



US008650299B1

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

(10) **Patent No.:** **US 8,650,299 B1**
(45) **Date of Patent:** **Feb. 11, 2014**

(54) **SCALABLE CLOUD COMPUTING**

(56) **References Cited**

(75) Inventors: **Alex Huang**, Cupertino, CA (US); **Will Chan**, Santa Clara, CA (US); **Sheng Liang**, Cupertino, CA (US); **Chiradeep Vittal**, Cupertino, CA (US)

U.S. PATENT DOCUMENTS

2009/0222558 A1 9/2009 Xu et al.
2009/0276771 A1* 11/2009 Nickolov et al. 717/177
2009/0300605 A1 12/2009 Edwards et al.
2011/0022812 A1 1/2011 van der Linden et al.

(73) Assignee: **Citrix Systems, Inc.**, Fort Lauderdale, FL (US)

* cited by examiner

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 215 days.

Primary Examiner — Larry Donaghue
Assistant Examiner — Marshall McLeod

(74) *Attorney, Agent, or Firm* — BainwoodHuang

(21) Appl. No.: **13/020,799**

(57) **ABSTRACT**

(22) Filed: **Feb. 3, 2011**

A scalable cloud infrastructure serves two or more customers, where each customer is associated with at least one unit of virtual resources. The virtual resources are established by apportioning physical resources in the cloud infrastructure that are partitioned into pods within one or more zones in a scalable manner. Each pod comprises a strongly connected discrete set of physical resources in a zone. Additionally, the cloud infrastructure establishes one or more management server clusters each comprising one or more management servers. Each management server is configured for allocating the units of virtual resources to an account associated with each customer. The traffic from two or more customer accounts within the scalable cloud infrastructure is isolated and segmented.

Related U.S. Application Data

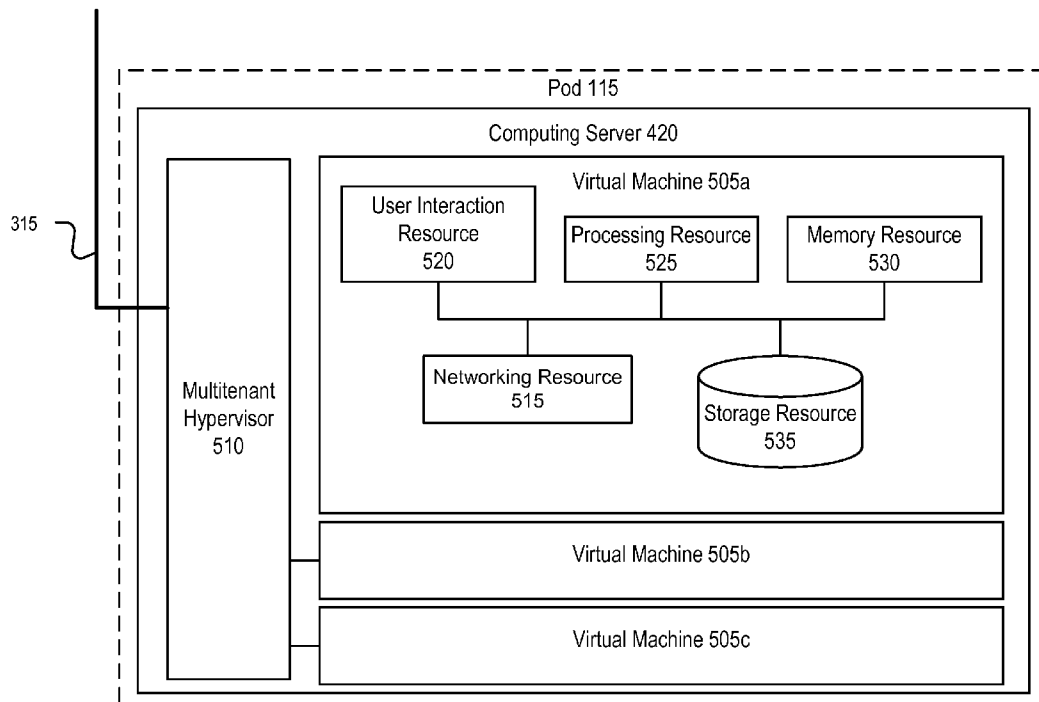
(60) Provisional application No. 61/301,168, filed on Feb. 3, 2010.

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

(52) **U.S. Cl.**
USPC **709/226**

(58) **Field of Classification Search**
USPC 709/223, 226
See application file for complete search history.

19 Claims, 12 Drawing Sheets



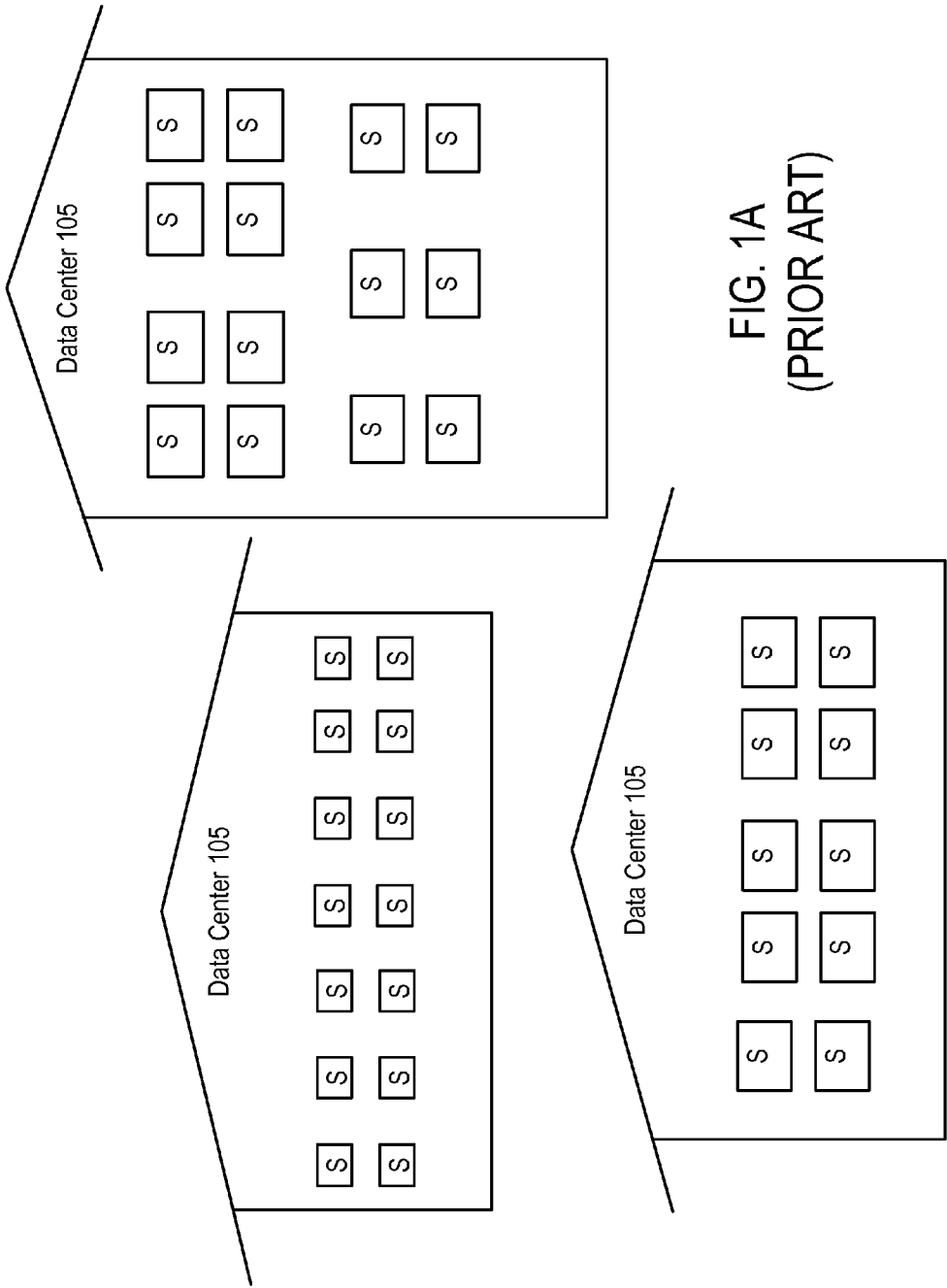


FIG. 1A
(PRIOR ART)

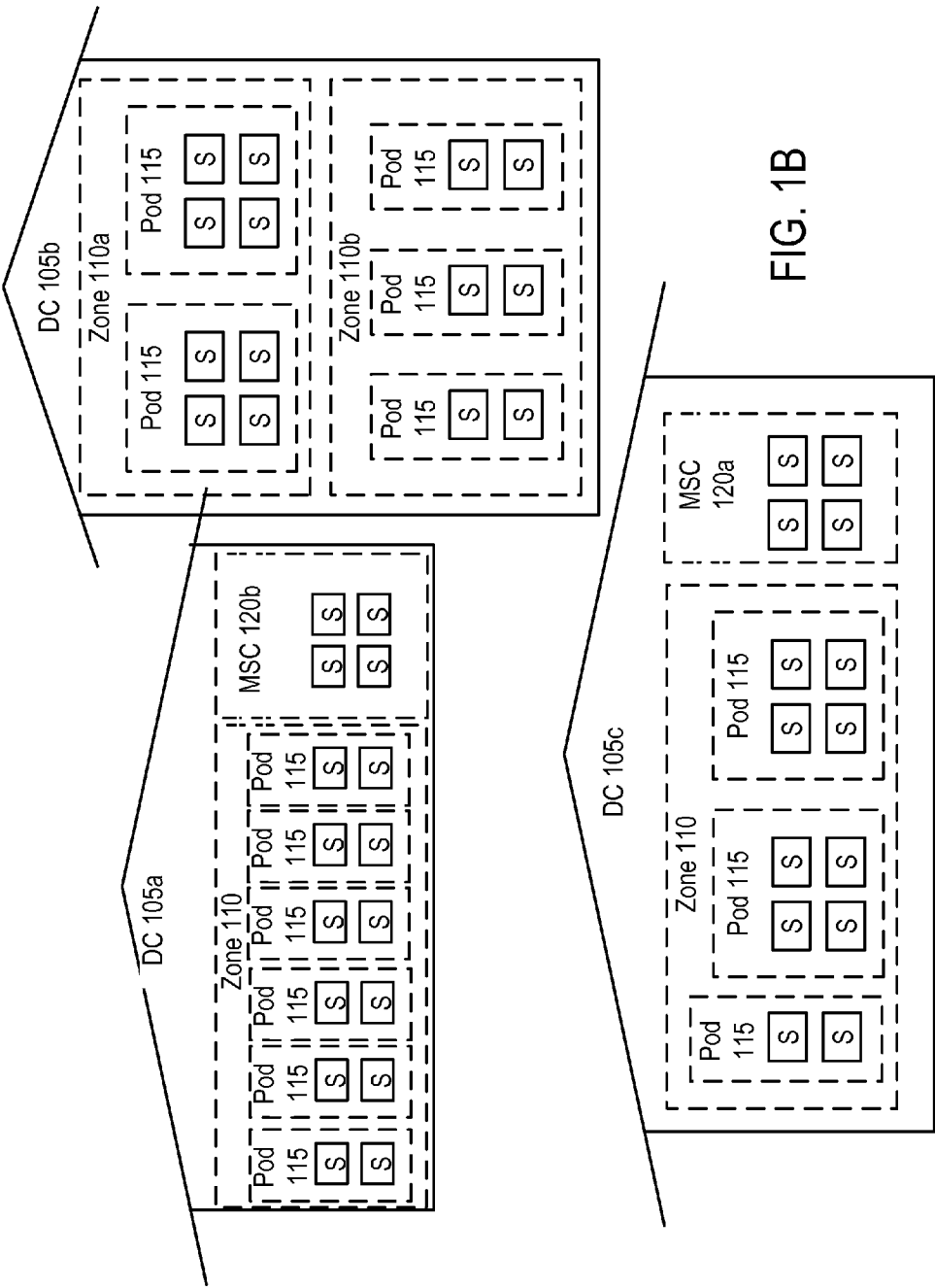
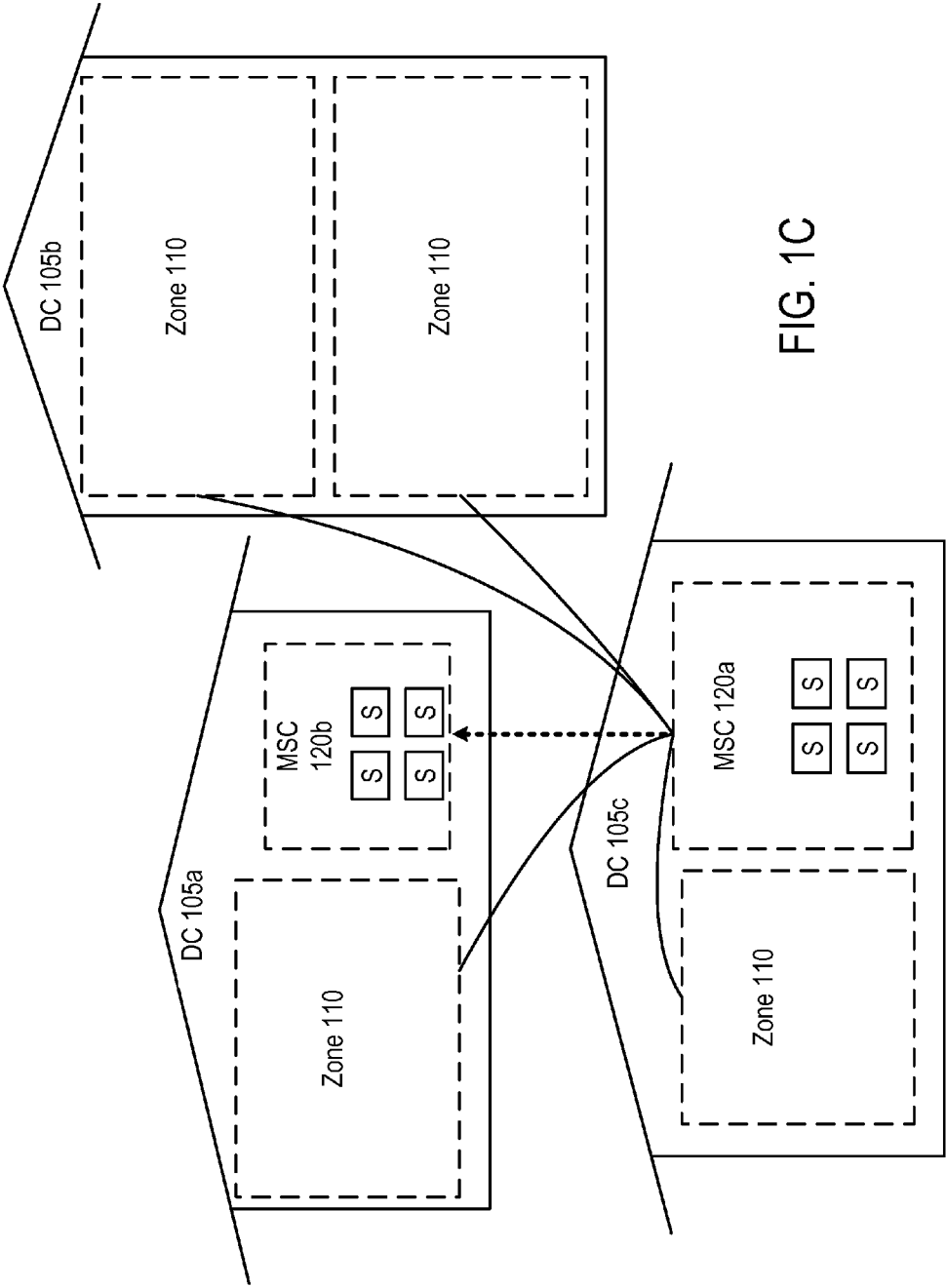


