CS 61C:
Great Ideas in Computer Architecture
What's Next and Course Review

Instructors:
Krste Asanović and Randy H. Katz
http://inst.eecs.Berkeley.edu/~cs61c/fa17

Agenda

- FireBox: A Hardware Building Block for the 2020 WSC
- Course Review
- Project 3 Performance Competition
- Course On-line Evaluations

Agenda

- FireBox: A Hardware Building Block for the 2020 WSC
- Course Review
- Project 3 Performance Competition
- Course On-line Evaluations

Warehouse-Scale Computers (WSCs)

- Computing migrating to two extremes:
  - Mobile and the "Swarm" (Internet of Things)
  - The "Cloud"
- Most mobile/swarm apps supported by cloud compute
- All data backed up in cloud
- Ongoing demand for ever more powerful WSCs

Three WSC Generations

1. ~2000: Commercial Off-The-Shelf (COTS) computers, switches, & racks
2. ~2010: Custom computers, switches, & racks but build from COTS chips
3. ~2020: Custom computers, switches, & racks using custom chips

- Moving from horizontal Linux/x86 model to vertical integration WSC_OS/WSC_SoC (System-on-Chip) model
- Increasing impact of open-source model across generations

WSC: Most Critical Tolerance

- Old Conventional Wisdom: Fault tolerance is critical for Warehouse-Scale Computer (WSC)
  - Build reliable whole from less reliable parts
- New Conventional Wisdom: Tail tolerance also critical for WSC, ***Slow = failure***
  - Build predictable response whole from less predictable parts

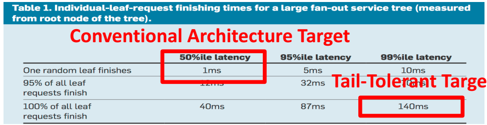


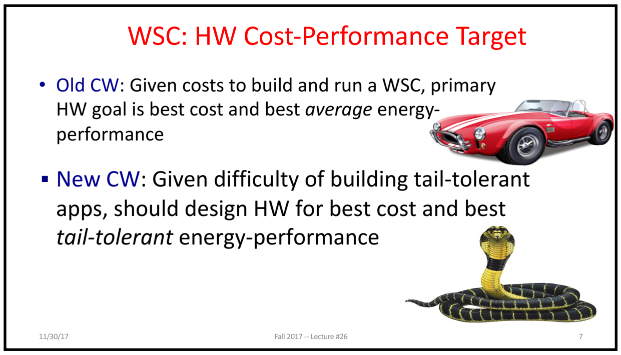
Table 1. Individual-leaf-request finishing times for a large fan-out service tree (measured from root node of the tree).

Conventional Architecture Target

	50%ile latency	95%ile latency	99%ile latency
One random leaf finishes	1ms	5ms	10ms
95% of all leaf requests finish	12ms	32ms	70ms
100% of all leaf requests finish	40ms	87ms	140ms

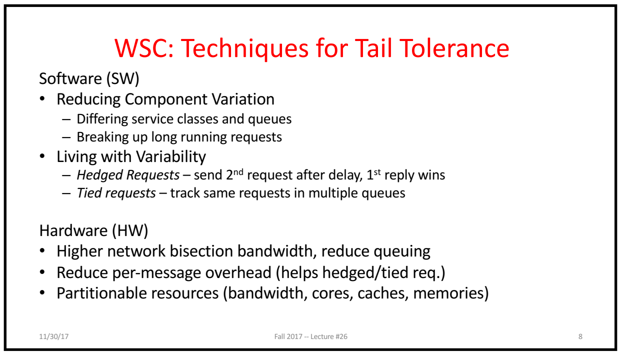
Tail-Tolerant Target

Dean, J., & Barroso, L. A. (2013). The tail at scale. *CACM*, 56(2), 74-80.



