(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 July 2016 (28.07.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/118162 A1

(51) International Patent Classification:
G06F 11/30 (2006.01) *G06F 12/00* (2006.01)

(21) International Application Number:
PCT/US2015/012722

(22) International Filing Date:
23 January 2015 (23.01.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP** [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).

(72) Inventors: **GONZALEZ DIAZ, Maria Teresa**; 1501 Page Mill Rd., Palo Alto, California 94304-1100 (US). **LI, Jun**; 1501 Page Mill Rd., Palo Alto, California 94304-1100 (US). **RIVERA REYNA, Maria Janneth**; Av. Camino al Iteso 8270, Colonia el Mante Tlaquepaque, 45080 Jalisco (MX).

(74) Agents: **KIRCHEV, Ivan T.** et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79 , Fort Collins, CO 80528 (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, 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, ST, 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, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

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

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

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

(54) Title: NON-UNIFORM MEMORY ACCESS AWARE MONITORING

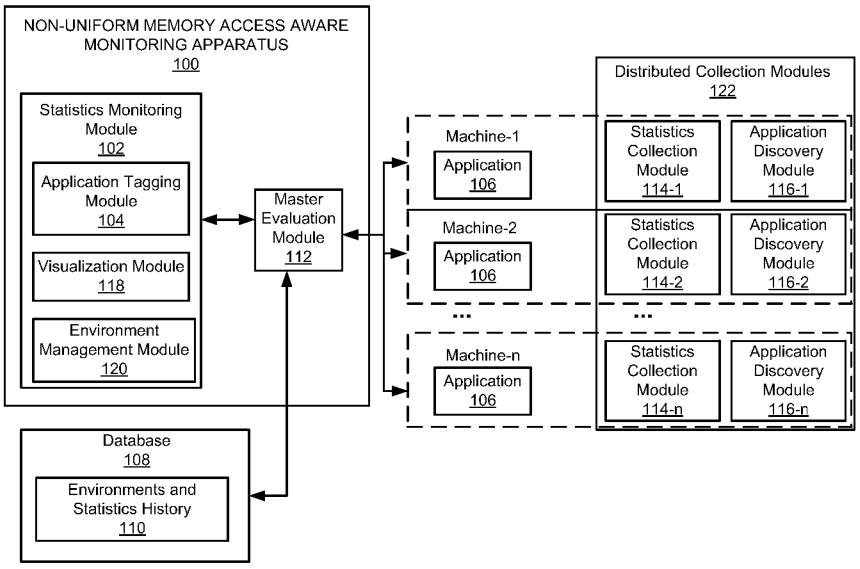

FIG. 1

(57) Abstract: According to an example, NUMA-aware monitoring may include identifying an application, and receiving statistics related to resource utilization. The statistics may be analyzed to determine application performance characteristics of the application.

WO 2016/118162 A1

NON-UNIFORM MEMORY ACCESS AWARE MONITORING

BACKGROUND

[0001] In computer systems, resource monitoring includes determining and displaying statistics related to the use of hardware (e.g., central processing unit (CPU), memory, disk, network, etc.) and software (e.g., machine readable instructions) resources. A computer system may include a resource monitoring application that is executed to generate statistics related to use of the hardware and software resources.

BRIEF DESCRIPTION OF DRAWINGS

[0002] Features of the present disclosure are illustrated by way of example and not limited in the following figure(s), in which like numerals indicate like elements, in which:

[0003] Figure 1 illustrates an architecture of a non-uniform memory access (NUMA) aware monitoring apparatus, according to an example of the present disclosure;

[0004] Figure 2 illustrates a method for NUMA-aware monitoring, according to an example of the present disclosure;

[0005] Figure 3 illustrates further details of the method for NUMA-aware monitoring, according to an example of the present disclosure;

[0006] Figure 4 illustrates further details of the method for NUMA-aware monitoring, according to an example of the present disclosure; and

[0007] Figure 5 illustrates a computer system, according to an example of the present disclosure.

DETAILED DESCRIPTION

[0008] For simplicity and illustrative purposes, the present disclosure is described by referring mainly to examples. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present disclosure. It will be readily apparent however, that the present disclosure may be practiced without limitation to these specific details. In other instances, some methods and structures have not been described in detail so as not to unnecessarily obscure the present disclosure.

[0009] Throughout the present disclosure, the terms "a" and "an" are intended to denote at least one of a particular element. As used herein, the term "includes" means includes but not limited to, the term "including" means including but not limited to. The term "based on" means based at least in part on.

[0010] With respect to relatively large datasets, data analytics includes the use of clusters of relatively high-end servers to provide fast in-memory processing of the datasets. The high-end servers may be characterized as including a relatively large amount of memory and a relatively large number of cores. These aspects related to high-end servers may expose non-uniform memory access (NUMA) architectures in which the cores are divided by sockets, and then each socket includes local memory banks. A core may include a logical execution unit including, for example, an L1 cache, an L2 cache, etc., and functional units. A processor socket (also referred to as a central processing unit (CPU) socket) may represent a connector on a motherboard that houses a CPU, and may form the electrical interface and contact with the CPU. A motherboard may include a plurality of sockets. Depending on the architectural specifications, high-end servers may include a different number of sockets, and each socket may include a different number of cores.

[0011] A process related to execution of an application (i.e., including machine readable instructions) may be bound to a specific socket, and then other memory banks (e.g., on other sockets) may become remote memory for the process. A

process may be described as a set of machine readable instructions related to an application. Understanding the NUMA effect for machines, such as high-end servers and other systems that encounter relatively large datasets can be challenging because the performance of an application may be impacted by data placement over memory banks across sockets. For example, if data is placed on remote memory banks (as opposed to a local memory bank), the remote data access may slow down data processing due to cache misses, which may impact the overall runtime performance. In this regard, monitoring of memory access and CPU utilization, and others such as traffic patterns on analytics workloads may be helpful for identifying, for example, bottlenecks, optimizing processes, and achieving overall higher system performance.

[0012] According to examples, a NUMA-aware monitoring apparatus and a method for NUMA-aware monitoring are disclosed herein. The apparatus and method disclosed herein may generally provide for NUMA-aware monitoring for distributed machines, and user-specific applications. Statistics related to the monitoring may be displayed via a statistics monitoring module as disclosed herein. The apparatus and method disclosed herein may identify a target application, and collect NUMA-aware resource utilization statistics related to execution of the target application. The NUMA-aware resource utilization statistics may include, for example, a core level, a NUMA node level, and/or an application level, where each of the NUMA-aware resource utilization statistics may be related to CPU and/or memory utilization for a plurality of machines that are being used by a particular application. The core level may represent the CPU and/or memory utilization for cores of a plurality of machines that are being used by the particular application. The NUMA node level may represent CPU and/or memory utilization for NUMA nodes of a plurality of machines that are being used by the particular application. The application level may represent the CPU and/or memory utilization, aggregated for the application across cores, NUMA nodes, and the set of machines that involve the application. For example, the NUMA-aware resource utilization

statistics may be used to determine core-level CPU utilization, NUMA node-level CPU utilization, NUMA node-level memory utilization, and/or application NUMA node-level memory utilization, that is NUMA node-level memory utilization across the entire application, each presented as a single view across multiple machines. The machines may include servers, and computer systems generally.

