CS370 Operating Systems
Colorado State University
Yashwant K Malaiya
Spring 2018 Lecture 27

Virtualization

Slides based on
• Some on text by Silberschatz, Galvin, Gagne
• Mostly: Various sources
Project Notes

• 5/1: Tu Project final report
• 5/1: Research Poster session 9:15 to 11:15 AM (electronic copy due 5 PM.)
  – Presenters: all others: A-L: 9:30-10:05  M-Z: 10:10-10:45
• 5/2 Project Slides for both options (3-4): Piazza
• 5/1-5/3: Planned Development demos
• 5/7 Project reviews using form provided
  – 1 Dev, 1 Res, also peer (using form)
• 5/10 Thurs Final
Implementation of VMMs

- **Type 1 hypervisors** - Operating-system-like software built to provide virtualization. Runs on ‘bare metal’.
  - Including VMware ESX, Joyent SmartOS, and Citrix XenServer

- Also includes general-purpose operating systems that provide standard functions as well as VMM functions
  - Including Microsoft Windows Server with HyperV and RedHat Linux with KVM

- **Type 2 hypervisors** - Applications that run on standard operating systems but provide VMM features to guest operating systems
  - Including VMware Workstation and Fusion, Parallels Desktop, and Oracle VirtualBox
Implementation of VMMs

https://microkerneldude.files.wordpress.com/2012/01/type1-vs-2.png
Building Block – Trap and Emulate

• VM needs two modes: both in real user mode
  – virtual user mode and virtual kernel mode
• When Guest OS attempts to execute a privileged instruction, what happens?
  – Causes a trap
  – VMM gains control, analyzes error, executes operation as attempted by guest
  – Returns control to guest in user mode
  – Known as trap-and-emulate
• This was the technique used for implementing floating point instructions in CPUs without floating point coprocessor
Memory Management

Memory mapping:

- On a bare metal machine: Each process has its own virtual address space. OS uses page table/TLB to map Virtual page number (VPN) to Physical page number (PPN) (physical memory is shared). Each process has its own page table/TLB.

VPN -> PPN
Memory Management

Memory mapping:

• On a bare metal machine:
  – VPN -> PPN

• VMM: Real physical memory (machine memory) is shared by the OSs. Need to map PPN of each VM to MPN (Shadow page table)
  – PPN -> MPN

• Where is this done?
  – In Full virtualization?
  – In Para virtualization?
Memory Management

• VMM: Real physical memory (*machine memory*) is shared by the OSs. Need to map PPN of each VM to MPN (Shadow page table)

  \[ \text{PPN} \rightarrow \text{MPN} \]

• Where is this done?
  – In Full virtualization? Has to be done by hypervisor. Guest OS knows nothing about MPN.
  – In Para virtualization? May be done by guest OS. It knows about hardware. Commands to VMM are “hypercalls”

• **Full virtualization:** PT/TLB updates are trapped to VMM. It needs to do \( \text{VPN}\rightarrow\text{PPN} \rightarrow \text{MPN} \). It can do \( \text{VPN}\rightarrow\text{MPN} \) directly (VMware ESX)
Handling memory oversubscription

Oversubscription solutions:

- Deduplication by VMM determining if same page loaded more than once, memory mapping the same page into multiple guests
- Double-paging, the guest page table indicates a page is in a physical frame but the VMM moves some of those to disk.
- Install a pseudo-device driver in each guest (it looks like a device driver to the guest kernel but really just adds kernel-mode code to the guest)
  - Balloon memory manager communicates with VMM and is told to allocate or deallocate memory to decrease or increase physical memory use of guest, causing guest OS to free or have more memory available.
Live Migration

Running guest can be moved between systems, without interrupting user access to the guest or its apps
  – for resource management,
  – maintenance downtime windows, etc

• Migration from source VMM to target VMM
  – Needs to migrate all pages gradually, without interrupting execution (details in next slide)
  – Eventually source VMM freezes guest, sends vCPU’s final state, sends other state details, and tells target to start running the guest
  – Once target acknowledges that guest running, source terminates guest
Live Migration

