



(51) International Patent Classification:

H04L 29/08 (2006.01) *H04L 12/66* (2006.01)
H04L 12/933 (2013.01)

(21) International Application Number:

PCT/US2016/063593

(22) International Filing Date:

23 November 2016 (23.11.2016)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

14/998,308 26 December 2015 (26.12.2015) US

(71) Applicant (for all designated States except DJ, KW): **IN-TEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95054 (US).

(72) Inventors: **BAYON-MOLINO, Victor**; 197 Griffin Rath Hall, Celbridge Road, Maynooth (IE). **ESTRADA, Giovanni**; 148 Killala Road, Dublin (IE).

(74) Agent: **KELLETT, Glen M.**; Barnes & Thornburg LLP, c/o CPA Global, P.O. Box 52050, Minneapolis, Minnesota 55402 (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, DJ, 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, KW, 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 applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

— as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

Published:

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

(54) Title: TECHNOLOGIES FOR EXECUTION ACCELERATION IN HETEROGENEOUS ENVIRONMENTS

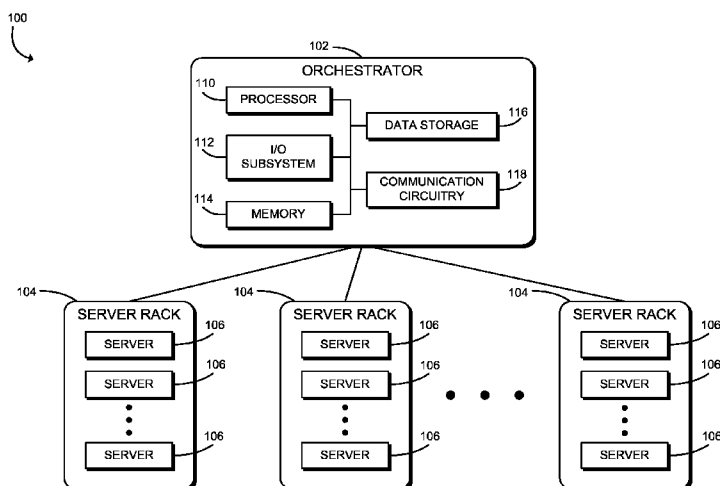


FIG. 1

(57) Abstract: Technologies for provisioning jobs to servers at a datacenter include a compute device to identify a class of each of a plurality of available servers in the datacenter, wherein the class is indicative of a set of hardware included on the corresponding server. The compute device selects a workload that includes a plurality of jobs to provision to a subset of the available servers and determines a provisioning configuration for the workload from a set of prospective configurations as a function of a rack utilization of each prospective configuration, the classes utilized by each prospective configuration, and a number of racks utilized by each prospective configuration. The provisioning configuration identifies a subset of the available servers to which to provision the jobs of the workload.

TECHNOLOGIES FOR EXECUTION ACCELERATION IN HETEROGENEOUS ENVIRONMENTS

CROSS-REFERENCE TO RELATED APPLICATION

[0001] The present application claims priority to U.S. Utility Patent Application Serial No. 14/998,308, entitled “TECHNOLOGIES FOR EXECUTION ACCELERATION IN HETEROGENEOUS ENVIRONMENTS,” which was filed on December 26, 2015.

BACKGROUND

[0002] High performance computing (HPC) and “big data” analytics generally involve the use of multiple servers in a datacenter to compute data faster and more efficiently than possible with standalone computing devices. However, typical “big data” and other high-throughput applications often require intimate knowledge of the execution environment, fine-tuning, and/or a static environment. Generally, workloads consisting of many jobs/tasks are provisioned to various computing resources (e.g., rack-based servers), which may be homogenous computing environments consisting, for example, only a particular class of servers or may be heterogeneous computing environments consisting of any number of different classes (e.g., different architectures and/or server generations) of servers. In some systems, virtualization and/or cloud abstraction layers may present heterogeneous infrastructures as homogeneous infrastructures but, in doing so, may exact significant allocation inefficiencies.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] The concepts described herein are illustrated by way of example and not by way of limitation in the accompanying figures. For simplicity and clarity of illustration, elements illustrated in the figures are not necessarily drawn to scale. Where considered appropriate, reference labels have been repeated among the figures to indicate corresponding or analogous elements.

[0004] FIG. 1 is a simplified block diagram of at least one embodiment of a system for execution acceleration in heterogeneous environments;

[0005] FIG. 2 is a simplified block diagram of at least one embodiment of provisioning jobs to a heterogeneous server rack;

[0006] FIG. 3 is a simplified block diagram of at least one embodiment of an environment of an orchestrator of the system of FIG. 1;

[0007] FIG. 4 is a simplified flow diagram of at least one embodiment of a method for provisioning jobs to servers in a datacenter; and

[0008] FIG. 5 is a simplified flow diagram of at least one embodiment of a method for determining a server provisioning configuration.

DETAILED DESCRIPTION OF THE DRAWINGS

[0009] While the concepts of the present disclosure are susceptible to various modifications and alternative forms, specific embodiments thereof have been shown by way of example in the drawings and will be described herein in detail. It should be understood, however, that there is no intent to limit the concepts of the present disclosure to the particular forms disclosed, but on the contrary, the intention is to cover all modifications, equivalents, and alternatives consistent with the present disclosure and the appended claims.

[0010] References in the specification to “one embodiment,” “an embodiment,” “an illustrative embodiment,” etc., indicate that the embodiment described may include a particular feature, structure, or characteristic, but every embodiment may or may not necessarily include that particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it is submitted that it is within the knowledge of one skilled in the art to effect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described. Additionally, it should be appreciated that items included in a list in the form of “at least one A, B, and C” can mean (A); (B); (C); (A and B); (B and C); (A and C); or (A, B, and C). Similarly, items listed in the form of “at least one of A, B, or C” can mean (A); (B); (C); (A and B); (B and C); (A and C); or (A, B, and C).

[0011] The disclosed embodiments may be implemented, in some cases, in hardware, firmware, software, or any combination thereof. The disclosed embodiments may also be implemented as instructions carried by or stored on one or more transitory or non-transitory machine-readable (e.g., computer-readable) storage medium, which may be read and executed by one or more processors. A machine-readable storage medium may be embodied as any storage device, mechanism, or other physical structure for storing or transmitting information in a form readable by a machine (e.g., a volatile or non-volatile memory, a media disc, or other media device).

[0012] In the drawings, some structural or method features may be shown in specific arrangements and/or orderings. However, it should be appreciated that such specific arrangements and/or orderings may not be required. Rather, in some embodiments, such features may be arranged in a different manner and/or order than shown in the illustrative

figures. Additionally, the inclusion of a structural or method feature in a particular figure is not meant to imply that such feature is required in all embodiments and, in some embodiments, may not be included or may be combined with other features.

[0013] Referring now to FIG. 1, a system 100 for execution acceleration in heterogeneous environments includes an orchestrator 102 and one or more server racks 104. Additionally, in the illustrative embodiment of FIG. 1, each of the server racks 104 may include one or more servers 106. Although only one orchestrator 102 is illustratively shown in FIG. 1, the system 100 may include any number of orchestrators 102 in other embodiments (e.g., a distributed orchestration system). It should be appreciated that, in some embodiments, the system 100 may be embodied as a datacenter having a plurality of servers 106, for example, arranged into server racks 104 or other server groupings.

[0014] As described in detail below, in the illustrative embodiment, the orchestrator 102 computes a provisioning score that may be used for the selection of hosts (i.e., for job provisioning). For example, in some embodiments, the provisioning score may minimize/reduce the difference between the best server 106 performance and the worst server 106 performance, which may be critical to the scheduling of map/reduce, array, and/or HPC jobs. In the illustrative embodiment, the provisioning score is utilized to identify a set of similar servers 106 for job deployment/provisioning across a minimal number of racks that maximizes server utilization of the servers 106 within the utilized server racks 104. It should be appreciated that by prioritizing job placements within a heterogeneous infrastructure, the workload may run in a more predictable way and provide improved performance guarantees while maximizing datacenter rack utilization.

[0015] The orchestrator 102 may be embodied as any type of computing device capable of performing the functions described herein. For example, the orchestrator 102 may be embodied as a server, a rack-mounted server, a blade server, a desktop computer, laptop computer, tablet computer, notebook, netbook, Ultrabook™, cellular phone, smartphone, personal digital assistant, mobile Internet device, wearable computing device, Hybrid device, and/or any other compute/communication device. It should be appreciated that the illustrative orchestrator 102 is configured to handle the assignment/provisioning of workloads/jobs (e.g., map/reduce and array jobs) to the appropriate servers 106 in the system 100 (e.g., within the corresponding datacenter).

