CS 61C:
Great Ideas in Computer Architecture
(formerly called Machine Structures)

Course Introduction

Instructor:
David A. Patterson

http://inst.eecs.Berkeley.edu/~cs61c/sp12

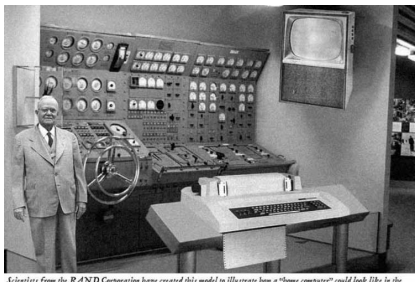
Agenda

- Great Ideas in Computer Architecture
- Administrivia
- PostPC Era: From Phones to Datacenters
- Software as a Service
- Cloud Computing
- Technology Break
- Warehouse Scale Computers in Depth

CS61c is NOT really about C Programming

- It is about the hardware-software interface
  - What does the programmer need to know to achieve the highest possible performance
- Languages like C are closer to the underlying hardware, unlike languages like Python!
  - Allows us to talk about key hardware features in higher level terms
  - Allows programmer to explicitly harness underlying hardware parallelism for high performance

Old School CS61c

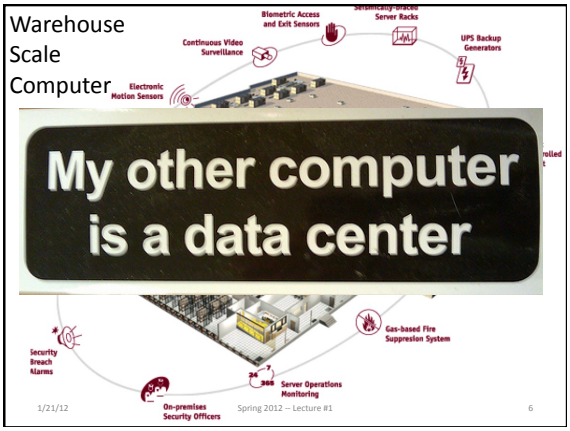


Scientists from the RAND Corporation have created this model to illustrate how a "home computer" could look like in the year 2004. However the needed technology will not be economically feasible for the average home. Also the scientists readily admit that the computer will require not yet invented technology to actually work, but 30 years from now scientific progress is expected to solve these problems. With teletype interface and the Fortran language, the computer will be easy to use.

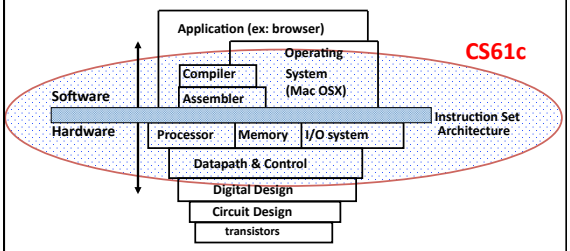
Personal Mobile Devices

New School CS61c

Warehouse Scale Computer



Old-School Machine Structures



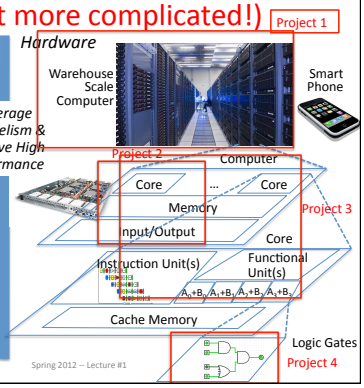
New-School Machine Structures (It's a bit more complicated!)

