ostensibly functions like a synapse, or a data pathway.

Further, nBoxen 200 can be thought of as transforms. That is, one can view the
15 nBoxen as black box technology; something (e.g., data) comes in, something (e.g., transformed data) comes out. The programmer provides the transforms. That is, nBoxen 200, as far as Limbix 500 is concerned, is just a transform. By thinking of the nBoxen as transforms, it is possible to model/exhibit characteristics of neural connection. A Limbix 500 physically has 4 Fibre Channel NICs that are connected to 4
20 nBoxen via 16 independent HBAs. Procedurally, though, the Limbix 500 is only sending out one output, and expecting one input. The internal underlying structure of the hardware provides a neural loop that acts like an intelligent synapse.

In a preferred implementation, the network cards are not only used to get information to go from one machine to another. Rather, Limbix 500 solves a
25 computational task by sending information "down a pipe" and waiting for information to come back.

OPERATING SYSTEM AND SOFTWARE

The hardware of present invention described thus far is put into operation via a combination of an overall operating system and application software. Preferably, Limbix 500 is the front end of the overall system. The front end is used to distribute
5 the load on the individual nodes and preferably also acts as a firewall. As mentioned previously, Limbix 500 preferably runs a version of Linux which inherently includes routing capabilities. Thus, a router is not a necessary component of the present invention. Limbix 500 preferably also runs PVM (parallelized virtual machine). By using Linux, proxying is also possible with the present invention.

10 Limbix 500 further preferably controls, for example, the operating system used by each nBoxen by controlling the boot image the nBoxen sees on startup. Finally, Limbix 500 connects the machine of the present invention with the outside world through a modem, ethernet, other kind of LAN, or any other communications system. That is, the Limbix can be configured with one or more adapter cards in its several
15 available motherboard slots whereby the overall system can be connected to any number of different external communications systems.

In accordance with the present invention, Limbix preferably runs a Linux kernel patch called Mosix (available from <http://www.mosix.org>) which operates to migrate processes from one machine, i.e., node, to another, transparently. In other words, when
20 Mosix is running, a map of the nodes is accessed whereby it is possible to determine which nodes are "up" and available, how much memory each currently has, what the nodes' processor feed is, and what the respective loads are, whereby if a CPU-intensive process is running on one node, Mosix can scatter other processes off of the busy node on to other nodes, and utilize the other processors to maintain overall functionality.

25 A functional layering scheme of the environment of the machine in accordance with the present invention is as follows:

Daisy (High level language that greatly benefits/simplifies development of AI)

Daisy Microkernel

DSI (Data Space for the Interpreter) virtual Machine

Host Operating System

Hardware

5 The standard programming language for implementing applications on the machine of the present invention is C/C++. On the other hand, such applications have inherent "complexity barriers", namely they require a large degree of explicit resource management and scheduling (allocating memory, scheduling threads, explicit parallelizing, etc.).

10 Therefore, the preferred programming language for the present invention is Lisp, a language that is more abstracted and thus well suited for building complex algorithms. Daisy is a language derived from LISP that adds some additional functionality such as suspended construction. More specifically, resource management issues are handled transparently by the Daisy microkernel and DSI virtual machine. The underlying hardware of the present invention provides an ideal (i.e., robust, highly
15 interconnected) environment for the virtual machine to utilize.

When a programmer programs using Daisy, and wants to, for example, send a packet from point A to point B, he does not have to know that there are, e.g., 4 network cards in each node. Daisy, the Daisy Microkernel and the virtual machine are organized such that the programmer need only execute a 'send packet' function. Such
20 programming techniques are well known to those skilled in the art. The difference with respect to the present invention, however, is that the underlying hardware architecture permits higher levels of computing. This is even more evident with the implementation of Mosix which operates to intercept messages/packets and to reroute them to other nodes. Preferably, Mosix is built into the kernel of each nBoxen and Limbix.

25 Via Mosix, the individual nodes can operate transparently; each can operate independently or work with the others as a group. This operation possibility follows the Janus principle, i.e., reticulation and arborization.

Moreover, a symbolic language such as LISP frees a programmer from resource constraints. In C programming, for example, pointers are required and memory must be allocated. However, a Lisp programmer never allocates memory.

Another type of load balancing in the present invention is preferably
5 implemented with the use of Eddieware (available from <http://www.eddieware.org>), a software application particularly designed to load balance a machine when operated as a world wide web server over the Internet. Eddieware uses a handful of front-end nodes as traffic controllers, which redirect incoming requests to a cluster of backend nodes in round-robin fashion, thereby distributing server loads. The Limbix is/are well
10 suited for performing the task of the front-end nodes, delegating web traffic to the farm of nBoxen under their control. The Eddieware software manages details such as failover so that the client will not notice if the node they had been accessing data from suddenly failed. Eddieware ensures that another node resumes the connection transparently and finishes the clients' requested tasks.

15 In conventional computer systems, firewalls are achieved via hardware, and often, by stand alone, expensive boxes. Many companies that run NTTM networks run the network from one central router (which might cost thousands of dollars) that provides packet filtering, network address translation, and routing. However, Linux inherently accomplishes all of this. With a utility called "IP chains", for example,
20 Linux can set up and apply packet filtering rules, perform IP masquerading, and route packets from one subnet to another. While there are some proxying abilities included in the Linux kernel, many others are readily available, as is well known by those skilled in the art. Accordingly, in accordance with the present invention, packet filtering, IP masquerading, and routing can be easily achieved through software.

25 The present invention is also particularly amenable to PVM, a well-known message caching interface layer which facilitates parallel programming. PVM allows one to write a program in parallel without knowing how many nodes are available. Essentially, it is a software layer that knows how many processors or computers are available within its cluster to effectively distribute parallelized program processes. In

view of the multiple node system of the present invention, the use of PVM is particularly desirable.

MONITORING

The present invention also provides monitoring and early warning capabilities. Specifically, in a preferred embodiment of the present invention, the main board of
5 each computer preferably includes onboard "health monitoring" (chip 269) for, e.g., CPU speed, CPU temperature, and system voltage. Also, there is preferably provided a hardware watchdog chip 271 which detects a hardware lockup or impasse and automatically restarts (reboots) the processor. In other words, if watchdog chip 271
10 senses that one of the NIC cards (e.g., fibre channel cards) has gone bad and there is an impasse or lock up because of this failure and no processing is occurring, hardware watchdog 271 will reboot the computer.

The monitoring capabilities are preferably implemented with AI and Daisy lists. For example, the computer preferably trains itself in pattern recognition. It notes, for
15 instance, that in last 2 months the temperature in a particular node is constant with a standard deviation of .3 degrees. It is thereafter determined that micro changes of .1 or .2 degree are occurring. Such changes can be interpreted as the beginning of a silicon failure inside the CPU. If the CPU thereafter actually fails and dies, AI recognizes this pattern wherein if and when a CPU temperature starts fluctuating in such a manner, the
20 CPU is deemed likely to crash, e.g., within 7 days. The AI system preferably then notifies the system operator before the crash happens. In other words, the AI system preferably learns to recognize the physical characteristics of the CPU silicon and predicts a silicon failure and acts to initiate a replacement of the processor or perform backup.

