CS 61C: Great Ideas in Computer Architecture (Machine Structures)  
Lecture 17 – Datacenters and Cloud Computing  
Instructor: Dan Garcia  
http://inst.eecs.Berkeley.edu/~cs61c/  

Computer Eras: Mainframe 1950s-60s  
- Processor (CPU)  
- “Big Iron”: IBM, UNIVAC, ... build $1M computers for businesses ➔ COBOL, Fortran, timesharing OS

Minicomputer Eras: 1970s  
- Using integrated circuits, Digital, HP... build $10k computers for labs, universities ➔ C, UNIX OS

PC Era: Mid 1980s - Mid 2000s  
- Using microprocessors, Apple, IBM, ... build $1k computer for 1 person ➔ Basic, Java, Windows OS

PostPC Era: Late 2000s - ??  
- Personal Mobile Devices (PMD): Relying on wireless networking, Apple, Nokia, ... build $500 smartphone and tablet computers for individuals ➔ Objective C, Java, Android OS + iOS  
- Cloud Computing: Using Local Area Networks, Amazon, Google, ... build $200M Warehouse Scale Computers with 100,000 servers for Internet Services for PMDs ➔ MapReduce, Ruby on Rails

Why Cloud Computing Now?  
- "The Web Space Race": Build-out of extremely large datacenters (10,000’s of commodity PCs)  
  ➔ Build-out driven by growth in demand (more users)  
  ➔ Infrastructure software and Operational expertise  
- Discovered economy of scale: 5-7x cheaper than provisioning a medium-sized (1000 servers) facility  
- More pervasive broadband Internet so can access remote computers efficiently  
- Commoditization of HW & SW  
  ➔ Standardized software stacks
March 2013 AWS Instances & Prices

<table>
<thead>
<tr>
<th>Instance</th>
<th>Price/ Hour</th>
<th>Per Hour</th>
<th>Compute Virtual Units</th>
<th>Memory (GB)</th>
<th>Disk (GB)</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Small</td>
<td>$0.065</td>
<td>1.0</td>
<td>1</td>
<td>17</td>
<td>169</td>
<td>32.94</td>
</tr>
<tr>
<td>Standard Large</td>
<td>$0.260</td>
<td>4.0</td>
<td>2</td>
<td>75</td>
<td>859</td>
<td>64.94</td>
</tr>
<tr>
<td>Standard Extra Large</td>
<td>$0.520</td>
<td>8.0</td>
<td>4</td>
<td>2.00</td>
<td>169</td>
<td>64.94</td>
</tr>
<tr>
<td>High-Memory Extra Large</td>
<td>$0.460</td>
<td>5.9</td>
<td>2</td>
<td>3.25</td>
<td>122</td>
<td>64.94</td>
</tr>
<tr>
<td>High-Memory Double Extra Large</td>
<td>$1.840</td>
<td>12.5</td>
<td>8</td>
<td>68.4</td>
<td>1699</td>
<td>64.94</td>
</tr>
<tr>
<td>High-CPU Medium</td>
<td>$0.165</td>
<td>2.0</td>
<td>5</td>
<td>2.50</td>
<td>1.7</td>
<td>350</td>
</tr>
<tr>
<td>High-CPU Extra Large</td>
<td>$0.660</td>
<td>8.0</td>
<td>2</td>
<td>2.50</td>
<td>7.0</td>
<td>1699</td>
</tr>
</tbody>
</table>

- Closest computer in WSC example is Standard Extra Large
- At these low rates, Amazon EC2 can make money!
  - even if used only 50% of time

Warehouse Scale Computers

- Massive scale datacenters: 10,000 to 100,000 servers + networks to connect them together
  - Emphasize cost-efficiency
  - Attention to power: distribution and cooling
- (relatively) homogeneous hardware/software
- Offer very large applications (Internet services): search, social networking, video sharing
- Very highly available: < 1 hour down/year
  - Must cope with failures common at scale
  - “...WSCs are no less worthy of the expertise of computer systems architects than any other class of machines” Barroso and Hoelzle 2009

Design Goals of a WSC

- Unique to Warehouse-scale
  - Ample parallelism:
    - Batch apps: large number independent data sets with independent processing. Also known as Data-Level Parallelism
  - Scale and its Opportunities/Problems
    - Relatively small number of these make design cost expensive and difficult to amortize
    - But price breaks are possible from purchases of very large numbers of commodity servers
    - Must also prepare for high # of component failures
  - Operational Costs Count:
    - Cost of equipment purchases << cost of ownership

Containers in WSCs

- Inside WSC
- Inside Container

Equipment Inside a WSC

Server (in rack format): 1 ¾ inches high "1U", x 19 inches x 16-20 inches: 8 cores, 16 GB DRAM, 4x1 TB disk 7 foot Rack: 40-80 servers + Ethernet local area network (1-10 Gbps) switch in middle ("rack switch")

Array (aka cluster): 16-32 server racks + larger local area network switch ("array switch") 10X faster → cost 100X: cost f(N²)
Defining Performance

- What does it mean to say X is faster than Y?

