CS 61C:
Great Ideas in Computer Architecture
Course Introduction

Instructor:
Randy H. Katz
http://inst.eecs.Berkeley.edu/~cs61c/fa13

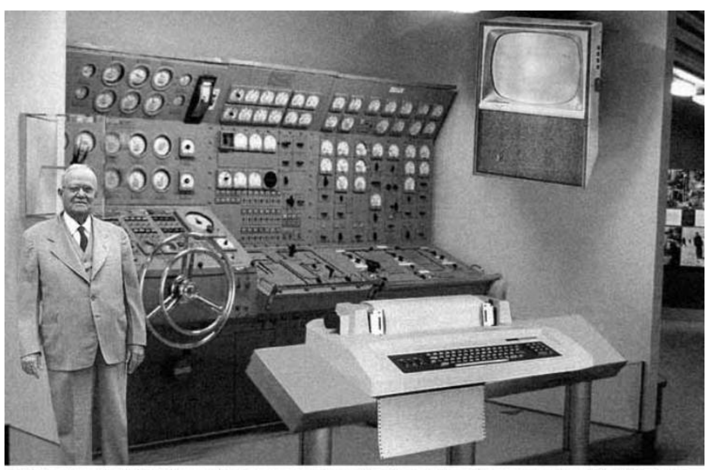
Agenda

- 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 about C Programming

- It's 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: "programming for performance"

Old Machine Structures

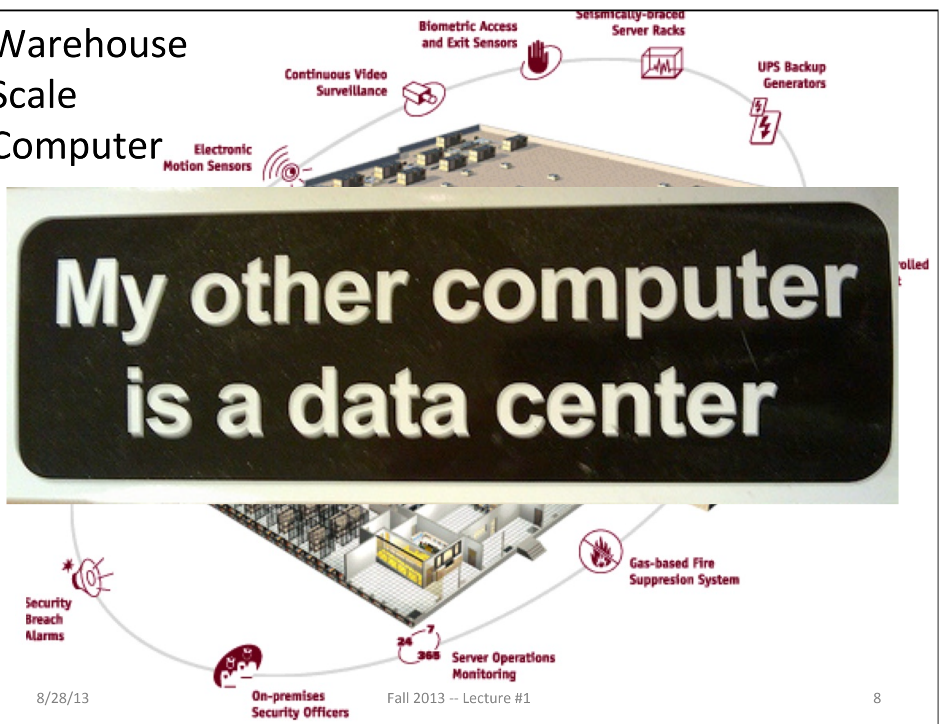


Scientists from the RAND Corporation have created this model to illustrate how a "home computer" could look like in the year 2004. However the needed technology will not be economically feasible for the average home. Also the scientists readily admit that the computer will require not yet invented technology to actually work, but 50 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.

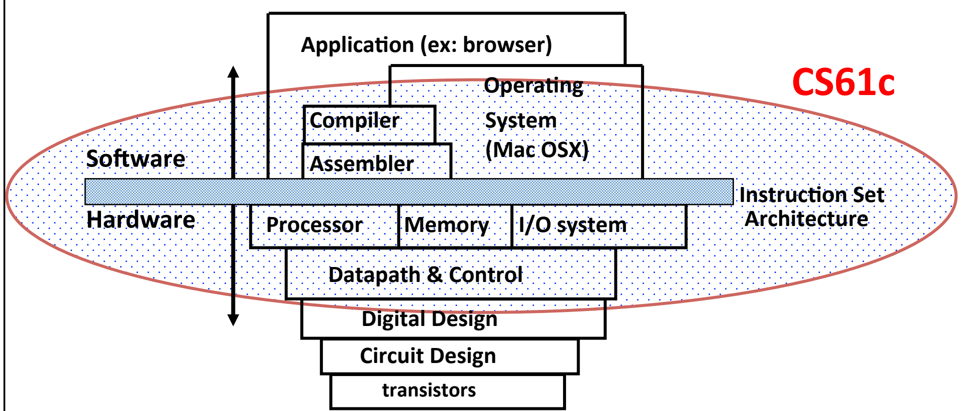
Personal Mobile Devices

New "Great Ideas"

Warehouse Scale Computer



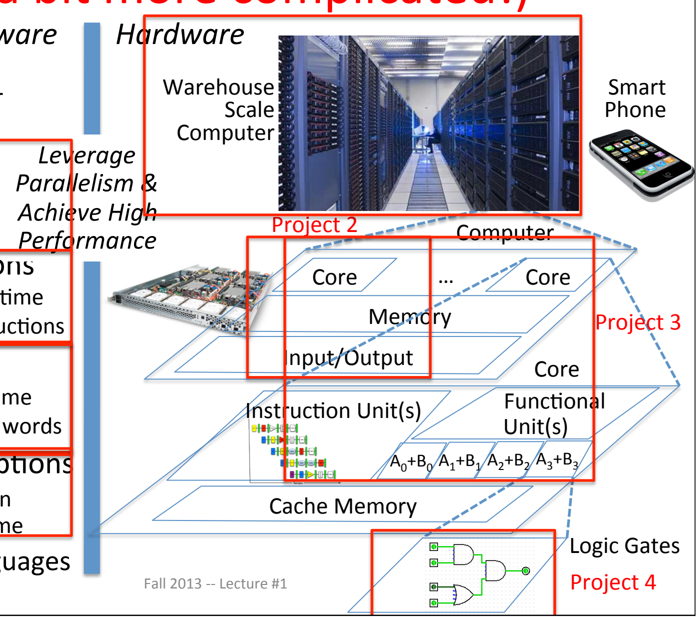
Old Machine Structures



New "Great Ideas"
(It's a bit more complicated!)

