

(51) International Patent Classification:
G06F 9/445 (2006.01)

(21) International Application Number:

PCT/US2013/057075

(22) International Filing Date:

28 August 2013 (28.08.2013)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

13/597,104	28 August 2012 (28.08.2012)	US
13/597,125	28 August 2012 (28.08.2012)	US
13/597,150	28 August 2012 (28.08.2012)	US

(71) Applicant: VCE COMPANY LLC [US/US]; 1500 North Greenville Avenue, Suite 1100, Richardson, Texas 75081 (US).

(72) Inventors: **KSHIRSAGAR, Mohit**; 980 Kiely Blvd., #215, Santa Clara, California 95051 (US). **OLSZEWSKI, David**; 759 Marilyn Drive, Campbell, California 95008 (US). **MAHAPATRA, Akshaya**; 6071 Loma Prieta Drive, San Jose, California 95123 (US). **CHOUDRY, Faisal**; 80 Wroughton Road, London Greater London SW11 6BG (GB). **KUMAR, Archana**; 111 Saratoga Avenue, Apt. #2212, Santa Clara, California 95051 (US). **TALEKAR, Rahul**; 190 Ryland Street, #3213, San Jose, California 95110 (US). **KANNAN, Hariharan**; 660 Perkins Court, San Jose, California 95127 (US). **GILMOUR, Sean**; 7776 Ridgeline Drive, Dublin, California 94568 (US).

(74) Agents: STEWART, Jon K. et al.; Patterson & Sheridan, LLP, 3040 Post Oak BLVD., Suite 1500, Houston, Texas 77056 (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, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, 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, 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).

Published:

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

(54) Title: INTEGRATED COMPUTING PLATFORM DEPLOYED IN AN EXISTING COMPUTING ENVIRONMENT

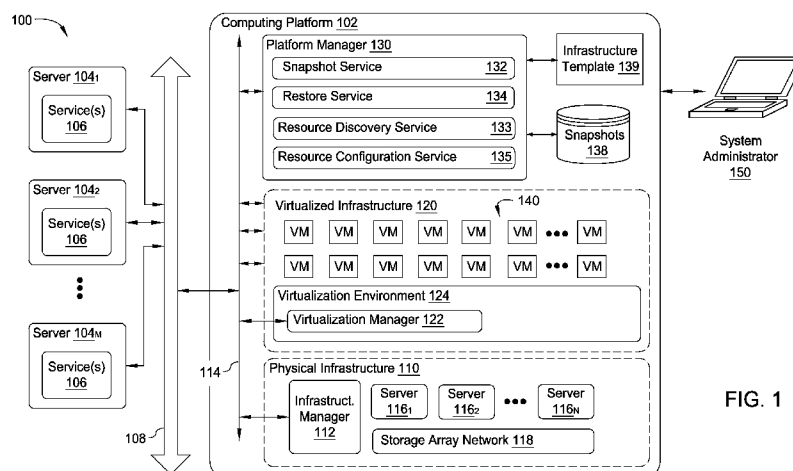


FIG. 1

(57) Abstract: A method and apparatus for integrating a converged infrastructure platform in a data center is provided. The described apparatus configures the physical and virtual resources that run on the converged infrastructure to communicate with the rest of the data center. The described apparatus further enables a system administrator to rollback configurations of the physical and virtual resources of the converged infrastructure to restore the converged infrastructure to a factory state.

INTEGRATED COMPUTING PLATFORM DEPLOYED IN AN EXISTING COMPUTING ENVIRONMENT

BACKGROUND OF THE INVENTION

Field of the Invention

5 [0001] Embodiments of the present invention generally relate to a method for installing a computing block platform within an existing data center.

Description of the Related Art

[0002] Generally, in a virtualized computer environment, virtual machines are configured to run on one or more host computers. Each virtual machine uses CPU and
10 memory resources of one of the host computers and runs its own operating system and application programs to give the appearance that it is operating as a stand-alone computer system. The amount of CPU and memory resources provisioned for each of the virtual machines on a host computer can be designated by an administrator of the virtualized computer environment. In some virtualized environments, load balancing is
15 enforced across multiple host computers by software that monitors resource usage on different host computers. Such software migrates virtual machines from one host computer to another, e.g., from a busy host computer to one that has excess capacity.

[0003] Additionally, enterprises engaged in developing, testing, and deploying software applications need to deal with many layers of the targeted platform. These
20 layers include application services, virtualization, and hardware infrastructure with compute, network, storage, and management at all levels. Information technology (IT) and engineering groups also acquire, deploy, and provide ongoing management, as well as ensure the layers work seamlessly together. This increases an enterprise's initial and ongoing cost, extends the development cycle, and reduces flexibility needed to
25 respond to changes in the market.

[0004] Further, the traditional enterprise information technology (IT) roles such as server administrator, UNIX administrator, and network, storage or exchange administrator, have been generally static. The roles operate in isolation, or "silos", which cause friction within IT organizations, as well as between developers and IT.
30 Further, it is difficult for developers to play the role of IT administrator when their developed applications are deployed as "cloud applications." This difficulty is due not

only to various access-controls, authentication and authorization complexities, but also to a model of organizational separation between developers and IT, where the default behavior is to deny access to the resources.

[0005] As such, there is a demand for a more efficient operational model for
5 administrating computing infrastructure.

SUMMARY OF THE INVENTION

[0006] Embodiments of the present disclosure provide a method for installing an integrated computing platform into an existing computing environment. The method includes configuring an integrated computing platform to communicate with at least one
10 service running in an existing computing environment outside of the integrated computing platform. The integrated computing platform includes an application having a plurality of functional components using a plurality of networking, storage, computing, and virtualized resources of the integrated computing platform. The method further includes restoring each of the plurality of networking, storage, computing, and virtualized
15 resources of the integrated computing platform to a state prior to the configuring the integrated computing platform to communicate with the service running in the existing computing environment outside of the integrated computing platform.

[0007] Embodiments of the present disclosure provide an integrated computing platform. The integrated computing platform includes a plurality of networking, storage,
20 computing, and virtualized resources having an application with a plurality of functional components executing thereon. The integrated computing platform includes a management server configured to configure the integrated computing platform to communicate with at least one service running in an existing computing environment outside of the integrated computing platform. The management server is further
25 configured to restore each of the plurality of networking, storage, computing, and virtualized resources of the integrated computing platform to a state prior to the configuring the integrated computing platform to communicate with the service running in the existing computing environment outside of the integrated computing platform.

[0008] Embodiments of the present disclosure provide a non-transitory computer-
30 readable storage medium comprising instructions that, when executed in a computing

device, install an integrated computing platform into an existing computing environment, by performing the steps of configuring an integrated computing platform to communicate with at least one service running in an existing computing environment outside of the integrated computing platform. The integrated computing platform may include an
5 application having a plurality of functional components using a plurality of networking, storage, computing, and virtualized resources of the integrated computing platform. The instructions, when executed, also perform the steps of restoring each of the plurality of networking, storage, computing, and virtualized resources of the integrated computing platform to a state prior to the configuring the integrated computing platform to
10 communicate with the service running in the existing computing environment outside of the integrated computing platform.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] So that the manner in which the above recited features of the present invention can be understood in detail, a more particular description of the invention,
15 briefly summarized above, may be had by reference to embodiments, some of which are illustrated in the appended drawings. It is to be noted, however, that the appended drawings illustrate only typical embodiments of this invention and are therefore not to be considered limiting of its scope, for the invention may admit to other equally effective embodiments.

20 [0010] Figure 1 illustrates an integrated computing platform configured for providing a virtualized environment according to one embodiment.

[0011] Figure 2 illustrates an application-specific deployment of the integrated computing platform of Figure 1, according to one embodiment.

25 [0012] Figure 3 is a flow diagram of method steps for deploying an integrated computing platform in an existing data center, according to one embodiment.

[0013] Figure 4 is a flow diagram of method steps for capturing state of an integrated computing platform, according to one embodiment.

30 [0014] Figure 5 is a flow diagram of method steps for restoring configuration changes to an integrated computing platform installed in an existing data center, according to one embodiment.

[0015] Figure 6 is a state diagram illustrating a workflow for resetting state of an integrated computing platform during installation of an application, according to one embodiment.

5 [0016] Figure 7 illustrates an example operation for deploying an application in the computing platform of Figure 1, according to one embodiment.

[0017] Figure 8 is a flow diagram of a method for deploying an application on computer resources in a virtualized environment, according to one embodiment.

10 [0018] Figure 9 is a flow diagram of a method for integrating a computing platform having an application executing thereon into an existing computing environment, according to one embodiment.

[0019] Figure 10 illustrates an example workflow for integrating the computing platform of Figure 2 within an existing data center, according to one embodiment.

15 [0020] To facilitate understanding, identical reference numerals have been used, where possible, to designate identical elements that are common to the figures. It is contemplated that elements disclosed in one embodiment may be beneficially utilized on other embodiments without specific recitation.

DETAILED DESCRIPTION

20 [0021] Embodiments of the present disclosure use a computing block platform, sometimes referred to as converged infrastructure, that consolidates IT infrastructure into a pool of virtualized physical as well as computing, storage, and networking capacity that is shared by multiple application and lines of businesses have been proposed to address the problem of silo architectures and IT sprawl. When an enterprise uses a converged infrastructure platform, the enterprise may be faced with the challenge of integrating the new converged infrastructure platform within any

25 existing computing infrastructure, such as a data center. Before a converged infrastructure platform can be used by an enterprise, the physical resources (e.g., blade servers, network switches, storages) and virtual machines that are “inside” the converged infrastructure platform have to be configured to communicate with physical resources and virtual machines that are “outside” the configured infrastructure platform,

and vice versa. Further, a converged infrastructure platform may be configured to run an application having multiple functional components (e.g., executing on multiple virtual machines). At least one of the functional components may rely on existing services running outside of the converged infrastructure, and must be configured to communicate
5 with any required services running in the existing data center.

[0022] As such, to install a converged infrastructure platform having an application pre-loaded therein, a number of changes may have to be made to the networking, storage, and computing resources of the converged infrastructure platform. For example, the network within the converged infrastructure has to understand the data
10 center network already in place, extend the network setup into the converged infrastructure platform, and enable communication between the converged infrastructure and the rest of the network in the data center. To complete installation, the converged infrastructure platform may create new VLANs and port groups, as well as change new virtual machines and edit settings of existing virtual machines.

[0023] However, if the installation fails (e.g., unable to connect with the network of the existing data center), it may be desirable to “rollback” changes done to the system such that the installation may start again from a clean state. Accordingly, embodiments of the present disclosure provide an administrative application that generates snapshots of the various resources (e.g., networking, storage, computing, virtual) of the converged
20 infrastructure platform before and after configuration for an application. The administrative application may initiate a rollback operation that restores configurations to the networking, storage, computing, and virtual resources. A rollback or reset of the application means not only reverting to known snapshots on the virtual infrastructure (e.g., virtual machines), but also roll back of configurations changes of physical
25 resources to free up those resources for future needs.

[0024] Additionally, manually deploying an application on any such converged infrastructure is error prone, as doing so typically requires working knowledge of both the converged infrastructure and the particular application to be deployed. Accordingly, embodiments presented herein include a packaged application delivery mechanism to
30 deploy applications having multiple functional components, which themselves may require complex configuration and setup. For example, in one embodiment, a deterministic approach is used to allocate resources per application within a pool of

resources from the computing block platform. The application can be layered on the allocated resources and delivered in a prepackaged form. The packaged application delivery mechanism discovers a set of available converged infrastructure resources and determines the resources required for a given application that is to be deployed within the computing block platform. The mechanism provides a model that deploys services, functional components, tiers, nodes, etc. required by the application layer on top of the allocated services.

[0025] Further, a converged infrastructure platform may be pre-configured to run an application having multiple functional components. At least one of the functional components may rely on existing services running outside of the converged infrastructure. As such, certain functional components running in the converged infrastructure platform must be configured to communicate with any required services running in the existing data center (e.g., outside of the converged infrastructure platform). Embodiments presented herein use an infrastructure template that describes the existing computing environment (e.g., the data center). The converged infrastructure platform uses the information specified in the infrastructure template to configure its physical resources and virtual resources (e.g., VMs) to allow communication between the converged infrastructure and the rest of the data center. Accordingly, embodiments presented herein reduce the time to integrate components of a converged infrastructure with services of a data center. Further, embodiments presented herein reduce the risk of error during configuration of network resources and security services,

[0026] Figure 1 illustrates an integrated computing platform 102 configured to provide a virtualized environment, according to one embodiment of the present disclosure. A system administrator 150 desires to deploy the integrated computing platform 102 within an existing computing environment (e.g., data center 100). The data center 100 may include a plurality of servers (illustrated as servers 104₁, 104₂, 104_M) that run one or more services 106. It should be recognized that the servers 104 may include conventional computing components (e.g., processor, memory, storage) or may be virtual machines (VMs) executing on such physical hardware. The services 106 running on the servers 104 provide one or more IT functions within the data center, including directory services, web server, database server, accounting, application

serving, file management, storage, backup services, etc. As described in detail below, the system administrator may wish to deploy the integrated computing platform 102 such that the physical resources and virtual resources (e.g., VMs) running inside of the integrated computing platform 102 may communicate with the services 106 of the existing data center 100.