FIG. 1B



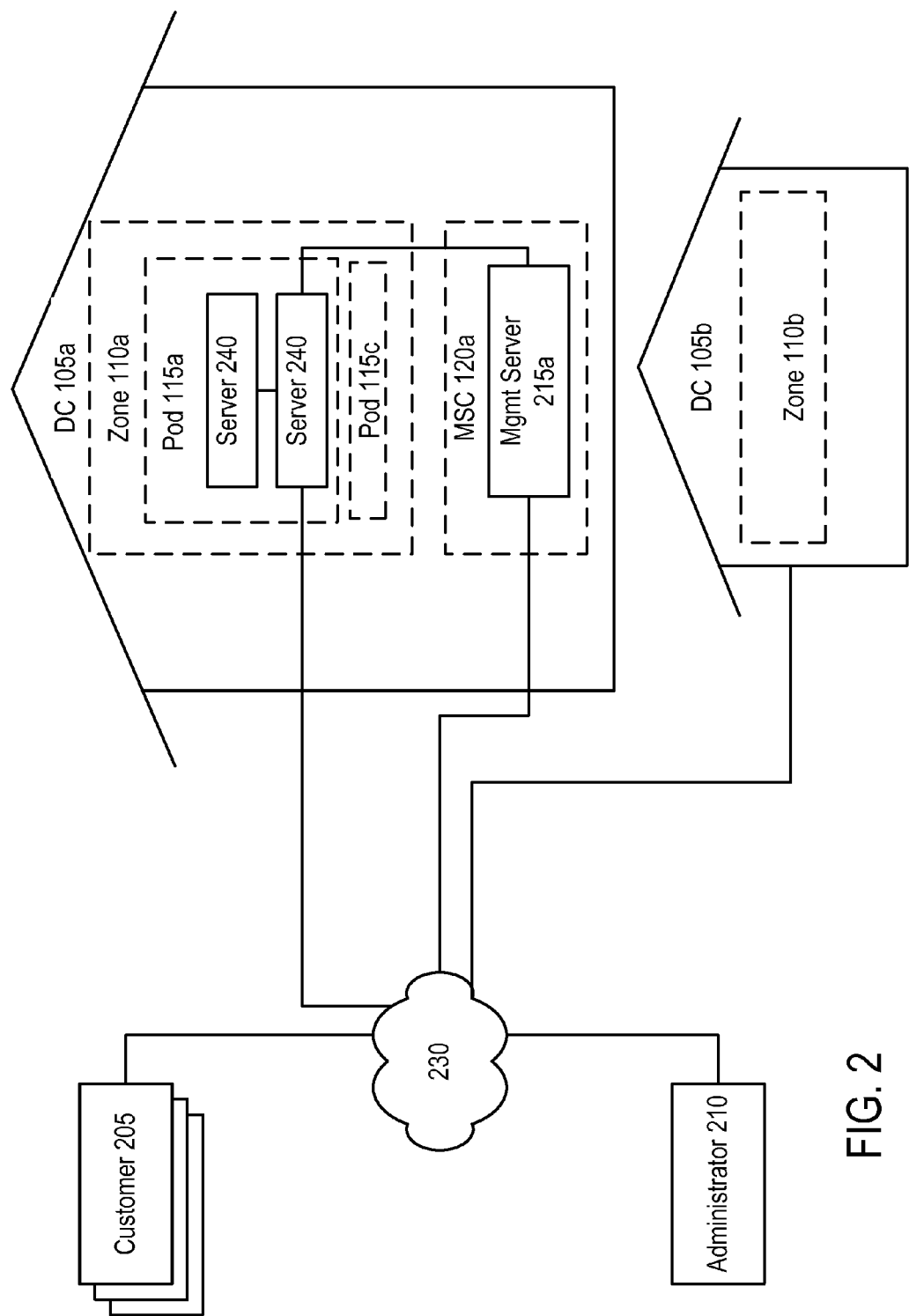


FIG. 2

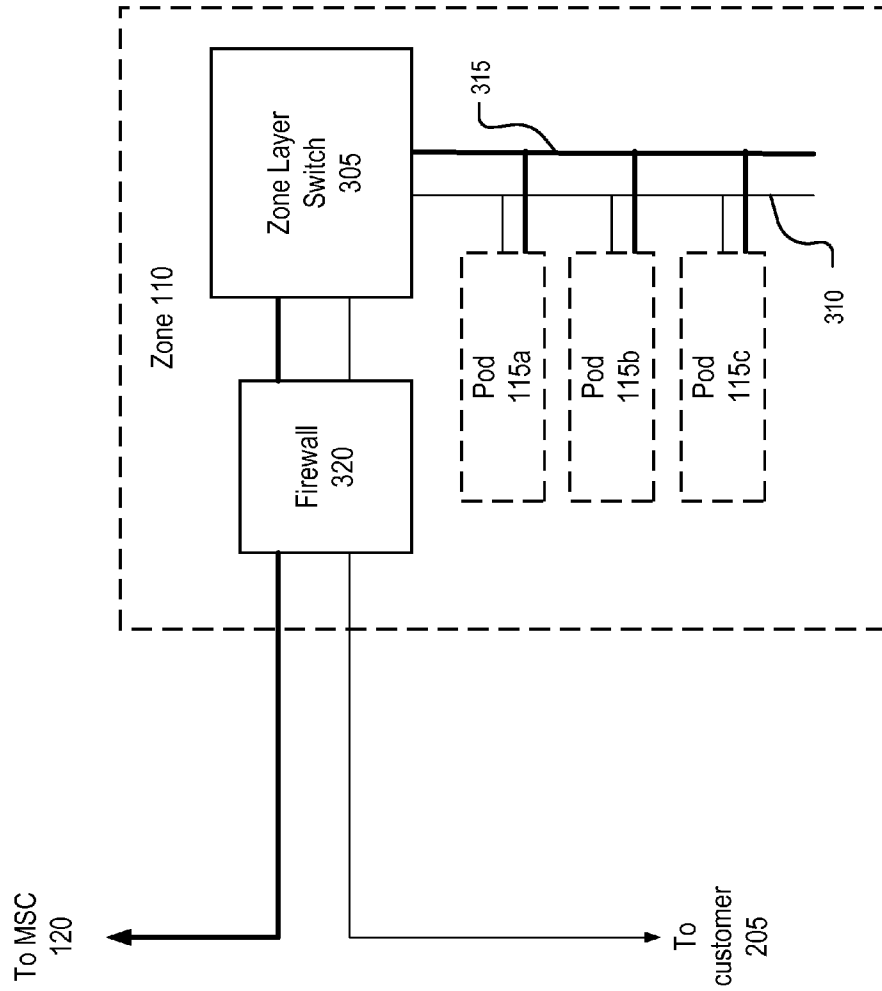


FIG. 3

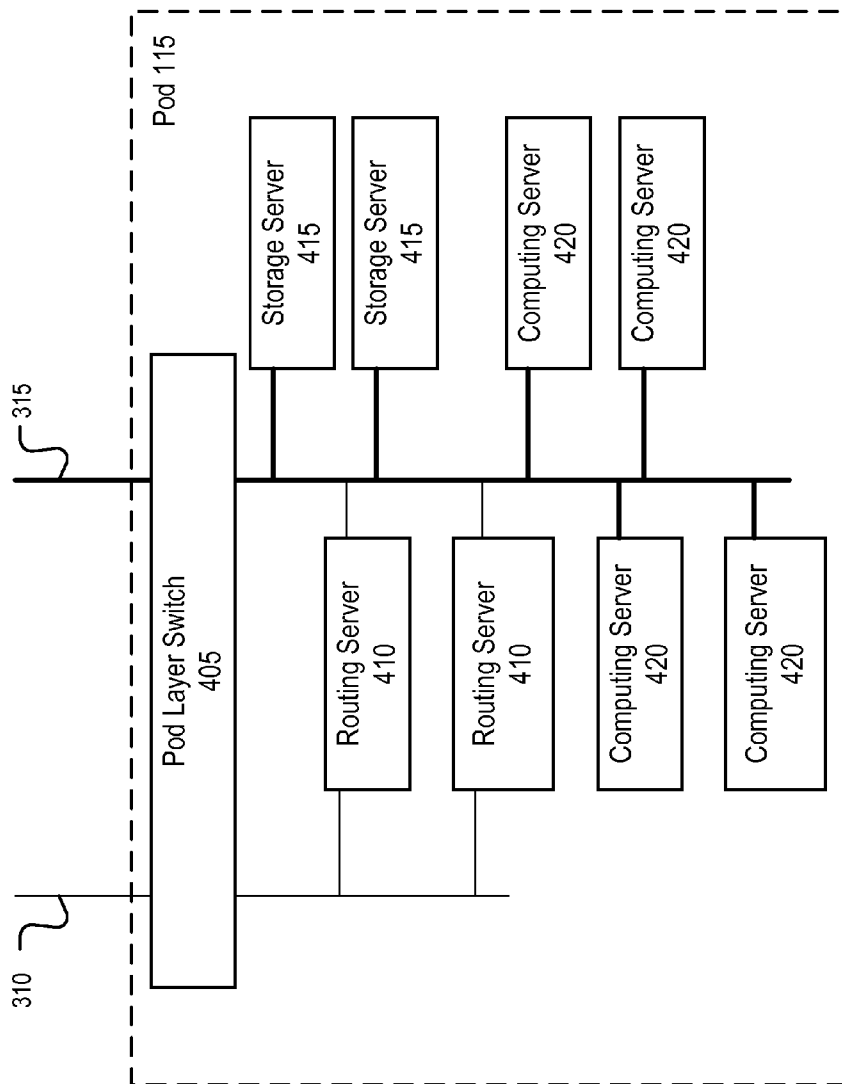


FIG. 4

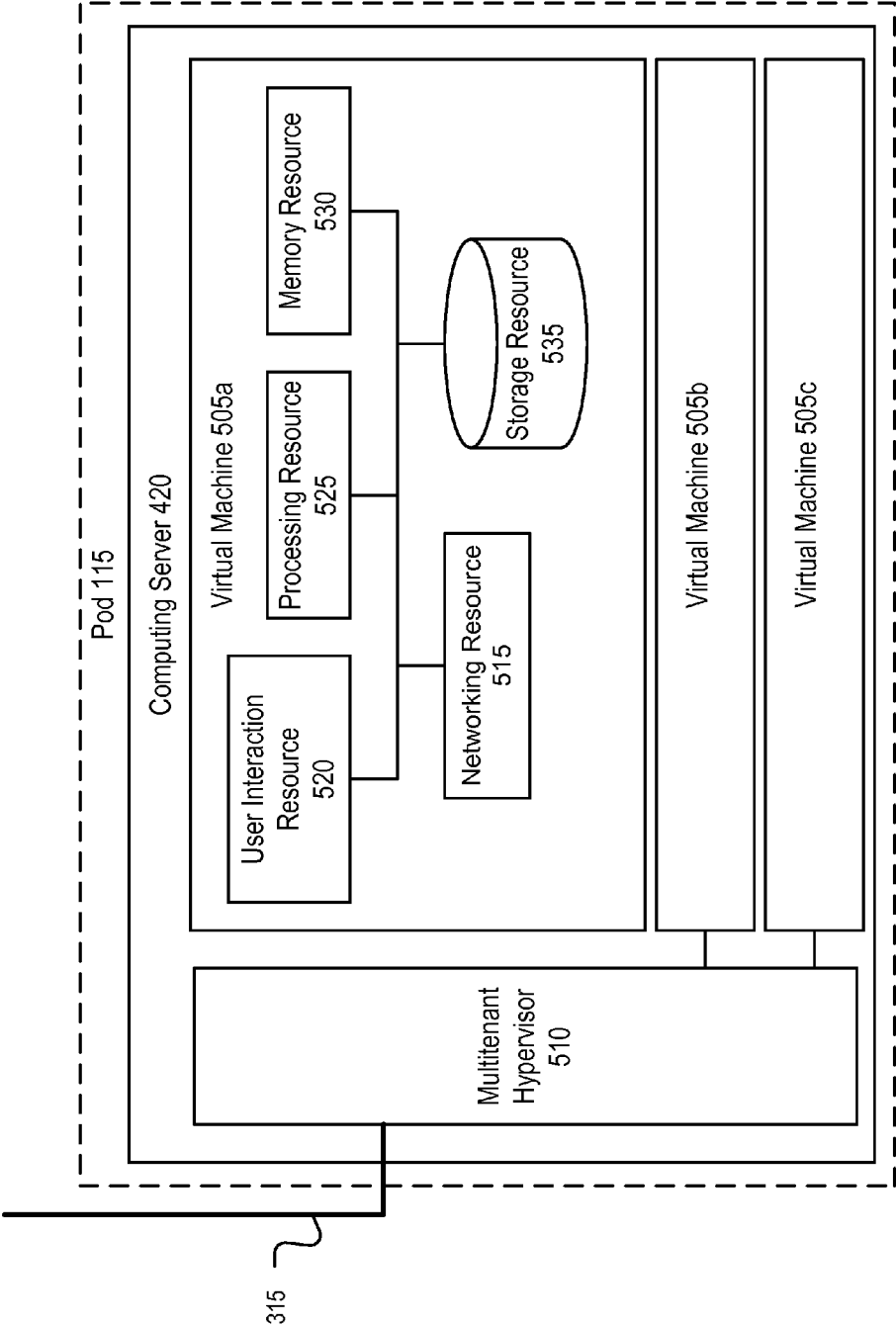


FIG. 5

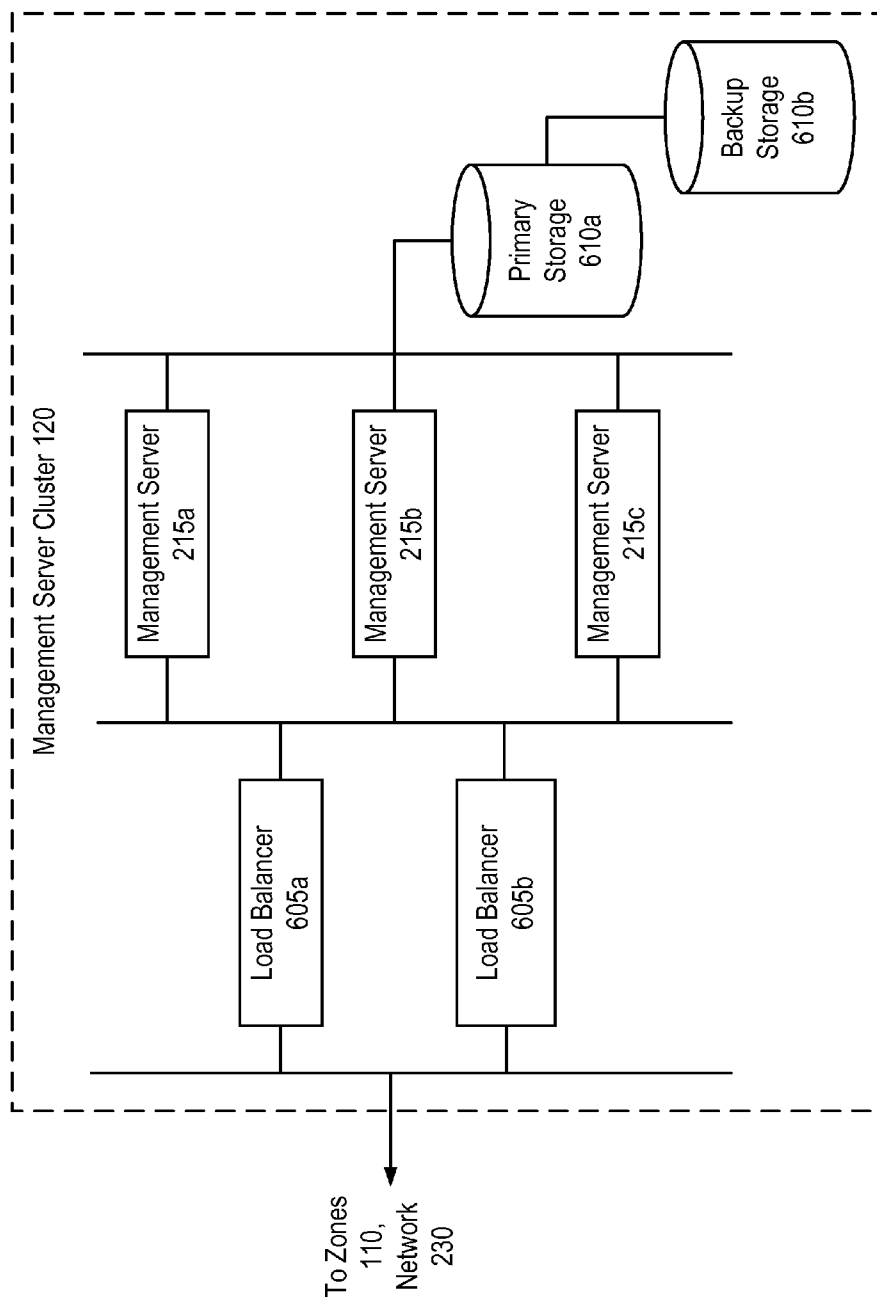


FIG. 6

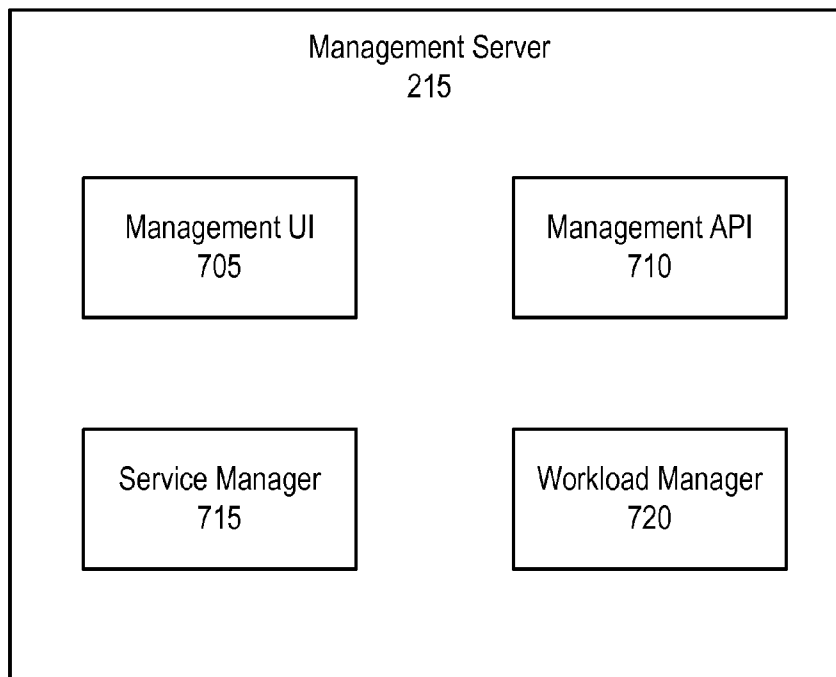


FIG. 7

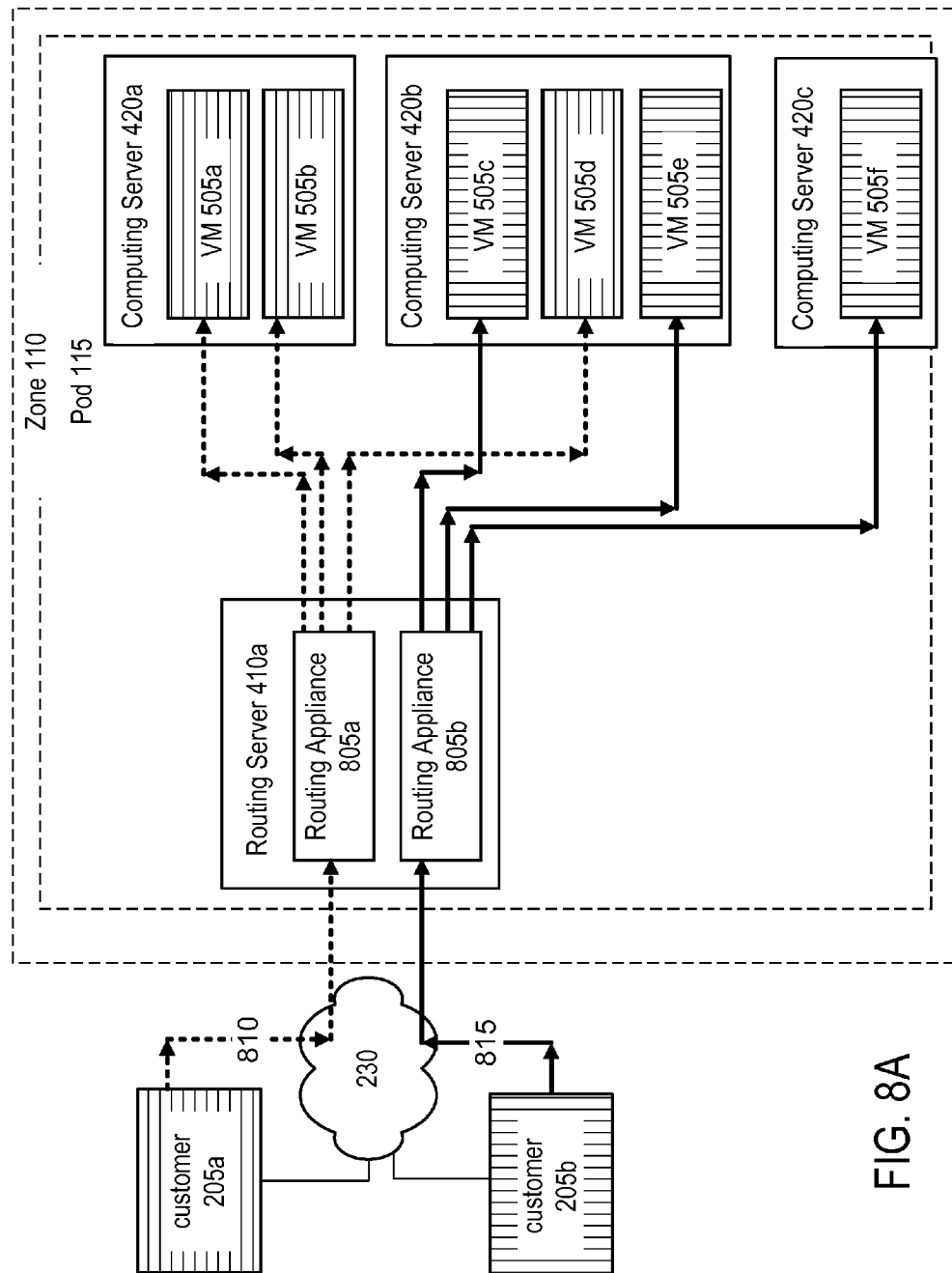


FIG. 8A

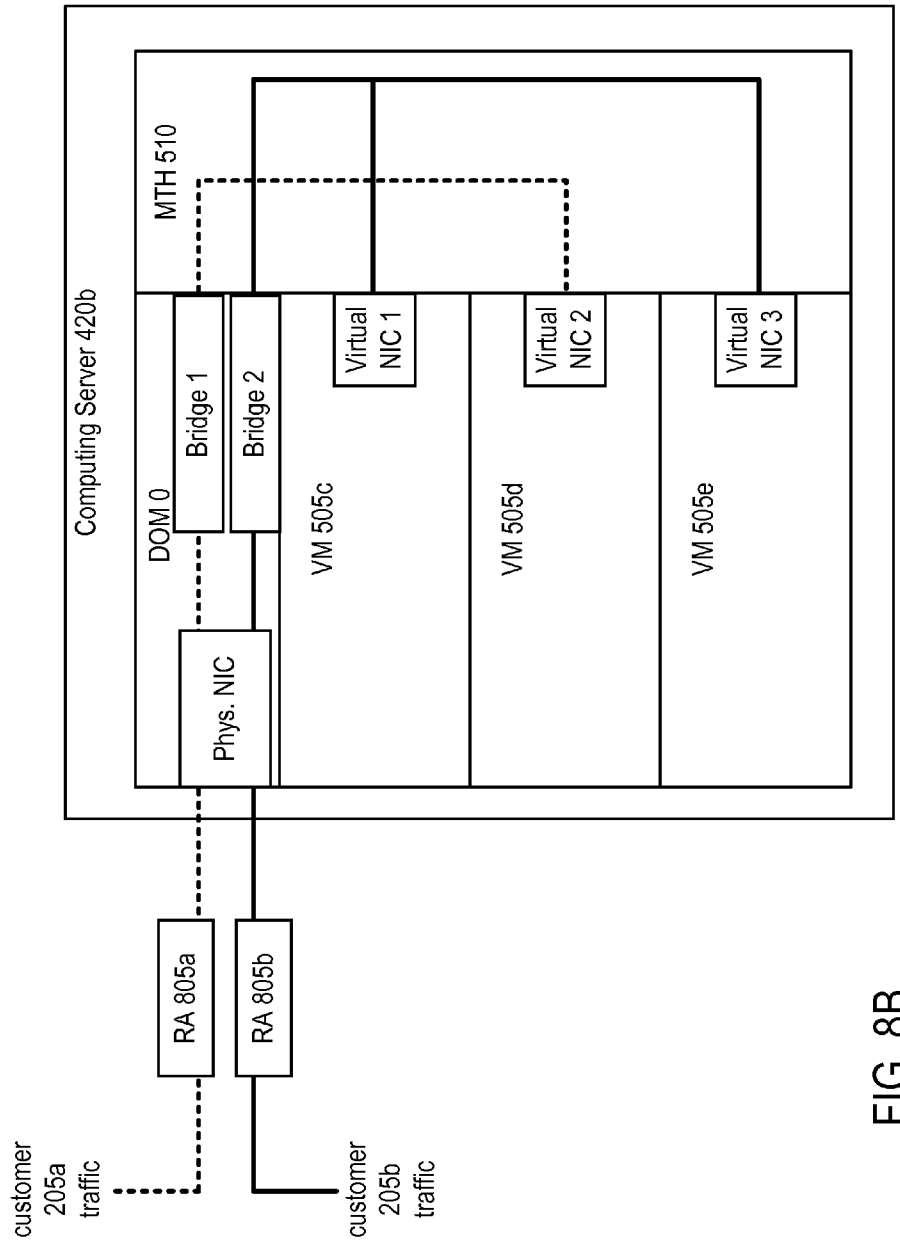


FIG. 8B

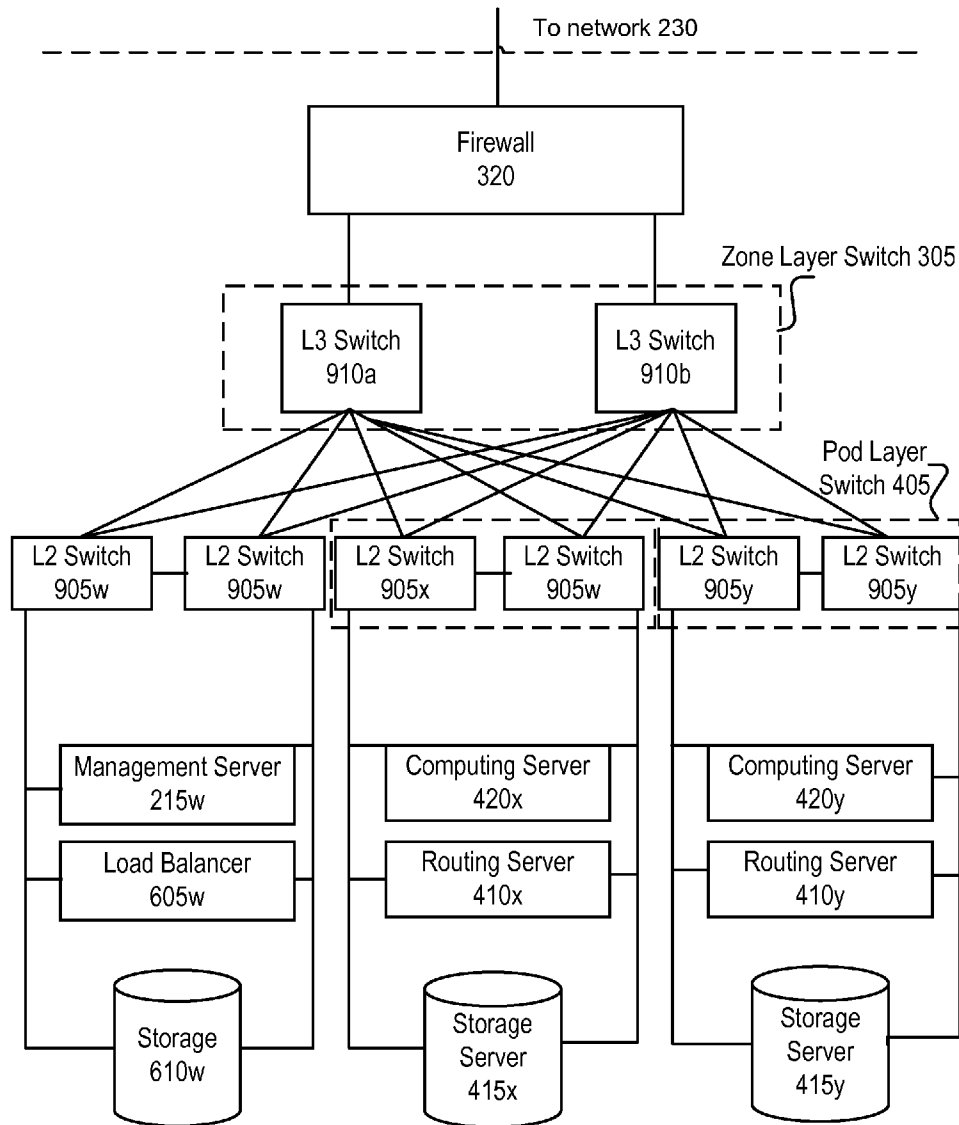


FIG. 9

SCALABLE CLOUD COMPUTING

CROSS REFERENCE TO OTHER
APPLICATIONS

This application claims the benefit, under 35 U.S.C. §119 (e), of U.S. Patent Application No. 61/301,168, filed on Feb. 3, 2010, and incorporated by reference herein.

BACKGROUND

This disclosure generally relates to cloud computing, and, more particularly, to enabling infrastructure information technology services, including computing, storage, and networking services to be provisioned on demand and delivered over the Internet in a scalable manner.

Service providers and enterprises have traditionally relied on large local installations of commodity data center hardware, including computing, storage, and networking devices, to provide information technology services and applications to their users. The advent of computing services that enable applications to run in “the cloud,” or on remote data centers that provision computing, storage, and networking services to applications, has left many service providers and enterprises with a large inventory of unused commodity data center hardware.

A cloud infrastructure provides on-demand computing resources to a customer (whether an individual or an enterprise) of the cloud operator through virtualization. The customer typically is physically remote from the computing resource and is agnostic to the location of the physical resources that support the computing resources. In a virtualized cloud infrastructure, the computing resource generally comprises a virtual machine characterized by some amount of processor, memory, storage, networking capability or capacity. Virtualization allows the physical resources support a large number of computing resources, often well beyond the limited number of actual physical devices. Physical resources in the cloud infrastructure are shared amongst the different customers of the cloud infrastructure. Each customer gets the illusion of operating a physically distinct computing resource.

Traditional virtualization infrastructure is built on shared storage and shared Layer-2 (Data Link) networking. These requirements severely limit scalability of conventional cloud systems. In shared storage, the physical disks may be physically separate from the computing server. These disks are typically controlled by a dedicated computer known as a storage controller or storage server. The storage controller provides access to the physical server via network protocols such as NFS and iSCSI. The virtual machines therefore access their storage over the network, but in a transparent fashion such that their storage appears to be locally attached. Each storage server provides storage to multiple physical servers. The virtual machines access their virtual disks over the network via a hypervisor deployed on the physical servers hosting the virtual machines. The hypervisor is responsible for managing the virtual machines’ access to the storage servers.

