



US009270543B1

(12) **United States Patent**
Giardina et al.

(10) **Patent No.:** **US 9,270,543 B1**

(45) **Date of Patent:** **Feb. 23, 2016**

(54) **APPLICATION CENTERED NETWORK
NODE SELECTION**

(56) **References Cited**

U.S. PATENT DOCUMENTS

(71) Applicant: **CA, Inc.**, Islandia, NY (US)
(72) Inventors: **Nathan J. Giardina**, San Diego, CA (US); **David S. Tyree**, Denver, CO (US); **Robert Hucik**, Simi Valley, CA (US); **Beau Croteau**, Bay Shore, NY (US)

2009/0287825	A1*	11/2009	Walker et al.	709/226
2010/0162260	A1*	6/2010	Ibrahim	718/105
2011/0080850	A1*	4/2011	Klein et al.	370/254
2011/0231477	A1*	9/2011	Safruti et al.	709/203
2012/0198073	A1*	8/2012	Srikanth et al.	709/226
2012/0303807	A1*	11/2012	Akelbein et al.	709/224
2013/0346543	A1*	12/2013	Benantar et al.	709/217
2014/0215045	A1*	7/2014	Wang et al.	709/223

(73) Assignee: **CA, Inc.**, New York, NY (US)

* cited by examiner

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 250 days.

Primary Examiner — Mohamed Ibrahim

(74) *Attorney, Agent, or Firm* — Myers Bigel & Sibley, P.A.

(21) Appl. No.: **13/791,966**

(57) **ABSTRACT**

(22) Filed: **Mar. 9, 2013**

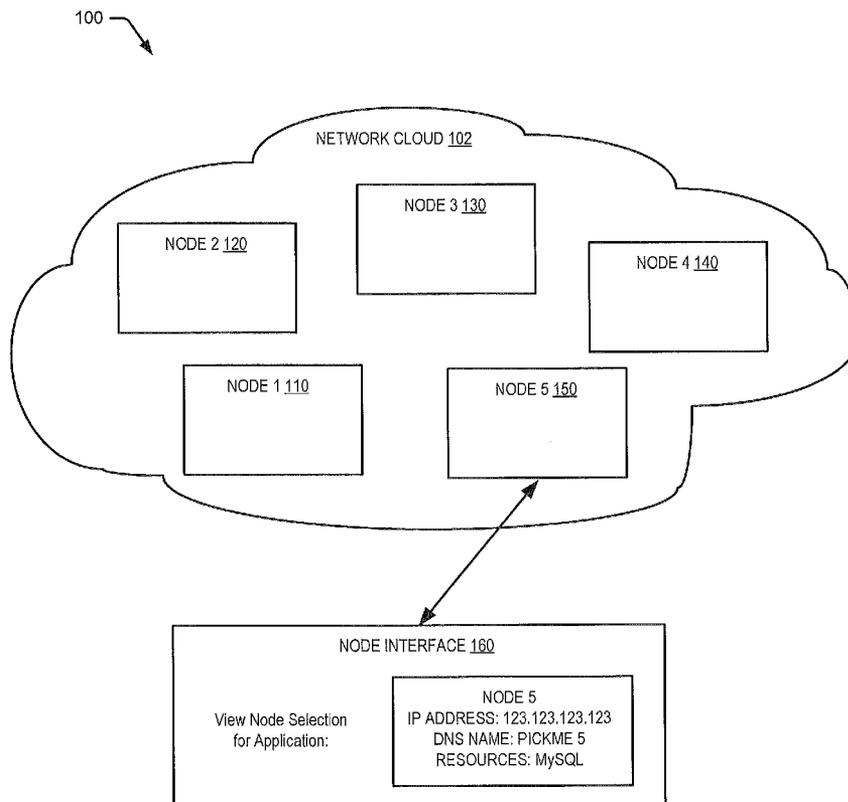
Methods, systems and computer program products for application centered node selection are provided. A plurality of nodes of a network having node attributes for executing applications are identified. Each node of the plurality of nodes is linked to an application based on a node attribute identified for the respective node. Each node linked to a linked application has a corresponding node attribute for executing the linked application. A node request is received for a target application. The node request may be independent of a node configuration, such as an IP address or DNS name. A node linked to the target application is selected and provided. The target application is executed by the selected node.

(51) **Int. Cl.**
G06F 15/173 (2006.01)
H04L 12/26 (2006.01)

(52) **U.S. Cl.**
CPC **H04L 43/04** (2013.01)

(58) **Field of Classification Search**
None
See application file for complete search history.