WSC: HW Cost-Performance Target

- Old CW: Given costs to build and run a WSC, primary HW goal is best cost and best *average* energy-performance
- New CW: Given difficulty of building tail-tolerant apps, should design HW for best cost and best *tail-tolerant* energy-performance



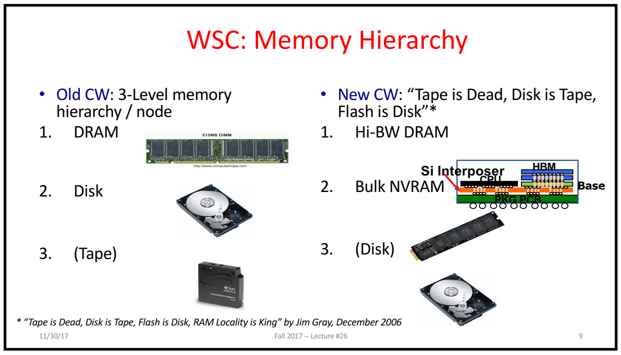
WSC: Techniques for Tail Tolerance

Software (SW)

- Reducing Component Variation
  - Differing service classes and queues
  - Breaking up long running requests
- Living with Variability
  - *Hedged Requests* – send 2nd request after delay, 1st reply wins
  - *Tied requests* – track same requests in multiple queues

Hardware (HW)

- Higher network bisection bandwidth, reduce queuing
- Reduce per-message overhead (helps hedged/tied req.)
- Partitionable resources (bandwidth, cores, caches, memories)



WSC: Memory Hierarchy

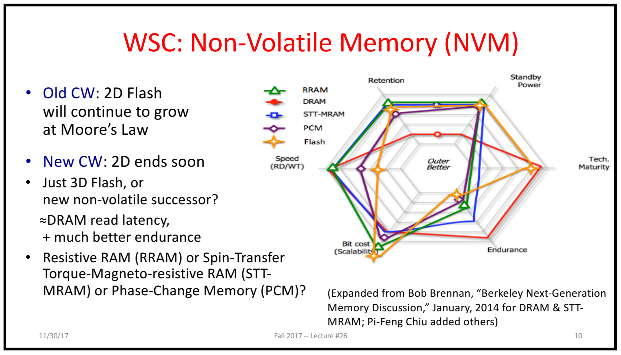
- Old CW: 3-Level memory hierarchy / node

1. DRAM
2. Disk
3. (Tape)

- New CW: "Tape is Dead, Disk is Tape, Flash is Disk"*

1. Hi-BW DRAM
2. Bulk NVRAM
3. (Disk)

* *"Tape is Dead, Disk is Tape, Flash is Disk, RAM Locality is King" by Jim Gray, December 2006*



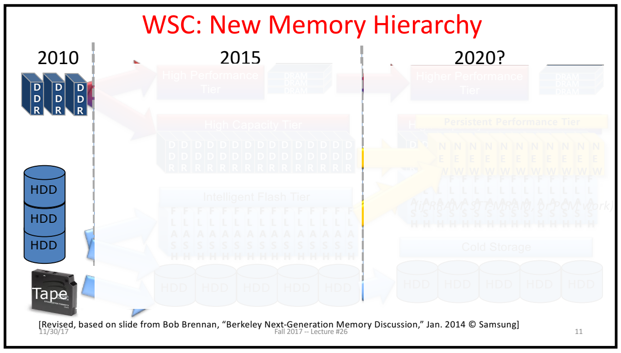
WSC: Non-Volatile Memory (NVM)

- Old CW: 2D Flash will continue to grow at Moore's Law
- New CW: 2D ends soon
- Just 3D Flash, or new non-volatile successor?

≈DRAM read latency,
+ much better endurance

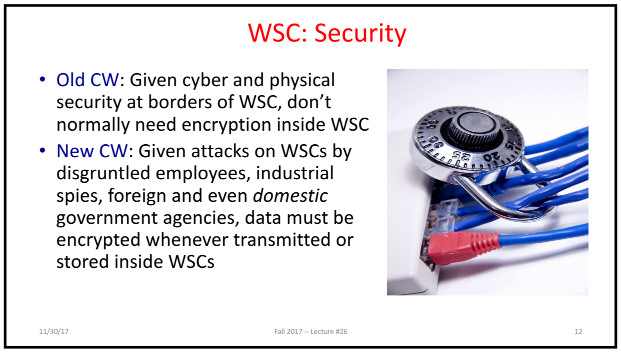
- Resistive RAM (RRAM) or Spin-Transfer Torque-Magneto-resistive RAM (STT-MRAM) or Phase-Change Memory (PCM)?