Software

- 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



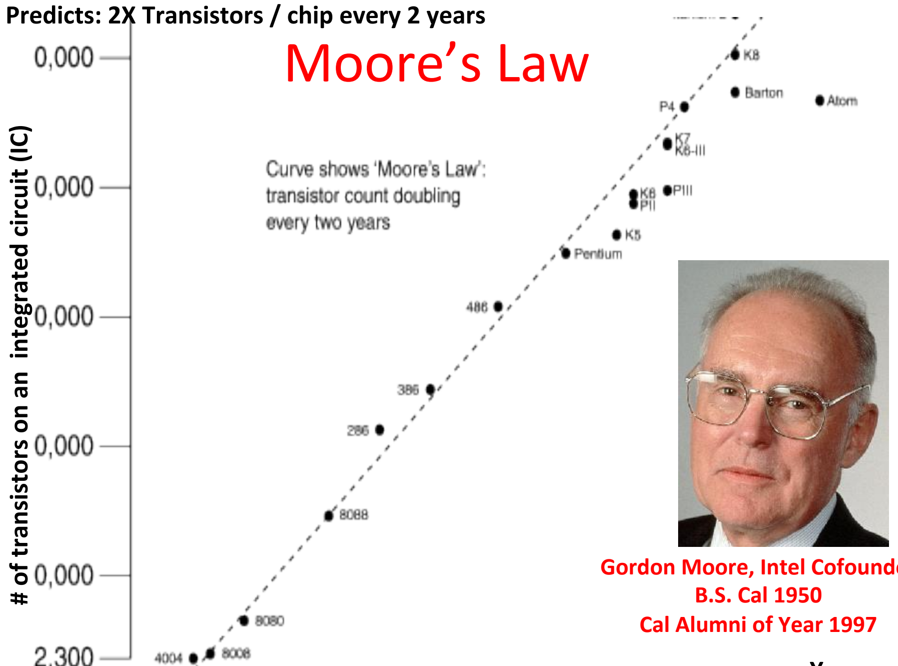
Great Ideas in Computer Architecture

1. Design for Moore's Law
2. Abstraction to Simplify Design
3. Make the Common Case Fast
4. Dependability via Redundancy
5. Memory Hierarchy
6. Performance via Parallelism/Pipelining/ Prediction

8/28/13 Fall 2013 -- Lecture #1 11

Predicts: 2X Transistors / chip every 2 years

Moore's Law



Gordon Moore, Intel Cofounder
B.S. Cal 1950
Cal Alumni of Year 1997

8/28/13 Fall 2013 -- Lecture #1 12

Abstraction via Layers of Representation

High Level Language Program (e.g., C)

Compiler

Assembly Language Program (e.g., MIPS)

Assembler

Machine Language Program (MIPS)

Machine Interpretation

Hardware Architecture Description (e.g., block diagrams)

Architecture Implementation

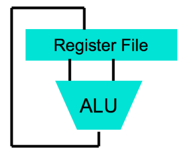
Logic Circuit Description (Circuit Schematic Diagrams)

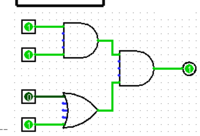
```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

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

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