Project 1

Software | *Hardware*

- Parallel Requests
  Assigned to computer
  e.g., Search "Katz"
- Parallel Threads
  Assigned to core
  e.g., Lookup, Ads
- Parallel Instructions
  >1 instruction @ one time
  e.g., 5 pipelined instructions
- Parallel Data
  >1 data item @ one time
  e.g., Add of 4 pairs of words
- Hardware descriptions
  All gates functioning in parallel at same time
- Programming Languages

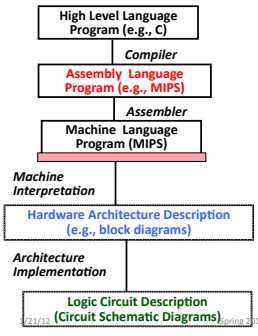
Leverage Parallelism & Achieve High Performance



6 Great Ideas in Computer Architecture

1. Layers of Representation/Interpretation
2. Moore's Law
3. Principle of Locality/Memory Hierarchy
4. Parallelism
5. Performance Measurement & Improvement
6. Dependability via Redundancy

Great Idea #1: Levels of Representation/Interpretation

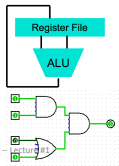



```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

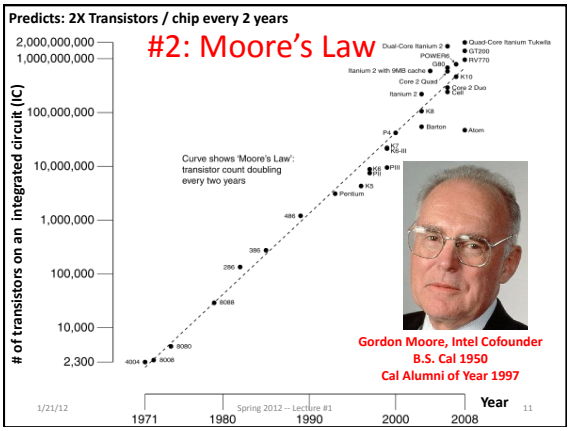
```
lw   $t0, 0($2)
lw   $t1, 4($2)
sw   $t1, 0($2)
sw   $t0, 4($2)
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

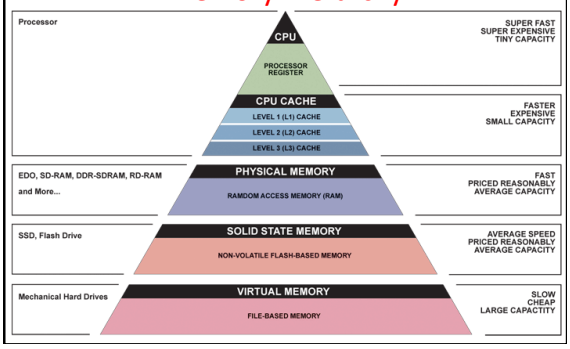
Anything can be represented as a *number*, i.e., data or instructions



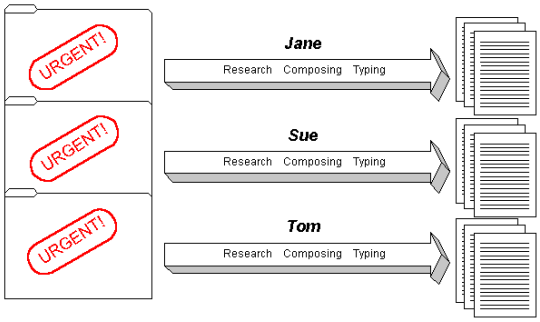
#2: Moore's Law



Great Idea #3: Principle of Locality/ Memory Hierarchy



Great Idea #4: Parallelism



Great Idea #5: Performance Measurement and Improvement

- Matching application to underlying hardware to exploit:
  - Locality
  - Parallelism
  - Special hardware features, like specialized instructions (e.g., matrix manipulation)
- Latency
  - How long to set the problem up
  - How much faster does it execute once it gets going
  - It is all about *time to finish*

