

(51) International Patent Classification:
G06F 13/14 (2006.01)(21) International Application Number:
PCT/US2014/047703(22) International Filing Date:
22 July 2014 (22.07.2014)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT-PACKARD DEVELOPMENT COMPANY, LP** [US/US]; 11445 Compaq Center Drive W, Houston, Texas 77070 (US).(72) Inventors: **LI, Sheng**; 1501 Page Mill Rd., Palo Alto, California 94304-1100 (US). **ZHAO, Jishen**; 1501 Page Mill Rd., Palo Alto, California 94304-1100 (US). **LIM, Kevin T**; 1501 Page Mill Rd., Palo Alto, California 94304-1100 (US). **FARABOSCHI, Paolo**; 3000 Hanover St., Palo Alto, California 94304-1112 (US).(74) Agents: **KUIPERS, Luke** et al.; Hewlett-Packard Company, Intellectual Property Administration, 3404 E. Harmony Road, Mail Stop 35, Fort Collins, Colorado 80528-9599 (US).

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

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,

[Continued on next page]

(54) Title: NODE-BASED COMPUTING DEVICES WITH VIRTUAL CIRCUITS

(57) Abstract: According to an example, a node-based computing device includes memory nodes communicatively coupled to a processor node. The memory nodes may form a main memory address space for the processor node. The processor node may establish a virtual circuit through memory nodes. The virtual circuit may dedicate a path within the memory nodes. The processor node may then communicate a message through the virtual circuit. The memory nodes may forward the message according to the path dedicated by the virtual circuit.

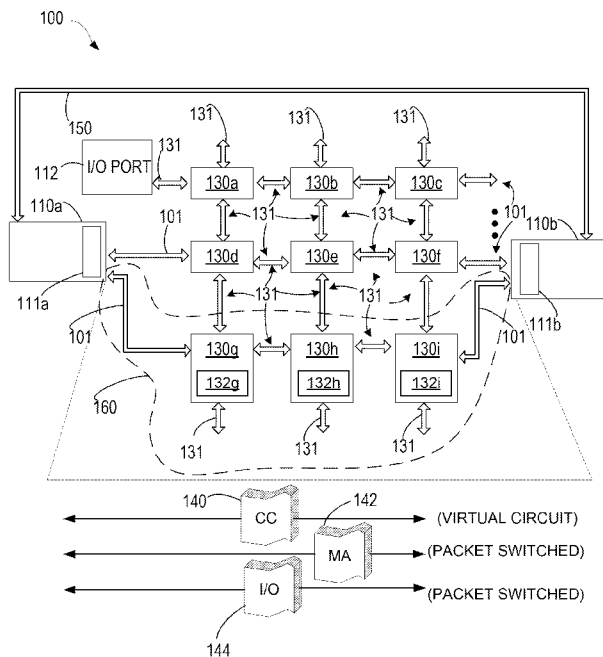


FIG. 1



TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, **Published:**
KM, ML, MR, NE, SN, TD, TG).

— *with international search report (Art. 21(3))*

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

NODE-BASED COMPUTING DEVICES WITH VIRTUAL CIRCUITS

BACKGROUND

[0001] Computer networks and systems have become indispensable tools for modern business. Today terabytes or more of information on virtually every subject imaginable are stored and accessed across networks. Some applications, such as telecommunication network applications, mobile advertising, social media applications, etc., demand short response times for their data. As a result, new memory-based implementations of programs, such as in-memory databases, are being employed in an effort to provide the desired faster response times. These memory-intensive programs primarily rely on large amounts of directly addressable physical memory (e.g., random access memory) for storing terabytes of data rather than hard drives to reduce response times.

-2-

BRIEF DESCRIPTION OF DRAWINGS

[0002] The following description illustrates various examples with reference to the following figures:

[0003] FIG. 1 is a diagram showing a node-based computing device, according to an example;

[0004] FIG. 2 is flowchart illustrating a method for using a virtual circuit to communicate messages in a node-based computing device, according to an example;

[0005] FIG. 3 is a flowchart illustrating a method for dynamically establishing a virtual circuit during run-time execution of a node-based computing device, according to an example;

[0006] FIG. 4 is a diagram showing the node-based computing device of FIG. 1 with voltage and frequency domains, according to an example; and

[0007] FIG. 5 is a block diagram of a computing device capable of communicating data using a virtual circuit, according to one example.

DETAILED DESCRIPTION

[0008] For simplicity and illustrative purposes, the principles of this disclosure are described by referring mainly to examples thereof. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the examples. It is apparent that the examples may be practiced without limitation to all the specific details. Also, the examples may be used together in various combinations.

[0009] A node-based computing device, according to an example, includes a processor node and memory nodes. The processor node may be communicatively coupled to the memory nodes via interconnects, such as point-to-point links. Further, the memory nodes may also be communicatively coupled to each other via interconnects, such as point-to-point links. Each memory node may be a memory subsystem including a memory controller and memory to store data. Each memory node may also include routing logic to route message data to a destination, which may be another memory node, processor node, or an input/output (“I/O”) port in the node-based computing device. Collectively, the memory nodes may provide a main memory address space for processor nodes.

[0010] Examples may use the memory nodes and the point-to-point links of a node-based computing device as a messaging fabric to communicate different protocol types carrying different types of messages, such as cache coherency messages, memory access command messages, and I/O messages, to given memory nodes, I/O ports, or processors of the node-based computing device.

