RAD Lab
UC Berkeley

The Potential of Cloud Computing: Challenges, Opportunities, Impact

Armando Fox, UC Berkeley
Reliable Adaptive Distributed Systems Laboratory

NAE Frontiers of Engineering, September 2010

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

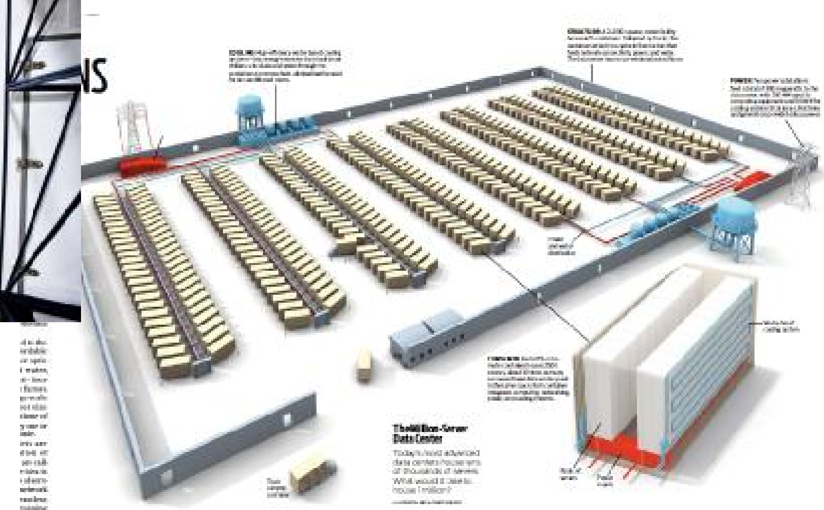
Get Your Own Supercomputer

- 41.88 Teraflops on LINPACK benchmark
  – Faster than #145 entry on current Top500 list
- No batch queues—start your job in minutes
- No grantwriting—just a credit card ($1400/hour, 1 hour minimum purchase)
- No need to predict usage in advance—pay only for what you use, get more on demand
- Lease several simultaneously
- MPI and Matlab available, among other packages

Warehouse Scale Computers

- Built to support consumer demand of Web services (email, social networking, etc.)
  - “Private clouds” of Google, Microsoft, Amazon, ...
- “Warehouse scale” buying power = 5-7x cheaper hardware, networking, administration cost





photos: Cnet News, Sun Microsystems, datacenterknowledge.com

2008: Public Cloud Computing Arrives

Type & Price/Hour	1GHz core eqv.	RAM	Disk and/or I/O
Small - $0.085	1	1.7 GB	160 GB
Large - $0.34	4	7.5 GB	850 GB, 2 spindles
XLarge - $0.68	8	15 GB	1690 GB, 3 spindles
Cluster (“HPC”) - $1.60	32*	23 GB	1690 GB + 10Gig Ethernet

