CS61C
Memory Hierarchy Introduction and Eight Week Review

Lecture 16

March 12, 1999

Dave Patterson
(http.cs.berkeley.edu/~patterson)

www-inst.eecs.berkeley.edu/~cs61c/schedule.html

Review 1/1

- Magnetic Disks continue rapid advance: 60%/yr capacity, 40%/yr bandwidth, slow on seek, rotation improvements, MB/$ improving 100%/yr?
  - Designs to fit high volume form factor
  - Quoted seek times too conservative, data rates too optimistic for use in system
- RAID
  - Higher performance with more disk arms per $
  - Adds availability option at modest cost

Outline

- Memory Hierarchy Analogy
- Illusion of Large, Fast, Cheap Memory
- Principle of Locality
- Terms
- Who manages each level of Hierarchy?
- Administrivia, “Computer in the News”
- Big Ideas on 61C: What we’ve seen so far
- Conclusion

Hierarchy Analogy: Term Paper in Library

- Working on paper in library at a desk
- Option 1: Every time need a book
  - Leave desk to go to shelves (or stacks)
  - Find the book
  - Bring one book back to desk
  - Read section interested in
  - When done with section, leave desk and go to shelves carrying book
  - Put the book back on shelf
  - Return to desk to work
  - Next time need a book, go to first step

Hierarcgy Analogy: Library

- Option 2: Every time need a book
  - Leave some books on desk after fetching them
  - Only go to shelves when need a new book
  - When go to shelves, bring back related books in case you need them; sometimes you’ll need to return books not used recently to make space for new books on desk
  - Return to desk to work
  - When done, replace books on shelves, carrying as many as you can per trip
- Illusion: whole library on your desktop

Technology Trends

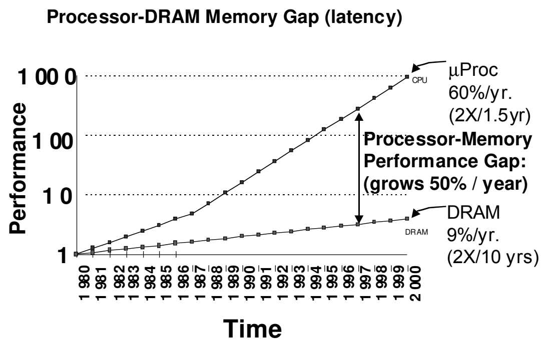
	Capacity	Speed (latency)
Processor	--	4x in 3 yrs
DRAM:	4x in 3 yrs	2x in 10 yrs
Disk:	4x in 3 yrs	2x in 10 yrs

DRAM

Year	Size	Cycle Time
1980	64 Kb	250 ns
1983	256 Kb	220 ns
1986	1 Mb	190 ns
1989	4 Mb	165 ns
1993	16 Mb	145 ns
1997	64 Mb	120 ns

1000:1! **2:1!**

Who Cares About the Memory Hierarchy?

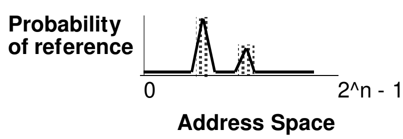


The Goal: Illusion of large, fast, cheap memory

- Fact: Large memories are slow, fast memories are small
- How do we create a memory that is large, cheap and fast (most of the time)?
- Hierarchy of Levels
  - Similar to Principle of Abstraction: hide details of multiple levels

Why Hierarchy works: Natural Locality

- The Principle of Locality:
  - Program access a relatively small portion of the address space at any instant of time.

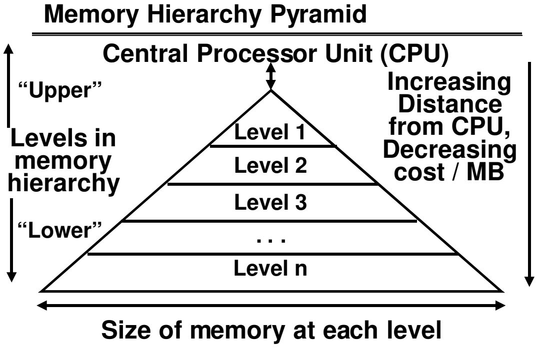


- What programming constructs lead to Principle of Locality?

Memory Hierarchy: How Does it Work?