• Migration from source VMM to target VMM
  – Source VMM establishes a connection with the target VMM
  – Target creates a new guest by creating a new VCPU, etc
  – Source sends all read-only memory pages to target
  – Source starts sending all read-write pages to the target, marking them as clean
    • repeats, as during that step some pages were modified by the guest and are now dirty.
  – Source VMM freezes guest, sends VCPU’s final state, other state details, final dirty pages, and tells target to start running the guest
    • Once target acknowledges that guest running, source terminates guest
VIRTUAL APPLIANCES: “shrink-wrapped” virtual machines

• Developer can construct a virtual machine with
  – required OS, compiler, libraries, and application code
  – Freeze them as a unit ... ready to run
• Customers get a complete working package
• Virtual appliances: “shrink-wrapped” virtual machines
• Amazon’s EC2 cloud offers many pre-packaged virtual appliances examples of Software as a service
CS370 Operating Systems
Colorado State University
Yashwant K Malaiya
Spring 2018 Lecture 27

Data Centers & Cloud Computing

Slides based on
- Text by Silberschatz, Galvin, Gagne
- Various sources
Data Centers

• Large server and storage farms
  – 1000s-100,000 of servers
  – Many PBs of data

• Used by
  – Enterprises for server applications
  – Internet companies
  – Some of the biggest DCs are owned by Google, Facebook, etc

• Used for
  – Data processing
  – Web sites
  – Business apps
Data Center architecture

Traditional - static
- Applications run on physical servers
- System administrators monitor and manually manage servers
- Storage Array Networks (SAN) or Network Attached Storage (NAS) to hold data

Modern – dynamic with larger scale
- Run applications inside virtual machines
- Flexible mapping from virtual to physical resources
- Increased automation, larger scale
Data Center architecture

Giant warehouses with:

- Racks of servers
- Storage arrays
- Cooling infrastructure
- Power converters
- Backup generators

Or with containers

- Each container filled with thousands of servers
- Can easily add new containers
- “Plug and play”
- Pre-assembled, cheaper, easily expanded
Server Virtualization

Allows a server to be “sliced” into Virtual Machines

- VM has own OS/applications
- Rapidly adjust resource allocations
- VM migration within a LAN

• Virtual Servers
  - Consolidate servers
  - Faster deployment
  - Easier maintenance

• Virtual Desktops
  - Host employee desktops in VMs
  - Remote access with thin clients
  - Desktop is available anywhere
  - Easier to manage and maintain
Data Center Challenges

Resource management
- How to efficiently use server and storage resources?
- Many apps have variable, unpredictable workloads
- Want high performance and low cost
- Automated resource management
- Performance profiling and prediction

Energy Efficiency
- Servers consume huge amounts of energy
- Want to be “green”
- Want to save money
Data Center Challenges

Efficiency captured as *Power Usage Effectiveness*
- Ratio of IT Power / Total Power
- typical: 1.7, Google PUE ~ 1.1)

Larger data centers can be cheaper to buy and run than smaller ones

- Lower prices for buying equipment in bulk
- Cheaper energy rates
- Automation allows small number of sys admins to manage thousands of servers
- General trend is towards larger mega data centers
- 100,000s of servers
- Has helped grow the popularity of cloud computing
## Economy of Scale

<table>
<thead>
<tr>
<th>Resource</th>
<th>Cost in Medium DC</th>
<th>Cost in Very Large DC</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU cycle cost</td>
<td>2 picocents</td>
<td>&lt; 0.5 picocents</td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td>$95 / Mbps / month</td>
<td>$13 / Mbps / month</td>
<td>7.1x</td>
</tr>
<tr>
<td>Storage</td>
<td>$2.20 / GB / month</td>
<td>$0.40 / GB / month</td>
<td>5.7x</td>
</tr>
<tr>
<td>Administration</td>
<td>≈140 servers/admin</td>
<td>&gt;1000 servers/admin</td>
<td>7.1x</td>
</tr>
</tbody>
</table>
Reliability Challenges

