CS61C : Machine Structures

Lecture #25
Input / Output, Networks II, Disks

CPS today!

2005-11-30

There is one handout today at the front and back of the room!

Lecturer PSOE, new dad Dan Garcia

www.cs.berkeley.edu/~ddgarcia

Maxell's 300GB HVDs! ⇒

We all fondly remember the days of Zip and Syquest drives. InPhase Technologies has developed 300GB Holographic Versatile discs, w/1.6TB discs to come later!

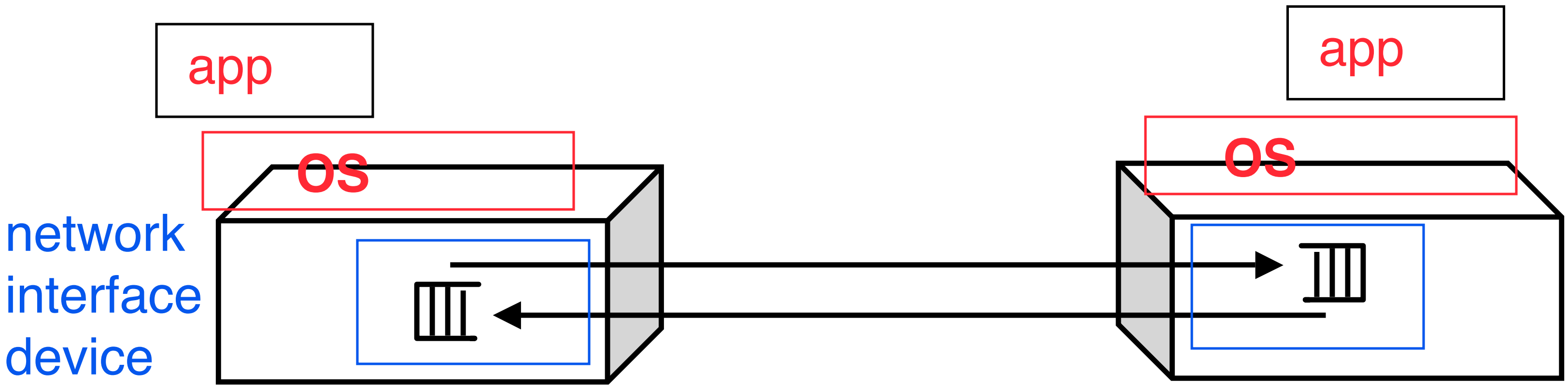
www.theregister.com/2005/11/24/maxell_holo_storage/

Review

- I/O gives computers their 5 senses
- I/O speed range is 12.5-million to one
- Processor speed means must synchronize with I/O devices before use
- Polling works, but expensive
  - processor repeatedly queries devices
- Interrupts works, more complex
  - devices cause exception, OS runs and deal with the device
- I/O control leads to Operating Systems
- Integrated circuit ("Moore's Law") revolutionizing network switches as well as processors
  - Switch just a specialized computer
- Trend from shared to switched networks to get faster links and scalable bandwidth

ABCs of Networks: 2 Computers

- **Starting Point: Send bits between 2 computers**



- **Queue (First In First Out) on each end**
- **Can send both ways ("Full Duplex")**
  - **One-way information is called "Half Duplex"**
- **Information sent called a "message"**
  - **Note: Messages also called packets**

