Virtualization deployments
A discussion of various challenges in virtualized environments

Iustin Pop
Ganeti team
Google Switzerland
Overview

• Introduction
• Abstract model
• Scaling the machine pool size
  ▪ Up to 10
  ▪ Up to 100
  ▪ Up to 1,000
• VM types
• Hardware evolution
• Ganeti in Google
Introduction

Use cases of virtualization:

• traditional:
  ▪ improved resource utilization
  ▪ consolidate low-usage machines

• but the technology has matured:
  ▪ focus moves to higher layers (management)
  ▪ large numbers of both physical and virtual machine
  ▪ more uses (VM types)

Disclaimer:

• the presentation content is not representative of Google's usage of virtualization

• the presentation solely refers to the use of virtualization in Google for internal, corporate purposes and not external services or products (e.g. www.google.com)
Abstract model

- **monitoring**
- **access controls**
- **machine lifecycle**

**resource mgmt.**
- **base OS**
- **hardware**

**single-app**

- **application layer**
- **integration layer**
- **resource mgmt.**

- **cluster**
- **resource mgmt.**
- **machine resource mgmt.**

- **hypervisor**
- **base OS**
- **hardware**

**multiple app** or virtualized
Scaling the number of machines

The machine pool size influences the ROI for different features

• at lower sizes, efficiency at lower levels is most important
• growing the number of machines increases the importance of the management layer
• HW failure rate importance varies
• automation cost is more-or-less constant, but benefits vary greatly
Up to 10 physical machines

Main characteristics:

• Hypervisor and HW efficiency is paramount
• Small number of machines translates into:
  ▪ small customer base
  ▪ low number of HW failures
  ▪ reduced benefit of automation
  ▪ greater chance of same HW profile

Challenges:

• application compatibility
Up to 100 physical machines

Similar to non-virtualized environments:

• component failure is rare, but part of normal life
• automation benefits start to show (deployment, configuration, etc)

Specific to virtual environments:

• diverse configurations and machine mobility mean VMs will be shifted around and their HW profile can change dynamically
• cost savings are split between resource utilization and operational gains

Challenges:

• accommodating various customer types
• integration of VM and non-VM environments
Up to 1,000 physical machines

Number of machines affect cost profile:

- HW costs are linear, but operations costs no longer

- Automation and standardization across all layers have big benefits
  - the hypervisor and HW layers can be deeply abstracted by the management tools
  - automation of all procedures is paramount to keeping the VM environment healthy

Challenges:

- multiple customers, same management toolset
- software upgrades for physical machines
- dealing with multiple HW generations
VM types

Deployments will have multiple VM types:

• server/desktop/lab/etc.
• central administration versus end-user administration
• integration of all these into the same machine pool
VM types - server

• stable, long-life
• monitoring is important
• resource usage has smoother variation
• usually continuous operation
VM types - desktop

• less stable life
• bursty usage
• 'business hours' type of operation
• GUI/user friendly interface to VM-specific operations very important
• monitoring integration good for debugging, but less for the big mass of end-users
VM types - lab

• short lived
• end-user provisioning out of own resource pool
• monitoring less useful than quick provisioning
• snapshot, rollback and similar features important
Keeping pace with hardware

Commodity hardware:

- rapidly changing specifications:
  - cores
  - memory
  - disk size
- and not so rapid:
  - network/disk bandwidth
  - single-CPU speed

This results in asymmetric growth:

- resource allocation needs to handle this
- cluster architecture changes over time
Design goals and principles

• sits between hardware layer and integration layer
• fully automate node- and cluster-level resource management
• needs base OS and hypervisor
• does not provide monitoring, access controls or global provisioning (across clusters)

• principles:
  ▪ not dependent on specific hardware (e.g. external shared storage)
  ▪ scales (almost) linearly with the number of systems
Questions & Answers