(Expanded from Bob Brennan, "Berkeley Next-Generation Memory Discussion," January, 2014 for DRAM & STT-MRAM; Pi-Feng Chiu added others)



WSC: New Memory Hierarchy

[Revised, based on slide from Bob Brennan, "Berkeley Next-Generation Memory Discussion," Jan. 2014 © Samsung]



WSC: Security

- Old CW: Given cyber and physical security at borders of WSC, don't normally need encryption inside WSC
- New CW: Given attacks on WSCs by disgruntled employees, industrial spies, foreign and even *domestic* government agencies, data must be encrypted whenever transmitted or stored inside WSCs

WSC: Moore's Law

- Old CW: Moore's Law, each 18-month technology generation, transistor performance/energy improves, cost/transistor decreases
- New CW: generations slowing to 3 year -> 5+ year, transistor performance/energy slight improvement, co increases!

2020: Moore's Law has ended for logic, SRAM, & DRAM (Maybe 3D Flash & new NVM continues?)

WSC: Hardware Design

- Old CW: Build WSC from cheap Commercial Off-The-Shelf (COTS) Components, which run LAMP stack
  - Microprocessors, racks, NICs, rack switches, array switches, …

- New CW: Build WSC from custom components, which support SOA, tail tolerance, fault tolerance detection recovery prediction, …
  - Custom high radix switches, custom racks and cooling, System on a Chip (SoC) integrating processors & NIC

Why Custom Chips in 2020?

- Without transistor scaling, improvements in system capability have to come above transistor-level
  - More specialized hardware
- WSCs proliferate @ $100M/WSC
  - Economically sound to divert some $ if yield more cost-performance-energy effective chips
- Good news: when scaling stops, custom chip costs drop
  - Amortize investments in capital equipment, CAD tools, libraries, training, … over decades vs. 18 months
- New HW description languages supporting parameterized generators improve productivity and reduce design cost
  - E.g., Stanford Genesis2; Berkeley's Chisel, based on Scala

Berkeley RISC-V ISA

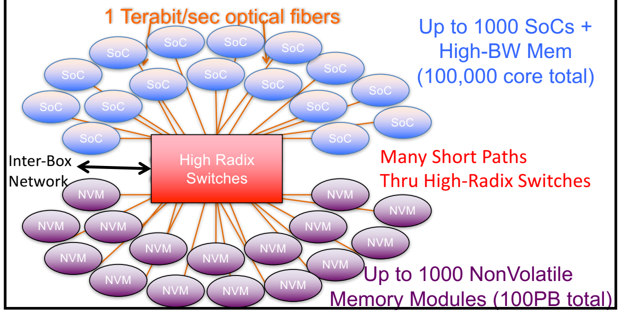
`www.riscv.org`

- A new completely open ISA
  - Already runs GCC, Linux, glibc, LLVM, …
  - RV32, RV64, and RV128 variants for 32b, 64b, and 128b address spaces defined
- Base ISA only 40 integer instructions, but supports compiler, linker, OS, etc.
- Extensions provide full general-purpose ISA, including IEEE-754/2008 floating-point
- Comparable ISA-level metrics to other RISCs
- Designed for extension, customization
- Eight 64-bit silicon prototype implementations completed at Berkeley so far (45nm, 28nm)

Open-Source & WSCs

- 1st generation WSC leveraged open-source software
- 2nd generation WSC also pushing open-source board designs (OpenCompute), OpenFlow API for networking
- 3rd generation – open-source chip designs?
  - FireBox WSC chip generator

FireBox



1 Terabit/sec optical fibers

Up to 1000 SoCs + High-BW Mem (100,000 core total)

Inter-Box Network

High Radix Switches

Many Short Paths Thru High-Radix Switches

Up to 1000 NonVolatile Memory Modules (100PB total)

FireBox Big Bets

- Reduce OpEx, manage units of 1,000+ sockets
- Support huge in-memory (NVM) databases directly
- Massive network bandwidth to simplify software
- Re-engineered software/processor/NIC/network for low-overhead messaging between cores, low-latency high-bandwidth bulk memory access
- Data always encrypted on fiber and in bulk storage
- Custom SoC with hardware to support above features
- Open-source hardware generator to allow customization within WSC SoC template