[0027] As shown, the computing platform 102 includes a physical infrastructure 110 configured to support a virtualized infrastructure 120. In the embodiment shown in Figure 1, physical infrastructure 110 includes hardware resources, such as servers 116₁ to 116_N (sometimes referred to as “hosts”) and one or more storage array networks (SAN), such as SAN 118, connected by a network 114. The virtualized infrastructure 120 may include a virtualization environment 124 which itself includes one or more virtual machines 140. The computing platform 102 may be connected to other computing systems, such as workstations, personal computers, data center servers, via a network 128, such as the Internet. In one embodiment, components of the computing platform 102 (e.g., servers, network, storage, software, etc.) may be organized into a single integrated framework sometimes referred to as “converged infrastructure.” The components of the computing platform 102 provide a pool of virtualized server, storage, network resources shared by multiple applications and/or organizations within an enterprise.

[0028] According to one embodiment, the physical infrastructure 110 may be organized into a “computing-block” based infrastructure, wherein physical infrastructure units are characterized by repeatable units of construction having similar performance, operational characteristics, and discrete requirements of power, space, and cooling that facilitate rapid deployment, integration, and scalability. For example, the “computing block” may dynamically provision hardware resources based on performance demands placed on the physical infrastructure 110. One example of physical infrastructure 110 is a Vblock™ System available from the VCE Company, LLC.

[0029] The physical infrastructure 110 further includes an infrastructure manager 112 configured to manage the configuration, provisioning, and policy compliance of the physical infrastructure 110. Infrastructure manager 112 provides an interface to manage the provisioning of hardware resources (e.g., computing, networking, storage) with policy-based automation. According to one embodiment, the infrastructure

manager 112 may be included in each unit of physical infrastructure 110 to manage the configuration, provisioning, and compliance of each distinct computing block. The infrastructure manager 112 may simplify deployment and integration into IT service catalogs and workflow engines, and dramatically simplifies computing-block platform deployment by abstracting the overall provisioning while offering granular access to individual components for troubleshooting and fault management.

[0030] In one embodiment, the infrastructure manager 112 may include a configuration including a list of IP address and system credentials to assign newly provisioned systems. The platform manager 130 and/or the virtualized infrastructure 120 may connect to and communicate with the infrastructure manager 112 of the physical infrastructure 110 to manage and/or configure the physical infrastructure 110. One example of an infrastructure manager includes EMC Ionix Unified Infrastructure Manager (UIM) available from EMC Corporation. Similarly, the network 114 may include a network manager configured to configure network devices (e.g., switches, routers) and manage addressing, subnets, virtual local area networks (VLANs), and other network configurations. One example of a network manager includes a Cisco Switch accessible via a Cisco IOS command line interface (CLI) available from Cisco System, Inc.

[0031] The virtualized infrastructure 120 includes a virtualization environment 124 configured to simulate (i.e., to virtualize) conventional components of a computing device, e.g., a processor, system memory, a hard disk drive, for executing one or more virtual machines 140. For example, each virtual machine 140 may include a virtual processor and a virtual system memory configured to execute an application. In one example implementation of an embodiment similar to that of Figure 1, virtualization environment 124 may be implemented by running VMware vSphere®- or VMware ESX®-based hypervisor technologies on servers 116₁ to 116_n provided by VMware, Inc. of Palo Alto, California (although it should be recognized that any other virtualization technologies, including Xen® and Microsoft Hyper-V virtualization technologies may be utilized consistent with the teachings herein). As described above, a hypervisor application may provide a foundation for building and managing a virtualized IT infrastructure, such as the virtualized infrastructure 120. The hypervisor may abstract

processor, memory, storage and networking resources into multiple virtual machines that run unmodified operating systems and applications.

[0032] In one embodiment, the virtualized infrastructure 120 may be managed by a virtualization manager 122 (e.g., implemented as a process running in a virtual machine in one embodiment). In one embodiment, the virtualization manager 122 may provide end-to-end datacenter management through a set of APIs that enable integration with third-party management tools. The virtualization manager 122 may be configured to manage provision of VMs 140 within the virtualized infrastructure 120 and to configure VMs 140 with computing, network, and storage configurations suitable for interoperability with other VMs 140 within the virtualized infrastructure 120. One example of the virtualization manager 122 may be the VMware vCenter virtualized management platform from available from VMware, Inc.

[0033] As shown, the computing platform 102 also includes a platform manager 130 connected to the virtualized infrastructure 120 and physical infrastructure 110 by the communications network. The platform manager 130 is configured to provision and configure resources from physical infrastructure 110 and virtualized infrastructure 120 for use in deploying an application within the computing platform 102. For example, if virtualized infrastructure 120 requires additional VMs to scale a currently running application during peak traffic, platform manager 130 can coordinate with the virtualization manager 122 to instantiate additional virtual machines to support such needs and configure the instantiated virtual machines with network settings matching those existing virtual machines. In another example, the platform manager 130 may modify an existing VM 140 to add, remove, or configure virtual resources, such as a virtual network interface card (vNIC), assigned to the VM.

[0034] As shown, the platform manager 130 includes a snapshot service 132 configured to save a configuration state of the computing platform 102, across all levels of resources of the computing platform 102, including networking resources (e.g., network 114), storage resources (e.g., SAN 118), computing resources (e.g., servers 116), and virtual resources (e.g., VMs 140). In one embodiment, the snapshot service 132 may save the configuration state as "snapshots" 138 of the networking, storage, computing, and virtual machines. The type of a snapshot 138 may vary depending on the type of resource captured. For example, a snapshot of a VM 140 may comprise a

container file that encapsulates the state of a VM including its documents and data, and information about its virtual hardware (e.g., CPU, memory, disk, etc.); a reference or link to a container file; or a delta file describing differences between one or more container files. In another example, a snapshot of a networking resource (e.g., network 114)

5 includes a log of administrative operations performed on the networking sources, such as adding VLANs, creating DMZs, etc.

[0035] As shown, the snapshot service 132 may maintain a repository of snapshots 138 that may be later used to restore configuration changes to the resources of the computing platform 102. In one embodiment, the repository of snapshots 138 may be
10 implemented as a repository of operations performed on the networking, storage, computing, and virtual resources of the computing platform 102. In operation, the snapshot service 132 may add, remove, and update entries in the repository as needed. For example, the snapshot service 132 may purge a set of entries in the repository of snapshots 138 in response to determining a successful installation and deployment of
15 the computing platform 102 within the data center 100, the set of entries now extraneous.

[0036] In one embodiment, the platform manager 130 includes a restore service 134 configured to restore the computing platform 102 to a previously saved state. The restore service 134 may use snapshots retrieved from the repository of snapshots 138
20 to restore the computing platform 102 to a previous state, for example, such as to a state saved prior to an erroneous installation of the computing platform 102. In some embodiments, the restore service 134 may execute a series of operations selected to undo configuration changes made to networking, storage, computing, and virtual resources. For example, the restore service 134 may instruct a network manager of the
25 network 114 to remove a VLAN that was added during installation, or instruct the infrastructure manager 112 to de-allocate a blade 116 or storage volume that were provisioned during installation.

[0037] In some embodiments, prior to the installation of the computing platform 102, an application having one or more functional components may be installed or loaded
30 onto the computing platform 102. Each functional component of the application performs one or more tasks of the application and/or provide a functional layer of the application (e.g., in a multi-tiered application). The functional components include a

variety of software components, operating systems, and configurations (e.g., executing on a VM 140) that inter-operate to function as a multi-tiered application. For example, the functional components of a deployed web application may include a web server, application server, and database server, each executing on a VM 140 from the virtualized infrastructure 120.

[0038] In some embodiments, an installation process for the computing platform 102 into an existing data center 100 includes configuring resources of the computing platform 102 to enable the pre-loaded application and its functional components that are running within the computing platform 102 to communicate with existing services 106 of the data center 100. In one example, the platform manager 130 may make configuration changes to the networking, storage, computing, and virtual resources of the computing platform 102 for a virtual desktop infrastructure (VDI) running within the computing platform 102, and is shown in greater detail in Figure 2.

[0039] As shown, the platform manager 130 also includes a resource discovery service 133 configured to identify resources (e.g., servers 104, services 106) that are of interest to the system administrator 150 seeking to connect the newly-deployed computing platform 102 with the rest of the data center 100. In some embodiments, the resource discovery service 133 may identify network switches and blades of network 108 that have to be configured to allow communication between the computing platform 102 and the rest of existing data center 100.

[0040] In one embodiment, the platform manager 130 is configured to generate an infrastructure template 139 based on the resources identified by the resource discovery service 133. The infrastructure template 139 includes infrastructure parameters that describe the existing computing environment (e.g., data center 100) in which the integrated computing platform 102 is being deployed. Examples of infrastructure parameters specified by the infrastructure template 139 include: VLAN identifiers that the network 114 of the computing platform 102 should use for communicating with the network 108 of the existing data center 100, IP addresses of a DNS service running within the data center 100, and a range of IP addresses, subnet masks, and gateway IP address to be assigned to VMs 140 of the computing platform 102. In some embodiments, the infrastructure template 139 may be generated specific to an application deployed within the computing platform 102. As such, the infrastructure

parameters contained in the infrastructure template 139 may address specific configurations, settings, and information needed by the application executing within the computing platform 102. Infrastructure template 139 may be implemented using any suitable structured or semi-structured data structure, such as an Extensible Markup Language (XML) document, relational database, and key-value data store.

[0041] In some embodiments, the platform manager 130 may be configured to import and export infrastructure templates 139 allowing them to be transferred between other instances of converged infrastructure (e.g., computing platform 102). The platform manager 130 may also import and export of infrastructure templates 139 to backup known configurations and settings of the computing platform 102, test and deploy a known infrastructure template 139, and perform other such administrative tasks.

[0042] As shown, the platform manager 130 also includes a resource configuration service 135 for configuring physical resources and virtual resources of the computing platform 102 based on an infrastructure template 139. For example, the resource configuration service 135 may configure the network switches and blades based on the infrastructure template 139 to allow communication between the converged infrastructure (e.g., computing platform 102) and the existing data center 100. In another example, the resource configuration service 135 may configure network interface cards (NICs) of VMs 140 running within the computing platform 102 based on infrastructure parameters of the infrastructure template 139.

[0043] Figure 2 illustrates a computing platform 102 executing an application 200 to be integrated into the existing data center 100, according to one embodiment of the present disclosure. As shown, the platform manager 130 may deploy a virtual desktop infrastructure (VDI) 200 on a plurality of VMs 140 within the virtualized infrastructure 120. One example of VDI system 200 includes the VMware View system available from VMware, Inc.

[0044] In a VDI system 200, an end user 210 uses a VDI client software program (e.g., VDI client 212), running on an operating system of a local computing device, to access their desktop which may be running in one of VMs 140 in the computing platform 102 that may be remote from the end user's location. Note that the term "desktop" generally refers to the instance of an interactive operating environment provided by a

computer operating system and software applications, typically in the form of a display and sound output and keyboard and mouse input. With VDI clients 212, users can access remote desktops 206 running in a remote data center (e.g., computing platform 102) through the network 128, from any location, using a general purpose computer
5 running a commodity operating system and a VDI client software program such as VMware® View™, or a special purpose thin client such as those available from Dell, HP, NEC, Sun Microsystems, Wyse, and others.

[0045] As shown, the VDI system 200 includes a connection server 202 that provides user authentication for remote desktops 206 and direct incoming desktop requests (e.g.,
10 from VDI client 212) to a corresponding remote desktop 206. Illustratively, the VDI system 200 further includes one or more security servers 204 (e.g., executing in one or more VMs 140) that enable secure access to the remote desktops 206 from an external network, such as the Internet. The security server 204 may act as a proxy host for connections inside a trust network (e.g., network 114) and shields the connection server
15 202 from request a public-facing Internet. For simplicity, a single network is shown but it should be recognized that, in actual implementations, the components of VDI system 200 may be connected over the same network or different networks. Furthermore, a particular configuration of the virtualized desktop infrastructure is described above and illustrated in Figure 2, but it should be recognized that one or more embodiments of the
20 present invention may be practiced with other configurations of the virtualized desktop infrastructure.