[0013] A NUMA node may include cores hosted on a socket and memory that is bound locally to these cores. In this regard, local memory may include memory that is on the same NUMA-node (that is, the socket) as the CPU currently executing the thread to access this memory. Further, remote memory may include any memory that does not belong to the NUMA node on which a thread is currently executing.

[0014] With respect to core-level CPU utilization, CPU utilization may refer to a computer's usage of processing resources, or the amount of work handled by a CPU. Actual CPU utilization may vary depending on the amount and type of managed computing tasks. Certain tasks may utilize a relatively greater amount of CPU time, while other tasks may utilize a relatively lesser amount of CPU time because of non-CPU resource needs, such as waiting for a disk based input/output (IO), communication IO, etc. In this regard, core-level CPU utilization may include utilization of the CPU resources measured at each core by an operating system monitoring tool such as collectl. For example, a computer system may include ten sockets, and eight cores for each socket. The operating system monitoring tool may be used to determine, for example, NUMA Node0: core0 – utilization 15%, NUMA Node0: core8 – utilization 45%, etc.

[0015] NUMA node-level CPU utilization may include the total core utilization aggregated by the cores that are hosted on the same NUMA node. For example, for a computer system including ten sockets, and eight cores for each socket, on a NUMA Node0, two cores, core0 and core8, may be currently active, and include the utilization of NUMA Node0: core0 – utilization 15%, and NUMA Node0: core8 – utilization 45%. In this case, the NUMA node-level CPU utilization may include

NUMA Node 0: 60% (which may be obtained by adding the utilizations of 15% and 45%). For a NUMA node, the NUMA node-level CPU utilization may be expressed as an absolute value or an average value (e.g., for NUMA Node0, the absolute NUMA node-level CPU utilization may be represented as 60%, and the average NUMA node-level CPU utilization may be represented as 60/8 = 7.5%).

[0016] NUMA node-level memory utilization may include memory used by all applications that are executing on a NUMA node. For example, for a computer system including ten sockets, and eight cores for each socket, Node0 may include 22GB of memory being consumed by all applications that are executing on the NUMA Node0.

[0017] Application NUMA node-level memory utilization may include memory used for a user-specific application by a NUMA node. For example, for a computer system including ten sockets, and eight cores for each socket, on NUMA Node0, an image search application may use 100GB of memory, and on NUMA Node1, the same image search application may use 120GB of memory.

[0018] The apparatus and method disclosed herein may provide for user application tags to identify which application is to be tracked, for example, based on a user selection. The statistics monitoring module may provide for the tagging of application instances, and thus processes related to the tagged application for tracking. Based on the identified application, the NUMA node related to the application may be identified for application-specific tracking. The apparatus and method disclosed herein may also provide for user-specific application monitoring supported by user tagging and discovery of processes related to an application.

[0019] The statistics monitoring module may be used to manage logical clusters in a common physical execution environment. For example, certain machines of a cluster may be used to execute a first application and other machines of the cluster may be used to execute a second application. While the machines in the cluster may be physically connected, they are also logically divided with respect to the first and second applications. Thus, in order to track a particular application, the

particular machines that are used to execute the particular application may be tracked.

[0020] The apparatus and method disclosed herein may provide a distributed view of NUMA node information, such as process level information for user-specific applications. For example, various processes related to an application may be executed on different machines. Thus, a distributed view of the processes related to the application may be provided with respect to different NUMA nodes.

[0021] The apparatus and method disclosed herein may provide for publishing of statistics using, for example, time series charts and heat maps in real time. For example, a heat map may represent CPU utilization, or memory utilization, at the NUMA node level. The statistics monitoring module may provide a history snapshot from past executions of an application (e.g., a tagged application), and comparison of the statistics, for example, by time window and/or specific machines. For example, the statistics monitoring module may provide for comparison of a past execution of an application with a current execution of the application for analysis of various configurations of machines that are used to execute the application. For example, an application may operate differently on one machine versus another machine. The statistics monitoring module may provide for the analysis of the statistics by NUMA node, core, and/or application, and time window.

[0022] The statistics monitoring module may provide application snapshots from history using heat maps to highlight workload bottlenecks and unbalance. For example, a heat map may identify busy versus idle timeframes for application utilization, and identify various cores that are busy versus idle during these timeframes.

[0023] Figure 1 illustrates an architecture of a NUMA-aware monitoring apparatus (hereinafter also referred to as "apparatus 100"), according to an example of the present disclosure. Referring to Figure 1, the apparatus 100 is depicted as including a statistics monitoring module 102 to generally provide for the monitoring of statistics over a plurality of machines *1-n*. The statistics monitoring

module 102 may include an application tagging module 104 to identify, and tag, an application 106 that is executed on the machines *1-n* (where the machines*1-n* may represent a set of machines on which the application 106 is executed from a plurality of machines, or alternatively, a set of machines to execute the application 106 may be selected from the machines *1-n*). The tag may be stored as part of an execution history of the application 106 in a database 108 as the environments and statistics history 110. A master evaluation module 112 may receive, from the set of machines *1-n*, NUMA-aware resource utilization statistics that include a core level, a NUMA node level, and/or an application level, where each of the NUMA-aware resource utilization statistics may be related to CPU and/or memory utilization for the set of machines *1-n.* With respect to NUMA node level, a NUMA node for a machine of the set of machines *1-n* may include cores hosted on a socket, and memory that is bound locally to the cores hosted on the socket. The application level may represent the CPU and/or memory utilization, aggregated for the application across cores, NUMA nodes, and the machines *1-n* that involve the application. For each particular machine *1-n*, a statistics collection module 114 may call an associated application discovery module 116 to provide for the tracking of the application 106. The master evaluation module 112 may analyze the NUMA-aware resource utilization statistics to generate, based on the operation of a visualization module 118 and/or an environment management module 120, a display of application performance characteristics of the application 106 on the set of machines *1-n*. Examples of application performance characteristics may include whether the application 106 includes uniform CPU utilization across the NUMA nodes, whether the application 106 is memory-demanding, that is, the application major tasks are characterized by demanding a large amount of memory allocation, whether the application 106 is CPU bound, that is, the application major tasks are characterized by needing a large amount of CPU computations on many cores, whether the scheduler for the application 106 provides adequate balance of CPU utilization/memory utilization across a NUMA node and NUMA nodes, etc. The environment management module 120 may also generate a view of previous

execution snapshots of a single analytics application execution, and comparison on the same application 106 that has different inputs or different configuration parameters, on the same execution environment being monitored.

[0024] The modules and other elements of the apparatus 100 may be machine readable instructions stored on a non-transitory computer readable medium. In this regard, the apparatus 100 may include or be a non-transitory computer readable medium. In addition, or alternatively, the modules and other elements of the apparatus 100 may be hardware or a combination of machine readable instructions and hardware.

