



- (51) International Patent Classification:
G06F 9/50 (2006.01) *G06F 9/46* (2006.01)
- (21) International Application Number:
PCT/US2015/014008
- (22) International Filing Date:
31 January 2015 (31.01.2015)
- (25) Filing Language: English
- (26) Publication Language: English
- (71) Applicant: HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).
- (72) Inventor: WILKINSON, Lyle Eric; 1501 Page Mill Rd., Palo Alto, California 94304 (US).
- (74) Agents: ADEKUNLE, Olaolu et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (US).

- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

[Continued on next page]

- (54) Title: RESOURCE ALLOCATION

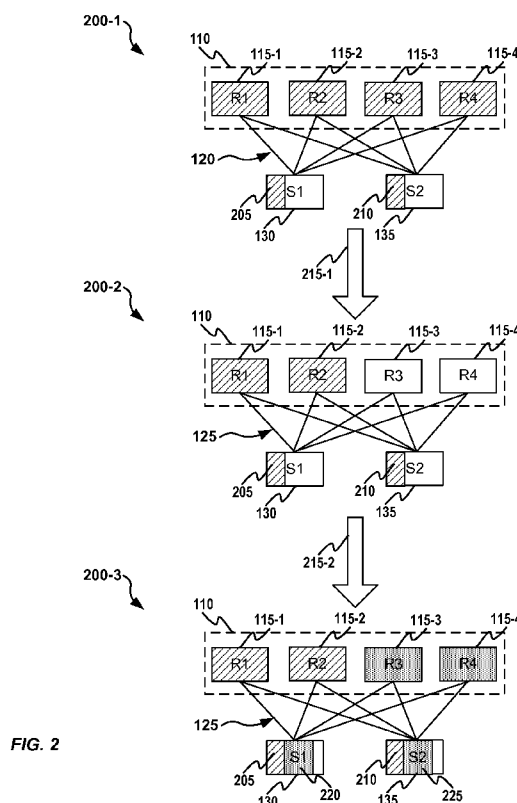


FIG. 2

[Continued on next page]

**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))*

(57) Abstract: An example method includes determining resources needed for activation of a new application; identifying redundant resources currently available; identifying shared resources with inter-relationship with available redundant resources; freeing shared resources with inter-relationship with available redundant resources from existing applications currently using the shared resources; and allocating freed shared resources and available redundant resources to new application.

RESOURCE ALLOCATION

BACKGROUND

[0001] Typical solutions for resource partitioning or allocation entail careful administrator preplanning and deliberate (manual) actions. Introduction of a new application to a complex computer system, e.g. a proof of concept, may require utilization of new or unused resources (e.g. server, storage, networking hardware). An administrator may identify the availability of resources and potentially free up resources to allow introduction of a new application.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] For a more complete understanding of various examples, reference is now made to the following description taken in connection with the accompanying drawings in which:

[0003] Figure 1 illustrates an example computer system topological map;

[0004] Figure 2 illustrates the example topological map at various stages of an example resource allocation;

[0005] Figure 3 illustrates an example computing device for resource allocation; and

[0006] Figure 4 illustrates an example flow diagram of an example process for resource allocation.

DETAILED DESCRIPTION

[0007] Example systems and methods described herein utilize a computer system topology configuration template, or topological map, for a complex system and analyze the configuration topology map for allocation of resources to an application. The example management system receives a set of resource requirements for a new application and uses its knowledge of the complex system's redundant and shared resources to identify resources that can be reallocated to the new application, while continuing the functionality of existing applications that are currently active on the computer system. The example management system may repurpose a subset of components (e.g., server, networking, storage, etc.) for the new application. Preexisting applications on the complex system may continue operating, perhaps at a reduced performance level, while enabling activation of the new application. In various examples, resources may be optimized to support both the preexisting applications as well as the new application.

[0008] Often, it may be desirable to redeploy a portion of redundant allocated resources to another application or potentially a new version of the same application without eliminating the existing application. However, reallocating a portion of the existing allocated resources to the new application is error prone and administrator time intensive.

[0009] As described herein, providing software plugin methods to free existing allocated resources (redundant and/or shared resources) and allocating a portion of these freed resources to the new application enables a generalized method to redeploy a portion of a complex system. A suitable topological description of the redundant and shared resources enables invocation of these plugin methods in a manner to minimize risk to the existing applications (potentially at lower performance and with less availability) while facilitating introduction of the new application.

[0010] Referring now to the figures, Figure 1 illustrates an example computer system topology 100 used by systems and methods described herein. Data indicative of the computer system topology 100 may be stored in memory including descriptive parameters such as, by way of example, an identifier of a component, such as a unique identifier, name, serial number, etc.; functioning or type of that component, including a purpose of that component within the system; dependency of each component relative to other components in the system; whether the component in the system is redundant to another component in the system; and whether the component is a shared resource providing services to multiple other components, etc.