[0011] In an example discussed herein, a node-based computing device may include processor nodes and memory nodes. Each memory node may include local memory. The local memory from the memory devices may collectively form a main memory address space of the node-based computing device. Point-to-point links may communicatively couple the memory nodes to one of the processor nodes, the memory nodes to another processor node, and the memory nodes to each other. One of the processor nodes may include a processor-side memory

-4-

controller. The processor-side memory controller may establish a virtual circuit between the processor node and another processor node. The virtual circuit may dedicate a path through the memory nodes. The processor-side memory controller may communicate a cache coherency message to the another processor node using the path dedicated through the virtual circuit.

[0012] In another example, a processor node may detect a high use memory node. A high use memory node may be a memory node from memory nodes communicatively coupled to the processor node. The memory nodes may form an addressable memory space for the processor node. The processor node may establish a virtual circuit that dedicates a communication path from the processor node of the high use memory node. The processor node may also communicate subsequent messages through the virtual circuit. The memory nodes may forward the subsequent messages according to the dedicated path.

[0013] These and other examples are now described.

[0014] FIG. 1 is a diagram showing a node-based computing device 100, according to an example. The node-based computing device 100 may include processor nodes 110a,b and memory nodes 130a-i. The processor nodes 110a,b may be compute units that are configured to execute computer-readable instructions and operate on data stored in the memory nodes 130a-i. As FIG. 1 shows, the processor nodes 110a,b may include processor-side memory controllers 111a,b that are connected (directly or indirectly) to the memory nodes 130a-i of the node-based computing device 100 via point-to-point links 101. A point-to-point link may be a wire or other connection medium that links two circuits. In an example, a point-to-point link connects only two circuits which is unlike a shared bus or crossbar switches that connect more than two circuits or devices. A processor node and processor-side memory controller connected to the node-based computing device 100, such as 110a and 111a or 110b and 111b, may be provided on the same chip, or may be provided on separate chips. Also, more or fewer processor nodes, processor-side memory controllers and memory nodes than

-5-

shown in FIG. 1 may be used in the node-based computing device 100. Also, an I/O port 112 may be connected to the node-based computing device 100. The I/O port 112 may be linked to a network device, a memory device, a data link or bus, a display terminal, a user input device, or the like.

[0015] The processor nodes 110a,b may, in some cases, be connected to each other with a direct processor-to-processor link 150, which may be a point-to-point link that provides a direct communication channel between the processor nodes 110a,b. In some cases, the processor nodes 110a,b may be configured to use the direct processor-to-processor link 150 to communicate high priority message data that is destined for each other. For example, the processor node 110a may send a cache coherency message destined for the processor node 110b through the direct processor-to-processor link 150 to avoid multiple hops through the memory nodes 130a-i.

[0016] The node-based computing device 100 may include memory nodes 130a-i that may also be connected together via point-to-point links 131, which are inter-node point-to-point links. Each memory node can operate as a destination of message data if the data to be accessed is stored at the memory node, and as a router that forwards message data along a path to an appropriate destination, such as another memory node, one of the processor nodes 110a,b, or the I/O port 112. For example, the processor-side memory controllers 111a,b can send memory access command messages, e.g., read, write, copy, etc., to the memory nodes 130a-i to perform memory access operations for the processor nodes 110a,b. Each memory node receiving message data may execute the command if that memory node is the destination or route the command to its destination memory node. The node-based computing device 100 may provide memory scalability through the point-to-point links 131 and through the ability to add memory nodes as needed, which may satisfy the memory capacity requirements of big-data workloads. Scaling up memory capacity in the node-based computing device 100 may involve, in some cases, cascading additional memory nodes.

-6-

[0017] The node-based computing device 100 may establish virtual circuits to provide quality of service (“QoS”) provisions for messages communicated through the node-based computing device 100. For example, the node-based computing device 100 may establish a virtual circuit, such as the virtual circuit 160, to provide performance bounds (and thereby, latency bounds) and band bandwidth allotments for given types of messages communicated through the memory nodes 130a-i. The virtual circuit 160 may be based on connection oriented packet switching, meaning that data may be delivered along the same memory node path. A possible advantage with a virtual circuit over connectionless packet switching is that in some cases bandwidth reservation during the connection establishment phase is supported, making guaranteed QoS possible. For example, a constant bit rate QoS class may be provided, resulting in emulation of circuit switching. Further, in some cases, less overhead may be used, since the packets (e.g., messages) are not routed individually and complete addressing information is not provided in the header of each data packet. Instead, a virtual channel identifier is included in each packet. Routing information may be transferred to the memory nodes during the connection establishment phase.

[0018] In FIG. 1, the virtual circuit 160 may be used to communicate cache coherency messages 140 from the processor node 110a to the processor node 110b. Accordingly, the node-based computing device 100 may provide QoS services to cache coherency messages using the virtual circuit, while memory access messages 142 and I/O messages 144 are transmitted to destination nodes within the node-based computing device 100 using packet switching. That is, subsequent messages sent from one node to another node may not necessarily travel through the same path of nodes.