[0046] During installation of the VDI system 200 and the computing platform 102 within the data center 100, the connection server 202 may be connected to a domain controller 208, such as Microsoft® Active Directory®, already running within the existing
25 data center 100 (e.g., on a server 104₂). The domain controller 208 manages user accounts 214 (e.g., a user account for end user 210) including user log-in information and credentials. Further, the connection server 202 and security server 204 may be connected to a domain name system (DNS) service 216 which is installed on a server 104₁ outside of the computing platform 102 to provide domain names to the functional
30 components of the VDI system 200 (e.g., connection server 202, security server 204, and remote desktops 206). The virtualization manager 122 may need to be connected with a “management plane” within the data center 100 and be able to access the

connection server 202 and security server 204. Network architecture for the VDI system 200 may be set up that includes one or more virtual local access networks (VLANs) between the connection server 202, security server 204, remote desktops 206, domain controller 208, DNS service 216 across networks 108 and 114. For example, a system administrator 150 needs access to the connection server 202 and security server 204 to configure them. Further, a VDI client 212 operated by an end user 210 would need access to the connection server 202 and the security server 204 (e.g., via the networks 128, 108, 114). The one-to-many relationships between the functional components of the VDI system 200 and the network 128 of the data center 100 that would need to be extended onto the computing platform 102 to ensure proper operation of the VDI system 200.

[0047] While Figure 2 depicts a particular example of an application (e.g., VDI system 200), other applications running within the computing platform 102 may be deployed that have components that connect to and communicate with services 106 executing in the data center 100. For example, a computing platform 102 having an application with an application server layer, a data grid layer, and a database layer may be integrated within the services of the existing data center 100.

[0048] Figure 3 is a flow diagram of method steps for deploying an integrated computing platform in an existing data center, according to certain aspects of the present disclosure. As shown, the method 300 begins at step 302, where the platform manager 130 captures a “factory state” of the computing platform 102. The term “factory state” may be used to describe a state of the computing platform 102 prior to the resources of the computing platform 102 being initialized and configured (e.g., as in steps 304 and 308 below). In one embodiment, the snapshot service 132 takes an initial snapshot of the networking resources (e.g., network 114), storage resources (e.g., SAN 118), computing resources (e.g., servers 116), and virtual resources (e.g., VMs 140) on the computing platform 102. The operations for capturing a state of the computing platform are described in greater detail in conjunction with Figure 4.

[0049] At step 304, the platform manager 130 provision resources of the computing platform 102 for executing an application and configures the resources to communicate with the data center 100. In some embodiments, the platform manager 130 allocates physical and virtual resources from the computing platform 102 for executing functional

components of the application. In one embodiment, the platform manager 130 configures the resources of the computing platform 102 to inter-operate with services 106 of the data center 100. The platform manager 130 may direct the infrastructure manager 112 to configure the networking resources (e.g., network 114) and hosts (e.g.,
5 servers 116) to assign VLANs, port groups, set quality of service (QoS) settings, and other parameters need to connect the functional components of the application to the services 106 of the data center. The platform manager 130 may direct the virtualization manager 122 to configure VMs 140 to add a network interface card (NIC) for accessing services 106 outside of the computing platform 102 and for services 106 outside of the
10 computing platform 102 to communicate with the VMs 140.

[0050] In the example of the VDI system 200, the platform manager 130 may configure the resources of the computing platform 102 to enable communication between the connection server 202 to the domain controller 208, to enable communication between the security server 204 and the network 108, and to enable
15 connection between the plurality of remote desktops 206 and the network 108. Examples of configurations that may be set include assigning an IP address configured for management of the domain controller 208, assigning an IP address configured for access to the domain controller 208, assigning authentication information for the domain controller 208, assigning a VLAN ID that the computing platform 102 should use for
20 communicating with the “management plane” of the data center 100, assigning VLAN ID that the computing platform 102 should use for communicating with the data center “access” network, assigning IP addresses for management and user access to be assigned to the functional components of the VDI system 200, assigning the IP address of the DNS service 216 on the data center network 108, and assigning a subnet mask
25 and gateway IP address for the VMs 140 within the computing platform 102.

[0051] In one embodiment, the platform manager 130 may configure the resources of the physical infrastructure 110 and virtualized infrastructure 120 based on values provided a system administrator 150 (e.g., entered via a graphical user interface).

[0052] In another embodiment, the platform manager 130 may configure the
30 resources of the physical infrastructure 110 and virtualized infrastructure 120 based on an infrastructure template. An infrastructure template specifies infrastructure parameters that describe the existing computing environment (e.g., data center 100) in

which the integrated computing platform 102 is being deployed. Examples of infrastructure parameters specified by the infrastructure template may include VLAN identifiers that the network 114 of the computing platform 102 should use for communicating with the network 108 of the existing data center 100, IP addresses of a DNS service running within the data center 100, and a range of IP addresses, subnet masks, and gateway IP address to be assigned to VMs 140 of the computing platform 102. In some embodiments, the infrastructure template may be provided for a specific application deployed within the computing platform 102. As such, the infrastructure parameters contained in the infrastructure template may address specific configurations, settings, and information needed by the application executing within the computing platform 102. In one implementation, the infrastructure template may be implemented as an Extensible Markup Language (XML) document, though any suitable structured or semi-structured data structure may be used, such as a relational database or key-value data store. An example of an infrastructure template provided for installing a computing platform 102 having a deployed VDI system 200 with the data center 100 is shown in Table 1 below.

```

<?xml version="1.0" encoding="UTF-8" ?>
<Infra-Template>
  <VLans>
    <Vlan>
      <id>4040</id>
      <name>Infra</name>
    </Vlan>
    <Vlan>
      <id>4051</id>
      <name>Client</name>
    </Vlan>
  </VLans>

  <DHCPVirtualMachines>
    <VMNameFilters>
      <VMNameFilter>*CLIENT*</VMNameFilter>
      <VMNameFilter>SC-*</VMNameFilter>
    </VMNameFilters>
  </DHCPVirtualMachines>

  <VirtualMachinesWithStaticIp>
    <VM>
      <VMName>VM01</VMName>
      <Ip>192.168.15.150</Ip>
      <GatewayIp> 192.168.15.1</GatewayIp>
      <mask>255.255.255.0</mask>
    </VM>
  </VirtualMachinesWithStaticIp>

```

```

        <DNS>192.168.15.150</DNS>
        <Domain>example.com</Domain>
    </VM>
    <VM>
5        <VMName>VM02</VMName>
        <Ip> 192.168.15.151</Ip>
        <GatewayIp> 192.168.15.1</GatewayIp>
        <mask>255.255.255.0</mask>
        <DNS>192.168.15.150</DNS>
10        <Domain>example.com</Domain>
    </VM>
</VirtualMachinesWithStaticIp>
</Infra-Template>

```

Table 1 - Example Infrastructure Template

[0053] In embodiments where the platform manager 130 configures the resources of the computing platform 102 using an infrastructure template, the platform manager 130 may store the infrastructure template within the repository of snapshots 138, as

20 information to be used for any rollback operations in the future.

[0054] At step 306, the snapshot service 132 captures a post-configuration state of the computing platform 102. As described in greater detail later, the restore service 134 may use the captured post-configuration state to restore the computing platform 102 to its factory state.

25 [0055] At step 308, the platform manager 130 installs the application having a plurality of functional components using resources of the computing platform 102. For example, the platform manager 130 may deploy a VDI application (e.g., VDI system 200) that includes a plurality of VMs 140 executing a connection server 202, security server 204, and remote desktops 206. In one embodiment, the platform manager 130

30 may invoke a call that directs the virtualization manager 122 (e.g., through a set of APIs) to create one or more VMs (e.g., VMs 140) having virtual resources (e.g., VRAM, storage) to execute the functional components of the application. The platform manager 130 deploys instances of the functional components on the allocated resources. For example, the platform manager 130 may install software packages onto the provisioned

35 VMs 140, or alternatively, the platform manager 130 may invoke a call that directs the virtualization manager 122 (e.g., through a set of APIs) to create one or more VMs (e.g., VMs 140) based on a pre-packaged VMs having the application components and guest

operating system pre-installed thereon. In some embodiments, the platform manager 130 may create instances of a functional component based on a VM template that defines a VM having pre-installed software components, operating system, and configurations corresponding to a particular functional component.

5 [0056] At step 310, after the installation of the application has been completed, the snapshot service 132 may take a snapshot of the networking, storage, computing, and virtual resources of the computing platform 102 that captures the post-installation state of the converged infrastructure platform (e.g., computing platform 102). The post-installation snapshot may be captured similarly to the snapshots generated in steps 302
10 and 306 above.

[0057] At step 312, the platform manager 130 finalizes set up of the application and launches the application. In the example of the VDI system 200, the platform manager may deploy a pool of remote desktops 206 supported by VMs 140 executing in the virtualized infrastructure 120. In one embodiment, the platform manager 130 may
15 import a VM template that serves as a model for the remote desktops 206 (sometimes referred to as a “gold image”) and deploys multiple VMs based on the gold image to create a pool of remote desktops for the VDI system 200.

[0058] Figure 4 is a flow diagram of method steps for capturing state of an integrated computing platform, according to certain aspects of the present disclosure. Persons
20 skilled in the art will understand that, even though the method 400 is described in conjunction with the systems of Figures 1 and 2, any system configured to perform the method steps, in any order, is within the scope of embodiments of the present disclosure.

[0059] At step 402, the snapshot service 132 takes a snapshot of network
25 configurations for network 114 of the computing platform 102. In some embodiments, the snapshot service 132 records information regarding VLANs assignments, logical networks, port groups, and other network configurations of virtual switches, IP switches, Ethernet switches, and storage switches (e.g., fibre channel) that comprise the networking resources. The snapshot service 132 may store the snapshot of the network
30 configurations in the repository of snapshots 138.

[0060] At step 404, the snapshot service 132 takes a snapshot of computing resources before configuration of the application is complete. For example, the snapshot service 132 records a state of the servers 116 (e.g., blades, server chassis, fabric interconnects). At step 406, the snapshot service 132 takes a snapshot of storage configuration for the storage resources (e.g., SAN 118). For example, the snapshot service 132 records a state of the storage configurations including storage arrays, logical volumes, RAID volumes, data replication, backup and recovery settings. In some embodiments, the snapshot service 132 communicates with the infrastructure manager 112 to obtain configuration states for the computing and storage resources. The snapshot service 132 may store the snapshot of the computing and storage configurations in the repository of snapshots 138.

[0061] At step 408, the snapshot service 132 takes a snapshot of the state of the virtualized infrastructure 120 including the plurality of VMs 140. As shown, the snapshot service 132 determines, for each of the VMs 140, whether a VM had been cloned or instantiated from a VM template. For example, the snapshot service 132 may determine that a VM 140 executing connection server 202 may have been derived from VMs templates for functional components of the VDI system 200. In another example, the snapshot service 132 may determine that a VM 140 may have been created from a VM template from a library of VM templates provided by the virtualization manager 122.

[0062] At step 410, responsive to determining that the VM was not derived from a template, the snapshot service 132 may take a snapshot of the VM 140 and record the snapshot in the repository of snapshots 138. In one embodiment, the snapshot may include a container file containing serialization of all data, environment, and state of the VM 140, for example, formatted in an Open Virtualization Format (OVF) or other suitable structure. At step 412, responsive to determining that a VM was derived from a VM template, rather than take a snapshot of the VM, the snapshot service 132 records an association or link between the VM template and the derived VM. The snapshot service 132 may record the association or link between the derived VM and the VM template within the repository of snapshots 138. It should be recognized that steps 408, 410, and 412 may be repeated for capturing snapshots of each of the VMs 140 in the virtualized infrastructure 120.

[0063] In some embodiments, the snapshots 138 may be implemented as a recordation of all configuration operations executed on resources of the computing platform 102. To capture snapshots, the snapshot service 132 may gather (e.g., from the infrastructure manager and virtualization manager) logs that report what
5 configuration operations were performed on what resources of the computing platform 102. The snapshot service 132 may limit what logged operations it collects based on a time period. For example, the snapshot service 132 may define a “before” snapshot as a point in time prior to configuration of the resources and define an “after” snapshot as a point in time after the configuration of the resources (e.g., step 304 above). The
10 snapshot service 132 may store this collection of operations in the repository of snapshots 138.

[0064] After installation of the application has been completed and the application is deployed, a system administrator 150 may later wish to undo the installation of the application and restore the computing platform 102 to its factory state (or other states
15 during the installation process). For example, the system administrator 150 may determine that the application is operating improperly and may desire to redo the installation process. In another use case, the system administrator 150 may be experimenting with a variety of different deployment schemes during a proof of concept (POC) or testing phase, and wishes to start “fresh” from the beginning of the installation
20 process each time.

[0065] Figure 5 is a flow diagram of method 500 for restoring configuration changes to an integrated computing platform installed in an existing data center, according to certain aspects of the present disclosure. Persons skilled in the art will understand that, even though the method 500 is described in conjunction with the systems of Figures 1
25 and 2, any system configured to perform the method steps, in any order, is within the scope of embodiments of the present disclosure.

