CS528
Virtualization in Cloud

A. Sahu
Dept of CSE, IIT Guwahati

http://www.iitg.ac.in/asahu/cs528-2017e/
Outline

• What is Cloud Computing?
• (HPC, Data Center, Grid) Vs Cloud
• **Virtualization**
  • Power Consumption Model
  • Prediction Aware Scheduling of Cloud
• Cost Aware Scheduling, Reliability Aware
Cloud computing takes virtualization to the next step

• You don’t have to own the hardware
• You “rent” it as needed from a cloud
• There are public clouds
  – e.g. Amazon EC2, and now many others (Microsoft, IBM, Sun, and others ...)
• A company can create a private one
  – With more control over security, etc.
Virtualization

- Abstraction of computer resources.
- Virtualization hides the physical characteristics of computing resources
  - From their users, be they applications, or end users.
Virtualization

• Virtualization includes making a **single** physical resource
  – such as a server, an operating system, an application, or storage device
  – appear to function as **multiple** virtual resources

• Also include making **multiple** physical resources
  – such as storage devices or servers
  – appear as a **single** virtual resource
Virtualization Basic: Truck on Train
Virtualization Basic

• In OS, Classic example `FILE` as abstract virtual object
• File read/write:
  – fwrite: write data to File
  – fread: read data from File
• Underlying target `File` may be in HDD, Buffer, SSD, Network File, CDROM
• Internal may be diff but externally the same call
Virtualization Basic

• Example: Virtual BOX, QEMU, Wine, Dalvik, JVM
• QEMU: ARM emulation on X86
• JVM and Dalvik: Java byte code and in Andriod
• Oracle Virtual BOX
  – Running MS Window OS on Linux Host
  – Running Linux on MS Window Host
• Cygwin: Running Linux App on Window
  – Assume running shell script and GIMP in Window
• Wine
  – Running MS Window app on Linux, Running your favorite MS-Office in Linux
Virtualization Basic

• In OS, Classic example **FILE** as abstract virtual object

• File read/write:
  – fwrite: write data to File
  – fread: read data from File

• Underlying target **File** may be in HDD, Buffer, SSD, Network File, CDROM

• Internal may be diff but externally the same call
Virtual Machine

• What is Virtual Machine (VM)?
  – **VM** is a software implementation of a machine (i.e. a computer) that executes programs like a real machine.

• Terminology:
  – Host (Target) : The primary environment where will be the target of virtualization.
  – Guest (Source) : The virtualized environment where will be the source of virtualization.
The Use of Computers
Virtualization

- Applications
- Operating System
- Hypervisor
- Hardware
Virtualization -- a Server for Multiple Applications/OS
Virtualization -- a Server for Multiple Applications/OS

• **Hypervisor** is a software program
  – that manages multiple operating systems (or multiple instances of the same operating system)
  – on a single computer system.

• The hypervisor manages the system's
  – processor, memory, and other resources to allocate what each operating system requires.

• Hypervisors are designed for a particular processor architecture
  – and may also be called **virtualization managers**.
Capacity Utilization

- Virtualized system (high)
  - High utilized*
  - Low utilized
- Stand alone system (low)
  * But not overloaded…
Logical view of Cloud System

- $t_1$, $t_2$, $t_n$: Tasks
- $v_{t1}$, $v_{t2}$, $v_{t_k}$: VM types
- $k$ VM types
- $m$ PMs: PM$_1$, PM$_2$, PM$_3$, PM$_m$
- $n$ Tasks

Cloud System
Why now?

• 1960—1999
  – IBM, CP-40, CP/CMS, S/360-370, VM370, Virtual PC, VMware

• 2000—2005
  – IBM z/VM, Xen

• 2006
  – Intel VT-x
  – AMD’s AMD-V

• 2008—
Hardware evolution

• Faster CPU clock than ever
  – Though almost hit its top

• More CPU cores in a single chip
  – 4-core CPUs already in the market
  – 6- or 8-core CPUs will be there soon

• Multi-core architectures make parallel processing more realizable

• Virtualization support on chip from CPU manufacturers (e.g., Intel, AMD)
Software maturity

• More than one credible player in the market
• Available and stable open-sourced software
  – OS, DB, Web server, Java, PHP, gcc, etc.
• Established and mature software standards
  – Web service, XML, SOAP, COM, etc.
Virtualization

- Binary translation is the most established technology for full virtualization
- Hardware assist is the future of virtualization, but it still has a long way to go
- Paravirtualization delivers performance benefits with maintenance costs
  - Xen
  - VMWare
Issues in Virtualization for Cloud-Computing