[0019] As FIG. 1 shows, memory nodes may include memory-side memory controllers. For example, memory nodes 130g-i may include memory-side memory controllers 132g-i, respectively. The memory-side memory controllers 132g-i may include logic for accessing local memory (e.g., the memory-side memory controller

-7-

132g may include logic for accessing memory of the memory node 132) and routing messages to other nodes. A memory-side memory controller may include hardware, logic, and/or machine readable instructions stored on a storage device and executable by hardware. As just mentioned, a memory-side memory controller may perform the operations involved in executing memory access operations on memory local to a memory node. For example, the memory-side memory controller 132g can receive packets from other memory nodes, decode the packets to extract the memory access commands and enforce memory management mechanisms that may be implemented and the actual execution of the read, write, and block copy commands from local memory. To illustrate, after receiving a read command from the processor-side memory controller 111a of the processor node 110a, the memory-side memory controller 132g can fetch the data from local memory and notify the processor-side memory controller 111a that the data is ready to be accessed or directly sends a data packet with a transaction identifier and the requested data back to the processor-side memory controller 111a. These mechanisms depend on the specific type of the memory technology employed in the node; for example, the memory-side co-memory controller for DRAM is different from the co-memory controller for a DRAM stack, for flash memory, or for other forms of non-volatile memory.

[0020] In terms of routing, the memory-side memory controller 132g (or any other memory-side memory controller) may receive message data, determine whether the message data relate to a memory address mapped to local memory of the memory node 130g, and, if so, the memory-side memory controller 132g fetches data from the local memory. If the memory node 130g is not the destination, the memory-side memory controller 132g may send the message data to a next hop in the node-based computing device 100 toward the destination along one of the point-to-point links 131.

[0021] FIG. 2 is flowchart illustrating a method 200 for using a virtual circuit to communicate messages in a node-based computing device, according to an

-8-

example. The method 200 may be performed by the modules, logic, components, or systems shown in FIG. 1 and, accordingly, is described herein merely by way of reference thereto. It will be appreciated that the method 200 may, however, be performed on any suitable hardware. The method of 200 may be performed by the node-based computing device 100 during system start-up to reserve or otherwise establish a virtual circuit usable to provide performance bounds (and thereby, latency bounds) and band bandwidth allotments for given types of messages communicated through the memory nodes 130a-i.

[0022] The method 200 may begin at operation 202 when the processor-side memory controller 111a establishes the virtual circuit 160. The virtual circuit 150 may include a path within the memory nodes between the memory processor node 110a and the processor node 110b. In some cases, the virtual circuit 150 may act as a dedicated path (e.g., memory nodes 130g-i, and corresponding point-to-point links) between the processor nodes 110a,b which may be used to communicate messages of a given type. Cache coherency messages are an example of a message type in which the virtual circuit 150 may be used to transmit. In some cases, establishing the virtual circuit 150 may involve the processor-side memory controller 111a reserving performance properties for the virtual circuit, such as a bandwidth or a priority. With virtual circuits, the processor-side memory controller 111a can also apply dynamic voltage and frequency scaling (“DVFS”) on different virtual channels in the node-based computing device to favorably deliver power to the memory nodes and links with high priority virtual channels, ensuring, in some cases, that messages are delivered in time to meet the QoS goals. In an example, the node-based computing device 100 can have a power budget, and DVFS can be applied to speed up the virtual circuits by increasing the voltage and/or frequency of the point-to-point links and memory nodes in those circuits. The power budget is maintained by adjusting (e.g., decreasing) the voltage and/or frequency of other paths (e.g., the point-to-point links and memory nodes) in the node-based computing device 100. Thus, the speed of the connections in the node-based

-9-

computing device can vary while maintaining an overall energy budget. Note that applying DVFS in memory nodes 130a-i and point-to-point links 131 may lead to asynchronous network designs. To solve this problem, memory nodes can include buffers to allow for additional packet/message buffering at each node to compensate for the slower rates.

[0023] At operation 204, once the virtual circuit 150 between the processor nodes 110a,b is established, the processor node 110a may communicate a cache coherency message to the processor node 110b using the virtual circuit 150. As just discussed above, the virtual circuit 150 may be used to provide a dedicated path through the memory nodes 130a-i for a given type of message. Accordingly, according to the virtual circuit 160, the cache coherency message 140 may travel from the processor-side controller 111a to the memory-side memory controller 132g, to the memory-side memory controller 132h, to the memory-side memory controller 132i, and, finally, to the processor-side memory controller 111b.

[0024] As discussed above, the method of 200 may be performed by the node-based computing device 100 during system start-up to reserve or otherwise establish a virtual circuit usable to provide performance bounds (and thereby, latency bounds) and bandwidth allotments for given types of messages communicated through the memory nodes 130a-i. In additional or alternative cases, a virtual circuit may be established dynamically during runtime. An example of a case where a virtual circuit can be established during runtime is where a processor node is likely to access a given memory node (e.g., the memory node is known to have data relevant to the processor node). For example, a virtual circuit can be established for a processor node executing a Hadoop worker compute node and the memory node holding its associated map/reduce data. The benefits of these virtual circuits may be to provide dedicated routing paths (and thereby, latency bounds) and bandwidth allotments for specific traffic.

[0025] FIG. 3 is a flowchart illustrating a method 300 for dynamically establishing a virtual circuit during run-time execution of a node-based computing

-10-

device 100, according to an example. Similar to the method 200 of FIG. 2, the method 300 may be performed by the modules, logic, components, or systems shown in FIG. 1 and, accordingly, is described herein merely by way of reference thereto. It will be appreciated that the method 300 may, however, be performed on any suitable hardware.