[0066] At step 502, the platform manager 130 may receive a “rollback” command, e.g., from the system administrator, for restoring configuration changes to the resources (e.g., network 114, SAN 118, servers 116, VMs 140, etc.) of the computing platform
30 102. In one embodiment, the rollback command may select a snapshot from the repository of snapshots 138 to which the state of the computing platform 102 is to be restored. In another embodiment, the rollback command may indicate a subset of the

resources of the computing platform 102 to be restored. For example, the rollback command may indicate that only configuration changes to the network 114, and not to the storage 118 or VMs 140, are to be restored to the state of the selected snapshot. As such, the rollback command may provide a component-by-component granularity in its ability to restore configuration state of the resources of the computing platform 102. For sake of the foregoing discussion, the rollback command is assumed to indicate all levels of resources (e.g., networking, storage, computing, and virtual machines) have been selected for restoration.

[0067] At step 504, the restore service 134 determines the changes made to the networking, storage, and computing resources based on a comparison of the snapshots taken before and after configuration. In one embodiment, the restore service 134 determines an inverse operation that undoes a configuration change between the before snapshot and the after snapshot. For example, the restore service 134 may determine that a port group was added to the network 114. The restore service 134 may then formulate a “delete” operation to delete the port group from the network 114. In another example, the restore service 134 may determine that a “delete vlan” operation is needed to undo an “add vlan” operation that was performed between snapshots. The restore service 134 may determine similar inverse operations for the storage and computing resources.

[0068] At step 506, the restore service 134 performs the determined inverse operation to undo changes to the network, storage, and computing resources of the computing platform 102. In one embodiment, the restore service 134 communicates with the infrastructure manager 112 (e.g., via an API call) to execute the inverse operation and restore the network 114, storage 118, and servers 116 to their factory state.

[0069] Beginning at step 508, the restore service 134 determines whether the VMs 140 are derived from a VM template. At step 510, the restore service 134 deletes any VMs 140 that were derived from a VM template, retrieves the original VM template, and clones a new instance of the VM based on the VM template. Alternatively, at step 512, responsive to detecting a VM that was not derived from a VM template, the restore service 134 retrieve a snapshot of the VM 140 from the repository of snapshots 138. The restore service 134 directs the virtualization manager 122 to perform a revert

operation to return the state of the VM to that of the snapshot retrieved from the repository. It should be recognized that steps 508, 510, and 512 may be repeated for restoring state of each of the VMs 140 in the virtualized infrastructure 120.

[0070] Figure 6 is a state diagram 600 illustrating a workflow for resetting state of an integrated computing platform, according to certain aspects of the present disclosure. The state diagram 600 includes states 602, 604, 606, 608 that represent the different states of the computing platform 102 during installation of a VDI application (e.g., VDI system 200).

[0071] As shown, after the installation process has been completed (e.g., at state 608), a system administrator 150 may invoke rollback operation that restores the state of the computing platform 102 to one of a variety of states captured during the installation process. In one embodiment, the rollback operation may restore the computing platform 102 to a factory state 602, which undoes any configuration changes made during the installation process. For example, a system administrator 150 may wish to factory reset the computing platform when relocating the computing platform 102 to a new site (e.g., new data center 100).

[0072] In another embodiment, the rollback operation may revert the installation of the VDI application 200 and restore the computing platform 102 to a post-configuration state 604, whereby any changes made to the computing platform 102 during installation of the functional components of the VDI system 200 are reverted. For example, restoring to the post-configuration state 604 would remove the instances of the connection servers 202 and security servers 204 deployed during the installation process. In yet another embodiment, the rollback operation may delete the pool of remote desktops and restore the computing platform 102 to a post-installation state 606, wherein the VDI application 200 and its functional components have been installed. This operation enables a system administrator 150 to replace the “gold image” or VM template that defines instances of the remote desktop 206 and deploy a new pool of remote desktops.

[0073] Accordingly, embodiments of the present disclosure advantageously enables system administrators 150 to easily test their applications and reset the converged infrastructure platform to defaults in order to test various use cases. As a result, the

time and cost to do a proof of concept (POC) or test deployment is greatly reduced, thereby reducing the time to integrate components of a converged infrastructure with services of a data center. Embodiments of the present disclosure further reduce the risk of error during configuration of network resources and security services, which may be a manual and error-prone process.

[0074] As noted above, embodiments presented herein include a packaged application delivery mechanism to deploy applications having multiple functional components. Figure 7 illustrates an example operation for deploying an application in the computing platform of Figure 1, according to one embodiment. A system administrator may provide the platform manager 130 with an application package 136 to be deployed on the computing platform 100 (e.g., on provisioned physical and virtual resources). The application package 700 represents a container of one or more virtual machines having pre-installed software components, operating systems, and configurations that comprise a multi-tiered application. Each VM included in the application package 700 may represent a functional component 702 of the application that performs tasks of the application or provide a functional layer of the application (e.g., in a multi-tiered application). For example, an application package for a typical web application may include a first VM having a pre-installed web server, application server, and application code for the web application, and another VM having a pre-installed database server and configured to be connected to the first VM. The application package 700 may include files, including data, library, and metadata files in a format configured for packaging and distributing virtual machines, such as the Open Virtualization Format (OVF).

[0075] In one embodiment, the application package 700 also includes one or more models 704 indicating relationships between usage of the physical and virtual resources (e.g., VMs 140) and the components 702 of the application. The models 704 may represent the relationships between resources of the computing platform 100 and components 702 of the application using mathematical equations. For example, a model 704 may indicate a relationship between a number of instances (e.g., X) of a particular component of the application and a number of expected users (e.g., Y) of the application. For example, the model 704 may indicate that for a deployment to support Y users of the application, the deployment should include X instances of a component,

where $X=N+1$, and $N=(Y/500)$. Thus, a model 704 for an application may incorporate “best practices” for deploying the application. The particular “best practices” in a given case may be pre-determined, e.g., according to testing and empirical results determined by system engineers or may reflect industry-wide knowledge. In one embodiment, the models 704 may further include a detailed listing of physical resources, virtual resources, settings, and configurations typically needed to deploy an instance of the application. Examples of settings and configurations provided by the models 704 include networking settings, such as network configurations that indicate which components 702 may be located in a “management” portion of the network, in an externally accessible portion of the network, in a DMZ portion of the network, etc.

[0076] As described in detail below, the platform manager 130 is configured to adjust the deployment of the application based on the one or more models 704 and further based on deployment parameters 706 provided by a system administrator. For example, the platform manager 130 may adjust the deployment of a web application by increasing the number of VMs having a pre-installed web server based on a model 704 that indicates an increased number of web server VMs should be deployed based on input from the system administrator indicating the expected amount of traffic for the web application. The system administrator may provide parameters 706 indicating deployment details specific to the instance of the application being deployed. Examples of parameters 706 include an anticipated number of users (e.g., 500 users), volume of traffic (e.g., 200 requests/second; 500 MB/sec), a desired uptime percentage (e.g., 99.999%), and a target resource utilization (e.g., 75% CPU utilization).

[0077] After determining whether (and how) to adjust the deployment of application based on the model 704 and parameters 706, the platform manager 130 may provision virtual and physical resources (e.g., VMs 140-1, 140-2, 140-3, etc.) from computing platform 100 and instantiate individual components 202 of the application package 700 to deploy the application 710 to virtualized infrastructure 120. While Figure 7 depicts individual components 702 as modules executing on provisioned VMs (e.g., VMs 140-1, 140-2, 140-3, etc.), the platform manager 130 may alternatively instantiate complete VMs having pre-installed software components and operating systems based on information specified by the application package 700. The operations of the platform manager 130 are described in greater detail in conjunction with Figure 8.

[0078] Figure 8 is a flow diagram of a method 800 for deploying an application on computer resources in a virtualized environment, according to one embodiment. As shown, the method 800 begins at step 802, where the platform manager 130 receives an application package having one or more functional components (e.g., components 702) for the application to be deployed within the computing platform 100. For example, a system administrator may provide an application package for a VDI system to the platform manager 130. The application package may contain multiple pre-packaged VMs corresponding to each functional component of the application. For example, an application package for a VDI system may contain a pre-configured VM having pre-installed application components, software, libraries, and guest operating system for running a connection server, a pre-configured VM having pre-installed application components, software, libraries, and guest operating system for running a security server, and a pre-configured VM having pre-installed application components, software, libraries, and guest operating system for running a remote desktop.

[0079] At step 804, the platform manager 130 discovers what physical and virtual resources are available for deploying the application in the computing platform 100. The platform manager 130 may process the application package and determine resource requirements (e.g., memory, compute, storage, networking) specified for executing the application, and then query the virtualized infrastructure 120 and physical infrastructure 110 to determine whether such resources are available. For example, the platform manager 130 may determine that a VM corresponding to a functional component of the application requires 4 GB of RAM, a 64-bit processor, at least 200 GB of storage, and at least a gigabit Ethernet connection. The platform manager 130 may then communicate with the infrastructure manager 112 and the virtualization manager 122 to determine whether such resources are available from the physical infrastructure 110 and the virtualized infrastructure 120, respectively.

[0080] At step 806, the platform manager 130 receives deployment parameters (e.g., parameters 706) specifying how to adjust the deployment of the application using models 704. The models 704 indicate how the resources provided by the computing platform 100 may be used by, for example, the functional components specified by the application package 132. The models 704 may indicate a function relating the use of physical resources and virtual resources (e.g., VMs) to a number of instances of each of

the functional components. In one embodiment, the model may specify resources of the computing platform 100 to be used as a function of deployment parameters (e.g., as provided by the system administrator). For example, a model may specify that a deployment of a VDI system (e.g., VDI system 300) should include X instances of a connection server, where $X=N+1$ and $N=(Y/500)$, in order to support Y users of the VDI system. As such, in this example, the system administrator provides a value of "1000" for the deployment parameter of "number of users" (e.g., Y) to the platform manager 130 to adjust the deployment of the VDI system.

[0081] At step 808, the platform manager 130 determines what physical and virtual resources to allocate to the deployment based on a model 704 for the application. Continuing the VDI example described above, the platform manager 130 may evaluate the model 704 using the deployment parameters of "500 users" to determine that the deployment should have three instances of a connection server (e.g., $(1000/500)+1=3$). The platform manager 130 determines an amount of virtual resources (e.g., VMs 140) sufficient to support execute of the three instances of the connection server.

[0082] At step 810, the platform manager 130 allocates physical and virtual resources from the computing platform 100. In one embodiment, the platform manager 130 may invoke a call that directs the virtualization manager 122 (e.g., through a set of APIs) to create one or more VMs (e.g., VMs 140) having virtual resources (e.g., VRAM, storage) to execute one of the functional components of the application. At step 812, the platform manager 130 deploys instances of the functional components 702 on the allocated resources. In embodiments where the functional components 702 are software packages, e.g., the platform manager 130 installs the software packages onto the provisioned VMs 140. Alternatively, in embodiments where the functional components 702 of the application package 700 are pre-packaged VMs, the platform manager 130 may invoke a call that directs the virtualization manager 122 (e.g., through a set of APIs) to create one or more VMs (e.g., VMs 140) as clone of the pre-packaged VMs having the application components and guest operating system pre-installed thereon. For example, the platform manager 130 may create three instances of the connection server VM, one instance of the security server VM, and 50 instances of the remote desktop VM for a deployment of a VDI system.

[0083] At step 814, the platform manager 130 configures the allocated resources and the deployed instances of the functional components as specified by the model 138. In one embodiment, the platform manager 130 may “tune” the allocated resources of the computing platform 100 to meet the functional requirements of the application. For example, the platform manager 130 may modify default networking settings of the allocate resources to meet requirements of a VDI system 300 as specified by the model. Specifically, the platform manager 130 may configure the network 114 to create VLANs for a DMZ, for internal data network, for external access (e.g., via limited ports of a firewall), and for management purposes, where each VLAN may have a particular quality of service (QoS) setting. The platform manager 130 may further configure the deployed instances of the functional components 702 of the application based on adjustments made according to a model 704. For example, referring to the VDI example above, an instance of the connection server may have default configurations that assume the instance is the sole connection server within the VDI system. However, in deployments where multiple connection servers are instantiated (e.g., based on the model 704), the platform manager 130 may modify the default configurations of multiple instances of the connection server to function together (e.g., for replication and load balancing purposes). For example, the platform manager 130 may elect one of the instances of the connection server as a “master” instance and enable a “replica” mode within the other instances, provide each instance with the known IP addresses of the other instances, etc.

[0084] Figure 9 is a flow diagram of a method 900 for integrating a computing platform having an application executing thereon into an existing computing environment, according to one embodiment. As shown, the method 900 begins at step 902, where the platform manager 130 deploys an application having one or more functional components using resources of the converged infrastructure (e.g., computing platform 102). Each functional component of the application performs one or more tasks of the application and/or provide a functional layer of the application (e.g., in a multi-tiered application). The functional components include a variety of software components, operating systems, and configurations (e.g., executing on a VM 140) that inter-operate to function as a multi-tiered application. For example, the functional components of a deployed web application may include a web server, application

server, and database server, each executing on a VM 140 from the virtualized infrastructure 120.