FireBox SoC Highlights

- ~100 (homogenous) cores per SoC
  - Simplify resource management, software model
- Each core has vector processor++ (>> SIMD)
  - "General-purpose specialization"
- Uses RISC-V instruction set
  - Open source, virtualizable, modern 64-bit RISC ISA
  - GCC/LLVM compilers, runs Linux
- Cache coherent on-chip so only need one OS per SoC
  - Core/outer caches can be split into local/global scratchpad/cache to improve tail tolerance
- Compress/Encrypt engine so reduce size for storage and transmission yet always encrypted outside node
- Implemented as parameterized Chisel chip generator
  - Easy to add custom application accelerators, tune architectural parameters

FireBox Hardware Highlights

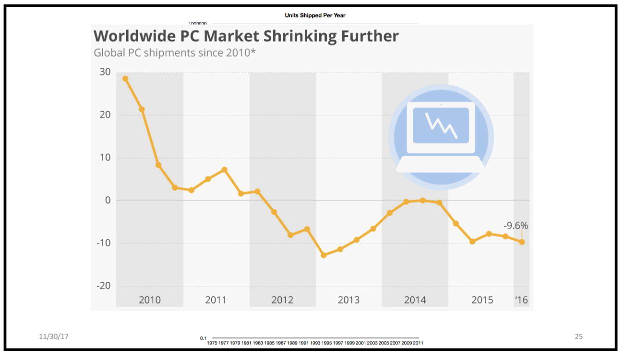
- 8-32 DRAM chips on interposer for high BW
  - 32Gb chips give 32-128GB DRAM capacity/node
  - 500GB/s DRAM bandwidth
- Message Passing is RPC: can return/throw exceptions
  - ≈20 ns overhead for send or receive, including SW
  - ≈100ns latency to access Bulk Memory: ≈2X DRAM latency
- Error Detection/Correction on Bulk Memory
- No Disks in Standard Box; special Disk Boxes instead
  - Disk Boxes for Cold Storage
- ≈50 KW/box
- ≈35KW for 1000 sockets
  - 20W for socket cores, 10W for socket I/O, 5W for local DRAM
- ≈15KW for Bulk NVRAM + Crossbar switch
  - 10^{-12} joule/bit transfer => Terabit/sec/Watt

Revised FireBox Vision, 2017

- Not too many mispredicts – we were surprisingly mostly on track
- By 2015, we realized that flash was going to dominate, so bulk memory will be DRAM+Flash for forseeable future
  - Other NVM technology very slow to market, unclear value proposition
  - Flash arrays became huge business
- Custom hardware in datacenter happened faster than expected
  - Microsoft Catapult, Brainwave; Google TPU/TPU2; Amazon F1 instances
- RISC-V took off far faster than expected
- Monolithic photonics becoming credible
- From special-purpose FPGA boards, to F1 to run WSC simulations
- Services as unit of work in datacenter still/more popular
- Security still a big problem

Agenda

- FireBox: A Hardware Building Block for the 2020 WSC
- Course Review
- Project 3 Performance Competition
- Course On-line Evaluations