21 Claims, 9 Drawing Sheets



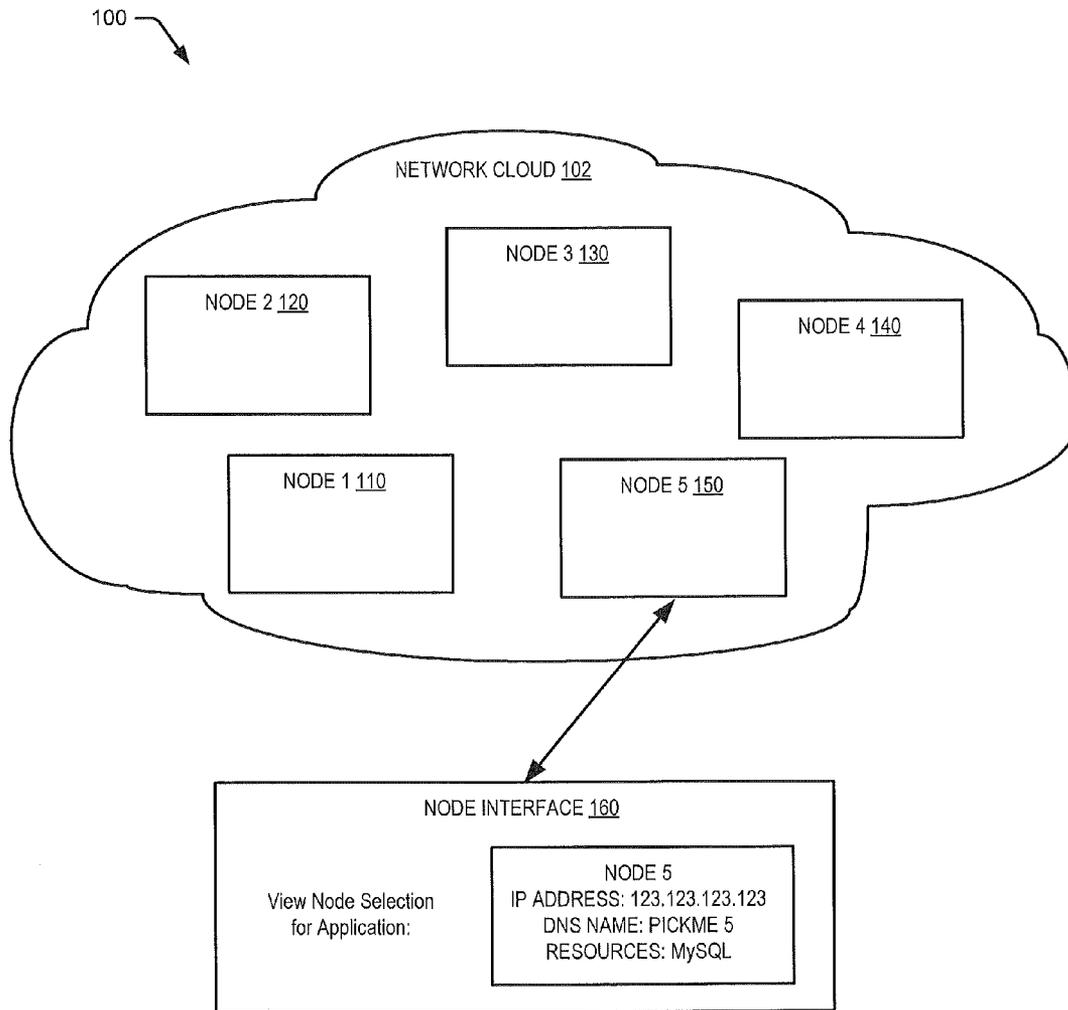


FIGURE 1

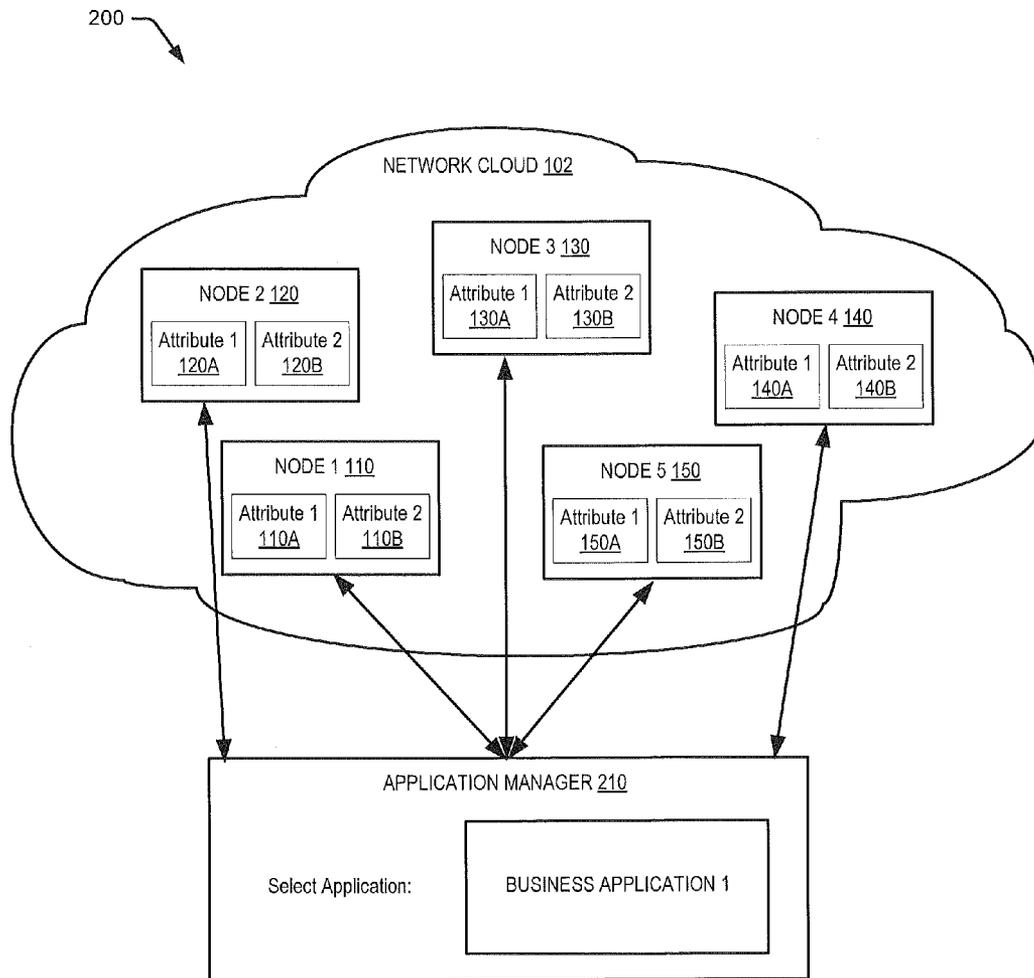


FIGURE 2

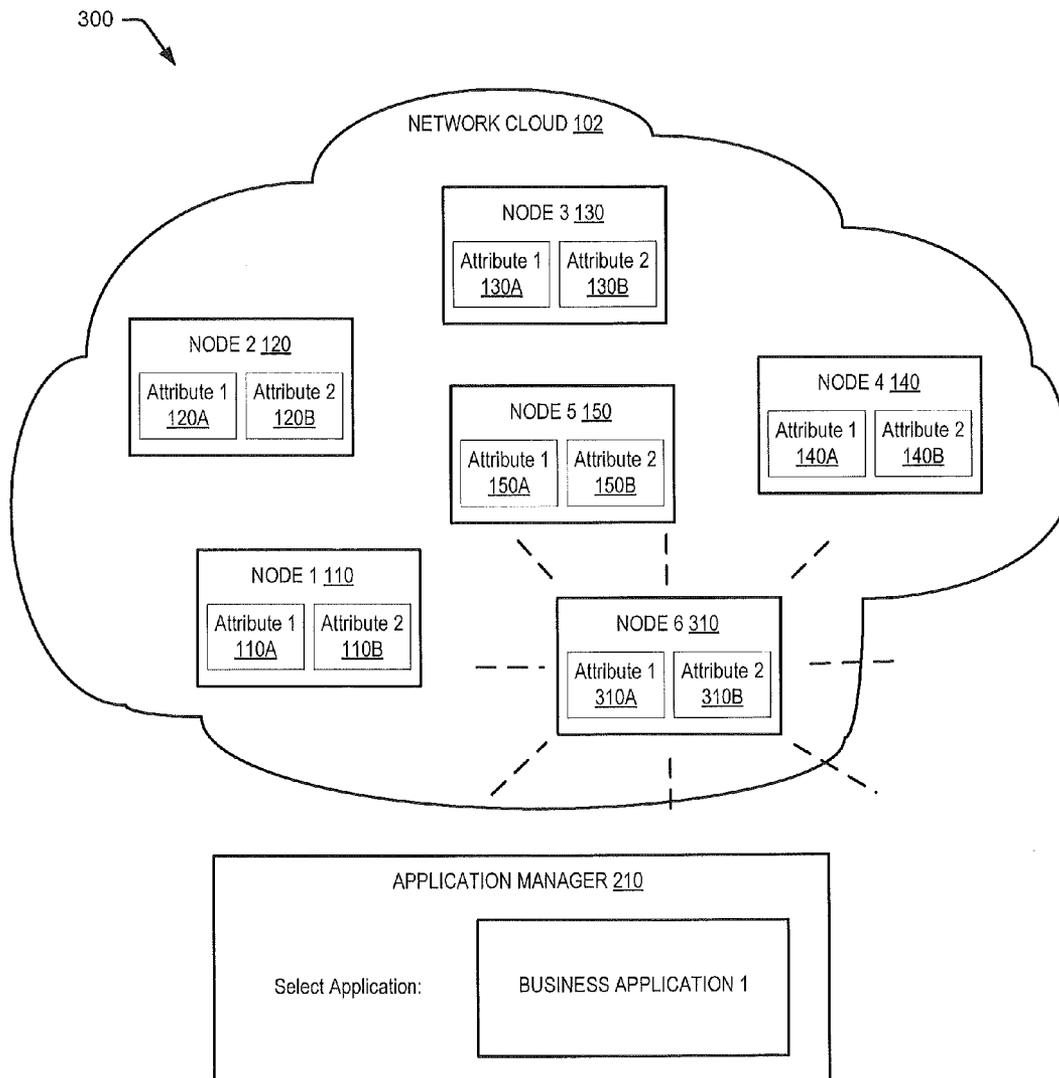


FIGURE 3

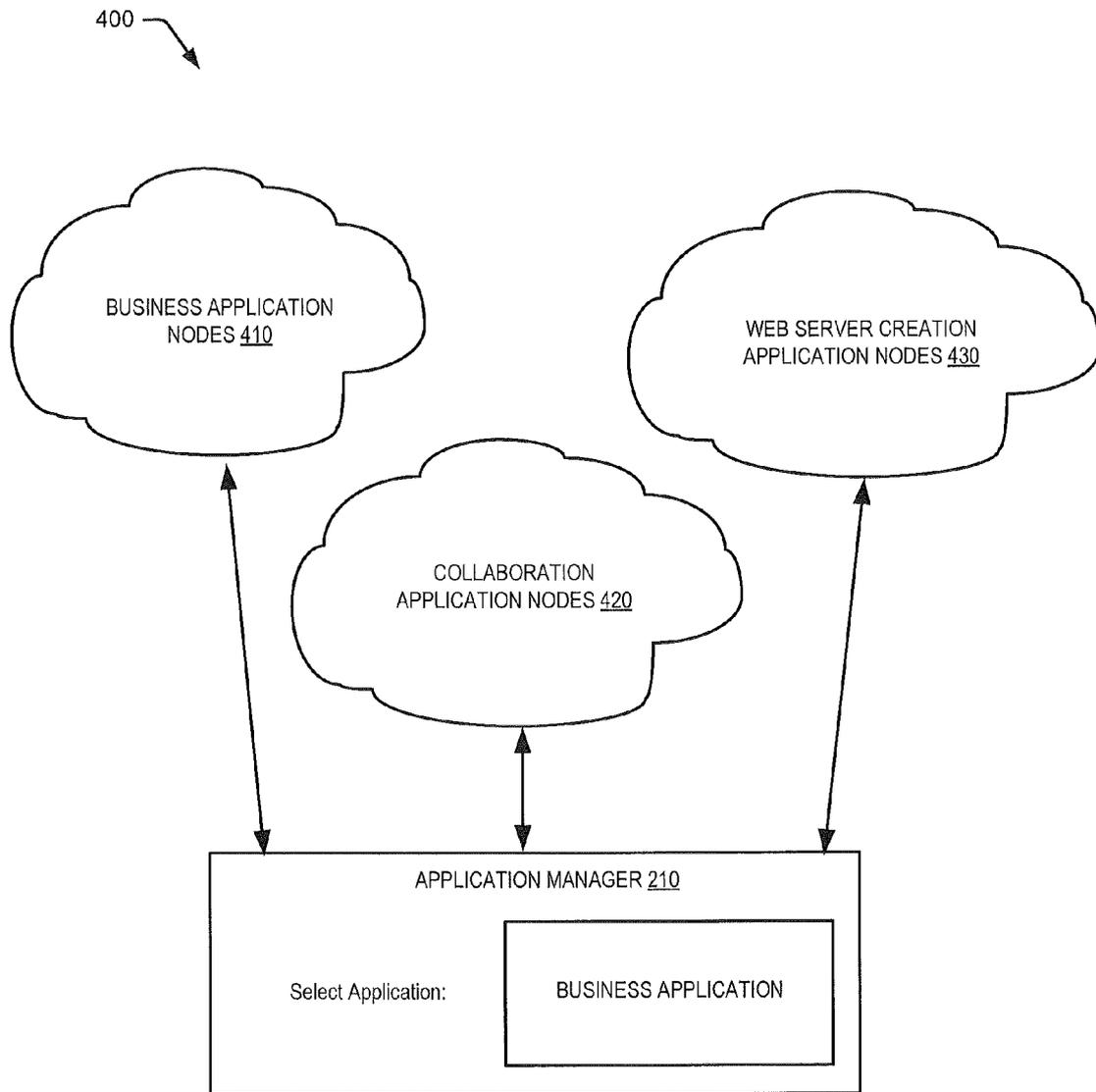


FIGURE 4

