CS370 Operating Systems
Colorado State University
Yashwant K Malaiya
Spring 2020 L26
Virtualization

Slides based on
- Text by Silberschatz, Galvin, Gagne
- Various sources
Virtualization:

• How to allocate vCPUs to virtual machines?
  – Guidelines/experience. pCPU:vCPU is often 1:1 to 1:3 or more. A VM may not use a CPU all the time.

• How much memory should be allocated per VM? Depends. 0.5-1GB min?

• Running a VM inside a VM?
  • Possible for modern CPUs. Restrictions possible for older.

• How can V machines run concurrently? How can two OSs run at the same time? Just like processes..

Kernel vs OS: OS = {Kernel, UI, libraries/binaries}
Wish things were back to normal ..
Implementation of VMMs

https://microkerneldude.files.wordpress.com/2012/01/type1-vs-2.png
Memory mapping:

- On a bare metal machine: 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

- 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
• Linux containers (LXC) are “lightweight” VMs

• Comparison between LXC/docker and VM

• Containers provide “OS-level Virtualization” vs “hardware level”.

• Containers can be deployed in seconds.

• Very little overhead during execution, just like Type 1.
Microservices

eShopOnContainers reference application
(Development environment architecture)

CS370 Operating Systems
Colorado State University
Yashwant K Malaiya

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)

Economy of Scale

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>
Data Center Challenges

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

User has a variable need for capacity. User can choose among

**Fixed resources: Private data center**
- Under-provisioning when demand is too high, or
- Provisioning for peak

**Variable resources:**
- Use more or less depending on demand
- Public 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 many users, their peaks may occur at different times
- Prices set low for low overall demand periods
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 (cheap)
• request spare Amazon EC2 computing capacity for up to 90% off
• Applications that have flexible start and end times

Reserved Instances (expensive)
• 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>

ECU = EC2 Compute Unit (perf), EBS: elastic block store (storage), automatically replicated
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 an 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
# The Service Models

<table>
<thead>
<tr>
<th>On-Premises</th>
<th>IaaS: Infrastructure as a Service</th>
<th>PaaS: Platform as a Service</th>
<th>SaaS: Software as a Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications</td>
<td>Applications</td>
<td>Applications</td>
<td>Applications</td>
</tr>
<tr>
<td>Data</td>
<td>Data</td>
<td>Data</td>
<td>Data</td>
</tr>
<tr>
<td>Runtime</td>
<td>Runtime</td>
<td>Runtime</td>
<td>Runtime</td>
</tr>
<tr>
<td>Middleware</td>
<td>Middleware</td>
<td>Middleware</td>
<td>Middleware</td>
</tr>
<tr>
<td>O/S</td>
<td>O/S</td>
<td>O/S</td>
<td>O/S</td>
</tr>
<tr>
<td>Virtualization</td>
<td>Virtualization</td>
<td>Virtualization</td>
<td>Virtualization</td>
</tr>
<tr>
<td>Servers</td>
<td>Servers</td>
<td>Servers</td>
<td>Servers</td>
</tr>
<tr>
<td>Storage</td>
<td>Storage</td>
<td>Storage</td>
<td>Storage</td>
</tr>
<tr>
<td>Networking</td>
<td>Networking</td>
<td>Networking</td>
<td>Networking</td>
</tr>
</tbody>
</table>

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

Global Amazon Web Services (AWS) Infrastructure

GovCloud (US ITAR Region) 2
US West (Northern California) 3
US East (Northern Virginia) 5
Europe West 3 (Dublin)
Asia Pacific (Singapore) 2
Asia Pacific (Tokyo) 2

US West (Oregon) 3

# Zones
- AWS Regions
- AWS Edge Locations (CloudFront & Route 53)
Updates

- Piazza/Canvas
- cs370@cs.colostate.edu: TAs and instructors
- Final is comprehensive but questions will mostly be from the second half.
- Review on Thursday next week.
Reflecting on Part 1

• System structure and program compilation/execution
• Processes & Threads:
  – creation
  – scheduling
  – termination
• Inter-process communication
  – Synchronization
  – Deadlocks (included in Part 2)
Part 2

• We will review these on next Thursday.
• Virtual and physical address spaces
  – Pages and frames
    • Translation using page tables and TLBs
    • Effective access time
  – Virtual memory
    • Demand paging, page replacement algorithms
  – File systems
    • Disk organization, block allocation, scheduling
    • RAIDs
  – Virtual machines and containers
  – Data centers and cloud