[0085] In some embodiments, the platform manager 130 allocates physical and virtual resources from the computing platform 102 for executing the functional components. In one embodiment, the platform manager 130 may invoke a call that directs the virtualization manager 122 (e.g., through a set of APIs) to create one or more VMs (e.g., VMs 140) having virtual resources (e.g., VRAM, storage) to execute the functional components of the application. The platform manager 130 deploys instances of the functional components on the allocated resources. For example, the platform manager 130 may install software packages onto the provisioned VMs 140, or alternatively, the platform manager 130 may invoke a call that directs the virtualization manager 122 (e.g., through a set of APIs) to create one or more VMs (e.g., VMs 140) based on a pre-packaged VMs having the application components and guest operating system pre-installed thereon.

[0086] At step 904, the platform manager 130 determines the resources of the converged infrastructure (e.g., computing platform 102) that are to be connected to components of the existing computing environment (e.g., data center 100). In some embodiments, the platform manager 130 may receive input, e.g., from the system administrator 150, that identifies which resources of the converged infrastructure have to be configured and what configuration information the identified resources need (e.g., infrastructure parameters). The system administrator 150 may indicate to the platform manager 130 which VMs inside the converged infrastructure need to be accessed from outside of the converged infrastructure. The system administrator 150 may further indicate which networking components (e.g., network switches, hosts) need to be configured to allow integrated of the converged infrastructure 102 with the data center 100. For example, a system administrator 150 may provide input to the platform manager 130 that identifies a VM (e.g., "VM01") running a web server as part of an application executing in the computing platform 102. In this example, the system administrator 150 indicates the web server needs to be configured with at least one port with network connectivity (e.g., via a particular VLAN) to a public Internet for receiving web requests.

[0087] At step 906, the platform manager 130 generates a precursor to an infrastructure template that specifies the infrastructure parameters needed for the converged infrastructure to connect to the existing computing environment 100. The precursor to the infrastructure template may be a “blank” infrastructure template that names which parameters are needed but lacks the corresponding values. In some embodiments, the precursor to the infrastructure template may be pre-determined, and may be pre-generated for a specific application deployed within the computing platform 102.

[0088] At step 908, the platform manager 130 determines values for the infrastructure parameters contained in the infrastructure template 138. In some embodiments, the platform manager 130 may process the blank infrastructure template (e.g., generated at step 906) to determine what infrastructure parameters need to be determined. The platform manager 130 may prompt the system administrator 150 (e.g., via a step-by-step graphical “wizard”) for values to the infrastructure parameters. In some embodiments, the platform manager 130 may derive values for some infrastructure parameters based on values of other infrastructure parameters received from the system administrator. For example, the platform manager 130 may deduce an “N+1” domain naming scheme based on infrastructure parameter values (e.g., “VM01.example.com,” “VM02.example.com”) for servers 104 within the existing data center 100.

[0089] At step 910, the platform manager 130 generates an infrastructure template 138 that describes the environment of the data center 100 using the determined infrastructure parameters and their corresponding values. At step 912, the platform manager 130 configures the resources (e.g., network 114, servers 116, storage 118) of the physical infrastructure 110 based on the infrastructure template 139 to be able to communicate with the data center 100. At step 914, the platform manager 130 configures the resources (e.g., VMs 140) of the virtualized infrastructure 120 based on the infrastructure template 139 to be able to communicate with one or more services 106 of the data center 100. At step 916, the platform manager 130 may optionally export the generated infrastructure template 139 for later use, as described above.

[0090] In some embodiments, an infrastructure template 139 may be configured for integrating a computing platform 102 into an existing data center 100. That is, an

infrastructure template 139 may be configured for a particular application and associated functional components running within the computing platform 102. As such, the platform manager 130 determines values for infrastructure parameters (e.g., configurations, settings, and information) specific to the application executing within the computing platform 102. In one example, an application-dependent infrastructure template 139 may be generated for a virtual desktop infrastructure (VDI) running within the computing platform 102.

[0091] Figure 10 illustrates an example workflow for integrating the computing platform of Figure 2 within an existing data center 100, according to one embodiment of the present disclosure. As shown, the resource discovery service 133 identifies resources (e.g., servers 104) that are of interest to the VDI system 200. For example, the resource discovery service 133 may identify the existing data center 100 having at least a domain controller 208, a DNS service 216, and a network 108. In some embodiments, the resource discovery service 133 discovers resources the VMs 140 that host the functional components of the VDI system 200 that need to be access from outside of the computing platform 102. The resource discovery service 133 further discovers the networking components (e.g., network switches, hosts) that need to be configured to allow integration of the computing platform 102 with the data center 100.

[0092] In one embodiment, the platform manager 130 may use a predetermined blank infrastructure template associated with the VDI system 200 for integrating the computing platform 102 with the data center 100. The infrastructure template 139 identifies configurations, settings, and other set-up anticipated to be needed by the functional components of the VDI system 200. In some embodiments, the infrastructure template 139 for an instance of VDI system 200 may contain configurations for properly connecting a connection server to the domain controller 208, for properly connecting a security server to the network 108, and for properly connecting a plurality of remote desktops to the network 108. Examples of infrastructure parameters specified by the infrastructure template 139 include an IP address configured for management of the domain controller 208, an IP address configured for access to the domain controller 208, authentication information for the domain controller 208, a VLAN ID that the computing platform 102 should use to communicate with a management plane of the data center 100, a VLAN ID that the computing platform 102 should use for communicating with the

data center access network, a range of IP addresses for management and user access to be assigned to the functional components of the VDI system 200, the IP address of the DNS service 216 on the data center network 108, and a subnet mask and gateway IP address for the VMs 140 within the computing platform 102.

5 [0093] As shown, the system administrator 150 provides the resource discovery service 133 with values for infrastructure parameters 1000 (e.g., via a graphical user interface). For example, the system administrator 150, in response to a GUI query, may specify the DNS service 216 running in the data center 100 is located at IP address “192.168.15.150”. In another example, the system administrator 150 may specify a
10 VLAN for functional components of the VDI system 200 having a VLAN ID of “4040” and a VLAN label of “Infra”. In some embodiments, the platform manager 130 may derive values for some infrastructure parameters for the VDI system 200 based on values of other infrastructure parameters received from the system administrator. For example, under certain deployments of a VDI system 200, it may be advantageous to have user
15 accounts 214 organized into a specific “organizational unit” for use with VDI system 200. As such, using the IP address and authentication information of the domain controller network, the platform manager 130 may connect to the domain controller 208 to determine whether such an organizational unit (OU) already exists, and if not, to create one for use with the VDI system 200.

20 [0094] The platform manager 130 generates the infrastructure template 139 for integrating the computing platform 102 having the specific application (e.g., VDI system 200) executing within. In some embodiments, the generated infrastructure template 139 may be exported for later re-use. Additionally, an infrastructure template 139 may be imported from an earlier deployment of an instance of the computing platform 102 and
25 used to integrate the computing platform 102 within the data center 100.

[0095] As shown, the resource configuration service 135 uses the infrastructure template 139 to configure the resources of the computing platform 102 to inter-operate with the services 106 of the data center 100. In one embodiment, the resource configuration service 135 configures the network components (e.g., network 114) and
30 hosts (e.g., servers 116) to create VLANs using parameters specified in the infrastructure template 139. In one embodiment, the resource configuration service 135 adds a network interface card (NIC) to VMs 140 that have need to be accessed from

outside of the computing platform 102 and configures the NICs using parameters specified in the resource configuration service 135.

[0096] Various embodiments of the present disclosure may be implemented as a program product for use with a computer system. The program(s) of the program product define functions of the embodiments (including the methods described herein) and can be contained on a variety of computer-readable storage media. Illustrative computer-readable storage media include, but are not limited to: (i) non-writable storage media (e.g., read-only memory devices within a computer such as CD-ROM disks readable by a CD-ROM drive, flash memory, ROM chips or any type of solid-state non-volatile semiconductor memory) on which information is permanently stored; and (ii) writable storage media (e.g., hard-disk drive, a USB flash memory device, etc.) on which alterable information is stored.

[0097] The invention has been described above with reference to specific embodiments and numerous specific details are set forth to provide a more thorough understanding of the invention. Persons skilled in the art, however, will understand that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The foregoing description and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

[0098] While the foregoing is directed to embodiments of the present disclosure, other and further embodiments of the present disclosure may be devised without departing from the basic scope thereof, and the scope thereof is determined by the claims that follow.

Claims:

1. A method for installing an integrated computing platform into an existing computing environment, the method comprising:
 - 5 configuring an integrated computing platform to communicate with at least one service running in an existing computing environment outside of the integrated computing platform, wherein the integrated computing platform includes an application having a plurality of functional components using a plurality of networking, storage, computing, and virtualized resources of the integrated computing platform; and
 - 10 restoring each of the plurality of networking, storage, computing, and virtualized resources of the integrated computing platform to a state prior to the configuring the integrated computing platform to communicate with the service running in the existing computing environment outside of the integrated computing platform.
- 15 2. The method of claim 1, wherein the integrated computing platform is configured to enable communication between the functional components of the application and the service running within the existing computing environment.
3. The method of claim 1, further comprising:
 - 20 prior to the restoring:
 - generating a snapshot of the plurality of networking, storage, computing, and virtualized resources of the integrated computing platform, wherein the snapshot represents a configuration state of the resources of the integrated computing platform; and
 - 25 determining a plurality of configuration operations performed on the plurality of networking, storage, computing, and virtualized resources of the integrated computing platform during installation of the application.
4. The method of claim 1, wherein restoring the plurality of networking, storage, computing, and virtualized resources of the integrated computing platform comprises:
 - 30 determining an inverse operation to undo a configuration change made to the plurality of networking, storage, computing, and virtualized resources of the integrated computing platform.

5. The method of claim 1, wherein restoring each of the plurality of networking, storage, computing, and virtualized resources comprises:

restoring a configuration state of the networking, storage, computing, and
5 virtualized resources based on a snapshot of the configuration state.

6. The method of claim 1, wherein the virtualized resources comprise a plurality of virtual machines (VMs), and wherein restoring each of the virtualized resources comprises:

10 responsive to determining a VM is derived from a VM template, deleting the VM and creating a new instance of the VM template; and

responsive to determining a VM is not derived from a VM template, retrieving a snapshot of the VM from a repository of snapshots and reverting the state of the VM based on the retrieved snapshot.

15

7. The method of claim 1, wherein the functional components comprise virtual machines executing application components of a virtual desktop infrastructure.

8. An integrated computing platform, comprising:

20 a plurality of networking, storage, computing, and virtualized resources having an application with a plurality of functional components executing thereon; and
a management server configured to:

25 configure the integrated computing platform to communicate with at least one service running in an existing computing environment outside of the integrated computing platform, and

restore each of the plurality of networking, storage, computing, and
virtualized resources of the integrated computing platform to a state prior to the
configuring the integrated computing platform to communicate with the service
running in the existing computing environment outside of the integrated
30 computing platform.

9. The integrated computing platform of claim 8, wherein the integrated computing platform is configured to enable communication between the functional components of the application and the service running within the existing computing environment.

5 10. The integrated computing platform of claim 8, wherein the management server is further configured to, prior to restoring, to generate a snapshot of the plurality of networking, storage, computing, and virtualized resources of the integrated computing platform, wherein the snapshot represents a configuration state of the resources of the integrated computing platform.

10

11. The integrated computing platform of claim 8, wherein the management server is further configured to, prior to the restoring, to determine a plurality of configuration operations performed on the plurality of networking, storage, computing, and virtualized resources of the integrated computing platform during installation of the application.

15

12. The integrated computing platform of claim 8, wherein the management server is further configured to:

determine an inverse operation to undo a configuration change made to the plurality of networking, storage, computing, and virtualized resources of the integrated computing platform.

20

13. The integrated computing platform of claim 8, wherein the management server is further configured to:

restore a configuration state of the networking, storage, computing, and virtualized resources based on a snapshot of the configuration state.

25

14. The integrated computing platform of claim 8, wherein the virtualized resources comprise a plurality of virtual machines (VMs), and wherein the management server is further configured to:

responsive to determining a VM is derived from a VM template, delete the VM and creating a new instance of the VM template; and

30

responsive to determining a VM is not derived from a VM template, retrieve a snapshot of the VM from a repository of snapshots and revert the state of the VM based on the retrieved snapshot.

5 15. The integrated computing platform of claim 8, wherein the functional components comprise virtual machines executing application components of a virtual desktop infrastructure.

16. A computer-readable storage medium comprising instructions that, when executed in a computing device, install an integrated computing platform into an existing
10 computing environment, by performing the steps of:

configuring an integrated computing platform to communicate with at least one service running in an existing computing environment outside of the integrated computing platform, wherein the integrated computing platform includes an application having a plurality of functional components using a plurality of networking, storage,
15 computing, and virtualized resources of the integrated computing platform; and

restoring each of the plurality of networking, storage, computing, and virtualized resources of the integrated computing platform to a state prior to the configuring the integrated computing platform to communicate with the service running in the existing computing environment outside of the integrated computing platform.

