UNIT –III

INFRASTRUCTURE AS A SERVICE (IAAS)

Virtual Machines Provisioning And Migration Services

INTRODUCTION AND INSPIRATION

In this chapter, we shall focus on two core services that enable the users to get the best out of the IaaS model in public and private cloud setups. These services are named virtual machine provisioning and migration services. We will also cover their concepts, techniques, and research directions, along with an introductory overview about virtualization technology and its role as a fundamental component/block of the cloud computing architecture stack.

5.2.1 Virtualization Technology Overview

Virtualization has revolutionized data center's technology through a set of techniques and tools that facilitate the providing and management of the dynamic data center's infrastructure. It has become an essential and enabling technology of cloud computing environments. Virtualization can be defined as the abstraction of the four computing resources (storage, processing power, memory, and network or I/O). It is conceptually similar to emulation, where a system pretends to be another system, whereas virtualization is a system pretending to be two or more of the same system

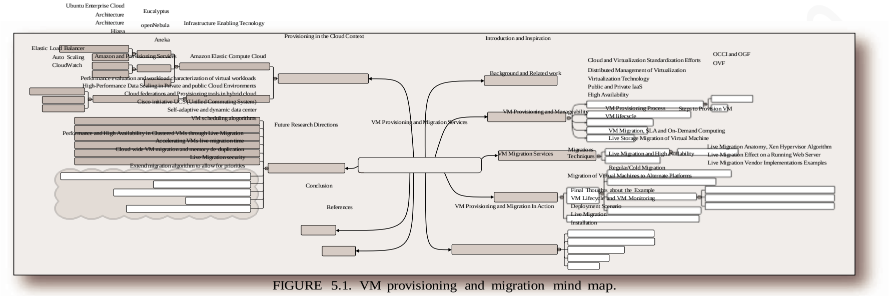

Ubuntu Enterprise Cloud
Architecture
Architecture
Hizea
Eucalyptus
openNebula
Infrastructure Enabling Tecnology
Aneka
Provisioning in the Cloud Context
Elastic Load Balancer
Auto Scaling
CloudWatch
Amazon and Provisioning Services
Amazon Elastic Compute Cloud
Performance evaluation and workload characterization of virtual workloads
High-Performance Data Scaling in Private and public Cloud Environments
Cloud federations and Provisioning tools in hybrid cloud
Cisco initiative UCS (Unified Commuting System)
Self-adaptive and dynamic data center
VM scheduling alogorithms
Performance and High Availability in Clustered VMs through Live Migration
Accelerating VMs live migration time
Cloud-wide VM migration and memory de-duplication
Live Migration security
Extend migration algorithm to allow for priorities
Future Research Directions
Conclusion
References
VM Provisioning and Migration Services
Introduction and Inspiration
Background and Related work
Cloud and Virtualization Standardization Efforts
OCCI and OGF
OVF
Distributed Management of Virtualization
Virtualization Technology
Public and Private IaaS
High Availability
VM Provisioning and Manageability
VM Provisioning Process
Steps to Provision VM
VM lifecycle
VM Migration, SLA and On-Demand Computing
Live Storage Migration of Virtual Machine
VM Migration Services
Migrations Techniques
Live Migration and High Availability
Live Migration Anatomy, Xen Hypervisor Algorithm
Live Migration Effect on a Running Web Server
Live Migration Vendor Implementations Examples
Regular/Cold Migration
Migration of Virtual Machines to Alternate Platforms
VM Provisioning and Migration In Action
Final Thoughts about the Example
VM Lifecycle and VM Monitoring
Deployment Scenario
Live Migration
Installation

FIGURE 5.1. VM provisioning and migration mind map.

125

126 VIRTUAL MACHINES PROVISIONING AND MIGRATION SERVICES

Virtual Machine	Virtual Machine	Virtual Machine	Virtual Machine
Workload 1	Workload 2	Workload 3	Workload n
Guest OS	Guest OS	Guest OS	Guest OS

Figure 5.2, the virtualization layer will partition the physical resource of the underlying physical server into multiple virtual machines with different work- loads. The fascinating thing about this virtualization layer is that it schedules, allocates the physical resource, and makes each virtual machine think that it totally owns the whole underlying hardware's physical resource

(processor, disks, rams, etc.) Virtual machine's technology makes it very flexible and easy to manage resources in cloud computing environments, because they improve the utiliza- tion of such resources by multiplexing many virtual machines on one physical host (server consolidation), as shown in Figure 5.1. These machines can be scaled up and down on demand with a high level of resources' abstraction.

Virtualization enables high, reliable, and agile deployment mechanisms and management of services, providing on-demand cloning and live migration services which improve reliability. Accordingly, having an effective manage- ment's suite for managing virtual machines' infrastructure is critical for any cloud computing infrastructure as a service (IaaS) vendor.

Public Cloud and Infrastructure Services

Public cloud or external cloud describes cloud computing in a traditional mainstream sense, whereby resources are dynamically provisioned via publicly accessible Web applications/Web services (SOAP or RESTful interfaces) from an off-site third-party provider, who shares resources and bills on a fine-grained utility computing basis , the user pays only for the capacity of the provisioned resources at a particular time. There are many examples for vendors who publicly provide infrastructure as a service. Amazon Elastic Compute Cloud (EC2)[4] is the best known example, but the market now bristles with lots of competition like GoGrid [5], Joyent Accelerator [6], Rackspace [7], AppNexus [8], FlexiScale [9], and Manjrasoft Aneka [10].

Private Cloud and Infrastructure Services

A private cloud aims at providing public cloud functionality, but on private resources, while maintaining control over an organization's data and resources to meet security and governance's requirements in an organization. Private cloud exhibits a highly virtualized cloud data center located inside your organi- zation's firewall. It may also be a private space dedicated for your company within a cloud vendor's data center designed to handle the organization's workloads.

Private clouds exhibit the following characteristics:

- Allow service provisioning and compute capability for an organization's users in a self-service manner.
- Automate and provide well-managed virtualized environments.
- Optimize computing resources, and servers' utilization.
- Support specific workloads.

