Above the Clouds:
A Berkeley View of Cloud Computing
Armando Fox, UC Berkeley
Reliable Adaptive Distributed Systems Lab
© 2009-2011

Image: John Curley http://www.flickr.com/photos/jay_que/1834540/

What is distributed computing?

Your PC vs. Datacenter Computer, in 1996 & today

Sun E-10000 “supermini” c.1996

<table>
<thead>
<tr>
<th>Machine</th>
<th>Processor cores</th>
<th>RAM</th>
<th>Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>E10000, 1996</td>
<td>64 x 250MHz</td>
<td>64 GB</td>
<td>20 TB</td>
</tr>
<tr>
<td>PC, 1996</td>
<td>1 x 250 MHz</td>
<td>32 MB</td>
<td>4 GB</td>
</tr>
<tr>
<td>Ratio</td>
<td>64:1</td>
<td>2000:1</td>
<td>5000:1</td>
</tr>
<tr>
<td>Datacenter</td>
<td>8 x 1 GHz</td>
<td>16 GB</td>
<td>2 TB</td>
</tr>
<tr>
<td>computer, 2010</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ratio</td>
<td>&lt; 2:1</td>
<td>4:1</td>
<td>4:1</td>
</tr>
<tr>
<td>PC, 2010</td>
<td>2 x 3 GHz</td>
<td>4 GB</td>
<td>0.5 TB</td>
</tr>
</tbody>
</table>

• The first demonstration of how to build really large Internet sites out of clusters of commodity computers was done by:
  (a) Stanford
  (b) Berkeley
  (c) Yahoo!
  (d) Google
  (e) IBM

NOW-0
1994
Four HP-735’s

UC Berkeley Networks Of Workstations (1994-1999)
NOW-1
1995
32 Sun SPARC-stations

NOW-2
1997
60 Sun SPARC-2

Challenge: how do you program a NOW? How do you keep it running as individual machines fail?

Trivia Fact

• The first full Web browser running on a mobile device was developed by:
  (a) Apple
  (b) Stanford
  (c) Berkeley
  (d) Nokia
  (e) Motorola

“Access Is the Killer App”
Project Daedalus, 1994-1999

• Faculty: Profs. Katz & Brewer
• Idea: Use the “cloud” for services!
  – First truly scalable search engine (Inktomi)
  – First mobile Web browser enabled by content transformation (TopGun)
  – Vision: Anywhere, anytime access to data & services, supported by the “cloud”

• A Google datacenter built c.2005 would be designed to house approximately _______ computers.
  (a) 1,000
  (b) 5,000
  (c) 10,000
  (d) 50,000
  (e) 100,000
Datacenter is new “server”

- “Program” => Web search, email, map/GIS, …
- “Computer” => 1000’s computers, storage, network
- Warehouse-sized facilities and workloads

RAD Lab 5-year Mission

Enable 1 entrepreneur to prototype a great Web app over 3-day weekend, then deploy at scale

- Key enabling technology: Statistical machine learning
- Highly interdisciplinary faculty & students
  - 7 faculty across CS, from theory to systems
  - 2 postdocs, ~30 PhD students, ~12 undergrads

2007: Public Cloud Computing Arrives

- Amazon Elastic Compute Cloud (EC2)
- “Compute unit” rental: $0.02-0.68/hr.
  - 1 CU = ~1 GHz x86 core
  - Virtual machine technology used to “slice up”
- No up-front cost, no contract, no minimum
- Billing rounded to nearest hour
  - pay-as-you-go storage also available
- “Computing as utility” — MULTICS, c.1969
- See abovetheclouds.cs.berkeley.edu

Why Now (not then)?

- The Web “Space Race”: Build-out of extremely large datacenters (10,000’s of commodity PCs)
- Driven by growth in demand (more users)
  - Discovered economy of scale: 5-7x cheaper than provisioning a medium-sized (100’s machines) facility
  - Infrastructure software: e.g., Google File System
  - Operational expertise
- More pervasive broadband Internet
- Dominance of Intel x86 architecture in servers
- Free & open source software availability
- What’s new: risk transfer & cost associativity

Cloud Economics 101

- Provisioning for peaks: wasteful, but necessary

Risk Transfer (or: who remembers Friendster?)

- Unused resources

- Lost revenue
- Lost users
Cost Associativity

- 1,000 CPUs for 1 hour same price as 1 CPU for 1,000 hours
- Washington Post converted Hillary Clinton’s travel documents to post on WWW
  - Conversion time: <1 day after released
  - Cost: less than $200
- RAD Lab graduate students demonstrate improved MapReduce scheduling—on 1,000 servers

Challenge: Cloud Programming

- Challenge: exposing parallelism
  - Programmers must (re)write problems to expose this parallelism, if it’s there to be found
- Challenge: operations
  - Failures a constant fact when use 10,000 machines
  - Automating the process of grabbing/releasing machines

Rising to the challenge

- Programming
  - BOOM (Berkeley Orders of Magnitude) simplifies creating cloud-scale storage services (Hellerstein et al.)
  - SEJITS (Selective Embedded Just-in-Time Specialization) lets same Python programs exploit cloud-scale or CPU-level parallelism (Fox et al.)
- Operations
  - RAD Lab expertise in using machine learning to auto-scale servers and storage in cloud

Success Stories: Karl’s Long Weekend

Presidents’ Day Weekend, Feb 21-13
Final demo on Feb 24

Cloud in Education

- Berkeley research culture: integrate leading research into teaching at all levels
- CS61C Great Ideas in Computer Architecture (reinvented Fall 2010): 190 students
- CS169 Software Engineering for SaaS (in its 4th iteration): 50+50+50+70 students
- CS162 Operating Systems: 70 students
- (New course) Intro. Data Science (Spring 2010): 30
- (New course) Programming Cloud Storage with BOOM (Fall 2011)
- CS260 Adv. topics in HCI: 20 students
- CS288 Natural language processing: 20 students

Cloud computing in courses

- New undergraduate teaching opportunities
  - SaaS: make a database fall over—would need 200 servers for ~20 project teams
  - deploy projects publicly, many continue after course
- Better use of resources
  - Heavy usage right before lab deadlines
- Better hardware
  - Better machines than students’ own laptops
  - Better machines than most UCB labs
Going back to NOW...

- **2000**: using medium-sized clusters for Internet services => several PhD’s
- **2010**: CS169 students do it in 6-8 weeks and deploy on cloud computing – *Everything delivered as SaaS now…*
- **2020**: ?

Summary

- Cloud computing *democratizes access* to large-scale computing resources
  - Pay-as-you-go => low risk, low entry cost
- *Accelerates “SaaS-ification”*
  - Economic benefits of delivering software as a service now available to anyone
- Allows students, academia to have even greater impact on industry
- Open up research/innovation opportunities

Relevant Topics?

- SaaS architecture & cloud (CS 169)
- Big data (CS 194 Intro to Data Science this semester)
- Machine learning (CS 188)
- Human-computer interaction (CS 160)
- *Non-goal:* “iPhone programming”, “Android programming”, etc. (why?)

Thank you!