[0011] The example computer system topology 100 includes a redundancy set 110 including redundant resources 115 such as, for example, servers and/or virtual machine hosts. A virtual machine hosts multiple virtual machines on a single device, allowing multiple operating systems running on the same device, for example. The example redundancy set 110 illustrated in Figure 1 includes four redundant resources 115 including a first redundant resource 115-1, a second redundant resource 115-2, a third redundant resource 115-3 and a fourth redundant resource 115-4.

[0012] The redundant resources 115 of the example redundancy set 110 may have an inter-relationship with two shared resources, as indicated by the dependency lines 125. The shared resources, in this example, include a first shared resource 130 and a second shared resource 135. The first shared resource 130 may be, for example, a network switch that provides a plurality of switch ports for individual applications running on the redundant resources 115. The second

shared resource 135 may be, for example, a storage array that provides memory for the individual applications running on the redundant resources 115. The example computer system topology 110 is only an example, and other topologies may have more or fewer redundant resources 115, more or fewer shared resources and/or more layers of redundant and/or shared resources.

[0013] Figure 2 illustrates the example topological map of Figure 1 at various stages of resource allocation in the computer system. The example computer system illustrated in Figure 2 is shown operating with a first resource allocation 200-1. In this example, a single current application is running on the redundant resources 115, as indicated by the angled lines highlights. In addition, the current application requires a first portion 205 of the first shared resource 130 and a first portion 210 of the second shared resource 135 to be allocated to the current application.

[0014] While the computer system is operating using the first resource allocation 200-1, a request that a new application is to be activated may be received. Upon receiving the request for the new application, a computing device running a resource allocation manager (e.g., a resource allocation manager 330 described in reference to Figure 3 below) determines the required resources needed to activate and process the new application. The required resources for the new application, as well as for existing applications, may be determined based on input from an administrator or based on parameters embedded in the computer code of the new application.

[0015] In the example of Figure 2, the new application requires two of the redundant resources 115 and about half of each of the first and second shared resources 130 and 135 (e.g., 2 networks at 1GB/s of a network switch, and 100GB of memory of a storage array). In this example, the current application being run on the computer system of Figure 2, requires only two of the redundant resources 115, and only a fraction (about a third) of each of the shared resources 130 and 135 to continue running at a safe level, which may be lower than a desired or optimal level. This may be determined by the resource allocation manager based on data previously obtained from the redundant resources 115 and from the first and second shared resources 130 and 135, as described below.

[0016] Upon determining that two of the redundant resources 115 are available, and that about two thirds of the first and second shared resources 130 and 135 are available for activating the new application, the resource allocation manager initiates a first resource reallocation 215-1

freeing up the third and fourth redundant resources 115-3 and 115-4, as indicated in a second stage of the resource allocation 200-2 by the removal of the angled line highlights. This first resource reallocation 215-1 may include the resource allocation manager sending a request signal to the third and fourth redundant resources 115-3 and 115-4 and the first and second shared resources 130 and 135, the request signal requesting that the redundant resources 115-3 and 115-4 currently allocated to the current application be freed-up and allocated to the new application and that portions of the first and second shared resources 130 and 135 also be allocated to the new application.

[0017] In response to sending the request signal, the example resource allocation manager receives response signals from the third and fourth redundant resources and from the first and second shared resources 130 and 135 indicating that the requested resources can be or have been freed-up and can be allocated to the new application. In response to receiving the response signals, the example resource allocation manager initiates a second reallocation 215-2 in which the third and fourth redundant resources 115-3 and 115-4 and half of each of the first and second shared resources 130 and 135 (a second portion 220 of the first shared resource 130 and a second portion 225 of the second shared resource 135, in this example) are allocated to the new application, as indicated by the portions highlighted with shaded area in a third stage of the resource allocation 200-3. The reallocation examples 200 illustrated in Figure 2 are very simple examples, but more complicated reallocations may be performed using the systems and methods described herein.

[0018] Referring now to Figure 3, an example computing device 300 that may utilize an example resource allocation manager 330 for allocating the redundant and shared resources of a computer system topology, such as the example computer system topology 100 of Figure 1 and as illustrated in Figure 2, is illustrated. The example computing device 300 may include embedded firmware and hardware components in order to continually collect data associated with amounts of redundant resources allocated to current active applications, and amounts of shared resources allocated to current active application, from the components from a computer system such as the components illustrated in Figures 1 and 2. The example computing device 300 may be a standalone server such as a blade server, a storage server or a switch, for example. The example computing device 300 may include a server CPU (central processing unit) 310, at least one memory device 320, and a power supply 340. The power supply 340 is coupled to an

electrical interface 345 that is coupled to an external power supply such as an AC power supply 350.