[0016] As shown in FIG. 1, the illustrative orchestrator 102 includes a processor 110, an input/output (“I/O”) subsystem 112, a memory 114, a data storage 116, and a communication circuitry 118. Of course, the orchestrator 102 may include other or additional components,

such as those commonly found in a typical computing device (e.g., various input/output devices and/or other components), in other embodiments. Additionally, in some embodiments, one or more of the illustrative components may be incorporated in, or otherwise form a portion of, another component. For example, the memory 114, or portions thereof, may be incorporated in the processor 110 in some embodiments.

[0017] The processor 110 may be embodied as any type of processor capable of performing the functions described herein. For example, the processor 110 may be embodied as a single or multi-core processor(s), digital signal processor, microcontroller, or other processor or processing/controlling circuit. Similarly, the memory 114 may be embodied as any type of volatile or non-volatile memory or data storage capable of performing the functions described herein. In operation, the memory 114 may store various data and software used during operation of the orchestrator 102 such as operating systems, applications, programs, libraries, and drivers. The memory 114 is communicatively coupled to the processor 110 via the I/O subsystem 112, which may be embodied as circuitry and/or components to facilitate input/output operations with the processor 110, the memory 114, and other components of the orchestrator 102. For example, the I/O subsystem 112 may be embodied as, or otherwise include, memory controller hubs, input/output control hubs, firmware devices, communication links (i.e., point-to-point links, bus links, wires, cables, light guides, printed circuit board traces, etc.) and/or other components and subsystems to facilitate the input/output operations. In some embodiments, the I/O subsystem 112 may form a portion of a system-on-a-chip (SoC) and be incorporated, along with the processor 110, the memory 114, and other components of the orchestrator 102, on a single integrated circuit chip.

[0018] The data storage 116 may be embodied as any type of device or devices configured for short-term or long-term storage of data such as, for example, memory devices and circuits, memory cards, hard disk drives, solid-state drives, or other data storage devices. The data storage 116 and/or the memory 114 may store various data during operation of the orchestrator 102 as described herein. For example, in some embodiments, the data storage 116 may store a provisioning policy 312 that identifies one or more provisioning configuration preferences as described below.

[0019] The communication circuitry 118 may be embodied as any communication circuit, device, or collection thereof, capable of enabling communications between the orchestrator 102 and other remote devices over a network. The communication circuitry 118 may be configured to use any one or more communication technologies (e.g., wireless or wired communications) and associated protocols (e.g., Ethernet, Bluetooth[®], Wi-Fi[®], WiMAX, LTE,

5G, etc.) to effect such communication. Additionally, in the illustrative embodiment, the communication circuitry 118 may include any communication circuit, device, or collection thereof, to enable the orchestrator 102 to communicate with each of servers 106 in the system 100 (e.g., via one or more communication links).

[0020] Each of the server racks 104 may be embodied as any device capable of organizing a subset of the servers 106 in the system 100 into a group. In the illustrative embodiment, each server rack 104 includes a chassis, enclosure, or housing having a plurality of slots/bays in which each of those slots/bays is adapted to receive a corresponding server 106 (or remain as an open slot/bay) as shown in the illustrative rack server 202 of FIG. 2. Further, in some embodiments, the slots/bays may be configured to receive servers 106 of different sizes (e.g., corresponding with different classes of servers 106). In other embodiments, it should be appreciated that one or more of the server racks 104 may be configured to group a subset of servers 106 together without physically receiving the servers 106 by a chassis, enclosure, or housing.

[0021] Each of the servers 106 may be embodied as any type of computing device capable of performing the functions described herein. For example, in some embodiments, each of the servers 106 may be embodied as a server, rack-mounted server, blade server, desktop computer, laptop computer, tablet computer, notebook, netbook, Ultrabook™, cellular phone, smartphone, personal digital assistant, mobile Internet device, wearable computing device, Hybrid device, and/or any other computing/communication device. Further, the servers 106 may include components similar to those of the orchestrator 102 discussed above. The description of those components of the orchestrator 102 is equally applicable to the description of components of the servers 106 and is not repeated herein for clarity of the description. Further, it should be appreciated that the one or more of the servers 106 may include other components, sub-components, and devices commonly found in a computing device, which are not discussed above in reference to the orchestrator 102 and not discussed herein for clarity of the description. In some embodiments, one or more of the components of the orchestrator 102 may be omitted from one or more of the servers 106.

[0022] It should be appreciated that, throughout the lifetime of a datacenter (e.g., the system 100), new servers 106 are added and old servers 106 are decommissioned. Rather than shuffling servers 106 throughout the datacenter any time there is an addition of another server 106 to the system 100, old and new server generations (and therefore classes) frequently coexist within a particular server rack 104. Accordingly, a server rack 104 having a homogenous architecture may evolve into a heterogeneous architecture over time due to the replacement of

servers 106 in the rack 104. For example, the illustrative server rack 202 of FIG. 2 includes seven servers 106 from three different classes. As described in greater detail below, the class of a server 106 is indicative of a set of particular hardware included on the corresponding server 106. For example, servers 106 having different chipsets, different hardware generations, and/or otherwise of a different type may have different classes depending on the particular embodiment. In particular, by way of example, Intel® Xeon Nahalem EP, Intel® Xeon Ivy Bridge EP, Intel® Xeon E5-4603, and Intel® Xeon E5-2450 servers have different classes relative to one another.

[0023] As shown in FIG. 2, the server rack 202 includes two open rack slots, one server 206 having a first class (class 1), three servers 206 having a second class (class 2), one server 206 having a third class (class 3), and two servers 206 having a fourth class (class 4). In the illustrative embodiment, the classes of servers 106 are defined such that a lower-numbered class has a slower processing speed than a higher-numbered class (e.g., class 1 is slower than class 2). As shown of FIG. 2, by way of a simplified illustration, seven jobs (e.g., parallel map/reduce jobs) are provisioned to the servers 106 of the server rack 202, one job to each of the servers 106. It should be appreciated that, assuming the workloads are homogeneous (and therefore require the same amount of processing), the class 1 server will process the job slower than the class 2 server and each of the other higher-class servers. As such, the “slowest” class of servers 106 in a particular rack 104 may most significantly influence the overall latency of the rack 104. Further, in actual deployment, the number of servers 106 included in a server rack 104 and/or the number of jobs provisioned to the servers 106 at a given time may be much greater in number.

[0024] Referring now to FIG. 3, in use, the orchestrator 102 establishes an environment 300 for execution acceleration in heterogeneous environments. The illustrative environment 300 includes a server identification module 302, a job allocation module 304, a provisioning module 306, and a communication module 308. Additionally, the illustrative job allocation module 304 includes a score calculation module 310. The various modules of the environment 300 may be embodied as hardware, software, firmware, or a combination thereof. For example, the various modules, logic, and other components of the environment 300 may form a portion of, or otherwise be established by, the processor 110 or other hardware components of the orchestrator 102. As such, in some embodiments, one or more of the modules of the environment 300 may be embodied as circuitry or collection of electrical devices (e.g., a server identification circuitry 302, a job allocation circuitry 304, a provisioning circuitry 306, a communication circuitry 308, and/or a score calculation circuitry 310). It should be appreciated

that, in such embodiments, one or more of the server identification circuitry 302, the job allocation circuitry 304, the provisioning circuitry 306, the communication circuitry 308, and/or the score calculation circuitry 310 may form a portion of one or more of the processor 110, the I/O subsystem 112, and/or other components of the orchestrator 102. Additionally, in some embodiments, one or more of the illustrative modules may form a portion of another module and/or one or more of the illustrative modules may be independent of one another. Further, in some embodiments, one or more of the modules of the environment 300 may be embodied as virtualized hardware components or emulated architecture, which may be established and maintained by the processor 110 or other components of the orchestrator 102.

[0025] The server identification module 302, which may be embodied as hardware, firmware, software, virtualized hardware, emulated architecture, and/or a combination thereof as discussed above, is configured to determine various information related to the servers 106 and the server racks 104 included in the system 100. For example, in the illustrative embodiment, the server identification module 302 determines which servers 106 are included on each of the server racks 104 and identifies the class of each of those servers 106. As indicated above, the class of a server 106 is indicative of the hardware (e.g., chipset) included on the server 106. It should be appreciated that the particular hardware components considered in distinguishing between the various classes of servers 106 may vary depending on the particular embodiment. Further, the number of classes of servers 106 included in a particular server rack 104 may also vary. In some embodiments, the server identification module 302 organizes each of the server racks 104 of the system 100 based on the server classes of the servers 106 in each of the respective racks 104. For example, in an embodiment, the server identification module 302 may determine the system 100 consists of sixty servers 106 organized into a first rack 104 that includes twenty servers 106 of a first class, a second rack 104 that includes ten servers 106 of a first class and ten servers 106 of a second class, and a third rack 104 that includes twenty servers 106 of a third class.