[0026] The method 300 may begin at operation 302 when the processor-side memory controller 111a detects that a memory node may be a high use memory node. A high use memory node may refer to a memory node that is likely to be the destination of subsequent memory access messages. A number of techniques can be used to signal that a communication path will be high use path. In some cases, the instruction set architecture ("ISA") of the processor node 110a can provide explicit instructions for a programmer/compiler to signal that memory access to a given region (e.g., memory address, data structure, memory node) should be optimized by the processor-side memory controller entity and the node-based computing device. The ISA may also have an instruction to disable a memory address as a high use path.

[0027] In other cases, a node-based computing device 100 can predict when a region or path to a memory node should be optimized. For example, the processor-side memory controller can make such predictions by using performance counters that create a virtual circuit after a rate of activity within a time frame exceeds a threshold amount, and then disables a virtual circuit after a rate of inactivity within a time frame exceeds a threshold amount. The performance counters may be specific to messages being sent to a given address (or range of addresses) or a given memory node. These predictions could detect so called hot zones and cold zones within the node-based computing device in a manner that does not involve programmer or compiler assistance.

[0028] Upon detecting the high use memory node, the processor-side memory controller 111a may then, at operation 304, establish a virtual circuit between the processor node 110a and the high use memory node. With virtual

-11-

circuits, the processor-side memory controller 111a can also apply DVFS on different virtual channels in the node-based computing device to ensure that power is favorably delivered to the memory nodes and links with high priority virtual channels, ensuring that important messages are delivered in time to meet the QoS goals. In an example, the node-based computing device can have a power budget, and DVFS can be applied to speed up the virtual circuits by boosting the voltage and/or frequency of the links and nodes in those circuits. The power budget is maintained by adjusting the voltage and/or frequency of other paths in the node-based computing device. Thus, the speed of the connections in the node-based computing device can vary while maintaining an overall energy budget. Note that applying DVFS in memory nodes and point-to-point links may lead to asynchronous network designs. To solve this problem, memory nodes can include buffers to allow for additional packet/message buffering at each node to compensate for the slower rates.

[0029] With continued reference to FIG. 3, at operation 306, the processor-side memory controller 111a may then transmit subsequent messages destined to the high use memory node via the virtual circuit. The memory-side controllers of the memory nodes 130a-i may manage the routing of messages between the processor node 110a and the memory node in accordance to the virtual circuit established at operation 304.

[0030] As described above, in some cases applying DVFS may lead to asynchronous network designs. Also described above, some implementations of the memory-side memory controllers may include storage buffers to allow for the memory-side memory controllers to buffer incoming messages at varying (e.g., slower) rates.

[0031] An additional option to ease the asynchronous challenges is to partition the node-based computing device into different voltage and frequency domains. Approaches adopting voltage and frequency domains can reduce the degree of asynchrony that each channel of a point-to-point link could potentially

-12-

observe. In these designs, DVFS is applied to a voltage and frequency domain rather than the node-based computing device as a whole. FIG. 4 is a diagram showing the node-based computing device 100 of FIG. 1 with voltage and frequency domains 402, 404, according to an example. A voltage and frequency domain (such as voltage and frequency domains 402, 404) may be a sub-set of the memory nodes of the node-based computing device in which a virtual circuit can be formed. Further, a processor-side memory controller may apply DVFS to the memory nodes of a voltage and frequency domain. Thus, in some cases, the performance of a path in one voltage and frequency domain can be optimized by increasing the energy used by that path and then lowering the frequency/voltage used by other paths in that voltage and frequency domain. In this way, optimizing a path in one voltage and frequency domain does not affect the operation of paths in other voltage and frequency domains.

[0032] Some voltage and frequency domains may include buffers that are optimized for a range of DVFS values. Thus, one domain may include buffers of greater sizes than the buffers found in other domains to accommodate a greater step down in speed. Furthermore, other domains may not allow DVFS and are therefore optimized for a single frequency/voltage model. This hybrid/nonhomogeneous configuration can balance runtime flexibility and design time ease.

[0033] FIG. 5 is a block diagram of a computing device 500 capable of communicating data using a virtual circuit, according to one example. The computing device 500 includes, for example, a processor 510, and a computer-readable storage device 520 including virtual circuit memory controller instructions 522. The computing device 500 may be, for example, a memory node, a processor node, (see FIG. 1) or any other suitable computing device capable of providing the functionality described herein.

[0034] The processor 510 may be a central processing unit (CPU), a semiconductor-based microprocessor, a graphics processing unit (GPU), other

-13-

hardware devices or circuitry suitable for retrieval and execution of instructions stored in computer-readable storage device 520, or combinations thereof. For example, the processor 510 may include multiple cores on a chip, include multiple cores across multiple chips, multiple cores across multiple devices, or combinations thereof. The processor 510 may fetch, decode, and execute one or more of the virtual circuit memory controller instructions 522 to implement methods and operations discussed above, with reference to FIGS. 1-4. As an alternative or in addition to retrieving and executing instructions, processor 510 may include at least one integrated circuit ("IC"), other control logic, other electronic circuits, or combinations thereof that include a number of electronic components for performing the functionality of virtual circuit memory controller instructions 522.

[0035] Computer-readable storage device 520 may be any electronic, magnetic, optical, or other physical storage device that contains or stores executable instructions. Thus, computer-readable storage device may be, for example, Random Access Memory (RAM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), a storage drive, a Compact Disc Read Only Memory (CD-ROM), non-volatile memory, and the like. As such, the machine-readable storage device can be non-transitory. As described in detail herein, computer-readable storage device 520 may be encoded with a series of executable instructions for communicating message data through a node-based computing device using a virtual circuit.