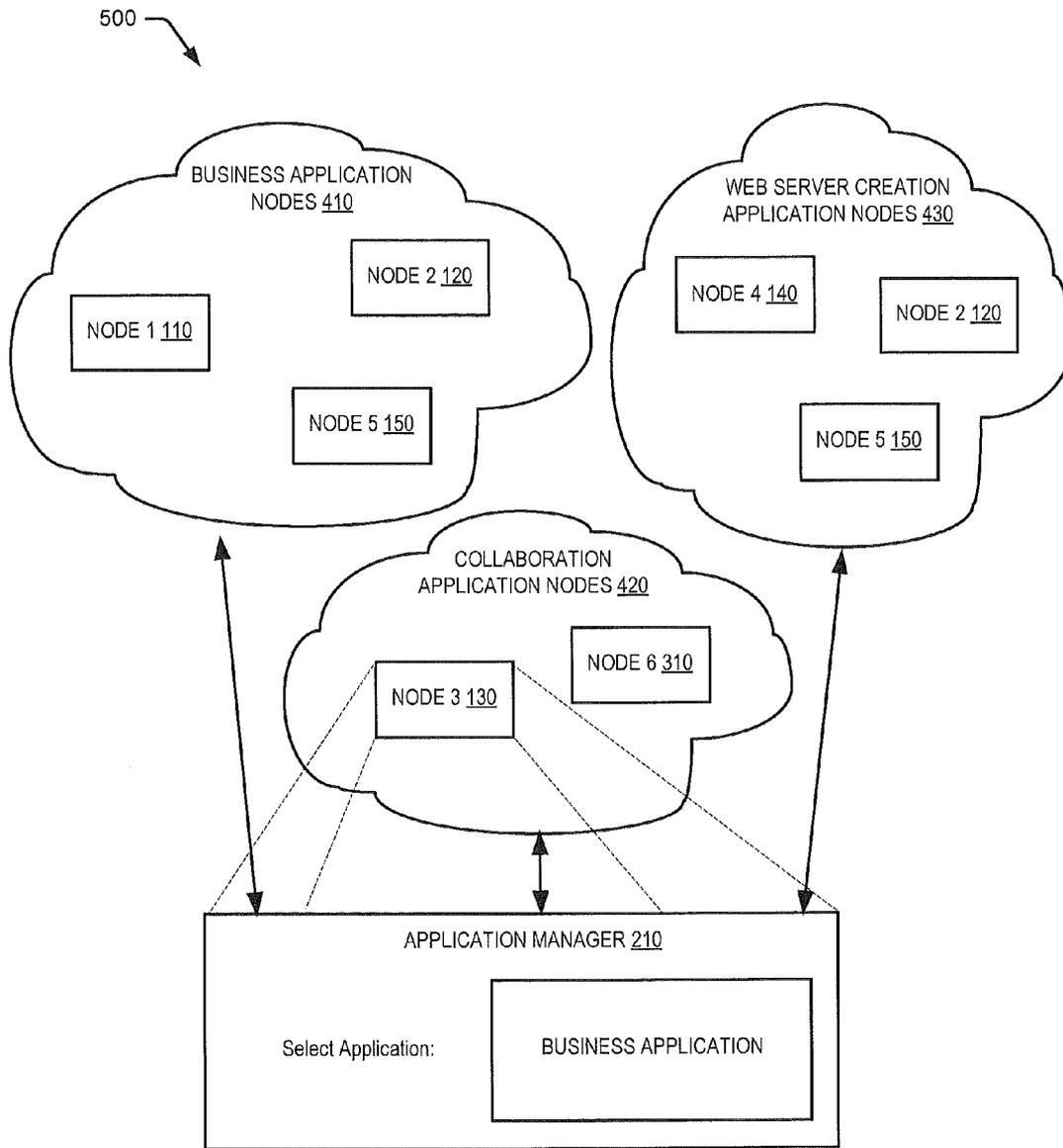


FIGURE 5

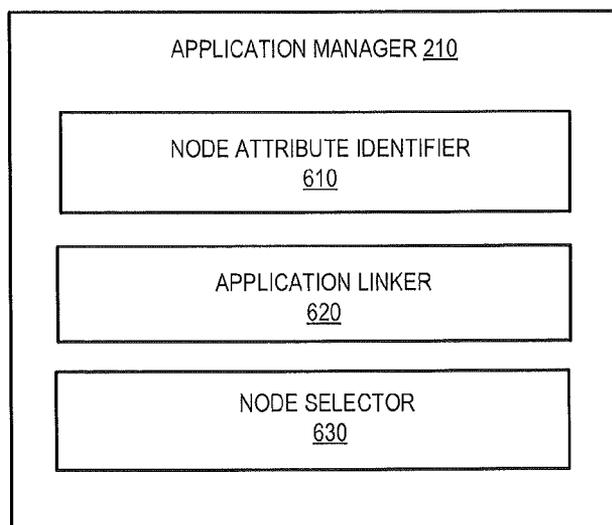


FIGURE 6

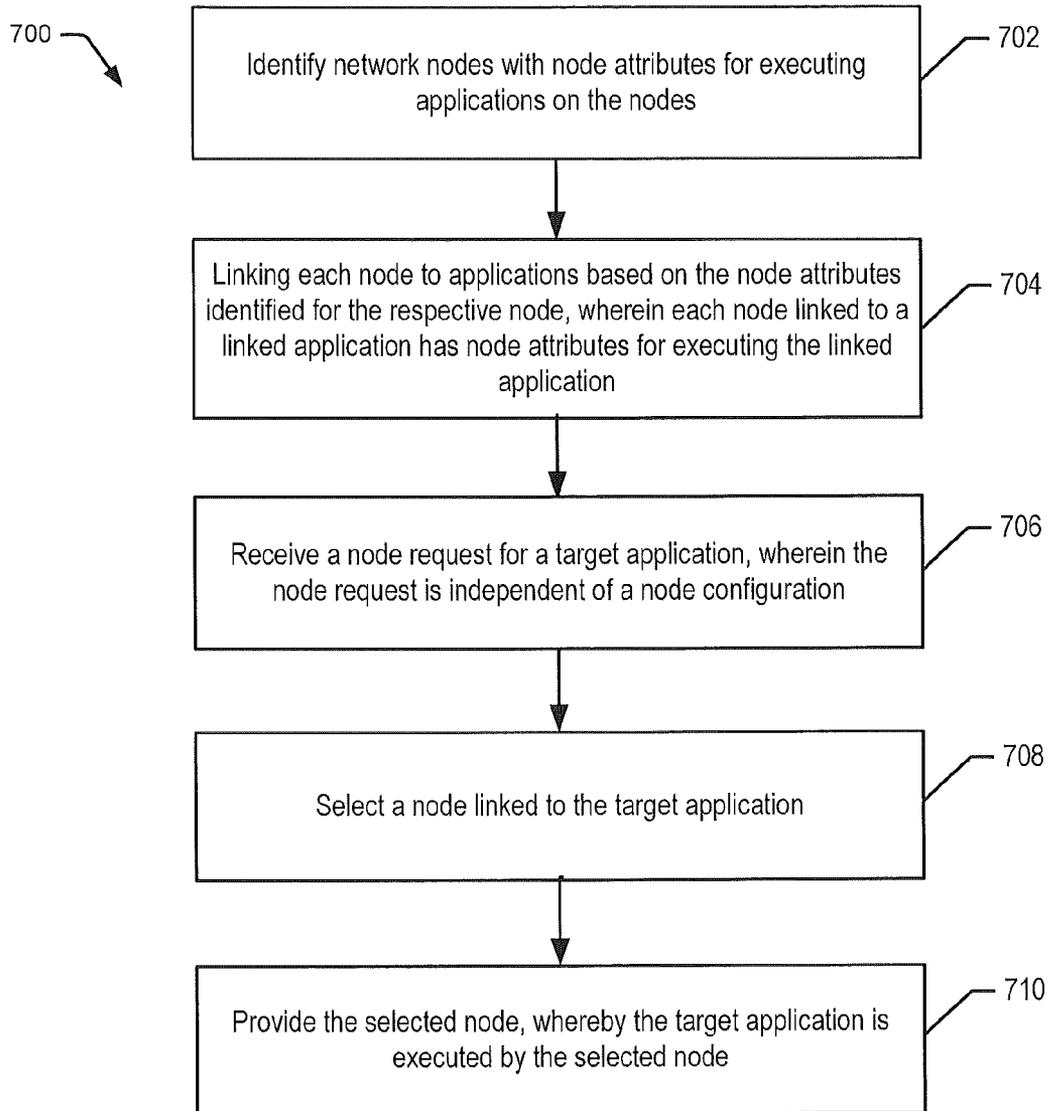


FIGURE 7

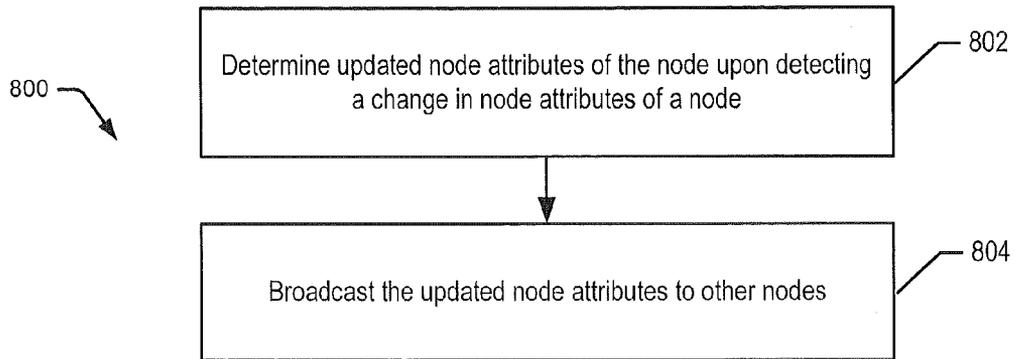


FIGURE 8

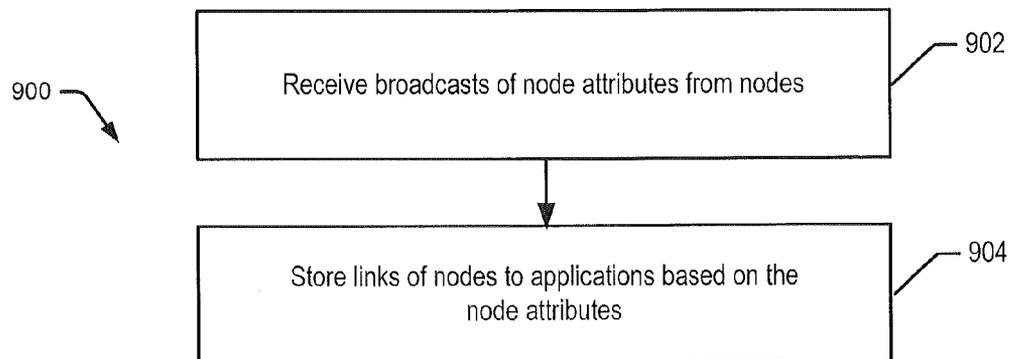


FIGURE 9