[0026] The job allocation module 304, which may be embodied as hardware, firmware, software, virtualized hardware, emulated architecture, and/or a combination thereof as discussed above, is configured to deploy/provision workloads that include a plurality of jobs (e.g., map/reduce or array jobs) to the servers 106 of the system 100. In doing so, the job allocation module 304 may select a workload available for provisioning and identify a provisioning configuration that defines the servers 106 to which each of the workload jobs are to be provisioned. In the illustrative embodiment, the provisioning configuration identifies the subset of the servers 106 to which the jobs are to be provisioned. That is, in some

embodiments, the provisioning configuration may identify the corresponding server 106 to which each of the jobs is to be deployed. As described in greater detail below, the job allocation module 304 may determine the provisioning configuration by selecting the configuration from a set of prospective configurations based on, or as a function of, a rack utilization of each prospective configuration, the classes of servers 106 utilized by each prospective configuration, and/or a number of racks utilized by each prospective configuration. In doing so, the job allocation module 304 may utilize the provisioning policy 312, which may identify various rules and/or preferences for selecting the appropriate configuration, and/or a provisioning score calculated by the score calculation module 310 for each of the prospective configurations.

[0027] The score calculation module 310, which may be embodied as hardware, firmware, software, virtualized hardware, emulated architecture, and/or a combination thereof as discussed above, determines a provisioning score for one or more prospective provisioning configurations (e.g., each possible configuration). In particular, in the illustrative embodiment, the score calculation module 310 may determine the provisioning score according to

$$s_i = (1/r) \sum (n/a),$$

wherein s is the provisioning score, i is an index that identifies the corresponding prospective configuration, r is a number of racks involved in the corresponding prospective configuration, n is a number of servers 106 involved in the corresponding prospective configuration, and a is a number of available servers 106 to which a job could be provisioned. In the illustrative embodiment, the score calculation module 310 only considers the servers 106 that have enough capacity to run one or more of the particular jobs of the workload (e.g., enough processor utilization and/or memory capacity). Further, it should be appreciated that the orchestrator 102 and/or the provisioning policy 312 may define an order among the various server classes (e.g., that class 2 is better than class 1, etc.). In the illustrative embodiment, the provisioning score incorporates a rack utilization ratio defined according to (n/a) and penalizes jobs spreading across multiple racks 104 (e.g., by virtue of the $(1/r)$ term). As described below, the provisioning scores may be compared in order to select a prospective configuration as the configuration to use for provisioning (e.g., the configuration associated with the greatest provisioning score). It should be appreciated that, in some embodiments, the score calculation module 310 of the job allocation module 304 may be separately embodied as hardware, firmware, software, virtualized hardware, emulated architecture, and/or a combination thereof.

[0028] The provisioning policy 312 may define various rules and/or preferences associated with selecting the provisioning configuration. For example, in some embodiments, the provisioning policy 312 may define rules or principles to be used as “tie breakers” in circumstances in which multiple configurations have the same “best” provisioning score. Of course, in some embodiments, the rules or principles may be utilized prior to the calculation of the provisioning score, for example, to filter a subset of prospective configurations from a larger set of prospective configurations under consideration such that the provisioning configuration is selected from one of the configurations in the filtered subset. For example, in some embodiments, the provisioning policy 312 may indicate to select the prospective configuration that utilizes the fewest racks and/or that has the greater rack utilization in response to a determination that multiple prospective configurations have the best provisioning score. Further, in some embodiments, the provisioning policy 312 may indicate that the jobs should be provisioned to servers 106 of the same class if possible.

[0029] The provisioning module 306, which may be embodied as hardware, firmware, software, virtualized hardware, emulated architecture, and/or a combination thereof as discussed above, is configured to provision the jobs of the workload to the appropriate servers 106. For example, in the illustrative embodiment, the provisioning module provisions the jobs to the subset of available servers 106 identified by the provisioning configuration as described herein.

[0030] The communication module 308, which may be embodied as hardware, firmware, software, virtualized hardware, emulated architecture, and/or a combination thereof as discussed above, handles the communication between the orchestrator 102 and other computing devices of the system 100 (e.g., the servers 106). For example, as described herein, the orchestrator 102 is configured to deploy various jobs of a workload to servers 106 selected based on the techniques described herein.

[0031] Referring now to FIG. 4, in use, the orchestrator 102 may execute a method 400 for provisioning jobs to servers 106, for example, in a datacenter. The illustrative method 400 begins with block 402 in which the orchestrator 102 identifies the classes of the available servers 106 of each of the racks 104. In doing so, in block 404, the orchestrator 102 may organize each of the server racks 104 based on the class(es) of the servers 106 in the corresponding server rack 104 or otherwise determine how many servers 106 belong to each class for each of the racks 104. For example, in an embodiment described above, the orchestrator 102 may determine that the system 100 consists of sixty servers 104 organized into a first rack 104 that includes twenty servers 106 of a first class, a second rack 104 that includes

ten servers 106 of a first class and ten servers 106 of a second class, and a third rack 104 that includes twenty servers 106 of a third class.

[0032] In block 406, the orchestrator 102 determines whether a workload is available for provisioning/deployment to the servers 106. If the orchestrator 102 determines that at least one workload is available for provisioning, the orchestrator 102 selects one of the workloads (e.g., assuming there are multiple workloads in the queue) for provisioning to the servers 106 in block 408. It should be appreciated that the orchestrator 102 may select the workload to provision based on any suitable techniques and/or algorithms, for example, for workload priority management. In block 410, the orchestrator 102 determines the server provisioning configuration. To do so, in some embodiments, the orchestrator 102 may execute a method 500 described below in reference to FIG. 5. In block 412, the orchestrator 102 provisions the jobs of the selected workload based on the determined server provisioning configuration.

[0033] Referring now to FIG. 5, in use, the orchestrator 102 may execute the method 500 for determining the server provisioning configuration. The illustrative method 500 begins with block 502 in which the orchestrator 102 determines the number of jobs, n , included in the workload. In block 504, the orchestrator 102 determines a provisioning score for a subset of servers 106. In particular, in some embodiments, the orchestrator 102 determines the provisioning score for the subset according to

$$s_i = (1/r) \sum (n/a),$$

wherein s is the provisioning score, i is an index that identifies the corresponding prospective configuration, r is a number of racks involved in the corresponding prospective configuration, n is a number of servers 106 involved in the corresponding prospective configuration, and a is a number of available servers 106 to which a job could be provisioned. As indicated above, it should be appreciated that the provisioning score incorporates a rack utilization ratio defined by (n/a) and penalizes jobs spreading across multiple racks 104 (e.g., by virtue of the $(1/r)$ term). In block 506, the orchestrator 102 determines whether to determine the provisioning score for another subset (e.g., for another provisioning configuration). If so, the method 500 selects another subset and returns to block 504 to determine the provisioning score for that subset.

[0034] Otherwise, the method 500 proceeds to block 508 in which the orchestrator 102 determines whether all of the jobs of the workload are provisionable to the same class of servers 106 in the system 100. If not, the orchestrator 102 selects the provisioning configuration(s) having the highest provisioning score in block 510. Returning to block 408, if the orchestrator

102 determines that the jobs are all provisionable to the same class (e.g., due to lack of servers 106 having that class, lack of processing/memory capacity of the servers 106 of that class, etc.), the orchestrator 102 filters the prospective configurations having a preferred class based on the provisioning policy 312 in block 512, and the provisioning configuration is selected from the filtered set of prospective configurations in block 510. It should be appreciated that the provisioning policy 312 may define the preferred class in any suitable way depending on the particular embodiment. For example, in the illustrative embodiment, the provisioning policy 312 indicates that less efficient classes have priority/preference over more efficient classes (e.g., class 1 has the highest priority, class 2 has priority over class 3, class 3 has priority over class 4, etc.). As indicated above, in other embodiments, the provisioning policy 312 may be applied to filter the prospective configurations as described above only if there are multiple configurations having the best (e.g., highest) provisioning score. Further, in some embodiments, the prospective configurations may only be filtered according to the rules described above if all of the jobs may be deployed to servers 106 of the highest priority of class of the available servers 106 (e.g., only if all of the jobs may be deployed to servers 106 of the slowest server class).