25 There are also software control watchdog components within certain operating systems, one of which happens to be Linux. Alternatively, one can obtain commercial software watchdog programs which monitor software applications. These programs first try to abort the application if there is a processing error, and, if aborting does not work, then the software control watchdog preferably causes a physical shut down and

restart of the machine. Thus, the present invention preferably includes both software and hardware monitoring components. Most preferably, a hardware watchdog component is provided on the mainboard of each nBoxen, nMime and Limbix.

5 PREFERRED USES OF THE COMPUTER/SERVER OF THE PRESENT INVENTION

In one embodiment, the present invention operates as an Internet server wherein the machine acts as a gateway, i.e., two network cards are provided: a network card that communicates with the internal network of, e.g., Figure 11 and a network card that communicates with the Internet. Such a configuration provides a "firewall"
10 architecture.

Significantly, in conventional computer systems, the only time that multiple interfaces are implemented, i.e., multiple HBA network cards are placed inside one computer, is for communicating with different destinations. However, the four-way connections of the present invention are provided not necessarily to "get to" four
15 different places, but to provide four different ways to compute or handle data, e.g., four different ways to move the same the data through four different nodes. Thus, using multiple network cards provides the side effect of high availability/redundancy and fault tolerance.

An important aspect of the present invention is that the connection topology
20 enables the machine to function as a connectionist machine, whereby it is possible to implement algorithms that can have more than one way to process data. In this respect, the network itself becomes instrumental to the computations involved, wherein algorithms execute purely in the network alone, with the processors operating as devices used for input/output to the network, which is unlike a conventional
25 relationship between a processor and its network today. Data moving through the network convene at the nBoxen, where, depending upon the pattern of data incoming from the various network interfaces, determines how the data should leave via the remaining interfaces. The machine can thus behave as a neural network, in which weighting coefficients, made available via input connections, determine output signals.

In another implementation, the present invention can be configured like a four lane highway: four separate paths or one big open four lane highway to the same information. Under the first scenario, one can write a program that, e.g., examines four different scenarios to find out which one functions best. This is, in effect, a program
5 that distribute four different scenarios simultaneously, one down each path, and then each path returns an answer at which point the originating source reads those answers and then makes a decision based on what the answers of those scenarios were. This is a conventional parallel processing scenario albeit one that the architecture of the present invention can easily support.

10 Alternatively, and in accordance with the present invention, so-called "lazy" concepts and genetic algorithms may be employed. Such concepts and algorithms are typically unworkable in a traditional parallel computing approach where, as described above, a master computer has, e.g., tasks 1, 2, 3, 4 to accomplish. So, if there are four nodes, the master computer farms out the tasks to the several processors: "you do task
15 1, you do task 2," etc., "and get back to me with the answer when you are done." In contrast, the present invention enables computing that is more analogous to how a brain functions.

Specifically, in the brain, one gains knowledge via neurons and neural activity. An electrical impulse passes through a neuron(s) and the neuron(s) may or may not
20 relate to that impulse. The neuron may have the knowledge that it is seeking, yet passes the information through. Even if three of four possible neurons have no idea how to respond to that impulse, the one remaining neuron still may. In reality, the brain comprises millions of neurons connected in a connectionist fashion. Accordingly, the idea of a connectionist machine is that between any two points it is not necessarily
25 known which node might have accomplished a particular task. All that is known is that somewhere within the network, at least one node had an answer and that node returned that answer to the program that initiated the process. Thus, in the connectionist machine of the present invention, there is no need for a "task master" that dictates the tasks of each individual node. Rather there are many nodes and connections and a
30 "survival of the species" reaction can take place, wherein whichever node knows the

information, or whichever node has developed an appropriate algorithm to obtain the desired information, sends the information through. Where two or more nodes execute a process, those nodes might accomplish the task using different algorithms. This is the concept of a genetic algorithm. Inside each one of these nodes a different little
5 environment might evolve with respect to how to process information, and so a requesting "neuron" (node) may actually get two answers back, and then from there the requesting node can choose which of the answers is better.

Of course, weighting factors may also be applied. So, rather than the taskmaster breaking the problem up into four parts and sending those parts to four sub slaves, with
10 each sub slave working on one problem, the task is "offered up" and the components (nodes) that know how to relate to the task act on the task and pass it through.

Another benefit achieved by the present invention is its natural affinity for programs that implement recursion. Recursion is a fundamental concept of LISP (and Daisy as well) because it allows for abstract/complex algorithms to develop. However,
15 most current schools of thought regard recursion as a poor programming technique since conventional hardware architecture (such as PC hardware), and the tools most favored for that platform (C/C++), can not easily support recursive techniques. Specifically, recursion is very resource intensive (requires high overhead, memory, CPU time, etc.) and is generally very slow on a single processor computer.

20 On the other hand, the architecture of the present invention is well-suited to recursive techniques, and, more particularly, to the implementation of programming using suspended construction techniques. Suspended construction is a way of chaining individual elements of an expression together in a recursive manner, so that an expression can be built symbolically. In traditional computer programming, one can
25 define, constants, for example, and thereafter use those constants in equations with multiple variables. Further, in conventional computing, each time one of the variables is assigned a value, memory must be allocated and a final value calculated and stored.

In suspended construction, memory is not allocated for assigning or determining values on a step-by-step basis. Rather, the entire calculation is seen as a process. A

mapping is stored of the calculations that might ultimately have to be accomplished. However, the value of each variable is not necessarily evaluated on step-by-step basis. Instead, each variable is created recursively based on each of its constituent elements. When a finite value is called for, the answer is calculated in a recursive manner, back
5 propagating through all of the elements that define the called-for finite value (those elements themselves may also be comprised of many other elements, which must now also be backward propagated as well). Each "suspended construct", therefore, is actually a separate process.

In view of potentially large computations resulting from the manipulation of
10 these suspended constructs, load balancing techniques preferably are implemented when it is necessary to calculate the value of a certain variable. This is a way of distributing processes, which the present invention is particularly capable of accomplishing. Specifically, Daisy, combined with the Linux kernel modifications provided by MOSIX, enable the present invention to operate in this desired fashion.

Further, the machine of the present invention can be dynamically configured to
15 handle various scenarios including running large databases, web servers (as mentioned above), FTP servers and/or mail servers. The actual configuration is accomplished through external connection to a Limbix 500, Limbix commands, and configuration files for Mosix. Preferably, a single graphic user interface is provided to configure
20 Limbix 500.