```
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
```



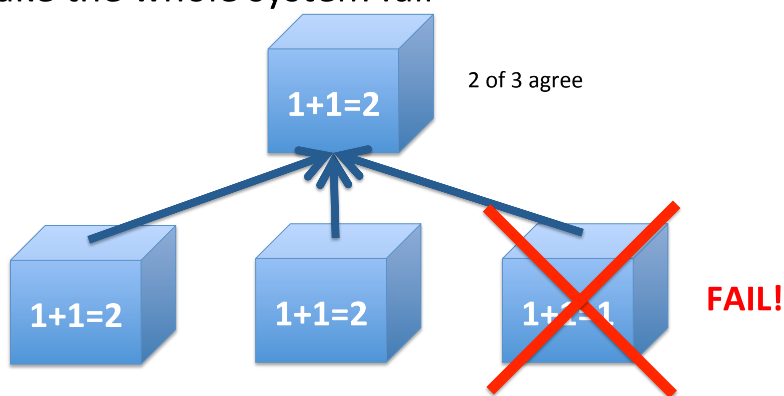


Make the Common Case Fast

- In making a design tradeoff, favor the common over the infrequent case
- Don't spend time optimizing code that is run infrequently
- Choose your performance metric and use measurement to determine the common case

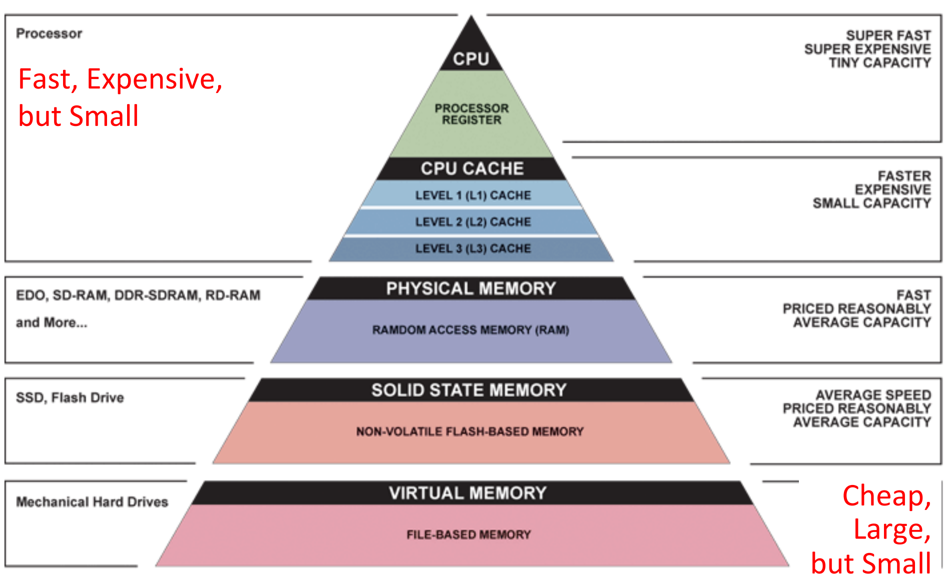
Dependability via Redundancy

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

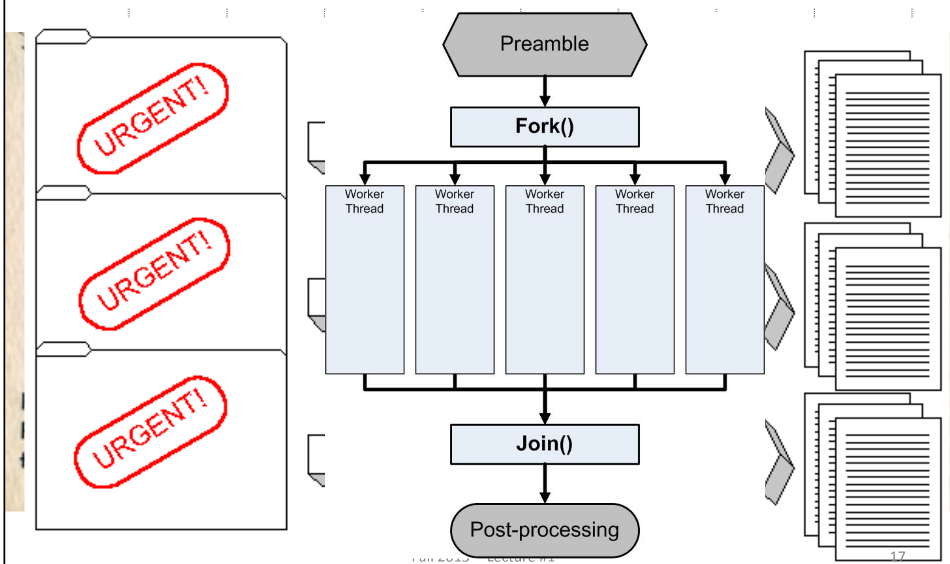


Increasing transistor density reduces the cost of redundancy

Memory Hierarchy



Parallelism/Pipelining/Prediction



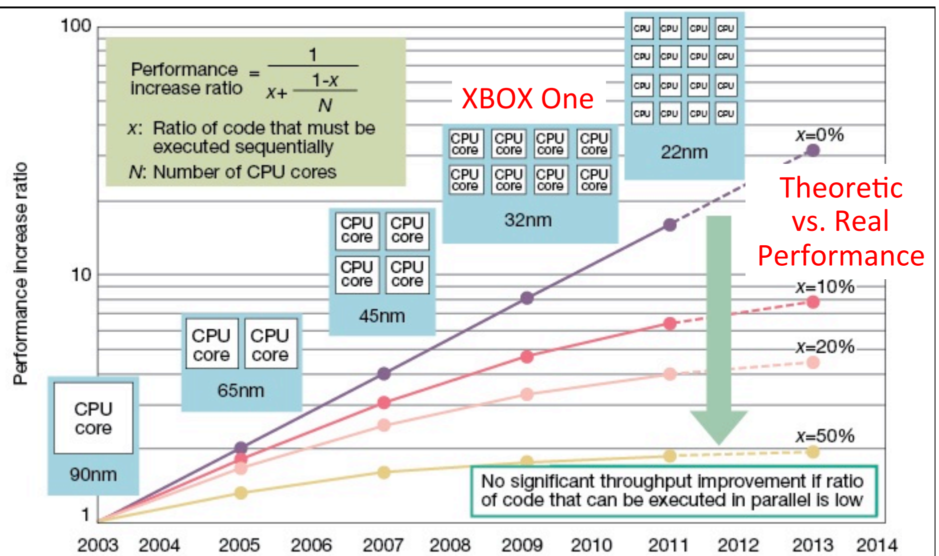
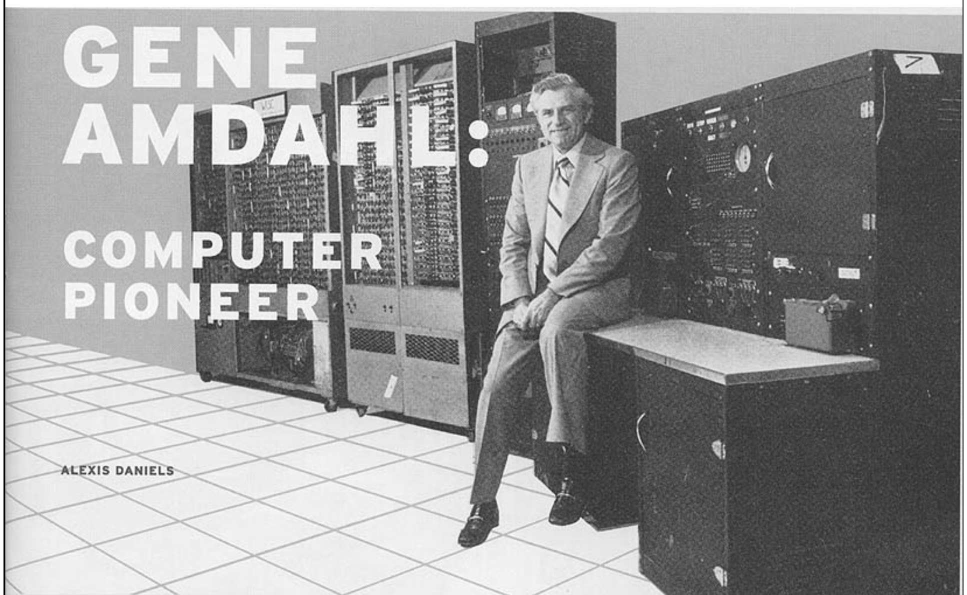
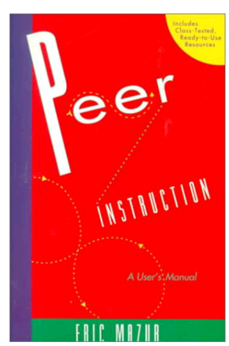


Fig 3 Amdahl's Law an Obstacle to Improved Performance Performance will not rise in the same proportion as the increase in CPU cores. Performance gains are limited by the ratio of software processing that must be executed sequentially. Amdahl's Law is a major obstacle in boosting multicore microprocessor 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.



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 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

21

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

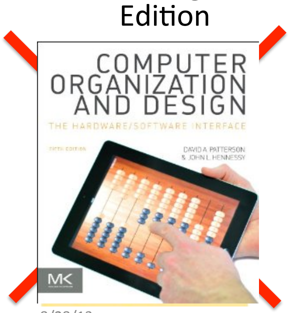
8/28/13 Fall 2013 -- Lecture #1 23

Course Information

- Course Web: http://inst.eecs.Berkeley.edu/~cs61c/fa13
- Instructor: Randy Katz
- Teaching Assistants: Kelvin Chou, Jeff Dong, Riyaz Faizullabhoy, Winston Hsu, Sagar Karandikar, Kevin Liston, Ajay Tripathi, Kevin Yeun, Sung Roa Yoon

Course Information

- Textbooks: Average 15 pages of reading/week
  - Barroso & Holzle, *The Datacenter as a Computer (free download from course page)*
  - Patterson & Hennessey, *Computer Organization and Design*, New 5th Edition (coming late September)
  - Kernighan & Ritchie, *The C Programming Language*, 2nd Edition



Chapters 1-3 available at Copy Central on Bancroft

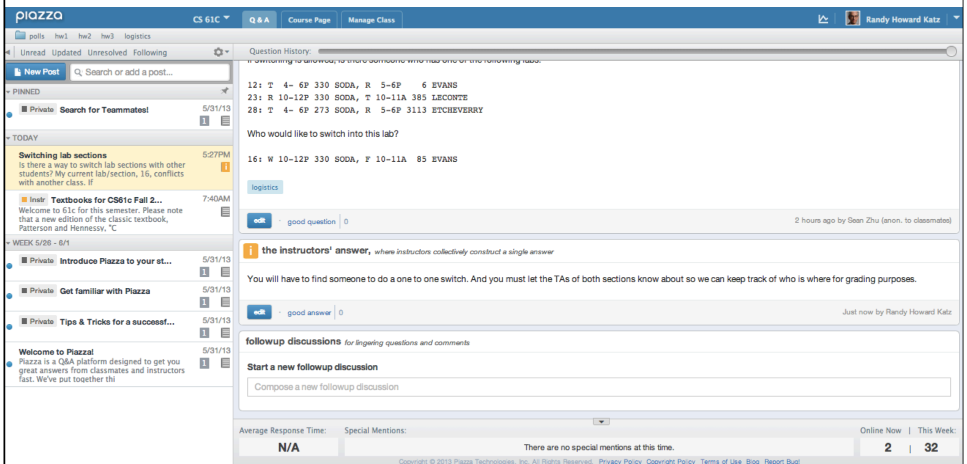
Course Organization