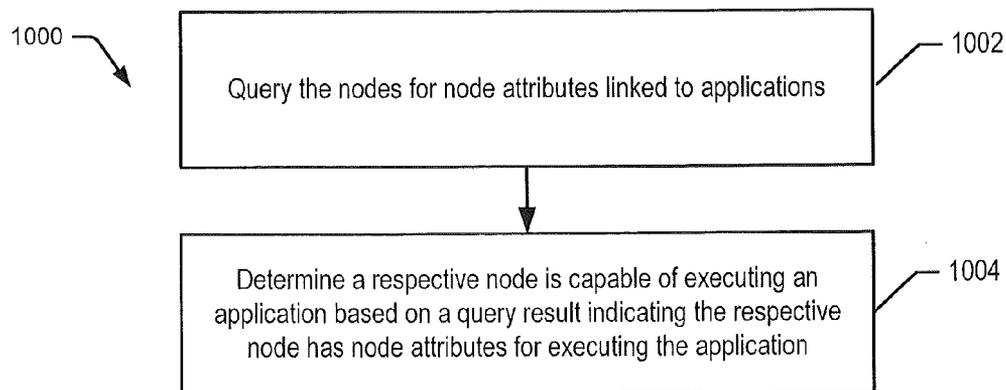


FIGURE 10

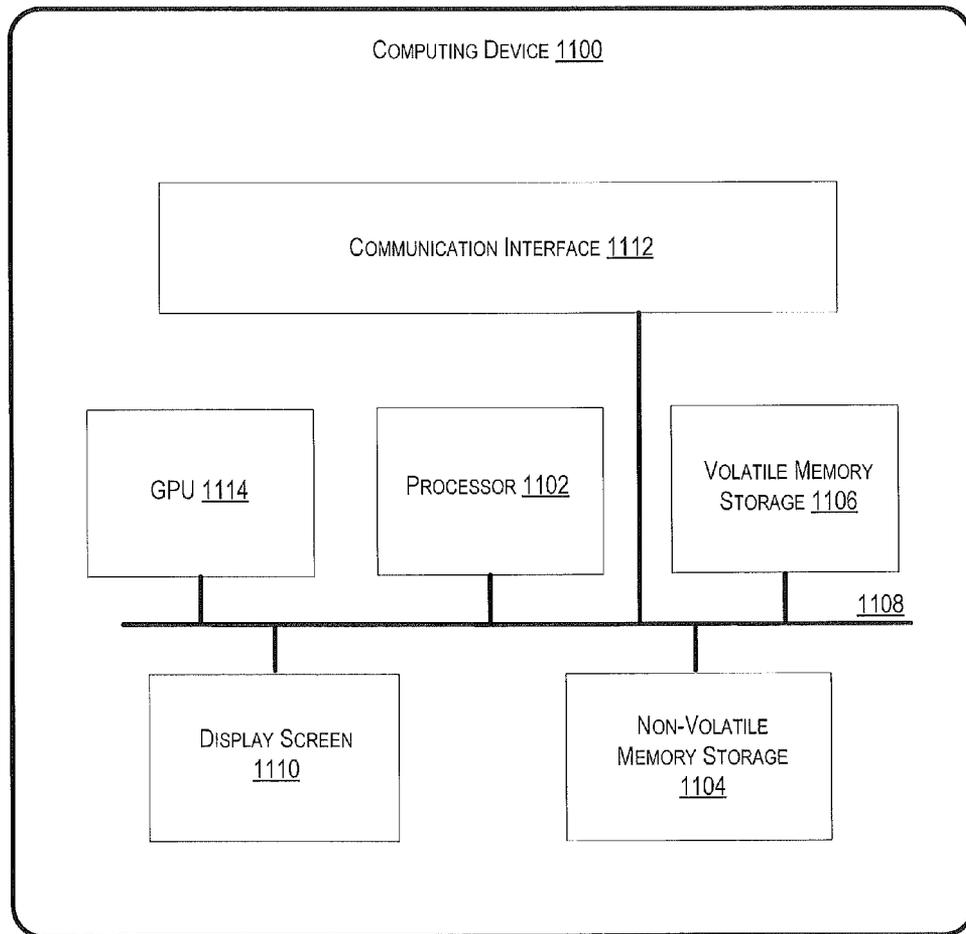


FIGURE 11

1

APPLICATION CENTERED NETWORK NODE SELECTION

BACKGROUND

The present disclosure relates generally to software applications in cloud computing networks.