[0036] As used herein, the term "computer system" may refer to one or more computer devices, such as the computer device 500 shown in FIG. 5. Further, the terms "couple," "couples," "communicatively couple," or "communicatively coupled" is intended to mean either an indirect or direct connection. Thus, if a first device, module, or engine couples to a second device, module, or engine, that connection may be through a direct connection, or through an indirect connection via other devices, modules, logic, engines and connections. In the case of electrical

-14-

connections, such coupling may be direct, indirect, through an optical connection, or through a wireless electrical connection.

[0037] While this disclosure makes reference to some examples, various modifications to the described examples may be made without departing from the scope of the claimed features.

-15-

CLAIMS

What is claimed is:

1. A node-based computing device comprising:
 - a first processor node and a second processor node;
 - memory nodes that each include local memory that collectively form a main memory address space of the node-based computing device;
 - point-to-point links communicatively coupling the memory nodes to the first processor node, the memory nodes to the second processor node, and the memory nodes to each other; and
 - the second processor node including a processor-side memory controller to:
 - establish a virtual circuit between the first processor node with the second processor node, the virtual circuit dedicating a path through the memory nodes, and
 - communicate a cache coherency message to the first processor node using the path dedicated through the virtual circuit.
2. The node-based computing device of claim 1, wherein the processor-side memory controller further to communicate a memory access message to one of the memory nodes.
3. The node-based computing device of claim 2, wherein the memory nodes further to communicate the memory access message to the one of the memory nodes using connectionless packet switching.

-16-

4. The node-based computing device of claim 2, wherein the processor-side memory controller message to communicate the cache coherency message and the memory access message to the memory nodes through the same point-to-point link.
5. The node-based computing device of claim 1, wherein the processor-side memory controller further to communicate an input output message to one of an input output port through the memory nodes using connectionless packet switching.
6. The node-based computing device of claim 1, wherein the processor-side memory controller further to apply dynamic voltage and frequency scaling to increase the voltage and frequency of the path dedicated through the virtual circuit.
7. The node-based computing device of claim 1, wherein the processor-side memory controller further to apply dynamic voltage and frequency scaling to decrease the voltage and frequency of paths other than the path dedicated through the virtual circuit.
8. The node-based computing device of claim 7, wherein a degree of the decrease is relative to a power budget of the memory nodes.
9. The node-based computing device of claim 7, wherein the processor-side memory controller further to select the paths based on the paths belonging to a voltage and frequency domain.

-17-

10 The node-based computing device of claim 1, wherein the processor-side memory controller further to establish the virtual circuit during a startup phase of the node-based computing device.

11. A method comprising:

 detecting, by a processor node, a high use memory node, the high use memory node being a memory node of a plurality of memory nodes communicatively coupled to the processor node, the plurality of memory nodes forming an addressable memory space for the processor node;

 establishing a virtual circuit that dedicates a communication path from the processor node of the high use memory node; and

 communicating subsequent messages through the virtual circuit, the memory nodes forwarding the subsequent messages according to the dedicated path.

12. The method of claim 11, wherein detecting the high use memory node comprises executing an instruction set architecture instruction that requests establishment of the virtual circuit.

13. The method of claim 11, wherein detecting the high use memory node comprises determining that a rate of activity associated with the high use memory node exceeds a threshold amount.

14. The method of claim 13, further comprising closing the virtual circuit based on determining that a rate of inactivity associated with the high use memory node exceeds a threshold amount.

-18-

15. A computer-readable storage device comprising instructions that, when executed, cause a processor of a computing device to:

detect a memory node from a plurality of memory nodes as a high use memory node, the plurality of memory nodes forming an addressable memory space for the processor;

establish a virtual circuit that dedicates a communication path from the processor node of the high use memory node; and

communicate subsequent messages to the high use memory node through the virtual circuit.