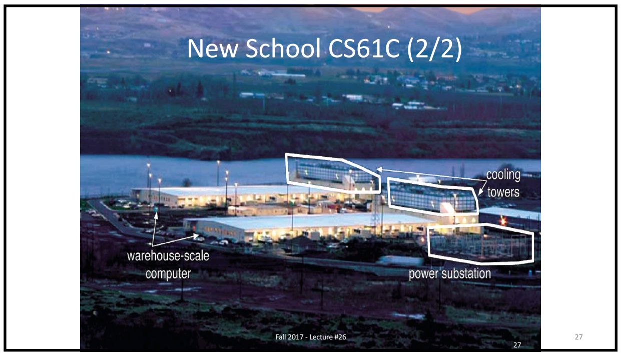


Worldwide PC Market Shrinking Further

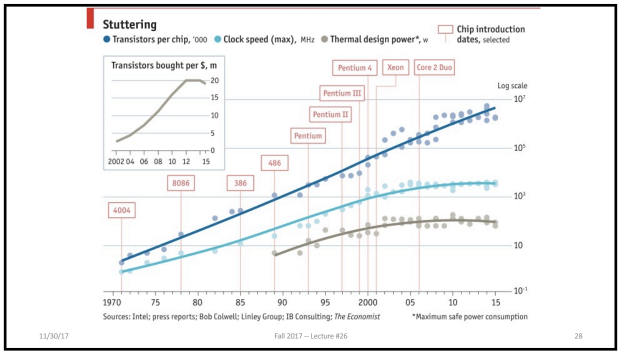
Global PC shipments since 2010*

New School CS61C (1/2)

Personal
Mobile
Devices



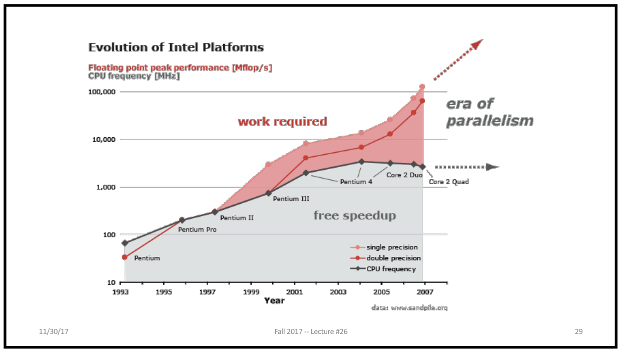
New School CS61C (2/2)



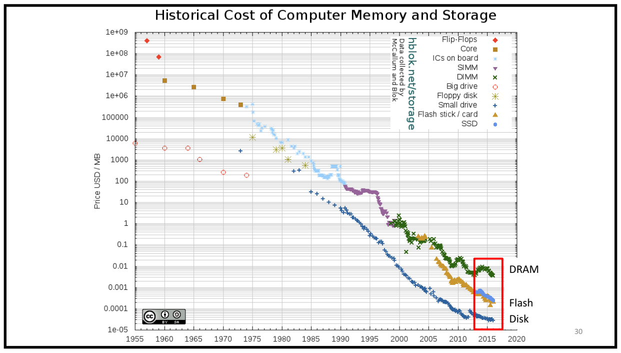
Stuttering

Sources: Intel; press reports; Bob Colwell; Linley Group; IB Consulting; *The Economist*