- 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 using cache blocking, SIMD, MIMD (OpenMP, with partner)
    4. Computer Processor Design (Logisim)
  - Extra Credit: Performance Improvement Competition, anything goes
  - Midterm (10%): **6-9 PM Th October 17**, Room TBD
  - Final (20%): **8-11 AM F December 20**

Do I Need to Know Java?

- Java used in Labs 2, 3; Project #1 (MapReduce)
- Prerequisites:
  - Official course catalog: "61A, along with either 61B or 61BL, or programming experience equivalent to that gained in 9C, 9F, or 9G"
  - Course web page: "The only prerequisite is that you have taken Computer Science 61B, or at least have solid experience with a C-based programming language"
  - *61a + Python alone is not sufficient*

Piazza for Course Q&A



Good answers enhance participation, reposting questions already asked yield *anti-participation* (aka negative participation)

TAs answer within 24 hours to encourage self-reliance and crowdsourced answers

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."
- Fall 2012: GPA 2.87
  22% A's, 52% B's, 21% C's, 2% D's, 3% F's
- Job/Intern Interviews: They grill you with technical questions, so it's what you say, not your GPA
  (61c gives you good stuff to say)

	Fall	Spring
2012	**2.87**	2.84
2011	2.72	**2.85**
2010	**2.81**	2.81
2009	2.71	2.81
2008	2.95	2.74

Labs and Discussions

- Waitlisted?
  - Limiting factor is lab space
  - You can add only if someone drops
- Want to switch?
  - Find someone in your desired lab section to swap with you who wants your lab, notify both TAs
  - Go to any discussion taught by your TA

Labs and Discussions

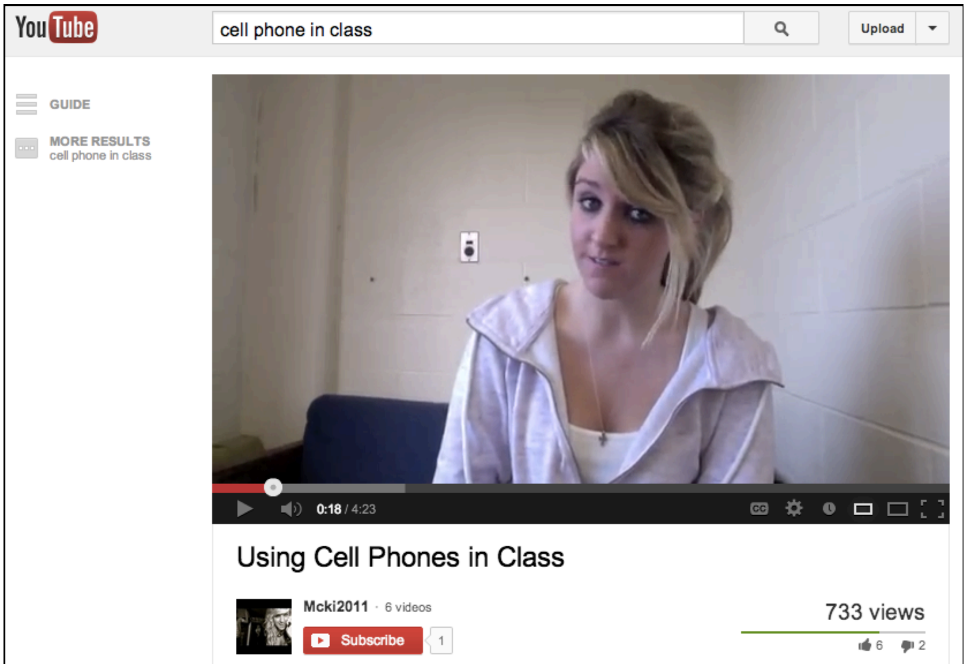
- Labs start week of 3 September
  - Project Partners: only Project 3 and extra credit, OK if partners mix sections but have same TA
- First homework assignment due Sunday, 8 September by 11:59:59 PM
  - Reading assignment on course page

Late Policy

- Assignments due Sundays at 11:59:59 PM
- There are no late homeworks (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"

Assignments and Independent Work