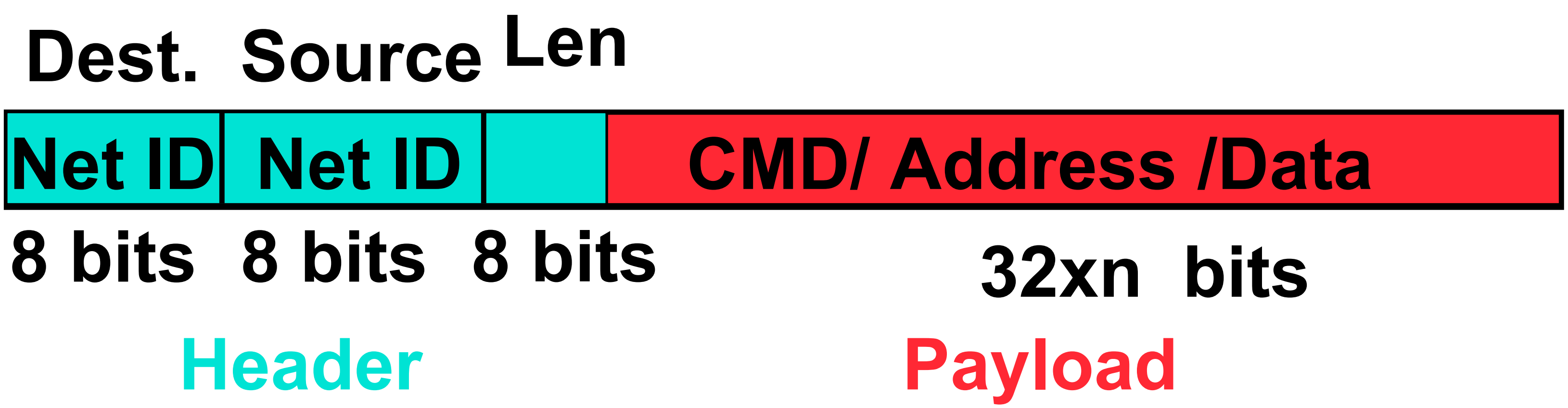
A Simple Example: 2 Computers

- **What is Message Format?**
  - **Similar idea to Instruction Format**
  - **Fixed size? Number bits?**

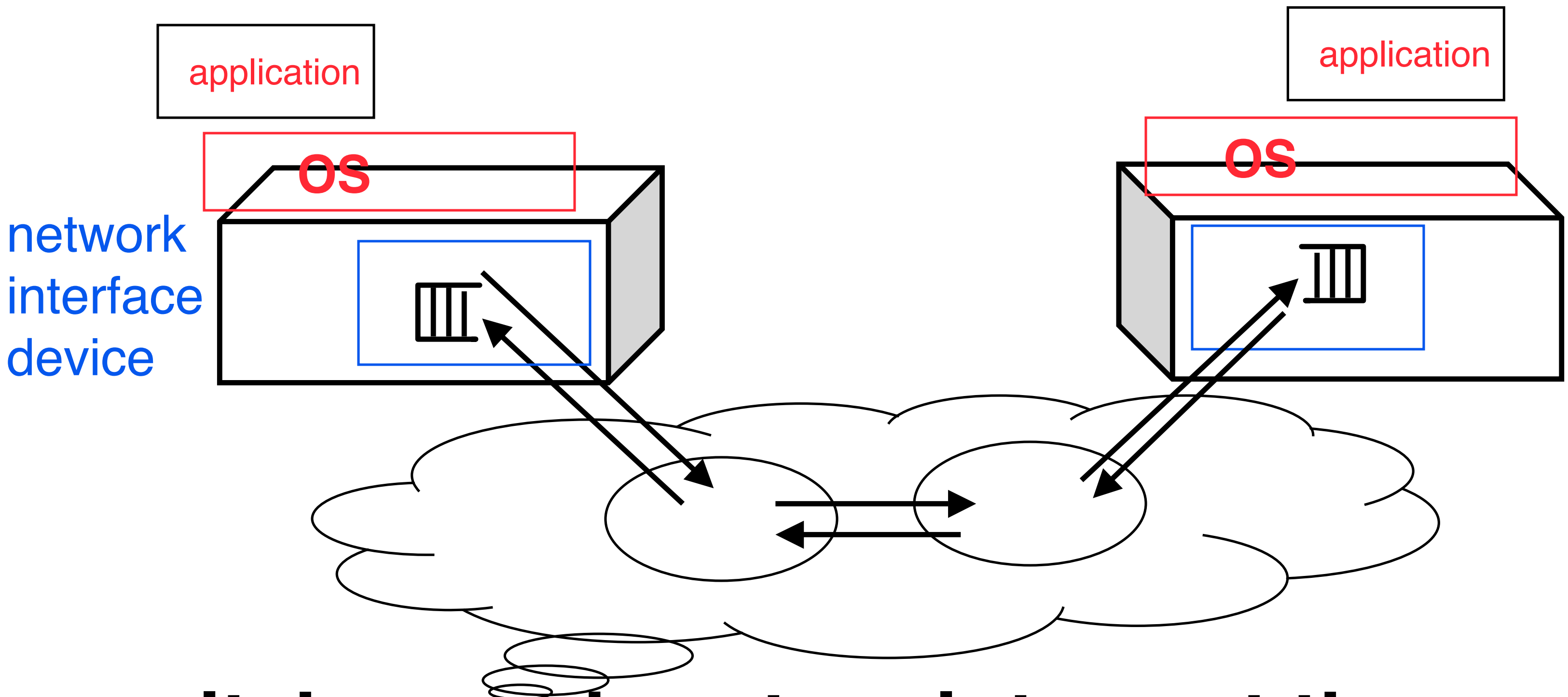
- **<u>Header(Trailer):</u> information to deliver message**
- **<u>Payload</u>: data in message**
- **What can be in the data?**
  - **anything that you can represent as bits**
  - **values, chars, commands, addresses...**