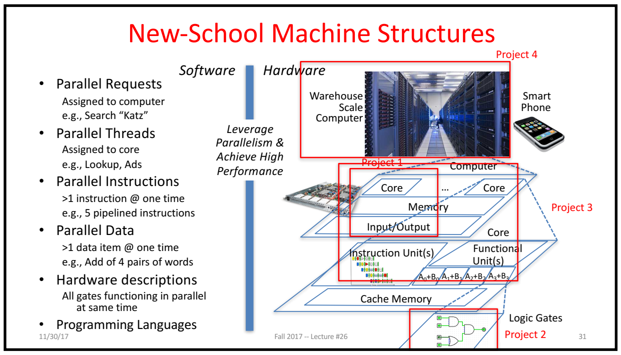
*Maximum safe power consumption



Evolution of Intel Platforms



Historical Cost of Computer Memory and Storage

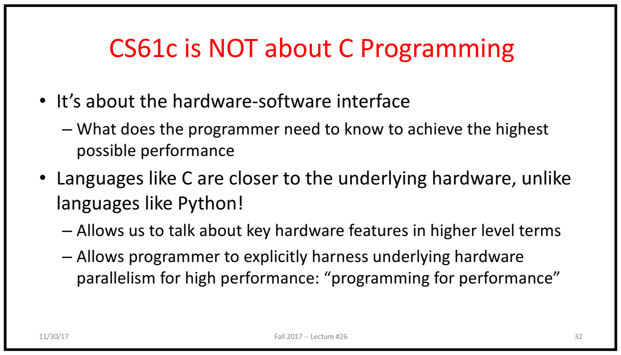


New-School Machine Structures

Software | *Hardware*

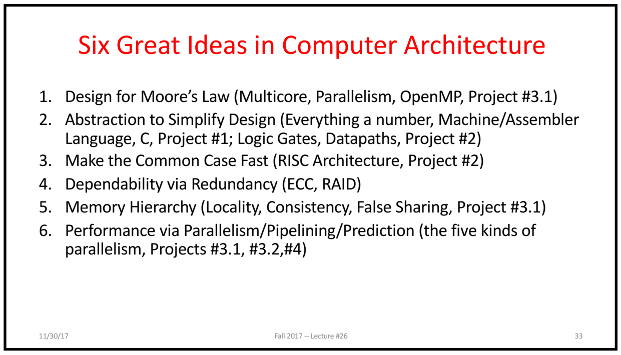
- Parallel Requests
  Assigned to computer
  e.g., Search "Katz"
- Parallel Threads
  Assigned to core
  e.g., Lookup, Ads
- Parallel Instructions
  >1 instruction @ one time
  e.g., 5 pipelined instructions
- Parallel Data
  >1 data item @ one time
  e.g., Add of 4 pairs of words
- Hardware descriptions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

Leverage Parallelism & Achieve High Performance



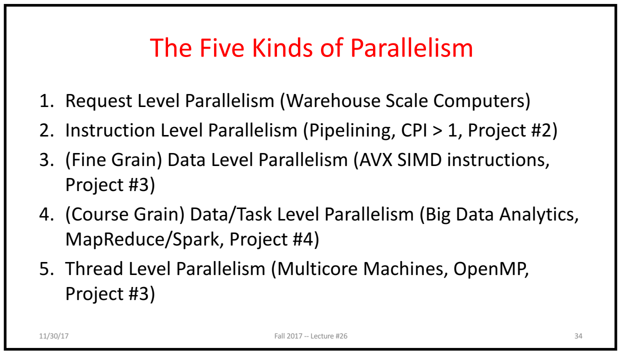
CS61c is NOT about C Programming

- 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"



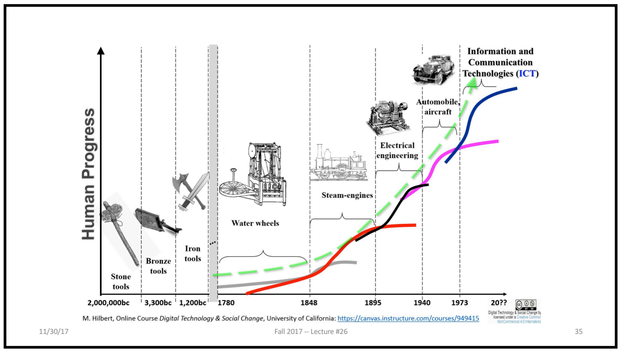
Six Great Ideas in Computer Architecture

1. Design for Moore's Law (Multicore, Parallelism, OpenMP, Project #3.1)
2. Abstraction to Simplify Design (Everything a number, Machine/Assembler Language, C, Project #1; Logic Gates, Datapaths, Project #2)
3. Make the Common Case Fast (RISC Architecture, Project #2)
4. Dependability via Redundancy (ECC, RAID)
5. Memory Hierarchy (Locality, Consistency, False Sharing, Project #3.1)
6. Performance via Parallelism/Pipelining/Prediction (the five kinds of parallelism, Projects #3.1, #3.2,#4)



The Five Kinds of Parallelism

1. Request Level Parallelism (Warehouse Scale Computers)
2. Instruction Level Parallelism (Pipelining, CPI > 1, Project #2)
3. (Fine Grain) Data Level Parallelism (AVX SIMD instructions, Project #3)
4. (Course Grain) Data/Task Level Parallelism (Big Data Analytics, MapReduce/Spark, Project #4)
5. Thread Level Parallelism (Multicore Machines, OpenMP, Project #3)



M. Hilbert, Online Course *Digital Technology & Social Change*, University of California: https://canvas.instructure.com/courses/949415

Prof. Katz's First Computer -- 1970

https://en.wikipedia.org/wiki/Programma_101
First commercial desktop computer? $3200 (1966 dollars)
240 bytes of memory; jump and jump conditional statements