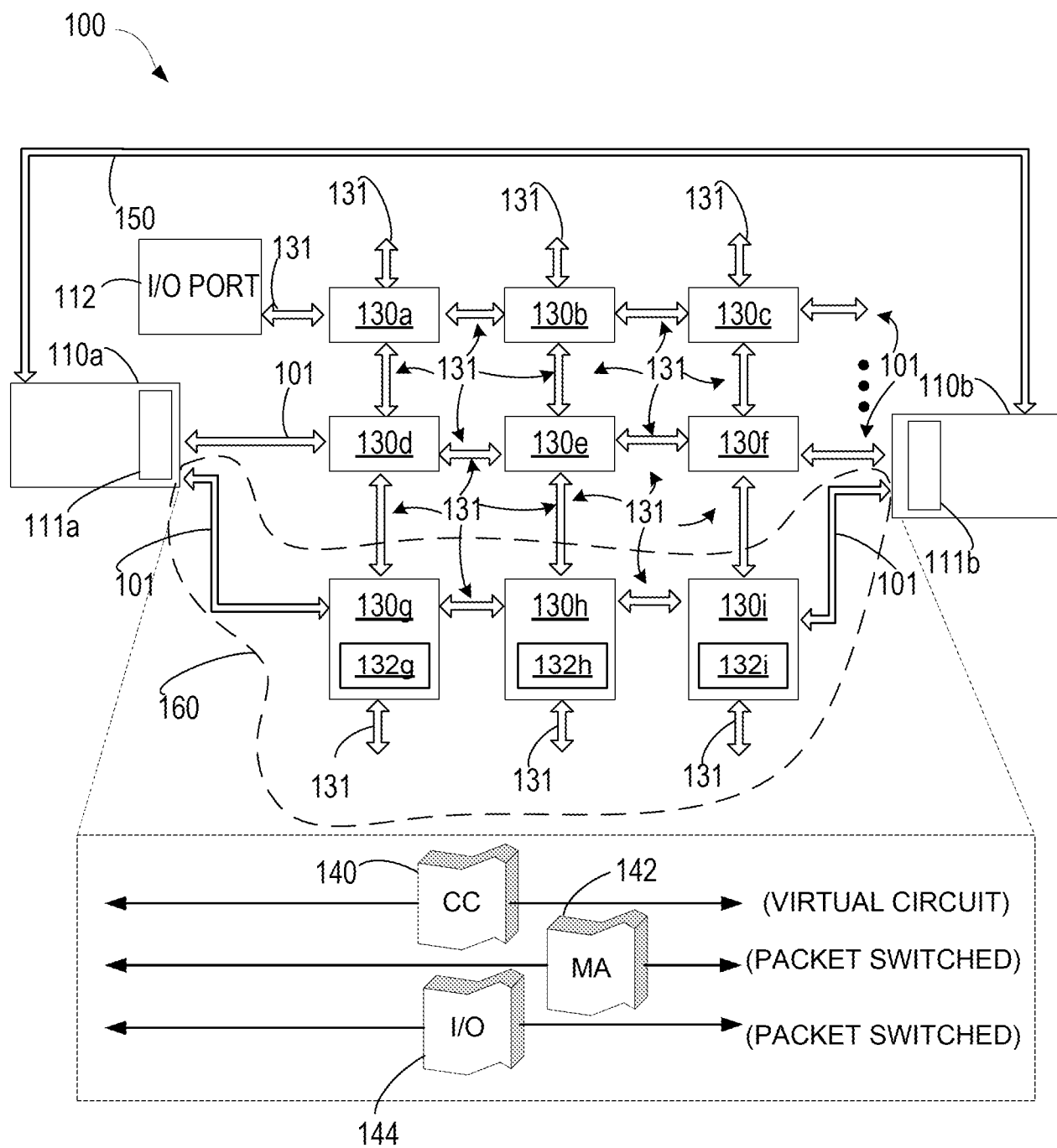


FIG. 1

2/5

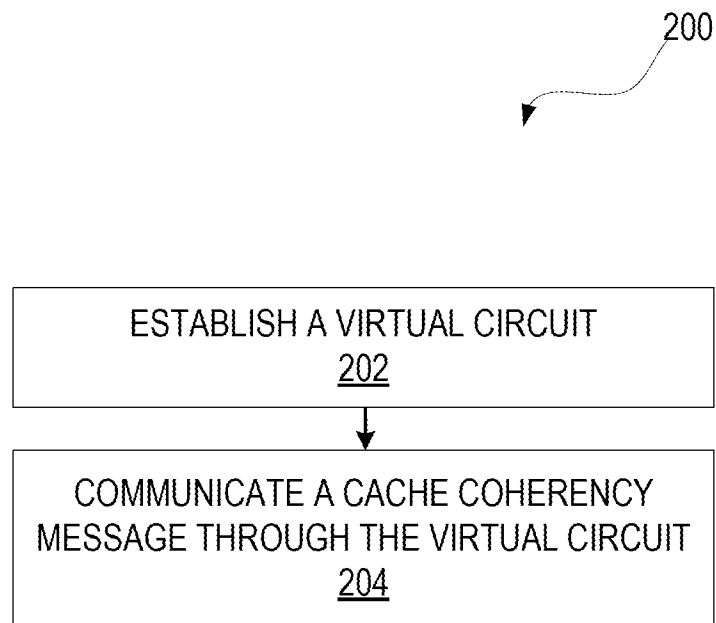
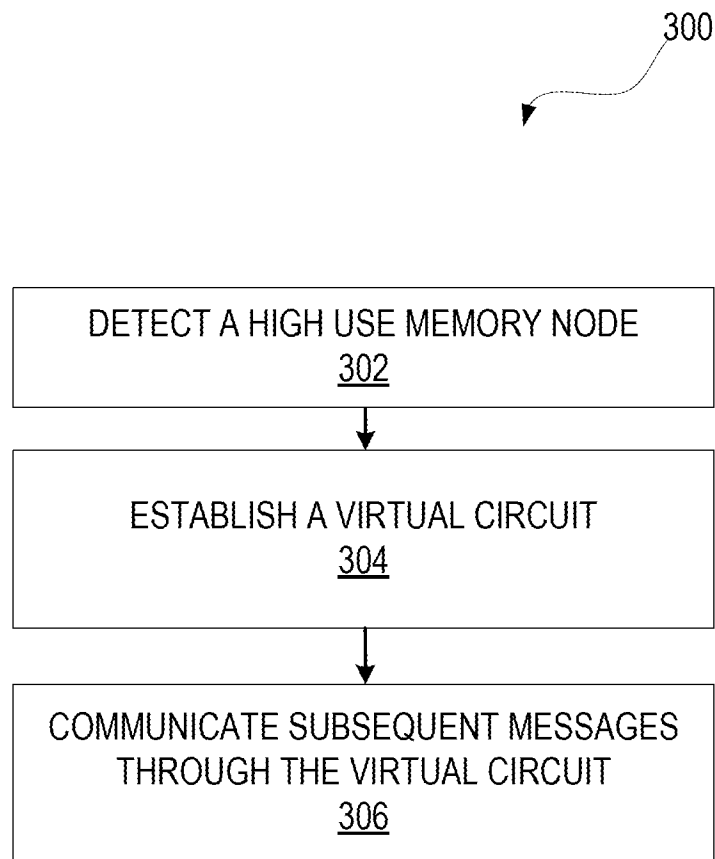