Questions About Simple Example

- **What if more than 2 computers want to communicate?**
  - **Need computer "address field" in packet to know which computer should receive it (destination), and to which computer it came from for reply (source) [just like envelopes!]**



ABCs: many computers



- **switches and routers interpret the header in order to deliver the packet**
- **source encodes and destination decodes content of the payload**

Questions About Simple Example

- What if message is garbled in transit?
- Add redundant information that is checked when message arrives to be sure it is OK
- 8-bit sum of other bytes: called "Check sum"; upon arrival compare check sum to sum of rest of information in message. `xor` also popular.

Math 55 talks about what a Check sum is…

Questions About Simple Example

- What if message never arrives?
- Receiver tells sender when it arrives (ack) [ala registered mail], sender retries if waits too long
- Don’t discard message until get “ACK” (for ACKnowledgment);
  Also, if check sum fails, don’t send ACK

Observations About Simple Example

- Simple questions such as those above lead to more complex procedures to send/receive message and more complex message formats
- Protocol: algorithm for properly sending and receiving messages (packets)

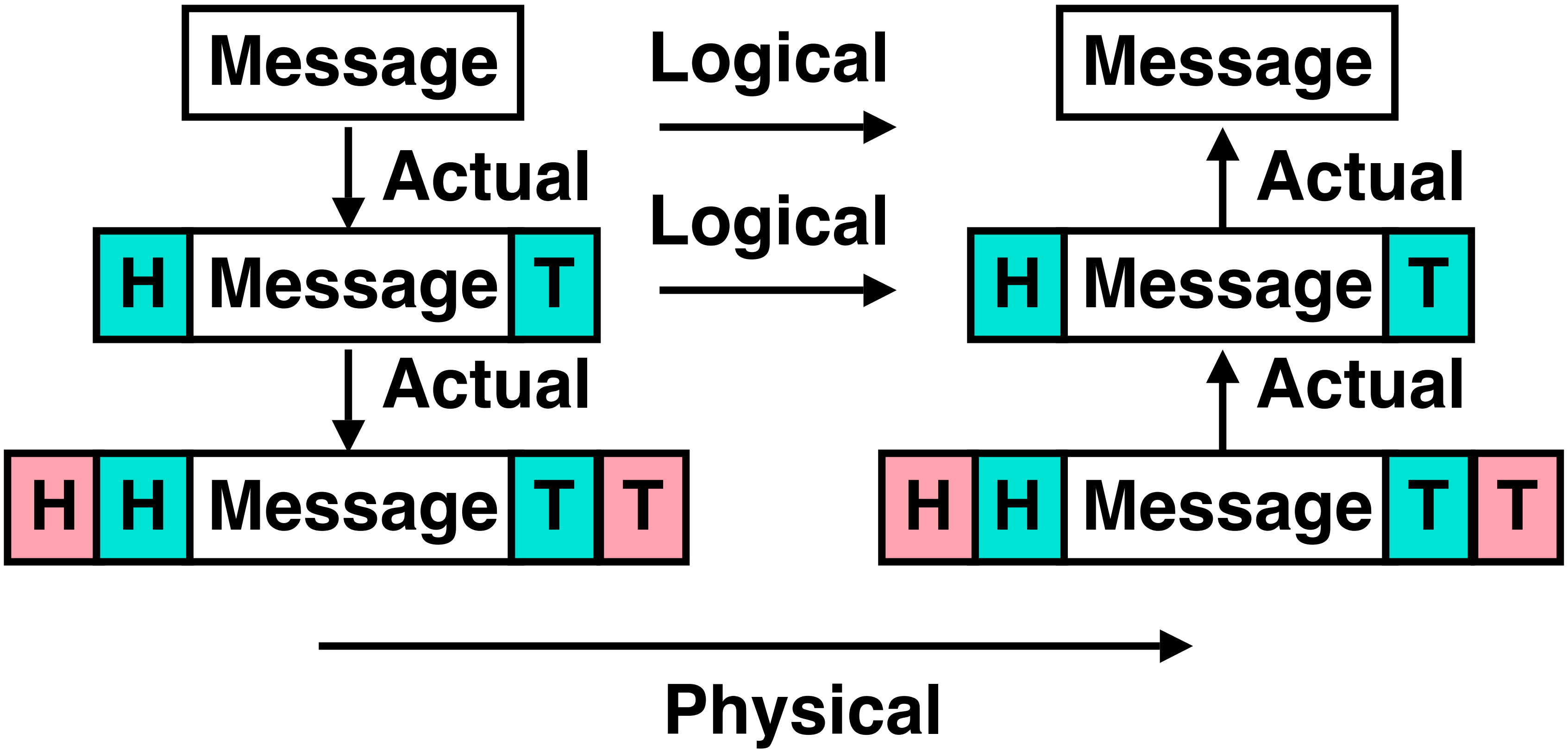
Software Protocol to Send and Receive

