OpenNebula

Sándor Ács
acs.sandor@sztaki.mta.hu
http://www.lpds.sztaki.hu/CloudResearch

This presentation is heavily based on multiple presentations of the following people: Gábor Kecskeméti, Ignacio M. Llorente, Rubén S. Montero, Jaime Melis, Javier Fontán, Rafael Moreno
Outline

• Virtual Infrastructure Managers
• OpenNebula as a whole
• Architectural view on OpenNebula
• Constructing a Private cloud
• Virtual Machines in OpenNebula
• Constructing a Hybrid cloud
VIRTUAL INFRASTRUCTURE
MANAGERS
Why a Virtual Infrastructure Manager?

- VMs are great!!...but something more is needed
  - Where did/do I put my VM? (*scheduling & monitoring*)
  - How do I provision a new cluster node? (*clone & context*)
  - What MAC addresses are available? (*networking*)
- Provides a *uniform view* of the resource pool
- *Life-cycle management* and monitoring of VM
- The VIM *integrates* Image, Network and Virtualization
MTA SZTAKI LPDS

Extending the Benefits of Virtualization to Clusters

- Dynamic deployment and replacement of virtual machines on a pool of physical resources
- Transform a rigid distributed physical infrastructure into a flexible and agile virtual infrastructure

- Backend of Public Cloud: Internal management of the infrastructure
- Private Cloud: Virtualization of cluster or data-center for internal users
- Cloud Interoperation: On-demand access to public clouds
WHAT IS OPENNEBULA?
What is OpenNebula?

IaaS Cloud Computing Tool for Managing a Data Center's Virtual Infrastructure
- Open source Apache license
- Interoperable, based on standards
- Adaptable

Private Clouds
- Virtualize your on-premise infrastructure

Public Clouds
- Expose standard cloud interfaces

Hybrid Clouds
- Extend your private cloud with resources from a remote cloud provider

Ready for end-users
- Advanced user management
- CLI and Web Interface
What is OpenNebula?

Rigorously Tested, Matured Through Vibrant Community and Many Release Cycles

OpenNebula.org
- Develop & innovate
- Support the community
- Collaborate

Third party scalability tests: 16,000 VMs

2005
2008
2009
2010
2011
2012
2013
2014

TP v1.0 v1.2 v1.4 v2.0 v2.2 v3.0 v3.2 v3.4 v3.6 v3.8 v4.0

5,000 downloads/month

Research Project
The Benefits of OpenNebula

- **For the Infrastructure Manager**
  - Centralized management of VM workload and distributed infrastructures
  - Support for VM placement policies: balance of workload, server consolidation...
  - Dynamic resizing of the infrastructure
  - Dynamic partition and isolation of clusters
  - Dynamic scaling of private infrastructure to meet fluctuating demands
  - Lower infrastructure expenses combining local and remote Cloud resources

- **For the Infrastructure User**
  - Faster delivery and scalability of services
  - Support for heterogeneous execution environments
  - Full control of the lifecycle of virtualized services management
Interoperability from the Cloud Provider perspective

- Interoperable (platform independent), innovative (feature-rich) and proven (mature to run in production).
Widely Used to Build Enterprise Private Clouds in Medium and Large Data Centers

Survey Q2/Q3 2012 (2,500 users http://c12g.com/resources/survey/)

- Central or South America: 3%
- China, Japan or Korea: 14%
- Middle East: 1%
- North America: 23%
- India or Pakistan: 4%
- Europe or Russia: 49%
- Australia or South East Asia: 4%
- Africa: 1%

- Industry: 42%
- Other: 20%
- Research: 15%
- Academic: 5%
- Non-profit: 5%

- <10: 5%
- 10-100: 20%
- 100-500: 10%
- >500: 5%

Development/testing: 58%
The Benefits for System Integrators

• Fits into any existing data center, due to its open, flexible and extensible interfaces, architecture and components
• Builds any type of Cloud deployment
• Open source software, Apache license
## Comparison with Similar Technologies

<table>
<thead>
<tr>
<th>Feature</th>
<th>VMware Vsphere</th>
<th>Eucalyptus</th>
<th>Nimbus</th>
<th>OpenStack</th>
<th>OpenNebula</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Virtualization Management</strong></td>
<td>VMware</td>
<td>Xen, KVM</td>
<td>Xen</td>
<td>Xen, KVM, VMware</td>
<td>Xen, KVM, VMware</td>
</tr>
<tr>
<td><strong>Virtual Network Management</strong></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Image Management</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Service Contextualization</strong></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Scheduling</strong></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Administration Interface</strong></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Hybrid Cloud Computing</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Cloud Interfaces</strong></td>
<td>vCloud</td>
<td>EC2*</td>
<td>WSRF, EC2*</td>
<td>EC2*, OGF OCCI</td>
<td>EC2*, OGF OCCI</td>
</tr>
<tr>
<td><strong>Flexibility and Extensibility</strong></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Open Source</strong></td>
<td>No</td>
<td>GPL</td>
<td>Apache</td>
<td>Apache</td>
<td>Apache</td>
</tr>
</tbody>
</table>
INSIDE OPENNEBULA
Front-end
- X509
- LDAP
- ssh keys
- ACL
- permissions,
- Groups
- Accounting

Service Networks
- SSH, Ganglia/Nagios
- Additional monitor agents

VM Instance Networks
- VLAN per user (layer2)
- Open vSwitch, 802.1q
- Ebtables

Hosts
- VMware
- Xen
- KVM
- ...