When the storage is networked in such a fashion, it may provide many advantages to the cloud operator. However, a typical infrastructure cloud is characterized by massive scale with hundreds or thousands of customers operating thousands of virtual machines simultaneously, with each customer getting the illusion of operating physically distinct computers. To support such scale, the operator needs to deploy hundreds

of physical servers and the networking elements and storage to support these physical servers.

While advantageous as outlined above, commercially available storage servers are not the ideal solution. The storage servers may not scale sufficiently to support such deployments due to architectural limitations. They may be prohibitively expensive or represent more capital outlay than warranted by the initial anticipated demand for the service. They may present single points of failure or increased cost due to deployment of redundant elements. Insurmountable performance bottlenecks may be present, for example, due to the limits of networking speed. Expensive large centralized storage may require long-term technology and vendor lock-in detrimental to the competitiveness of the cloud operator.

The networking elements may provide a similar challenge in large scale cloud deployments. Typically the network between the physical servers is provided by switched Ethernet since it provides performance at optimal price points. However, interconnecting all physical servers using Layer-2 switching has a number of drawbacks.

First, each physical server uses broadcasts and multicasts to discover services and advertise services on the network. As the number of physical servers increases to accommodate a growing number of virtual machines, the amount of broadcast traffic scales accordingly. Broadcast traffic is detrimental to the performance of the network since each server is interrupted by every broadcast even if it is not relevant to the server. Commercially available network switches can often only support a few dozen physical ports—each physical server is connected to one or more ports. Switches can be linked together with high speed switches but at great expense and potentially lower reliability.

Additionally, previous virtualization technologies resorted to one of two approaches: physical host-based network virtualization using software drivers integrated in the hypervisor or physical network/VLAN-based network virtualization, either via port-based VLANs or IEEE 802.1q tagged Ethernet frames. The popular IEEE 802.1Q standard defines a 12-bit tag, which allows more than 4000 VLANs to be supported within a broadcast domain. But neither of these approaches by themselves are sufficient to build a scalable cloud infrastructure.

SUMMARY

