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

New School CS61c

Personal Mobile Devices

Warehouse Scale Computer
Old-School Machine Structures

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

- 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

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

Great Idea #3: Principle of Locality/Memory Hierarchy

Predicts: 2X Transistors / chip every 2 years

#2: Moore’s Law

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

Architecture Implementation

Logic Circuit Description
(Circuit Schematic Diagrams)

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

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

Great Idea #6: Dependability via Redundancy

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

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

---

Course Information

- Course Web: [http://inst.eecs.berkeley.edu/~cs61c/sp12sp](http://inst.eecs.berkeley.edu/~cs61c/sp12sp)
- Instructor:
  - Dave Patterson
- Teaching Assistants:
  - Rimas Azziensis, Scott Beamer, Alan Christopher, Eric Liang, Paul Ruan, Ian Vorseggern
- Textbooks:
  - Barroso & Holste, *The Datacenter as a Computer* (free download from web page)
- 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

---

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, MIMO (OpenMP), with partner)
  4. Computer Processor Design (LogicSim)
  - 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](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

<table>
<thead>
<tr>
<th></th>
<th>Fall</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>2.72</td>
<td>2.85</td>
</tr>
<tr>
<td>2010</td>
<td>2.81</td>
<td>2.81</td>
</tr>
<tr>
<td>2009</td>
<td>2.71</td>
<td>2.81</td>
</tr>
<tr>
<td>2008</td>
<td>2.95</td>
<td>2.74</td>
</tr>
<tr>
<td>2007</td>
<td>2.67</td>
<td>2.76</td>
</tr>
</tbody>
</table>

---

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 22nd by 11:59:59 PM
- There is reading assignment as well on course page
It seems like he can no longer be fully focusing on his family. His wife, Brenda, complains, forgets things like dinner plans, and he has trouble getting from his electronic gadgets. He of data. Even after unplugging, he craves the connection to continue to struggle with the effects of the deluge of e-mail messages of his life landing in his inbox a few days after sending them. While he managed to salvage the $1.3 million deal he was writing the computer code he was writing.

The resulting distractions can have deadly consequences, as when cellphone-wielding drivers and train-engineer caused wrecks. And for millions of people like Mr. Campbell, these urges can affect jobs and costs on creativity and deep thought, interrupting work and family life.

While many people say multitasking makes them more productive, research shows otherwise. Many multitaskers actually have more trouble focusing and shifting their attention, scientists say, and they experience more stress. And scientists are discovering that even after the multitasking ends, transport thinking analysis of lower levels, making it take longer to recall. In other words, try not your brain off of computers.

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 8, 2010, by MaD Richtel

SAN FRANCISCO – When one of the most important e-mail messages of his life landed in his inbox a few years ago, Scott 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,’” Mr. Campbell said. “It’s kind of hard to miss an e-mail like that, but I did.”

The message had arrived by fax and as an electronic file: 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 apologizes, he cross the stimulation he gets from his electronic gadgets. He often cannot focus on anything but e-mails, even when he is writing a letter to his family. His wife, Brenda, complains. Scientists say he can no longer be fully in the moment.

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 the flood of information.

These days to a primitive urge to respond to immediate opportunities and threats. The stimulation provides excitement – a dopamine spurt that researchers say can be addictive. In its absence, people feel anxious.

The resulting distractions can have deadly consequences, as when cellphone-wielding drivers and train-engineer caused wrecks. And for millions of people like Mr. Campbell, these urges can affect jobs and costs on creativity and deep thought, interrupting work and family life.

While many people say multitasking makes them more productive, research shows otherwise. Many multitaskers actually have more trouble focusing and shifting their attention, scientists say, and they experience more stress. And scientists are discovering that even after the multitasking ends, transport thinking analysis of lower levels, making it take longer to recall. In other words, try not your brain off of computers.

Architecture of a Lecture

Full Attention

Time (minutes)

0  20  25  50  53  78  80

Administrivia  Tech break  “And in conclusion…”

Architecture of a Lecture

Full Attention

Time (minutes)

0  20  25  50  53  78  80

Administrivia  Tech break  “And in conclusion…”
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)

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 1SS 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

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 100s servers
     – Careful selection of identical HW/SW
     – Virtual Machine Monitors simplify operation
  4. Dependability via extensive redundancy

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

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
2012 AWS Instances & Prices

<table>
<thead>
<tr>
<th>Instance</th>
<th>Per Hour</th>
<th>CPU Cores</th>
<th>Memory (GB)</th>
<th>Disk (GB)</th>
<th>Address Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Small</td>
<td>$0.085</td>
<td>1.0</td>
<td>1.0</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Standard Large</td>
<td>$0.340</td>
<td>4.0</td>
<td>4.0</td>
<td>2.00</td>
<td>15.0</td>
</tr>
<tr>
<td>High Memory Extra Large</td>
<td>$0.500</td>
<td>6.5</td>
<td>6.5</td>
<td>3.25</td>
<td>17.1</td>
</tr>
<tr>
<td>High Memory Quadruple Extra Large</td>
<td>$0.400</td>
<td>26.2</td>
<td>26.2</td>
<td>8.75</td>
<td>68.4</td>
</tr>
<tr>
<td>High-CPU Medium</td>
<td>$0.270</td>
<td>2.8</td>
<td>2.8</td>
<td>2.50</td>
<td>7.0</td>
</tr>
<tr>
<td>High-CPU Extra Large</td>
<td>$0.360</td>
<td>6.8</td>
<td>6.8</td>
<td>5.96</td>
<td>96.8</td>
</tr>
<tr>
<td>Eight Extra Large</td>
<td>$0.500</td>
<td>15.3</td>
<td>15.3</td>
<td>15.99</td>
<td>22.9</td>
</tr>
<tr>
<td>Eight Extra Large</td>
<td>$0.400</td>
<td>28.2</td>
<td>28.2</td>
<td>21.75</td>
<td>60.5</td>
</tr>
</tbody>
</table>

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

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?

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

E.g., Google’s Oregon WSC

Server (in rack format): 1½ inches high “1U”, x 19 inches x 16-20 inches: 8 cores, 16 GB DRAM, 4x1 TB disk

Array (aka cluster): 16-32 server racks + larger local area network switch (“array switch”) 10X faster => cost 100X: cost f(N^2)

Equipment Inside a WSC
**Server, Rack, Array**

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

<table>
<thead>
<tr>
<th></th>
<th>Local</th>
<th>Rack</th>
<th>Array</th>
</tr>
</thead>
<tbody>
<tr>
<td>Racks</td>
<td></td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>Servers</td>
<td>1</td>
<td>80</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 (GB)</td>
<td>4,000</td>
<td>320,000</td>
<td>9,600,000</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>

**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

**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 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
  - An 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?

Uninterruptable Power Supply (battery)
Chiller cools warm water from Air Conditioner
Servers + Networking

Servers and Networking Power Only

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)
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

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