

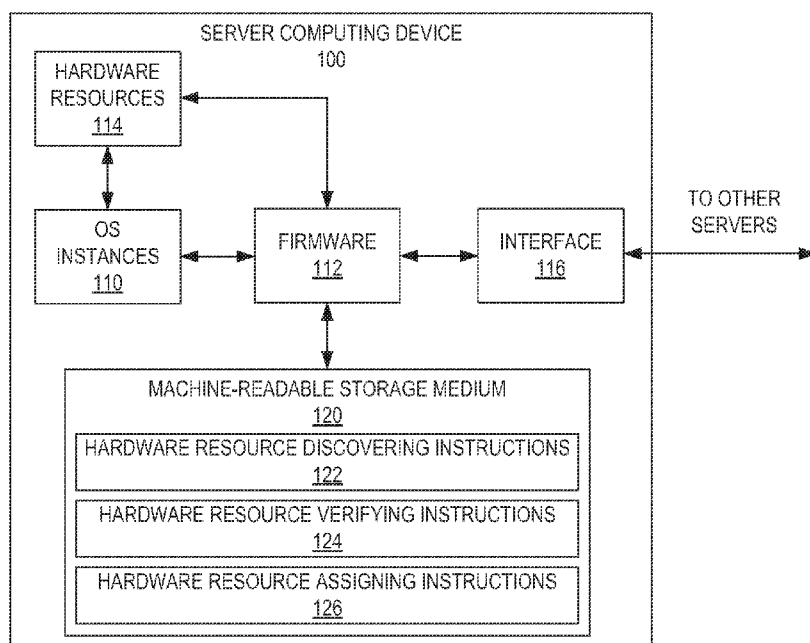


- (51) **International Patent Classification:**
G06F 9/44 (2006.01)
- (21) **International Application Number:**
PCT/US2014/069625
- (22) **International Filing Date:**
10 December 2014 (10.12.2014)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).
- (72) **Inventors:** MILOJICIC, Dejan S; 1501 Page Mill Rd., Palo Alto, California 94304 (US). SCHUMACHER, Derek; 8000 Foothills Blvd., Roseville, California 95747 (US). WANG, Zhikui; 1501 Page Mill Rd., Palo Alto, California 94304 (US).
- (74) **Agents:** BELL, Scott 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, LI, 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).

[Continued on next page]

- (54) **Title:** FIRMWARE-BASED PROVISIONING OF OPERATING SYSTEM RESOURCES



(57) **Abstract:** Examples relate to firmware-based provisioning of hardware resources. In some of the examples, firmware discovers and takes ownership of a hardware resource. At this stage, the firmware performs a test to verify the hardware resource. The firmware then assigns the hardware resource to an OS instance. At this stage, the firmware can suspend assigning further hardware resources to the OS instance in response to a satisfied notification from the OS instance.

FIG. 1

**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))*

FIRMWARE-BASED PROVISIONING OF OPERATING SYSTEM RESOURCES

BACKGROUND

[0001] In many-core, rack-size systems with large amounts of high-performance, high-density, low-cost non-volatile memory, resource provisioning can be quite complex. Further, such systems can be partitioned and run different operating systems (OS's). An OS holds ownership of a set of resources after the OS is booted.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] The following detailed description references the drawings, wherein:

[0003] FIG. 1 is a block diagram of an example server computing device for firmware-based provisioning of hardware resources;

[0004] FIG. 2 is a block diagram of an example system including multiple server computing devices for firmware-based provisioning of hardware resources;

[0005] FIG. 3 is a flowchart of an example method for execution by a server computing device for firmware-based provisioning of hardware resources; and

[0006] FIG. 4 is a flowchart of an example method for execution by a server computing device for firmware-based provisioning of hardware resources.

DETAILED DESCRIPTION

[0007] As discussed above, the management of systems' resources becomes more complex as computer systems grow in size and complexity. In particular, there are numerous (potentially heterogeneous) cores, multiple heterogeneous memories (dynamic random access memory (DRAM), non-volatile memory (NVM)) that are distributed across local and remote nodes, and rack-size computers that can cross multiple rack boundaries. Firmware initially controls the system, tests all resources, boots an OS, and then passes the resource control to the OS. During operation, firmware may occasionally execute some of the

privileged functionality on behalf of the OS, but the firmware does not have an active role.