Typical failures in a year of a Google data center:

• 20 rack failures (40-80 machines instantly disappear, 1-6 hours to get back)
• 3 router failures (have to immediately pull traffic for an hour)
• 1000 individual machine failures
• thousands of hard drive failures

• Pay by use instead of provisioning for peak

Static data center

Data center in the cloud

Unused resources
User has a variable need for capacity

- Underprovisioning
- Provisioning for peak
- Cloud has elastic capacity (i.e. way more than what the user needs)
- User can get exactly the capacity from the Cloud that is actually needed

- Why does this work for the provider?
  - Varying demand is statistically smoothed out over (very) many users
Amazon EC2 Instance types

On-Demand instances
• Users that prefer the low cost and flexibility of Amazon EC2 without any up-front payment or long-term commitment
• Applications with short-term, spiky, or unpredictable workloads that cannot be interrupted

Spot Instances
• request spare Amazon EC2 computing capacity for up to 90% off
• Applications that have flexible start and end times

Reserved Instances
• Applications with steady state usage
• Applications that may require reserved capacity

Dedicated Hosts
• physical EC2 server dedicated for your use.
• server-bound software licenses, or meet compliance requirements
Amazon EC2 Prices (samples from their site)

General Purpose - Current Generation  Region: US East (Ohio)

<table>
<thead>
<tr>
<th>Instance</th>
<th>vCPU</th>
<th>ECU</th>
<th>Memory (GiB)</th>
<th>Instance Storage (GB)</th>
<th>Linux/UNIX Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>t2.nano</td>
<td>1</td>
<td>Variable</td>
<td>0.5</td>
<td>EBS Only</td>
<td>$0.0058 per Hour</td>
</tr>
<tr>
<td>t2.small</td>
<td>1</td>
<td>Variable</td>
<td>2</td>
<td>EBS Only</td>
<td>$0.023 per Hour</td>
</tr>
<tr>
<td>t2.medium</td>
<td>2</td>
<td>Variable</td>
<td>4</td>
<td>EBS Only</td>
<td>$0.0464 per Hour</td>
</tr>
<tr>
<td>m5.4xlarge</td>
<td>16</td>
<td>61</td>
<td>64</td>
<td>EBS Only</td>
<td>$0.768 per Hour</td>
</tr>
<tr>
<td>m4.16xlarge</td>
<td>64</td>
<td>188</td>
<td>256</td>
<td>EBS Only</td>
<td>$3.2 per Hour</td>
</tr>
</tbody>
</table>
The Service Models

Service models

• **IaaS: Infrastructure as a Service**
  – infrastructure components traditionally present in an on-premises data center, including servers, storage and networking hardware
  – e.g., Amazon EC2, Microsoft Azure, Google Compute Engine

• **PaaS: Platform as a Service**
  – supplies a environment on which users can install applications and data sets
  – e.g., Google AppEngine, Heroku, Apache Stratos

• **SaaS: Software as a Service**
  – a software distribution model with provider hosted applications
  – Microsoft Office365, Amazon DynamoDB, gmail
Cloud Management models

• **Public clouds**
  - Utility model
  - Shared hardware, no control of hardware,
  - Self-managed (e.g., AWS, Azure)

• **Private clouds:**
  - More isolated (secure?)
  - Federal compliance friendly
  - Customizable hardware and hardware sharing

• **Hybrid clouds:**
  - a mix of on-premises, private cloud and third-party, public cloud services.
  - Allows workloads to move between private and public clouds as computing needs and costs change.
Different Regions to Achieve HA

- AWS datacenters is divided into regions and zones, that aid in achieving availability and disaster recovery capability.
- Provide option to create point-in-time snapshots to back up and restore data to achieve DR capabilities.
- The snapshot copy feature allows you to copy data to a different AWS region.
  - This is very helpful if your current region is unreachable or there is a need to create an instance in another region
  - You can then make your application highly available by setting the failover to another region.
Different Regions to Achieve HA