Prof. Katz's Second Computer -- 1971

- 25-bit word (1 sign bit plus 8 octal digits), single accumulator (A Register)
- 4096 words (magnetic drum, 3400 RPM)
- 100+ instructions
  - Opcode [23:18], Count [17:12], Address [11:0]
  - Add/Subtract: 1.5 ms
  - Multiply/Divide: 8 ms
  - Ld/St: 9 ms
- Paper tape input/output: 50 characters per second

http://bitsavers.informatik.uni-stuttgart.de/pdf/digiac/3080/Digiac_3080_Brochure_1964.pdf

Prof. Katz's Third Computer -- 1972

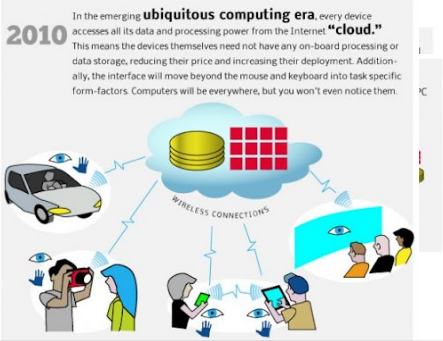
https://en.wikipedia.org/wiki/IBM_System/360

TechNewsDaily™ www.technewsdaily.com

Your Computer is Going Away

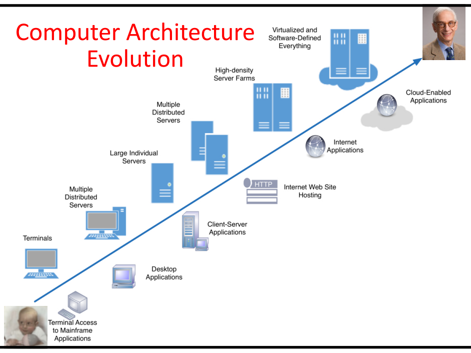
Soon, your smartphone, TiVo, laptop, television -- all of your current gadgets -- will be obsolete. The future is "ubiquitous computing." Think Google Docs, but on every screen you use, running every program you use -- every device drawing from the same pool of data and processing power. Here's how we got to this point.

Currently, all digital devices include these four components:



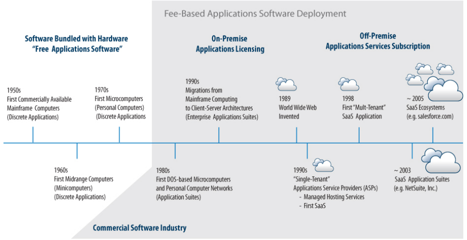
2010 In the emerging **ubiquitous computing era**, every device accesses all its data and processing power from the Internet **"cloud."** This means the devices themselves need not have any on-board processing or data storage, reducing their price and increasing their deployment. Additionally, the interface will move beyond the mouse and keyboard into task specific form-factors. Computers will be everywhere, but you won't even notice them.

Computer Architecture Evolution



Virtualized and Software-Defined Everything
High-density Server Farms
Multiple Distributed Servers
Large Individual Servers
Multiple Distributed Servers
Terminals
Terminal Access to Mainframe Applications
Desktop Applications
Client-Server Applications
HTTP
Internet Web Site Hosting
Internet Applications
Cloud-Enabled Applications

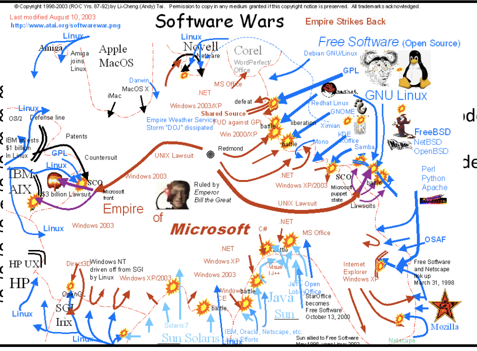
Software Application Evolution



- 198
- 198
- 198
- 199
- 199 rele
- 199
- 199
- 199
- 199

ıde

ıde



Administrivia (1/3)