[0008] Examples described herein include firmware that plays an active role in managing and owning all resources and gradually provisioning resources to OS's as requested by the OS's. Further, the firmware can also recover resources when they are no longer used by the OS or according to some resource sharing policy.

[0009] In some examples, firmware discovers and takes ownership of a hardware resource. Next, the firmware performs a test to verify the hardware resource while an OS instance is booting. The firmware then assigns the hardware resource to the OS instance. At this stage, the firmware can suspend assigning further hardware resources to the OS instance in response to a satisfied notification from the OS instance.

[0010] Referring now to the drawings, FIG. 1 is a block diagram of an example server computing device 100 for firmware-based provisioning of hardware resources. The example server computing device 100 may be implemented as a rack-mount server, a desktop computing device, server, or other computing device capable of providing multiple OS environments. In the example of FIG. 1, server computing device 100 includes OS instance(s) 110, firmware 112, hardware resource(s) 114, interface 116, and machine-readable storage medium 120.

[0011] OS instance 110 is partitioned computing system within server computing device 100 that provides the architecture and functions of a separate computer system. Any number of OS instances 110 can be included in server computing device 100, where each OS instance 110 executes as if it were a separate computer system (i.e., hardware resources 114 are partitioned between the OS instances 110). Each OS instance may have varied characteristics such as different operating systems, different proportions of hardware resources, different system requirements, different hardware profiles, etc.

[0012] Firmware 112 includes persistent memory and executable code that is embedded in a computing component such as a basic input/output system (BIOS). Examples of persistent memory used for firmware 112 include, but are not limited to, read-only memory (ROM), erasable programmable read only memory

(EPROM), and flash memory. The persistent memory is programmed using a firmware image that is flashed or otherwise installed onto the firmware 112. BIOS firmware typically includes elementary functions for booting and initializing a computing device; however, firmware 112 has been expanded as described below to provide additional functionality for dynamically provisioning hardware resources 114.

[0013] Hardware resources 114 can include various resources of server computing device 100 that can be assigned to OS instances 110. Examples of hardware resources 114 include, but are not limited to, processors/cores, memory, networking ports, etc. In some cases, hardware resources 114 can be virtualized such that a single hardware resource can be virtually shared among multiple OS instances 110. A processor may be one or more central processing units (CPUs), microprocessors, and/or other hardware devices suitable for retrieval and execution of instructions.

[0014] Interface 116 may include a number of electronic components for communicating with other server devices. For example, interface 116 may be an Ethernet interface, a Universal Serial Bus (USB) interface, an IEEE 1394 (Firewire) interface, an external Serial Advanced Technology Attachment (eSATA) interface, or any other physical connection interface suitable for communication with the other server devices. In another example, interface 116 may be a wireless interface.

[0015] Machine-readable storage medium 120 may be any electronic, magnetic, optical, or other physical storage device that stores executable instructions. Thus, machine-readable storage medium 120 may be, for example, Random Access Memory (RAM), ROM, EEPROM, flash memory, a storage drive, an optical disc, and the like. As described in detail below, machine-readable storage medium 120 may be encoded with executable instructions for firmware-based provisioning of hardware resources. In this example, instructions stored on machine-readable storage medium 120 are executed by a processor as described below.

[0016] Hardware resource discovering instructions 122 may discover hardware resources 114 while server computing device 100 is booting. BIOS firmware 112 discovers and tests hardware resources 114 during boot to determine if the hardware resources 114 are operating properly. Similarly, in this example, the firmware 112 locates and examines hardware resources 114 during boot to discover the amount and characteristics of the available hardware resources 114.

[0017] Hardware resource verifying instructions 124 may examine the hardware resources 114 to confirm they are operating properly. Specifically, firmware 112 may test RAM of server computing device 100 to determine if the RAM is operational (i.e., can be written to and read from). Firmware 112 may also test processors of server computing device 100 to ensure the processors are able to execute instructions. If hardware resources 114 are not operating properly, firmware 112 may interrupt the booting of server computing device 100 and post an appropriate error code that identifies the issue.