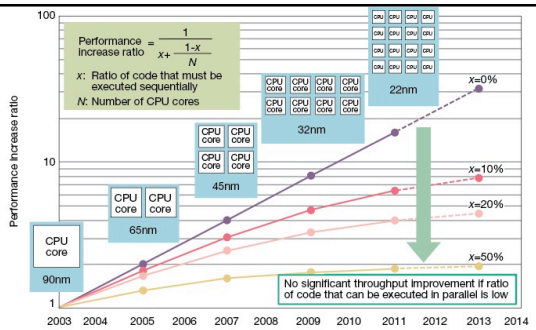
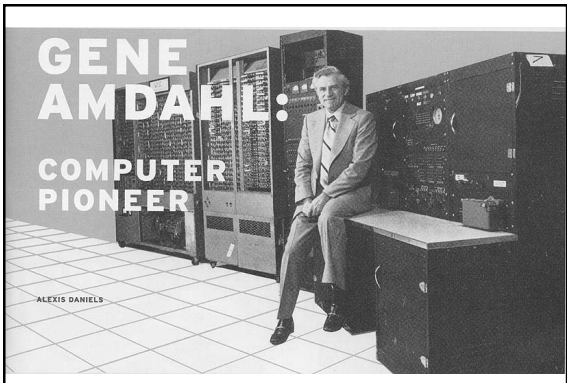
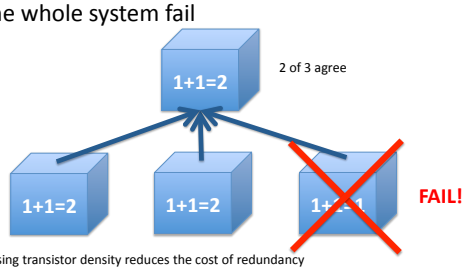


Fig 3 Amdahl's Law an Obstacle to Improved Performance Performance will not rise in the same proportion as the increase in CPU cores. Performance gains are limited by the ratio of software processing that must be executed sequentially. Amdahl's Law is a major obstacle in boosting multicore microprocessor performance. Diagram assumes no overhead in parallel processing. Years shown for design rules based on Intel planned and actual technology. Core count assumed to double for each rule generation.



Great Idea #6: Dependability via Redundancy

- Redundancy so that a failing piece doesn't make the whole system fail



Increasing transistor density reduces the cost of redundancy

Great Idea #6: Dependability via Redundancy

- Applies to everything from datacenters to storage to memory
  - Redundant datacenters so that can lose 1 datacenter but Internet service stays online
  - Redundant disks so that can lose 1 disk but not lose data (Redundant Arrays of Independent Disks/RAID)
  - Redundant memory bits of so that can lose 1 bit but no data (Error Correcting Code/ECC Memory)

Question: Which statement is TRUE about Big Ideas in Computer Architecture?

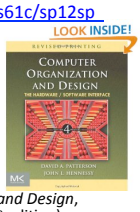
- To offer a dependable system, you must use components that almost never fail
- Memory hierarchy goal: look ≈as fast as most expensive memory, ≈ as big as cheapest
- Moore's Law means computers get twice as fast every ≈ 1.5 years
- The goal of levels of interpretation is to build the most efficient hardware and software

Agenda

- Great Ideas in Computer Architecture
- Administrivia
- PostPC Era: From Phones to Datacenters
- Software as a Service
- Cloud Computing
- Technology Break
- Warehouse Scale Computers in Depth

Course Information

- Course Web: http://inst.eecs.Berkeley.edu/~cs61c/sp12sp
- Instructor:
  - Dave Patterson
- Teaching Assistants:
  - Rimas Avizienis, Scott Beamer, Alan Christopher, Eric Liang, Paul Ruan, Ian Vonseggern
- Textbooks: Average 15 pages of reading/week
  - Barroso & Holzle, *The Datacenter as a Computer (free download from web page)*
  - Patterson & Hennessey, *Computer Organization and Design*, Revised 4th Edition (not ≤3rd Edition, not Asian 4th edition)
  - Kernighan & Ritchie, *The C Programming Language*, 2nd Edition
- Piazza for class announcements, Q&A:
  - Just go to Piazza web page and add yourself to the class
  - Staff reads them all; please keep it class related and professional



Reminders

- Discussions and labs will be held this week
  - Switching Sections: if you find another 61C student willing to swap discussion AND lab, talk to your TAs
  - Partner (projects 1, 3, and extra credit): OK if partners mix sections but have same TA
- First homework assignment due this Sunday January 22rd by 11:59:59 PM
  - There is reading assignment as well on course page