[0035] In some embodiments, it should be appreciated that multiple provisioning configurations may have the same provisioning score. Accordingly, in block 514, the orchestrator 102 determines whether there are multiple configurations having the highest and, therefore, matching provisioning score. If so, in the illustrative embodiment, the orchestrator 102 applies the rules of the provisioning policy 312 to identify the configuration to select for provisioning. That is, in block 516, the orchestrator 102 selects the provisioning configuration based on the provisioning policy 312. It should be appreciated that the particular provisioning rules of the provisioning policy 312 may vary depending on the particular embodiment and may be applied in any suitable order to identify the appropriate provisioning configuration. For example, in block 518, the orchestrator 102 may select the configuration that minimizes or utilizes a fewer number of server racks 104 (e.g., to minimize network overhead from inter-rack communication). Further, in block 520, the orchestrator 102 may select the configuration that maximizes server platform utilization within the racks 104. In some embodiments, the provisioning policy 312 may further identify a procedure by which to select the configuration if the application of the defined rules does not result in a clear choice for the configuration (e.g., by random selection).

[0036] By way of example, suppose the orchestrator 102 were to provision a sequence of three workloads to the servers 106 of the system 100 described above in which the system 100 consists of sixty servers 104 organized into a first rack 104 that includes twenty servers 106

of a first class, a second rack 104 that includes ten servers 106 of a first class and ten servers 106 of a second class, and a third rack 104 that includes twenty servers 106 of a third class based on the techniques described herein (e.g., based on the method 500). In such an example, suppose the first workload includes five jobs, the second workload includes fifteen jobs, and the third workload includes eleven jobs.

[0037] For the first workload, the orchestrator 102 may determine that there are enough servers 106 of the first class (twenty in the first rack 104 and ten in the second rack 104) to provision all of the jobs of the workload to servers 106 of the first class such that it is unnecessary to utilize the higher-class servers 106. Accordingly, the orchestrator 102 may calculate the provisioning formula, $s_i = (1/r) \sum (n/a)$, described above for the relevant server subsets/configurations. In particular, the provisioning scores may be calculated to be $s_1 = (1/1)(5/20) = 0.2$ and $s_2 = (1/1)(5/10) = 0.5$, where s_1 is the score associated with provisioning the jobs to the first class servers 106 of the first rack 104 and s_2 is the score associated with provisioning the jobs to the first class servers 106 of the second rack 104. In the illustrative embodiment, the orchestrator 102 selects the configuration having the greatest/maximum provisioning score (i.e., s_2), so the orchestrator 102 determines to provision the jobs to the first class servers 106 of the second rack 104. It should be appreciated that by assigning the workload to a single rack 104, inter-process communication and HPC jobs (e.g., OpenMP/MPI and CUDA) may run faster and with less network overhead (e.g., due to the servers 106 in the rack 104 generally being wired through the same network device).

[0038] After deployment of the first workload, it should be appreciated that the system 100 now has fifty-five available servers 106 grouped into a first rack 104 that includes twenty servers 106 of a first class, a second rack 104 that includes five servers 106 of a first class and ten servers 106 of a second class, and a third rack 104 that includes twenty servers 106 of a third class. For the second workload (fifteen jobs), the orchestrator 102 again determines that there are enough servers 106 of the first class (twenty in the first rack 104 and five in the second rack 104) to fulfill the workload request by provisioning all of the jobs to servers 106 of the first class. As such, the orchestrator 102 calculates the relevant provisioning scores to be $s_1 = (1/1)(15/20) = 0.7$ and $s_2 = (1/2)(5/5 + 10/20) = 0.7$, where s_1 is the score associated with provisioning all of the jobs to the first class servers 106 of the first rack 104 and s_2 is the score associated with provisioning five of the jobs to the available first class servers 106 of the second rack 104 and the remaining ten jobs to the first class servers 106 of the first rack 104. Because the provisioning scores are equal, the orchestrator 102 applies the rules of the

provisioning policy 312 in the illustrative embodiment. In particular, the orchestrator 102 selects the configuration that minimizes the number of racks 104 used for the deployment and, therefore, provisions all fifteen of the jobs to the first class servers 106 of the first rack 104.

[0039] After deployment of the second workload, it should be appreciated that the system 100 now has forty available servers 106 from a first rack 104 that includes five servers 106 of a first class, a second rack 104 that includes five servers 106 of a first class and ten servers 106 of a second class, and a third rack 104 that includes twenty servers 106 of a third class. For the third workload (eleven jobs), the orchestrator 102 determines that there are too few servers 106 of the first class to provision all of the jobs to servers 106 of the first class and, therefore, must assign at least one job to a server 106 of a higher class (e.g., class 2). Accordingly, the orchestrator 102 calculates the relevant provisioning scores to be $s_1 = (1/3)(5/5 + 5/5 + 1/10) = 0.7$ and $s_2 = (1/3)(5/5 + 5/5 + 1/20) = 0.6$, where s_1 is the score associated with provisioning five of the jobs to the available first class servers 106 of the first rack 104, five of the jobs to the available first class servers 106 of the second rack 104, and the last job to a second class server of the second rack 104 and s_2 is the score associated with provisioning five of the jobs to the available first class servers 106 of the first rack 104, five of the jobs to the available first class servers 106 of the second rack 104, and the last job to a third class server 106 of the third rack 104. In the illustrative embodiment, the orchestrator 102 selects the configuration having the greatest/maximum provisioning score (i.e., s_1) and, therefore, determines to provision five of the jobs to the available first class servers 106 of the first rack 104, five of the jobs to the available first class servers 106 of the second rack 104, and the last job to a second class server of the second rack 104. It should be appreciated that the illustrative example described herein represents the application of one embodiment of the provisioning policy 312 and the techniques described herein are, therefore, not so limited.

EXAMPLES

[0040] Illustrative examples of the technologies disclosed herein are provided below. An embodiment of the technologies may include any one or more, and any combination of, the examples described below.

[0041] Example 1 includes a compute device for provisioning jobs to servers at a datacenter, the compute device comprising a server identification module to identify a class of each of a plurality of available servers in the datacenter, wherein the class is indicative of a set of hardware included on the corresponding server; and a job allocation module to (i) select a workload that includes a plurality of jobs to provision to a subset of the available servers (ii)

determine a provisioning configuration for the workload from a set of prospective configurations as a function of a rack utilization of each prospective configuration, the classes utilized by each prospective configuration, and a number of racks utilized by each prospective configuration, wherein the provisioning configuration identifies a subset of the available servers to which to provision the jobs of the workload.

[0042] Example 2 includes the subject matter of Example 1, and further including a provisioning module to provision the jobs of the workload to the identified subset of the available servers.

[0043] Example 3 includes the subject matter of any of Examples 1 and 2, and wherein the compute device comprises an orchestrator of the datacenter.

[0044] Example 4 includes the subject matter of any of Examples 1-3, and where to identify the class of each of the plurality of available servers comprises to organize each server rack associated with the plurality of available servers based on the class of the available servers.

[0045] Example 5 includes the subject matter of any of Examples 1-4, and wherein to determine the provisioning configuration comprises to determine a provisioning score for each prospective configuration of the set of prospective configuration.

[0046] Example 6 includes the subject matter of any of Examples 1-5, and wherein to determine the provisioning score for each prospective configuration comprises to determine the provisioning score according to

$$s_i = (1/r) \sum (n/a),$$

wherein s is the provisioning score, i is an index that identifies the corresponding prospective configuration, r is a number of racks involved in the corresponding prospective configuration, n is a number of servers involved in the corresponding prospective configuration, and a is a number of available servers.

[0047] Example 7 includes the subject matter of any of Examples 1-6, and wherein to determine the provisioning configuration comprises to select the prospective configuration having a greatest provisioning score as the provisioning configuration.

[0048] Example 8 includes the subject matter of any of Examples 1-7, and wherein the job allocation module is further to select a prospective configuration as the provisioning configuration based on a provisioning policy of the compute device in response to a determination that multiple prospective configurations have the greatest provisioning score.

[0049] Example 9 includes the subject matter of any of Examples 1-8, and wherein the provisioning policy indicates to select the prospective configuration that utilizes a lesser

number of racks in response to a determination that multiple prospective configurations have a matching provisioning score.

[0050] Example 10 includes the subject matter of any of Examples 1-9, and wherein the provisioning policy indicates to select the prospective configuration that has a greater rack utilization in response to a determination that the multiple prospective configurations have a matching provisioning score.

[0051] Example 11 includes the subject matter of any of Examples 1-10, and wherein to determine the provisioning configuration comprises to determine whether each of the jobs in the workload can be provisioned to available servers having a same class.