Cloud computing provides for applications to execute on any number or arrangement of computers or network nodes. Some systems allow a user or administrator to request a network node for use in executing an application. Existing cloud environments require a network configuration of a node, such as an internet protocol (IP) address or domain name service (DNS) name, to be specified in order to proceed with node selection of for installation or execution of an application. For example, FIG. 1 shows a network cloud 102 with multiple servers labeled node 1 to node 5 (110-150). Node interface 160 is concerned with identifying the specific node to be utilized by an application, including network configuration information such as IP address and DNS name. Such node specificity adds a level of complexity for users.

BRIEF SUMMARY

Embodiments for application centered node selection are disclosed. According to an aspect, a plurality of nodes of a network having node attributes for executing applications are identified. Each node is linked to one or more applications based on one or more node attributes identified for the respective node. Each node linked to a linked application has node attributes for executing the linked application. A node request is received for a target application. The node request may be independent of a node configuration, such as an IP address or DNS name. A node linked to the target application is selected and provided. The target application is executed by the selected node.

Updated node attributes of the node may be determined upon detecting a change in a node attribute of a node. The updated node attributes are broadcast to other nodes.

Also, broadcasts of node attributes are received from nodes of the plurality of nodes. Links of nodes to applications based on the node attributes are stored. The nodes may be queried for node attributes linked to applications. A respective node may be determined to be capable of executing an application based on a query result indicating the respective node has one or more node attributes for executing the application.

In some cases, a node with an open TCP/IP port is linked to a database used for web applications. In other cases, a node with software for creating an instance of an application server is linked to the application server.

According to an aspect, a node with resources associated with a unique application identifier is linked to an application corresponding to the unique application identifier. A resource attribute may be received and a node with node attributes may be selected corresponding to the resource attribute. A customer name may be received and a node may be selected among nodes linked to applications associated with the customer name.

Some other embodiments are directed to related methods, systems and computer program products.

It is noted that aspects described with respect to one embodiment may be incorporated in different embodiments although not specifically described relative thereto. That is, all embodiments and/or features of any embodiments can be combined in any way and/or combination. Moreover, other systems, methods, and/or computer program products according to embodiments will be or become apparent to one

2

with skill in the art upon review of the following drawings and detailed description. It is intended that all such additional systems, methods, and/or computer program products be included within this description, be within the scope of the present invention, and be protected by the accompanying claims.

BRIEF DESCRIPTION OF THE DRAWINGS

Aspects of the present disclosure are illustrated by way of example and are not limited by the accompanying figures with like references indicating like elements.

FIG. 1 illustrates a system for selecting a node for an application;

FIG. 2 illustrates a system for application centered node selection, according to an embodiment;

FIG. 3 illustrates a system for application centered node selection;

FIG. 4 illustrates a conceptual view of application centered node selection;

FIG. 5 further illustrates a conceptual view of application centered node selection;

FIG. 6 further illustrates a system for application centered node selection, according to an embodiment;

FIG. 7 illustrates a process for application centered node selection, according to an embodiment;

FIG. 8 illustrates another process for application centered node selection;

FIG. 9 illustrates a further process for application centered node selection;

FIG. 10 illustrates another process for application centered node selection; and

FIG. 11 is a block diagram of a computing device in which embodiments can be implemented.

DETAILED DESCRIPTION

Embodiments of the present disclosure will be described more fully hereinafter with reference to the accompanying drawings. Other embodiments may take many different forms and should not be construed as limited to the embodiments set forth herein. Like numbers refer to like elements throughout.

The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting to other embodiments. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises," "comprising," "includes" and/or "including" when used herein, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

Unless otherwise defined, all terms (including technical and scientific terms) used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this invention belongs. It will be further understood that terms used herein should be interpreted as having a meaning that is consistent with their meaning in the context of this specification and the relevant art and will not be interpreted in an idealized or overly formal sense unless expressly so defined herein.

As will be appreciated by one skilled in the art, aspects of the present disclosure may be illustrated and described herein in any of a number of patentable classes or context including any new and useful process, machine, manufacture, or com-

position of matter, or any new and useful improvement thereof. Accordingly, aspects of the present disclosure may be implemented as entirely hardware, entirely software (including firmware, resident software, micro-code, etc.) or combined software and hardware implementation that may all generally be referred to herein as a "circuit," "module," "component," or "system." Furthermore, aspects of the present disclosure may take the form of a computer program product embodied in one or more computer readable media having computer readable program code embodied thereon.

Existing cloud environments require a network configuration, such as an IP address or DNS name of a node, to be specified in order to proceed with installation or execution of an application on the specified node. This is an extra level of complexity and sometimes another hurdle for an application user or system administrator. A user may have an application in mind and would just like a straightforward way to execute the application within a large and dynamic network of resources in the cloud.

Rather than focusing on a node and its network configuration, an application can learn the capabilities needed for the application and the nodes capable of handling the application. A node having those capabilities can be selected and provided. This way the user or administrator can be focused on the application and not on the specific nodes and the various capabilities of the specific nodes.

FIG. 2 illustrates an exemplary system 200 of a computing environment involving network cloud 102. Network cloud 102 may include any combination of multiple computing devices, including, but not limited to servers, mainframe, personal computer (PC), handheld computers, mobile computers, tablet, laptops, set top boxes, game consoles or any other appropriate computer hardware. Such computing devices may include computing resources such as a central processing unit (CPU); networking controllers; communication controllers; a display unit; a program and data storage device; memory controllers; input devices (such as a keyboard, a mouse, etc.) and output devices such as printers. The CPU may be any conventional processor, such as the AMD Athlon™ 64, Intel® Core™ Duo, or ARM® processor sets. Network cloud 102 may also include nodes that are virtualized instances of computers.

Network cloud 102 shows nodes 1-5 (110-150). Although some embodiments of nodes 110-150 may be configured to operate as a computer server, nodes 110-150 are not limited thereto and can be configured to provide other functionality, such as data processing, communications routing, etc. Nodes 110-150 may be network cloud 102 or a portion of network cloud 102. Nodes 110-150 may be securely associated with one another through a virtual private network (VPN) or any other grouping or configuration that identifies the nodes as available resources to a user.

Nodes 110-150 are communicatively coupled in network cloud 102, which may be part of a larger communicate network. Network cloud 102 facilitates wireless or wired communication, and may communicate using, for example, IP packets, Frame Relay frames, Asynchronous Transfer Mode (ATM) cells, voice, video, data, and other suitable information between network addresses. Network cloud 102 may include one or more local area networks (LANs), radio access networks (RANs), metropolitan area networks (MANS), wide area networks (WANs), a portion of the global computer network known as the Internet, and/or any other communication system or systems at one or more locations.

Users or administrators may use application manager 210 to install, execute or otherwise handle applications. An application may be chosen for installation or execution on nodes of

cloud network 102. No network configuration is required in a request for resources to install or execute the application. Application manager 210 is aware of the capabilities of nodes 110-150 and will select one or more of nodes 110-150 for the application based on nodes that have indicated resources or attributes that are capable of handling the application. In various embodiments, Application manager 210 is a web application or website hosted on each of nodes 110-150. Nodes 110-150, or application manager 210 on each of nodes 110-150, may be aware of the capabilities of each of the other nodes.

Each node may have certain attributes. For example, node 110 may have node attributes 110A and 110B and nodes 120-150 may have node attributes 120A-150B. Attributes may include resources or information that corresponds with specific applications. Attributes may be capabilities for handling applications. Capabilities may include computing device configurations, network configurations or software necessary for handling one or more applications. Examples of such capabilities may include, but are not limited to open source or proprietary databases, port configurations, and software implementations such as application servers, web servers or other related software for creating server instances. Capabilities may also include hardware features and functionality, CPU architecture, operating system versions, and available resources such as disk space, memory and network connectivity. In some cases, attributes may include customer names and/or customer specific applications.

Node attributes 110A to 150B of nodes 110-150 may be identified by application manager 210. These attributes may be determined by querying nodes of cloud network 102 and analyzing query results, according to some embodiments. Queries may be periodic or event based. Queries may be directed in a general broadcast to all associated nodes or direct to a specific node or specific set of nodes. Queries may request all node attribute, node, network configuration and software configuration information or only a subset of information. Queries for a specific node attribute may be made.

In other embodiments, each node broadcasts its node attributes to application manager 210 or other nodes. In some cases, this may be performed using multicast DNS (mDNS). Node attributes may be broadcast when a node changes its node attributes. In other embodiments, a new node entering cloud network 102 broadcasts its node attributes upon arrival or inclusion in the nodes available for selection by application manager 210. For example, FIG. 3 shows system diagram 300 with new node 6 310 joining nodes 110-150 in network cloud 102. Node 6 310 has node attributes 310A and 310B. According to this embodiment, node 6 310 broadcasts node attributes 310A and 310B to nodes 110-150. Application manager 210 may also receive this broadcast, record node attributes 310A and 310B and link them to corresponding applications.