There are many examples for vendors and

frameworks that provide infrastruc- ture as a service in private setups. The best-known examples are Eucalyptus [11] and OpenNebula [12] (which will be covered in more detail later on).

5.2.4 Distributed Management of Virtualization

Virtualization’s benefits bring their own challenges and complexities presented in the need for a powerful management capabilities. That is why many commercial, open source products and research projects such as OpenNebula [12], IBM Virtualization Manager, Joyent, and VMware DRS are being developed to dynamically provision virtual machines, utilizing the physical infrastrcture. There are also some commercial and scientific infrastructure cloud computing initiatives, such as Globus VWS, Eucalyptus [11] and Amazon, which provide remote interfaces for controling and monitoring virtual resources.

High Availability

High availability is a system design protocol and an associated implementation that ensures a certain absolute degree of operational continuity during a given measurement period. Availability refers to the ability of a user's community to access the system—whether for submiting new work, updating or altering existing work, or collecting the results of the previous work

Cloud and Virtualization Standardization Efforts

Standardization is important to ensure interoperability between virtualization mangement vendors, the virtual machines produced by each one of them, and cloud computing. Here, we will have look at the prevalent standards that make cloud computing and virtualization possible

OCCI and OGF

Another standardization effort has been initiated by Open Grid Forum (OGF) through organizing an official new working group to deliver a standard API for cloud IaaS, the Open Cloud Computing Interface Working Group (OCCI- WG). This group is dedicated for delivering an API specification for the remote management of cloud computing's infrastructure and for allowing the devel- opment of interoperable tools for common tasks including deployment, autonomic scaling, and monitoring.

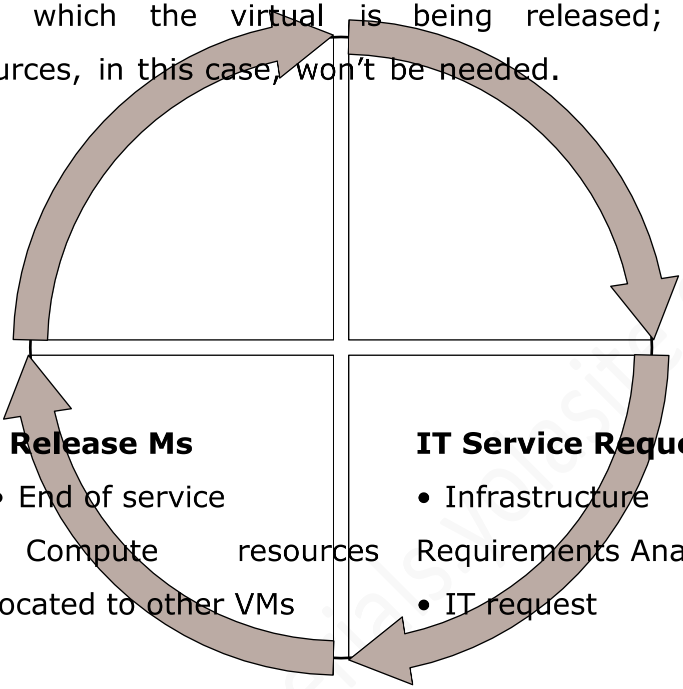
- Consumers to interact with cloud computing infrastructure on an ad hoc basis.
- Integrators to offer advanced management services.
- Aggregators to offer a single common interface to multiple providers.
- Providers to offer a standard interface that is compatible with the available tools.
- Vendors of grids/clouds to offer standard interfaces for dynamically scalable service's delivery in their products.

VIRTUAL MACHINES PROVISIONING AND MANAGEABILITY

The typical life cycle of VM and its major possible states of operation, which make the management and automa- tion of VMs in virtual and cloud environments easier than in traditional computing environments.

As shown in Figure 5.3, the cycle starts by a request delivered to the IT department, stating the requirement for creating a new server for a particular service. This request is being processed by the IT administration to start seeing the servers' resource pool, matching these resources with the requirements, and starting the provision of the needed virtual machine.

Once it is provisioned and started, it is ready to provide the required service according to an SLA, or a time period after which the virtual is being released; and free resources, in this case, won't be needed.



Release Ms

- End of service
- Compute resources deallocated to other VMs

IT Service Request

- Infrastructure Requirements Analysis
- IT request

VMs In Operation

- Serving web requests
- Migration services
- Scal on-demand compute resources

VM Provision

- Load OS Appliances
- Customize and Configure
- Start the server

FIGURE 5.3. Virtual machine life cycle.

VM Provisioning Process

Provisioning a virtual machine or server can be explained and illustrated as in Figure 5.4:

Steps to Provision VM. Here, we describe the common and normal steps of provisioning a virtual server:

- Firstly, you need to select a server from a pool of available servers (physical servers with enough capacity) along with the appropriate OS template you need to provision the virtual machine.
- Secondly, you need to load the appropriate software (operating system you selected in the previous step, device drivers, middleware, and the needed applications for the service required).
- Thirdly, you need to customize and configure the machine (e.g., IP address, Gateway) to configure an associated network and storage resources.

Finally, the virtual server is ready to start with its newly loaded software. Typically, these are the tasks required or being performed by an IT or a data center's specialist to provision a particular virtual machine.

Migrations Techniques

Live Migration and High Availability. Live migration (which is also called hot or real-time migration) can be defined as the movement of a virtual machine from one physical host to another while being powered on. When it is properly carried out,

Live Migration Anatomy, Xen Hypervisor Algorithm. In this section we will explain live migration's mechanism and how memory and virtual machine states are being transferred, through the network, from one host A to another host B :

Stage 0: Pre-Migration. An active virtual machine exists on the physical host A.

Stage 1: Reservation. A request is issued to migrate an OS from host A to host B (a precondition is that the necessary resources exist on B and on a VM container of that size).

Stage 2: Iterative Pre-Copy. During the first iteration, all pages are transferred from A to B. Subsequent iterations copy only those pages dirtied during the previous transfer phase.

Stage 3: Stop-and-Copy. Running OS instance at A is suspended, and its network traffic is redirected to B. As described in reference 21, CPU state and any remaining inconsistent memory pages are then transferred. At the end of this stage, there is a consistent suspended copy of the VM at both A and B. The copy at A is considered primary and is resumed in case of failure.