[0052] Example 12 includes the subject matter of any of Examples 1-11, and wherein to determine the provisioning configuration comprises to select a prospective configuration associated with a least efficient class in response to a determination that each of the jobs in the workload can be provisioned to available servers having the same class for multiple different classes.

[0053] Example 13 includes the subject matter of any of Examples 1-12, and wherein the plurality of jobs comprises parallel map/reduce jobs.

[0054] Example 14 includes a method for provisioning jobs to servers at a datacenter, the method comprising identifying, by a compute device, a class of each of a plurality of available servers in the datacenter, wherein the class is indicative of a set of hardware included on the corresponding server; selecting, by the compute device, a workload that includes a plurality of jobs to provision to a subset of the available servers; and determining, by the compute device, a provisioning configuration for the workload from a set of prospective configurations as a function of a rack utilization of each prospective configuration, the classes utilized by each prospective configuration, and a number of racks utilized by each prospective configuration, wherein the provisioning configuration identifies a subset of the available servers to which to provision the jobs of the workload.

[0055] Example 15 includes the subject matter of Example 14, and further including provisioning, by the compute device, the jobs of the workload to the identified subset of the available servers.

[0056] Example 16 includes the subject matter of any of Examples 14 and 15, and wherein the compute device comprises an orchestrator of the datacenter.

[0057] Example 17 includes the subject matter of any of Examples 14-16, and where identifying the class of each of the plurality of available servers comprises organizing each

server rack associated with the plurality of available servers based on the class of the available servers.

[0058] Example 18 includes the subject matter of any of Examples 14-17, and wherein determining the provisioning configuration comprises determining a provisioning score for each prospective configuration of the set of prospective configuration.

[0059] Example 19 includes the subject matter of any of Examples 14-18, and wherein determining the provisioning score for each prospective configuration comprises determining the provisioning score according to

$$s_i = (1/r) \sum (n/a),$$

wherein s is the provisioning score, i is an index that identifies the corresponding prospective configuration, r is a number of racks involved in the corresponding prospective configuration, n is a number of servers involved in the corresponding prospective configuration, and a is a number of available servers.

[0060] Example 20 includes the subject matter of any of Examples 14-19, and wherein determining the provisioning configuration comprises selecting the prospective configuration having a greatest provisioning score as the provisioning configuration.

[0061] Example 21 includes the subject matter of any of Examples 14-20, and further including selecting, by the compute device, a prospective configuration as the provisioning configuration based on a provisioning policy of the compute device in response to a determination that multiple prospective configurations have the greatest provisioning score.

[0062] Example 22 includes the subject matter of any of Examples 14-21, and wherein the provisioning policy indicates to select the prospective configuration that utilizes a lesser number of racks in response to a determination that multiple prospective configurations have a matching provisioning score.

[0063] Example 23 includes the subject matter of any of Examples 14-22, and wherein the provisioning policy indicates to select the prospective configuration that has a greater rack utilization in response to a determination that the multiple prospective configurations have a matching provisioning score.

[0064] Example 24 includes the subject matter of any of Examples 14-23, and wherein determining the provisioning configuration comprises determining whether each of the jobs in the workload can be provisioned to available servers having a same class.

[0065] Example 25 includes the subject matter of any of Examples 14-24, and wherein determining the provisioning configuration comprises selecting a prospective configuration associated with a least efficient class in response to a determination that each of the jobs in the

workload can be provisioned to available servers having the same class for multiple different classes.

[0066] Example 26 includes the subject matter of any of Examples 14-25, and wherein the plurality of jobs comprises parallel map/reduce jobs.

[0067] Example 27 includes a computing device comprising a processor; and a memory having stored therein a plurality of instructions that when executed by the processor cause the computing device to perform the method of any of Examples 14-26.

[0068] Example 28 includes one or more machine-readable storage media comprising a plurality of instructions stored thereon that in response to being executed result in a computing device performing the method of any of Examples 14-26.

[0069] Example 29 includes a computing device comprising means for performing the method of any of Examples 14-26.

[0070] Example 30 includes a compute device for provisioning jobs to servers at a datacenter, the compute device comprising means for identifying a class of each of a plurality of available servers in the datacenter, wherein the class is indicative of a set of hardware included on the corresponding server; means for selecting a workload that includes a plurality of jobs to provision to a subset of the available servers; and means for determining a provisioning configuration for the workload from a set of prospective configurations as a function of a rack utilization of each prospective configuration, the classes utilized by each prospective configuration, and a number of racks utilized by each prospective configuration, wherein the provisioning configuration identifies a subset of the available servers to which to provision the jobs of the workload.

[0071] Example 31 includes the subject matter of Example 30, and further including means for provisioning the jobs of the workload to the identified subset of the available servers.

[0072] Example 32 includes the subject matter of any of Examples 30 and 31, and wherein the compute device comprises an orchestrator of the datacenter.

[0073] Example 33 includes the subject matter of any of Examples 30-32, and where the means for identifying the class of each of the plurality of available servers comprises means for organizing each server rack associated with the plurality of available servers based on the class of the available servers.

[0074] Example 34 includes the subject matter of any of Examples 30-33, and wherein the means for determining the provisioning configuration comprises means for determining a provisioning score for each prospective configuration of the set of prospective configuration.

[0075] Example 35 includes the subject matter of any of Examples 30-34, and wherein the means for determining the provisioning score for each prospective configuration comprises means for determining the provisioning score according to

$$s_i = (1/r) \sum (n/a),$$

wherein s is the provisioning score, i is an index that identifies the corresponding prospective configuration, r is a number of racks involved in the corresponding prospective configuration, n is a number of servers involved in the corresponding prospective configuration, and a is a number of available servers.

[0076] Example 36 includes the subject matter of any of Examples 30-35, and wherein the means for determining the provisioning configuration comprises means for selecting the prospective configuration having a greatest provisioning score as the provisioning configuration.

[0077] Example 37 includes the subject matter of any of Examples 30-36, and further including means for selecting a prospective configuration as the provisioning configuration based on a provisioning policy of the compute device in response to a determination that multiple prospective configurations have the greatest provisioning score.

[0078] Example 38 includes the subject matter of any of Examples 30-37, and wherein the provisioning policy indicates to select the prospective configuration that utilizes a lesser number of racks in response to a determination that multiple prospective configurations have a matching provisioning score.

[0079] Example 39 includes the subject matter of any of Examples 30-38, and wherein the provisioning policy indicates to select the prospective configuration that has a greater rack utilization in response to a determination that the multiple prospective configurations have a matching provisioning score.

[0080] Example 40 includes the subject matter of any of Examples 30-39, and wherein the means for determining the provisioning configuration comprises means for determining whether each of the jobs in the workload can be provisioned to available servers having a same class.

[0081] Example 41 includes the subject matter of any of Examples 30-40, and wherein the means for determining the provisioning configuration comprises means for selecting a prospective configuration associated with a least efficient class in response to a determination that each of the jobs in the workload can be provisioned to available servers having the same class for multiple different classes.

[0082] Example 42 includes the subject matter of any of Examples 30-41, and wherein the plurality of jobs comprises parallel map/reduce jobs.

WHAT IS CLAIMED IS:

1. A compute device for provisioning jobs to servers at a datacenter, the compute device comprising:

a server identification module to identify a class of each of a plurality of available servers in the datacenter, wherein the class is indicative of a set of hardware included on the corresponding server; and

a job allocation module to (i) select a workload that includes a plurality of jobs to provision to a subset of the available servers (ii) determine a provisioning configuration for the workload from a set of prospective configurations as a function of a rack utilization of each prospective configuration, the classes utilized by each prospective configuration, and a number of racks utilized by each prospective configuration, wherein the provisioning configuration identifies a subset of the available servers to which to provision the jobs of the workload.

2. The compute device of claim 1, further comprising a provisioning module to provision the jobs of the workload to the identified subset of the available servers.

3. The compute device of claim 1, wherein the compute device comprises an orchestrator of the datacenter.

4. The compute device of any of claims 1-3, where to identify the class of each of the plurality of available servers comprises to organize each server rack associated with the plurality of available servers based on the class of the available servers.

5. The compute device of claim 1, wherein to determine the provisioning configuration comprises to determine a provisioning score for each prospective configuration of the set of prospective configuration.

6. The compute device of claim 5, wherein to determine the provisioning score for each prospective configuration comprises to determine the provisioning score according to

$$s_i = (1/r) \sum (n/a),$$

wherein s is the provisioning score, i is an index that identifies the corresponding prospective configuration, r is a number of racks involved in the corresponding prospective configuration, n

is a number of servers involved in the corresponding prospective configuration, and a is a number of available servers.

7. The compute device of claim 6, wherein to determine the provisioning configuration comprises to select the prospective configuration having a greatest provisioning score as the provisioning configuration.