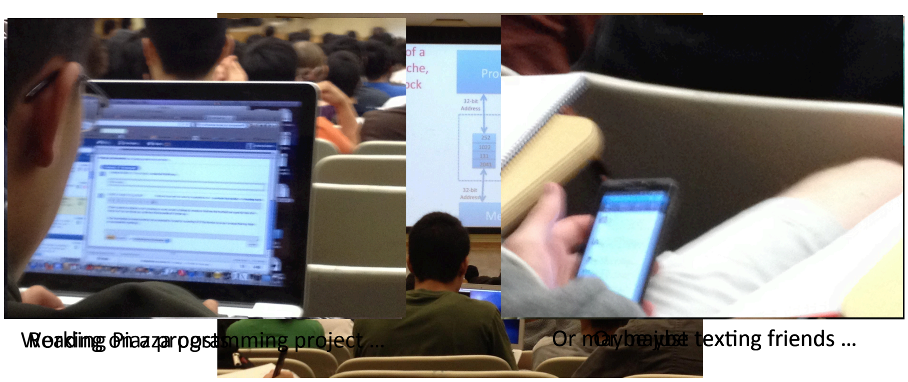
- With the exception of laboratories and assignments that explicitly permit you to work in groups, all homeworks and projects are to be YOUR work and your work ALONE.
- You are encouraged to discuss your assignments with other students, and extra credit will be assigned to students who help others, particularly 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.
- It is NOT acceptable to hire someone to do your project for you (we know all about those programming for hire websites, and we do scan them!).
- 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 catch people every time we teach 61C!)



8/28/13 Fall 2013 -- Lecture #1 34

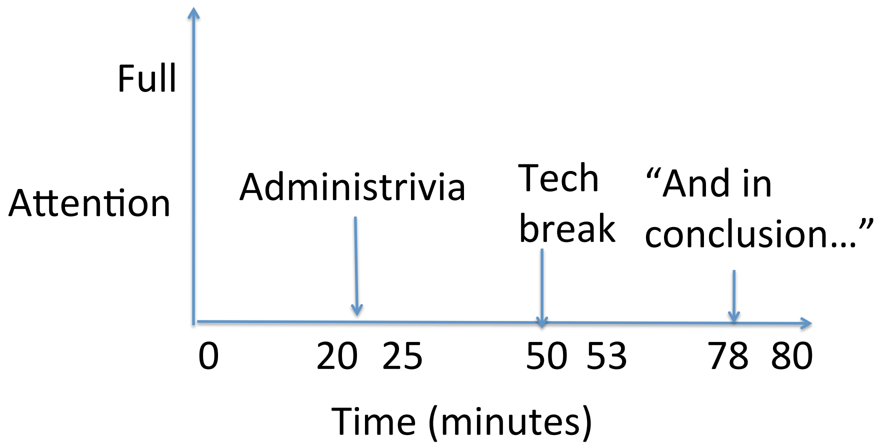
Phones and Laptops in Lecture

- "I like to take notes and follow along the lecture on my laptop …"

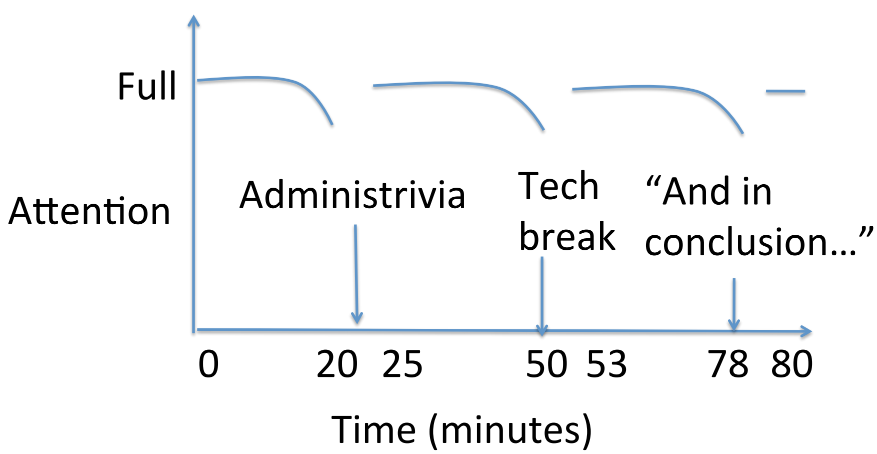


Fall 2013 -- Lecture #1 35

Architecture of a Lecture



Architecture of a Lecture



Question: Which statements are TRUE about this class?

- ☐ The midterm is Tuesday October 15 during class (12:30-2)
- ☐ The midterm is Thursday October 17 in the evening (6-9PM)
- ☐ It's OK to book airline tickets before December 20; Katz will surely let me take final early
- ☐ It is OK to buy the 4th edition of Computer Organization and Design

38

Question: Which statements are TRUE about this class?

- ☐ The midterm is Tuesday October 17 during class (12:30-2)
- ☐ The midterm is Thursday October 17 in the evening (6-9PM)
- ☐ It's OK to book airline tickets before December 20; Katz will surely let me take final early
- ☐ It is OK to buy the 4th edition of Computer Organization and Design

39


Young Tech Sees Itself in Microsoft's Ballmer

James Best Jr./The New York Times

By QUENTIN HARDY
Published: August 25, 2013

SAN FRANCISCO — As a young executive at Microsoft, Steven A. Ballmer helped topple older, slower-moving technology giants like the Digital Equipment Corporation, Wang and Novell.