Stage 4: Commitment. Host B indicates to A that it has successfully received a consistent OS image. Host A acknowledges this message as a commit- ment of the migration transaction. Host A may now discard the original VM, and host B becomes the primary host.

Stage 5: Activation. The migrated VM on B is now activated. Post-migration code runs to reattach the device's drivers to the new machine and advertise moved IP addresses

VM Migration, SLA and On-Demand Computing

virtual machines' migration plays an important role in data centers by making it easy to adjust resource's priorities to match resource's demand conditions.

This role is completely going in the direction of meeting SLAs; once it has been detected that a particular VM is consuming more than its fair share of resources at the expense of other VMs on the same host, it will be eligible, for this machine, to either be moved to another underutilized host or assign more resources for it, in case that the host machine still has resources; this in turn will highly avoid the violations of the SLA and will also, fulfill the requirements of on-demand computing resources.

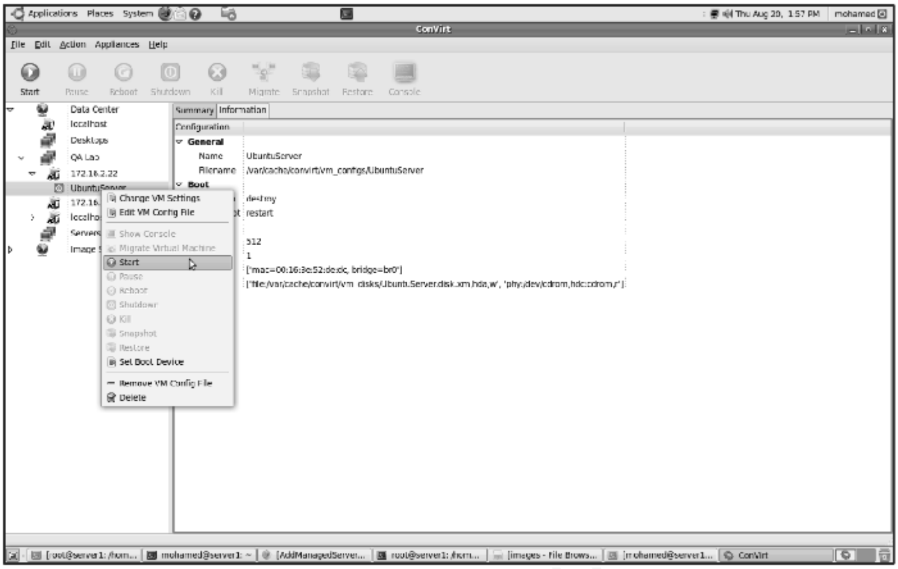
VM PROVISIONING AND MIGRATION IN ACTION

Now, it is time to get into business with a real example of how we can manage the life cycle, provision, and migrate a virtual machine by the help of one of the open source frameworks used to manage virtualized infrastructure. Here, we will use ConVirt [25] (open source framework for the management of open source virtualization like Xen [26] and KVM [27], known previously as XenMan).

Deployment Scenario: ConVirt deployment consists of at least one ConVirt workstation, where ConVirt is installed and ran, which provides the main console for managing the VM life cycle, managing images, provisioning new VMs, monitoring machine resources, and so on.

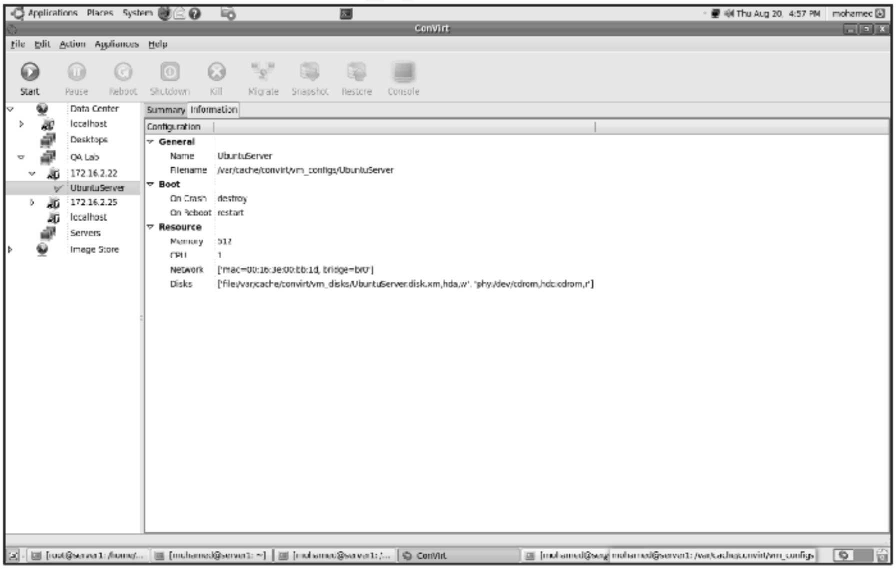
VM Life Cycle and VM Monitoring

You can notice through working with ConVirt that you are able to manage the whole life cycle of the virtual machine; start, stop, reboot, migrate, clone, and so on. Also, you noticed how easy it is to monitor the resources of the managed server and to monitor the virtual machine's guests that help you balance and control the load on these managed servers once needed. In the next section, we are going to discuss how easy it is to migrate a virtual machine from host to host.