8. The compute device of claim 7, wherein the job allocation module is further to select a prospective configuration as the provisioning configuration based on a provisioning policy of the compute device in response to a determination that multiple prospective configurations have the greatest provisioning score.

9. The compute device of claim 8, wherein the provisioning policy indicates to select the prospective configuration that utilizes a lesser number of racks in response to a determination that multiple prospective configurations have a matching provisioning score.

10. The compute device of claim 8, wherein the provisioning policy indicates to select the prospective configuration that has a greater rack utilization in response to a determination that the multiple prospective configurations have a matching provisioning score.

11. The compute device of claim 1, wherein to determine the provisioning configuration comprises to determine whether each of the jobs in the workload can be provisioned to available servers having a same class.

12. The compute device of claim 11, wherein to determine the provisioning configuration comprises to select a prospective configuration associated with a least efficient class in response to a determination that each of the jobs in the workload can be provisioned to available servers having the same class for multiple different classes.

13. A method for provisioning jobs to servers at a datacenter, the method comprising:

identifying, by a compute device, a class of each of a plurality of available servers in the datacenter, wherein the class is indicative of a set of hardware included on the corresponding server;

selecting, by the compute device, a workload that includes a plurality of jobs to provision to a subset of the available servers; and

determining, by the compute device, a provisioning configuration for the workload from a set of prospective configurations as a function of a rack utilization of each prospective configuration, the classes utilized by each prospective configuration, and a number of racks utilized by each prospective configuration, wherein the provisioning configuration identifies a subset of the available servers to which to provision the jobs of the workload.

14. The method of claim 13, further comprising provisioning, by the compute device, the jobs of the workload to the identified subset of the available servers.

15. The method of claim 13, wherein the compute device comprises an orchestrator of the datacenter.

16. The method of claim 13, where identifying the class of each of the plurality of available servers comprises organizing each server rack associated with the plurality of available servers based on the class of the available servers.

17. The method of claim 13, wherein determining the provisioning configuration comprises determining a provisioning score for each prospective configuration of the set of prospective configuration.

18. The method of claim 17, wherein determining the provisioning score for each prospective configuration comprises determining the provisioning score according to

$$s_i = (1/r) \sum (n/a),$$

wherein s is the provisioning score, i is an index that identifies the corresponding prospective configuration, r is a number of racks involved in the corresponding prospective configuration, n is a number of servers involved in the corresponding prospective configuration, and a is a number of available servers.

19. The method of claim 18, wherein determining the provisioning configuration comprises selecting the prospective configuration having a greatest provisioning score as the provisioning configuration.

20. The method of claim 19, further comprising selecting, by the compute device, a prospective configuration as the provisioning configuration based on a provisioning policy of the compute device in response to a determination that multiple prospective configurations have the greatest provisioning score.

21. The method of claim 20, wherein the provisioning policy indicates to select the prospective configuration that utilizes a lesser number of racks in response to a determination that multiple prospective configurations have a matching provisioning score.

22. The method of claim 20, wherein the provisioning policy indicates to select the prospective configuration that has a greater rack utilization in response to a determination that the multiple prospective configurations have a matching provisioning score.

23. The method of claim 13, wherein determining the provisioning configuration comprises determining whether each of the jobs in the workload can be provisioned to available servers having a same class.

24. The method of claim 23, wherein determining the provisioning configuration comprises selecting a prospective configuration associated with a least efficient class in response to a determination that each of the jobs in the workload can be provisioned to available servers having the same class for multiple different classes.

25. One or more machine-readable storage media comprising a plurality of instructions stored thereon that in response to being executed result in a computing device performing the method of any of claims 13-24.