Course Organization

- Grading
  - Participation and Altruism (5%)
  - Homework (5%)
  - Labs (20%)
  - Projects (40%)
    1. Data Parallelism (Map-Reduce on Amazon EC2, with partner)
    2. Computer Instruction Set Simulator (C)
    3. Performance Tuning of a Parallel Application/Matrix Multiply using cache blocking, SIMD, MIMD (OpenMP, with partner)
    4. Computer Processor Design (Logisim)
  - Extra Credit: Matrix Multiply Competition, anything goes
  - Midterm (10%): 6-9 PM Tuesday March 6, 155 Dwinelle
  - Final (20%): 11:30-2:30 PM Wednesday May 9

EECS Grading Policy

- http://www.eecs.berkeley.edu/Policies/ugrad.grading.shtml
  "A typical GPA for courses in the lower division is 2.7. This GPA would result, for example, from 17% A's, 50% B's, 20% C's, 10% D's, and 3% F's. A class whose GPA falls outside the range 2.5 - 2.9 should be considered atypical."
- Spring 2011: GPA 2.85
  24% A's, 49% B's, 18% C's, 6% D's, 3% F's
- Job/Intern Interviews: They grill you with technical questions, so it's what you say, not your GPA
  (New 61c gives good stuff to say)

	Fall	Spring
2011	2.72	2.85
2010	2.81	2.81
2009	2.71	2.81
2008	2.95	2.74
2007	2.67	2.76

Late Policy

- Assignments due Sundays at 11:59:59 PM
- Late homeworks not accepted (100% penalty)
- Late projects get 20% penalty, accepted up to Tuesdays at 11:59:59 PM
  - No credit if more than 48 hours late
  - No "slip days" in 61C
    - Used by Dan Garcia and a few faculty to cope with 100s of students who often procrastinate without having to hear the excuses, but not widespread in EECS courses
    - More late assignments if everyone has no-cost options; better to learn now how to cope with real deadlines

Policy on Assignments and Independent Work

- With the exception of laboratories and assignments that explicitly permit you to work in groups, all homeworks and projects are to be YOUR work and your work ALONE.
- You are encouraged to discuss your assignments with other students, and extra credit will be assigned to students who help others, particularly by answering questions on the Google Group, but we expect that what you hand is yours.
- It is NOT acceptable to copy solutions from other students.
- It is NOT acceptable to copy (or start your) solutions from the Web.
- We have tools and methods, developed over many years, for detecting this. You WILL be caught, and the penalties WILL be severe.
- At the minimum a ZERO for the assignment, possibly an F in the course, and a letter to your university record documenting the incidence of cheating.
- (We caught people last time taught 61C!)

YOUR BRAIN ON COMPUTERS; Hooked on Gadgets, and Paying a Mental Price

NY Times, June 7, 2010, by Matt Richtel

SAN FRANCISCO -- When one of the most important e-mail messages of his life landed in his in-box a few years ago, Kord Campbell overlooked it.

Not just for a day or two, but 12 days. He finally saw it while sifting through old messages: a big company wanted to buy his Internet start-up.

"I stood up from my desk and said, 'Oh my God, oh my God, oh my God,' " Mr. Campbell said. "It's kind of hard to miss an e-mail like that, but I did."

The message had slipped by him amid an electronic flood: two computer screens alive with e-mail, instant messages, online chats, a Web browser and the computer code he was writing.

While he managed to salvage the $1.3 million deal after apologizing to his suitor, Mr. Campbell continues to struggle with the effects of the deluge of data. Even after he unplugs, he craves the stimulation he gets from his electronic gadgets. He forgets things like dinner plans, and he has trouble focusing on his family. His wife, Brenda, complains, "It seems like he can no longer be fully in the moment."

This is your brain on computers.

Scientists say juggling e-mail, phone calls and other incoming information can change how people think and behave. They say our ability to focus is being undermined by bursts of information.

These play to a primitive impulse to respond to immediate opportunities and threats. The stimulation provokes excitement -- a dopamine squirt -- that researchers say can be addictive. In its absence, people feel bored.