[0018] Hardware resource assigning instructions 126 assigns hardware resources 114 to OS instances 110. While server computing device 100 is booting, firmware 112 allocates and assigns subsets of hardware resources 114 to specific OS instances 110. For example, the assignments can be made by firmware 112 according to configurations specified by an administrator of server computing device 100. In another example, OS instances 110 may request a quantity of resources 114, where firmware 112 satisfies the requests by appropriately assigning resources from the resource pool. In this manner, the different requirements of each OS instance 110 can be satisfied by the assignment of hardware resources 114 while the server computing device 100 is booting.

[0019] FIG. 2 is a block diagram of an example system 200 including server computing devices (e.g., server computing device A 202A, server computing device N 202N) that allow for firmware-based provisioning of hardware resources. The components of the server computing devices may be similar to the corresponding components of server computing device 100 described with respect to FIG. 1.

[0020] As illustrated, server computing device A 202A includes integrated circuits (i.e., chips) 208A, 208B, firmware 212, interface 216, and hardware resources 224. Interface 216 and hardware resources 224 may be similar to the corresponding components of server computing device 100 that are described above with respect to FIG. 1. Each integrated circuit 208A, 208B includes an OS instance 210A, 210B, respectively, which may be similar to the corresponding OS instances 110 of server computing device 100 of FIG. 1. Firmware 212 may be similar to the corresponding firmware 112 of server computing device 100 of FIG. 1.

[0021] In this example, firmware 212 uses interface 216 to communicate with other server computing devices (e.g., server computing device N 202N, etc.). Integrated circuit A 208A corresponds to a management circuit because it includes firmware 212. Firmware 212 is configured to allocate resources to different OS instances 210A, 210B, 210N in different integrated circuits (e.g., integrated circuit B 208B, etc.) server computing device A 202A and even remote integrated circuits (e.g., integrated circuit 208N, etc.) of other server computing device N 202N. Specifically, firmware 212 can allocate resources dynamically while server computing device 202A is booting and/or after boot while OS instances 210A, 210B are running. In other words, firmware 212 is configured to centrally manage hardware resources 224 of server computing device A 202A. For example, firmware 212 may disable hardware resources 224 to optimize for power saving, allocate hardware resources 224 to different OS instances 210A, 210B, 210N to balance the load, set hardware resources 224 to be idle or deactivated for maintenance purpose, react to security threats, etc. In some cases, firmware 212 can also be configured to manage hardware resources of other server computing devices such as server computing device N 202N.

[0022] In some cases, firmware 212 allocates hardware resources 224 by invoking low-level commands to either assign resources to or recapture resources from OS instances 210A, 210B, 210N. Firmware 212 may recapture hardware resources 224 from an OS instance by simply reassigning the hardware resources 224 to a different OS instance. Further, such functions or primitives as described

above can be accomplished through techniques, such as ballooning to regain ownership of memory or containers and partitioning techniques to regain the ownership of cores.

[0023] In other cases, OS instances 210A, 210B, 210N may request hardware resources 224 actively and firmware 212 can decide if the requests should be satisfied. In this case, the OS instances 210A, 210B, 210N can release hardware resources 224 back to resource pools that are managed by firmware 212.

[0024] In yet other cases, firmware 212 may be configured to incrementally test hardware resources 224 and pass the hardware resources 224 as soon as they are tested to the OS instances 210A, 210B, 210N during the bootstrap process to optimize performance and improve the bootstrap time (i.e., availability of the system). In such cases, firmware 212 testing the system is accomplished in parallel with the OS instances 210A, 210B, 210N booting. In other words, OS instances 210A, 210B initially run with minimal resources and then gradually receive additional resources 224 from firmware 212 as it continues to test and configure the hardware resources 224.