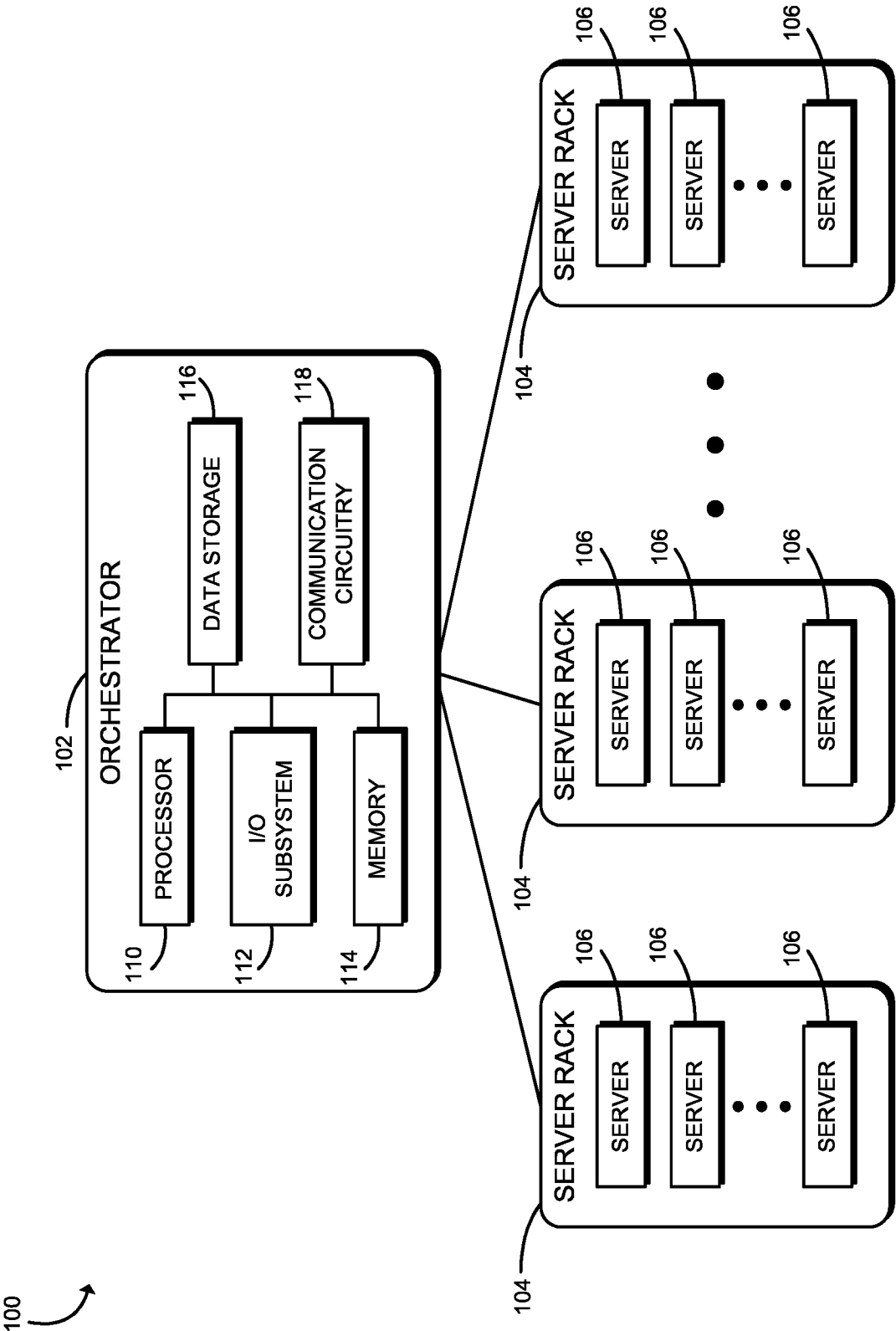


FIG. 1

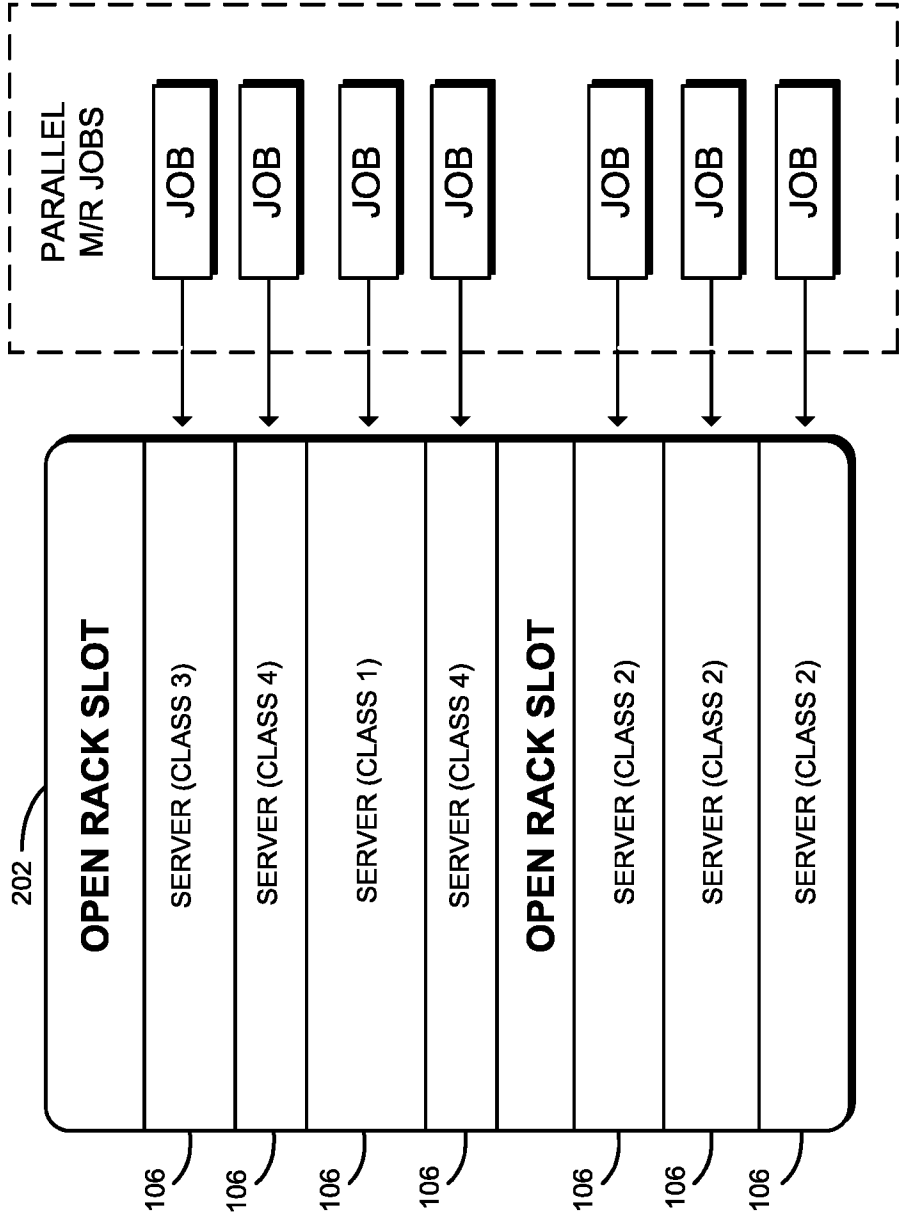


FIG. 2

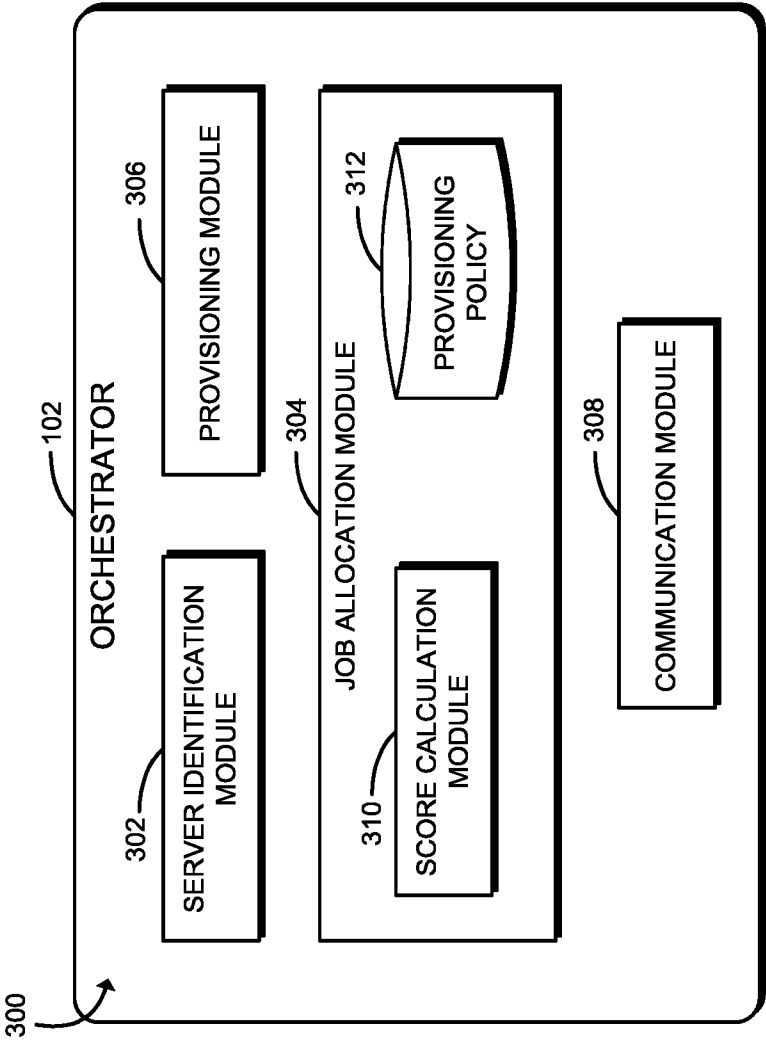


FIG. 3

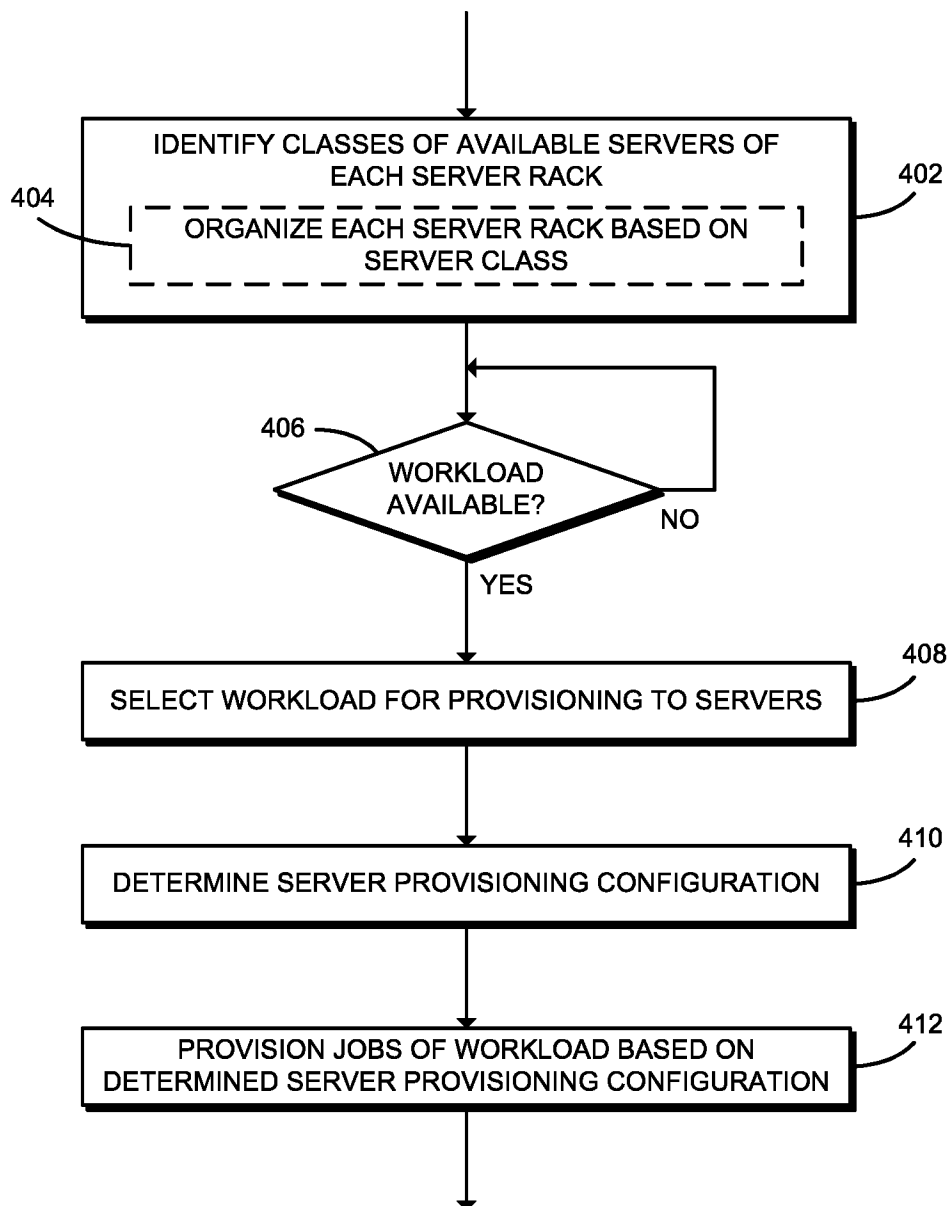
400
↓

FIG. 4

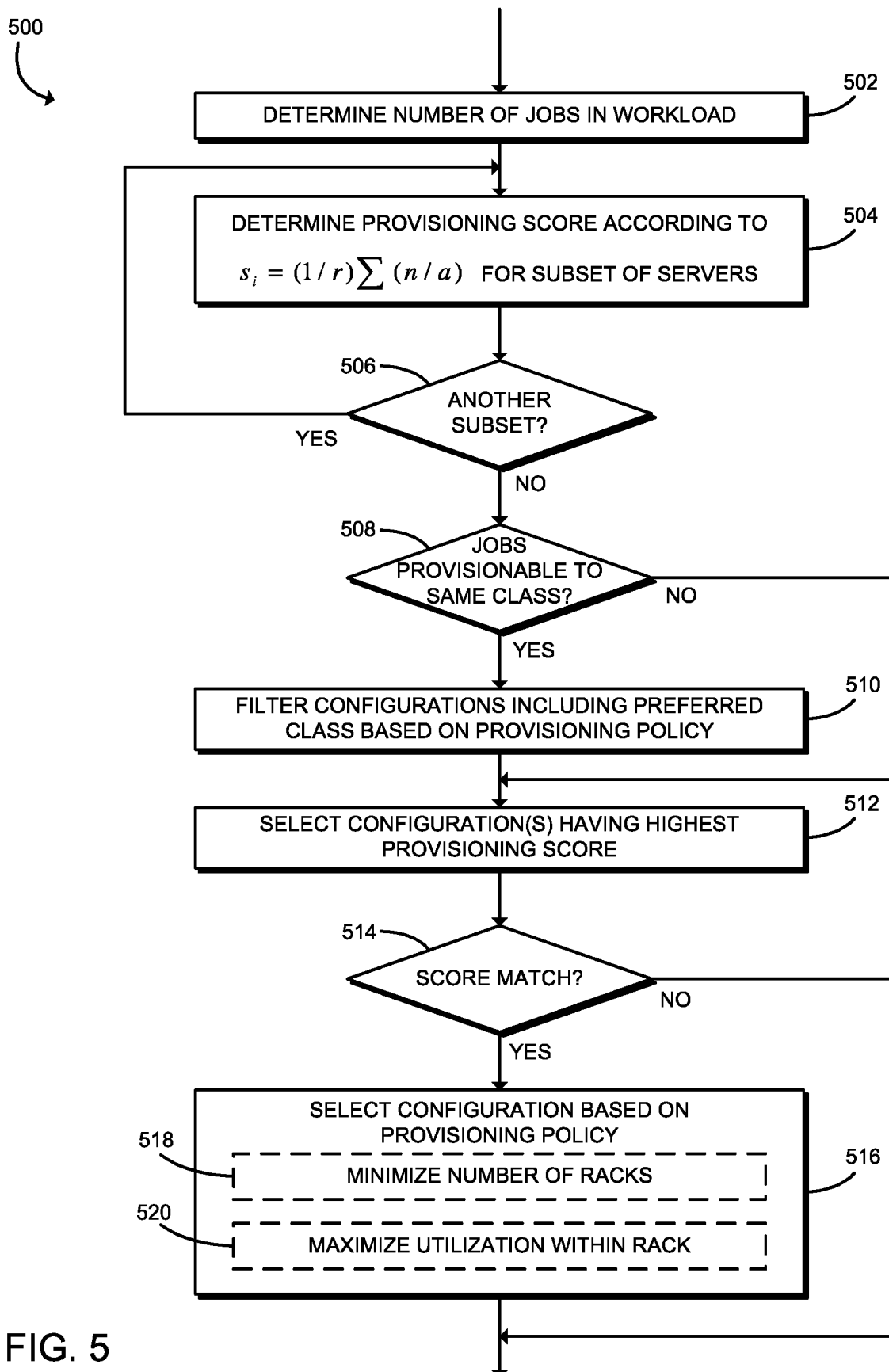


FIG. 5

A. CLASSIFICATION OF SUBJECT MATTER**H04L 29/08(2006.01)i, H04L 12/933(2013.01)i, H04L 12/66(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)

H04L 29/08; G06F 15/173; H04L 12/24; H04L 12/28; G05D 23/19; G06F 1/32; G05D 23/00; H04L 12/933; H04L 12/66

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: load balancing, server, class, group, rack, utilization, configuration

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2011-0077795 A1 (JAMES W. VANGILDER et al.) 31 March 2011 See paragraphs [0039], [0126], [0129], [0132]-[0134], [0137], [0143], [0149]-[0150], [0152] and figures 1, 7-9.	1-25
Y	US 2013-0290498 A1 (COMMScope, INC. OF NORTH CAROLINA) 31 October 2013 See paragraph [0053], claim 1 and figure 1.	1-25
Y	US 2007-0258465 A1 (WEIMIN MA et al.) 08 November 2007 See paragraphs [0035]-[0036] and figure 4.	12, 24
A	US 2009-0276095 A1 (WILLIAM THOMAS PIENTA et al.) 05 November 2009 See paragraphs [0028]-[0031] and figure 1.	1-25
A	US 2013-0290540 A1 (LOUIS J. CIOCCI et al.) 31 October 2013 See claims 1-11 and figure 1.	1-25



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

06 March 2017 (06.03.2017)

Date of mailing of the international search report

13 March 2017 (13.03.2017)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

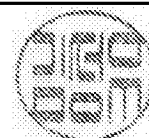
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

KIM, Seong Woo

Telephone No. +82-42-481-3348



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/063593

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2011-0077795 A1	31/03/2011	AU 2010-214029 A1	19/08/2010