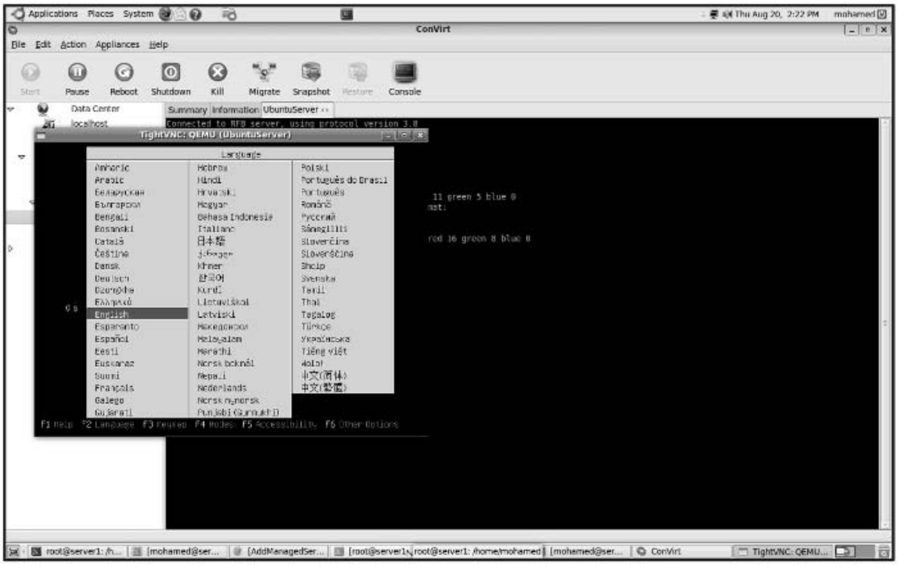
```
Configuration
General
    Name        UbuntuServer
    Filename    /var/cache/convirt/vm_configs/UbuntuServer
Boot
            t   destroy
            t   restart

                512
                1
                ['mac=00:16:3e:52:de:dc, bridge=br0']
                ['file:/var/cache/convirt/vm_disks/Ubuntu.Server.disk.xm,hda,w', 'phy:/dev/cdrom,hdc:cdrom,r']
```

FIGURE 5.14. Provisioned VM ready to be started.



```
Configuration
General
    Name        UbuntuServer
    Filename    /var/cache/convirt/vm_configs/UbuntuServer
Boot
    On Crash    destroy
    On Reboot   restart
Resource
    Memory      512
    CPU         1
    Network     ['mac=00:16:3e:00:bb:1d, bridge=br0']
    Disks       ['file:/var/cache/convirt/vm_disks/UbuntuServer.disk.xm,hda,w', 'phy:/dev/cdrom,hdc:cdrom,r']
```

FIGURE 5.15. Provisioned VM started.



```
Connected to RFB server, using protocol version 3.8
 11 green 5 blue 0
mat:
red 16 green 8 blue 8
```

FIGURE 5.16. VM booting from the installation CD to start the installation process.

5.5.2 Live Migration

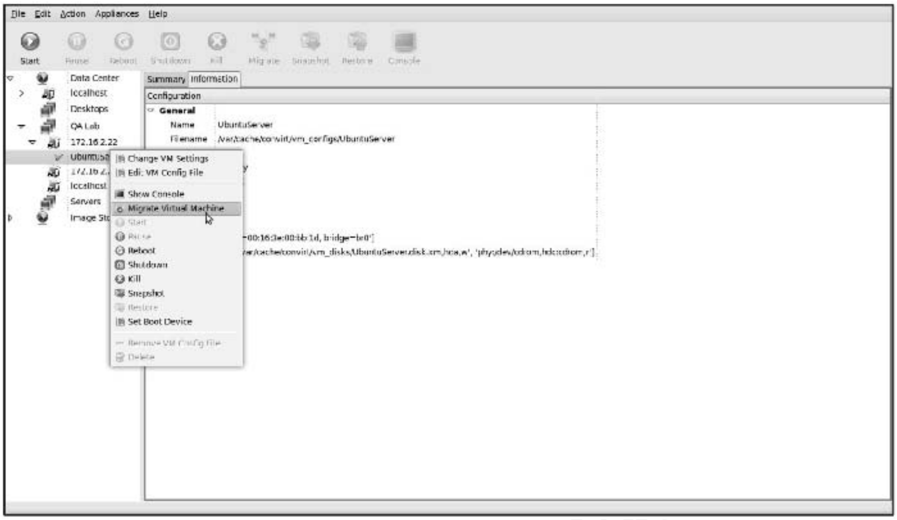
ConVirt tool allows running virtual machines to be migrated from one server to another [29].This feature makes it possible to organize the virtual machine to physical machine relationship to balance the workload; for example, a VM needing more CPU can be moved to a machine having available CPU cycles, or, in other cases, like taking the host machine for maintenance. For proper VM migration the following points must be considered

- Shared storage for all Guest OS disks (e.g., NFS, or iSCSI).
- Identical mount points on all servers (hosts).
- The kernel and ramdisk when using para-virtualized virtual machines should, also, be shared. (This is not required, if pygrub is used.)
- Centrally accessible installation media (iso).
- It is preferable to use identical machines with the same version of virtualization platform.
- Migration needs to be done within the same subnet.

Migration Process in ConVirt

To start the migration of a virtual machine from one host to the other, select it and choose a migrating virtual machine, as shown in Figure 5.17.

You will have a window containing all the managed servers in your data center (as shown in Figure 5.18). Choose one as a destination and start



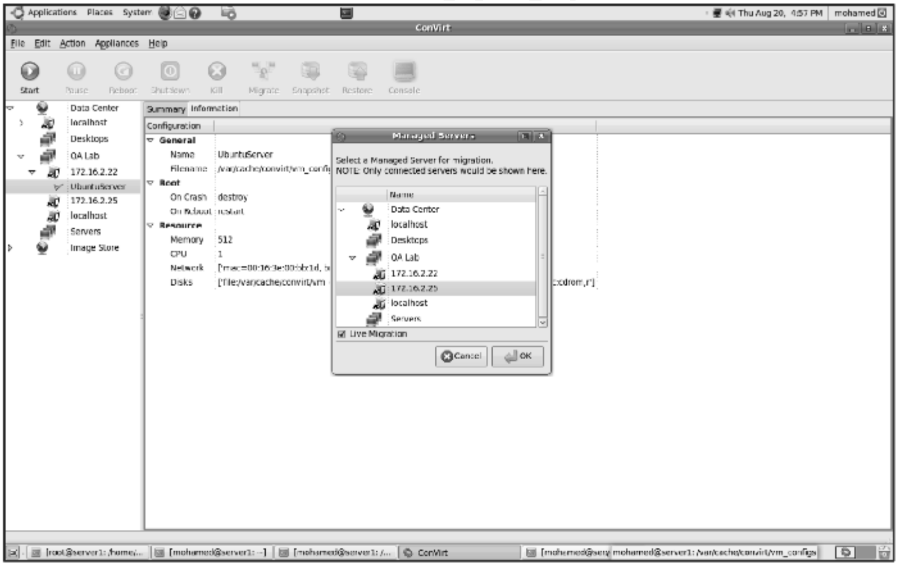
Name UbuntuServer

Filename /var/cache/convirt/vm_configs/UbuntuServer

=00:16:3e:00:bb:1d, bridge=br0']

/var/cache/convirt/vm_disks/UbuntuServer.disk.xm,hda,w', 'phy:dev/cdrom,hdc:cdrom,r']

FIGURE 5.17. VM migration.