20

17. The computer-readable storage medium of claim 17, wherein the operation further comprises, prior to the restoring:

generating a snapshot of the plurality of networking, storage, computing, and virtualized resources of the integrated computing platform, wherein the snapshot
25 represents a configuration state of the resources of the integrated computing platform; and

determining a plurality of configuration operations performed on the plurality of networking, storage, computing, and virtualized resources of the integrated computing platform during installation of the application.

30 18. An integrated computing platform, comprising:

means for configuring an integrated computing platform to communicate with at least one service running in an existing computing environment outside of the integrated

computing platform, wherein the integrated computing platform includes an application having a plurality of functional components using a plurality of networking, storage, computing, and virtualized resources of the integrated computing platform; and

5 means for restoring each of the plurality of networking, storage, computing, and virtualized resources of the integrated computing platform to a state prior to the configuring the integrated computing platform to communicate with the service running in the existing computing environment outside of the integrated computing platform.

19. The integrated computing platform of claim 18, further comprising:

10 means for generating, prior to the restoring, a snapshot of the plurality of networking, storage, computing, and virtualized resources of the integrated computing platform, wherein the snapshot represents a configuration state of the resources of the integrated computing platform; and

15 means for generating, prior to the restoring, determining a plurality of configuration operations performed on the plurality of networking, storage, computing, and virtualized resources of the integrated computing platform during installation of the application.

20. The integrated computing platform of claim 18, further comprising:

means for restoring a configuration state of the networking, storage, computing, and virtualized resources based on a snapshot of the configuration state.