[0025] In yet other cases, firmware 212 may be configured to gradually bootstrap an OS instance A 210A from increasingly further memory locations with respect to a management core of the OS instance A 210A (i.e., firmware 212 initially assigns resources that are closest to OS instance A 210A and then gradually begins to assign resources that are farther away from OS instance A 210A). Initially, firmware 212 boots OS instance A 210A from the non-volatile memory (NVM), which is on the same integrated circuit 208A as the management core of OS instance A 210A. But instead (if for example that memory has failed) or in addition (due to the size, or subsequent changes), the bootstrap can be redirected to NVM on a different integrated circuit 208B. Similarly, the bootstrap can be redirected even further to NVM on yet another server computing device N 202N to which the management core has load-store access. In each case, security and accessibility should be established (e.g., access rights and mapping of a memory management unit (MMU) of server computing device A 202A should

be set to be able to access memory on an a different integrated circuit than OS instance A 210A or on a different server).

[0026] FIG. 3 is a flowchart of an example method 300 for execution by a server computing device 100 for firmware-based provisioning of hardware resources. Although execution of method 300 is described below with reference to server computing device 100 of FIG. 1, other suitable devices for execution of method 300 may be used, such as server computing devices 202A, 202N of FIG. 2. Method 300 may be implemented in the form of executable instructions stored on a machine-readable storage medium, such as computer readable medium 120 of FIG. 1, and/or in the form of electronic circuitry.

[0027] Method 300 may start in block 305 and continue to block 310, where firmware of server computing device 100 discovers and takes ownership of hardware resources. In block 315, the firmware verifies the hardware resources to confirm they are operating properly.

[0028] In block 320, the firmware assigns hardware resources to OS instances of server computing device 100. Server computing device 100 may include any number of OS instances, and the firmware may apportion the hardware resources between the OS instances dynamically based on system requirements and/or statically based on a configuration file. In block 325, the firmware may suspend assigning further hardware resources to a particular OS instance in response to a satisfied notification from the machine. In other words, the firmware is configured to incrementally assign resources to the OS instance until the OS instance sends the satisfied notification. In other examples, the firmware may suspend assigning further hardware resources based on a preconfigured policy such as a load-balancing policy. Method 300 may then proceed to block 335, where method 300 stops.

[0029] FIG. 4 is a flowchart of an example method 400 for execution on a server computing device 202A for firmware-based provisioning of hardware resources. Although execution of method 400 is described below with reference to server computing device 202A of FIG. 2, other suitable devices for execution of method 400 may be used, such as server computing device 100 of FIG. 1.

Method 400 may be implemented in the form of executable instructions stored on a machine-readable storage medium and/or in the form of electronic circuitry.

[0030] In block 410, firmware of server computing device 202A discovers and takes ownership of hardware resources. In block 415, the firmware verifies the hardware resources to confirm they are operating properly.

[0031] In block 420, the firmware assigns hardware resources to an OS instance of server computing device 202A. Server computing device 202A may include any number of OS instances, and the firmware may apportion the hardware resources to the OS instance dynamically based on system requirements and/or statically based on a configuration file. In block 425, the firmware determines if the OS instance is satisfied with the hardware resources it has been assigned. For example, the firmware may determine if a satisfaction notification has been received from the OS instance. In another example, the firmware may determine if the OS instance is satisfied based on a preconfigured policy.

[0032] If the OS instance is not satisfied, method 400 may return to block 415 to continue verifying hardware resources for assignment. In this example, hardware resources are assigned as they are verified to reduce boot time (i.e., hardware resources are verified and assigned in parallel). Accordingly, hardware resources can be incrementally assigned to the OS instance as they are verified in segments.

[0033] If the OS instance is satisfied, method 400 proceeds to block 430, where the firmware may determine if a release notification has been received. For example, the OS instance may send a release notification when it no longer needs hardware resources because the OS instance's load has decreased or because the OS instance is suspending or shutting down. In another example, the firmware can initiate recapturing the hardware resources based on a preconfigured policy (e.g., load-balancing policy, etc.). If a release notification is received, the firmware adds the previously assigned hardware resources back to a resource pool in block 435. Because the hardware resources have been assigned to the resource pool, the firmware can then assign the hardware resources to a

different OS instance in block 440. In this manner, the firmware continues to centrally manage the de-allocation and allocation of hardware resources between OS instances. In block 445, method 400 may stop.