[0019] The example computing device 300 also includes an operating system component 355 including, for example, an operating system driver component and a pre-boot BIOS (Basic Input/Output System) component stored in ROM (read only memory), and coupled to the CPU 320. In various examples, the CPU 320 may have a non-transitory memory device 320. In various examples, the memory device 320 may be integrally formed with the CPU 120 or may be an external memory device. The memory device 320 may include program code that may be executed by the CPU 320. For example, one or more processes may be performed to execute a user control interface 375 and/or software applications 380. The memory device 320 also stores data 322 indicative of topological maps, data 324 for implementing an application tracker module, and parameter data 326.

[0020] The example topological map data 322 may be indicative of inter-relationships among a plurality of components of the computer system, the plurality of components including a plurality of components providing redundant resources and one or more components providing shared resources. The topological map data 322 may also include redundancy information identifying components of the plurality of components each capable of performing the same function and each having the same dependency relative to the computing device (e.g., the redundant resources 115 of Figures 1 and 2).

[0021] The example topology map data 322 may include topology map code representing a complex system. A complex system may be defined using various systems including, for example:

- A virtualization system, e.g. VMware vCenter/ESX or Microsoft SC VMM/HyperV;
- Database/Application system, e.g. SAP HANA;
- Large scale distributed processing, e.g. Hadoop.

[0022] An example of topology code that may be used to model the example computer system topology 100 of Figure 1 is presented below:

Sample Topology Code

System Definition Excerpt
System1

```

{
  Redundancy_Set: X1
  {
    Component: R1;
    Component: R2;
    Component: R3;
    Component: R4;
  }
  Component: S1;
  Component: S2;
}
R1
{
  Type: "Server";
}
R2
{
  Type: "Server";
}
R2
{
  Type: "Server";
}
R4
{
  Type: "Server";
}
S1
{
  Type: "NetworkSwitch";
}
S2
{
  Type: "StorageArray";
}

```

[0023] In this example code, the redundancy set 110 (labeled X1) includes four redundant servers R1, R2, R3 and R4 representing the four redundant resources 115. The example code also includes a network switch S1 representing the first shared resource 130 and a storage array S2 representing the second shared resource 135.

[0024] The example resource allocation manager 330 may be implemented in software, firmware and/or hardware. The example resource allocation manager 330 receives signals from the plurality of components in the computer system, the signals being associated with an amount

of redundant resources and an amount of shared resources currently allocated for one or more currently active applications. In addition, when the computing device 300 corresponds to one of the redundant resources 115 or one of the first and second resources 130 or 135, the resource allocation manager 330 may receive the signal from the application tracker module 324 that is implemented in the computing device 300.

[0025] When a new application is being added to the computer system, the example resource allocation manager 330 determines amounts of both redundant resources and shared resources needed by the new application in order to be activated on the computer system. The resource allocation manager 330 may determine the amounts of redundant resources and shared resources based on input received from the user control interface 375 (e.g., as input by an administrator) or from data embedded in the application software.

[0026] The example resource allocation manager 330 allocates a portion of the redundant resources and a portion of the shared services to the new application based on the received signals associated with the amount of redundant resources and amount of shared resources currently allocated for the one or more currently active application, and further based on the resources determined to be needed. The allocating may include freeing up a portion of the redundant resources currently allocated to the one or more currently active applications and allocating the freed-up portion of the redundant resources to the new application.

[0027] In various examples, the allocating may include the resource allocation manager 330 sending a request signal to one or more components requesting a portion of the redundant resources or a portion on the shared resources to be allocated to the new application. In response to the sending of the request signal, the resource allocation manager 330 may receive a response signal from respective components indicating whether or not the requested resources can be freed-up and allocated to the new application.

[0028] The allocating may also include using dependency information in the topological map data 322 to identify shared resources needed by the redundant resources for the currently active applications, and determining the shared resources that are available for the new application based on the dependency information.

[0029] The example computing device 300 may also include a display 360, a network interface 365 and other hardware 370 known to those skilled in the art. The network interface 365 may be coupled to a network such as an intranet, a local area network (LAN), a wireless

local area network (WLAN), the Internet, etc., where the other underlying components are a part of the network or at least coupled to the network

[0030] The display 360 may be used to display resource allocations of the redundant resources and of the shared resources for each active application. For example, displays similar to the resource allocations 200 illustrated in Figure 2 may be displayed to allow easy identification of the resources allocated to each application. In addition, the display 360 may display input screens of the user control interface 375 that may be configured to receive commands from a user or administrator inputting requirements for the redundant and shared resources required by a new application.

[0031] Figure 4 illustrates an example flow diagram for a process 400 for allocating redundant and shared resources of a computer system. The example process 400 of Figure 4 will now be described with further references to Figures 1 and 3.