- Temporal Locality (Locality in Time):
  - ⇒ Keep most recently accessed data items closer to the processor
  - Library Analogy: Recently read books are kept on desk
  - <u>Block</u> is unit of transfer (like book)
- Spatial Locality (Locality in Space):
  - ⇒ Move blocks consists of contiguous words to the upper levels
  - Library Analogy: Bring back nearby books on shelves when fetch a book; hope that you might need it later for your paper

Memory Hierarchy Pyramid



(data cannot be in level i unless also in i+1)

Big Idea of Memory Hierarchy

- Temporal locality: keep recently accessed data items closer to processor
- Spatial locality: moving contiguous words in memory to upper levels of hierarchy
- Uses smaller and faster memory technologies close to the processor
  - Fast hit time in highest level of hierarchy
  - Cheap, slow memory furthest from processor
- If hit rate is high enough, hierarchy has access time close to the highest (and fastest) level and size equal to the lowest (and largest) level

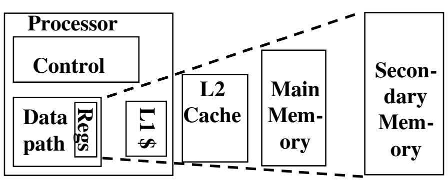
Memory Hierarchy: Terminology

- **Hit: data appears in some block in the upper level (example: Block X)**
  - Hit Rate: the fraction of memory access found in the upper level
  - Analogy: fraction of time find book on desk
- **Miss: data needs to be retrieve from a block in the lower level (Block Y)**
  - Miss Rate = 1 - (Hit Rate)
  - Analogy: fraction of time must go to shelves for book

Memory Hierarchy: Terminology

- **Hit Time: Time to access the upper level which consists of**
  - Time to determine hit/miss + Memory access time
  - Analogy: time to find, pick up book from desk
- **Miss Penalty: Time to replace a block in the upper level + Time to deliver the block the processor**
  - Analogy: time to go to shelves, find needed book, and return it to your desk, pick up
- **Note: Hit Time << Miss Penalty**

Current Memory Hierarchy



Speed(ns):	0.5ns	2ns	6ns	100ns	10,000,000ns
Size (MB):	0.0005	0.05	1-4	100-1000	100,000
Cost ($/MB):	--	$100	$30	$1	$0.05
Technology:	Regs	SRAM	SRAM	DRAM	Disk

Memory Hierarchy Technology

- **Random Access: "Random" is good: access time is the same for all locations (binary hardware tree to select table entry: billionths of a second)**
- **DRAM: Dynamic Random Access Memory**
  - High density, low power, cheap, slow
  - Dynamic: needs to be "refreshed" regularly
- **SRAM: Static Random Access Memory**
  - Low density, high power, expensive, fast
  - Static: content last "forever"(until lose power)
- **Sequential Access Technology: access time linear in location (e.g.,Tape)**

How is the hierarchy managed?

- **Registers ↔ Memory**
  - By compiler (or Asm Programmer)
- **Cache ↔ Main Memory**
  - By the hardware
- **Main Memory ↔Disks**
  - By the hardware and operating system (virtual memory; after the break)
  - By the programmer (Files)

Administrivia

- **Upcoming events**
  - Midterm Review Sunday 3/14 2PM, 1 Pimentel
  - Fill out questionnaire, answer questions
  - Conflict Midterm Mon 3/15 6PM, 405 Soda
  - Midterm on Wed. 3/17 5pm-8PM, 1 Pimentel
  - No discussion section 3/18-3/19
  - Friday before Break 3/19: video tape by Gordon Moore, "Nanometers and Gigabucks"
- **Copies of lecture slides in 271 Soda?**
  - Vote on 4 slides/page vs. 6 slides/page?
- **Copies before midterm in Copy Central?**

Administrivia Warnings

- "Is this going to be on the midterm?"
- Option 1: What you really mean is: "I don't understand this, can you explain it to me?"

∴ Our answer is: "Yes"

- Option 2: What you really mean is: "I'm behind in class, and haven't learned the assigned material yet. Do I have to try to learn this material?"

∴ Our answer is: "Yes"

- Bring to exam: Pencil, 1 sheet of paper with handwritten notes
    - No calculators, books

"Computers in the News"

- "Microsoft to Alter Software in Response to Privacy Concerns", NY Times, 3/7/99, Front Page
    - "Microsoft conceded that the feature...had the potential to be far more invasive than a traceable serial number in the ... new Pentium III that has privacy advocates up in arms. The difference is that the Windows number is tied to an individual's name, to identifying numbers on the hardware in his computer and even to documents that he creates."
    - "[M/S is] apparently building a data base that relates Ethernet adapter addresses to personal information... Ethernet adapters are cards inserted in a PC that enable it to connect to high-speed networks within organizations and through them to the Internet.