[0025] The application tagging module 104 may tag a user application (e.g., the application 106) that is to be executed on machines selected from the machines *1-n*. A user may add a tag to the master evaluation module 112, for example, as a name of an application. For example, a tag may be represented as "app, algorithmA, classifier_*1", where the algorithmA name may be related to the application 106 that may include machine readable instructions. The tag may be sent to the master evaluation module 112 to track the associated application that is being executed on machines selected from the machines *1-n*. The tag may be stored as part of an execution history of the application in the database 108 as the environments and statistics history 110.

[0026] The visualization module 118 may provide (i.e., generate, as needed, to display), a view of the NUMA node statistics related to the tagged application 106 that may be executed on the machines *1-n*. The visualization module 118 may generate a time-series view, for example, of the core and NUMA node level statistics across a plurality of the machines *1-n* in a cluster environment. Further, the visualization module 118 may generate a real-time display of the core and NUMA node level statistics. The visualization module 118 may also generate various heat map views. For example, the visualization module 118 may generate a heat map view of core-level CPU utilization, NUMA node-level CPU utilization, NUMA node-level memory utilization, and/or application NUMA node-level memory

utilization, and identify any bottlenecks and unbalanced processing. The heat map view may include options to zoom in or out of a time window, e.g., 10 minutes, 30 minutes, or 24 hours of history, etc.

[0027] The environment management module 120 may provide for the storage and analysis of history for applications that have been tagged. The history for the applications that have been tagged may be stored in the database 108. The environment management module 120 may also generate a view of previous execution snapshots of a single analytics application execution, and comparison on the same application that has different inputs or different configuration parameters, on the same execution environment being monitored. The environment management module 120 may manage logical groups of machines that are monitored in parallel. For example, a cluster A may be designated {machine1, machine2, machine3}. A cluster B may be designated {machine3, machine4, machine5}, where machine3 is shared in both cluster A and cluster B. This logical cluster grouping may facilitate the tracing of different applications that are executing in the same shared execution environments. For example cluster A may be executing on a first platform, while other applications may be executed using cluster B on a second platform. While machine3 may be physically shared on both platforms, the separated applications may be tracked for both cluster A and cluster B.

[0028] With respect to collection of statistics over the machines in the clusters, the master evaluation module 112 may identify the list of machines and clusters that are to be monitored, the type of statistics (e.g., memory usage, CPU utilization, etc.), and the name of the applications that are to be tracked. The master evaluation module 112 may use a data-pull mechanism to connect every *t* seconds to the remote machines *1-n*, execute the statistics collection modules 114 (on the appropriate machines *1-n*) to ascertain statistics related to a tagged application, and retrieve the statistics. Once all statistics are received, the master evaluation module 112 may forward the information to the visualization module 118 to display

appropriate charts for users. The collected statistics may be stored in the database 108 as the environments and statistics history 110 for time tracking. The master evaluation module 112 may provide parallel threads to simultaneously monitor the machines *1-n* within the same logical cluster.

[0029] Distributed collection modules 122 (including the statistics collection modules 114 and the application discovery modules 116) that gather statistics from the remote machines *1-n* may be implemented as components of the apparatus 100 or separately from the apparatus 100 as illustrated in Figure 1. The distributed collection modules 122 may operate as agent programs (i.e., machine readable instructions) on the remote machines *1-n*, or otherwise receive statistics data from the remote machines *1-n*. For each particular machine *1-n*, the statistics collection module 114 may call the associated application discovery module 116 to provide for the tracking of an application (e.g., the tagged application 106). The specific application that is tracked may be requested by a user, for example, by tagging. The discovery process for the application discovery module 116 may include identifying a process identifier (PID) by matching similar process names provided by the user, discovering the NUMA node bound for the PID, and linking the PID and the NUMA node (e.g., process1-node0 to a process monitoring list). Once a new PID is identified to match the user monitoring criteria on a local machine, the CPU utilization may be obtained for related cores of the NUMA node, and further, the memory usage for each application on each socket (or NUMA node) may be obtained.

[0030] The database 108 may be used to store statistics at various levels of granularity. For example, the levels of granularity may include global and user-specific statistics. Global statistics may capture overall statistics of the machines *1-n* without linking to specific applications, such as total memory usage at NUMA node0. User-specific statistics may be linked to workload identifiers in order to generate snapshots for a user, such as memory usage for Algorithm1 at NUMA node0. A logical catalog for available execution environments as a list of logical

clusters and the related physical machines may also be maintained in the database 108. For example, the logical catalog for available execution environments may include a plurality of clusters that are executing various applications.

[0031] Figures 2-4 respectively illustrate flowcharts of methods 200, 300, and 400 for NUMA-aware monitoring, corresponding to the example of the NUMA-aware monitoring apparatus 100 whose construction is described in detail above. The methods 200, 300, and 400 may be implemented on the NUMA-aware monitoring apparatus 100 with reference to Figure 1 by way of example and not limitation. The methods 200, 300, and 400 may be practiced in other apparatus.

[0032] Referring to Figure 2, for the method 200, at block 202, the method may include identifying an application including processes, where each process of the processes may be executed on a different machine of a set of machines. For example, an application may include multiple processes in execution on the same machine or on different machines. For example, referring to Figure 1, the application tagging module 104 may identify, and tag, the application 106 that is executed on the machines *1-n* (where the machines *1-n* may represent a set of machines on which the application 106 is executed from a plurality of machines, or alternatively, a set of machines to execute the application 106 may be selected from the machines *1-n*).

[0033] At block 204, the method may include receiving, from the set of machines, NUMA-aware resource utilization statistics that include a core level, a NUMA node level, and/or an application level, where each of the NUMA-aware resource utilization statistics are related to CPU and/or memory utilization for the set of machines. With respect to NUMA node level, a NUMA node for a machine of the set of machines may include cores hosted on a socket, and memory that is bound locally to the cores hosted on the socket. The application level may represent the CPU and/or memory utilization, aggregated for the application across cores, NUMA nodes, and the set of machines that involve the application. For example, referring to Figure 1, the master evaluation module 112 may receive, from

the set of machines *1-n*, NUMA-aware resource utilization statistics that include a core level, a NUMA node level, and/or an application level, where each of the NUMA-aware resource utilization statistics may be related to CPU and/or memory utilization for the set of machines *1-n.*

[0034] At block 206, the method may include analyzing the NUMA-aware resource utilization statistics to determine application performance characteristics of the application on the set of machines. For example, referring to Figure 1, the master evaluation module 112 may analyze the NUMA-aware resource utilization statistics to determine application performance characteristics of the application 106 on the set of machines *1-n*. Examples of application performance characteristics may include whether the application 106 includes uniform CPU utilization across the NUMA nodes, whether the application 106 is memory-demanding, that is, the application major tasks are characterized by demanding a large amount of memory allocation, whether the application 106 is CPU bound, that is, the application major tasks are characterized by needing a large amount of CPU computations on many cores, whether the scheduler for the application 106 provides adequate balance of CPU utilization/memory utilization across a NUMA node and NUMA nodes, etc.

[0035] At block 208, the method may include generating a display of the application performance characteristics of the application on the set of machines. For example, referring to Figure 1, the visualization module 118 and/or the environment management module 120 may generate a display of the application performance characteristics of the application 106 on the set of machines *1-n*.