[0032] In the illustrated example process 400 of Figure 4, the example resource allocation manager 330 may determine resources needed for a new application (block 410). In various examples, the resource allocation manager 330 may receive, via the user control interface 375, input representative of the required redundant resources and the required shared resources necessary to activate a new application. The required resources may be embodied in a form of programming code such as that described above. Receiving the user specified resource requirements may be optional, as pre-defined resource requirements may be embedded in computer program code embodying the new application, for example.

[0033] The example resource allocation manager 330 may identify redundant resources that are available (e.g., redundant resources that are currently not used or may be freed up from existing applications) (block 420). In this regard, the example resource allocation manager 330 may retrieve one or more topological maps from the topological map data 322 stored in the memory 320. The retrieved data is indicative of a topological map of the computer system for which the computing device 300 is allocating resources. The retrieved topological map data may include redundancy information identifying the redundant resources. The example resource allocation manager 330 may determine usage of the redundant resources indicated in the topological map, as well as identifying the resource requirements of existing applications. The resource allocation manager 330 may receive control signals from a plurality of components in a computer system. The control signals may be associated with an amount of redundant resources

currently allocated for one or more currently active applications running on the computer system. The control signals may be received from plugin application tracker modules running on each of the redundant resources and each of the shared resources, as described above. In various examples, plugin application tracker modules may track existing resource usage, as well as describe minimum and optimum requirements for an application.

[0034] The example resource allocation manager 330 may then identify shared resources with inter-relationship with the redundant resources identified as being available (block 430). The retrieved topological map data 322 may be indicative of inter-relationships among a plurality of components of the computer system, including a plurality of components in a redundancy set of resources and one or more shared resources. The interrelationships may further comprise dependency information identifying the shared resources needed by the redundant resources for the currently active applications.

[0035] The example resource allocation manager 330 may determine if the shared resources are available. As with the redundant resources, the example resource allocation manager 330 may receive control signals associated with an amount of shared resources currently allocated for one or more currently active applications running on the computer system. The control signals may be received from plugin application tracker modules running on each of the redundant resources and each of the shared resources, as described above. The control signals may also indicate requirements of the existing applications, allowing the example resource allocation manager 330 to determine which shared resources may be freed up.

[0036] If sufficient redundant resources and shared resources are available or can be freed up, the new application may be launched. In this regard, the example resource allocation manager 330 may free up the resources needed for the new application from the existing applications (block 440). In this regard, the plugin application tracker modules may cause the resources to be safely freed without disrupting operation of the existing applications.

[0037] At block 450, the example resource allocation manager 330 allocates a portion of the redundant resources and a portion of the shared services to the new application based on the signals associated with the amount of redundant resources and amount of shared services currently allocated for one or more currently active application, and further based on the resources determined to be needed, for example.

[0038] In various examples, the example resource allocation manager 330 may send a request signal to one or more components requesting a portion of the redundant resources and/or the shared resources to be allocated to the new application, and in response receive a response signal from respective components indicating whether the requested resources can be freed-up and allocated to the new application. In various examples, the response signal may be received from one of the components providing redundant resources and the response signal may include information regarding availability of a portion of a shared resource upon which the one component providing redundant resources depends.

[0039] When the interrelationships comprise redundancy information identifying the redundant resources and the interrelationships further comprise dependency information identifying the shared resources needed by the redundant resources for the currently active applications, the resource allocation manager 330 may further determine shared resources that are available for the new application based on the dependency information.

[0040] Various examples described herein are described in the general context of method steps or processes, which may be implemented in one example by a software program product or component, embodied in a machine-readable medium, including executable instructions, such as program code, executed by entities in networked environments. Generally, program modules may include routines, programs, objects, components, data structures, etc. which may be designed to perform particular tasks or implement particular abstract data types. Executable instructions, associated data structures, and program modules represent examples of program code for executing steps of the methods disclosed herein. The particular sequence of such executable instructions or associated data structures represents examples of corresponding acts for implementing the functions described in such steps or processes.

[0041] Software implementations of various examples can be accomplished with standard programming techniques with rule-based logic and other logic to accomplish various database searching steps or processes, correlation steps or processes, comparison steps or processes and decision steps or processes.

[0042] The foregoing description of various examples has been presented for purposes of illustration and description. The foregoing description is not intended to be exhaustive or limiting to the examples disclosed, and modifications and variations are possible in light of the above teachings or may be acquired from practice of various examples. The examples discussed

herein were chosen and described in order to explain the principles and the nature of various examples of the present disclosure and its practical application to enable one skilled in the art to utilize the present disclosure in various examples and with various modifications as are suited to the particular use contemplated. The features of the examples described herein may be combined in all possible combinations of methods, apparatus, modules, systems, and computer program products.