In view of the foregoing, the present invention, in contrast to the prior art, provides a high performance parallel processing computer that is very small, and has modular hot swappable components. Instead of putting 16 processors on one mother board as is common in conventional shared-memory parallel processing architectures,
25 the present invention provides 16 hot swappable independent mini computers (nodes) interconnected in a connectionist fashion, thereby providing a modular, scalable, self-contained computing environment.

A connectionist machine like that described herein is desirable because of the availability of all of the nodes, which can be used to distribute standard programming

tasks or higher level recursive tasks. Thus, theoretical AI algorithms can be implemented with the distributed memory of the present invention, i.e., nBoxen.

The foregoing disclosure of the preferred embodiments of the present invention has been presented for purposes of illustration and description. It is not intended to be
5 exhaustive or to limit the invention to the precise forms disclosed. Many variations and modifications of the embodiment described herein will be obvious to one of ordinary skill in the art in light of the above disclosure. The scope of the invention is to be defined only by the claims appended hereto, and by their equivalents.

WHAT IS CLAIMED IS:

1. A scalable, multi-processor connectionist computer system, comprising:
 - 5 an enclosure;
 - a plurality of independent computers each having a motherboard and two hard drives, the independent computers being mounted in the enclosure;
 - at least one secondary storage computer mounted in the enclosure, the secondary storage computer providing redundant storage for data stored on at least one
 - 10 of the two hard drives in each of the independent computers;
 - at least one front end computer mounted in the enclosure, the front end computer providing interfacing to the exterior of the connectionist computer system and controlling at least one of routing, load balancing, vectorizing and process migrating among the independent computers; and
 - 15 at least one network switch, the switch establishing network connections among each of independent computers, the secondary storage computer and the front end computer,
 - wherein the independent computers, the secondary storage computer and the front end computer are connected to the network by at least four different paths.
 - 20
2. The connectionist computer system of claim 1, wherein a first of the two hard drives in each of the independent computers stores an operating system of the independent computer and a second of the two hard drives in each of the independent computers stores data.
- 25
3. The connectionist computer system of claim 1, wherein each of the independent computers is hot-swappable.
4. The connectionist computer system of claim 1, wherein each of the
- 30 independent computers includes health monitoring and hardware watchdog components.

5. The connectionist computer system of claim 1, wherein each of the independent computers is connected to a backplane including at least four connectors each capable of receiving a network interface card.

5 6. The connectionist computer system of claim 5, wherein the network interface card is a fibre channel network interface card.

7. The connectionist computer system of claim 5, wherein the network interface card is a fiber optic network interface card.

10

8. The connectionist computer system of claim 1, wherein the each independent computer includes a quick-connect device enabling the independent computer to be easily removed and replaced in the enclosure.

15 9. The connectionist computer system of claim 1, wherein four of the plurality of independent computers are mounted side-by-side in the enclosure.

10. The connectionist computer system of claim 9, wherein the enclosure holds four rows of four independent computers mounted side-by-side.

20

11. The connectionist computer system of claim 1, wherein programs and applications are launched on the front end computer.

12. The connectionist computer system of claim 1, including two front end computers, wherein one of the two front end computers functions as a failover front end computer.

13. The connectionist computer system of claim 1, wherein the front end computer is operable to at least one of (i) monitor the health of the connectionist computer system, (ii) configure the connectionist computer system, (iii) maintain a firewall, (iv) assign tasks to one or more of the plurality of independent computers, and (v) effect routing within the connectionist computer system.

14. The connectionist computer system of claim 1, wherein the at least one secondary storage computer comprises a redundant array of independent drives (RAID) facility.

5

15. The connectionist computer system of claim 1, wherein the secondary storage computer and the front end computer comprise at least four network interface cards for connectivity to the network.

10

16. The connectionist computer system of the claim 1, wherein the groups of four independent computers are connected via the network in a tetrahedron topology.

17. The connectionist computer system of claim 1, wherein the secondary storage computer and the front end computer are hot swappable.

15

18. The connectionist computer system of claim 1, wherein the connectionist computer system is operable as an Internet server.

19. The connectionist computer system of claim 1, wherein the number of independent computers in the system is scalable by orders of 16 such that there are 16, 256 or 4096 independent computers.

20

20. A connectionist topology computer/server, comprising:

at least one cluster of four independent computers, the four independent computers being connected to each other in a tetrahedron topology via a plurality of network interface cards installed in each of the independent computers, the tetrahedron topology being self recursive whereby a plurality of clusters are connectable to achieve a super cluster of four clusters and whereby the super cluster has the same tetrahedron topology as the one cluster of four independent computers, each of the four independent computers having a communication pathway leading outside of the tetrahedron topology for connection to components other than the four independent computers;

25

30

at least one secondary storage computer connected to one of the communication pathways; and
at least one front end computer connected to another of the communication pathways,
wherein the at least one secondary storage computer and the at least one front end
5 computer are connected to each other via a communication pathway other than the
communication pathways leading outside of the tetrahedron topology.

21. The connectionist topology computer/server of claim 20, wherein each of
the independent computers comprises two hard drives, a first hard drive storing an
10 operating system and the hard drive storing data.

22. The connectionist topology computer/server of claim 20, wherein at least
the independent computers are hot swappable.

15 23. The connectionist topology computer/server of claim 20, wherein at least
one of the secondary storage computer and the front end computer is hot-swappable.

24. The connectionist topology computer/server of claim 20, wherein each of
the independent computers includes health monitoring and hardware watchdog
20 components.

25 25. The connectionist topology computer/server of claim 20, wherein each of
the independent computers is connected to a backplane including at least four
connectors, each capable of receiving a network interface card.

26. The connectionist topology computer/server of claim 25, wherein the
network interface card is a fibre channel network interface card.

27. The connectionist topology computer/server of claim 25, wherein the
30 network interface card is a fiber optic network interface card.

28. The connectionist topology computer/server of claim 20, wherein the communication pathway comprises fiber optics.

5 29. The connectionist topology computer/server of claim 20, wherein each independent computer includes a quick-connect device enabling the independent computer to be easily removed and replaced in a rack.

30. The connectionist topology computer/server of claim 20, wherein four clusters are connected to each other for a total of 16 independent computers.
10

31. The connectionist topology computer/server of claim 20, wherein programs and applications are launched from the front end computer.

32. The connectionist topology computer/server of claim 20, including two front end computers, wherein one of the two front end computers functions as a failover front end computer.
15

33. The connectionist topology computer/server of claim 20, wherein the front end computer is operable to at least one of (i) monitor the health of the connectionist topology computer/server, (ii) configure the connectionist topology computer/server, (iii) maintain a firewall, (iv) assign tasks to one or more of the plurality of independent computers, and (v) effect routing within the connectionist topology computer/server.
20

34. The connectionist topology computer/server of claim 20, further comprising a distributed RAID implemented among the independent computers.
25

35. The connectionist topology computer/server of claim 20, wherein at least one of the secondary storage computer and the front end computer is hot swappable.