• Aspects and expectation from
  – End-user
  – Operator/Manager
Issues in Virtualization for Cloud-Computing

- Virtualization implemented on
  - a single machine (with multi-core CPUs)
  - a cluster of machines (with multi-core CPUs)
- The state-of-the-art
  - Running a Xen or a cluster of Xens
Issues in Virtualization for Cloud-Computing

• Abiquo/abicloud may provide partial solutions
Running multiple OS and applications

- Virtualization: One physical hardware can run multiple OS and applications through a hypervisor.
- A hypervisor is the virtualization manager on a physical hardware.
User View of Virtualization

Virtualization Layer - Optimize HW utilization, power, etc.
Cloud Computing

• Features of Clouds
  – Scalable, Enhanced Quality of Service (QoS)
  – Specialized and Customized, Cost Effective
  – Simplified User Interface
Virtualization in Five Abstraction Levels

- **Application Level**: JVM/.NET CLR/Panot
- **Library/API Level**: WINE/LXRun/vCuda
- **OS Level**: Jail/Virtual Environment /FVM
- **H/W Abst Layer (HAL) Level**: Vmware/Xen/L4/Virtual PC/Virtual Box
- **ISA Level**: Vovhs/QEMU/BIRD/Dynamo
Emulation vs. Virtualization

• **Emulation technique**
  – Simulate an independent environment where guest ISA and host ISA are different.
  – Example: Emulate x86 architecture on ARM platform.

• **Virtualization technique**
  – Simulate an independent environment where guest ISA and host ISA are the same.
  – Example: Virtualize x86 architecture to multiple instances.
Virtualization at ISA (Instruction Set Architecture) level

• With the help of ISA emulation
  – Example: MIPS binary code can run on an x-86 host
  – Typical systems: Bochs, Crusoe, Quemu, BIRD, Dynamo, Simic/Gems

• Advantage
  – It can run a large amount of legacy binary codes written for various processors on any given new hardware host machines
  – best application flexibility

• Shortcoming & limitation
  – One source instruction may require 10-100 of target instructions to perform its function, which is relatively slow.
Virtualization at Hardware Abstraction level

- Generates virtual hardware environments for VMs,
  - And manages the underlying hardware through virtualization.
  - Typical systems: VMware, Virtual PC, Xen, Virtual Box

- Advantage:
  - higher performance and good application isolation

- Shortcoming & limitation:
  - Very expensive to implement (complexity)
Virtualization at Operating System (OS) level

- This virtualization creates isolated containers on a single physical server and the OS-instance to utilize the hardware and software in datacenters.
  - Typical systems: Jail / Virtual Environment / FVM/
  - Docker/Container/Kubernet

- Advantage
  - Has minimal startup/shutdown cost, low resource requirement, and high scalability; synchronize VM and host state changes.

- Shortcoming & limitation:
  - All VMs at the operating system level must have the same kind of guest OS
  - Poor application flexibility and isolation.
Virtual Machine Monitor

• What’s Virtual Machine Monitor (VMM) ?
  – **VMM** or **Hypervisor** is the software layer providing the virtualization.

• System architecture :

```
       App       App       App
       OS        OS        OS
     Operating System
     Hardware

Traditional Stack

             App
       Hypervisor
             OS
     Hardware

Virtualized Stack
```
Virtualization Types

• Virtualization Types:
  – Type 1 – Bare metal
    • VMMs run directly on the host's hardware as a hardware control and guest operating system monitor.
  – Type 2 – Hosted
    • VMMs are software applications running within a conventional operating system.
Virtualization Approaches

- Virtualization Approaches:
  - Full-Virtualization
    - VMM simulates enough hardware to allow an unmodified guest OS.
  - Para-Virtualization
    - VMM does not necessarily simulate hardware, but instead offers a special API that can only be used by the modified guest OS.
Virtualization Approaches

• Full-Virtualization

<table>
<thead>
<tr>
<th>Pros</th>
<th>Need not to modify guest OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cons</td>
<td>Significant performance hit</td>
</tr>
</tbody>
</table>
Virtualization Approaches

• Para-Virtualization

<table>
<thead>
<tr>
<th>Pros</th>
<th>Light weight and high performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cons</td>
<td>Require modification of guest OS</td>
</tr>
</tbody>
</table>

Pros:
- Light weight and high performance

Cons:
- Require modification of guest OS
Examples

Xen
• Type 1 Virtualization
• Para-Virtualization

KVM
• Type 2 Virtualization
• Full-Virtualization