20

1/9

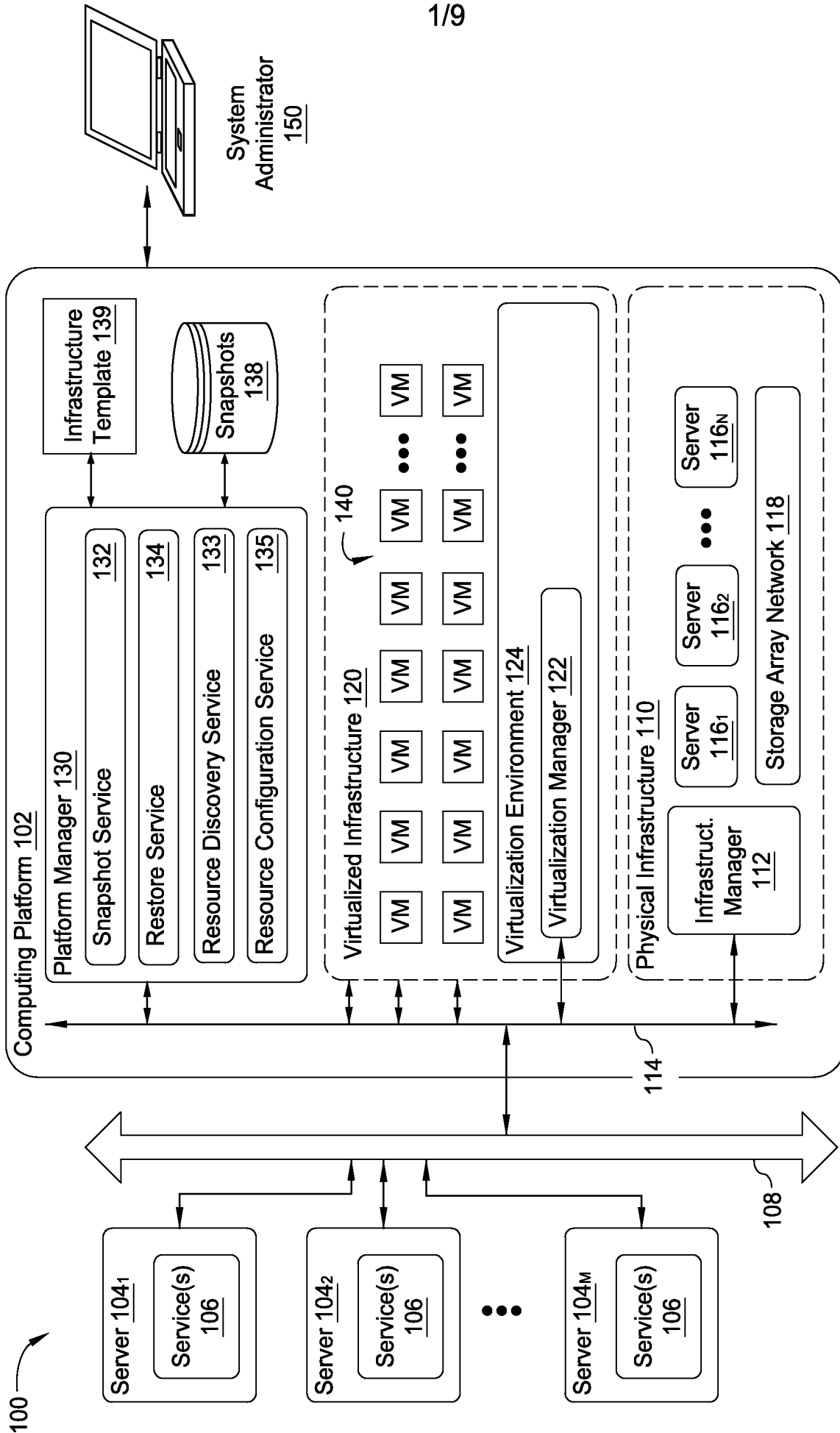


FIG. 1

2/9

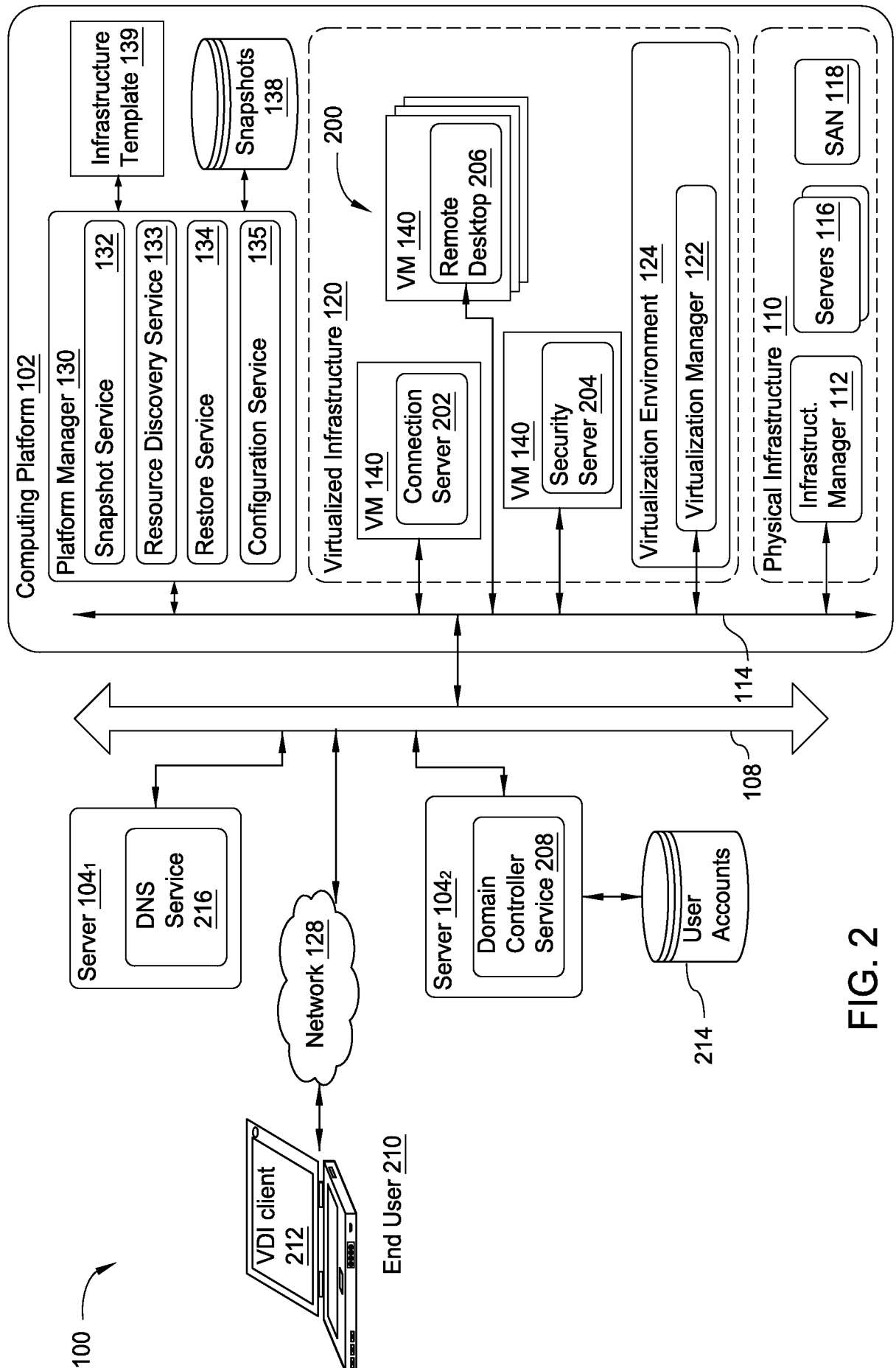


FIG. 2

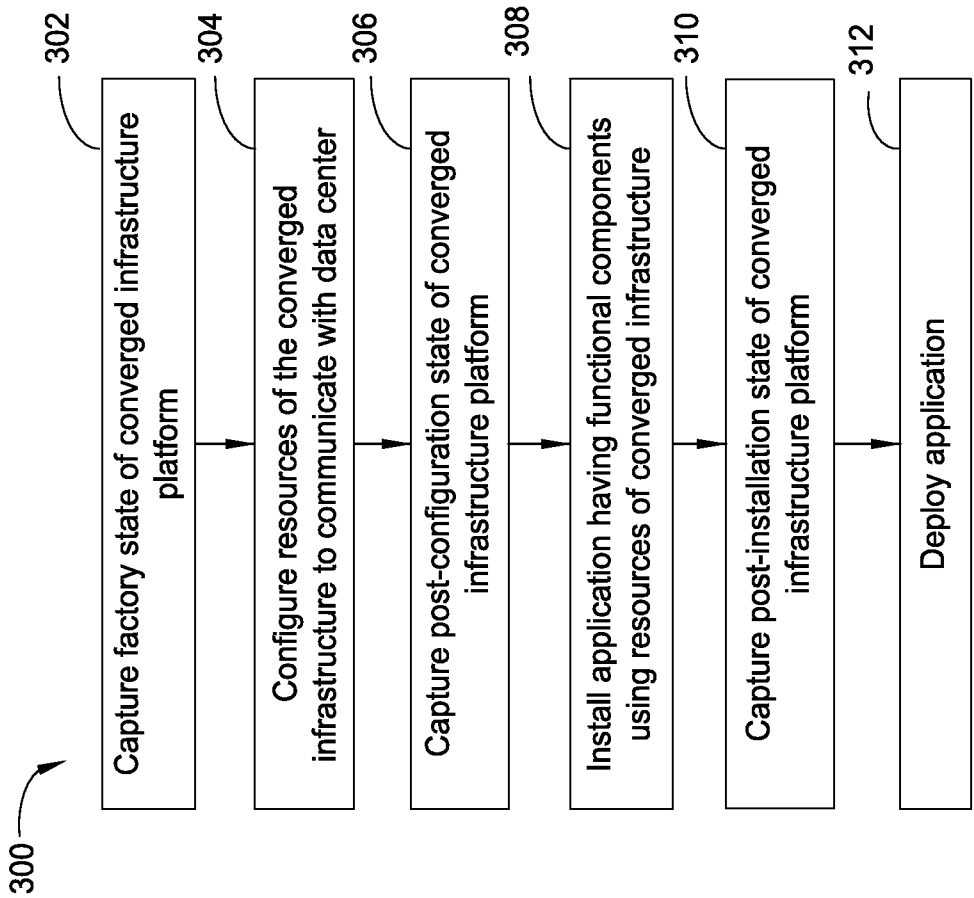


FIG. 3

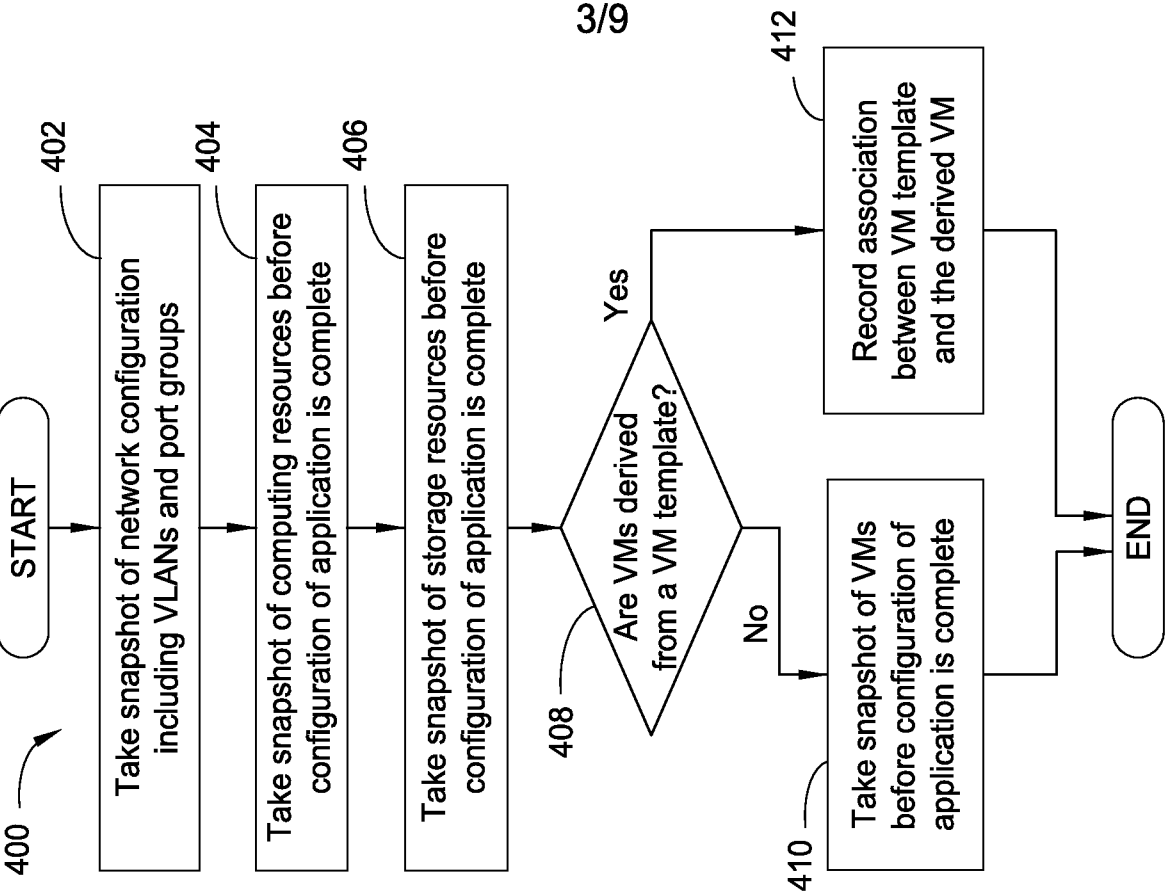


FIG. 4

4/9

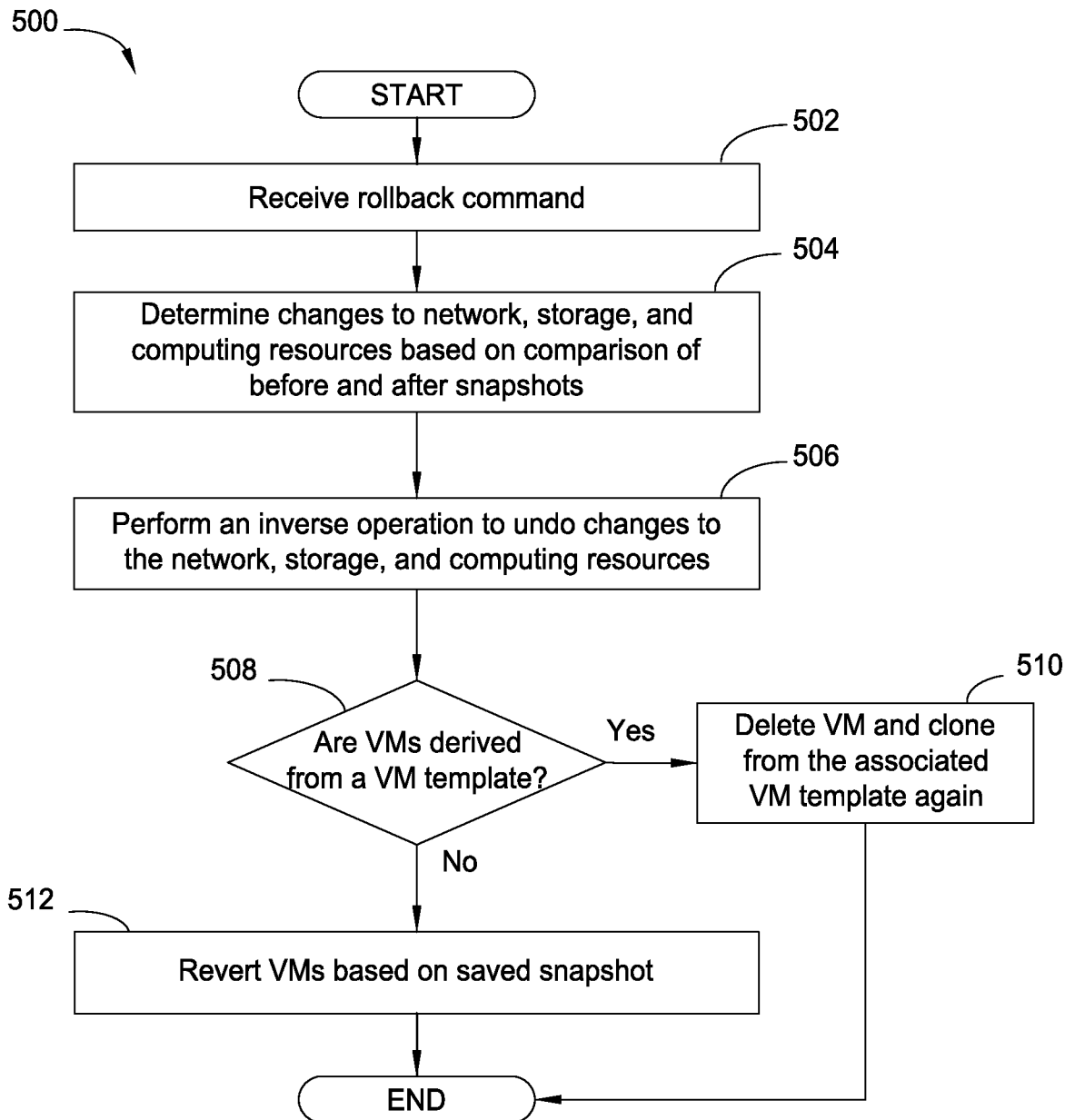
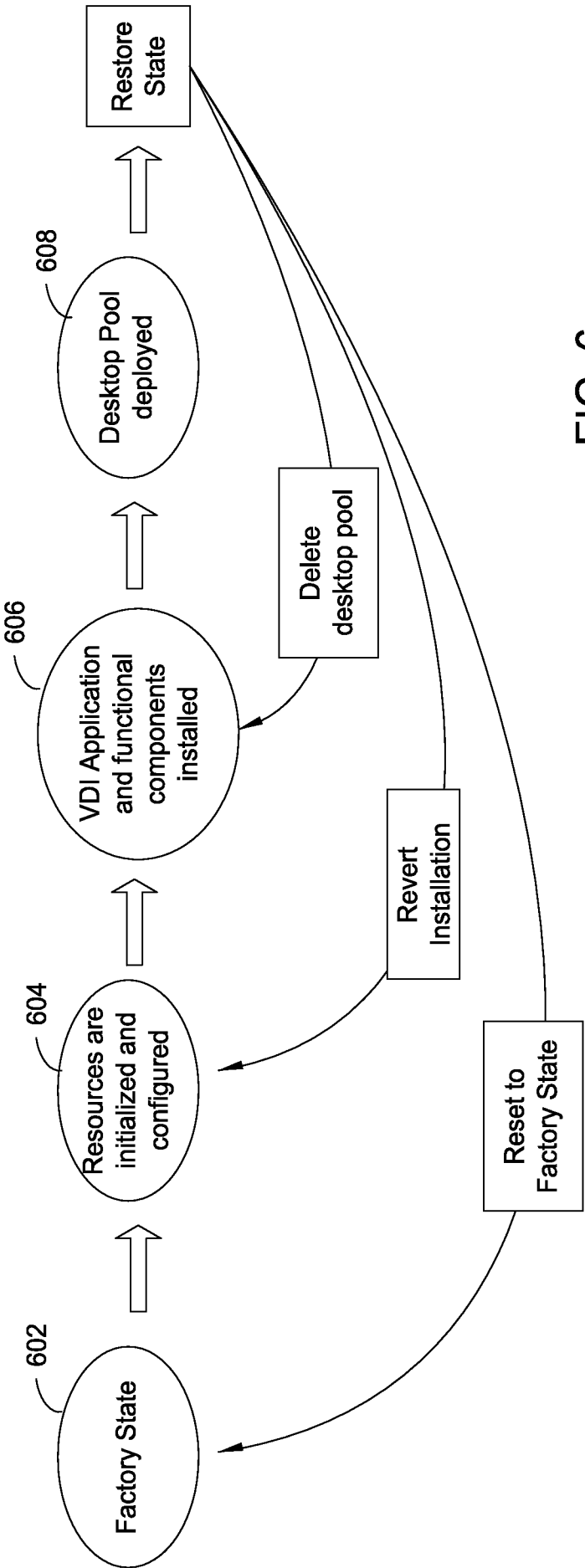