30 36. The connectionist topology computer/server of claim 20, wherein the connectionist topology computer/server is operable as an Internet server.

37. A modular, scalable, connectionist computer operable to perform complex processing, including genetic algorithms and artificial intelligence, the connectionist computer comprising:

a plurality of nodes connected to each other via at least one switch, each node being
5 one of (i) a hot swappable independent computer, (ii) a secondary storage computer, and (iii) a front end computer,

the independent computer comprising a first hard drive substantially dedicated to storing an operating system and a second hard drive substantially dedicated to storing data,

10 the secondary storage computer comprising a RAID facility that is operable to capture data from the second hard drive of a first independent computer that has been removed to the second hard drive of a second independent computer that has taken the place of the first independent computer,

the front end computer being operable as an interface for at least one of a user, another
15 computer system and a network external to the connectionist computer,

wherein each of the nodes includes at least four communication pathways to the at least one switch, whereby each node has more than one pathway to and from any one other node.

20 38. The connectionist computer of claim 37, wherein nodes that are independent computers are organized in clusters of four.

39. The connectionist computer of claim 38, wherein the clusters are, topologically, a tetrahedron.

25

40. The connectionist computer of claim 39, comprising 4 clusters.

41. The connectionist computer of claim 40, further comprising two secondary storage computers and two front end computers.

42. The connectionist computer of claim 41, wherein each of the secondary storage computers and the front end computers comprises at least four communication pathways to the at least one switch.
- 5 43. The connectionist computer of claim 37, wherein the operating system of the front end computer is Unix or a derivative thereof.

1/9

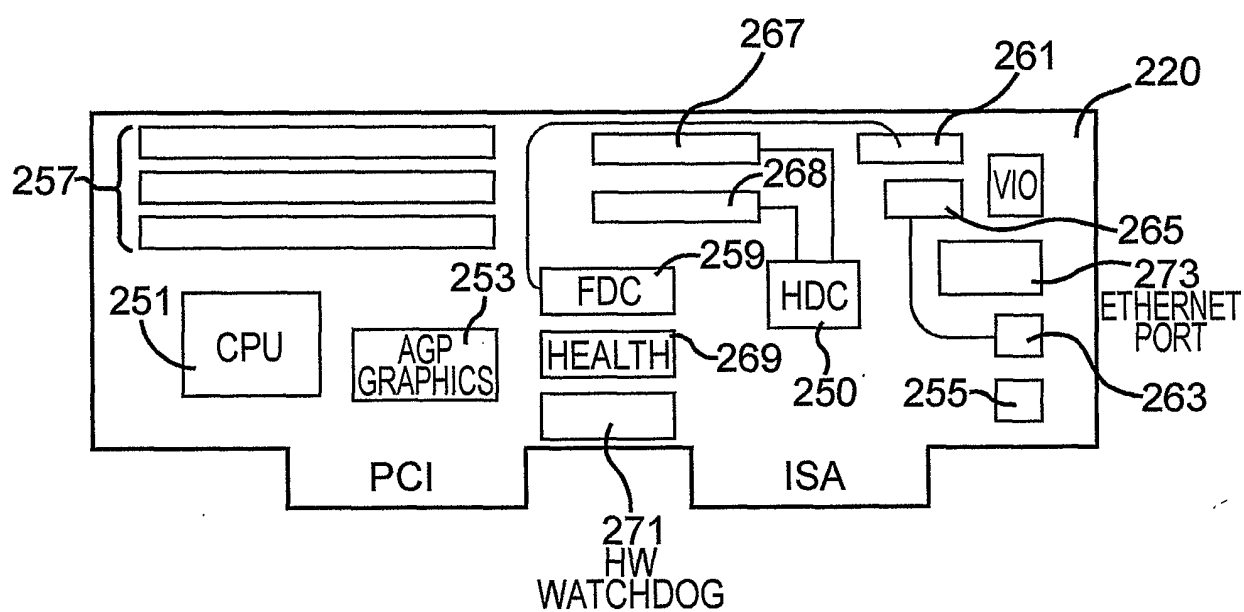
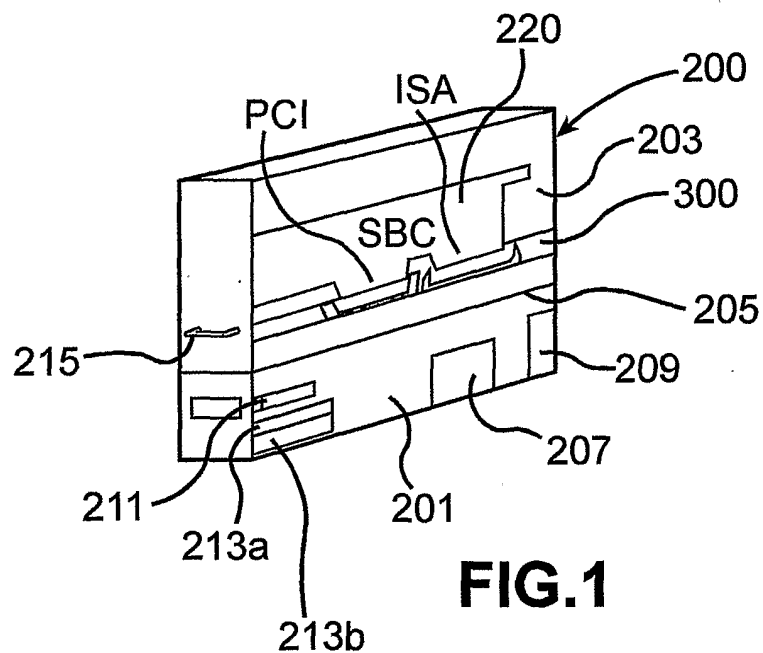