In a cloud infrastructure, physical resources are partitioned into pods within one or more zones in a scalable manner. The physical resources comprise physical compute, storage, and networking resources within data centers distributed across a network. Each zone comprises a subset of the pods and is physically isolated from other zones in the plurality. Each pod comprises a discrete set of physical resources in a zone, which resources are tightly connected via a communications network. The physical resources across pods are weakly connected via a communications network, in contrast to the physical resources within pods. Additionally, the cloud infrastructure establishes one or more management server clusters each comprising one or more management servers. In one embodiment, resources are strongly connected by physical and data link level protocols, and weakly connected by network or higher level protocols. In another embodiment, resources are strongly connected by having relatively low latency and/or high bandwidth network links between them, and weakly connected by having high latency and/or low bandwidth network link between them.

The cloud infrastructure serves two or more customers with authenticated accounts. Each customer is associated

3

with units of virtual resources on the cloud infrastructure. The cloud infrastructure establishes units of virtual resources by apportioning selected sets of the physical resources within the pods. The apportioned physical resources may be shared between two or more of the units of virtual resources. Each management server is configured for allocating the units of virtual resources to an account associated with each customer.

The cloud infrastructure comprises one or more data networks built from the distributed networking resources. A data network connects the pods and is configured for routing traffic to the pods from the customers of the cloud infrastructure and vice versa. The cloud infrastructure also comprises establishing one or more management networks. A management network connects the management servers to one of the zones, and connects the compute and storage resources partitioned within the pod.

Embodiments of the invention provide for a method to segment and isolate traffic from two or more customer accounts within the scalable cloud infrastructure. To segment traffic, VLANs may be employed by allocating a VLAN ID to a customer account at the time a virtual machine is created. To isolate virtual machines associated with two different customer accounts and hosted on the same physical resource, the cloud infrastructure may specify limits on the network bandwidth and storage bandwidth used by the virtual machines.

The features and advantages described in this summary and the following detailed description are not all-inclusive. Many additional features and advantages will be apparent to one of ordinary skill in the art in view of the drawings, specification, and claims hereof.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGS. 1A-1C provide one embodiment of a collective conceptual framework for a scalable cloud infrastructure.

FIG. 2 is one embodiment of a high-level entity diagram for a scalable cloud infrastructure.

FIG. 3 depicts one embodiment of a zone 110.

FIG. 4 illustrates one embodiment of a pod 115.