According to an embodiment, node attributes of nodes 110-150 and 310 may be identified as attributes that provide for the installation and execution of application. Each node may then be linked to applications based on node attributes identified for the respective node. Each node linked to an application will have node attributes for executing the linked application. One or more nodes may execute an instance of a certain application or type of application. The same application, or separate instances of the same application, can be executed by any nodes that are capable of executing the application. This allows for more real time flexibility of node resources and selection.

Application manager 210 may represent nodes 110-150 and 310 of cloud network 102 to a user or administrator in an application centric manner. Network configurations or iden-

5

tification of specific nodes and their network locations will not be necessary in a user resource request. For example, rather than viewing nodes as in FIG. 1, a user or administrator may be able to view network cloud 102 conceptually as having resources for different types of applications. For example, FIG. 4 shows a conceptual view 400 of network cloud 102 resources. Resources available to a user through application manager 210 include nodes for a business application 410, nodes for a collaboration application 420 and nodes for a web server creation application 430. In this case, only an application needs to be specified and a node with attributes corresponding to the application will be provided.

In the embodiment of conceptual view 400, application manager 210 may receive a request for a node or resources to install and/or execute the "business application," as shown in the window of application manager 210 in FIG. 4. This request may be independent of a network configuration of a specific node. Rather than identification of a node and its network location, only the application is necessary from the user to receive a node. A node is selected from business application nodes 410 and provided for use by application manager 210. The business application is then installed or executed using the selected node.

The user request is based upon application centric conceptual view 400. Because known node attributes of the nodes allow for linking of nodes to applications, node 1 110, node 2 120 and node 5 150 are linked to the business application. Upon a request for a node linked to the business application, node 1 110, node 2 120 or node 5 150 may be selected for the business application.

In various embodiments, many nodes may have similar node attributes and may link to more than one application. For example, in diagram 500 of FIG. 5, node 2 120 and node 5 150 are linked to both the business application and the web server creation application, as shown by the business application nodes 410 and web server creation application nodes 430. By contrast, node 1 110 has node attributes for only the business application. Also, in this example, node 3 130 and node 6 310 have node attributes that link the nodes to only the collaboration application as shown by the collaboration application nodes 420. Note that in various embodiments, the nodes are multi-tenant and separate instances of applications may be created in a single node or application cloud.

According to an embodiment, FIG. 6 shows a more detailed view of application manager 210. Application manager 210 may include node attribute identifier 610, application linker 620 and node selector 630. Application manager 210 and its components are implemented with a computing device.

Node attribute identifier 610 is configured to identify nodes of a network with node attributes for executing applications on the plurality of nodes. Application linker 620 is configured to link each node of the plurality of nodes to applications based on the node attributes identified for the respective node. Each node linked to a linked application has node attributes for executing the linked application. Node selector 630 may be configured to receive a node request for a target application. The node request is independent of a node configuration. Node selector 630 may then select a node linked to the target application.

The selected node may be provided. When a selected node is provided, the selected node is made available for execution of the target application. This may include providing information identifying the node to a user, computer system, application manager and/or node manager. This may also include automatically proceeding with installation and/or execution on the selected node. Providing may include creating and/or

6

associating a node configuration with an application. In other cases, this may include displaying a node configuration for the selected node to a user. Providing the selected node may include other operations that allow any combination of application life cycle stages to proceed on the selected node.

The target application is executed by the selected node. Each of these components may perform their respective functions, as described earlier and according to the processes described below in FIGS. 7-10.

FIG. 7 shows a process for application centered node selection, according to an embodiment. In block 702, network nodes with node attributes for executing applications on the nodes are identified. Node attributes may include capabilities and resources necessary for executing specific applications. Examples of node attributes may include databases, software, code bases, server configurations, physical or virtual processing power, quality of service (QoS) metrics, port configurations, or any other capability or resource for applications.

Some identified node attributes may signify other node attributes for linked applications. For example, an open TCP/IP port, such as port 3306, may signify a certain type of database, such as a MySQL® database, is available on the node for web applications. In another example, if a server has Java® and the tomcat-create-instance executable, it can be considered to have node attributes corresponding to a capability for Tomcat® software.

Each node may be linked to applications based on node attributes identified for a respective node, wherein each node linked to a linked application has node attributes for executing the linked application (block 704). Node attributes can be predefined by for use by certain applications. Links may be created and modified dynamically. Node attributes may change for some servers as certain software or applications are installed or removed on server nodes. In other cases, server nodes may go offline for maintenance, overcapacity or other reasons. New server nodes may become available. According to a further embodiment, each of these node attribute updates may be determined upon detection of a change (FIG. 8 block 802). Such node attribute updates may be broadcast to other nodes (FIG. 8 block 804).

According to another embodiment, nodes within cloud network 102 and/or application manager 210 may receive these broadcasts (FIG. 9 block 902). Links of nodes to applications can be formed and stored based on the node attributes (FIG. 9 block 904).

According to an embodiment, nodes may be queried for node attributes linked to applications (FIG. 10 block 1002). Nodes may be queried by application manager 210 on a node or other nodes. A respective node may be determined to be capable of executing an application based on a query result indicating the respective node has node attributes for executing the application (FIG. 10 block 1004).

A node with software for creating an instance of an application server may be linked to the application server, according to an embodiment. In another embodiment, a node with resources associated with a unique application identifier or an application key (appKey), may be linked to an application corresponding to the application identifier. Application identifiers may be used to uniquely identify applications and instances of applications, Application identifiers isolate or shield applications from other applications. Application identifiers also allow applications to find resources by type or name via DNS.

Returning to block 706 of FIG. 7, a node request is received for a target application. The node request is independent of a node configuration. No network location, IP address, server name or DNS name is asked of the user, entered by the user or

selected by the user. The request asks that one or more nodes be provided so that a user may install or execute the target application on the one or more nodes. In some embodiments, the request may provide one or more virtual machines of the one or more nodes. In many cases, providing the name or reference of the target application is all that is necessary to proceed with node selection for the target application.

A node linked to the target application is selected (FIG. 7 block 708). A node may be selected from a number of qualifying nodes having sufficient node attributes. The node that is selected may be selected based on a number of factors, including but not limited to, CPU availability, memory availability, physical location, virtual machine configuration, IP addresses, software version, or other optimization considerations. Node selection may also be sequential or random. Node selection may not be limited a single node and may include multiple nodes.

The selected node is provided (block 710). The target application is executed by the selected node. In some cases, a target application may be installed on a node or installed to some degree on the node. Node selection then results in execution of the target application on the node. In other cases, node selection may result in installation, back up, restoring, starting, stopping, removal of an application on the selected node or any other application lifecycle events. The selected node, having been determined to be linked to a target application, can perform as necessary to ensure that the target application is utilized with the selected node.

According to an embodiment, node attributes may include a customer name. If the customer name is specified, nodes linked to the customer name may be provided. Such customer to node links may be formed based on one or more applications associated with or used by a customer being linked to a node. In some cases, node to customer links may require two or more node to application links relevant to a customer.

In another embodiment, a named resource or node attribute may be specified. In this case, a node with the named resource is provided. In yet another embodiment, if a QoS metric is requested, a node with the specified QoS metric is provided. Further embodiments may include selection of nodes linked to application based on application centered or customer centered requests.

The embodiments described above provide for real time information for node selection within a dynamic network or system, where nodes, node attributes and network configurations change but the request may remain simple and application focused. In such a system, resources may be reallocated and moved about as needed while providing a more universal model. Specific knowledge of the machines in network cloud 102 is removed from the request and management processes. Instead, the focus may be on applications, resource types and QoS metrics. In such a system, resources may be reallocated and moved about as needed while providing a more universal model.

According to an embodiment, the functionality of application manager 210, as described in the embodiments above, may be implemented in any combination of nodes 110-150 or node 310 of cloud network 102, as shown in FIG. 5. The functionality may also be implemented in a separate node. In that case, requests may be made through a node and any of the nodes may be selected for the target application without a separate computing device providing the functionality of application manager 210.

According to some embodiments, Redis™ data stores may be used for configuration data. Knife™ command tools may be used to create nodes and Chef™ software may be used to install the functionality of application manager 210 on a

computing device or node. In a further embodiment, nodes 110-150 and 310 may have agents that work in coordination with application manager 210 or implement the functionality of application manager 210. Node.js® software may be used to implement these node agents. In yet another embodiment, RESTful APIs are used for internode connections.