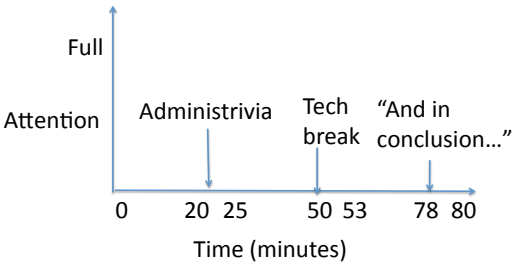
The resulting distractions can have deadly consequences, as when cellphone-wielding drivers and train engineers cause wrecks. And for millions of people like Mr. Campbell, these urges can inflict nicks and cuts on creativity and deep thought, interrupting work and family life.

While many people say multitasking makes them more productive, research shows otherwise. Heavy multitaskers actually have more trouble focusing and shutting out irrelevant information, scientists say, and they experience more stress.

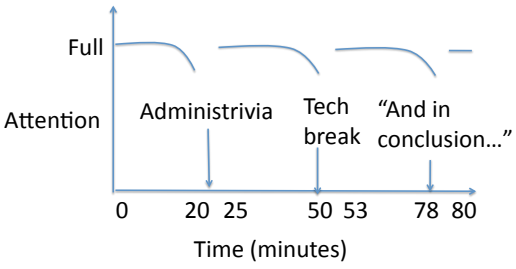
And scientists are discovering that even after the multitasking ends, fractured thinking and lack of focus persist. In other words, this is also your brain off computers.

The Rules (and we really mean it!)

Architecture of a Lecture

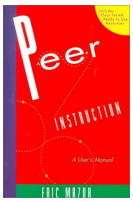


Architecture of a Lecture



Peer Instruction

- Increase real-time learning in lecture, test understanding of concepts vs. details
  mazur-www.harvard.edu/education/pi.phtml
- As complete a "segment" ask multiple choice question
  - <1 minute: decide yourself, vote
  - <2 minutes: discuss in pairs, then team vote; flash card to pick answer
    - Try to convince partner; learn by teaching
- Mark and save flash cards (get in discussion section)



1 2 3 4

Question: Which statements are TRUE about this class?

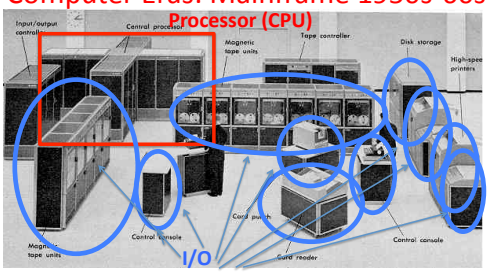
- ☐ The midterm is Tuesday March 6 during class (9:30-11)
- ☐ The midterm is Tuesday March 6 in the evening (6-9PM) in 155 Dwinelle
- ☐ It's OK to book airline tickets before May 9; Patterson will surely let me take final early
- ☐ I can save money by buying Asian edition of Computer Organization and Design

---

Agenda

- Great Ideas in Computer Architecture
- Administrivia
- PostPC Era: From Phones to Datacenters
- Software as a Service
- Cloud Computing
- Technology Break
- Warehouse Scale Computers in Depth

Computer Eras: Mainframe 1950s-60s



"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

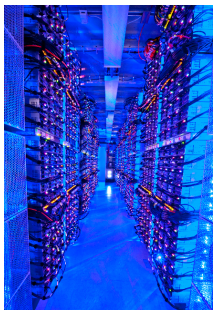
PC Era: Mid 1980s - Mid 2000s

Using microprocessors, Apple, IBM, … build $1k computer for 1 person => Basic, Java, Windows OS

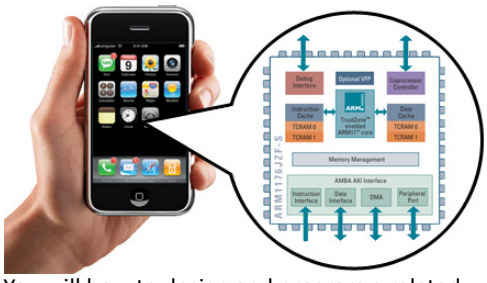
PostPC Era: Late 2000s - ??