FIG.2

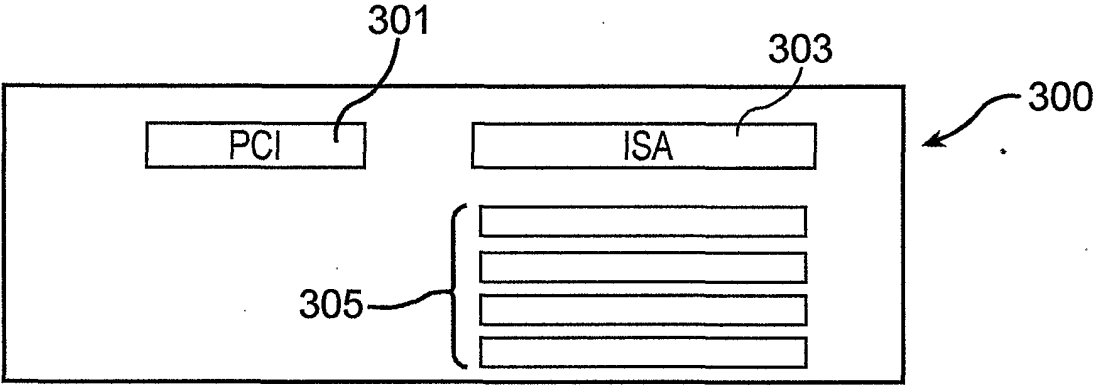


FIG.3

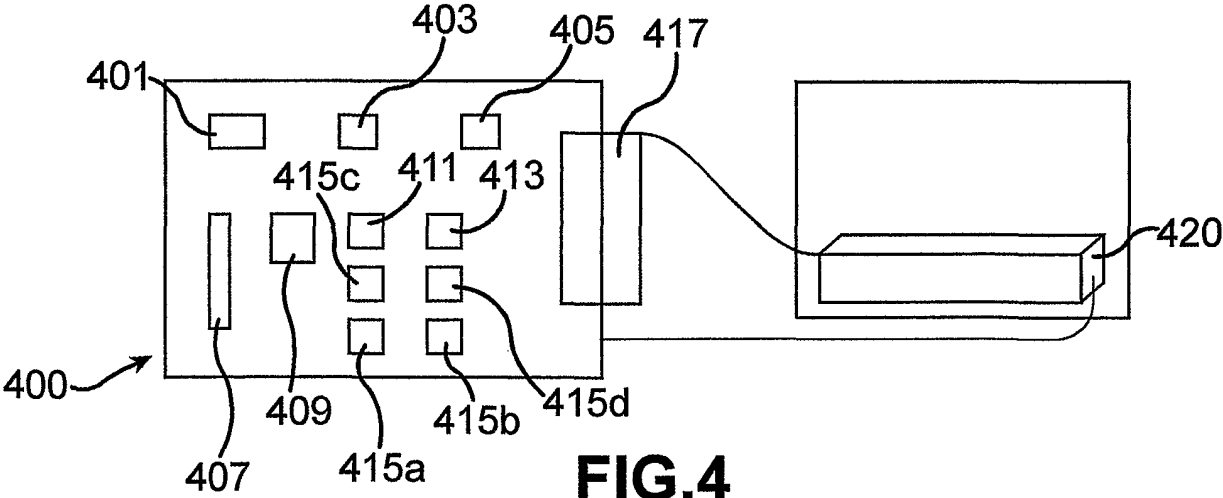


FIG.4

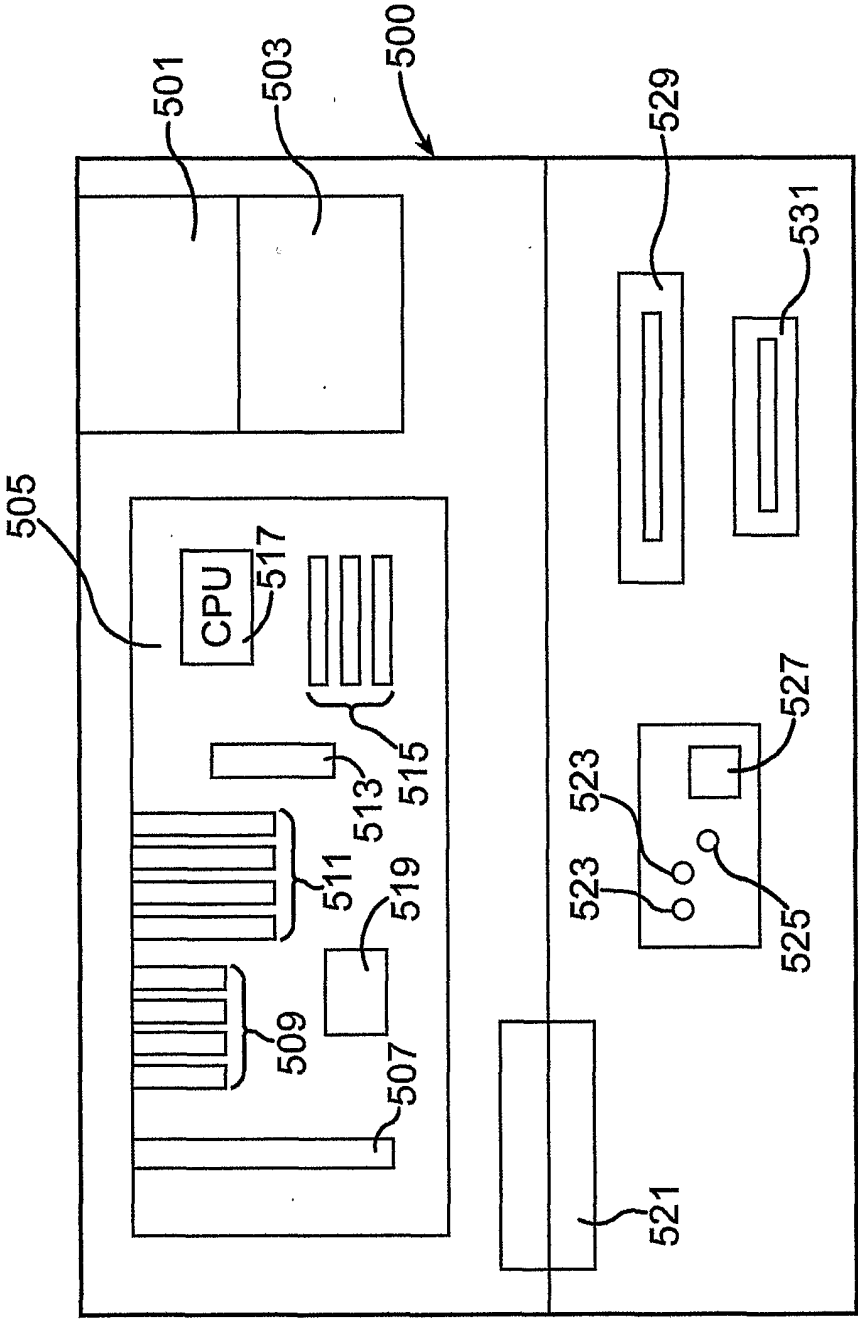