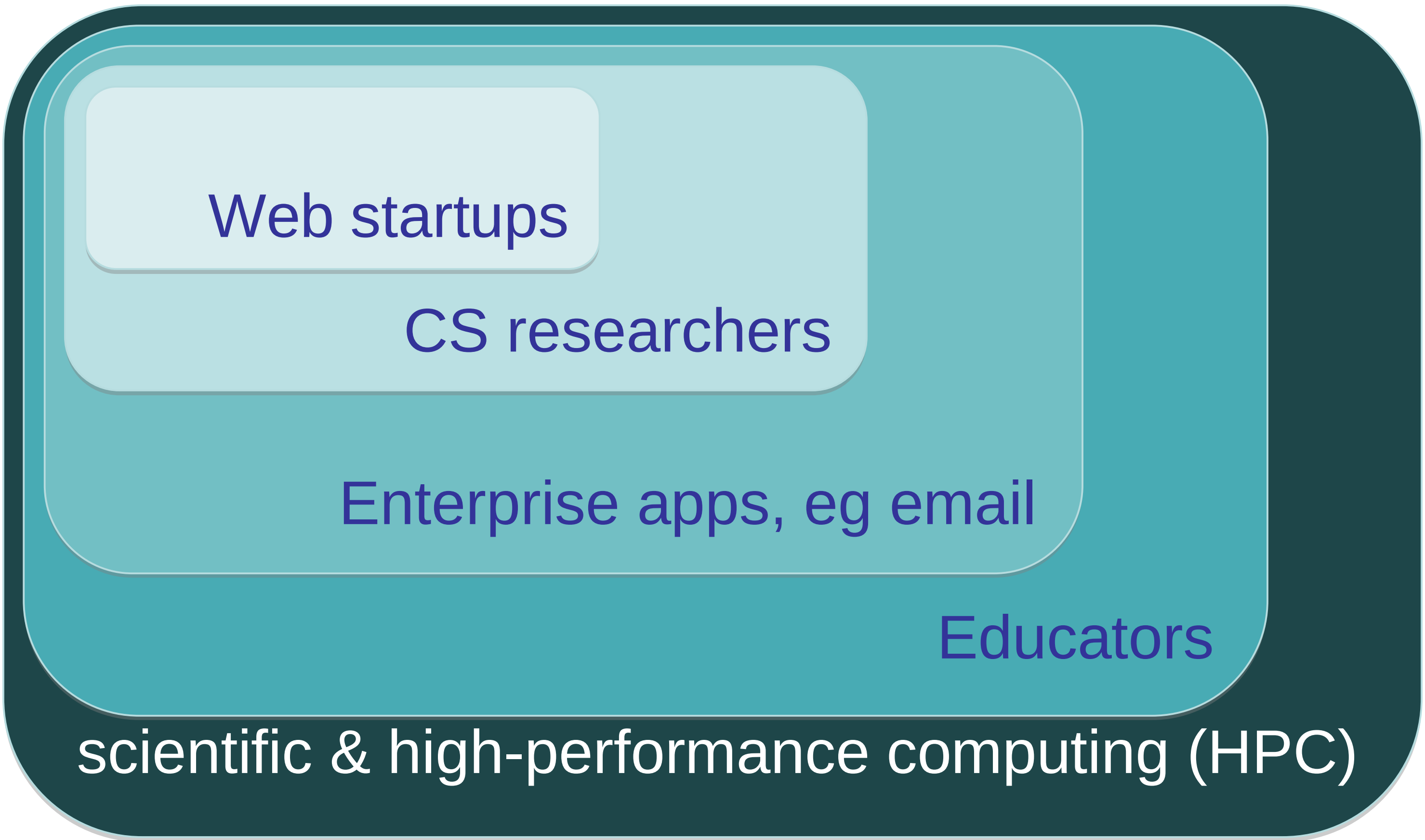
- Unlike Grid, computing on-demand *for the public*
- Virtual machines from $0.085 to $1.60/hr
  – Pay as you go with credit card, 1 hr. minimum
  – Cheaper if willing to share or risk getting kicked off
  – Machines provisioned & booted in a few minutes

1,000 machines for 1 hr = 1 machine × 1,000 hrs

* 2x quad-core Intel Xeon (Nehalem)

What’s In It For You?

Cloud Computing progress 2008-2010



What's in it for you?

- **What:** Low cost + on-demand + cost-associativity = Democratization of Supercomputing
- **How:** Sophisticated software & operational expertise mask unreliability of commodity components
- Example: **MapReduce**
  - LISP language construct (~1960) for elegantly expressing *data parallel* operations w/sequential code
  - MapReduce (Google) and Hadoop (open source) provide failure masking & resource management for performing this on commodity clusters

"Warehouse scale" software engineering issues ***hidden*** *from application programmer*

MapReduce in Practice

- Example: spam classification
  - training: 10^7 URLs x 64KB data each = 640GB data
  - One heavy-duty server: ~270 hours
  - 100 servers in cloud: ~3 hours (= ~$255)
- Rapid uptake in other scientific research
  - Large-population genetic risk analysis & simulation (Harvard Medical School)
  - Genome sequencing (UNC Chapel Hill Cancer Ctr)
  - Information retrieval research (U. Glasgow – Terrier)
  - Compact Muon Solenoid Expt. (U. Nebraska Lincoln)
- What's the downside?