Name UbuntuServer

Filename /var/cache/convirt/vm_confi

On Crash destroy

On Reboot restart

Memory 512

CPU 1

Network ['mac=00:16:3e:00:bb:1d, b

Disks ['file:/var/cache/convirt/vm

Select a Managed Server for migration.
NOTE: Only connected servers would be shown here.

☑ Live Migration

FIGURE 5.18. Select the destination managed server candidate for migration.

migration, or drag the VM and drop it on to another managed server to initiate migration.

Once the virtual machine has been successfully placed and migrated to the destination host, you can see it still living and working (as shown in Figure 5.19).

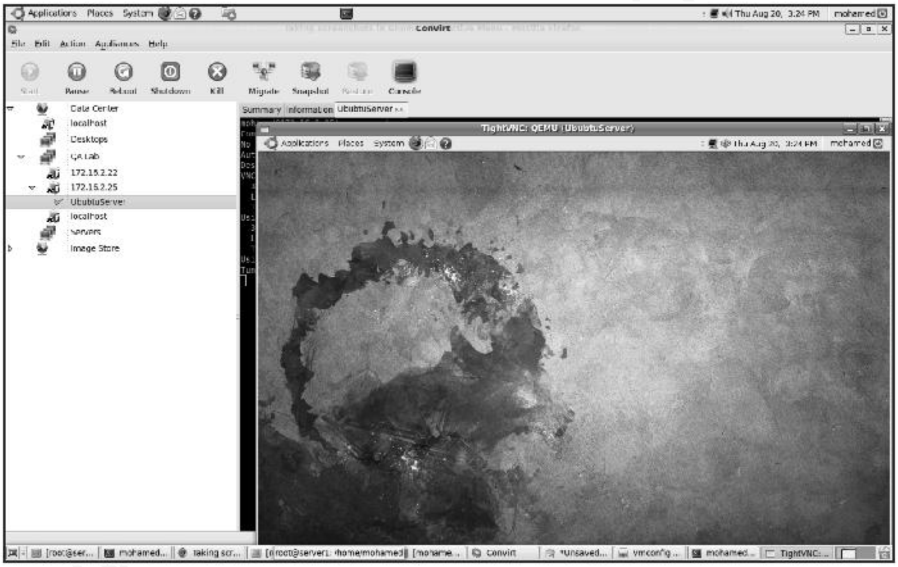


FIGURE 5.19. VM started on the destination server after migration.

5.6.1 Amazon Elastic Compute Cloud

The Amazon EC2 (Elastic Compute Cloud) is a Web service that allows users to provision new machines into Amazon's virtualized infrastructure in a matter of minutes; using a publicly available API (application programming interface), it reduces the time required to obtain and boot a new server. Users get full root access and can install almost any OS or application in their AMIs (Amazon Machine Images). Web services APIs allow users to reboot their instances remotely, scale capacity quickly, and use on-demand service when needed; by adding tens, or even hundreds, of machines.

Amazon EC2 provides its customers with three flexible purchasing models to make it easy for the cost optimization:

- On-Demand instances, which allow you to pay a fixed rate by the hour with no commitment.

- Reserved instances, which allow you to pay a low, one-time fee and in turn receive a significant discount on the hourly usage charge for that instance. It ensures that any reserved instance you launch is guaranteed to succeed (provided that you have booked them in advance). This means that users of these instances should not be affected by any transient limitations in EC2 capacity.

- Spot instances, which enable you to bid whatever price you want for instance capacity, providing for even greater savings, if your applications have flexible start and end times.

Amazon and Provisioning Services: Amazon provides an excellent set of tools that help in provisioning service; Amazon Auto Scaling [30] is a set of command line tools that allows scaling Amazon EC2 capacity up or down automatically and according to the conditions the end user defines. This feature ensures that the number of Amazon EC2 instances can scale up seamlessly during demand spikes to maintain performance and can scale down auto- matically when loads diminish and become less intensive to minimize the costs. Auto Scaling service and CloudWatch

Infrastructure Enabling Technology

Offering infrastructure as a service requires software and platforms that can manage the Infrastructure that is being shared and dynamically provisioned. For this, there are three noteworthy technologies to be considered: Eucalyptus, OpenNebula, and Aneka.

Eucalyptus

Eucalyptus is an open-source infrastructure for the implementation of cloud computing on computer clusters. It is considered one of the earliest tools developed as a surge computing (in which data center's private cloud could augment its ability to handle workload's spikes by a design that allows it to send overflow work to a public cloud) tool.

Interface compatibility with EC2, and S3 (both Web service and Query/ REST interfaces).

- Simple installation and deployment.
- Support for most Linux distributions (source and binary packages).

- Support for running VMs that run atop the Xen hypervisor or KVM. Support for other kinds of VMs, such as VMware, is targeted for future releases.
- Secure internal communication using SOAP with WS

security.

- Cloud administrator's tool for system's management and user's accounting.

- The ability to configure multiple clusters each with private internal network addresses into a single cloud.

Eucalyptus aims at fostering the research in models for service's provisioning, scheduling, SLA formulation, and hypervisors' portability.

FIGURE 5.20. Eucalyptus high-level architecture.

- Node controller (NC) controls the execution, inspection, and termination of VM instances on the host where it runs.
- Cluster controller (CC) gathers information about and schedules VM execution on specific node controllers, as well as manages virtual instance network.