[0036] According to an example, for the method 200, generating the display of the application performance characteristics of the application on the set of machines may further include generating a time series chart and/or a heat map of the application performance characteristics of the application on the set of machines.

[0037] According to an example, for the method 200, analyzing the NUMA-

aware resource utilization statistics to determine the application performance characteristics of the application on the set of machines may further include analyzing the NUMA-aware resource utilization statistics to determine, for each machine of the set of machines, core-level CPU utilization that includes utilization of a CPU of each machine of the set of machines measured at each core of each respective machine of the set of machines.

[0038] According to an example, for the method 200, analyzing the NUMA-aware resource utilization statistics to determine the application performance characteristics of the application on the set of machines may further include analyzing the NUMA-aware resource utilization statistics to determine, for each machine of the set of machines, NUMA node-level CPU utilization that includes a total core utilization aggregated by cores that are hosted on a same NUMA node for each machine of the set of machines.

[0039] According to an example, for the method 200, analyzing the NUMA-aware resource utilization statistics to determine the application performance characteristics of the application on the set of machines may further include analyzing the NUMA-aware resource utilization statistics to determine, for each machine of the set of machines, NUMA node-level memory utilization that includes memory used by all applications including the application that are executing on the NUMA node.

[0040] According to an example, for the method 200, analyzing the NUMA-aware resource utilization statistics to determine the application performance characteristics of the application on the set of machines may further include analyzing the NUMA-aware resource utilization statistics to determine, for each machine of the set of machines, application NUMA node-level memory utilization that includes memory used for the application by the NUMA node.

[0041] According to an example, for the method 200, identifying the application including processes may further include tagging the application (e.g., by using the application tagging module 104) from a plurality of applications to identify the

application.

[0042] According to an example, the method 200 may further include generating a history view (e.g., by using the visualization module 118) that compares past executions of the application on the set of machines to a current execution of the application on the set of machines.

[0043] According to an example, the method 200 may further include generating a history view (e.g., by using the visualization module 118) that compares past executions of the application on a different set of machines to a current execution of the application on the set of machines.

[0044] Referring to Figure 3, for the method 300, at block 302, the method may include identifying an application that is executed on a plurality of clusters of machines. Two or more clusters of the plurality of clusters of the machines include at least one common machine of the machines. For example, referring to Figure 1, the environment management module 120 may manage logical groups of machines that are monitored in parallel. For example, a cluster A may be designated {machine1, machine2, machine3}. A cluster B may be designated {machine3, machine4, machine5}, where machine3 is shared in both cluster A and cluster B. This logical cluster grouping may facilitate the tracing of different applications that are executing in the same shared execution environments. For example cluster A may be executing on a first platform, while other applications may be executed using cluster B on a second platform. While machine3 may be physically shared on both platforms, the separated applications may be tracked for both cluster A and cluster B.

[0045] At block 304, the method may include receiving, from the plurality of clusters of the machines, NUMA aware NUMA-aware resource utilization statistics that include a core level, a NUMA node level, and/or an application level, where each of the NUMA-aware resource utilization statistics are related to resource utilization for the plurality of clusters of the machines. For example, referring to Figure 1, the master evaluation module 112 may receive, from the plurality of

clusters of the machines, NUMA-aware resource utilization statistics that include a core level, a NUMA node level, and/or an application level, where each of the NUMA-aware resource utilization statistics may be related to resource utilization for the plurality of clusters of the machines.

[0046] At block 306, the method may include analyzing the NUMA-aware resource utilization statistics to determine application performance characteristics of the application on the plurality of clusters of the machines. For example, referring to Figure 1, the master evaluation module 112 may analyze the NUMA-aware resource utilization statistics to determine application performance characteristics of the application 106 on the plurality of clusters of the machines.

[0047] According to an example, for the method 300, identifying the application that is executed on the plurality of clusters of machines may further include identifying a NUMA node that corresponds to a process identifier (PID) corresponding to the application.

[0048] Referring to Figure 4, for the method 400, at block 402, the method may include identifying an application including processes, where each process of the processes may be executed on a different machine of a set of machines, wherein each different machine includes a core and a socket. For example, referring to Figure 1, the application tagging module 104 may identify, and tag, the application 106 that is executed on the machines *1-n* that each includes a core and a socket.

[0049] At block 404, the method may include receiving, from the set of machines, NUMA-aware resource utilization statistics related to resource utilization for the set of machines. For example, referring to Figure 1, the master evaluation module 112 may receive, from the set of machines, NUMA-aware resource utilization statistics related to resource utilization for the set of machines.

[0050] At block 406, the method may include analyzing the NUMA-aware resource utilization statistics to determine application performance characteristics of the application on the set of machines. The application performance

characteristics of the application may include a history view that compares past executions of the application on the set of machines or on a different set of machines to a current execution of the application on the set of machines. For example, referring to Figure 1, the master evaluation module 112 may analyze the NUMA-aware resource utilization statistics to determine application performance characteristics of the application on the set of machines.

[0051] According to an example, the method 400 may include organizing machines from the set of machines in a logical catalog, and tracking execution and the application performance characteristics of the application according to the logical catalog. For example, a logical catalog for available execution environments as a list of logical clusters and the related physical machines may also be maintained in the database 108. For example, the logical catalog for available execution environments may include a plurality of clusters that are executing various applications.

[0052] Figure 5 shows a computer system 500 that may be used with the examples described herein. The computer system 500 may represent a generic platform that includes components that may be in a server or another computer system. The computer system 500 may be used as a platform for the apparatus 100. The computer system 500 may execute, by a processor (e.g., a single or multiple processors) or other hardware processing circuit, the methods, functions and other processes described herein. These methods, functions and other processes may be embodied as machine readable instructions stored on a computer readable medium, which may be non-transitory, such as hardware storage devices (e.g., RAM (random access memory), ROM (read only memory), EPROM (erasable, programmable ROM), EEPROM (electrically erasable, programmable ROM), hard drives, and flash memory).

[0053] The computer system 500 may include a processor 502 that may implement or execute machine readable instructions performing some or all of the methods, functions and other processes described herein. Commands and data

from the processor 502 may be communicated over a communication bus 504. The computer system may also include a main memory 506, such as a random access memory (RAM), where the machine readable instructions and data for the processor 502 may reside during runtime, and a secondary data storage 508, which may be non-volatile and stores machine readable instructions and data. The memory and data storage are examples of computer readable mediums. The memory 506 may include a NUMA-aware monitoring module 520 including machine readable instructions residing in the memory 506 during runtime and executed by the processor 502. The NUMA-aware monitoring module 520 may include the modules of the apparatus 100 shown in Figure 1.

[0054] The computer system 500 may include an I/O device 510, such as a keyboard, a mouse, a display, etc. The computer system may include a network interface 512 for connecting to a network. Other known electronic components may be added or substituted in the computer system.

[0055] What has been described and illustrated herein is an example along with some of its variations. The terms, descriptions and figures used herein are set forth by way of illustration only and are not meant as limitations. Many variations are possible within the spirit and scope of the subject matter, which is intended to be defined by the following claims -- and their equivalents -- in which all terms are meant in their broadest reasonable sense unless otherwise indicated.