- 2009 Ferrari 599 GTB
  - 2 passengers, 11.1 secs for quarter mile (call it 10sec)

- 2009 Type D school bus
  - 54 passengers, quarter mile time? (let’s guess 1 min)
  - http://www.youtube.com/watch?v=KwyCoQuhUNA

- Response Time or Latency: time between start and completion of a task (time to move vehicle ¼ mile)

- Throughput or Bandwidth: total amount of work in a given time (passenger-miles in 1 hour)

Coping with Performance in Array

<table>
<thead>
<tr>
<th></th>
<th>Local</th>
<th>Rack</th>
<th>Array</th>
</tr>
</thead>
<tbody>
<tr>
<td>Racks</td>
<td>1</td>
<td>80</td>
<td>30</td>
</tr>
<tr>
<td>Servers</td>
<td></td>
<td>1</td>
<td>2400</td>
</tr>
<tr>
<td>Cores (Processors)</td>
<td>8</td>
<td>640</td>
<td>19,200</td>
</tr>
<tr>
<td>DRAM Capacity (GB)</td>
<td>16</td>
<td>1,280</td>
<td>38,400</td>
</tr>
<tr>
<td>Disk Capacity (TB)</td>
<td>4</td>
<td>320</td>
<td>9,600</td>
</tr>
<tr>
<td>DRAM Latency (microseconds)</td>
<td>0.1</td>
<td>100</td>
<td>300</td>
</tr>
<tr>
<td>Disk Latency (microseconds)</td>
<td>10,000</td>
<td>11,000</td>
<td>12,000</td>
</tr>
<tr>
<td>DRAM Bandwidth (MB/sec)</td>
<td>20,000</td>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td>Disk Bandwidth (MB/sec)</td>
<td>200</td>
<td>100</td>
<td>10</td>
</tr>
</tbody>
</table>

Impact of latency, bandwidth, failure, varying workload on WSC software?

- WSC Software must take care where it places data within an array to get good performance
- WSC Software must cope with failures gracefully
- WSC Software must scale up and down gracefully in response to varying demand
- More elaborate hierarchy of memories, failure tolerance, workload accommodation makes WSC software development more challenging than software for single computer

Coping with Workload Variation

- Online service: Peak usage 2X off-peak
Power vs. Server Utilization

- Server power usage as load varies idle to 100%
- Uses ½ peak power when idle!
- Uses ¾ peak power when 10% utilized! 90%@ 50%!
- Most servers in WSC utilized 10% to 50%
- Goal should be Energy-Proportionality:
  % peak load = % peak energy

Power Usage Effectiveness

- Overall WSC Energy Efficiency: amount of computational work performed divided by the total energy used in the process
- Power Usage Effectiveness (PUE): Total building power / IT equipment power
  - A power efficiency measure for WSC, not including efficiency of servers, networking gear
  - 1.0 = perfection

PUE in the Wild (2007)

High PUE: Where Does Power Go?

Google WSC A PUE: 1.24

- Careful air flow handling
  - Don’t mix server hot air exhaust with cold air (separate warm aisle from cold aisle)
  - Short path to cooling so little energy spent moving cold or hot air long distances
  - Keeping servers inside containers helps control air flow
- Elevated cold aisle temperatures
  - 82°F instead of traditional 65°- 68°F
  - Found reliability OK if run servers hotter
- Use of free cooling
  - Cool warm water outside by evaporation in cooling towers
  - Locate WSC in moderate climate so not too hot or too cold
- Per-server 12 V DC UPS
  - Rather than WSC wide UPS, place single battery per server board
  - Increases WSC efficiency from 90% to 99%
- Measure vs. estimate PUE, publish PUE, and improve operation

Summary

- Parallelism is one of the Great Ideas
  - Applies at many levels of the system – from instructions to warehouse scale computers
- Post PC Era: Parallel processing, smart phone to WSC
- WSC SW must cope with failures, varying load, varying HW latency bandwidth
- WSC HW sensitive to cost, energy efficiency
- WSCs support many of the applications we have come to depend on