[0043] It is also noted herein that while the above describes examples, these descriptions should not be viewed in a limiting sense. Rather, there are several variations and modifications which may be made without departing from the scope as defined in the appended claims.

WHAT IS CLAIMED IS:

1. A device, comprising:

a memory storing data indicative of a topological map of a computer system, the topological map data being indicative of inter-relationships among a plurality of components of the computer system, the plurality of components including a plurality of redundant resources and at least one shared resource; and

a processor for executing a resource allocation manager, the resource allocation manager to:

determine resources needed for activation of a new application;

identify redundant resources currently available;

identify shared resources with inter-relationship with available redundant resources;

free shared resources with inter-relationship with available redundant resources from existing applications currently using the shared resources; and

allocate freed shared resources and available redundant resources to new application.

2. The device of claim 1, wherein freeing shared resources includes reducing running of at least one existing application to lower than a desired or optimal level.

3. The device of claim 1, wherein the inter-relationships comprise redundancy information identifying the redundant resources and dependency information identifying the shared resources needed by the redundant resources for the currently active applications.

4. The device of claim 1, wherein the plurality of redundant resources includes at least one blade server.

5. The device of claim 1, wherein the at least one shared resource includes at least one of a network switch or a shared storage array.

6. The device of claim 1, wherein the resource allocation manager:

sends a request signal to at least one component requesting a portion of the redundant resources and/or the shared resources to be allocated to the new application; and

receives a response signal indicating whether the requested resources can be freed-up from an existing application and allocated to the new application.

7. The device of claim 6, wherein one of the response signals is received from one of the redundant resources and the response signal includes information indicating availability of a portion of a shared resource upon which the one of the redundant resources depends.

8. The device of claim 1, wherein the resource allocation manager determines the redundant resources and the shared resources needed by the new application based on input received via a user control interface.

9. A method, comprising:

determining resources needed for activation of a new application;

identifying redundant resources currently available;

identifying shared resources with inter-relationship with available redundant resources;

freeing shared resources with inter-relationship with available redundant resources from existing applications currently using the shared resources; and

allocating freed shared resources and available redundant resources to new application.

10. The method of claim 9, wherein freeing shared resources includes reducing running of at least one existing application to lower than a desired or optimal level.

11. The method of claim 9, wherein the interrelationships comprise redundancy information identifying the redundant resources and dependency information identifying the shared resources needed by the redundant resources for the currently active applications.

12. The method of claim 9, further comprising:

sending a request signal to at least one component requesting a portion of the redundant

resources and/or the shared resources to be allocated to the new application; and

receiving a response signal indicating whether the requested resources can be freed-up from an existing application and allocated to the new application.

13. A computer program product, embodied on a non-transitory computer-readable medium, comprising:

computer code to determine resources needed for activation of a new application;

computer code to identify redundant resources currently available;

computer code to identify shared resources with inter-relationship with available redundant resources;

computer code to free shared resources with inter-relationship with available redundant resources from existing applications currently using the shared resources; and

computer code to allocate freed shared resources and available redundant resources to new application.

14. The computer program product of claim 13, wherein the computer code to free shared resources includes computer code to reduce running of at least one existing application to lower than a desired or optimal level.

15. The computer program product of claim 13, further comprising:

computer code to send a request signal to at least one component requesting a portion of the redundant resources and/or the shared resources to be allocated to the new application; and

computer code to receive a response signal indicating whether the requested resources can be freed-up from an existing application and allocated to the new application.