[0034] The foregoing disclosure describes a number of examples for firmware-based provisioning of hardware resources. In this manner, the examples disclosed herein facilitate centralized management of hardware resources between OS instances by configuring firmware of a server to manage the hardware resources during and after bootstrapping.

CLAIMSWe claim:

1. A system for firmware-based provisioning of hardware resources, comprising:

a hardware resource;

an OS instance operatively connected to the hardware resource;

firmware operatively connected to the hardware resource, the firmware

to:

discover and take ownership of the hardware resource;

perform a test to verify the hardware resource;

assign the hardware resource to the OS instance; and

in response to a release notification, add the hardware resource to a resource pool for allocation to other OS instances.

2. The system of claim 1, wherein the firmware is further to recapture the hardware resource from the OS instance.

3. The system of claim 1, wherein the firmware is further to redirect a bootstrap of the OS instance to memory on a different chip than the OS instance.

4. The system of claim 1, wherein the firmware is further to, in response to a satisfied notification from the OS instance, suspend assigning further hardware resources to the OS instance.

5. The system of claim 1, wherein the OS instance is located on a different chip than the firmware.

6. The system of claim 1, wherein the OS instance is one of a plurality of OS instances, and wherein the plurality of OS instances are distributed across a plurality of server computing devices.

7. A method for firmware-based provisioning of hardware resources, comprising:

discovering, by firmware, and taking ownership of a hardware resource;

performing, by the firmware, a test to verify the hardware resource;

assigning, by the firmware, the hardware resource to an OS instance of a plurality of OS instances;

adding, by the firmware, the hardware resource to a resource pool for allocation to other OS instances in response to a release notification; and

revoking, by the firmware, access to the hardware resource by the OS instance.

8. The method of claim 7, further comprising redirecting, by the firmware, a bootstrap of the OS instance to memory on a different chip than the firmware.

9. The method of claim 7, further comprising, suspending, by the firmware, assigning further hardware resources to the OS instance in response to a satisfied notification from the OS instance.

10. The method of claim 7, wherein the OS instance is located on a different chip than the firmware.

11. The method of claim 7, wherein the plurality of OS instances are distributed across a plurality of server computing devices.

12. A non-transitory machine-readable storage medium encoded with instructions executable by a processor for firmware-based provisioning of hardware resources, the machine-readable storage medium comprising instructions to:

discover, by firmware, and take ownership of a hardware resource;

perform, by the firmware, a test to verify the hardware resource;

assign, by the firmware, the hardware resource to an OS instance of a plurality of OS instances that are distributed across a plurality of server computing devices; and

suspend, by the firmware, assigning further hardware resources to the OS instance in response to a satisfied notification from the OS instance.

13. The non-transitory machine-readable storage medium of claim 12, wherein the instructions are further to recapture, by the firmware, the hardware resource from the OS instance.

14. The non-transitory machine-readable storage medium of claim 12, wherein the instructions are further to redirect, by the firmware, a bootstrap of the OS instance to memory on a different chip than the firmware.

15. The non-transitory machine-readable storage medium of claim 12, wherein the instructions are further to add, by the firmware, the hardware resource to a resource pool for allocation to other OS instances in response to a release notification from the OS instance.