Datastores
- DFS: NFS, Ceph, GlusterFS…
- SAN: Fibre Channel, iSCSI, LVM…
- SSH
Internal OpenNebula Architecture - The Cloud Integrator Perspective

Design Principles

- Modularity
- Lightness
- Openness

OpenNebula core

CLI
GUI
Cloud Servers
OCA (Ruby, Java)
XML-RPC API

Scheduler

Monitoring
Storage
Network
DB
Virtualization
Images
Auth
Sysadmin-centric Approach - The Cloud Integrator Perspective

OpenNebula core

OCA (Ruby, Java)

XML-RPC API

CLI, GUI, Cloud Servers, Scheduler

Monitoring, Storage, Network, Virtualization, Images, Auth

DB

Easy to adapt
Easy to create new ones

OpenNebula drivers

- Small script for each action, written in any language
- Simple interaction done through arguments, std/err output, exit code
- Different drivers can co-exist in heterogeneous environments
- Can be executed locally or in the remote Host
- The Host monitorization updates the remote driver directory
How to Develop Drivers - The Cloud Integrator Perspective

An example: the migrate script

- Each script performs a small, synchronous task
- Helper scripts provide commonly-used functions for log, ssh execution, error reporting, etc.

```bash
#!/bin/bash
source $(dirname $0)/xenrc
source $(dirname $0)/../../scripts_common.sh
deploy_id=$1
dest_host=$2
eexec_and_log "$XM_MIGRATE $deploy_id $dest_host"
  "Could not migrate $deploy_id to $dest_host"
```
How to Interact with OpenNebula - The Cloud Integrator Perspective

**XML-RPC**
- Simple, fast
- Works in any language
- OCA (OpenNebula Cloud API)
- High level bindings
- Complete functionality
- Ruby, Java, Python

OpenNebula distribution

**OpenNebula**

Sunstone

Cloud APIs

- Occi
- OCCI
- OVF
- DMTF
- SNIA
- CDMI

Community Contributions

- Chef
- Puppet Labs
- JClouds
- Libcloud
How to Interact with OpenNebula - The Cloud Integrator Perspective

OCA Ruby Example:

Shutdown all my Virtual Machines

```ruby
#!/usr/bin/env ruby

require 'OpenNebula'

CREDENTIALS = "oneuser:onepass"
ENDPOINT = "http://localhost:2633/RPC2"

client = OpenNebula::Client.new(CREDENTIALS, ENDPOINT)

vm_pool = VirtualMachinePool.new(client, OpenNebula::Pool::INFO_MINE)

rc = vm_pool.info
if OpenNebula.is_error?(rc)
  puts rc.message
  exit -1
end

vm_pool.each do |vm|
  rc = vm.shutdown
  if OpenNebula.is_error?(rc)
    puts "Virtual Machine #{vm.id}: #{rc.message}"
  else
    puts "Virtual Machine #{vm.id}: Shutting down"
  end
end

exit 0
```
Clusters

- Pools of hosts that share datastores and networks
- Used for load balancing, high availability, and high performance computing

Multiple Datastores per Cluster

- Balance I/O operations between storage servers
- Define different SLA policies (e.g. backup) and performance features for different VM types or users
How are the images transferred to the hosts?

OpenNebula includes 6 different ways to distribute datastore images to the hosts:

- **Shared**, the datastore is exported in a shared filesystem to the hosts.
- **SSH**, datastore images are copied to the remote hosts using the ssh protocol.
- **iSCSI**, hosts access datastore targets opening and closing iSCSI sessions dynamically.
- **VMFS**, image copies are done using the vmkfstools (VMware filesystem tools).
- **QCOW**, a driver specialized to handle qemu-qcow format and take advantage of its snapshotting capabilities.
- **CEPH**, a driver that deletes to libvirt/KVM the management of Ceph RBDs.
**System**, to hold images for running VMs, depending on the storage technology used these temporal images can be complete copies of the original **image**, **qcow deltas** or **simple filesystem links**.

**File-system**, to store disk images in a **file form**. The files are stored in a directory mounted from a SAN/NAS server.