- Storage controller (SC) is a put/get storage service that implements Amazon's S3 interface and provides a way for storing and accessing VM images and user data.

- Cloud controller (CLC) is the entry point into the cloud for users and administrators. It queries node managers for information about resources, makes high-level scheduling decisions, and implements them by making requests to cluster controllers.

- Walrus (W) is the controller component that manages access to the storage services within Eucalyptus. Requests are communicated to Walrus using the SOAP or REST-based interface.

Live migration security.

Live migration security is a very important area of research, because several security's vulnerabilities exist; check reference 38 for an empirical exploitation of live migration.

Extend migration algorithm to allow for priorities.

Cisco initiative UCS (Unified Commuting System) and its role in dynamic just-in-time provisioning of virtual machines and increase of business agility.

ON THE MANAGEMENT OF VIRTUAL MACHINES FOR CLOUD INFRASTRUCTURES

This chapter focuseson the subject of IaaS clouds and, more specifically, on the efficient manage- ment of virtual machines in this type of cloud. Section 6.1 starts by discussing the characteristics of IaaS clouds and the challenges involved in managing these clouds.

RESERVOIR: (Resources and Services Virtualization without Barriers), a European Union

FP7-funded project. Section 6.2 starts by discussing the problem of managing virtual infrastructures; Section 6.3 presents scheduling techniques that can be used to provide advance reservation of capacity within these infrastructures; Section 6.4 focuses on service-level agreements (or SLAs) in IaaS clouds and discusses capacity management techniques supporting SLA commitments. Finally, the chapter concludes with a discussion of remaining challenges and future work in IaaS clouds.

DISTRIBUTED MANAGEMENT OF VIRTUAL INFRASTRUCTURES

Managing VMs in a pool of distributed physical resources is a key concern in IaaS clouds, requiring the use of a virtual infrastructure manager. To address some of the shortcomings in existing VI solutions, we have developed the open source OpenNebula[1] virtual infrastructure engine. OpenNebula is capable of managing groups of interconnected VMs—with support for the Xen, KVM, and VMWare platforms—within data centers and private clouds that involve a large amount of virtual and physical servers. OpenNebula can also be used to build hybrid clouds by interfacing with remote cloud sites [14]. This section describes how OpenNebula models and manages VMs in a virtual infrastructure.

VM Model and Life Cycle

The primary target of OpenNebula is to manage VMs. Within OpenNebula, a VM is modeled as having the following attributes:

- A capacity in terms of memory and CPU.
- A set of NICs attached to one or more virtual networks.
- A set of disk images. In general it might be necessary to transfer some of these image files to/from the physical machine the VM will be running in.
- A state file (optional) or recovery file that contains the memory image of a running VM plus some hypervisor-specific information.

The life cycle of a VM within OpenNebula follows several stages:

Resource Selection: Once a VM is requested to OpenNebula, a feasible placement plan for the VM must be made. OpenNebula’s default scheduler provides an implementation of a rank scheduling policy, allowing site administrators to configure the scheduler to prioritize the resources that are more suitable for the VM, using information from the VMs and the physical hosts. As we will describe in Section 6.3, OpenNebula can also use the Haizea lease manager to support more complex scheduling policies.

Resource Preparation: The disk images of the VM are transferred to the target physical resource. During the boot process, the VM is contextua- lized, a process where the disk images are specialized to work in a given environment

VM Creation: The VM is booted by the resource hypervisor.

VM Migration: The VM potentially gets migrated to a more suitable resource (e.g., to optimize the power consumption of the physical resources).

VM Termination: When the VM is going to shut down, OpenNebula can transfer back its disk images to a known location. This way, changes in the VM can be kept for a future use.

VM Management: OpenNebula manages a VMs life cycle by orchestrating three different management areas: virtualization by interfacing with a physical resource's hypervisor, such as Xen, KVM, or VMWare, to control (e.g., boot, stop, or shutdown) the VM; image management by transferring the VM images from an image repository to the selected resource and by creating on-the-fly temporary images; and networking by creating local area networks (LAN) to interconnect the VMs and tracking the MAC addresses leased in each network.

Virtualization: OpenNebula manages VMs by interfacing with the physical resource virtualization technology (e.g., Xen or KVM) using a set of pluggable drivers that decouple the managing process from the underlying technology. Thus, whenever the core needs to manage a VM, it uses high-level commands such as "start VM," "stop VM," and so on, which are translated by the drivers into commands that the virtual machine manager can understand.

Image Management: VMs are supported by a set of virtual disks or images, which contains the OS and any other additional software needed by the VM. OpenNebula assumes that there is an image repository that can be any storage medium or service, local or remote, that holds the base image of the VMs. There are a number of different possible configurations depending on the user's needs. For example, users may want all their images placed on a separate repository with only HTTP access.. OpenNebula uses the following concepts for its image management model (Figure 6.1):

- Image Repositories refer to any storage medium, local or remote, that hold the base images of the VMs. An image repository can be a dedicated file server or a remote URL from an appliance provider, but they need to be accessible from the OpenNebula front-end.

- Virtual Machine Directory is a directory on the cluster node where a VM is running. This directory holds all deployment files for the hypervisor to boot the machine, checkpoints, and images being used or saved—all of them specific to that VM. This directory should be shared for most hypervisors to be able to perform live migrations. Any given VM image goes through the following steps along its life

cycle:

Preparation implies all the necessary changes to be made to the machine's image so it is prepared to offer the service to which it is intended. OpenNebula assumes that the images that conform to a particular VM are prepared and placed in the accessible image repository.

SCHEDULING TECHNIQUES FOR ADVANCE RESERVATION OF CAPACITY

Commercial cloud providers, such as Amazon, rely on an immediate provisioning model where VMs are provisioned right away, since their data centers' capacity is assumed to be infinite. Thus, there is no need for other provisioning models, such as best-effort provisioning where requests have to be queued and prioritized or advance provisioning where resources are pre-reserved so they will be guaranteed to be available at a given time period; queuing and reservations are unnecessary when resources are always available to satisfy incoming requests. However, when managing a private cloud with limited resources, an immediate provisioning model is insufficient.