		AU 2011-305487 A1	18/04/2013
		AU 214029 B2	09/07/2015
		AU 305487 B2	21/05/2015
		CA 2752613 A1	19/08/2010
		CN 102395955 A	28/03/2012
		CN 102395955 B	06/08/2014
		CN 103221898 A	24/07/2013
		CN 103221898 B	22/04/2015
		EP 2619637 A1	31/07/2013
		US 2010-0211669 A1	19/08/2010
		US 8560677 B2	15/10/2013
		US 9519517 B2	13/12/2016
		WO 2010-093515 A1	19/08/2010
		WO 2012-040334 A1	29/03/2012
US 2013-0290498 A1	31/10/2013	GB 201112290 D0	31/08/2011
		GB 2480020 A	02/11/2011
		GB 2480020 B	08/07/2015
		US 2010-0214940 A1	26/08/2010
		US 8472333 B2	25/06/2013
		US 9246758 B2	26/01/2016
		WO 2010-096528 A1	26/08/2010
US 2007-0258465 A1	08/11/2007	CN 101438609 A	20/05/2009
		CN 101438609 B	08/02/2012
		EP 2014109 A2	14/01/2009
		US 7640023 B2	29/12/2009
		WO 2007-130290 A2	15/11/2007
		WO 2007-130290 A3	02/10/2008
US 2009-0276095 A1	05/11/2009	CA 2723407 A1	12/11/2009
		CA 2723442 A1	12/11/2009
		CA 2723908 A1	12/11/2009
		CN 102016754 A	13/04/2011
		CN 102084316 A	01/06/2011
		CN 102084316 B	18/05/2016
		CN 102150100 A	10/08/2011
		CN 102150100 B	17/06/2015
		EP 2277092 A2	26/01/2011
		EP 2277092 B1	16/03/2016
		EP 2277093 A2	26/01/2011
		EP 2277093 B1	31/08/2016
		EP 2277094 A1	26/01/2011
		KR 10-1557177 B1	02/10/2015
		KR 10-1563031 B1	23/10/2015
		KR 10-1578961 B1	18/12/2015
		KR 10-2011-0015597 A	16/02/2011
		MX 2010012107 A	30/11/2010

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/063593

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		US 2009-0276528 A1	05/11/2009
		US 2009-0292811 A1	26/11/2009
		US 8260928 B2	04/09/2012
		US 8782234 B2	15/07/2014
		US 8954197 B2	10/02/2015
		WO 2009-137026 A2	12/11/2009
		WO 2009-137026 A3	07/01/2010
		WO 2009-137027 A2	12/11/2009
		WO 2009-137027 A3	28/01/2010
		WO 2009-137028 A1	12/11/2009
US 2013-0290540 A1	31/10/2013	US 2016-0044097 A1	11/02/2016
		US 9176790 B2	03/11/2015