These days, it is Microsoft's turn to fend off the upstarts as it struggles to compete in a computing world that is increasingly mobile and based in a "cloud" of Internet-connected computers to which many customers gain access at the same time. It's all

Related

Needed at Microsoft: A Catch-Up Artist (August 25, 2013)

Pogue's Posts Blog: How Ballmer Missed the Tidal Shifts in Tech (August 24, 2013)

FACEBOOK
TWITTER
GOOGLE+
SAVE
E-MAIL
SHARE
PRINT
SINGLE PAGE

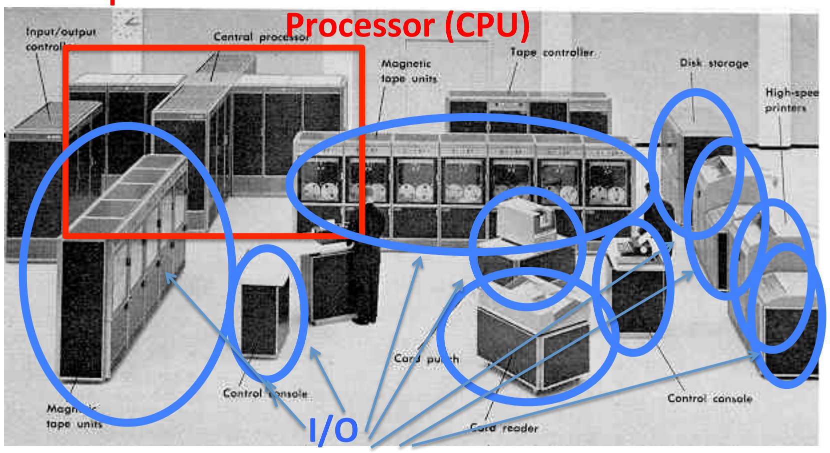
40

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

8/28/13 Fall 2013 -- Lecture #1 41

Computer Eras: Mainframe 1950s-60s



"Big Iron": IBM, UNIVAC, … build $1M computers for businesses => timesharing OS (Multics)

Minicomputer Eras: 1970s

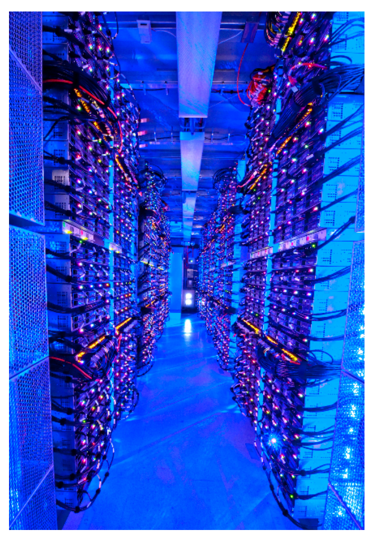
Using integrated circuits, Digital, HP… build $10k computers for labs, universities => UNIX OS

PC Era: Mid 1980s - Mid 2000s

Using microprocessors, Apple, IBM, … build $1k computers for individuals => Windows OS, Linux

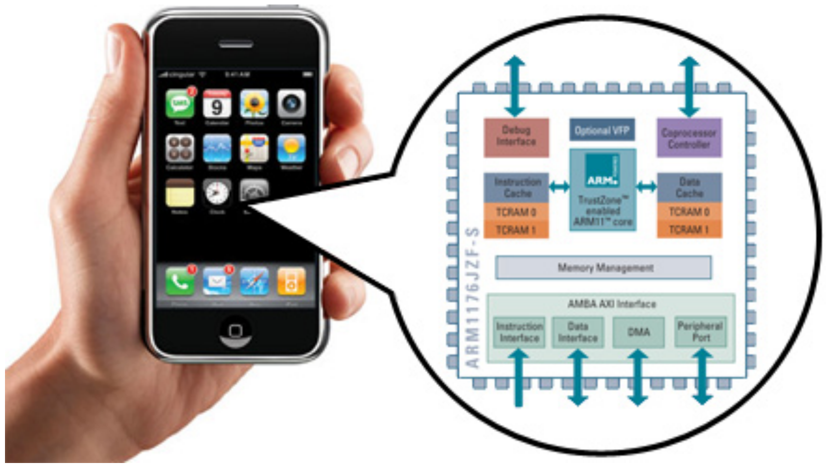
PostPC Era: Late 2000s - ??

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



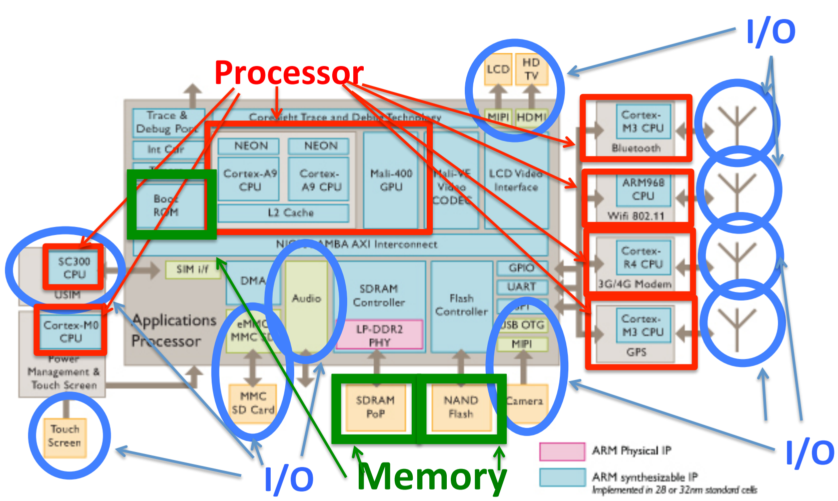
Cloud Computing: Using Local Area Networks, Amazon, Google, … build $200M Warehouse Scale Computers with 100,000 servers for Internet Services for PMDs
=> MapReduce/Hadoop

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:

  2012 Revenue share ($56.5 billion): Intel 65.3%, *Qualcomm 9.4%*, *Samsung 8.2%* , AMD 6.4%

  2012 Processor Share: approx. 6B ARM vs. 200M 80x86 (30X more)

Microprocessors Revenue

Leading MPU Suppliers ($M)

2012 Rank	Company	2011	2012	Percent Change	Percent Marketshare	Main Product Lines
1	Intel	37,435	36,892	-1%	65.3%	x86 PC, server MPUs
2	Qualcomm	4,152	5,322	28%	9.4%	ARM mobile app processors
3	Samsung (+Apple)*	2,614	4,664	78%	8.2%	ARM mobile app processors
4	AMD	4,552	3,605	-21%	6.4%	x86 PC, server MPUs
5	Freescale	1,210	1,070	-12%	1.9%	ARM and embedded MPUs
6	Nvidia	591	764	29%	1.4%	ARM mobile app processors
7	TI	510	565	11%	1.0%	ARM mobile app processors
8	ST-Ericsson**	660	540	-18%	1.0%	ARM mobile app processors