1/4

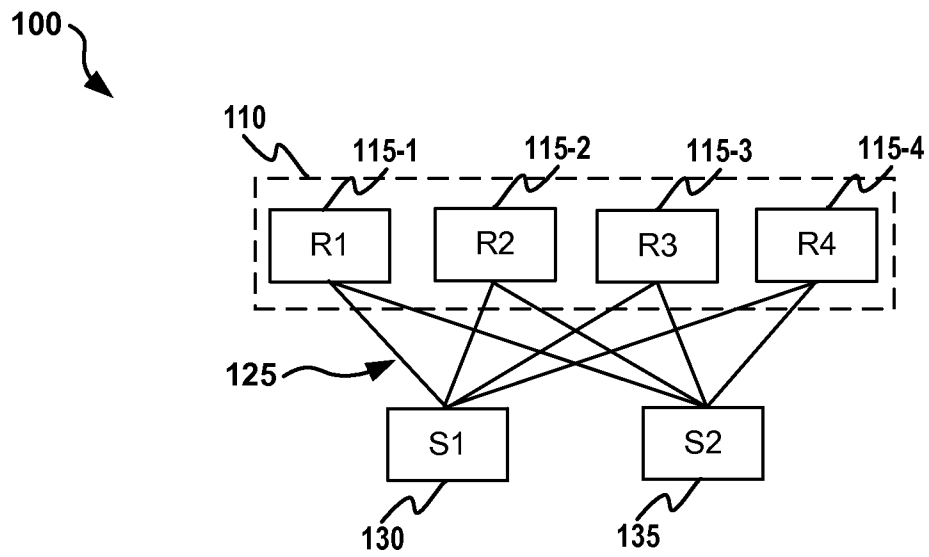


FIG. 1

2/4

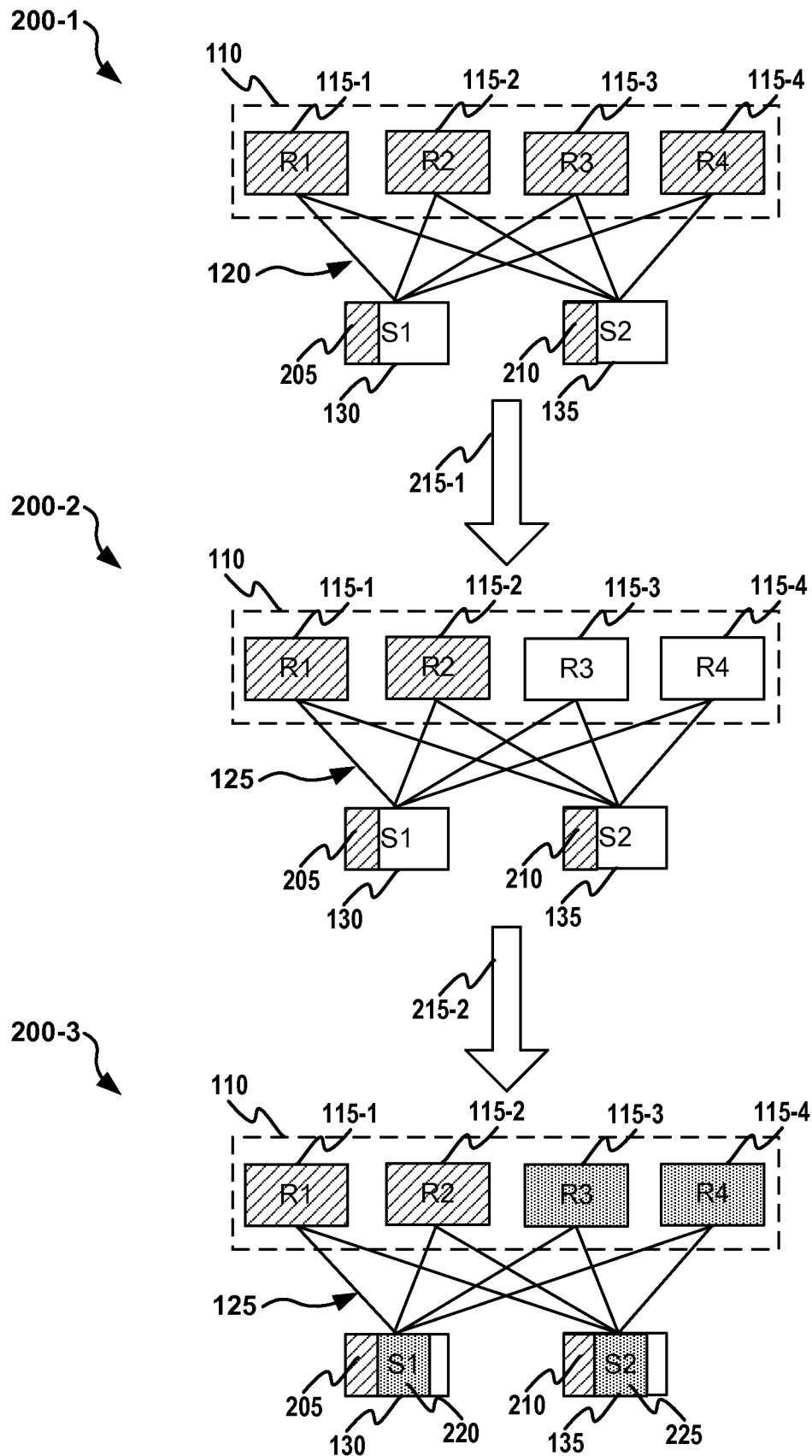