Existing Approaches to Capacity Reservation

Efficient reservation of resources in resource management systems has been studied considerably, particularly in the context of job scheduling. In fact, most modern job schedulers support advance reservation of resources, but their implementation falls short in several aspects. First of all, they are constrained by the job abstraction; when a user makes an advance reservation in a job- based system, the user does not have direct and unfettered access to the resources, the way a cloud users can access the VMs they requested, but, rather, is only allowed to submit jobs to them.

Additionally, it is well known that advance reservations lead to utilization problems [10 13], caused by the need to vacate resources before a reservation can begin. Unlike future reservations made by backfilling algorithms, where the start of the reservation is determined on a best-effort basis, advance reservations introduce roadblocks in the resource schedule. Thus, traditional job schedulers are unable to efficiently schedule workloads combining both best-effort jobs and advance reservations.

While preemption can be accomplished trivially by canceling a running job, the least disruptive form of preemption is checkpointing, where the preempted job's entire state is saved to disk, allowing it to resume its work from the last checkpoint. Additionally, some schedulers also support job migration, allowing checkpointed jobs to restart on other available resources, instead of having to wait until the preempting job or reservation has completed.

An application can be made checkpointable by explicitly adding that function- ality to an application (application-level and library-level checkpointing) or transparently by using OS-level checkpointing, where the operating system (such as Cray, IRIX, and patched versions of Linux using BLCR [17]) checkpoints a process, without rewriting the program or relinking it with checkpointing libraries. However, this requires a checkpointing-capable OS to be available.

Although the BLCR project does provide a checkpointing x86 Linux kernel, this kernel still has several limitations, such as not being able to properly checkpoint network traffic and not being able to checkpoint MPI applications unless they are linked with BLCR-aware MPI libraries.

Reservations with VMs

Virtual machines are also an appealing vehicle for implementing efficient reservation of resources due to their ability to be suspended, potentially migrated, and resumed without modifying any of the applications running inside the VM. However, virtual machines also raise additional challenges related to the overhead of using VMs:

Preparation Overhead: When using VMs to implement reservations, a VM disk image must be either prepared on-the-fly or transferred to the physical node where it is needed. Since a VM disk image can have a size in the order of gigabytes, this preparation overhead can significantly delay the starting time of leases. This delay may, in some cases, be unacceptable for advance reservations that must start at a specific time.

Runtime Overhead: Once a VM is running, scheduling primitives such as checkpointing and resuming can incur in significant overhead since a VM's entire memory space must be saved to disk, and then read from disk. Migration involves transferring this saved memory along with the VM disk image. Similar to deployment overhead, this overhead can result in noticeable delays.

Infrastructure SLAs

IaaS can be regarded as a giant virtual hardware store, where computational resources such as virtual machines (VM), virtual application networks (VAN) and virtual disks (VD) can be ordered on demand in the matter of minutes or even seconds. Virtualization technology is sufficiently versatile to provide virtual resources on a almost continuous granularity scale. Chandra et al. quantitatively study advantages of fine-grain resource allocation in a shared hosting platform.

These advantages come at a cost of increased management, accounting, and billing overhead. For this reason, in practice, resources are typically provided on a more coarse discrete scale. For example, Amazon EC2 [1] offers small, large, and extra large general-purpose VM instances and high-CPU medium and extra large instances.

Thus, to deploy a service on a cloud, service provider orders suitable virtual hardware and installs its application software on it. From the IaaS provider, a given service configuration is a virtual resource array of black box resources, which correspond to the number of instances of resource type.

In an IaaS model it is expected from the service provider that it sizes capacity demands for its service. If resource demands are provided correctly and are indeed satisfied upon request, then desired user experience of the service will be guaranteed. A risk mitigation mechanism to protect user experience in the IaaS model is offered by infrastructure SLAs (i.e., the SLAs formalizing capacity availability) signed between service provider and IaaS provider.

ENHANCING CLOUD COMPUTING ENVIRONMENTS USING A CLUSTER AS A SERVICE

Amazon Elastic Compute Cloud (EC2)

An IaaS cloud, EC2 offers “elastic” access to hardware resources that EC2 clients use to create virtual servers. Inside the virtual servers, clients either host the applications they wish to run or host services of their own to access over the Internet. As demand for the services inside the virtual machine rises, it is possible to create a duplicate (instance) of the virtual machine and distribute the load across the instances

Google App Engine

Google App Engine [5] is a PaaS cloud that provides a complete Web service environment: All required hardware, operating systems, and software are provided to clients. Thus, clients only have to focus on the installation or creation of their own services, while App Engine runs the services on Google's servers.

Microsoft Windows Azure

Another PaaS cloud, Microsoft's Azure [4] allows clients to build services using developer libraries which make use of communication, computational, and storage services in Azure and then simply upload the completed services.

To ease service-based development, Azure also provides a discovery service within the cloud itself. Called the .NET Service Bus [14], services hosted in Azure are published once and are locatable even if they are frequently moved. When a service is created/started, it publishes itself to the Bus using a URI [15] and then awaits requests from clients.

Salesforce: Salesforce [16] is a SaaS cloud that offers customer relations management (CRM) software as a service. Instead of maintaining hardware and software licenses, clients use the software hosted on Salesforce servers for a minimal fee. Clients of Salesforce use the software as though it is their own one and do not have to worry about software maintenance costs.

RVWS DESIGN

While Web services have simplified resource access and management, it is not possible to know if the resource(s) behind the Web service is (are) ready for requests. Clients need to exchange numerous messages with required Web services to learn the current activity of resources and thus face significant overhead loss if most of the Web services prove ineffective. Furthermore, even in ideal circumstances where all resources behind Web services are the best choice, clients still have to locate the services themselves.

SECURE DISTRIBUTED DATA STORAGE IN CLOUD COMPUTING

CLOUD STORAGE: FROM LANs TO WANs

Cloud computing has been viewed as the future of the IT industry. It will be a revolutionary change in computing services. Users will be allowed to purchase CPU cycles, memory utilities, and information storage services conveniently just like how we pay our monthly water and electricity bills. However, this image will not become realistic until some challenges have been addressed. In this section, we will briefly introduce the major difference brought by distributed data storage in cloud computing environment. Then, vulnerabilities in today's cloud computing platforms are analyzed and illustrated.