In an implementation, the functionality of application manager 210 may be provided through a browser on a node or computing device. The browser may be any commonly used browser, including any multithreading browser. Application manager 210 may be software in a browser or software displayed by the browser. Application manager 210 may be software hosted by a server and served to client devices over networks, including cloud network 102.

As will be appreciated by one of skill in the art, aspects of the disclosure may be embodied as a method, data processing system, and/or computer program product. Furthermore, embodiments may take the form of a computer program product on a tangible computer readable storage medium having computer program code embodied in the medium that can be executed by a computing device,

FIG. 11 is an example computer system 1100 in which embodiments of the present disclosure, or portions thereof, may be implemented as computer-readable code. For example, the components of nodes 110-150, node 310, application 210 or any other components of systems 200-600 or methods 700-1000 may be implemented in one or more computer devices 1100 using hardware, software implemented with hardware, firmware, tangible computer-readable storage media having instructions stored thereon, or a combination thereof and may be implemented in one or more computer systems or other processing systems. Components and methods in FIGS. 2-10 may be embodied in any combination of hardware and software.

Computing device 1100 may include one or more processors 1102, one or more non-volatile storage mediums 1104, one or more memory devices 1106, a communication infrastructure 1108, a display screen 1110 and a communication interface 1112. Computing device 1100 may also have networking or communication controllers, input devices (keyboard, a mouse, touch screen, etc.) and output devices (printer or display).

Processor(s) 1102 are configured to execute computer program code from memory devices 1104 or 1106 to perform at least some of the operations and methods described herein, and may be any conventional or special purpose processor, including, but not limited to, digital signal processor (DSP), field programmable gate array (FPGA), application specific integrated circuit (ASIC), and multi-core processors.

GPU 1114 is a specialized processor that executes instructions and programs, selected for complex graphics and mathematical operations, in parallel.

Non-volatile storage 1104 may include one or more of a hard disk drive, flash memory, and like devices that may store computer program instructions and data on computer-readable media. One or more of non-volatile storage device 1104 may be a removable storage device.

Memory devices 1106 may include one or more volatile memory devices such as but not limited to, random access memory. Communication infrastructure 1108 may include one or more device interconnection buses such as Ethernet, Peripheral Component Interconnect (PCI), and the like.

Typically, computer instructions are executed using one or more processors 1102 and can be stored in non-volatile storage medium 1104 or memory devices 1106.

Display screen 1110 allows results of the computer operations to be displayed to a user or an application developer.

Communication interface **1112** allows software and data to be transferred between computer system **1100** and external devices. Communication interface **1112** may include a modem, a network interface (such as an Ethernet card), a communications port, a PCMCIA slot and card, or the like. Software and data transferred via communication interface **1112** may be in the form of signals, which may be electronic, electromagnetic, optical, or other signals capable of being received by communication interface **1112**. These signals may be provided to communication interface **1112** via a communications path. The communications path carries signals and may be implemented using wire or cable, fiber optics, a phone line, a cellular phone link, an RF link or other communications channels. According to an embodiment, a host operating system functionally interconnects any computing device or hardware platform with users and is responsible for the management and coordination of activities and the sharing of the computer resources.

Any combination of one or more computer readable media may be utilized. The computer readable media may be a computer readable signal medium or a computer readable storage medium. A computer readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer readable storage medium would include the following: a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

A computer readable signal medium may include a propagated data signal with computer readable program code embodied therein, for example, in baseband or as part of a carrier wave. Such a propagated signal may take any of a variety of forms, including, but not limited to, electro-magnetic, optical, or any suitable combination thereof. A computer readable signal medium may be any computer readable medium that is not a computer readable storage medium and that can communicate, propagate, or transport a program for use by or in connection with an instruction execution system, apparatus, or device. Program code embodied on a computer readable signal medium may be transmitted using any appropriate medium, including but not limited to wireless, wireline, optical fiber cable, RF, etc., or any suitable combination of the foregoing.

Computer program code for carrying out operations for aspects of the present disclosure may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, JavaScript, Scala, Smalltalk, Eiffel, JADE, Emerald, C++, C#, VB.NET, Python or the like, conventional procedural programming languages, such as the "C" programming language, Visual Basic, Fortran 2003, Perl, COBOL 2002, PHP, ABAP, dynamic programming languages such as Python, Ruby and Groovy, or other programming languages. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the

user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider) or in a cloud computer environment or offered as a service such as a Software as a Service (SaaS).

Aspects of the present disclosure are described herein with reference to flowchart illustrations and/or block diagrams of methods, systems and computer program products according to embodiments. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create a mechanism for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

These computer program instructions may also be stored in a computer readable medium that when executed can direct a computer, other programmable data processing apparatus, or other devices to function in a particular manner, such that the instructions when stored in the computer readable medium produce an article of manufacture including instructions which when executed, cause a computer to implement the function/act specified in the flowchart and/or block diagram block or blocks. The computer program instructions may also be loaded onto a computer, other programmable instruction execution apparatus, or other devices to cause a series of operational steps to be performed on the computer, other programmable apparatuses or other devices to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide processes for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

It is to be understood that the functions/acts noted in the blocks may occur out of the order noted in the operational illustrations. For example, two blocks shown in succession may in fact be executed substantially concurrently or the blocks may sometimes be executed in the reverse order, depending upon the functionality/acts involved. Although some of the diagrams include arrows on communication paths to show a primary direction of communication, it is to be understood that communication may occur in the opposite direction to the depicted arrows.

Many different embodiments have been disclosed herein, in connection with the above description and the drawings. It will be understood that it would be unduly repetitious and obfuscating to literally describe and illustrate every combination and subcombination of these embodiments. Accordingly, all embodiments can be combined in any way and/or combination, and the present specification, including the drawings, shall support claims to any such combination or subcombination.

The foregoing description of the specific embodiments will so fully reveal the general nature of the invention that others can, by applying knowledge within the skill of the art, readily modify and/or adapt for various applications such specific embodiments, without undue experimentation, without departing from the general concept of the present invention. Therefore, such adaptations and modifications are intended to

11

be within the meaning and range of equivalents of the disclosed embodiments, based on the teaching and guidance presented herein.

The breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments or any actual software code with the specialized control of hardware to implement such embodiments, but should be defined only in accordance with the following claims and their equivalents.

What is claimed is:

1. A computer-implemented method, comprising:

acquiring, at an application manager, node attributes from a plurality of computing nodes of a network, wherein the node attributes correspond to resources of ones of the plurality of computing nodes;

identifying, by the application manager responsive to acquiring the node attributes, a portion of the plurality of computing nodes of the network having a node attribute for executing a software application on the portion of the plurality of computing nodes,

wherein the software application is a computer program selected by a user of the network to be installed or executed on one or more computing nodes of the portion of the plurality of computing nodes, and

wherein the node attribute for executing the software application indicates that a respective computing node of the portion of the plurality of computing nodes has a computing device configuration, a network configuration or required software to handle execution of the software application on the respective computing node of the portion of the plurality of computing nodes;

linking, within a data repository by the application manager, each respective computing node of the portion of the plurality of computing nodes to the software application based on the node attribute for executing the software application identified for the respective computing node of the portion of the plurality of computing nodes, whereby each respective computing node of the portion of the plurality of computing nodes linked to the software application has the corresponding node attribute for executing the software application;

receiving at the application manager a node request for the software application to be executed by one of the plurality of computing nodes, wherein the node request is independent of a computing node IP address and a computing node DNS name for any computing node of the plurality of computing nodes;

selecting, by the application manager, a selected computing node linked to the software application from among the portion of the plurality of computing nodes responsive to receiving the node request for the software application; and

providing, by the application manager, the selected computing node for execution of the software application responsive to the node request.

2. The method of claim 1, further comprising:

detecting a change in the node attribute of an updated computing node of the plurality of computing nodes, and, responsively, determining an updated node attribute of the updated computing node of the plurality of computing nodes; and

broadcasting the updated node attribute to ones of the plurality of computing nodes.

3. The method of claim 1, further comprising:

receiving broadcasts of broadcast node attributes from the plurality of computing nodes of the network; and

12

storing links of ones of the plurality of computing nodes to the software application based on the broadcast node attributes in the broadcasts.

4. The method of claim 1, wherein identifying comprises: querying the plurality of computing nodes for node attributes linked to the software application; and determining that the respective computing node of the portion of the plurality of computing nodes is capable of executing the software application based on a query result indicating that the respective computing node of the portion of the plurality of computing nodes has the node attribute for executing the software application.