FIG.5

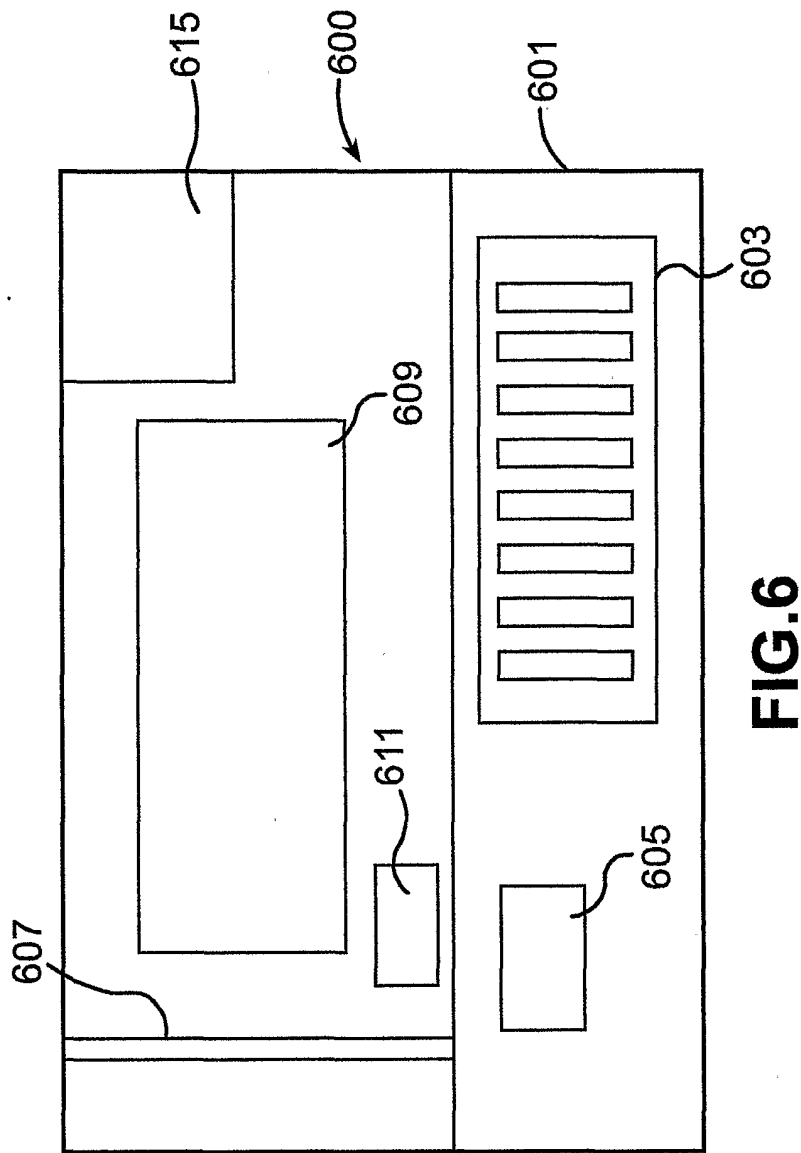


FIG. 6

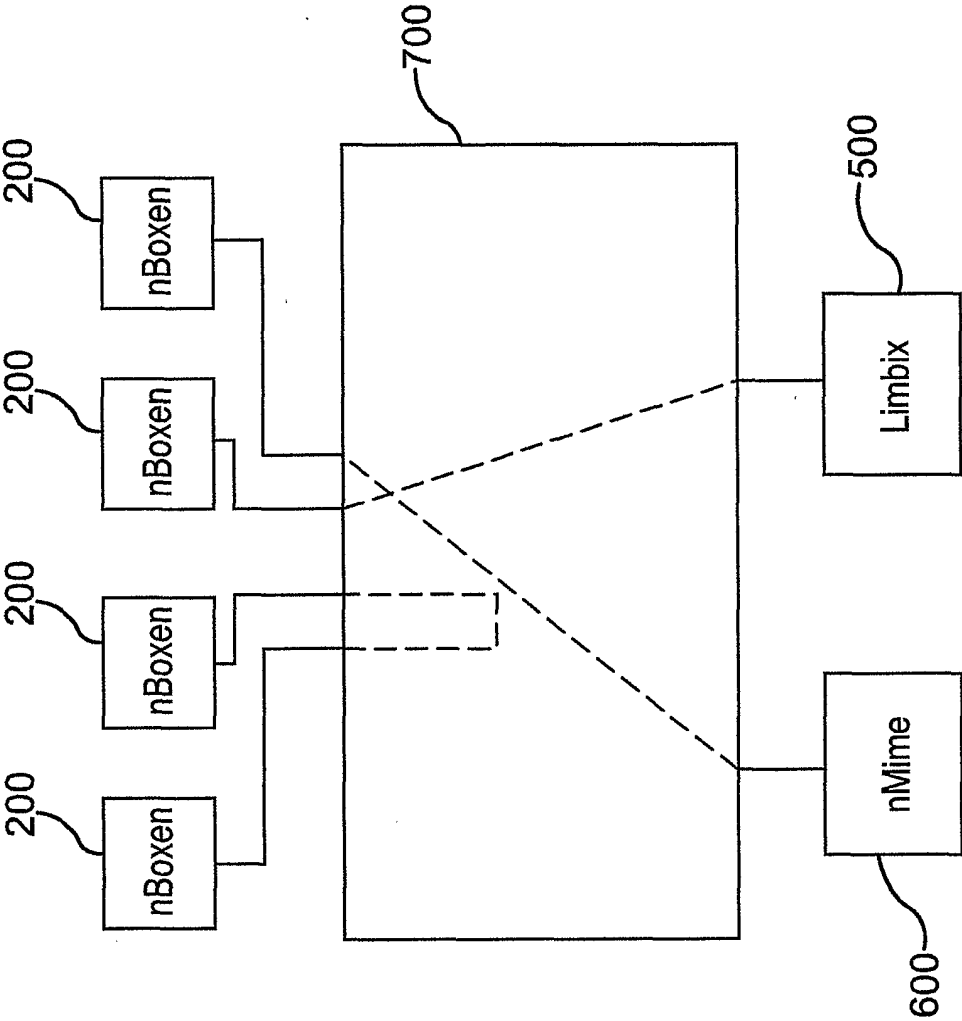
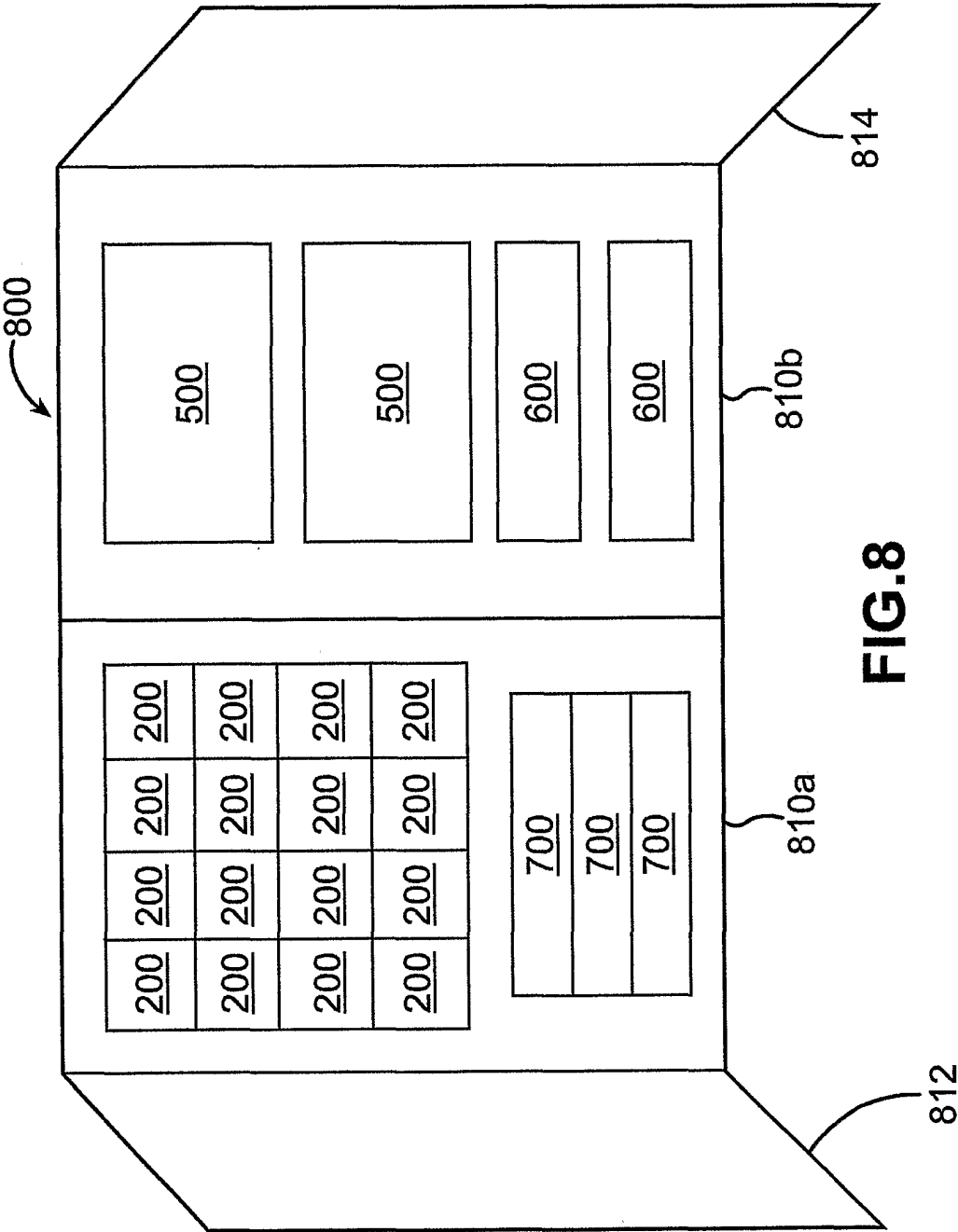