FIG. 2

3/5

*FIG. 3*

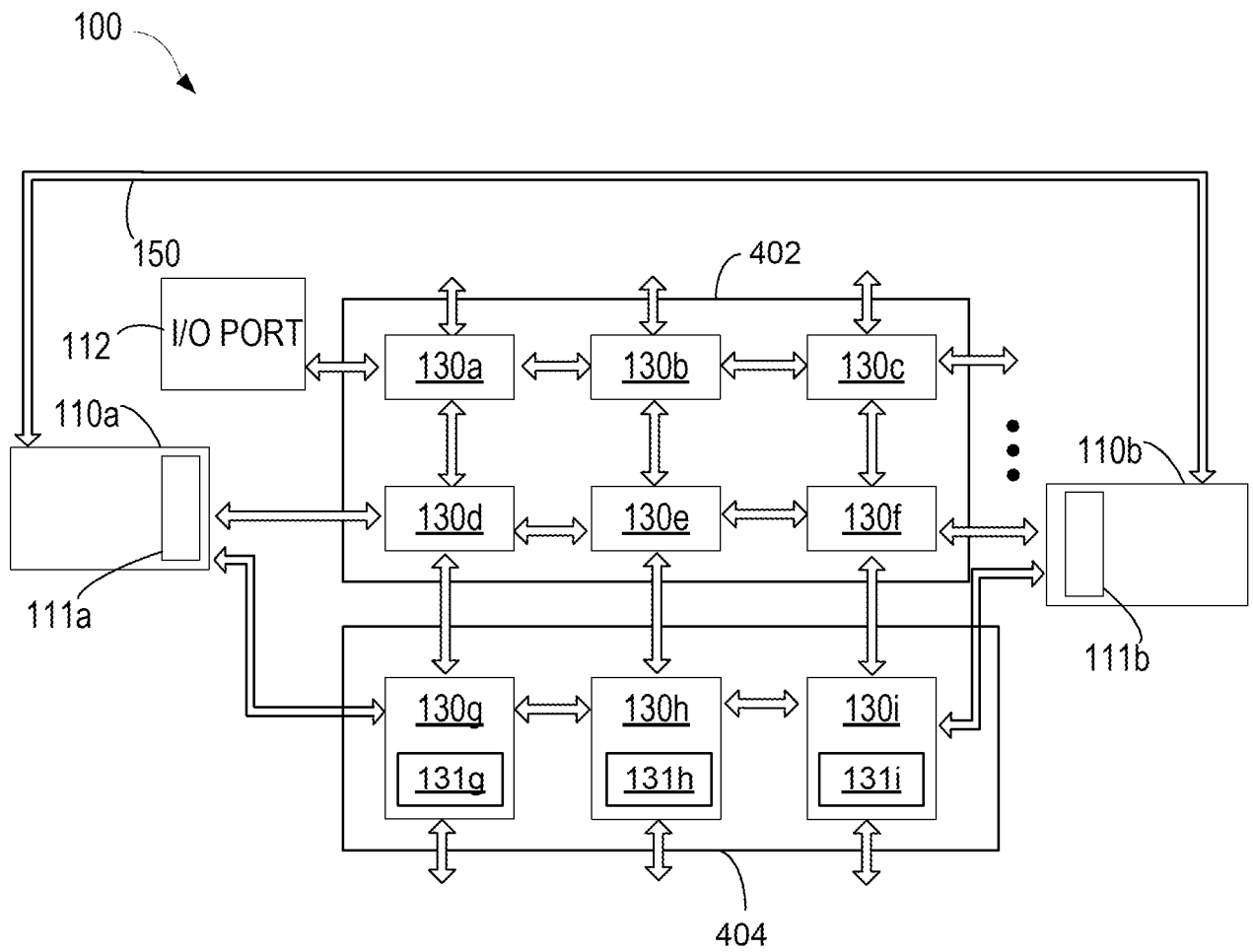


FIG. 4

5/5

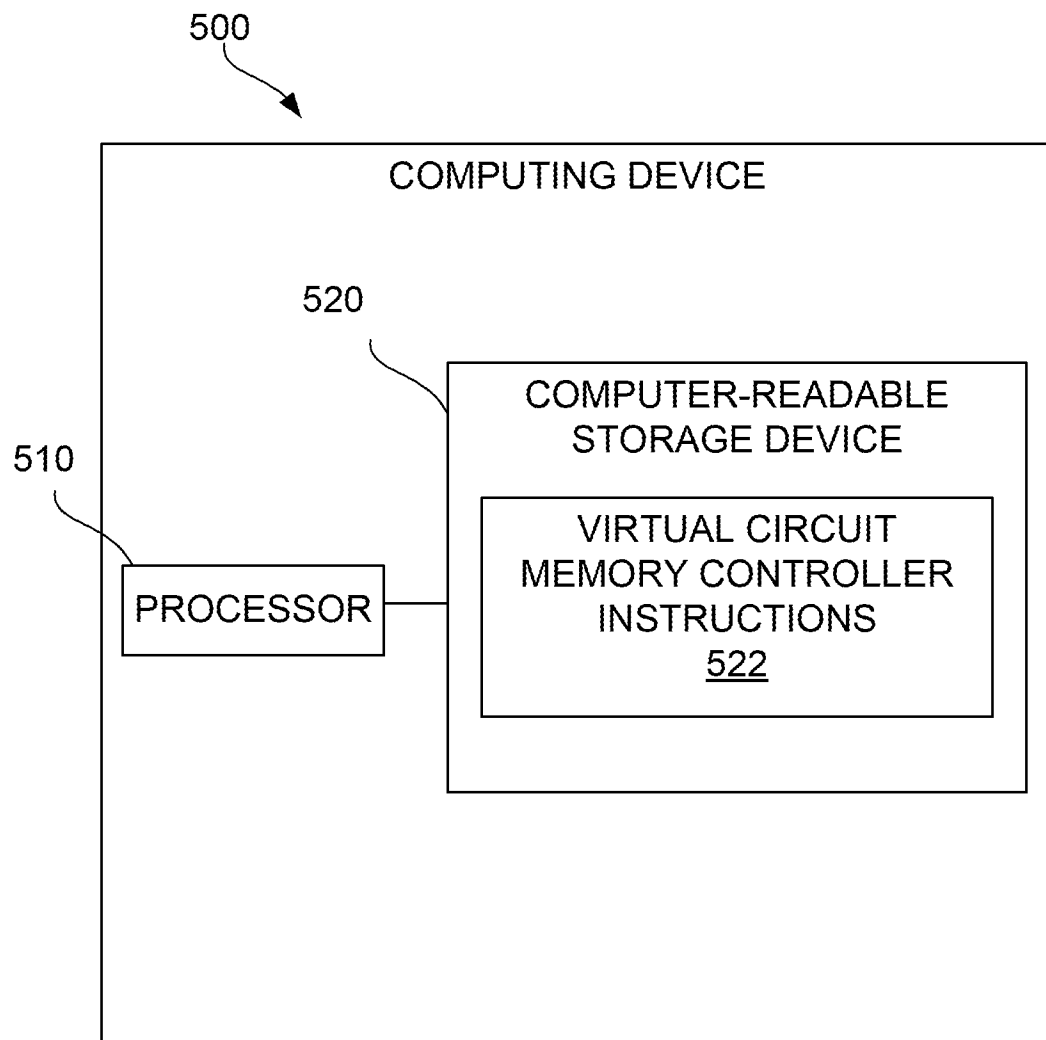


FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/047703**A. CLASSIFICATION OF SUBJECT MATTER****G06F 13/14(2006.01)i**

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 13/14; H04L 12/46; G06F 15/00; G06F 9/02; G06F 15/80; G06F 12/00; G06F 15/177; H04L 12/56

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: processor nodes, memory nodes, virtual circuit, point to point, and similar terms

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 98-18239 A1 (CASCADE COMMUNICATIONS CORPORATION) 30 April 1998 See page 3, lines 26-27; page 4, lines 2-3; and page 6, lines 5-6.	1-15
A	US 2004-0139297 A1 (JON M. HUPPENTHAL) 15 July 2004 See paragraphs [0015] and [0030]; and figure 3.	1-15
A	US 2012-0011323 A1 (SUNG-JAE BYUN et al.) 12 January 2012 See paragraphs [0012]-[0013] and [0042]; and figure 1A.	1-15
A	US 2011-0072237 A1 (GERALD GEORGE PECHANKE) 24 March 2011 See paragraphs [0011] and [0045]-[0052]; and figure 4A.	1-15
A	US 2007-0073993 A1 (KENNETH R. ALLEN et al.) 29 March 2007 See paragraphs [0011] and [0032]-[0038]; and figure 2.	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"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)

"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

"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

"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

"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