Personal Mobile Devices (PMD): Relying on wireless networking, Apple, Nokia, … build $500 smartphone and tablet computers for individuals
=> Objective C, Android OS

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

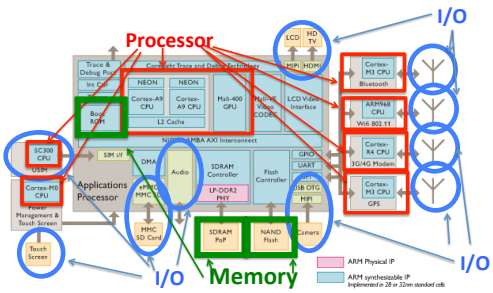


Advanced RISC Machine (ARM) instruction set inside the iPhone



You will how to design and program a related RISC computer: MIPS

iPhone Innards



You will about multiple processors, data level parallelism, caches in 61C

Why Not 80x86 vs. MIPS?

- Once learn one, easy to pick up others
- 80x86 instruction set is not beautiful
  - ≈ Full suitcase then add clothes on way to plane
  - Class time precious; why spend on minutiae?
- MIPS represents energy efficient processor of client (PostPC era) vs. fast processor of desktop (PC era)
- MIPS represents more popular instruction set: 2010: 6.1B ARM, 0.3B 80x86 (20X more)

Agenda

- Great Ideas in Computer Architecture
- Administrivia
- PostPC Era: From Phones to Datacenters
- Software as a Service
- Cloud Computing
- Technology Break
- Warehouse Scale Computers in Depth

Software as a Service: SaaS

- Traditional SW: binary code installed and runs wholly on client device
- SaaS delivers SW & data as service over Internet via thin program (e.g., browser) running on client device
  - Search, social networking, video
- Now also SaaS version of traditional SW
  - E.g., Microsoft Office 365, TurboTax Online

6 Reasons for SaaS

1. No install worries about HW capability, OS
2. No worries about data loss (at remote site)
3. Easy for groups to interact with same data
4. If data is large or changed frequently, simpler to keep 1 copy at central site
5. 1 copy of SW, controlled HW environment => no compatibility hassles for developers
6. 1 copy => simplifies upgrades for developers *and* no user upgrade requests

43

SaaS Infrastructure?

- SaaS demands on infrastructure

1. Communication: allow customers to interact with service
2. Scalability: fluctuations in demand during + new services to add users rapidly
3. Dependability: service and communication continuously available 24x7

44

Clusters

- Clusters: Commodity computers connected by commodity Ethernet switches

1. More scalable than conventional servers
2. Much cheaper than conventional servers
   - 20X for equivalent vs. largest servers
3. Few operators for 1000s servers
   - Careful selection of identical HW/SW
   - Virtual Machine Monitors simplify operation
4. Dependability via extensive redundancy

45

Agenda

- Great Ideas in Computer Architecture
- Administrivia
- PostPC Era: From Phones to Datacenters
- Software as a Service
- Cloud Computing
- Technology Break
- Warehouse Scale Computers in Depth

Warehouse Scale Computers

- Economies of scale pushed down cost of largest datacenter by factors 3X to 8X
  - Purchase, house, operate 100K v. 1K computers
- Traditional datacenters utilized 10% - 20%
- Make profit offering pay-as-you-go use at less than your costs for as many computers as you need

47

Utility Computing / Public Cloud Computing

- Offers computing, storage, communication at pennies per hour
- No premium to scale:

  1000 computers @ 1 hour
  = 1 computer @ 1000 hours
- Illusion of infinite scalability to cloud user
  - As many computers as you can afford
- Leading examples: Amazon Web Services, Google App Engine, Microsoft Azure