- Final exam: the last Thursday examination slot!
  - 14 December, 7-10 PM, Wheeler Auditorium (for everybody!)
  - Three double sided Cheat Sheets (Mid #1, Mid #2, material since Mid #2)
  - Contact us about conflicts
  - Review **Lectures** and **Book** with eye on the important concepts of the course, e.g., the Great Ideas in Computer Architecture and the Different Kinds of Parallelism
- Electronic Course Evaluations this week! See https://course-evaluations.berkeley.edu

Administrivia (2/3)

2 Final Review Sessions			
Led by:	Time	Location	Style:
Tutors	Saturday Dec 2, 11-1pm	Cory 540AB	OH, small group
TAs	Friday Dec 8, 5-8pm	VLSB 2050	Lecture style, problem-solving

- Lab 11 (Spark) is due any day this week
- Lab 13 (VM) is due any day next week
- Last Guerrilla Session is next Tuesday, 7-9 PM @ Cory 293
  - Will review the most difficult topics this semester

Administrivia (3/3)

- Project 3-2 Contest Results!
  - 3rd Place: Neelesh Dodda and Matthew Trepte at **138x speedup**
  - 2nd Place: Mohammadreza Mottaghi at **263x speedup**
  - 1st Place: Alvin Hsu and Jonathan Xia at **323x speedup!**
- Project 3 grades will be entered by the end of today!

CS61c In The News!

WESTERN DIGITAL TO ACCELERATE THE FUTURE OF NEXT-GENERATION COMPUTING ARCHITECTURES FOR BIG DATA AND FAST DATA ENVIRONMENTS

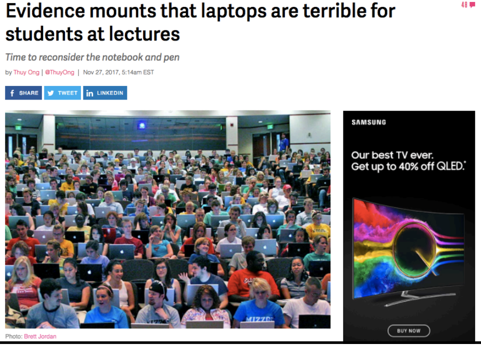
San Jose and Milpitas, Ca - November 28, 2017

Company to Transition Consumption of Over One Billion Cores Per Year to RISC-V to Drive Momentum of Open Source Processors for Data Center and Edge Computing

Western Digital Corp. (NASDAQ: WDC) announced today at the 7th RISC-V Workshop that the company intends to lead the industry transition toward open, purpose-built compute architectures to meet the increasingly diverse application needs of a data-centric world. In his keynote address, Western Digital's Chief Technology Officer Martin Fink expressed the company's commitment to help lead the advancement of data-centric compute environments through the work of the RISC-V Foundation. RISC-V is an open and scalable compute architecture that will enable the diversity of Big Data and Fast Data applications and workloads proliferating in core cloud data centers and in remote and mobile systems at the edge. Western Digital's leadership role in the RISC-V initiative is significant in that it aims to accelerate the advancement of the technology and the surrounding ecosystem by transitioning its own consumption of processors – over one billion cores per year – to RISC-V.

Agenda

- FireBox: A Hardware Building Block for the 2020 WSC
- Course Review
- Project 3 Performance Competition
- Course On-line Evaluations



Evidence mounts that laptops are terrible for students at lectures

Time to reconsider the notebook and pen

by Thuy Ong | @ThuyOng | Nov 27, 2017, 5:14am EST

Photo: Brett Jordan

What Next?

- EECS151 (spring/fall) if you liked digital systems design
- CS152 (spring) if you liked computer architecture
- CS162 (spring/fall) operating systems and system programming
- CS168 (fall) computer networks

And, in Conclusion …

- As the field changes, cs61c had to change too!
- It is still about the software-hardware interface
  - Programming for performance!
  - Parallelism: Task-, Thread-, Instruction-, and Data- MapReduce, OpenMP, C, AVX intrinsics
  - Understanding the memory hierarchy and its impact on application performance
- Interviewers ask what you did this semester!

Agenda

- FireBox: A Hardware Building Block for the 2020 WSC
- Course Review
- Project 3 Performance Competition
- Course On-line Evaluations:
  - HKN Evaluations Today and Electronic Course Evaluations until end of RRR Week! See https://course-evaluations.berkeley.edu