From First Lecture; How much so far?

- 15 weeks to learn big ideas in CS&E
    - Principle of abstraction, used to build systems as layers
    - Compilation v. interpretation to move down layers of system
    - Pliable Data: a program determines what it is
    - Stored program concept: instructions are data
    - Principle of Locality, exploited via a memory hierarchy (cache)
    - Greater performance by exploiting parallelism
    - Principles/pitfalls of performance measurement

Principle of abstraction, systems as layers

- Programming Languages:
    - C / Assembly / Machine Language
    - Pseudoinstructions in Assembly Language
- Translation:
    - Compiler / Assembler / Linker / Loader
- Network Protocol Suites:
    - TCP / IP / Ethernet
- Memory Hierarchy:
    - Registers / Caches / Main memory / Disk
- Others?

Compilation v. interpretation to move down

- Programming Languages:
    - C / Assembly / Machine Language
    - Compilation
- Network Protocol Suites:
    - TCP / IP / Ethernet
    - Interpretation
- Memory Hierarchy:
    - Caches / Main memory / Disk: Interpretation
    - Registers / Cache: Compilation
- Others?

Pliable Data: a program determines what it is

- Instructions (fetched from memory using PC)
- Types include Signed Integers, Unsigned Integers, Characters, Strings, Single Precision Floating Point, Double Precision Floating Point
- Everything has an address (⇒ pointers)
- TCP packet? IP packet? Ethernet packet?
- Others?

Stored program concept: instructions as data

- Allows computers to switch personalities
- Simplifies compile, assembly, link, load
- Distributing programs easy: on any disk, just like data
  - ⇒ binary compatibility, upwards compatibility (8086, 80286, 80386, 80486, Penitum I, II, III)
- Allows for efficient Dynamic Libraries: modify the code to patch in real address
- Makes it easier for viruses: Send message that overflows stack, starts executing code in stack area, take over machine

Principle of Locality

- Exploited by memory hierarchy
- Registers assume Temporal Locality: data in registers will be reused
- Disk seeks faster in practice: short seeks are much faster, so disk accesses take less time ⇒ due to Spatial Locality
- Disks transfer in 512 Byte blocks assuming spatial locality: more than just 4 bytes useful to program
- Networks: most traffic is local, so local area network vs. wide area network

Greater performance by exploiting parallelism

- RAID (Redundant Array of Inexp. Disks)
  - Replace a few number of large disks with a large number of small disks ⇒ more arms moving, more heads transferring (even though small disks maybe slower)
- Switched Networks
  - More performance in system since multiple messages can transfer at same time, (even though network latency is no better between 2 computers on unloaded system)
- Others?

Performance measurement Principles/Pitfalls

- Network performance measure: only looking peak bandwidth, not including software start-up overhead for message
- Disk seek time:
  - Its much better than what manufacturer quotes (3X to 4X)
- Disk transfer rate (internal media rate):
  - Its worse than what manufacturer quotes (0.75X)
  - See if profs in OS class know these!
- Others?

Rapid Change AND Little Change

- Continued Rapid Improvement in Computing
  - 2X every 1.5 years (10X/5yrs, 100X/10yrs)
  - Processor speed, Memory size - Moore's Law as enabler (2X transistors/chip/1.5 yrs); Disk capacity too (not Moore's Law)
- 5 classic components of all computers

  $\left.\begin{array}{l}\text{1. Control}\\ \text{2. Datapath}\end{array}\right\}$ Processor (or CPU)

  3. Memory
  4. Input
  5. Output

"And in Conclusion ..."

- Principle of Locality + Hierarchy of Memories of different speed, cost; exploit locality to improve cost-performance
- Hierarchy Terms: Hit, Miss, Hit Time, Miss Penalty, Hit Rate, Miss Rate, Block, Upper level memory, Lower level memory
- Review of Big Ideas (so far):
  - Abstraction, Stored Program, Pliable Data, compilation vs. interpretation, Performance via Parallelism, Performance Pitfalls
  - Applies to Processor, Memory, and I/O
- Next Midterm, then Gordon Moore, Break