- SW Send steps
  - 1: Application copies data to OS buffer
  - 2: OS calculates checksum, starts timer
  - 3: OS sends data to network interface HW and says start
- SW Receive steps
  - 3: OS copies data from network interface HW to OS buffer
  - 2: OS calculates checksum, if OK, send ACK; if not, delete message (sender resends when timer expires)
  - 1: If OK, OS copies data to user address space, & signals application to continue

Protocol for Networks of Networks?

- Internetworking: allows computers on independent and incompatible networks to communicate reliably and efficiently;
  - Enabling technologies: SW standards that allow reliable communications without reliable networks
  - Hierarchy of SW layers, giving each layer responsibility for portion of overall communications task, called protocol families or protocol suites
- Abstraction to cope with complexity of communication vs. Abstraction for complexity of computation

Protocol Family Concept



Protocol Family Concept

- Key to protocol families is that communication occurs logically at the same level of the protocol, called peer-to-peer…

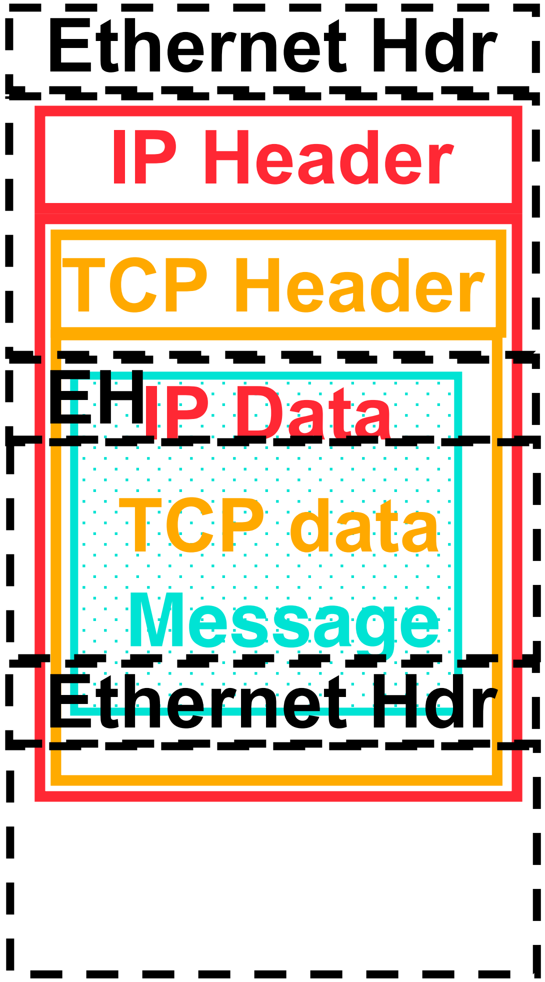
  …but is implemented via services at the next lower level
- Encapsulation: carry higher level information within lower level "envelope"
- Fragmentation: break packet into multiple smaller packets and reassemble