Moving From LANs to WANs

Most designs of distributed storage take the form of either storage area networks (SANs) or network-attached storage (NAS) on the LAN level, such as the networks of an enterprise, a campus, or an organization. SANs are constructed on top of block-addressed storage units connected through dedicated high-speed networks. In contrast, NAS is implemented by attaching specialized file servers to a TCP/IP network and providing a file-based interface to client machine [6]. For

SANs and NAS, the distributed storage nodes are managed by the same authority. The system administrator has control over each node, and essentially the security level of data is under control. The reliability of such systems is often achieved by redundancy, and the storage security is highly dependent on the security of the system against the attacks and intrusion from outsiders. The confidentiality and integrity of data are mostly achieved using robust cryptographic schemes.

TECHNOLOGIES FOR DATA SECURITY IN CLOUD COMPUTING

This section presents several technologies for data security and privacy in cloud computing. Focusing on the unique issues of the cloud data storage platform, this section does not repeat the normal approaches that provide confidentiality, integrity, and availability in distributed data storage applications. Instead, we select to illustrate the unique requirements for cloud computing data security from a few different perspectives:

Database Outsourcing and Query Integrity Assurance. Researchers have pointed out that storing data into and fetching data from devices and machines behind a cloud are essentially a novel form of database outsourcing. Section 8.3.1 introduces the technologies of Database Out- sourcing and Query Integrity Assurance on the clouding computing platform.

Data Integrity in Untrustworthy Storage: One of the main challenges that prevent end users from adopting cloud storage services is the fear of losing data or data corruption. It is critical to relieve the users' fear by providing technologies that enable users to check the integrity of their data. Section 8.3.2 presents two approaches that allow users to detect whether the data has been touched by unauthorized people.

Web-Application-Based Security: Once the dataset is stored remotely, a Web browser is one of the most convenient approaches that end users can use to access their data on remote services. In the era of cloud computing, Web security plays a more important role than ever. Section 8.3.3 discusses the most important concerns in Web security and analyzes a couple of widely used attacks.

Multimedia Data Security: With the development of high-speed network technologies and large bandwidth connections, more and more multi- media data are being stored and shared in cyber space. The security requirements for video, audio, pictures, or images are different from other applications. Section 8.3.4 introduces the requirements for multimedia data security in the cloud.

Data Integrity in Untrustworthy Storage

While the transparent cloud provides flexible utility of network-based resources, the fear of loss of control on their data is one of the major concerns that prevent end users from migrating to cloud storage services. Actually it is a potential risk that the storage infrastructure providers become self-interested, untrustworthy, or even malicious.

8.3.3 Web-Application-Based Security

Web security plays a more important role than ever. The Web site server is the first gate that guards the vast cloud resources. Since the cloud may operate continuously to process millions of dollars’ worth of daily on-line transactions, the impact of any Web security vulnerability will be amplified at the level of the whole cloud.

Web attack techniques are often referred as the class of attack. When any Web security vulnerability is identified, attacker will employ those techniques to take advantage of the security vulnerability. The types of attack can be categorized in Authentication, Authorization, Client-Side Attacks, Comm- and Execution, Information Disclosure, and Logical Attacks

Authentication: Authentication is the process of verifying a claim that a subject made to act on behalf of a given principal. Authentication attacks target a Web site’s method of validating the identity of a user, service, or application, including Brute Force, Insufficient Authentication, and Weak Password Recovery Validation. Brute Force attack employs an automated process to guess a person’s username and password by trial and error.

Authorization: Authorization is used to verify if an authenticated subject can perform a certain operation. Authentication must precede authorization. For example, only certain users are allowed to access specific content or functionality.

Client-Side Attacks: The Client-Side Attacks lure victims to click a link in a malicious Web page and then leverage the trust relationship expectations of the victim for the real Web site. In Content Spoofing, the malicious Web page can trick a user into typing user name and password and will then use this information to impersonate the user.

Command Execution: The Command Execution attacks exploit server-side vulnerabilities to execute remote commands on the Web site. Usually, users supply inputs to the Web-site to request services. If a Web application does not properly sanitize user-supplied input before using it within application code, an attacker could alter command execution on the server

Information Disclosure: The Information Disclosure attacks acquire sensi- tive information about a web site revealed by developer comments, error messages, or well-know file name conventions. For example, a Web server may return a list of files within a requested directory if the default file is not present. This will supply an attacker with necessary information to launch further attacks against the system. Other types of Information Disclosure includes using special paths such as "." and ".." for Path Traversal, or uncovering hidden URLs via Predictable Resource Location.

Logical Attacks: Logical Attacks involve the exploitation of a Web applica- tion's logic flow. Usually, a user's action is completed in a multi-step process. The procedural workflow of the process is called application logic. A common Logical Attack is Denial of Service (DoS). DoS attacks will attempt to consume all available resources in the Web server such as CPU, memory, disk space, and so on, by abusing the functionality provided by the Web site. When any one of any system resource reaches some utilization threshold, the Web site will no long be responsive to normal users.

8.3.4 Multimedia Data Security Storage

Multimedia Data Security plays an important role in the data storage to protect multimedia data. Recently, how storage multimedia contents are delivered by both different providers and users has attracted much attentions and many applications. This section briefly goes through the most critical topics in this area.

Protection from Unauthorized Replication: Contents replication is requi- red to generate and keep multiple copies of certain multimedia contents. For example, content distribution networks (CDNs) have been used to manage content distribution to large numbers of users, by keeping the replicas of the same contents on a group of geographically distributed surrogates .

Protection from Unauthorized Replacement: As the storage capacity is limited, a replacement process must be carried out when the capacity exceeds its limit. It means the situation that a currently stored content must be removed from the storage space in order to make space for the new coming content. However, how to decide which content should be removed is very important. If an unauthorized replacement happens, the content which the user doesn't want to delete will be removed resulting in an accident of the data loss.

Protection from Unauthorized Pre-fetching. The Pre-fetching is widely deployed in Multimedia Storage Network Systems between server databases and end users’ storage disks .