FIG. 5 depicts one embodiment of a computing server 420.

FIG. 6 shows one embodiment of a management server cluster 120.

FIG. 7 is a block diagram illustrating one embodiment of a management server 215 architecture.

FIG. 8A is a conceptual diagram showing one embodiment of an overlay of the virtual machines on the physical devices.

FIG. 8B shows additional detail for one embodiment of traffic routed between the different virtual machines 505c-505e within computing server 420b, when the traffic is to/from two different customers 205a, 205b.

FIG. 9 illustrates switching at different layers in the network architecture of the scalable cloud infrastructure in accordance with one embodiment.

The figures and the following description relate to preferred embodiments of the present invention by way of illustration only. It should be noted that from the following discussion, alternative embodiments of the structures and methods disclosed herein will be readily recognized as viable alternatives that may be employed without departing from the principles of the claimed invention.

DETAILED DESCRIPTION

Reference will now be made in detail to several embodiments of the present invention(s), examples of which are illustrated in the accompanying figures. It is noted that wher-

4

ever practicable similar or like reference numbers may be used in the figures and may indicate similar or like functionality.

FIG. 1A shows three separate data centers 105, each housing a number of servers S. A server S is a computer system based upon any type of hardware, including for example low-cost commodity data center hardware: x86 servers connected together using gigabit Ethernet. Service providers and enterprises have traditionally relied upon large local installations of this type of commodity data center hardware, including computing, storage, and networking devices, to provide information technology services and applications to their customers. Currently there is a large inventory of unused data center hardware in the market.

The present invention improves the organization and utilization of data center hardware to provide a scalable cloud infrastructure, by building a system of small blocks within data centers, and connecting the physical servers across data centers using high speed networks. FIGS. 1B-1C illustrate the overall framework for the scalable cloud infrastructure.

FIG. 1B shows several data centers ("DC") 105 and servers S partitioned into groupings of zones 110, pods 115 and management server clusters (MSC) 120. Each zone 110 is physically isolated from each other zone 110. Typically each data center 105 includes only one zone 110, however, a data center 105 can include multiple zones 110, e.g., as shown in data center 105b. When two zones such as 110a, 110b are not geographically distributed, they may still be physically isolated, e.g., via use of a separate power supplies, network uplinks, and other supporting infrastructure. Each zone 110 includes a grouping of pods 115.

Each pod 115 is a self-contained, physical grouping of servers that acts as a management unit for the scalable cloud infrastructure. Each pod 115 includes a discrete (i.e., non-overlapping) set of the physical resources in the zone 110. The physical resources in each pod have strong connectivity with each other and weak connectivity with the physical resources outside the pod. The strong connectivity and weak connectivity refer to the latency for traffic between connected physical resources. The latency for traffic may be determined by many different factors.

In one embodiment, strong connectivity and weak connectivity may be switching of traffic between any two connected physical resources on a particular layer of a standard, for instance the OSI model. As an example, strong connectivity may imply Layer-2 switching. Similarly, weak connectivity may imply layer-3 or higher layer switching. In another embodiment, strong connectivity and weak connectivity may be based on the bandwidth available for traffic between the connected physical resources. As an example, strong connectivity may be implemented by provisioning a minimum of 10 GHz links between the connected physical resources. Similarly, weak connectivity may be implemented by a minimum of 1 GHz links between the connected physical resources. Additionally, geographic proximity may also be used to define strong connectivity and weak connectivity. It is possible that geographically distant physical resources have a higher latency for traffic than geographically closer physical resources.

The discrete set of physical resources in a pod 115 may be based on the anticipated processor, memory, network, and storage requirements of potential customers of resources within a zone 110. For example, a customer's storage and network requirements can be significant. Given a specification of resources, e.g., an average and peak throughput in terms of input-output operations per second (IOPS), and assuming that that throughput is to be divided equally

5

amongst the devices (e.g., virtual machines) in a pod **115**, then the IOPS capacity of the servers determines an overall total number of virtual machines for a pod **115**. If each server within the pod **115** can host a specified number of virtual machines, then a pod **115** could be sized accordingly in terms of the number of servers, storage, and networking requirements.

The pods **115** can be coupled with any number of other pods **115** using Layer-3 switching, thereby enabling unlimited scaling of the number of customers using the scalable cloud infrastructure. The pods **115** allow the scalable cloud infrastructure to be built from smaller units of management, and without a large up-front hardware investment.

FIG. 1B also shows one or more management server clusters (MSC) **120** to the data centers **105** housing the zones **110**. The management server cluster **120** is a cluster of front-end servers **S** and their associated backend storage. The servers that make up the management server cluster **120** allocate and manage use of the physical resources in the associated zones **110** by one or more customer accounts associated with units of virtual resources as shown in FIG. 1C and further described below. A virtual machine is characterized by a combination of the units of virtual resources for processor, memory, storage and networking. In brief, units of virtual resources are established by apportioning selected physical resources within the pods **115** and zones **110**; the physical resources may be shared between units of virtual resources (i.e., may overlap).

Typically the management server cluster **120** is deployed as a primary management server cluster **120a** in a first data-center **105a**, with a back up management server cluster **120b** installation at a second datacenter **105c**.

In one embodiment, data stored by the first management cluster **120a** is replicated and transmitted for storage by the second management cluster **120b**. In response to a failure of the first management cluster **120a**, the domain name system (DNS) records are updated in the servers in the zones **110** such that network traffic that would normally be directed to the first management server cluster **120a** instead is directed to the second management server cluster **120b**. Thus, operation of the servers in the zones **110** is unaffected by failure of the first management server cluster **120a**. The two management server clusters **120** are isolated from one another by residing in different data centers **105a**, **105c** to minimize the likelihood of simultaneous failure of the management server clusters **120**.

It is within the architectural framework of FIGS. 1B-1C that the scalable cloud infrastructure is further described herein.

FIG. 2 is a high-level entity diagram for one embodiment of a scalable cloud infrastructure. The entities included are one or more customers **205**, an administrator **210**, one or more management server clusters **120** and servers **240**, which are physical resources within zones **110** in data centers **105**, connected to a network **230**. The data center **105** are partitioned into a plurality of zones **110**, and each zone contains one or more pods **115**, as described above. Each zone **110** is connected to a management server **215** in the management server cluster **120** and to the network **230**. The network **230** is the Internet according to one embodiment, but may be any other network, including but not limited to any combination of a LAN, a WAN, a MAN, a mobile, a wired or wireless network, a private network, or a virtual private network, where Layer 2 and Layer 3 switching is required.

The administrator **210**, or operator of the cloud infrastructure, deploys the various resources as discussed above. In one embodiment, the operator of the cloud infrastructure is distinct from one or more providers of the resources, e.g., a

6

provider may own any number of virtual machines that will be used by the administrator **210** in the cloud infrastructure. In one embodiment, the provider owns a large number of servers **240**.

One or more customers **205** are users of portions of the resources of the cloud infrastructure. The term customer may refer to either the device or the user (entity or person). Customers **205** access the resources associated with their respective customer accounts and are connected to the resources via the network **230**. Details of customer **205** account resource allocation and access to those resources are discussed further in conjunction with FIGS. 8 and 9.

The management server cluster **120** is a cluster of management servers **215** and their associated database **220**. As indicated in conjunction with FIG. 1C, the management server cluster **120** serves multiple zones **110** and pods **115** within data centers **105**. The management server cluster **120** also maintains customer account information. For example, the management server cluster **120** may establish virtual machines, units of virtual resources, by apportioning the physical resources within the pods to each of the units. The physical resources may be shared between two or more of the virtual machines. Although one management server cluster **120** is depicted, multiple management server clusters **120** may be distributed throughout the cloud infrastructure, each serving a plurality of data centers **105** and zones **110**. The details of the management server cluster **120** are described further in conjunction with FIG. 6.

Each management server **215** allocates and manages use of the physical resources within one or more zones **110**. For example, management server **215a** manages the resources in zone **110a**, management server **215b** manages the resources in zone **110b**, and management server **215c** manages the resources in zone **110c**. A management network may connect each of the management servers **215** to a zone **110**, and may also connect the physical resources within the pod **115**. A management server **215** can allocate to, and manage, units of virtual resources associating customer accounts with the physical resources in a zone **110** associated with that management server **215**. The details of the management server **215** are described further in conjunction with FIG. 7. The database **220** can be any database commonly used for storage. According to one embodiment, the database **220** is a MySQL database.

The data centers **105**, zones **110**, and pods **115** were described briefly in conjunction with FIG. 1B-1C. Switching within the zones **110** typically is Layer 3 switching, while switching within the pods **115** typically is Layer 2 switching, providing unlimited scaling of the cloud infrastructure. Zones **110** are described in greater detail in conjunction with FIG. 3, pods **115** are described in greater detail in conjunction with FIG. 4, and the servers in each pod **115** are described further in conjunction with FIG. 5.

FIG. 3 depicts one embodiment of a zone **110**. As discussed above, a zone **110** is a grouping of physical devices, including a grouping of pods **115**, that is physically isolated from other zones **110**. Devices within a zone **110** utilize a one or more power supplies and/or network uplink(s) that are dedicated solely to the zone **110**. The zone **110** includes a zone layer switch **305** that process traffic in and out of the zone **110** over a data network **310** and a management network **315**. The zone **110** is separated from other devices on the network **230** by a firewall **320**.

The zone layer switch **305** manages the network traffic to and from the pods **115** within the zone **110** and comprises one or more Layer 3 ("L3") (i.e., network layer) switches. For a zone **110** comprising the servers in a small data center **105**, a

pair of L3 switches may suffice. For large data centers, high-end core switches may implement core routing capabilities and include line cards that support firewall capabilities. A router redundancy protocol like VRRP also may be deployed. Traffic within the zone **110** (and/or pods **115**) also is switched at other levels of either the OSI model or other networking models, as described further below.

One embodiment of the zone layer switch **305** supports two internal networks within the zone **110**: a data network **310** and a management network **315**. The data network **310** is used to carry traffic to and from the zone **110** from customers **205** and other entities on the network **230**. For example, operation of virtual machines implemented on the servers within the pods **115** of the zone **110**, e.g., if a virtual machine within the zone **110** issues an HTTP request over the network **230**, both the HTTP request and any associated HTTP response will be routed via the data network **310**.

The management network **315** is used to carry traffic to and from the zone **110** from the management server cluster **120** (and individual management servers **215** within the management server cluster **120**), as well as traffic generated internally within the zone **110**. Each pod **115** in the zone **110** is communicatively coupled to both the data network **310** and the management network **315**.

All traffic in and out of the zone **110** passes through a firewall **320**. The firewall **320** may comprise one or more network firewalls specific to each internal network **310**, **315**. The firewall **320** provides connection to the public network space (i.e., network **230**) and is configured in routing mode.

In one embodiment, the firewall **320** operates in a transparent mode for the data network **310** such that the data network **310** comprises the same IP address space as the public Internet. For example, if the data network **310** utilizes public IP addresses, the zone **110** is assigned a unique set of IP addresses that do not overlap with the IP addresses assigned to other zones **110**.

In one embodiment, the firewall **320** operates in a network address translation (NAT) mode for the management network **315**. For example, the zone **110** can be assigned an IP address in the 192.168.0.0/16 Class B private address space, and each pod **115** within the zone **110** can be assigned an IP address in the 192.168.*.0/24 Class C private address space; the firewall **320** remaps between the two address spaces as data is sent to or from the pods **115** within the zone **110**. Hence, it is possible for the pods **115** of different zones **110** to have overlapping or identical private IP addresses. In some embodiments, the firewall **320** is outside of the zone **110** such that it filters traffic to both the zone **110** and the management server cluster **120**. In some embodiments, the firewall **320** enables site-to-site VPN such that servers in different zones **110** can reach each other within a virtual private network.

FIG. 4 illustrates one embodiment of a pod **115**. The pod **115**, as described above comprises a discrete set of the plurality of physical resources in the zone **110**. The servers in a pod **115** typically comprise one or more each of routing servers **410**, storage servers **415**, and computing servers **420**. In one embodiment, distinct routing servers **410** and computing servers **420** are used, whereas in another embodiment (not shown), they are one and the same and provide both functionalities described below. Traffic on the management network **310** and data network **315** in and out of the pod **115** is switched by one or more pod layer switches **405**. The servers in a pod, both individually and collectively are implemented by one or more computer systems.

The routing servers **410** are configured primarily to provide networking for computer data by the inclusion of suitable networking hardware (network interfaces, networking ports,

and the like). The routing servers **410** can be implemented using any manner of suitable networking hardware and software, and in some instances are combined with computing servers **420**.

The storage servers **415** are implemented as any device or combination of devices capable of persistently storing data in non-transitory computer-readable storage media, such as a hard disk drive, RAM, a writable compact disk (CD) or DVD, a solid-state memory device, or other optical/magnetic storage mediums. Other types of computer-readable storage mediums can be used, and it is expected that as new storage mediums are developed in the future, they can be configured in accordance with the teachings here. In addition, the storage servers **415** support local or distributed databases for storing customer information; in one embodiment the database are MySQL databases.