Protocol for Network of Networks

- Transmission Control Protocol/Internet Protocol (TCP/IP)
  - This protocol family is the basis of the Internet, a WAN protocol
  - IP makes best effort to deliver
  - TCP guarantees delivery
  - TCP/IP so popular it is used even when communicating locally: even across homogeneous LAN

TCP/IP packet, Ethernet packet, protocols

- Application sends message
- TCP breaks into 64KiB segments, adds 20B header
- IP adds 20B header, sends to network
- If Ethernet, broken into 1500B packets with headers, trailers (24B)
- All Headers, trailers have length field, destination, ...



Overhead vs. Bandwidth

- Networks are typically advertised using peak bandwidth of network link: e.g., 100 Mbits/sec Ethernet ("100 base T")
- Software overhead to put message into network or get message out of network often limits useful bandwidth
- Assume overhead to send and receive = 320 microseconds (μs), want to send 1000 Bytes over "100 Mbit/s" Ethernet
  - Network transmission time: 1000Bx8b/B /100Mb/s = 8000b / (100b/μs) = 80 μs
  - Effective bandwidth: 8000b/(320+80)μs = 20 Mb/s

Peer Instruction

(T / F) P2P filesharing has been the dominant application on many links!

Suppose we have 2 networks, Which has a higher effective bandwidth as a function of the transferred data size?

- BearsNet
  TCP/IP overhead 300 μs, peak BW 10Mb/s
- CalNet
  TCP/IP overhead 500 μs, peak BW 100Mb/s

```
   TRUE
1: B always
2: C always
3: B small
   C big
4: B big
   C small
5: The same!

   FALSE
6: B always
7: C always
8: B small
   C big
9: B big
   C small
0: The same!
```