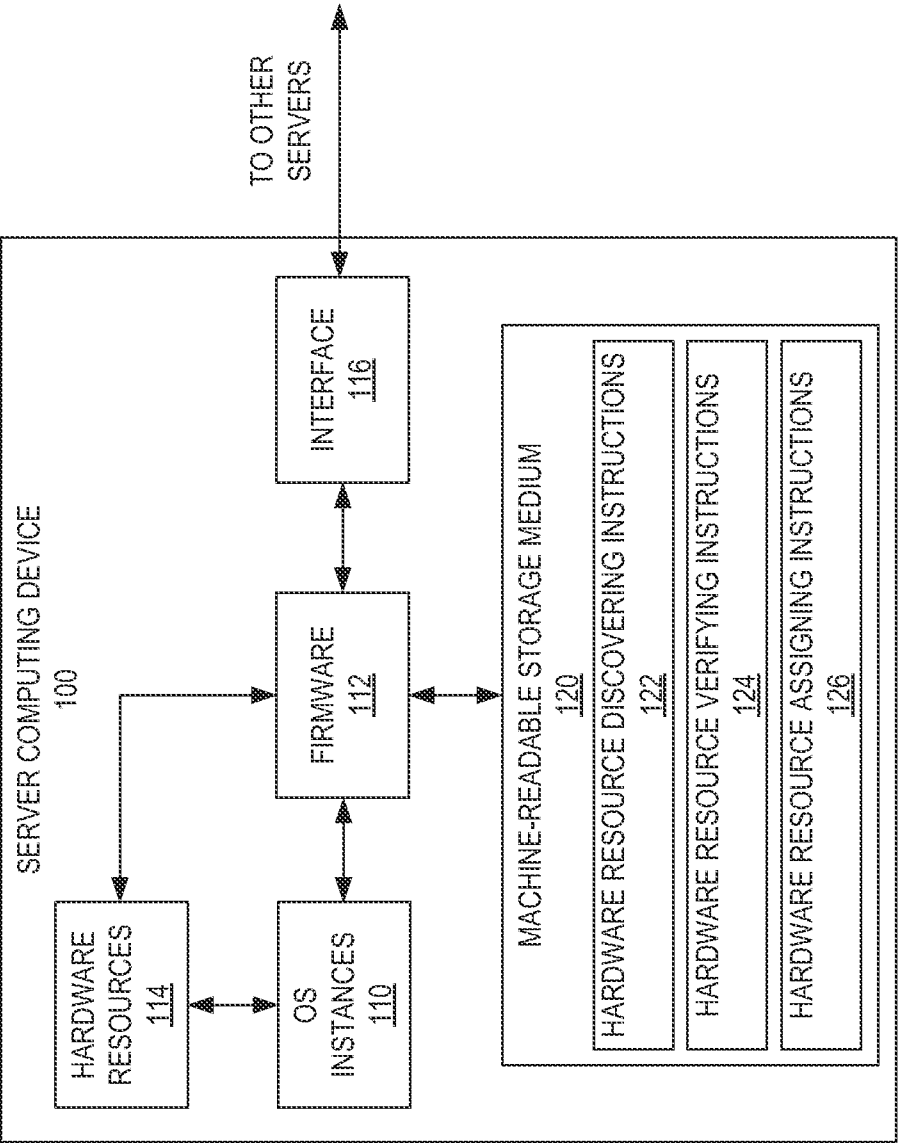


FIG. 1