FIG. 2

3/4

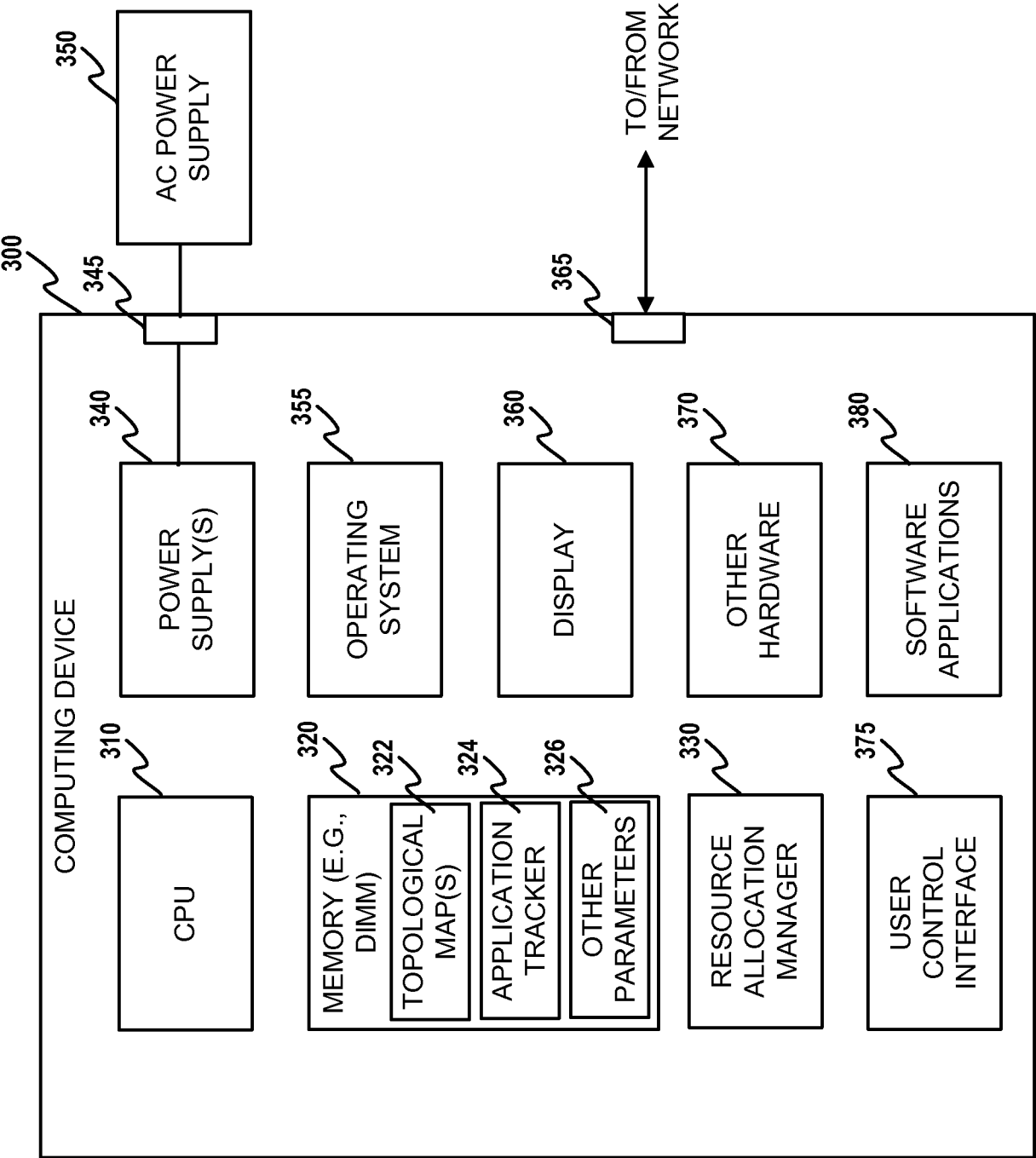
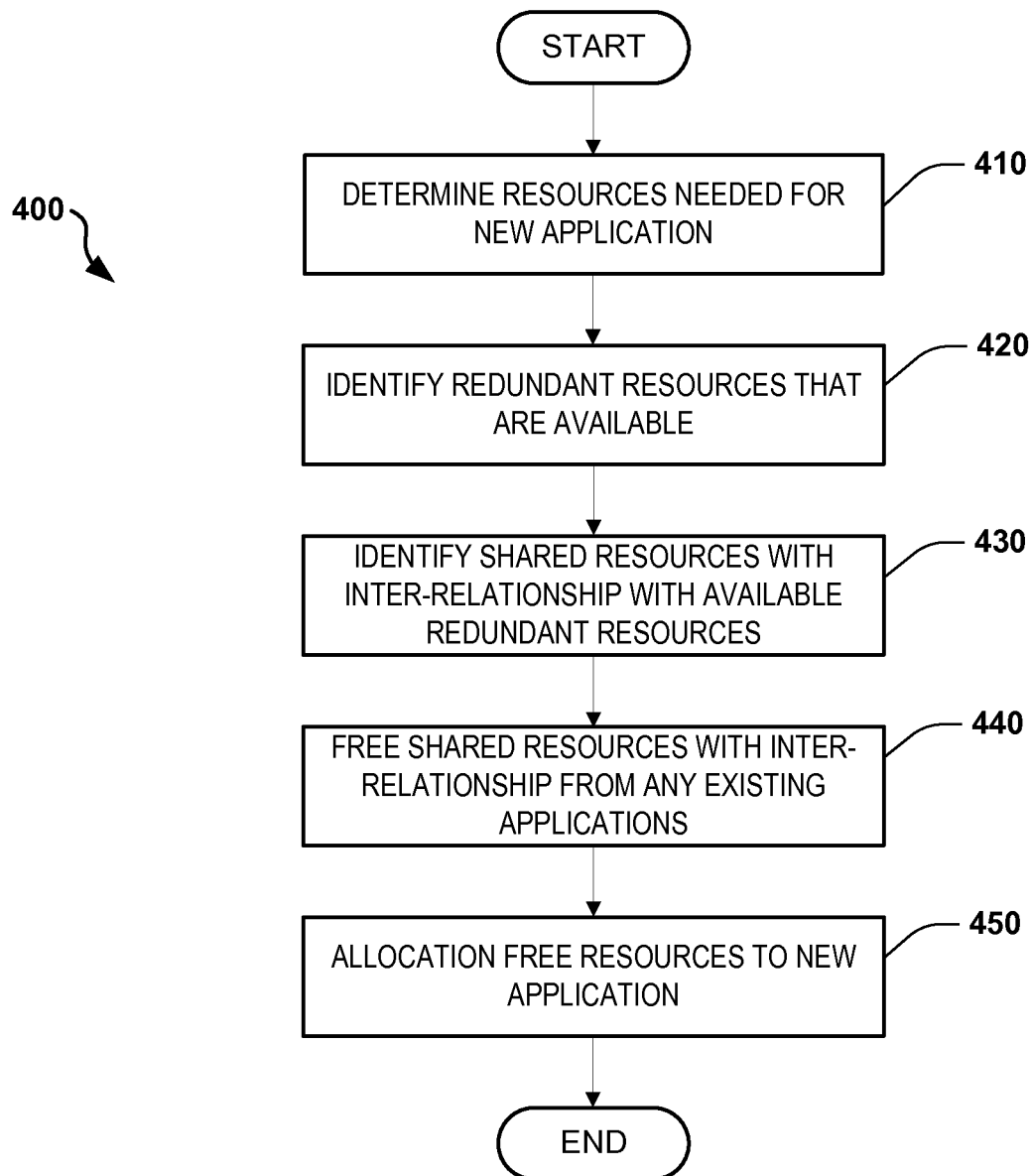