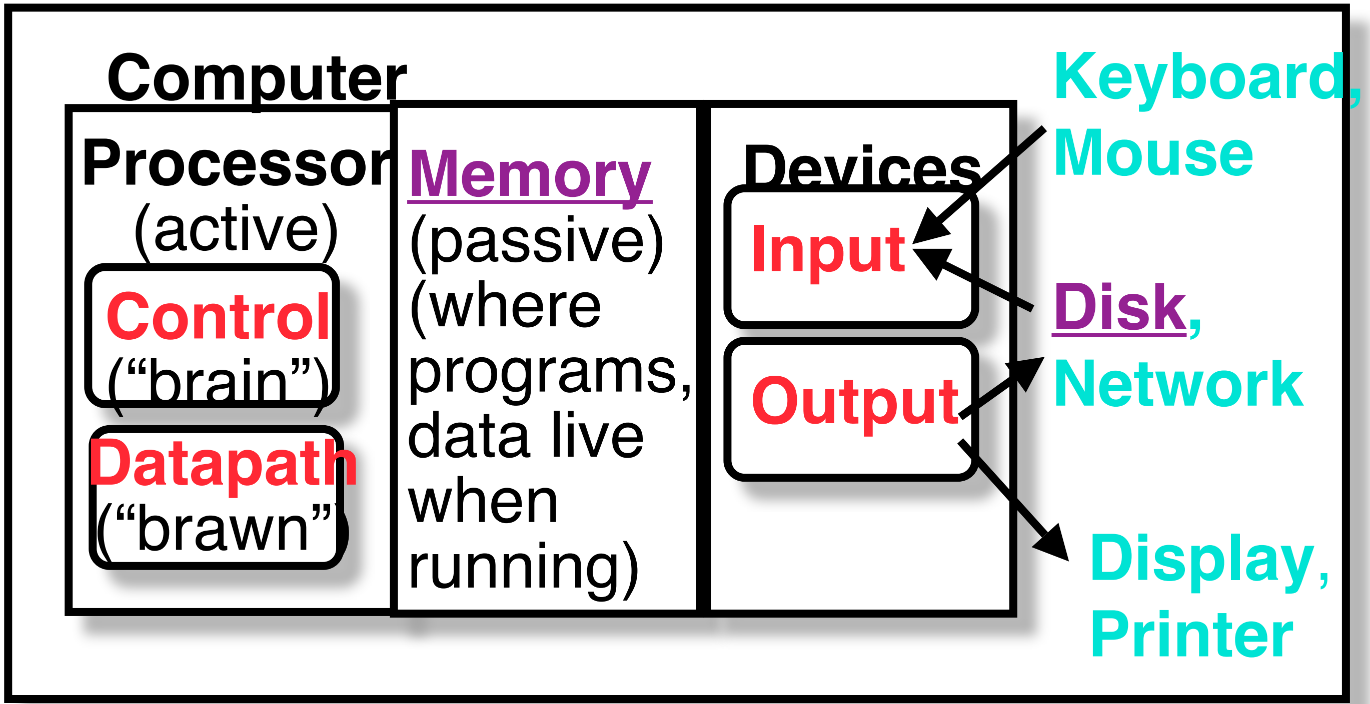
Administrivia

- Only 2 lectures to go (after this one)! :-(
- Project 4 (Cache simulator) due friday
- Compete in the Performance contest!
  - Deadline is Mon, 2005-12-12 @ 11:59pm, ~12 days from now
- HW4 and HW5 are done
  - Regrade requests are due by 2005-12-05
- Project 3 will be graded face-to-face, check web page for scheduling
- Final: 2005-12-17 @ 12:30pm in 2050 VLSB!