48

2012 AWS Instances & Prices

Instance	Per Hour	Ratio to Small	Compute Units	Virtual Cores	Compute Unit/ Core	Memory (GB)	Disk (GB)	Address
Standard Small	$0.085	1.0	1.0	1	1.00	1.7	160	32 bit
Standard Large	$0.340	4.0	4.0	2	2.00	7.5	850	64 bit
Standard Extra Large	$0.680	8.0	8.0	4	2.00	15.0	1690	64 bit
High-Memory Extra Large	$0.500	5.9	6.5	2	3.25	17.1	420	64 bit
High-Memory Double Extra Large	$1.200	14.1	13.0	4	3.25	34.2	850	64 bit
High-Memory Quadruple Extra Large	$2.400	28.2	26.0	8	3.25	68.4	1690	64 bit
High-CPU Medium	$0.170	2.0	5.0	2	2.50	1.7	350	32 bit
High-CPU Extra Large	$0.680	8.0	20.0	8	2.50	7.0	1690	64 bit
Cluster Quadruple Extra Large	$1.300	15.3	33.5	16	2.09	23.0	1690	64 bit
Eight Extra Large	$2.400	28.2	88.0	32	2.75	60.5	1690	64 bit

49

Supercomputer for hire

- Top 500 supercomputer competition
- 290 Eight Extra Large (@ $2.40/hour) = 240 TeraFLOPS
- 42nd/500 supercomputer @ ~$700 per hour
- Credit card => can use 1000s computers
- FarmVille on AWS
  - Prior biggest online game 5M users
  - What if startup had to build datacenter? How big?
  - 4 days =1M; 2 months = 10M; 9 months = 75M

50

IBM Watson for Hire?

- Jeopardy Champion IBM Watson
- Hardware: 90 IBM Power 750 servers
  - 3.5 GHz 8 cores/server
- 90 @ ~$2.40/hour = ~$200/hour
- Cost of human lawyer or account
- For what tasks could AI be as good as highly trained person @ $200/hour?
- What would this mean for society?

51

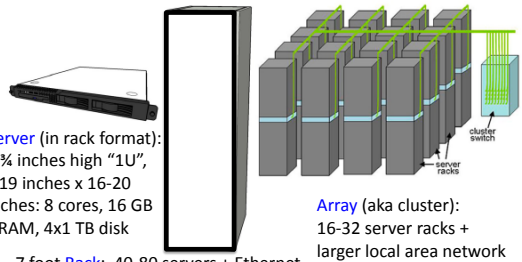
Which statements are NOT true about SaaS and Cloud Computing?

- ☐ Clusters are collections of commodity servers connected by LAN switches
- ☐ The Internet supplies the communication for SaaS
- ☐ Cloud computing uses HW clusters + SW layer using redundancy for dependability
- ☐ Private datacenters could match cost of Warehouse Scale Computers if they just purchased the same type of hardware

52

E.g., Google's Oregon WSC

Equipment Inside a WSC



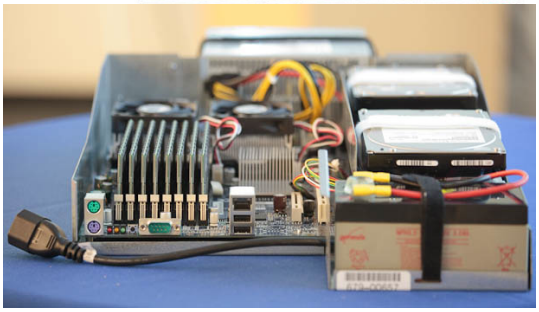
Server (in rack format): 1 ¾ inches high "1U", x 19 inches x 16-20 inches: 8 cores, 16 GB DRAM, 4x1 TB disk

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(N^2)

Server, Rack, Array

Google Server Internals

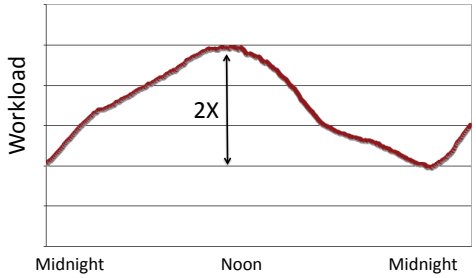


Coping with Performance in Array

Lower latency to DRAM in another server than local disk
Higher bandwidth to local disk than to DRAM in another server

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

Coping with Workload Variation

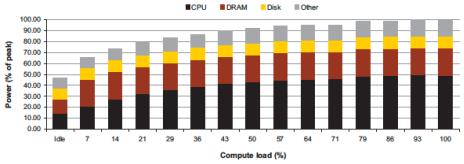


- Online service: Peak usage 2X off-peak

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

Power vs. Server Utilization



- Server power usage as load varies idle to 100%
- Uses ½ peak power when idle!
- Uses ⅔ 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
  - An power efficiency measure for WSC, *not* including efficiency of servers, networking gear
  - 1.0 = perfection