FIG. 3

4/4

**FIG. 4**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/014008**A. CLASSIFICATION OF SUBJECT MATTER****G06F 9/50(2006.01)i, G06F 9/46(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 9/50; G06F 9/48; G06F 9/46

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: resource allocation, availability, redundant resource, shared resource, free up

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2007-0094666 A1 (ODE, N.) 26 April 2007 See abstract; paragraphs [0022]-[0026],[0047]-[0052]; and figs. 2, 4, and 5.	1-15
A	US 2013-0290977 A1 (SIEMENS AKTIENGESELLSCHAFT) 31 October 2013 See abstract; paragraphs [0367]-[0041]; and claims 1 and 7.	1-15
A	US 2005-0283784 A1 (SUZUKI, T.) 22 December 2005 See abstract; fig. 1; and claims 1 and 8-10.	1-15
A	US 2014-0137121 A1 (HITACHI, LTD.) 15 May 2014 See abstract; paragraphs [0077]-[0100]; and fig. 4.	1-15
A	US 2010-0131959 A1 (SPIERS, A. Z. et al.) 27 May 2010 See abstract; paragraphs [0032]-[0039]; and fig. 2.	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

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

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

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

21 October 2015 (21.10.2015)

Date of mailing of the international search report

21 October 2015 (21.10.2015)

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. +82-42-472-7140

Authorized officer

YU, Jintae

Telephone No. +82-42-481-8530



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/014008

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2007-0094666 A1	26/04/2007	CN 1806228 A CN 1806228 C EP 1640861 A1 EP 1640861 A4 JP 2005-004350 A KR 10-1019308 B1 KR 10-2006-0023986 A US 7831975 B2 WO 2004-111847 A1	19/07/2006 27/02/2008 29/03/2006 13/02/2008 06/01/2005 07/03/2011 15/03/2006 09/11/2010 23/12/2004
US 2013-0290977 A1	31/10/2013	None	
US 2005-0283784 A1	22/12/2005	JP 2005-346433 A JP 4188281 B2 US 7779413 B2	15/12/2005 26/11/2008 17/08/2010
US 2014-0137121 A1	15/05/2014	WO 2014-054079 A1	10/04/2014
US 2010-0131959 A1	27/05/2010	None	