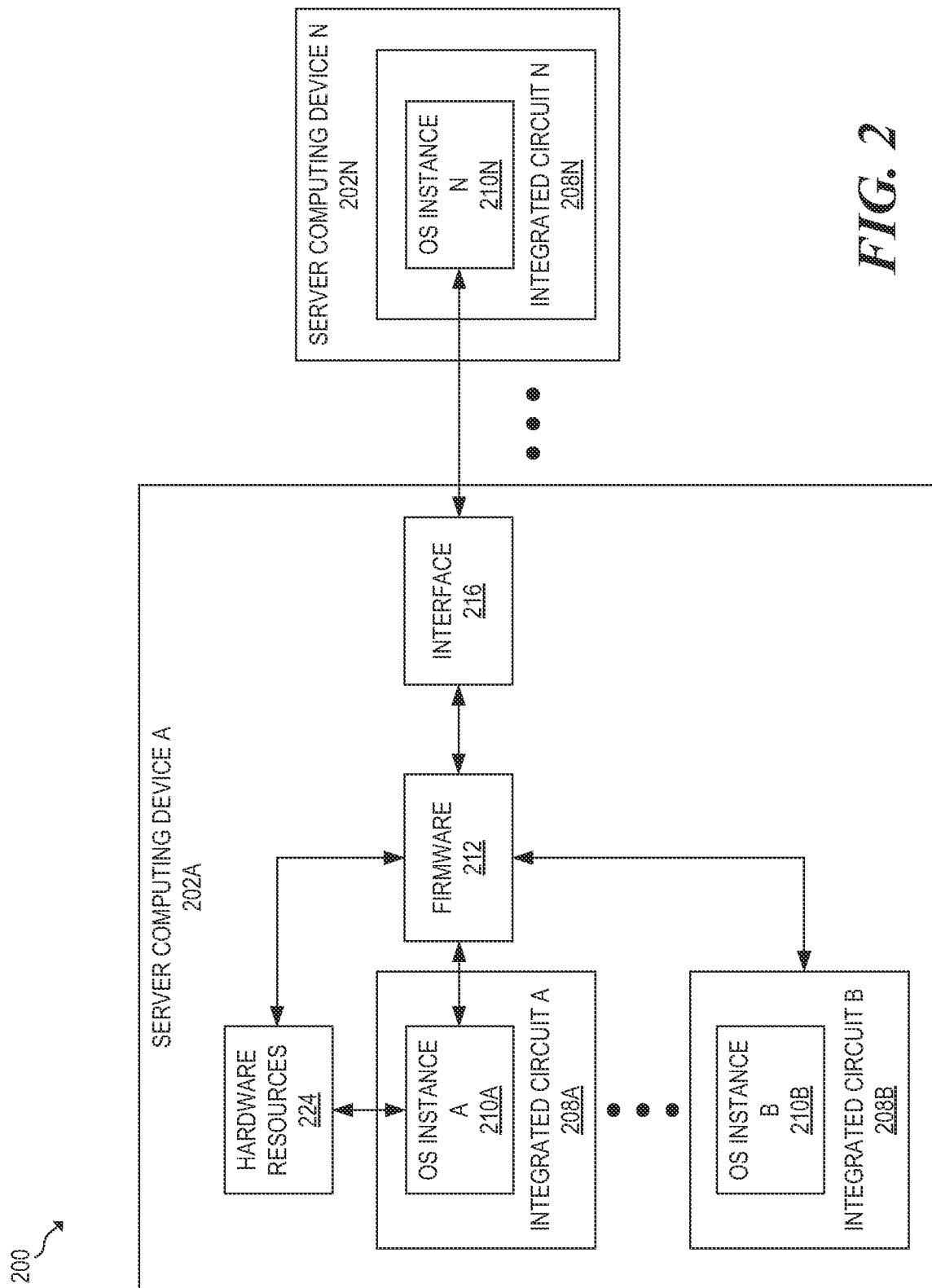
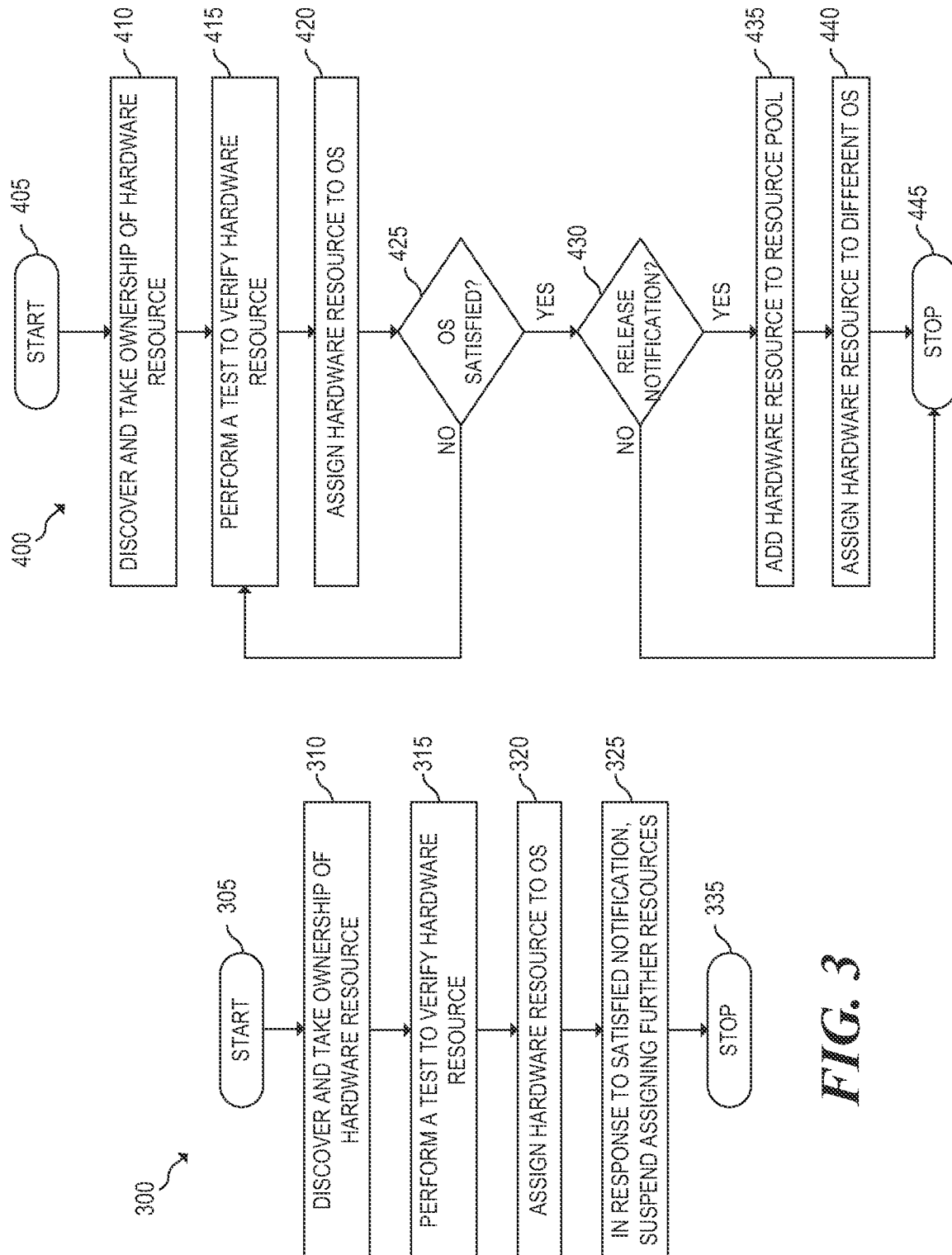