**iSCSI/LVM**, to store disk images in a **block device form**. Images are presented to the hosts as iSCSI targets.

**VMware**, a datastore specialized for the **VMware** hypervisor that handles the **vmdk format**.

**vmfs**, a datastore specialized in **VMFS format** to be used with VMware hypervisors. Cannot be mounted in the OpenNebula front-end since VMFS is not *nix compatible.

**Ceph**, to store disk images using **Ceph block devices**.

**Files**, This is a special datastore used to **store plain files** and not disk images. The plain files can be used as **kernels**, **ramdisks** or **context files**.
- OpenNebula management operations use ssh connections.
- **Image traffic**, may require the movement of heavy files (VM images, checkpoints). Dedicated storage links may be a good idea.
- **VM demands**, consider the typical requirements of your VMs. Several NICs to support the VM traffic may be a good idea.
- OpenNebula relies on bridge networking for the VMs.
Example network setup in a private cloud
Preparing VMs for OpenNebula

• You can use any VM image prepared for the target hypervisor

• **Hint I:** Use contextualisation.

• **Hint II:** Do not pack useless information in the images

• **Hint III:** Install once and deploy many; prepare master images

• **Hint IV:** Do not put private information (e.g. ssh keys) in the master images, use the contextualisation
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>• Name that the VM will get for description purposes.</td>
</tr>
<tr>
<td>CPU</td>
<td>• Percentage of CPU divided by 100 required for the Virtual Machine.</td>
</tr>
<tr>
<td>Memory</td>
<td>• Required amount of memory</td>
</tr>
<tr>
<td>DISK (SOURCE, TARGET, CLONE, TYPE)</td>
<td>• Description of a disk image to attach to the VM.</td>
</tr>
<tr>
<td>NIC (NETWORK)</td>
<td>• Definition of a virtual network the VM will be attached to.</td>
</tr>
</tbody>
</table>

- Multiple disk and network interfaces can be specified just adding more disk/nic statements.
- To create swap images you can specify TYPE=swap, SIZE=<size in MB>.
- By default disk images are cloned, if you do not want that to happen CLONE=no can be specified and the VM will attach the original image.
VM States overview

VMs

- Pending
- Prolog
- Boot
- Running
- Shutdown
- EpiLog
- Done
- Suspended
- Stopped
- Hold
- Migrate

MTA SZTAKI LPDS
CLOUD RESEARCH
Pending state

• After submitting a VM description to ONE it is added to the database and its state is set to PENDING.
• In this state IP and MAC addresses are also chosen if they are not explicitly defined.
• The scheduler awakes every 30 seconds and looks for VM descriptions in PENDING state and searches for a physical node that meets its requirements. Then a deploy XML-RPC message is sent to oned to make it run in the selected node.
• In PROLOG state the Transfer Driver prepares the images to be used by the VM.

• Transfer actions:
  – **CLONE**: Makes a copy of a disk image file to be used by the VM. If Clone option for that file is set to false and the Transfer Driver is configured for NFS then a symbolic link is created.
  – **MKSING**: Creates a swap disk image on the fly to be used by the VM if it is specified in the VM description.
• In this state a deployment file specific for the virtualization technology configured for the physical host is generated using the information provided in the VM description file. Then Virtual Machine Driver sends deploy command to the virtual host to start the VM.

• The VM will be in this state until deployment finishes or fails.
Contextualization

- Requirements against the VM:
  - Should be prepared to use the contextualization ISO image.
  - Should mount the ISO image at boot time.
  - After boot it should use the scripts on the ISO image to make use of the information provided.
While the VM is in RUNNING state it will be periodically polled to get its consumption and state.

In SHUTDOWN state Virtual Machine Driver will send the shutdown command to the underlying virtual infrastructure.
In EPILOG state the Transfer Manager Driver is called again to perform this actions:

- Copy back the images that have `SAVE=yes` option.
- Delete images that were cloned or generated by `MKSWAP`.
HYBRID CLOUD
Overview

- VMs can be local or remote
- VM connectivity has to be configured, usually VPNs

- External Clouds are like any other host
- Placement constraints
Using an EC2 hybrid cloud

- Virtual Machines can be instantiated locally or in EC2
- The VM template must provide a description for both instantiation methods.
- The EC2 counterpart of your VM (AMI_ID) must be available for the driver account
- The EC2 VM template attribute should describe not only the VM’s properties but the contact details of the external cloud provider
Hybrid cloud Use Cases

On-demand Scaling of Computing Clusters
- Elastic execution of a SGE computing cluster
- Dynamic growth of the number of worker nodes to meet demands using EC2
- Private network with NIS and NFS
- EC2 worker nodes connect via VPN

On-demand Scaling of Web Servers
- Elastic execution of the NGinx web server
- The capacity of the elastic web application can be dynamically increased or decreased by adding or removing NGinx instances
Questions?

Thank you for the attention!

https://www.lpds.sztaki.hu/CloudResearch