FIG. 7



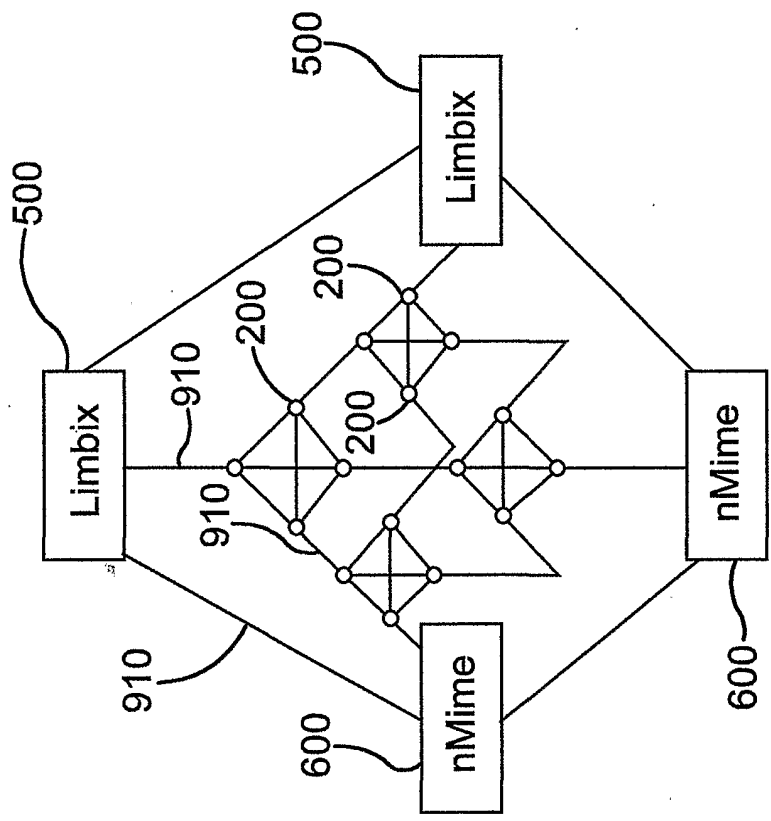


FIG.9

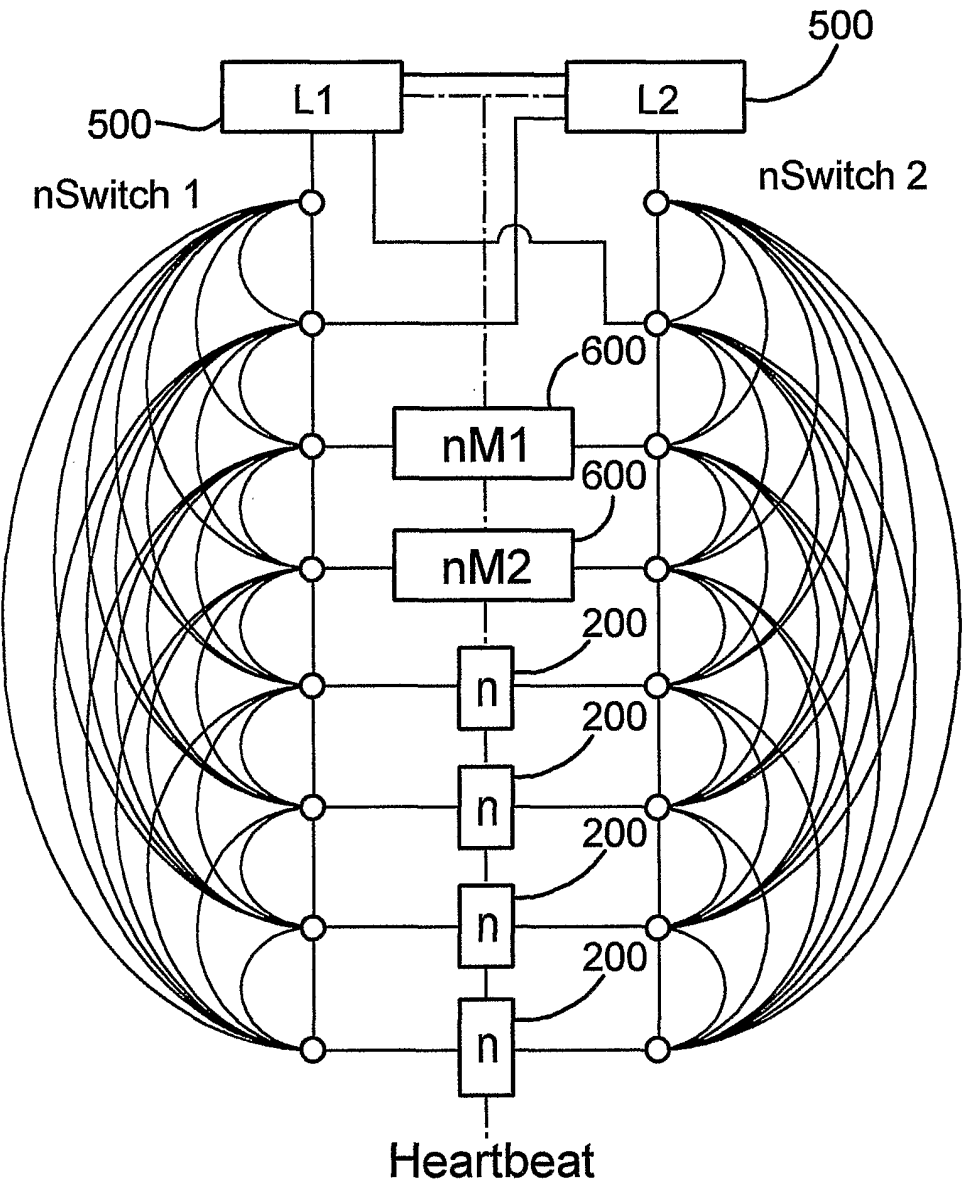
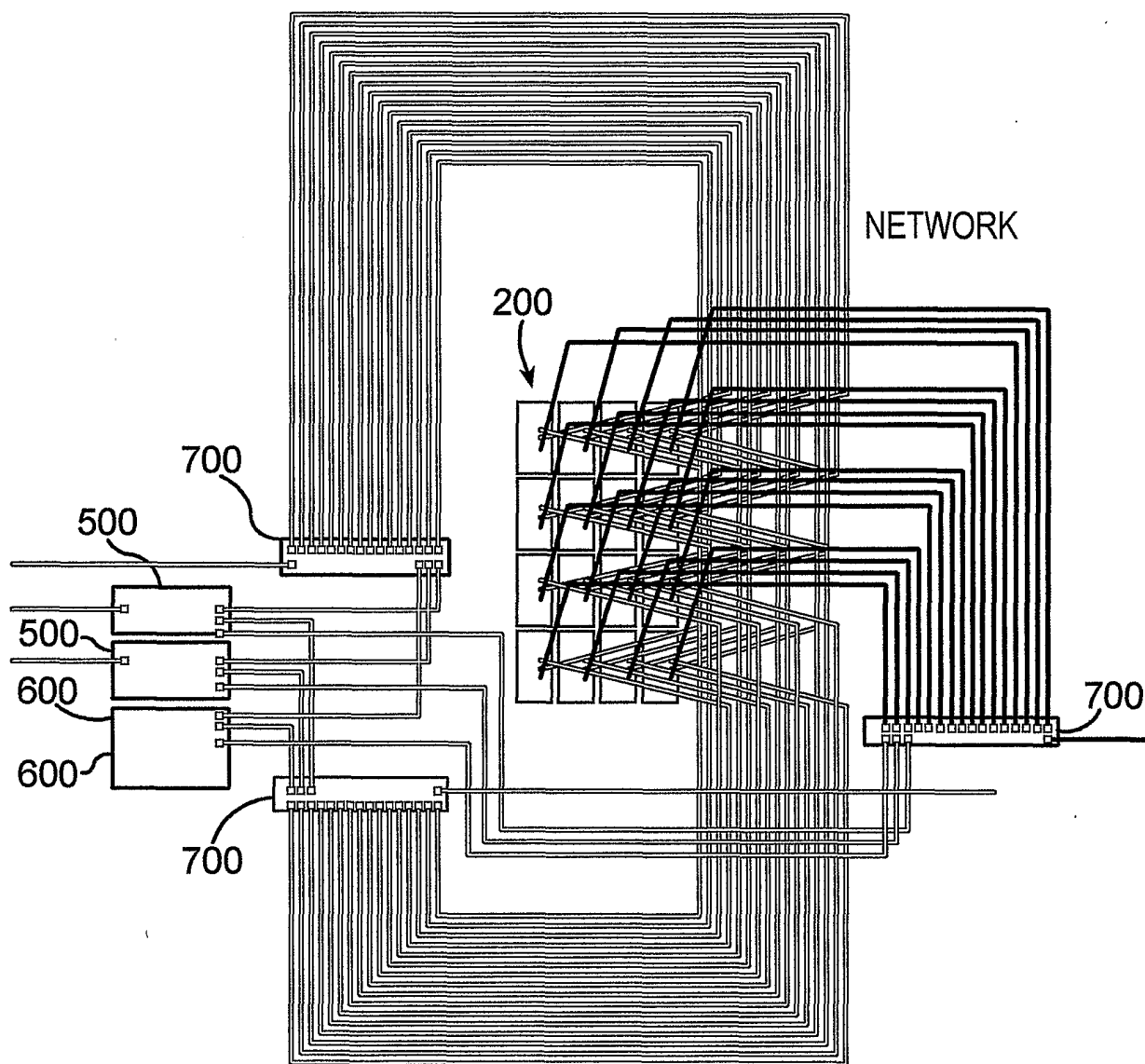


FIG.10

**FIG.11**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/15128

A. CLASSIFICATION OF SUBJECT MATTER		
IPC(7) : G06F 11/00 US CL : 714/11		
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)		
U.S. : 714/11, 12, 13, 152, 6; 711/114; 709/201, 203238, 248;		
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 practicable, search terms used)		
APS, WEST, EPAB, JPAB, DWPI, TDBD, PGPUB		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,630,045 A (KRYGOWSKI et al) 13 May 1997, Entire document.	1-43
Y	US 5,812,757 A (OKAMOTO et al) 22 September 1998, Entire document.	1-43
Y	US 5,790,776 A (SONNIER et al) 04 August 1998, Entire document.	1-43
Y	US 5,579,491 A (JEFFRIES et al) 26 November 1996, Entire document.	1-43
Y	US 5,448,723 A (ROWETT) 05 September 1995, Entire document.	1-43
Y	US 5,892,928 A (WALLACH et al) 06 April 1999, Entire document.	1-43
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents:	"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
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"E" earlier document published on or after the international filing date	"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
"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)	"&"	document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means		
"P" document published prior to the international filing date but later than the priority date claimed		
Date of the actual completion of the international search	Date of mailing of the international search report	
18 JULY 2001	14 AUG 2001	
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer NADEEM IQBAL <i>James R. Matthews</i> Telephone No. (703) 308-5228	