FIG. 2



INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/069625**A. CLASSIFICATION OF SUBJECT MATTER****G06F 9/44(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 9/44; G06F 15/173; G06F 15/177; G06F 9/50

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: firmware, resource, assign

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2008-0216085 A1 (ARNDT RICHARD LOUIS et al.) 04 September 2008	1-3,5-8,10-11
A	See abstract; paragraphs [0046]-[0049], [0081]-[0104]; and figures 7, 17-18.	4,9,12-15
Y	US 2012-0158923 A1 (MOHAMED ANSARI et al.) 21 June 2012	1-3,5-8,10-11
A	See abstract; paragraphs [0015]-[0051]; and figures 1-5.	4,9,12-15
A	US 2012-0096167 A1 (FREE JAMES DOUGLAS et al.) 19 April 2012	1-15
	See abstract; paragraphs [0017]-[0035]; and figures 1-7.	
A	US 2011-0179422 A1 (STEGARU CRISTIAN et al.) 21 July 2011	1-15
	See abstract; paragraphs [0015]-[0041]; and figures 1-7.	



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

21 July 2015 (21.07.2015)

Date of mailing of the international search report

21 July 2015 (21.07.2015)

Name and mailing address of the ISA/KR

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

Facsimile No. +82-42-472-7140

Authorized officer

CHOI, Jeong Kwon

Telephone No. +82-42-481-8507



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2014/069625

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2008-0216085 A1	04/09/2008	US 2006-0212620 A1 US 7376770 B2 US 8028105 B2	21/09/2006 20/05/2008 27/09/2011
US 2012-0158923 A1	21/06/2012	None	
US 2012-0096167 A1	19/04/2012	None	
US 2011-0179422 A1	21/07/2011	None	