What is claimed is:

1. A method for non-uniform memory access (NUMA) aware monitoring, the method comprising:

identifying an application including processes, wherein each process of the processes is executed on a different machine of a set of machines;

receiving, from the set of machines, NUMA-aware resource utilization statistics that include at least one of a core level, a NUMA node level, and an application level, wherein each of the NUMA-aware resource utilization statistics are related to at least one of central processing unit (CPU) and memory utilization for the set of machines, wherein, for the NUMA node level, a NUMA node for a machine of the set of machines includes cores hosted on a socket, and memory that is bound locally to the cores hosted on the socket, and wherein the application level represents the at least one of CPU and memory utilization aggregated for the application across cores, NUMA nodes, and the set of machines that involve the application;

analyzing, by a processor, the NUMA-aware resource utilization statistics to determine application performance characteristics of the application on the set of machines; and

generating a display of the application performance characteristics of the application on the set of machines.

2. The method of claim 1, wherein generating the display of the application performance characteristics of the application on the set of machines further comprises:

generating at least one of a time series chart and a heat map of the application performance characteristics of the application on the set of machines.

3. The method of claim 1, wherein analyzing the NUMA-aware resource utilization statistics to determine the application performance characteristics of the application on the set of machines further comprises:

analyzing the NUMA-aware resource utilization statistics to determine, for each machine of the set of machines, core-level CPU utilization that includes utilization of a CPU of each machine of the set of machines measured at each core of each respective machine of the set of machines.

4. The method of claim 1, wherein analyzing the NUMA-aware resource utilization statistics to determine the application performance characteristics of the application on the set of machines further comprises:

analyzing the NUMA-aware resource utilization statistics to determine, for each machine of the set of machines, NUMA node-level CPU utilization that includes a total core utilization aggregated by cores that are hosted on a same NUMA node for each machine of the set of machines.

5. The method of claim 1, wherein analyzing the NUMA-aware resource utilization statistics to determine the application performance characteristics of the application on the set of machines further comprises:

analyzing the NUMA-aware resource utilization statistics to determine, for each machine of the set of machines, NUMA node-level memory utilization that includes memory used by all applications including the application that are executing on the NUMA node.

6. The method of claim 1, wherein analyzing the NUMA-aware resource utilization statistics to determine the application performance characteristics of the application on the set of machines further comprises:

analyzing the NUMA-aware resource utilization statistics to determine, for each machine of the set of machines, application NUMA node-level memory utilization that includes memory used for the application by the NUMA node.

7. The method of claim 1, wherein identifying the application including processes further comprises:

tagging the application from a plurality of applications to identify the application.

8. The method of claim 1, further comprising:

generating a history view that compares past executions of the application on the set of machines to a current execution of the application on the set of machines.

9. The method of claim 1, further comprising:

generating a history view that compares past executions of the application on a different set of machines to a current execution of the application on the set of machines.

10. A non-uniform memory access (NUMA) aware monitoring apparatus comprising:

a processor; and

a memory storing machine readable instructions that when executed by the processor cause the processor to:

identify an application that is executed on a plurality of clusters of machines, wherein at least two clusters of the plurality of clusters of the machines

include at least one common machine of the machines;

receive, from the plurality of clusters of the machines, NUMA-aware resource utilization statistics that include at least one of a core level, a NUMA node level, and an application level, wherein each of the NUMA-aware resource utilization statistics are related to resource utilization for the plurality of clusters of the machines, wherein, for the NUMA node level, a NUMA node includes cores hosted on a socket, and memory that is bound locally to the cores hosted on the socket, and wherein the application level represents the resource utilization aggregated for the application across cores, NUMA nodes, and the clusters of machines that involve the application; and

analyze the NUMA-aware resource utilization statistics to determine application performance characteristics of the application on the plurality of clusters of the machines.

11. The NUMA-aware monitoring apparatus according to claim 10, wherein the resource utilization for the plurality of clusters of the machines further comprises:

at least one of central processing unit (CPU) and memory utilization for the machines.

12. The NUMA-aware monitoring apparatus according to claim 10, wherein the machine readable instructions to identify the application that is executed on the plurality of clusters of machines further comprise instructions to:

identify a NUMA node that corresponds to a process identifier corresponding to the application.

13. A non-transitory computer readable medium having stored thereon machine readable instructions to provide non-uniform memory access (NUMA) aware

monitoring, the machine readable instructions, when executed, cause a processor to:

identify an application including processes, wherein each process of the processes is executed on a different machine of a set of machines, wherein each different machine includes a core and a socket;

receive, from the set of machines, non-uniform NUMA-aware resource utilization statistics related to resource utilization for the set of machines; and

analyze the NUMA-aware resource utilization statistics to determine application performance characteristics of the application on the set of machines, wherein the application performance characteristics of the application include a history view that compares past executions of the application on the set of machines or on a different set of machines to a current execution of the application on the set of machines.

14. The non-transitory computer readable medium according to claim 13, wherein the machine readable instructions to receive, from the set of machines, NUMA-aware resource utilization statistics related to resource utilization for the set of machines further comprise instructions to:

receive, from the set of machines, the NUMA-aware resource utilization statistics that include at least one of a core level, a NUMA node level, and an application level, wherein each of the NUMA-aware resource utilization statistics are related to at least one of central processing unit (CPU) and memory utilization for the set of machines, wherein, for the NUMA node level, a NUMA node includes cores hosted on the socket, and memory that is bound locally to the cores hosted on the socket, and wherein the application level represents the at least one of CPU and memory utilization aggregated for the application across cores, NUMA nodes, and the set of machines that involve the application.

15. The non-transitory computer readable medium according to claim 13, further comprising machine readable instructions to:

organize machines from the set of machines in a logical catalog; and

track execution and the application performance characteristics of the application according to the logical catalog.