Typically single uniform storage servers do not scale to more than a few dozen servers. A system architecture using pods **115**, however, allows multiple smaller storage servers **415** to be associated with each pod **115**. Pod-level shared storage delivers the benefit of shared storage such as the ability to restart virtual machines on a server different than where the virtual machine last ran, which provides the ability to start up, shut down, and reallocate available servers.

The computing servers **420** host the virtual machines within the pod **115** as will be discussed further in conjunction with FIG. 5. The computing servers **420** may comprise computing, routing, and/or storage servers having different processing, memory, storage, and networking capabilities according to various embodiments, and may perform all of the functions of routing servers **410** in embodiments that exclude separate routing servers **410**. Additionally, the computing servers **420** can utilize different data storage systems, such as direct-attached storage (DAS), network-attached storage (NAS), or a storage area network (SAN).

The pod layer switch **405** switches network traffic into and out of the pod **115**. The pod layer switch **405** may comprise one or more pod layer switches. The pod layer switch **405** typically is a Layer 2 switch, but switching is also possible at other levels of either the OSI model or other networking models. In alternate embodiments, the pod **115** may implement internal networks in addition to or distinct from the data network **310** and the management network **315**.

In one embodiment, a public switch may be used for public traffic (e.g. traffic on the data network **310**) and a private switch for management traffic (e.g. traffic on the management network **315**). Storage traffic (e.g., traffic between the computing servers **420** and the storage servers **415** via the private switch) may be isolated from other traffic to avoid packet loss and delays that are common with regular TCP/IP traffic, and to protect the servers **240** from potentially malicious Internet traffic. In addition, the storage traffic may be directed over a higher speed switch to meet the higher performance demands of the storage system.

In one embodiment of the pod **115**, the pod layer switch(es) **405** are duplicated for redundancy, with each computing server **420** connected to multiple switches. Further, it should be noted that multiple layers of switches can be coupled to effectively form a pod layer switch **405** with an increased number of ports.

In another embodiment of the pod **115** design, Virtual LANs (VLANs) are used to segment traffic while using a single pod layer switch **405**. The pod layer switch **405** may support quality of service guarantees for virtual machines within the pod; the storage VLAN is then given the appropriate share of the available bandwidth on the switch to meet the IOPS requirements of such virtual machines.

The VLANs for the management, public and storage traffic may be created before the pod is deployed into production, by programming the pod layer switch 405 using the usual management interfaces. The storage server 415 is also configured to use the storage and management VLANs using the usual management interfaces of the storage server 415. In other embodiments, more than one traffic type can be directed over a single switch while other traffic types can have their own separate switches. For example, guest traffic, which refers of traffic between the virtual machines of each customer, may be segmented using VLANs and directed on the same switch as management traffic.

FIG. 5 depicts one embodiment of a computing server 420. The computing server 420 hosts one or more virtual machines 505 and a multitenant hypervisor 510. The computing server 420 is communicatively coupled to at least the management network 315, and may also be coupled to the data network 310 in embodiments without separate routing servers 410. In other embodiments, the virtual machines 505 and the multitenant hypervisor 510 each are individually coupled to the management network 315.

In one embodiment, the multitenant hypervisor 510 is implemented on the computing server 420 as a set of computer-executable instructions encoded onto a non-transitory computer-readable storage medium included in the computing server 420 and executed by a processor included in the computing server 420. The multitenant hypervisor 510 therefore can comprise, for example, a software layer that manages the physical computing elements (e.g., processors, memory, network cards and interfaces, data storage disks) of the computing server 420. The virtual machines 505 access and use these physical computing elements as dictated by the multitenant hypervisor 510. The multitenant hypervisor 510 can alter the allocation and accessibility of computing elements within the computing server 420 over time in response to changes in the number and configurations of hosted virtual machines 505. The changes in the number and configurations of hosted virtual machines 505 may occur, for example, because the customers 205 associated with the hosted virtual machines 505 made such a request or changes have occurred at other virtual machines 505 associated with the customer 205.

A number of virtual machines 505 may run on the computing server 420. The virtual machine 505 comprises an allocation of the computer hardware and computer software elements of the computing server 420. The virtual machine 505 simulates a physical computing device and can be configured to perform any computing task via inclusion of appropriate computer hardware and/or computer software elements therein. For example, a virtual machine 505 can simulate a physical computing device having a 1 GHz processor, 1 GB of memory, and a 16 GB hard drive.

A virtual machine 505 is associated exclusively with a single customer 205. However, a computing server 420 may host a set of virtual machines 505, each associated with different customers 205. For example, in FIG. 5, one customer 205 may be associated with virtual machine 505a, where as another customer may be associated with virtual machine 505b and 505c.

A single computing server 420 simultaneously can host virtual machines 505 associated with different customers 205, but the multitenant hypervisor 510, along with the associated management server 215, manages the hosted virtual machines 505 such that each virtual machine 505 appears to the corresponding customer 205 as a physically distinct and self-contained computing device.

In one embodiment, the virtual machine 505 comprises a processing resource 525, a memory resource 530, a storage resource 535, a networking resource 515, and a user interaction resource 520. The resources 515-535 of the virtual machine 505 comprise allocations of the computer hardware and/or computer software elements of the computing server 420, according to the units of virtual resources designated for a given customer account. The processing resource 525 comprises an allocation portion of one or more computer processors. The memory resource 530 and the storage resource 535 can comprise an allocation of any physical device or combination of physical devices capable of persistently storing computer data, such as a hard disk drive, random-access memory (RAM), a writable compact disk (CD) or DVD, a solid-state memory device, or other optical/magnetic storage mediums. Other types of non-transitory computer readable storage mediums can be used for the memory resource 530 and/or the storage resource 535, and it is expected that as new storage mediums are developed in the future, they can be configured in accordance with the teachings here. In one embodiment, the memory resource 530 provides operational memory and comprises a specified amount of RAM. The storage resource 535 provides long-term data storage and, again, can comprise any suitable type of non-transitory computer readable storage medium such as one or more hard disk drives.

The networking resource 515 comprises an allocation of computer networking hardware and software elements to enable the virtual machine 505 to communicate with other networked entities over the management network 315. One or more IP addresses can be associated with the virtual machine 505 and supported by the networking resource 515. Accordingly, the networking resource 515 can include any type of communication interface suitable for transmitting and receiving data over the network 315. For example, the networking resource 515 can comprise an Internet interface, a serial interface, a parallel interface, a USB (Universal Serial Bus) interface, an Ethernet interface, a T1 interface, a Bluetooth interface, IEEE 802.11 interface, IEEE 802.16 interface, or any other type of wired or wireless communication interface.

The user interaction resource 530 comprises hardware and software elements to enable the customer 205 or administrator 210 to interact with the virtual machine 505. For example, the user interaction resource 520 can provide display elements, such as a graphical user interface (GUI) whereby either the customer 205 or administrator 210 can interact with and manage operation of the virtual machine 505. The user interaction resource 530 can also support a keyboard, and mouse, or the like to further enable the customer 205 or administrator 210 to manage operation of the virtual machine 505.

In some embodiments, the resources 515-535 of the virtual machine 505 are supported within a single computing device, such as a server. In other embodiments, portions of the virtual machine 505 can be distributed among any number of physically separate computing devices that are communicatively coupled via the network 315. For example, the storage resource 535 can comprise multiple hard-drives residing on multiple servers. Because the resources 515-535 are communicatively coupled to one another, the virtual machine 505 appears to the customer 205 to be a single computing device that includes the cumulative capabilities of the resources 515-535 regardless of the actual physical location or distribution of any computer hardware or software elements associated with of the resources 515-535.

In some embodiments, there is a specially designated virtual machine called the management domain that provides a

11

standard set of commands to control the multitenant hypervisor **510**, for example, to start and stop virtual machines, and to control the networking stack of the multitenant hypervisor **510**. In other embodiments, the multitenant hypervisor **510** is hosted by a host operating system and the virtual machines operate as processes of the operating system with the multitenant hypervisor **510** providing isolation. The multitenant hypervisor **510** ensures that the virtual machines share the physical resources of the compute host such as the processor, memory, network interfaces and storage. In some cases, the multitenant hypervisor **510** defers the operation and management of the network stack and storage interfaces to the management domain or host operating system.

The physical network interfaces are shared among the virtual machines by the multitenant hypervisor **510**. In one embodiment, the virtual machines get the illusion of possessing a standard physical network interface such as those provided by commercially available network interface cards. The multitenant hypervisor **510** ensures that these virtual interfaces are connected to the underlying physical network interfaces of the compute host.

The cloud infrastructure comprises a management server **215** interacting with agents that in turn interact with and control the multitenant hypervisor **510** using its standard set of commands. In one embodiment there may be one agent running in each operating system or management domain on a server **240**. In other embodiments one agent may interact with a group of servers **240** whose multitenant hypervisors have been clustered using cluster management software. The agents are controlled by the management server **215**.

The management server **215** also interacts with the storage servers **415** in order to create and destroy the virtual disks for the virtual machines. In one embodiment, a special version of the agent known as the storage agent runs on the processor subsystem of the storage server **415** to perform these activities. In another embodiment, the management server **215** uses the standard set of commands provided by the management server **215**, or its Application Programming Interface (API) to create and destroy virtual disks.

The storage server **415** presents virtual disks to the computing server **420**. In one embodiment, the virtual disks are visible as networked file systems to the multitenant hypervisor **510**. In another embodiment the virtual disks are presented as block devices to the multitenant hypervisor **510**. The multitenant hypervisor **510** ensures that these virtual disks are presented to the virtual machine while giving the illusion of locally attached storage to the virtual machines.

The multitenant hypervisor **510** provides a standard set of commands that the agent uses. Some examples of the command set are: start a virtual machine, stop a virtual machine, reboot a virtual machine, add or remove a virtual disk for a virtual machine, add or remove a virtual network interface for a virtual machine, mount/dismount a virtual disk from the storage server **415**, add or remove VLANs from the physical network interfaces of the server **240**.

The agents collect information from the computing servers **420** and storage servers **415** and report to the management server **215**. The management server **215** maintains the reported information in database tables. The database includes for example: the state of the multitenant hypervisor **510**, the state of the virtual machines, the configuration of the networking stack, such as configured VLANs (explained subsequently in the description of FIG. **8**), IP addresses, speeds and aggregations of the physical network interfaces, storage resources visible from the compute server **420**, the capabilities of the multitenant hypervisor **510**, the capabilities of the storage server **415**, the capacity, used and allocated

12

size of the storage server **415**, statistics such as network traffic consumed by the virtual machine, processor and memory usage of virtual machine.

The management server commands the agent to perform certain actions in response to actions by the cloud customer or cloud operator at the user interface or API. For example, when a customer starts a virtual machine, the management server **215** may look up its table of servers **240** and identify a server **240** that has enough spare capacity to satisfy the requested processing, memory and network resources for the virtual machine. The management server **215** may also look up its table of virtual machines to verify if the virtual machine is already running or if its virtual disks already exist on a storage server **415**. If the virtual machine is not already running or its virtual disks do not exist, the management server **215** may command the storage server **415** or storage agent to create the disks.

The management agent then sends a start virtual machine instruction to the agent for the compute host chosen in step 1. In the start instruction are included information such as: the portion of the physical processors to allocate to the virtual machine, the virtual network interfaces to create for the virtual machine and their mapping to the physical interfaces and identification of the virtual disks to present to the virtual machine.

The agent may then instructs the multitenant hypervisor **510** on the computing server **240** (or computing server **420** in the example of FIG. **5**) to mount the virtual disks from the storage server **415**, create the virtual network interfaces and start the virtual machine with the desired processors, memory, network interfaces and disk. The agent verifies that the virtual machine has started using the hypervisor command set and reports the success to the management server **215**. The management server **215** updates its table of virtual machines with information about the virtual machine including the server **240** it is running on.

The management network **315** handles traffic associated with the multitenant hypervisor **510** executed by the computing servers **420** within the pods **115** of the zone **110**. Hence, traffic on the management network **315** may additionally comprise messages between a multitenant hypervisor **510** and a virtual machine **505** (e.g., messages related to the allocation of the physical computing elements of the particular computing server **420** to a particular virtual machine **505**).

In an embodiment of the invention, network and storage isolation may comprise isolation of data alone and not quality of service. In other embodiments, the quality of service capabilities built into the multitenant hypervisor's **510** as well as network and storage hardware may be used to support enhanced isolation.

Network bandwidth throttling is used to limit the rate of traffic from each virtual machine. The operator of the cloud infrastructure may configure the management server **215** with the desired maximum network bandwidth for each virtual machine. The configuration can be done on a per-service-offering basis, a per-customer account basis or across the cloud. A management server **215** includes this maximum in the start instructions sent down to the multitenant hypervisor **510** when starting a virtual machine. The multitenant hypervisor **510** uses a standard set of commands to limit the guest traffic from the new virtual machine.

Storage bandwidth throttling is used to limit the rate of storage traffic from each virtual machine while reading or writing to a virtual disk. The operator of the cloud infrastructure may configure the management server **215** with the desired maximum IOPs bandwidth for each virtual machine. As before, the configuration can be done on a per-service-

13

offering basis, a per-consumer basis or across the cloud. The management server includes this maximum in the start instructions send down to the multitenant hypervisor **510** when starting a virtual machine. The multitenant hypervisor **510** uses a standard set of commands to limit the storage traffic from the new virtual machine to a maximum number of IOPS as defined by the operator of the cloud infrastructure.