5. The method of claim 1,

wherein the software application comprises a database used for web applications and

wherein linking each respective computing node of the portion of the plurality of computing nodes to the software application based on the node attribute comprises creating a link within the data repository between the respective computing node of the portion of the plurality of computing nodes and the software application based on the node attribute of the respective computing node of the portion of the plurality of computing nodes comprising an open TCP/IP port.

6. The method of claim 1, wherein linking each respective computing node of the portion of the plurality of computing nodes to the software application based on the node attribute comprises a creating a link within the data repository between the respective computing node of the portion of the plurality of computing nodes with resources associated with a unique application identifier and the software application corresponding to the unique application identifier.

7. The method of claim 1, wherein receiving the node request for the software application comprises receiving a customer name, and

wherein selecting the selected computing node linked to the software application comprises selecting the selected computing node among the portion of the plurality of computing nodes linked to the software application associated with the customer name.

8. The method of claim 1, wherein the software application is not installed on the selected computing node, and

wherein providing the selected computing node for execution of the software application comprises automatically proceeding with installation of the software application on the selected computing node prior to the execution of the software application on the selected computing node and responsive to the selection of the selected computing node.

9. The method of claim 1,

wherein the node attribute for executing the software application is a first node attribute acquired from the respective computing node of the portion of the plurality of computing nodes, and

wherein linking each respective computing node of the portion of the plurality of computing nodes to the software application based on the first node attribute comprises:

determining that a second node attribute not acquired from the respective computing node of the portion of the plurality of computing nodes indicates the computing device configuration, the network configuration or the required software to handle execution of the software application;

13

determining that the first node attribute acquired from the respective computing node of the portion of the plurality of computing nodes signifies the second node attribute.

10. A computer system comprising:

a processor; and

a memory coupled to the processor, the memory comprising computer readable program code embodied therein that, when executed by the processor, provides the following modules:

a node attribute acquisition module to acquire node attributes from a plurality of computing nodes of a network, wherein the node attributes correspond to resources of ones of the plurality of computing nodes;

a node attribute identifier module to identify, responsive to acquiring the node attributes, a portion of the plurality of computing nodes of the network having a node attribute for executing a software application on the portion of the plurality of computing nodes,

wherein the software application is a computer program selected by a user of the network to be installed or executed on one or more computing nodes of the portion of the plurality of computing nodes, and

wherein the node attribute for executing the software application indicates that a respective computing node of the portion of the plurality of computing nodes has a computing device configuration, a network configuration or required software to handle execution of the software application on the respective computing node of the portion of the portion of the plurality of the computing nodes;

an application linker module to link within a data repository each respective computing node of the portion of the plurality of computing nodes to the software application based on the node attribute for executing the software application identified for the respective computing node of the portion of the plurality of the computing nodes, whereby each respective computing node of the portion of the plurality of computing nodes linked to the software application has the corresponding node attribute for executing the software application; and

a node selector module to:

receive a node request for the software application to be executed by one of the plurality of computing nodes, wherein the node request is independent of a computing node IP address and a computing node DNS name for any computing node of the plurality of computing nodes;

select a selected computing node linked to the software application from among the portion of the plurality of computing nodes responsive to receiving the node request for the software application; and

provide the selected computing node for execution of the software application responsive to the node request.

11. The system of claim 10, wherein the application linker module further:

receives broadcasts of broadcast node attributes from the plurality of computing nodes of the network; and stores links of ones of the plurality of computing nodes to the software application based on the broadcast node attributes in the broadcasts.

12. The system of claim 10, wherein the node identifier module further:

14

queries the plurality of computing nodes for node attributes linked to the software application; and

determines that the respective computing node of the portion of the plurality of computing nodes is capable of executing the software application based on a query result indicating that the respective computing node of the portion of the plurality of computing nodes has the node attribute for executing the software application.

13. The system of claim 10, the software application comprises a database used for web applications, and wherein the application linker module further:

creates a link within the data repository between between the respective computing node of the portion of the plurality of computing nodes and the software application based on the node attribute of the respective computing node of the portion of the plurality of computing nodes comprising an open TCP/IP port.

14. The system of claim 10, wherein the application linker module further:

creates a link within the data repository between the respective computing node of the portion of the plurality of computing nodes with resources associated with a unique application identifier and the software application corresponding to the unique application identifier.

15. The system of claim 10, wherein the node selector module further:

receives a customer name; and

selects the selected computing node among the portion of the plurality of computing nodes linked to the software application associated with the customer name.

16. The computer system of claim 10, wherein the software application is not installed on the selected computing node, and

wherein the node selector module provides the selected computing node for execution of the software application by automatically proceeding with installation of the software application on the selected computing node prior to the execution of the software application on the selected computing node and responsive to the selection of the selected computing node.

17. The computer system of claim 10,

wherein the node attribute for executing the software application is a first node attribute acquired from the respective computing node of the portion of the plurality of computing nodes, and

wherein the application linker module links each respective computing node of the portion of the plurality of computing nodes to the software application based on the first node attribute by:

determining that a second node attribute not acquired from the respective computing node of the portion of the plurality of the computer nodes indicates the computing device configuration, the network configuration or the required software to handle execution of the software application;

determining that the first node attribute acquired from the respective computing node of the portion of the plurality of the computer nodes signifies the second node attribute.

18. The computer system of claim 10, wherein the software application is already installed on the selected computing node of the plurality of computing nodes as a first installation, wherein the node selector module provides the selected computing node of the plurality of computing nodes for execution of the software application by automatically proceeding with a second installation of the software application on the selected computing node of the plu-

15

rality of computing nodes prior to the execution of the software application on the selected computing node of the plurality of computing nodes, and
 wherein the second installation is performed using a unique application identifier that identifies the second installation and isolates the second installation from the first installation.

19. A computer program product for operating a computer system, the computer program product comprising:
 a non-transitory computer readable storage medium having computer readable program code embodied in the medium, the computer readable program code comprising:
 computer readable program code to:
 acquire, at an application manager, node attributes from a plurality of computing nodes of a network, wherein the node attributes correspond to resources of ones of the plurality of computing nodes,
 wherein the application manager is executed on a management computing node that is distinct from the plurality of computing nodes of the network;
 identify, by the application manager responsive to acquiring the node attributes, a computing node of the plurality of computing nodes having a node attribute for executing a software application on the computing node,
 wherein the software application is a computer program selected by a user of the network to be installed or executed on one or more computing nodes of the plurality of computing nodes, and
 wherein the node attribute for executing the software application indicates that the computing node of the plurality of computing nodes has a computing device configuration, a network configuration or required software to handle execution of the software application on the computing node of the plurality of computing nodes;
 link, within a data repository by the application manager, the computing node of the plurality of computing nodes to the software application responsive to the computing node of the plurality of computing nodes having the node attribute for executing the software application on the computing node of the plurality of computing nodes;
 receive at the application manager a node request for the software application to be executed by one of the plurality of computing nodes,
 wherein the node request is independent of a computing node IP address and a computing node DNS name for any of the plurality of computing nodes;
 query, by the application manager, the data repository to identify linked computing nodes of the plurality of computing nodes which are linked to the software application;

16

select, by the application manager, the computing node of the plurality of computing nodes linked to the software application responsive to the query of the data repository,
 wherein the software application is not installed on the computing node of the plurality of computing nodes;
 initiate automatic installation, by the application manager, of the software application on the computing node of the plurality of computing nodes prior to the execution of the software application on the computing node of the plurality of computing nodes, and responsive to the selection of the computing node of the plurality of computing nodes;
 provide, by the application manager, the computing node of the plurality of computing nodes for execution of the software application responsive to the node request;
 receive, at the application manager, updated node attributes from an updated computing node of the plurality of computing nodes; and
 update, by the application manager, a link within the data repository between the updated computing node of the plurality of computing nodes and the software application responsive to the updated node attributes received from the updated computing node of the plurality of computing nodes.

20. The computer program product of claim **19**, wherein the software application comprises a database used for web applications, and further comprising:
 computer readable program code to create a link within the data repository between the respective computing node of the portion of the plurality of computing nodes and the software application based on the node attribute of the respective computing node of the portion of the plurality of computing nodes comprising an open TCP/IP port.

21. The computer program product of claim **19**, wherein the node attribute for executing the software application is a first node attribute acquired from the respective computing node of the portion of the plurality of computing nodes, and
 wherein the computer readable program code to link each respective computing node of the portion of the plurality of computing nodes to the software application based on the first node attribute comprises computer readable code to:
 determine that a second node attribute not acquired from the respective computing node of the portion of the plurality of the computer nodes indicates the computing device configuration, the network configuration or the required software to handle execution of the software application;
 determine that the first node attribute acquired from the respective computing node of the portion of the plurality of the computer nodes signifies the second node attribute.

* * * * *