Upcoming Calendar

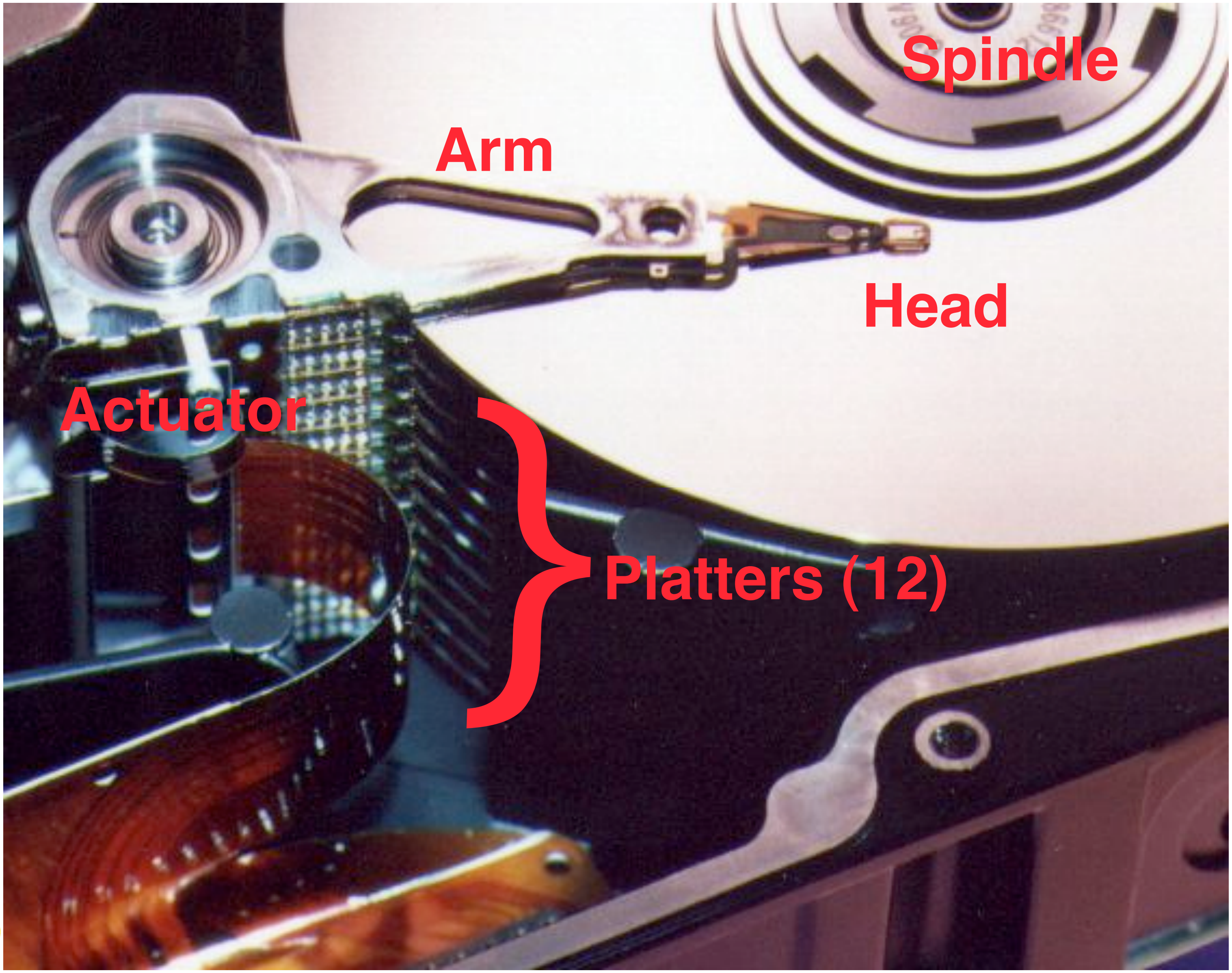
Week #	Mon	Wed	Thu Lab	Sat
#14 This week	I/O Basics & Networks I	I/O Networks II & Disks	I/O Polling	Cache project due yesterday
#15 Last Week o' Classes	Performance	LAST CLASS Summary, Review, & HKN Evals	I/O Networking & 61C Feedback Survey	
#16 Sun 2pm Review 10 Evans	Performance competition due tonight @ midnight			FINAL EXAM SAT 12-17 @ 12:30pm-3:30pm 2050 VLSB Performance awards