FIG. 6 shows one embodiment of a management server cluster **120**. As noted above, the management server cluster **120** is a cluster of front-end servers and their associated backend storage. The servers **215** that make up the management server cluster **120** allocate and manage use of the physical resources in the associated zones **110** by one or more customer accounts associated with units of virtual resources.

The depicted management server cluster **120** comprises three management servers **215**, two load balancers **205**, primary storage **610a**, and backup storage **610b**. Other embodiments of a management server cluster **120** may comprise different numbers of management servers **215** and load balancers **605**.

The management servers **215** are communicatively coupled to the load balancers **605**, and the load balancers **605** are communicatively coupled to the networks **310**, **315**. Thus, the management servers **215** can transmit and receive data and commands via the networks **310**, **315** through the load balancers **605**. The load balancers **605** distribute traffic from the networks **310**, **315** and associated workload among the management servers **215** to optimize utilization of the management servers **215**. In some embodiments, the load balancers **605** comprise dedicated hardware devices (e.g., multi-layer hardware switching devices or servers configured to provide load balancing functionality). In other embodiments, the load balancers **605** comprise load balancing software installed on the management servers **215**.

The primary storage **610a** and the backup storage **610b** can be implemented as any device or combination of devices capable of persistently storing data in non-transitory computer-readable storage media, such as a hard disk drive, RAM, a writable compact disk (CD) or DVD, a solid-state memory device, or other optical/magnetic storage mediums. Other types of computer-readable storage mediums can be used, and it is expected that as new storage mediums are developed in the future, they can be configured in accordance with the teachings here. In one embodiment, the primary storage **610a** and the backup storage **610b** further comprise MySQL databases. The primary storage **610a** and backup storage **610b** can also comprise one or more dedicated storage servers. The primary storage **610a** for the management server cluster **120** is communicatively coupled to the management servers **215** and provides data storage as required by the management servers **215**. The backup storage **610b** is communicatively coupled to the primary storage **610a** and comprises a replicated version of the data stored by the primary storage **610a**. The devices in the management server cluster **120** may be Layer 2 switched.

FIG. 7 is a block diagram illustrating one embodiment of a management server **215** architecture. The operations of the management server **215** as described herein can be controlled through either hardware (e.g., dedicated computing devices or daughter-boards in general purpose computers), or through computer programs installed in computer storage of the management server **215** and executed by one or more processors of the management server **215** to perform the functions described herein. One of skill in the art of system engineering will readily determine from the functional and algorithmic descriptions herein the construction and operation of such computer programs.

14

In one embodiment, the management server **215** comprises a management user interface (UI) **705**, a management application programming interface (API) **710**, a service manager **715**, and a workload manager **720**.

In one embodiment, the management UI **705** provides the primary user interface for customers **205** and administrators **210**. The management UI **705** can provide a graphical user interface (GUI) that is accessible over the networks **310**, **315** via a conventional web browser using any networked computing device. A customer can, for example, input specifications for configuring a new virtual machine **505** using a web GUI provided by the management UI **705**. More particularly, a customer configures a virtual machine by specifying the amount of processor, storage, memory and networking resources, in appropriate units (e.g., processor speed for processors, amount in mega- or giga-bytes for storage and memory, and throughput for networking).

A customer can also interact with a configured virtual machine **505** using a web GUI provided by the management UI **705** by, for example, inputting data for processing by the virtual machine **505**, viewing outputs computed by the virtual machine **505**, and inputting commands related to a complete or partial shutdown of the virtual machine **505**. The management UI **705** can provide a different web GUI for a customer **205** than for an administrator **210**.

One embodiment of the management API **710** allows an administrator **210** (or a customer **205** with appropriate access credentials) to further oversee virtual machine **505**. For example, the management API **710** can enable customization of the primary user interface provided by the management UI **705**. The management API **710** can also allow customization of billing policies and procedures, as well as access policies (e.g., granting different levels of access to a virtual machine **505** based on credentials associated with a customer **205**).

The service manager **715** communicates with computing servers **420** to oversee the creation, operation, and shutdown of virtual machines **505**. For example, the service manager **715** can receive specifications for a virtual machine **505** from the management UI **705**, select a computing server **420** suitable for hosting the virtual machine **505**, and transmit commands to the selected computing server **420** that cause the computing server **420** to implement the virtual machine **505**. Once the virtual machine **505** is configured and implemented on the computing server **420**, the service manager **715** can monitor its operation and implement corresponding billing and access policies, e.g., via routing appliance **805** discussed further in FIG. 8. For example, the service manager **715** can bill a customer \$20 per hour of operation for a virtual machine **505** with processing equivalent to a 1 GHz processor, memory equivalent to 1 GB of RAM, and storage equivalent to a 250 GB hard drive, as well as \$0.10 per GB of network traffic associated with the networking resource of the virtual machine **505**.

The workload manager **720** interacts with the multitenant hypervisors **510** installed on the computing servers **420**. The workload manager **720** monitors the status (e.g., availability and workload) of the different physical computing elements included in the computing servers **720**. The workload manager **720** can also oversee the transfer of a virtual machine **505** from a first computing server **420** to a second computing server **420** upon failure of the first computing server **420** or an imbalance of workload between computing servers **420**.

FIG. 8A is a block diagram showing one embodiment of an overlay of the virtual machines allocated to two customers **205a**, **205b** on the physical devices, e.g., virtual machines **505a-505f**. As shown, virtual resources such as virtual machines **505**, are allocated to customer accounts associated

15

with customers **205a** and **205b**. In this example, virtual machines **505a**, **505b**, and **505d** are allocated to customer **205a** and virtual machines **505c**, **505e**, and **505f** are allocated to customer **205b**. The virtual resources correspond to physical resources within various computing servers **420** within a pod **115** and zone **110**. Note that some of the physical resources are shared, at least in part, by customers **205a**, **205b**. For example, computing server **420b** is shared by both customers **205a**, **205b**.

In a cloud infrastructure serving multiple customers, the administrator **210** may ensure that accesses are secure and protected against attacks from malicious customers, by allocating resources on demand for instance. Routing appliances **805** run as virtual machines in the system to provide functionalities such as routing, DNS, load balancing, console access, etc. When they are no longer needed, they are garbage collected to release any precious system resources that they are holding. The routing appliances **805** may be hosted by the routing server **410**.

The eth0 interface of a routing appliance **805** serves as the gateway for the guest virtual network and has the IP address of 10.1.1.1, which is configurable. The eth1 interface of the routing appliance **805** resides on the management network **315** and is used to configure the routing appliance. The eth2 interface is assigned a public IP address on the data network **310**.

As indicated in FIG. 8A, VMs associated with customers **205a** and **205b** are segmented on the same physical network. In order for the VMs associated with a customer **205** to access the internet or to accept connections from the internet (such as ssh), a routing appliance may be started up for the VM. In the example of FIG. 8A, customer **205a** accesses virtual machines VM **505a**, **505b** and **505d** through routing appliance **805a** hosted on computing server **420a**, while customer **205b** accesses virtual machines VM **505c**, **505e** and **505f** through routing appliance **805b**. While physical resources such as network and physical server are shared, the networks of customers **205** are segmented and cannot see each other.

When a customer **205** starts a VM in a certain zone **110**, a management server **215** determines if a routing appliance **805** for that customer **205** is already running within that zone **110**. If it is not, the routing appliance **805** is started prior to the actual start of the VM. As the VM starts, the routing appliance **805** may then provide network functionalities such as DHCP, DNS, routing, load balancing, and firewall protection to the VM. After the last VM associated with customer **205** is stopped, the management server garbage **215** may garbage collect the routing appliances **805** after a defined interval. One routing appliance **805** may be needed per customer account per zone **110**.

In one embodiment, each customer **205** is assigned a guest virtual network in each zone **110**. A guest virtual network may be configured to any private address space, for example the Class A network in 10.0.0.0/8 private address space. The guest virtual network is an overlay network on top of the management network **315** and is managed by the multitenant hypervisor **510**.

A guest virtual network may be valid within one zone **110**. Therefore virtual machines in different zones **110** cannot communicate with each other using their IP addresses in the guest virtual network. Virtual machines in different zones **110** communicate with each other by routing through a public IP address.

A routing appliance **805** is associated with each guest virtual network. The routing appliance **805** automatically assigns an IP address for each virtual machine associated with the customer **205**, for example, in the 10.0.0.0/8 network. The

16

customer **205** may manually reconfigure virtual machines to assume different IP addresses as long as the customer **205** does not introduce IP address conflicts.

Source NAT is automatically configured in the routing appliance **805** to forward out-bound traffic for all virtual machines associated with the customer **205**. The customer **205** may configure port forwarding rules to direct inbound public traffic to the virtual machines associated with the customer **205**. A management server **215** programs the routing appliance **805** and the firewall **320** according to the port forwarding rules specified by the customer **205**. A customer **205** may also configure a software load balancer running on the routing appliance **805** to dispatch inbound traffic to multiple virtual machines associated with the customer **205** according to customer-specified load balancing rules.

The cloud infrastructure can support multiple guest virtual networks per customer **205**. The concept of routing appliances **805** as virtual machines may be generalized to include virtual machines with multiple virtual NIC interfaces and connected to multiple guest virtual networks. The virtual NIC interfaces are discussed further in conjunction with FIG. 8B.

As shown in FIG. 8A, although some of the physical resources are shared by the customers **205a**, **205b**, traffic between the two customers **205a**, **205b** and their virtual machines **505** is segmented using two separate routing appliances **805**, as are the individual virtual machines **505** assigned to each customer **205**.

The customer **205** can provide a functional specification for the virtual machine **505** (i.e., that the virtual machine **505** be capable of performing one or more specified computing tasks while meeting specified performance criteria) or the customer **205** can provide a resource specification for the virtual machine **505** (i.e., that the virtual machine **505** include specified computing resources such as hardware and/or software elements). The virtual machine **505** can also be configured according to any number of default specifications. Once the virtual machine **505** is configured, the customer **205** can access the virtual machine **505** over the network **230** and thereby interact with the virtual machine **505** to accomplish tasks as desired. For example, the customer **205** can utilize remote access software such as secure shell and/or virtual displays to control the virtual machine **505**.

FIG. 8B shows additional detail for one embodiment of traffic routed between the different virtual machines **505c-505e** within computing server **420b**, when the traffic is to/from two different customers **205a**, **205b**. Traffic from customer **205a** goes through routing appliance **805a**, when it meets the physical NIC of domain zero ("DOM 0") associated with the corresponding virtual machine (**505d**), it goes through a separate bridge (1) reserved only for traffic from customer **205a** before being routed to Virtual NIC2 for virtual machine **505d**. In some embodiments of the invention, the domain zero is also the multitenant hypervisor **510**.

The agent running on the computing server **420**, on behalf of the management server **215** ordering the instantiation, sets up a number of virtual networking components that co-operate to achieve communication between the virtualized operating system and the hardware. The virtual NIC is an emulated network adapter that shuttles traffic back and forth between dom0 and VM **505**. The bridge is a virtual switch that shuttles traffic back and forth between two segments-in this case, the network adapter of the customer virtual machine and VLAN network adapter. The bridge may also comprise a VNIC in some embodiments. The VNIC is a pseudo-network adapter that tags and untags frames in the communication using a standard, for example the 802.1q trunking standard.

17

Similarly, traffic from customer **205b** goes through routing device **805b**, and when it meets the physical NIC of DOM 0, it goes instead through a separate bridge (2) for customer **205b** traffic, before reaching virtual NIC1 or virtual NIC3 of virtual machines **505c** and **505e**, respectively. Note that even though customer **205b** has two virtual machines (**505c**, **505e**) within the computing server **420b**, it has only one bridge (2), as only one is needed for each customer **205b**. Doing so allows traffic for the same customer (**205b**) to be seen, but not traffic to/from other customers (e.g., **205a**) on the same computing server **420b**.

The cloud infrastructure combines physical host-based network virtualization and physical network/VLAN-based network virtualization. Each customer's virtual machine gets the illusion of participating in a physically distinct local area network, with the customer's other virtual machines. It is possible for example that both customer A's private network and customer B's private network is in the IP address space 10.1.1.0/24 but they never see each other's packets. It is possible that the networks addresses are not shared as well.

A prospective customer **205x** may contact the administrator **210**, requesting resources from the cloud infrastructure. The administrator **210** registers a customer account in the name of customer **205x**. A management server **215** reserves and assigns a unique VLAN ID V_x , for example within the 802.1q VLAN range (12-bit number totaling **4000** possible unique IDs) to the customer. If no unused VLAN IDs are found, the customer **205x** is informed that it is not possible to start a virtual machine.

In some embodiments, this ID is stored in a database that the management server **215** uses. The customer **205x** now starts a virtual machine within the cloud infrastructure and the management server **215** orders the instantiation of a virtual machine within one of the computing servers **420**.