FIG. 5

600



5/9

FIG. 6

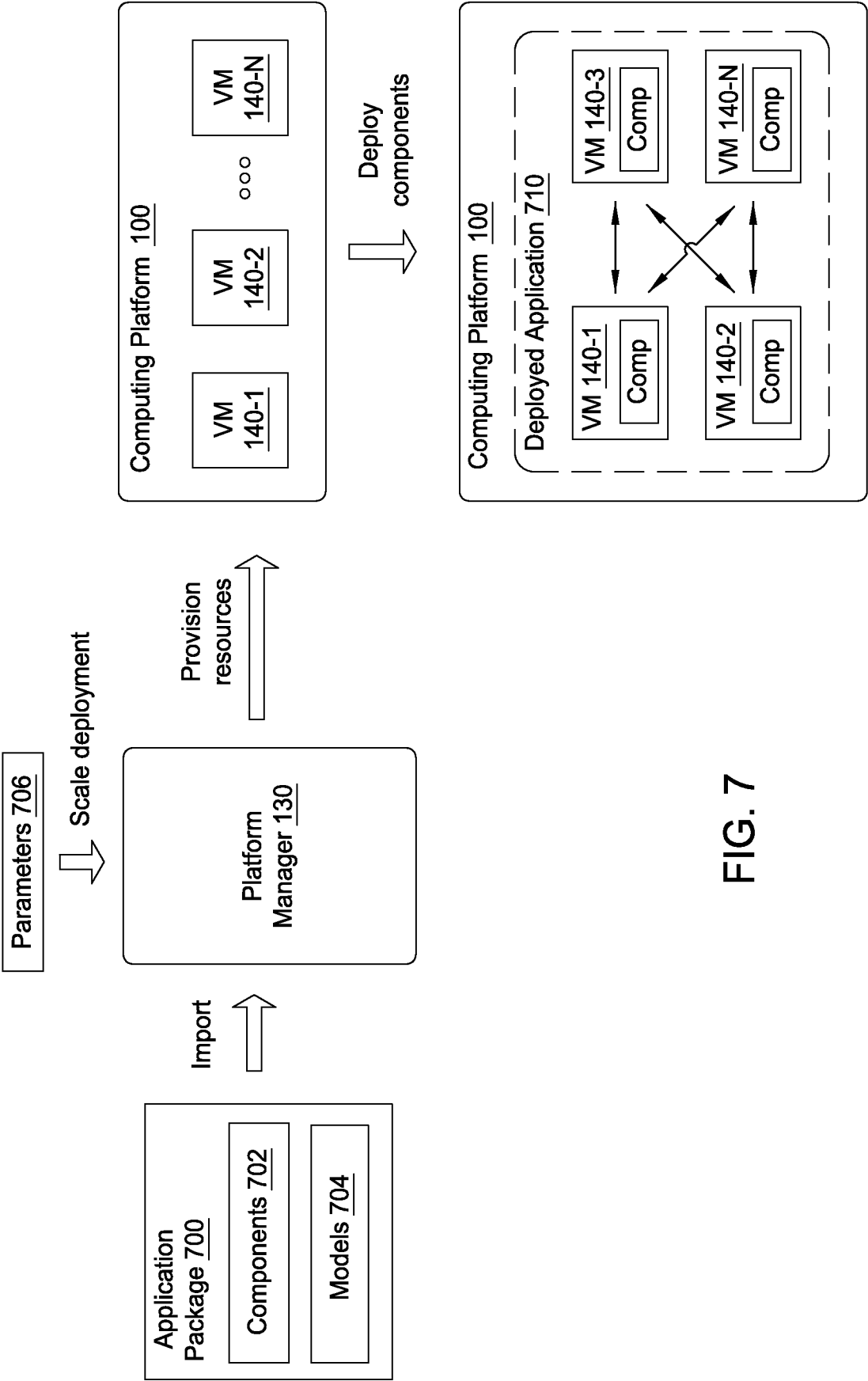


FIG. 7

7/9

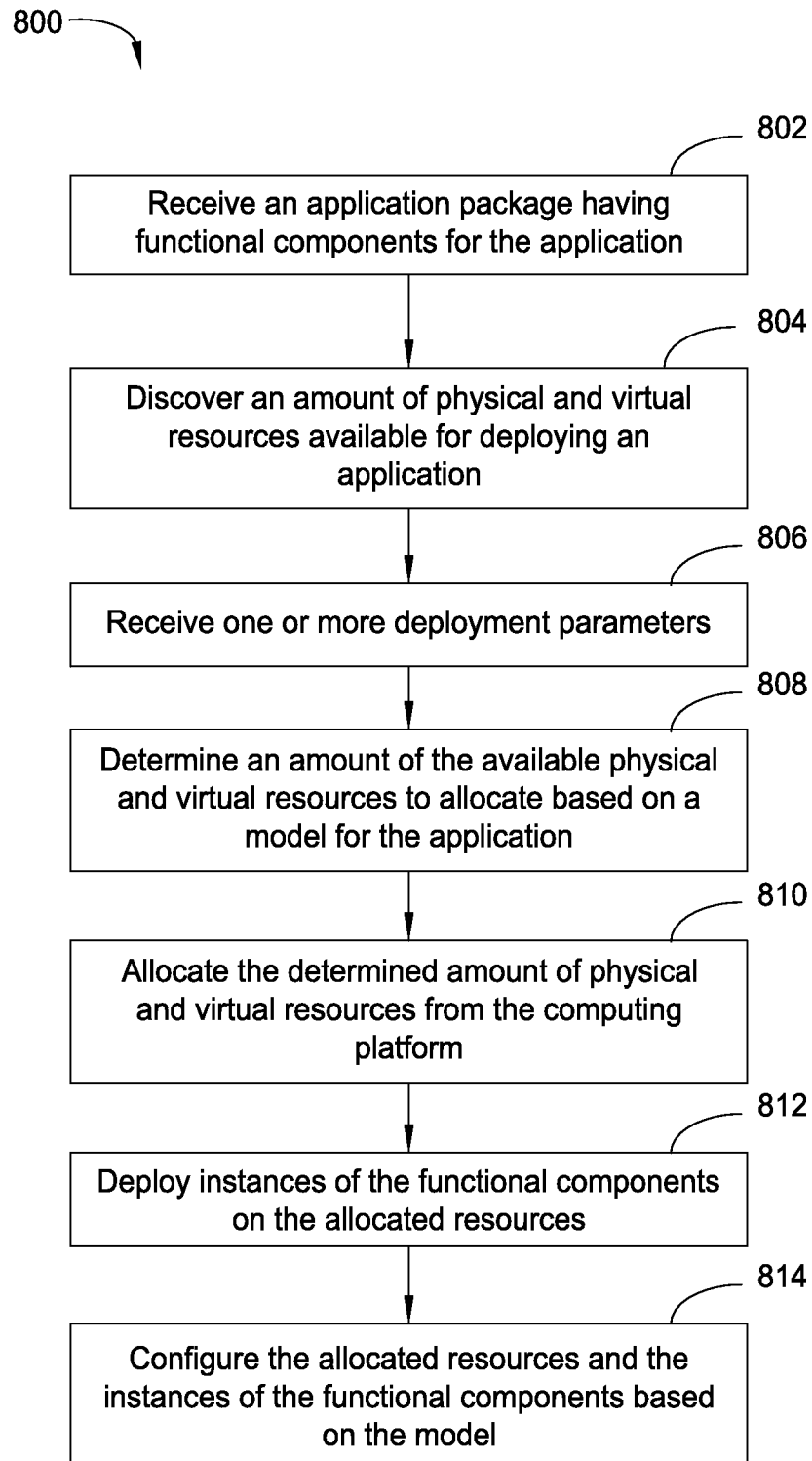


FIG. 8

8/9

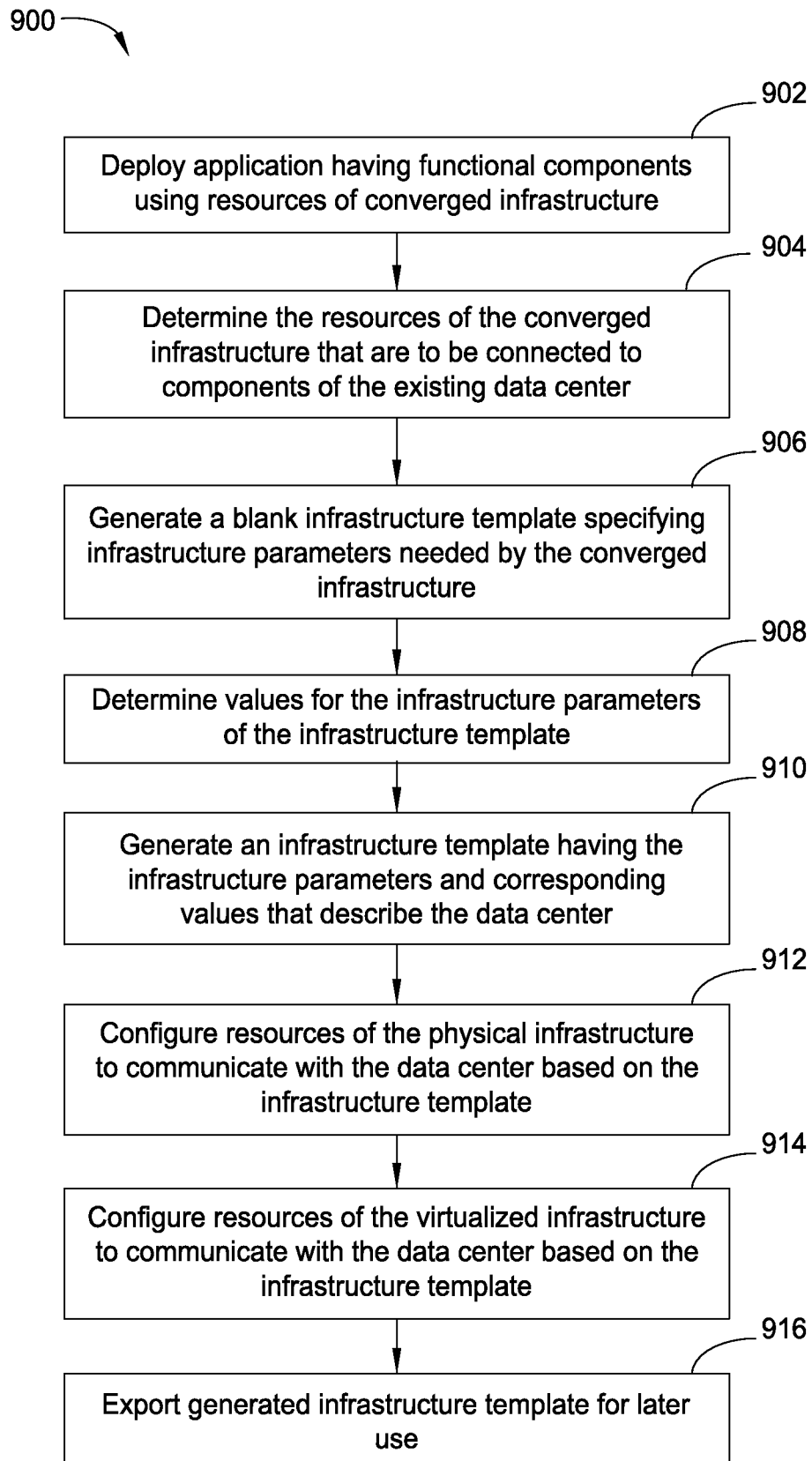


FIG. 9

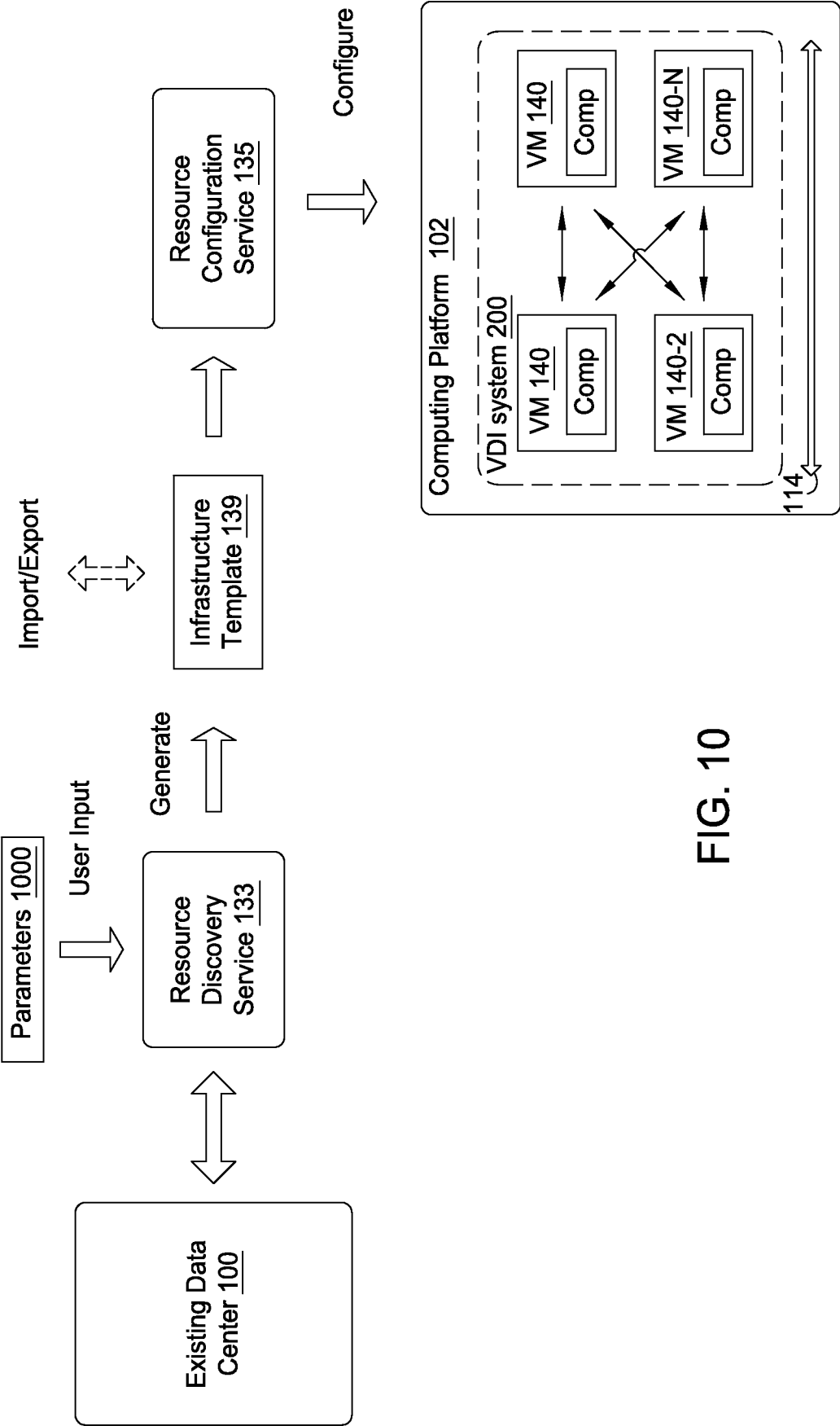


FIG. 10

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2013/057075

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/445
ADD.

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

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	F Oliveira ET AL: "Blutopia: Cluster Life-cycle Management", 7 November 2005 (2005-11-07), XP055089018, Retrieved from the Internet: URL:http://domino.research.ibm.com/library/cyberdig.nsf/papers/5A54BE032FC9E8F4852570B5005F82D6/\$File/rc23784.pdf [retrieved on 2013-11-19] page 3 - page 5 page 9	1-20
X	----- US 2008/250405 A1 (FARHANGI ALIREZA [US] ET AL) 9 October 2008 (2008-10-09) paragraph [0028] - paragraph [0030] ----- -/-	1-20



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

19 November 2013

Date of mailing of the international search report

28/11/2013

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Dewyn, Torkild

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2013/057075

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2003/126202 A1 (WATT CHARLES T [US]) 3 July 2003 (2003-07-03) the whole document -----	1-20
A	US 2011/161947 A1 (ASHOK ROHITH K [US] ET AL) 30 June 2011 (2011-06-30) the whole document -----	1-20

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2013/057075

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2008250405	A1	09-10-2008	NONE
US 2003126202	A1	03-07-2003	US 2003126202 A1 03-07-2003
		US 2007250608 A1	25-10-2007
US 2011161947	A1	30-06-2011	NONE