Challenge: Cloud Programming

- Challenge: exposing parallelism
  - MapReduce relies on “embarrassing parallelism”
- Programmers must (re)write problems to expose this parallelism, if it’s there to be found
- Tools still primitive, though progressing rapidly
  - MapReduce—early success story
  - Pig (Yahoo! Research) & Hive (Apache Foundation) – write database-like queries to run as Hadoop jobs
  - Mesos—share cluster among Hadoop, MPI, interactive jobs (UC Berkeley)
- Challenge for tool authors: parallel software hard to debug and operate reliably (YY Zhou)

Challenge: Big Data

Application	Data generated per day
DNA Sequencing (Illumina HiSeq machine)	1 TB
Large Synoptic Survey Telescope	30 TB; 400 Mbps sustained data rate between Chile and NCSA
Large Hadron Collider	60 TB

- Challenge: Long-haul networking is most expensive cloud resource, and improving most slowly
- Copy 8 TB to Amazon over ~20 Mbps network => ~35 days, ~$800 in transfer fees
- How about shipping 8TB drive to Amazon instead? => 1 day, ~$150 (shipping + transfer fees)

Source: Ed Lazowska, eScience 2010, Microsoft Cloud Futures Workshop, lazowska.cs.washington.edu/cloud2010.pdf

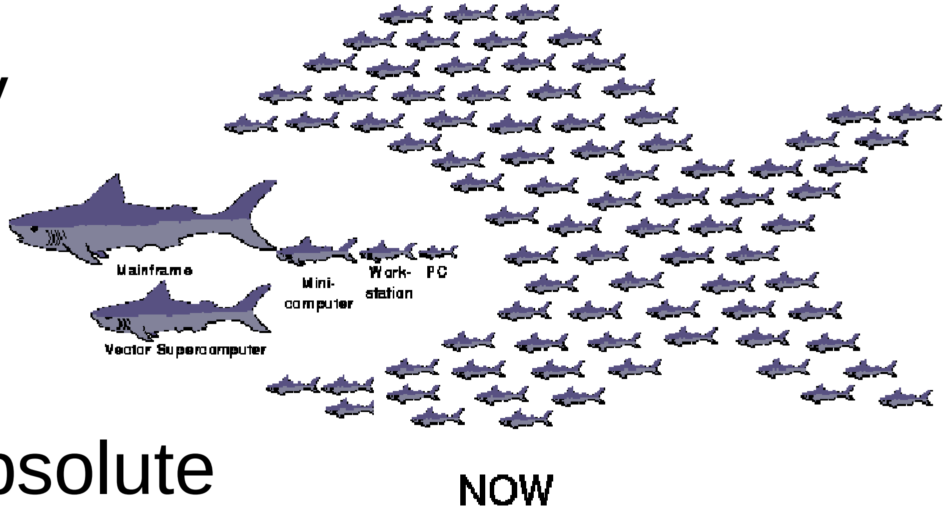
Challenge: “non-cloudy” scientific codes

- Challenge: scientific and HPC codes designed around “supercomputer-centric” assumptions
  - reliability, static configuration, exclusive resource use...
- Opportunity: resource management frameworks
- Time-to-answer may still be faster, since no wait!
- Cloud vendors are listening to HPC customers
  - July 2010: cluster of 880 EC2 “cluster compute” instances achieve 41.82 Tflops on LINPACK
  - MathWorks supporting “cloud MATLAB” on these

Better Education

- Capacity on demand is perfect for university courses
- Students can run new kinds of experiments at scale
- Unburdens system administrators, so provisioning is faster
- Students get access to state-of-the-art hardware
- Berkeley now using cloud computing from entry-level to graduate EECS courses

An Analogy: Clusters of Commodity PC's, c.1995



- Clusters of Commodity PC's vs. symmetric multiprocessors
- *Potential* advantages: incremental scaling, absolute capacity, economy of using commodity HW & SW
- 1995: SaaS on SMPs; software architecture SMP-centric
- 2010: Berkeley undergrads prototype SaaS in 6 8 weeks and deploy on cloud computing

Conclusion

- Democratization of supercomputing capability
  - Time to answer may be faster even if hardware isn't
  - Writing a grant proposal around a supercomputer?
- Software & hardware rapidly improving as vendors listen to scientific/HPC customers
- HPC opportunity to influence design of *commodity* equipment

Impact: Potential democratizing effect comparable to microprocessor

Acknowledgments

- Colleagues at UC Berkeley RAD Lab
  - Co-authors of *Above The Clouds:* Michael Armbrust, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia
  - And everyone else in the RAD Lab
- Colleagues at our industrial collaborators

Thank you!