"&" document member of the same patent family

Date of the actual completion of the international search

28 February 2015 (28.02.2015)

Date of mailing of the international search report

02 March 2015 (02.03.2015)

Name and mailing address of the ISA/KR

International Application Division
Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City, 302-701,
Republic of Korea

Facsimile No. ++82 42 472 7140

Authorized officer

NHO, Ji Myong

Telephone No. +82-42-481-8528



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2014/047703

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 98-18239 A1	30/04/1998	AU 5003397 A	15/05/1998
US 2004-0139297 A1	15/07/2004	AU 2003-282507 A1 CA 2511812 A1 EP 1586041 A1 JP 2006-513489 A WO 2004-063934 A1	10/08/2004 29/07/2004 19/10/2005 20/04/2006 29/07/2004
US 2012-0011323 A1	12/01/2012	JP 2007-157150 A JP 2014-096173 A KR 10-1275752 B1 KR 10-2007-0059462 A KR 10-2009-0113610 A TW I358640 B US 2007-0136536 A1 US 2009-0210691 A1 US 2011-0119477 A1 US 7882344 B2 US 8209527 B2 US 8423755 B2	21/06/2007 22/05/2014 17/06/2013 12/06/2007 02/11/2009 21/02/2012 14/06/2007 20/08/2009 19/05/2011 01/02/2011 26/06/2012 16/04/2013
US 2011-0072237 A1	24/03/2011	US 2006-0218376 A1 US 2009-0265512 A1 US 2011-0161625 A1 US 7581079 B2 US 7886128 B2 US 8156311 B2 US 8443169 B2	28/09/2006 22/10/2009 30/06/2011 25/08/2009 08/02/2011 10/04/2012 14/05/2013
US 2007-0073993 A1	29/03/2007	CN 100538661 C CN 1940891 A	09/09/2009 04/04/2007