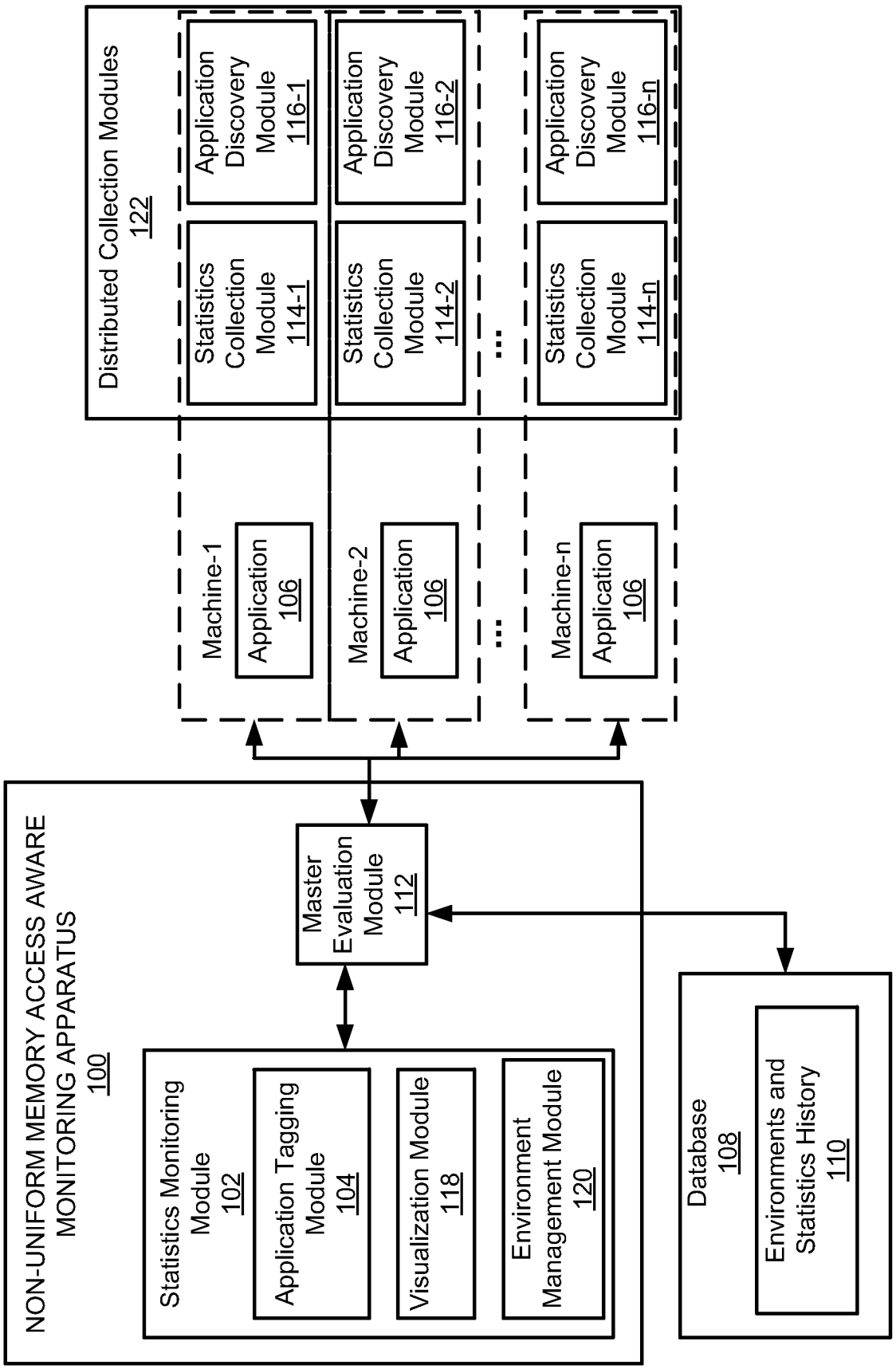
NON-UNIFORM MEMORY ACCESS AWARE MONITORING APPARATUS 100
Statistics Monitoring Module 102
Application Tagging Module 104
Visualization Module 118
Environment Management Module 120
Master Evaluation Module 112
Database 108
Environments and Statistics History 110
Distributed Collection Modules 122
Statistics Collection Module 114-1
Application Discovery Module 116-1
Statistics Collection Module 114-2
Application Discovery Module 116-2
Statistics Collection Module 114-n
Application Discovery Module 116-n
Machine-1
Application 106
Machine-2
Application 106
Machine-n
Application 106

FIG. 1

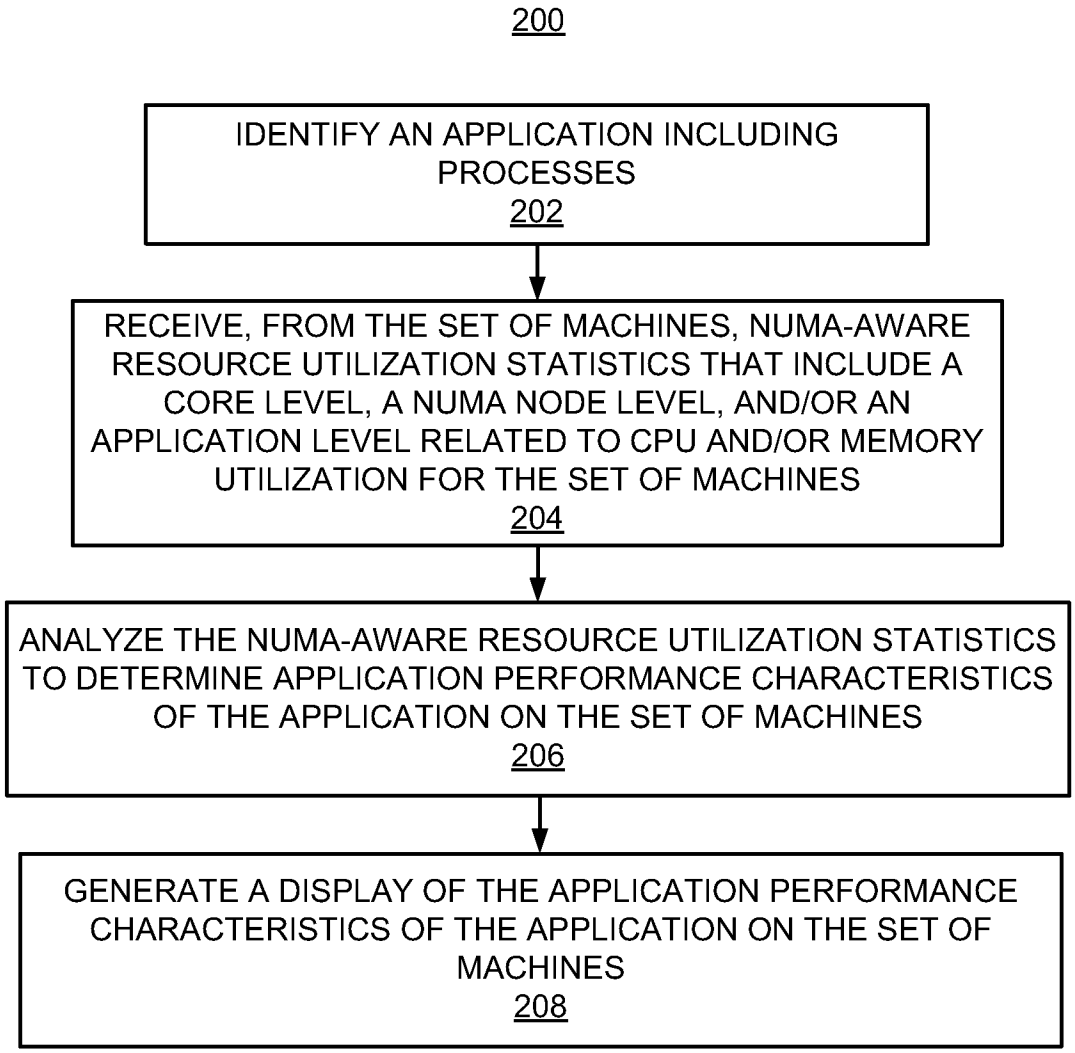
200
IDENTIFY AN APPLICATION INCLUDING PROCESSES
202
RECEIVE, FROM THE SET OF MACHINES, NUMA-AWARE RESOURCE UTILIZATION STATISTICS THAT INCLUDE A CORE LEVEL, A NUMA NODE LEVEL, AND/OR AN APPLICATION LEVEL RELATED TO CPU AND/OR MEMORY UTILIZATION FOR THE SET OF MACHINES
204
ANALYZE THE NUMA-AWARE RESOURCE UTILIZATION STATISTICS TO DETERMINE APPLICATION PERFORMANCE CHARACTERISTICS OF THE APPLICATION ON THE SET OF MACHINES
206
GENERATE A DISPLAY OF THE APPLICATION PERFORMANCE CHARACTERISTICS OF THE APPLICATION ON THE SET OF MACHINES
208