9	Broadcom	295	345	17%	0.6%	ARM mobile app processors
10	MediaTek	280	325	16%	0.6%	ARM mobile app processors

*Includes Apple's custom processors made by Samsung's foundry business.

Source: IC Insights **Cellphone IC joint venture to be dissovled by STMicroelectronics and Ericsson by 3Q13.

Agenda

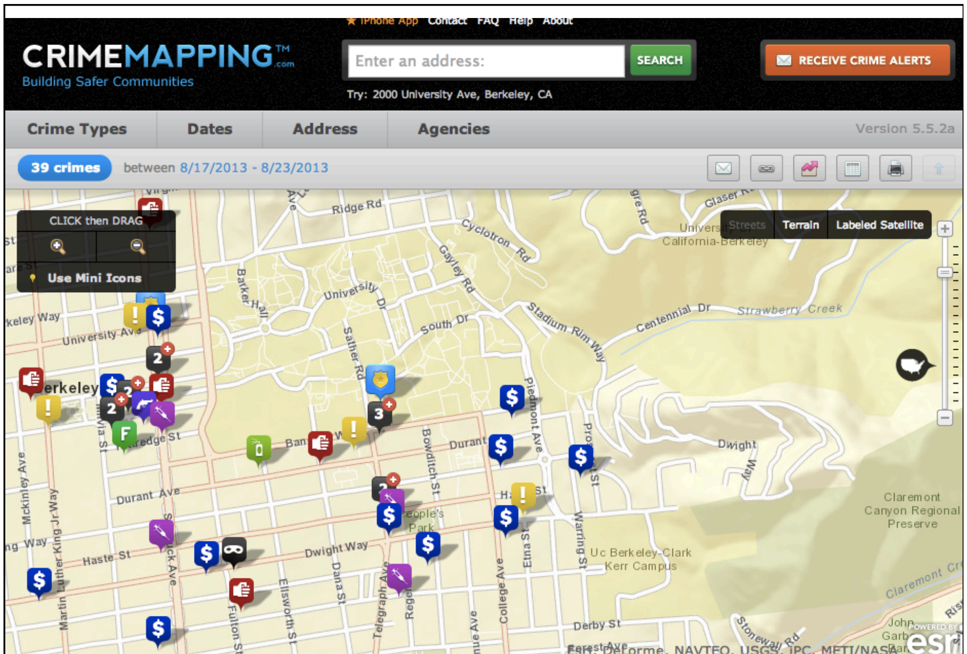
- Great Ideas in Computer Architecture
- Administrivia
- PostPC Era: From Phones to Datacenters
- Software as a Service
- Cloud Computing
- Technology Break
- Warehouse Scale Computers in Depth

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



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

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

The Big Switch: Cloud Computing

"A hundred years ago, companies stopped generating their own power with steam engines and dynamos and plugged into the newly built electric grid. The cheap power pumped out by electric utilities didn't just change how businesses operate. It set off a chain reaction of economic and social transformations that brought the modern world into existence. Today, a similar revolution is under way. Hooked up to the Internet's global computing grid, massive information-processing plants have begun pumping data and software code into our homes and businesses. This time, it's computing that's turning into a utility."

8/28/13 Fall 2013 -- Lecture #1 56

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

8/28/13 Fall 2013 -- Lecture #1 59

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

60

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

61

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

62

Which statements are NOT true about SaaS and Cloud Computing?

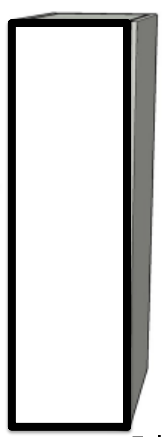
- ☐ Clusters are collections of commodity servers connected by LAN switches
- ☐ The Internet supplies the communication for SaaS
- ☐ Cloud computing uses HW clusters + SW layer using redundancy for dependability
- ☐ Private datacenters could match cost of Warehouse Scale Computers if they just purchased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