PUE in the Wild (2007)

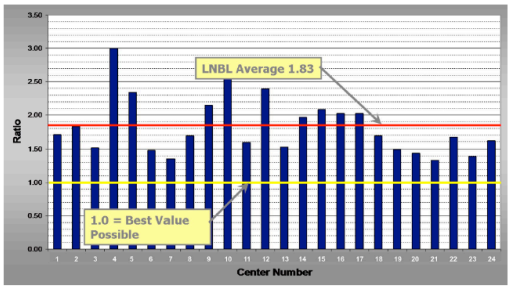
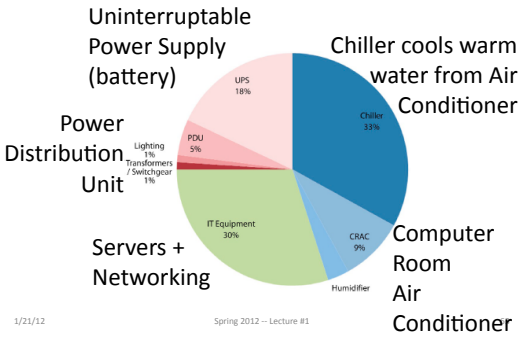
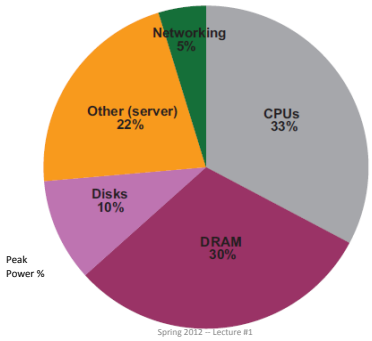


FIGURE 5.1: LBNL survey of the power usage efficiency of 24 datacenters, 2007 (Greenberg et al.)

High PUE: Where Does Power Go?



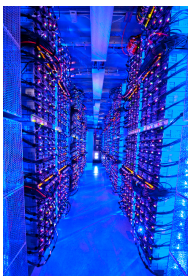
Servers and Networking Power Only



Containers in WSCs

Inside WSC

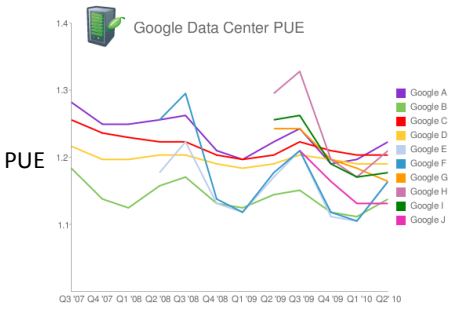
Inside Container



Google WSC A PUE: 1.24

1. Careful air flow handling
- Don't mix server hot air exhaust with cold air (separate warm aisle from cold aisle)
2. Elevated cold aisle temperatures
- 81°F instead of traditional 65°- 68°F
3. Measure vs. estimate PUE, publish PUE, and improve operation
- Note – subject of marketing
  - Average on a good day with artificial load (Facebook's 1.07) or real load for quarter (Google)

Google WSC PUE: Quarterly Avg



- www.google.com/corporate/green/datacenters/measuring.htm

Which statements are NOT true about Warehouse Scale Computing?

- ☐ Servers, IT equipment represent less than half of WSC power budget
- ☐ The Internet supplies the communication for SaaS
- ☐ Power Usage Effectiveness (PUE) also measures efficiency of the individual servers
- ☐ The goal of energy proportionality is energy usage should track equipment utilization

Summary

- CS61c: Learn 6 great ideas in computer architecture to enable high performance programming via parallelism, not just learn C
  1. Layers of Representation/Interpretation
  2. Moore's Law
  3. Principle of Locality/Memory Hierarchy
  4. Parallelism
  5. Performance Measurement and Improvement
  6. Dependability via Redundancy
- 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