V_y is then transmitted to the agent running on the computing server **420** in question, as part of the virtual machine instantiation procedure. The agent uses V_y to set up a tagged VLAN interface that will shuffle the correct traffic back and forth between the network and the virtual machine. In some embodiments, this setup communication happens through the management network **315**. The management network **315** shares the same physical network but is effectively segmented from other customers because it has a different (or no) tag.]

Traffic coming out of the virtual machines set up in this fashion will be tagged with the VLAN number as the packet travels through the multitenant hypervisor **510**; traffic coming into that interface tagged with V_y will be unwrapped from the tagged frame and delivered to the appropriate virtual machine based on the MAC address.

When a virtual machine associated with a customer actually starts on a particularly server **240**, the virtual NIC required to provide multi-tenancy is created on the server **240** and associated with the virtual machine to provide segmented network access. When the user VM is stopped, the virtual NIC is destroyed (i.e. garbage collected) and its resources returned to the server **240**.

In the embodiment of FIG. 8B, for VM **505c** and VM **505e**, there's only one virtual NIC and bridge created because they belong to the same customer and the virtual machines are allowed to see each other's traffic. This makes the virtual NIC garbage collection more difficult as the agent cannot terminate the bridge until all virtual machines for that customer are stopped on that server **240**. Each bridge and virtual NIC created requires resources on the server **240** in terms of memory. The agent may create and garbage collect bridges and virtual NICs as needed.

18

FIG. 9 illustrates switching in the network architecture of the scalable cloud infrastructure, in accordance with one embodiment. For the sake of illustration, two pods each with one computing server **420**, one routing server **410** and one storage server and an MSC **120** with one management server **215**, one load balancer **605** and one storage **610** are displayed. The firewall **320** provides connection to the network **230** and is configured in routing mode. The firewall **320** forwards HTTP requests and API calls from the network **230** to a management server **215**. The public traffic, which may be routed on a VLAN, also passes through the firewall **320** en-route to the zone layer switch **305**. Additionally, in some embodiments, the firewalls **320** may route traffic on internal networks between pods **115**. In certain embodiments, the firewall **320** enables site-to-site VPN such that servers **240** and management servers **215** in different zones **110** can reach each other.

In the embodiment of FIG. 9, the zone level switch **305** comprises two layer-3 switches L3 switches **910**. Additionally, redundant pairs of layer-2 switches L2 switches **905** are deployed in the pod layer switches **405**. In large deployments of pods **115** that include multiple racks, a second level of pod layer switch **405** may be deployed.

The present invention has been described in particular detail with respect to various embodiments, and those of skill in the art will appreciate that the invention may be practiced in other embodiments. In addition, those of skill in the art will appreciate the following aspects of the disclosure. First, the particular naming of the components, capitalization of terms, the attributes, data structures, or any other programming or structural aspect is not mandatory or significant, and the mechanisms that implement the invention or its features may have different names, formats, or protocols. Second, the system may be implemented via a combination of hardware and software, as described, or entirely in hardware elements. Third, the particular division of functionality between the various system components described herein is merely exemplary, and not mandatory; functions performed by a single system component may instead be performed by multiple components, and functions performed by multiple components may instead performed by a single component.

Some portions of above description describe the embodiments in terms of algorithms and symbolic representations of operations on information. These algorithmic descriptions and representations are commonly used by those skilled in the data processing arts to convey the substance of their work effectively to others skilled in the art. These operations, while described functionally, computationally, or logically, are understood to be implemented by computer programs or equivalent electrical circuits, microcode, or the like. Furthermore, it has also proven convenient at times, to refer to these arrangements of operations as modules, without loss of generality. The described operations and their associated modules may be embodied in software, firmware, hardware, or any combinations thereof.

In addition, the terms used to describe various quantities, data values, and computations are understood to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer

19

system memories or registers or other such information storage, transmission or display devices.

The present invention also relates to apparatus (e.g., computer systems) for performing the operations herein. These apparatuses may be specially constructed for the required purposes, or it may comprise a general-purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a non-transitory, computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, application specific integrated circuits (ASICs), or any type of non-transitory media suitable for persistently storing electronic instructions, and each coupled to a computer system bus. Furthermore, the computers referred to in the specification may include a single processor or may be architectures employing multiple processor designs for increased computing capability.

As used herein any reference to “one embodiment” or “an embodiment” means that a particular element, feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment. The appearances of the phrase “in one embodiment” in various places in the specification are not necessarily all referring to the same embodiment.

As used herein, the terms “comprises,” “comprising,” “includes,” “including,” “has,” “having” or any other variation thereof, are intended to cover a non-exclusive inclusion. For example, a process, method, article, or apparatus that comprises a list of elements is not necessarily limited to only those elements but may include other elements not expressly listed or inherent to such process, method, article, or apparatus. Further, unless expressly stated to the contrary, “or” refers to an inclusive or and not to an exclusive or. For example, a condition A or B is satisfied by any one of the following: A is true (or present) and B is false (or not present), A is false (or not present) and B is true (or present), and both A and B are true (or present).

In addition, use of the “a” or “an” are employed to describe elements and components of the embodiments herein. This is done merely for convenience and to give a general sense of the invention. This description should be read to include one or at least one and the singular also includes the plural unless it is obvious that it is meant otherwise.

Finally, it should be noted that the language used in the specification has been principally selected for readability and instructional purposes, and may not have been selected to delineate or circumscribe the inventive subject matter. Thus, upon reading this disclosure, those of skill in the art will appreciate still additional alternative structural and functional designs for a system and a process providing a scalable architecture for cloud computing. Accordingly, while particular embodiments and applications have been illustrated and described, it is to be understood that the disclosed embodiments are not limited to the precise construction and components disclosed herein. Various modifications, changes and variations, which will be apparent to those skilled in the art, may be made in the arrangement, operation and details of the method and apparatus disclosed herein without departing from the spirit and scope defined in the following claims.

What is claimed is:

1. A method for establishing a scalable cloud infrastructure, the method comprising:

partitioning a plurality of physical resources into a plurality of pods within a plurality of zones, wherein:

20

the physical resources comprise physical compute, storage, and networking resources within a plurality of data centers distributed across a network;

each zone comprises a weakly connected subset of the plurality of pods and is physically isolated from other zones in the plurality, and one or more zone-level layer-3 network switches providing interconnections among the pods of the zone and providing connections to other zones; and

each pod comprises a strongly connected discrete set of the plurality of physical resources in the zone, the physical resources in each pod including one or more compute servers, one or more storage servers, and one or more routing servers connected together by one or more pod-level layer-2 network switches;

establishing at least one data network from the distributed networking resources, the at least one data network connecting, and configured for routing traffic to, the plurality of pods from at least one customer;

establishing at least one management server cluster comprising a plurality of management servers;

establishing at least one management network from the distributed networking resources, the at least one management network connecting one of the plurality of management servers to one of the plurality of zones, and connecting the compute and storage resources partitioned within the pod;

establishing a plurality of units of virtual resources by apportioning selected of the physical resources within one or more of the plurality of pods to each of the plurality of units, wherein the apportioned physical resources may be shared between two or more of the plurality of units of virtual resources; and

wherein each management server is configured for allocating one or more of the plurality of units of virtual resources to an account associated with the at least one customer; and

wherein the compute servers of a pod execute a multitenant hypervisor, and the storage servers of the pod present virtual disks to the compute servers, the virtual disks being visible to the multitenant hypervisor as networked file systems or as block devices and being presented as locally attached storage to virtual machines of the compute servers.

2. A method for segmenting traffic from a plurality of customer accounts within a scalable cloud infrastructure, the method comprising:

allocating one or more units of virtual resources to each of a plurality of customer accounts, each of the one or more units of virtual resources corresponding to physical resources partitioned into a plurality of physically isolated zones within the scalable cloud infrastructure, the plurality of physical resources comprising a plurality of physical compute, storage, and networking resources within a plurality of data centers distributed across a network, wherein the plurality of physical resources are shared at least in part between two or more of the plurality of customer accounts, and wherein for each customer account, the physical resources are selected from one or more of a plurality of weakly connected pods within one of the plurality of zones, each zone including one or more zone-level layer-3 network switches providing interconnections among the pods of the zone and providing connections to other zones, each of the plurality of pods comprising a strongly connected discrete set of the plurality of physical resources within the zone, the physical resources in each pod including one or more

21

compute servers, one or more storage servers, and one or more routing servers connected together by one or more pod-level layer-2 network switches; and
 for the physical resources that are shared at least in part between two or more of the plurality of customer accounts, segmenting traffic from the one or more units of virtual resources allocated to a first of the two or more of the plurality of customer accounts from the one or more units of virtual resources allocated to a second of the two or more of the plurality of customer accounts;
 wherein the compute servers of a pod execute a multitenant hypervisor, and the storage servers of the pod present virtual disks to the compute servers, the virtual disks being visible to the multitenant hypervisor as networked file systems or as block devices and being presented as locally attached storage to virtual machines of the compute servers.

3. The method of claim 2, wherein segmenting further comprises:
 allocating a first subset of the at least in part shared physical resources exclusively to the first customer account and a second subset of the at least in part shared physical resources exclusively to the second customer account, the first and second subsets each comprising at least one virtual machine; and
 segmenting first customer account traffic across the at least in part shared physical resources corresponding to the one or more units of virtual resources allocated to the first customer account from second customer account traffic across the at least in part shared physical resources corresponding to the one or more units of virtual resources allocated to the first customer account.

4. A system for a scalable cloud infrastructure, the system comprising:
 a plurality of physical resources partitioned into a plurality of pods within a plurality of zones, wherein:
 the physical resources comprise physical compute, storage, and networking resources within a plurality of data centers distributed across a network;
 each zone comprises a weakly connected subset of the plurality of pods and is physically isolated from other zones in the plurality, and one or more zone-level layer-3 network switches providing interconnections among the pods of the zone and providing connections to other zones; and
 each pod comprises a strongly connected discrete set of the plurality of physical resources in the zone, the physical resources in each pod including one or more compute servers, one or more storage servers, and one or more routing servers connected together by one or more pod level layer-2 network switches;
 at least one data network from the distributed networking resources, the at least one data network connecting, and configured for routing traffic to, the plurality of pods from at least one customer;
 at least one management server cluster comprising a plurality of management servers;
 at least one management network from the distributed networking resources, the at least one management network connecting one of the plurality of management servers to one of the plurality of zones, and connecting the compute and storage resources partitioned within the pod;
 a plurality of units of virtual resources by apportioning selected of the physical resources within one or more of the plurality of pods to each of the plurality of units,

22

wherein the apportioned physical resources may be shared between two or more of the plurality of units of virtual resources; and
 wherein each management server is configured for allocating one or more of the plurality of units of virtual resources to an account associated with the at least one customer; and
 wherein the compute servers of a pod execute a multitenant hypervisor, and the storage servers of the pod present virtual disks to the compute servers, the virtual disks being visible to the multitenant hypervisor as networked file systems or as block devices and being presented as locally attached storage to virtual machines of the compute servers.

5. A system according to claim 4, wherein the storage servers of each pod provide pod-level shared storage including an ability to restart a virtual machine on a compute server of the pod different from another compute server of the pod where the virtual machine last ran.

6. A system according to claim 4, wherein storage bandwidth throttling is used to limit a rate of storage traffic from a virtual machine while reading from or writing to a virtual disk.

7. A system according to claim 6, wherein the storage throttling is specified by a desired maximum 10 per second value configured by a management server on either a per-service basis, a per-customer basis, or across the system.

8. A system according to claim 4, wherein virtual LANs are used to segment traffic within the data network between management, public and storage traffic.

9. A system according to claim 4, wherein virtual LANs are used to segment traffic within the data network among different customers.

10. A system according to claim 9, wherein a virtual LAN is allocated to a customer in response to a request from a customer for resources from the system.

11. A system according to claim 9, wherein network bandwidth throttling is used to limit a rate of network traffic from virtual machines of customers.

12. A system according to claim 11, wherein the network bandwidth throttling is specified by a desired maximum network bandwidth for each virtual machine, configured by a management server on either a per-service basis, a per-customer basis, or across the system.

13. A system according to claim 4, wherein a customer provides a resource specification for virtual machines to be used by the customer, the resource specification including specification of hardware and/or software elements.

14. A system according to claim 4, wherein the routing servers include routing appliances executed as virtual machines providing per customer secure and protected routing functionality, the routing appliances being garbage collected when no longer needed so as to release system resources for other uses.

15. A system according to claim 14, wherein the routing functionality includes automatic network address assignment and source network address translation for virtual machines of the respective customer.

16. A system according to claim 4, wherein the management servers are organized into one or more management server clusters.

17. A system according to claim 16, wherein each management server cluster is deployed as a primary management server cluster in a first data center with a backup management server cluster at a second distinct data center.

18. A system according to claim 17, wherein data stored by the primary management server cluster is replicated and

23

transmitted for storage by backup management server cluster,
and in the event of failure of the primary management server
cluster network traffic is automatically directed to the second-
ary management server cluster instead of the primary man-
agement server cluster such that operation of the servers in the 5
zones is unaffected by the failure of the primary management
server cluster.

19. A system according to claim **16**, wherein a manage-
ment server cluster serves multiple zones and pods within the
data centers.

10

* * * * *

24