65

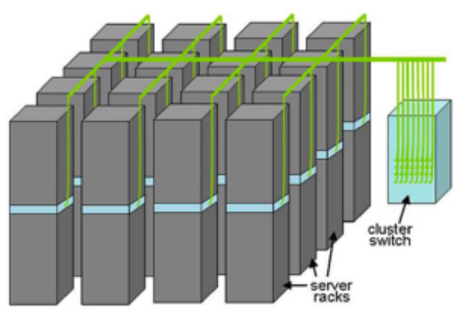
E.g., Google's Oregon WSC

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



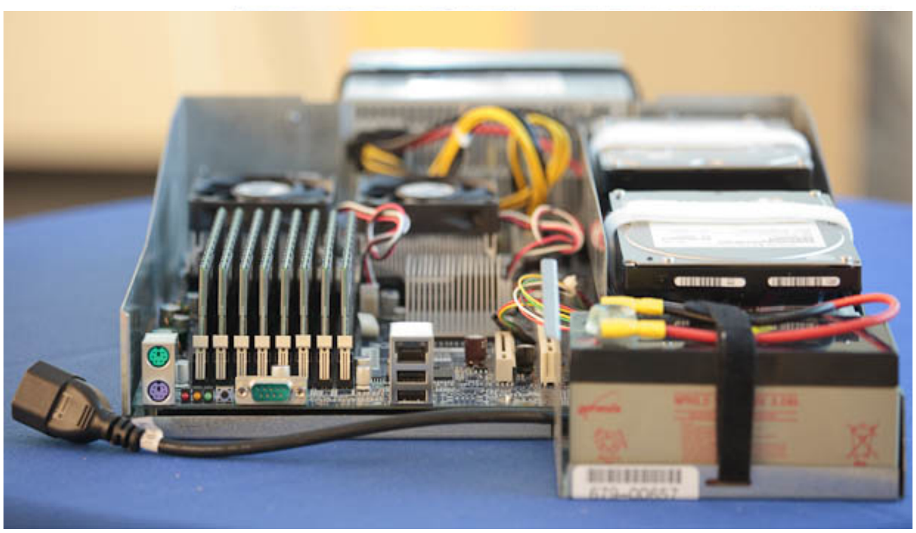
7 foot Rack: 40-80 servers + Ethernet local area network (1-10 Gbps) switch in middle ("rack switch")



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

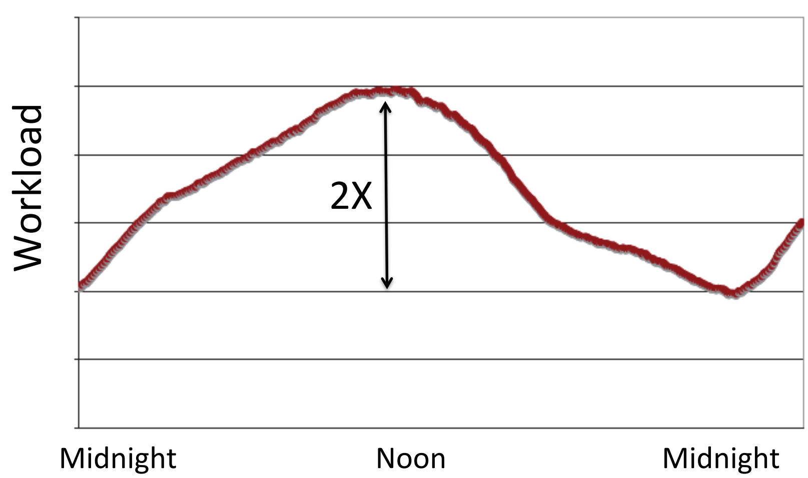


Coping with Performance in Array

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



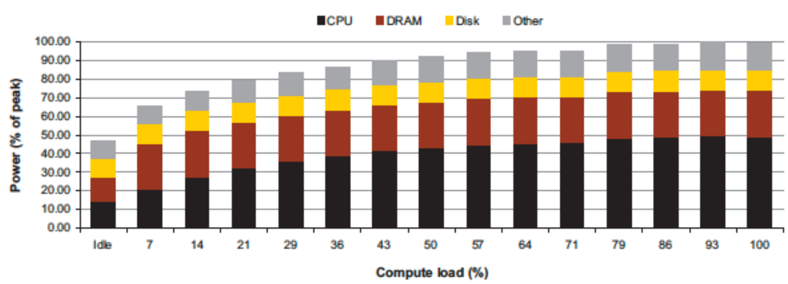
- Online service: Peak usage 2X off-peak

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

8/28/13 Fall 2013 -- Lecture #1 73

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

8/28/13 Fall 2013 -- Lecture #1 74

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
  - Power efficiency measure for WSC, *not* including efficiency of servers, networking gear
  - 1.0 = perfection

PUE in the Wild (2007)

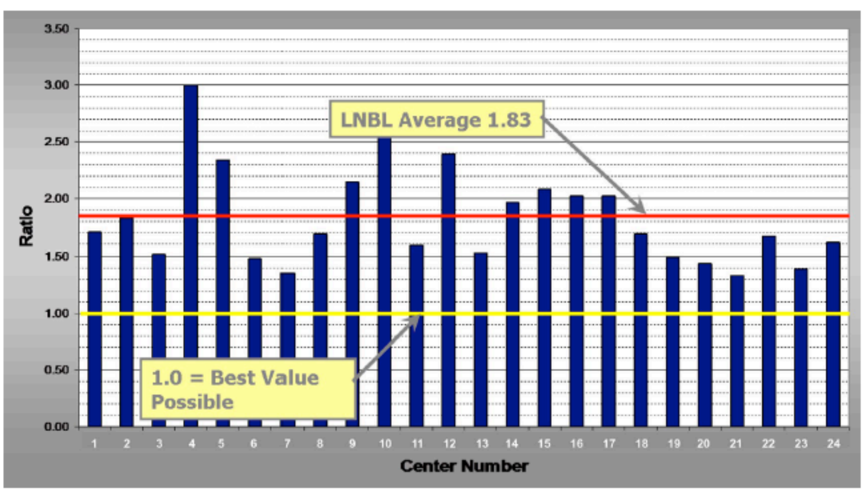


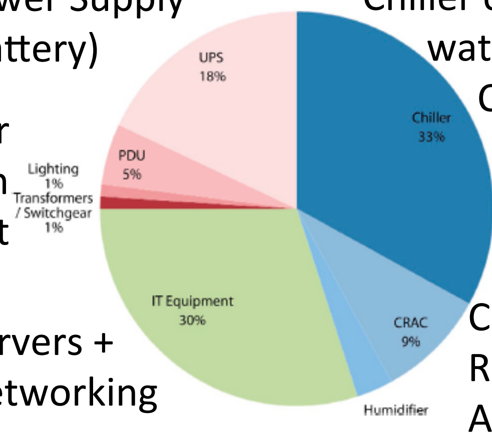
FIGURE 5.1: LBNL survey of the power usage efficiency of 24 datacenters, 2007 (Greenberg et al.)

High PUE: Where Does Power Go?

Uninterruptable Power Supply (battery)

Chiller cools warm water from Air Conditioner

Power Distribution Unit

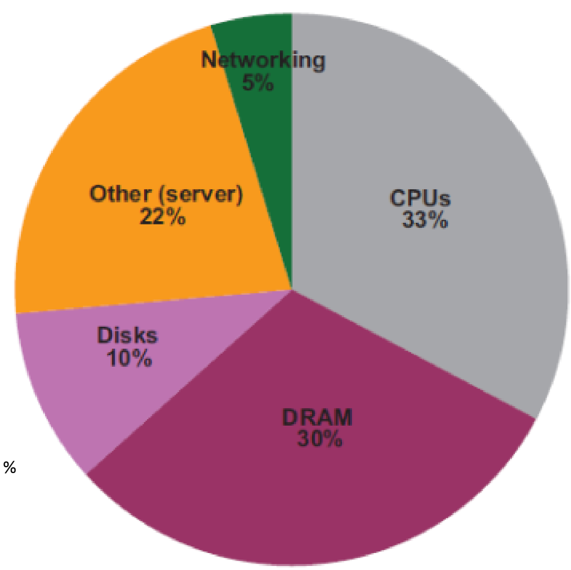


Servers + Networking

Computer Room Air Conditioner

8/28/13 Fall 2013 -- Lecture #1

Servers and Networking Power Only



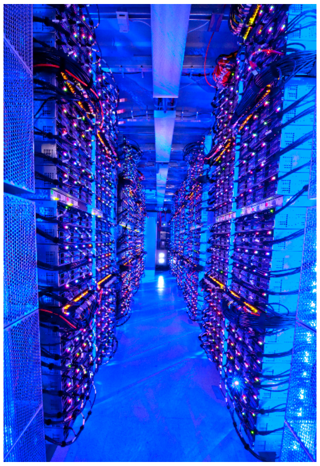
Peak Power %

8/28/13 78

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)

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

82

And In Conclusion …

- CS61c: Learn about great ideas in computer architecture to enable high performance programming via parallelism, not just learn C
  1. Design for Moore's Law
  2. Abstraction to Simplify Design
  3. Make the Common Case Fast
  4. Dependability via Redundancy
  5. Memory Hierarchy
  6. Performance via Parallelism/Pipelining/Predictiony
- 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

8/28/13 Fall 2013 -- Lecture #1 84