FIG. 2

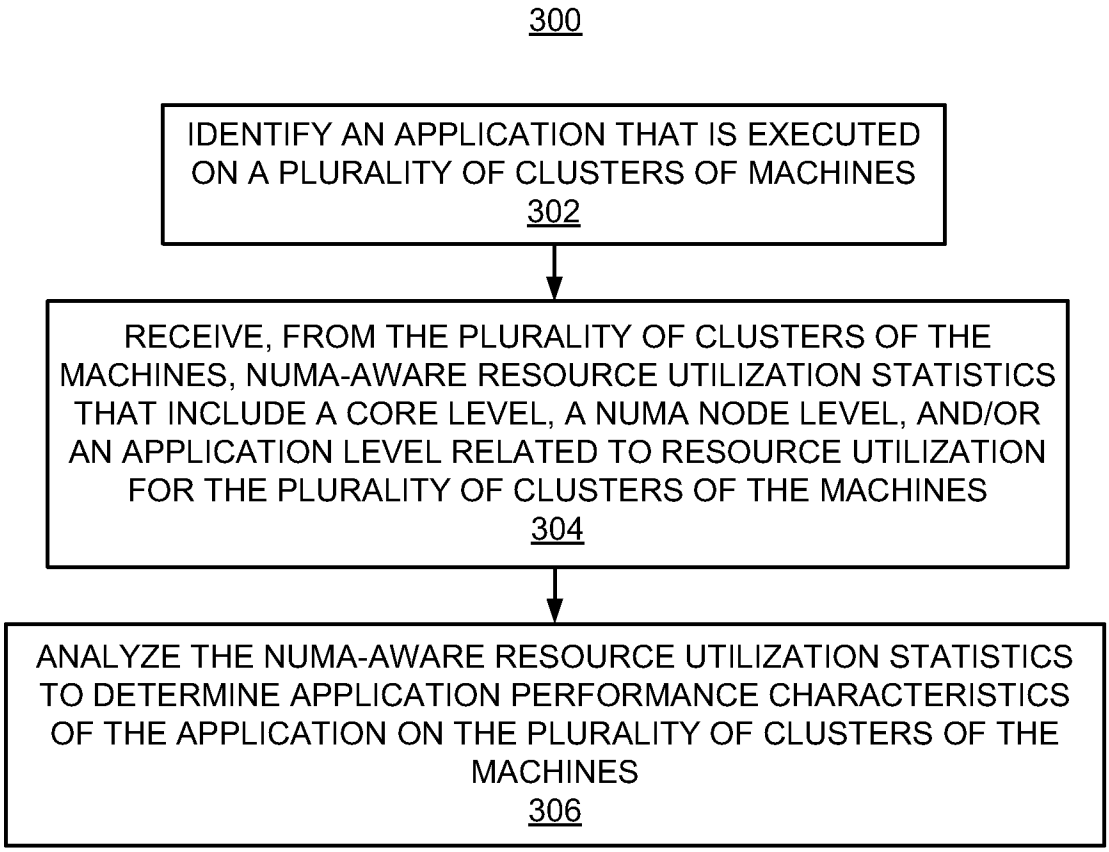
300
IDENTIFY AN APPLICATION THAT IS EXECUTED ON A PLURALITY OF CLUSTERS OF MACHINES
302
RECEIVE, FROM THE PLURALITY OF CLUSTERS OF THE MACHINES, NUMA-AWARE RESOURCE UTILIZATION STATISTICS THAT INCLUDE A CORE LEVEL, A NUMA NODE LEVEL, AND/OR AN APPLICATION LEVEL RELATED TO RESOURCE UTILIZATION FOR THE PLURALITY OF CLUSTERS OF THE MACHINES
304
ANALYZE THE NUMA-AWARE RESOURCE UTILIZATION STATISTICS TO DETERMINE APPLICATION PERFORMANCE CHARACTERISTICS OF THE APPLICATION ON THE PLURALITY OF CLUSTERS OF THE MACHINES
306

FIG. 3

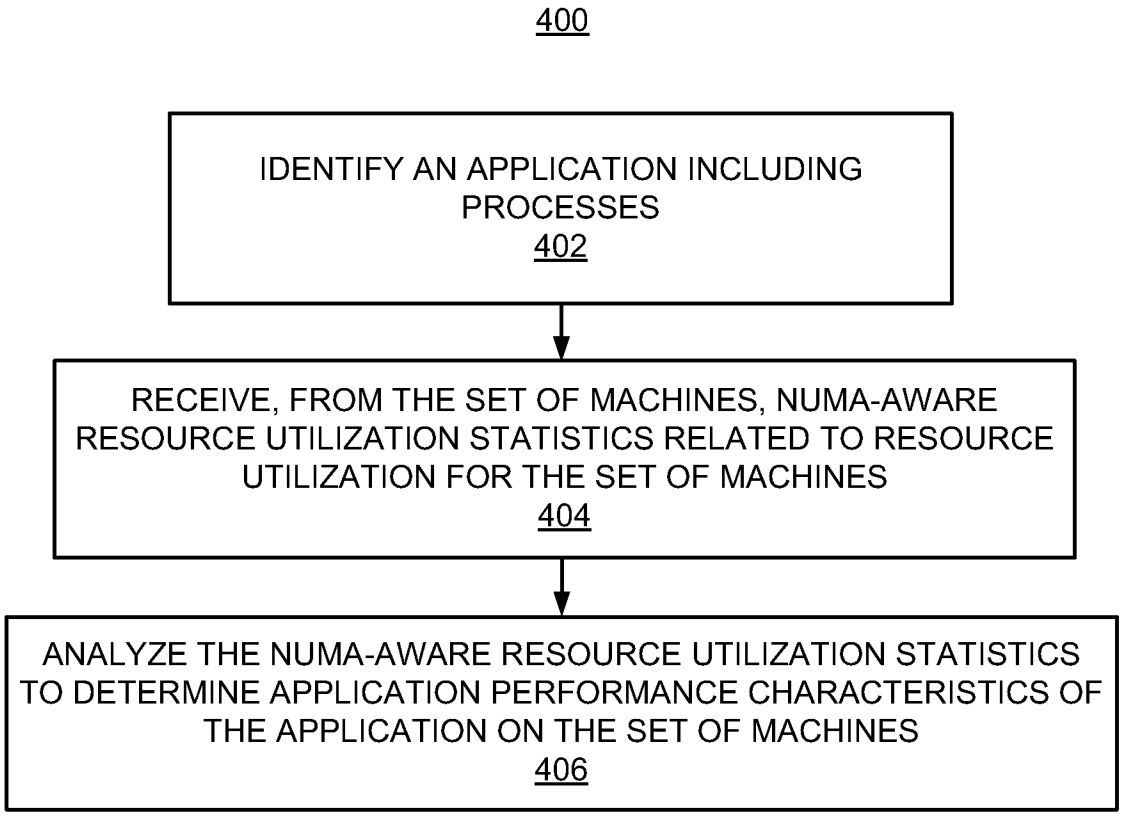
400
IDENTIFY AN APPLICATION INCLUDING PROCESSES
402
RECEIVE, FROM THE SET OF MACHINES, NUMA-AWARE RESOURCE UTILIZATION STATISTICS RELATED TO RESOURCE UTILIZATION FOR THE SET OF MACHINES
404
ANALYZE THE NUMA-AWARE RESOURCE UTILIZATION STATISTICS TO DETERMINE APPLICATION PERFORMANCE CHARACTERISTICS OF THE APPLICATION ON THE SET OF MACHINES
406

FIG. 4

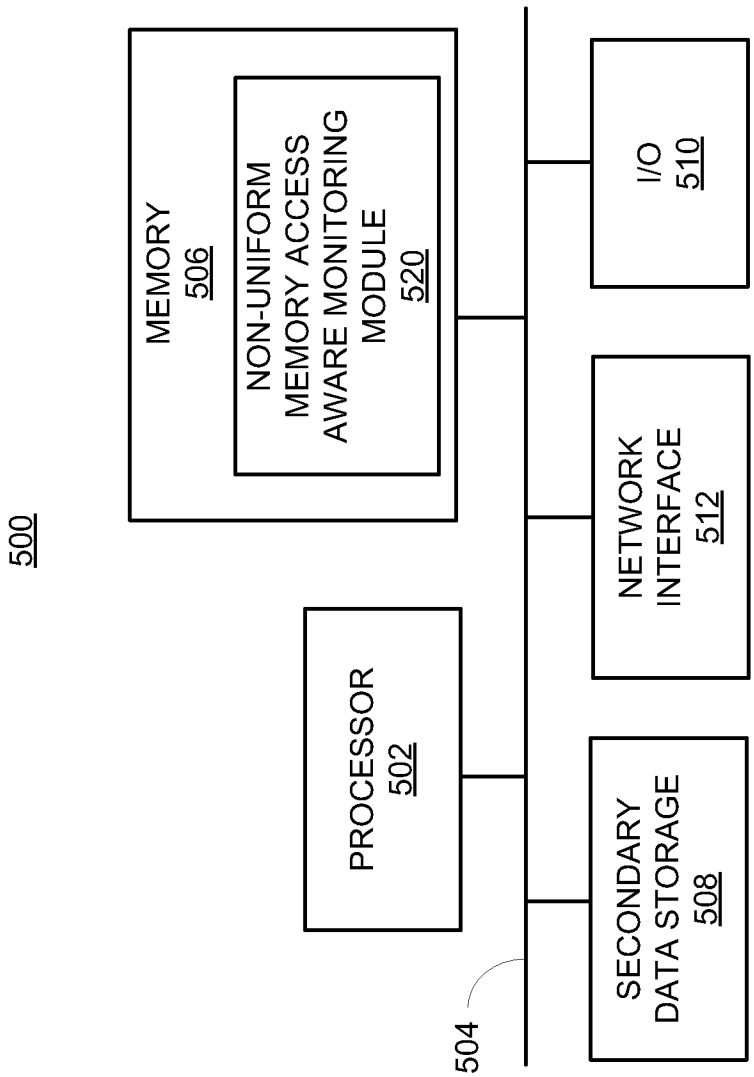
500
MEMORY
506
NON-UNIFORM
MEMORY ACCESS
AWARE MONITORING
MODULE
520
I/O
510
PROCESSOR
502
NETWORK
INTERFACE
512
SECONDARY
DATA STORAGE
508
504

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/012722

A. CLASSIFICATION OF SUBJECT MATTER

G06F 11/30(2006.01)i, G06F 12/00(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 11/30; G06F 13/40; G06F 3/00; G06F 9/46; G06F 1/26; G06F 9/50; G06F 12/00

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: non-uniform memory access (NUMA), monitoring, utilization, statistics, application, performance 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	US 2008-0229308 A1 (BLAIR WYMAN) 18 September 2008 See paragraphs [0015]-[0020], [0027], and [0047]; claim 1; and figures 1-2.	1-15
A	US 2008-0071939 A1 (TSUYOSHI TANAKA et al.) 20 March 2008 See paragraphs [0011]-[0017], [0159], and [0172]-[0181]; and figure 15.	1-15
A	US 2011-0202927 A1 (VLADIMIR MILOUSHEV et al.) 18 August 2011 See paragraphs [0311]-[0327] and figure 3.	1-15
A	WO 2012-036959 A1 (ORACLE INTERNATIONAL CORPORATION) 22 March 2012 See paragraphs [0002]-[0004] and [0047]; and figures 2 and 7.	1-15
A	US 2013-0246781 A1 (YONG QI et al.) 19 September 2013 See paragraphs [0023]-[0030] and figure 1.	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	Date of mailing of the international search report
31 August 2015 (31.08.2015)	**01 September 2015 (01.09.2015)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-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/US2015/012722

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2008-0229308 A1	18/09/2008	US 2006-0259704 A1	16/11/2006
		US 7383396 B2	03/06/2008
		US 7574567 B2	11/08/2009
US 2008-0071939 A1	20/03/2008	JP 2008-071237 A	27/03/2008
		JP 4839164 B2	21/12/2011
		US 7577770 B2	18/08/2009
US 2011-0202927 A1	18/08/2011	US 2006-0143350 A1	29/06/2006
		US 2009-0037585 A1	05/02/2009
		US 2014-207871 A1	24/07/2014
		US 7380039 B2	27/05/2008
		US 7934035 B2	26/04/2011
		US 8656077 B2	18/02/2014
WO 2012-036959 A1	22/03/2012	CN 103189844 A	03/07/2013
		CN 103189845 A	03/07/2013
		CN 103201722 A	10/07/2013
		CN 103210374 A	17/07/2013
		EP 2616934 A1	24/07/2013
		EP 2616935 A1	24/07/2013
		EP 2616936 A1	24/07/2013
		EP 2616937 A1	24/07/2013
		US 2012-0072621 A1	22/03/2012
		US 2012-0072622 A1	22/03/2012
		US 2012-0072624 A1	22/03/2012
		US 2012-0072627 A1	22/03/2012
		US 8725912 B2	13/05/2014
		US 8725913 B2	13/05/2014
		US 8782657 B2	15/07/2014
		US 8996756 B2	31/03/2015
		WO 2012-036898 A1	22/03/2012
		WO 2012-036960 A1	22/03/2012
		WO 2012-036961 A1	22/03/2012
US 2013-0246781 A1	19/09/2013	CN 103270470 A	28/08/2013
		JP 2014-507719 A	27/03/2014
		JP 5695766 B2	08/04/2015
		KR 10-1529016 B1	15/06/2015
		KR 10-2014-0064900 A	28/05/2014
		US 8671293 B2	11/03/2014
		WO 2013-040762 A1	28/03/2013