Magnetic Disks

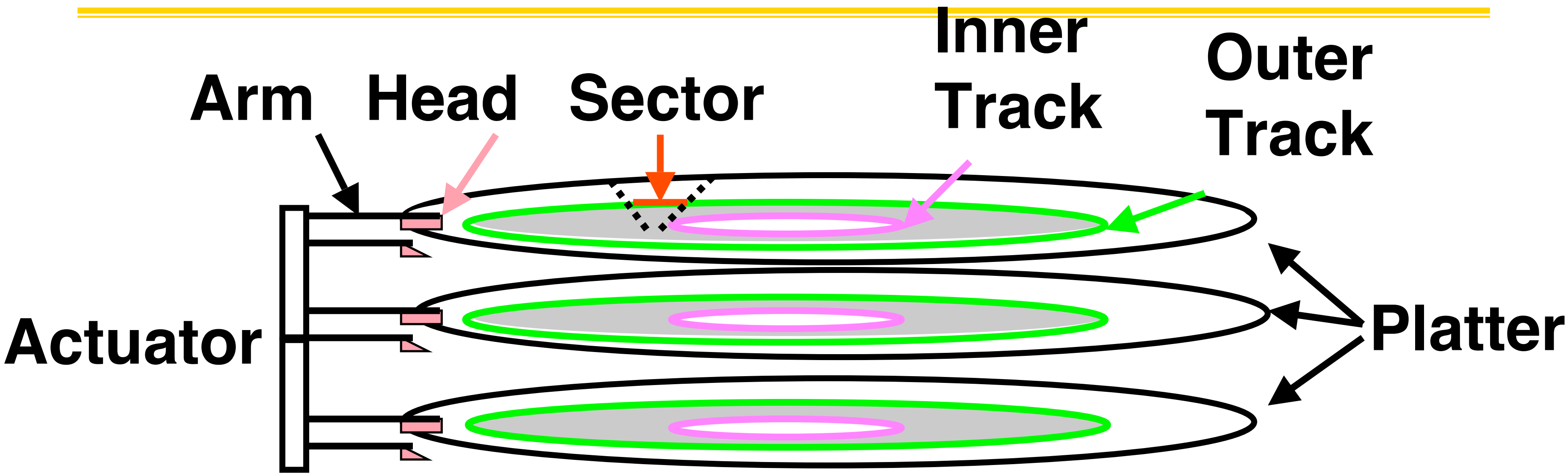


- **Purpose:**
  - **Long-term, nonvolatile, inexpensive storage for files**
  - **Large, inexpensive, slow level in the memory hierarchy**

Photo of Disk Head, Arm, Actuator

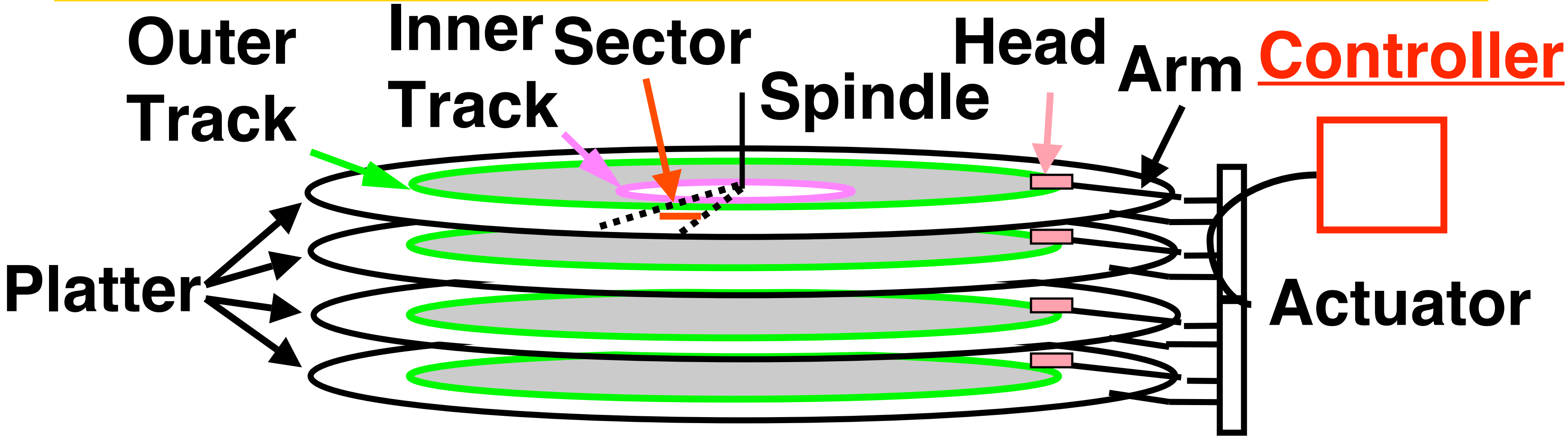


Disk Device Terminology



- Several <u>platters</u>, with information recorded magnetically on both <u>surfaces</u> (usually)
- Bits recorded in <u>tracks</u>, which in turn divided into <u>sectors</u> (e.g., 512 Bytes)
- <u>Actuator</u> moves <u>head</u> (end of <u>arm</u>) over track (<u>"seek"</u>), wait for <u>sector</u> rotate under <u>head</u>, then read or write

Disk Device Performance



- **Disk Latency = Seek Time + Rotation Time + Transfer Time + Controller Overhead**
  - Seek Time? depends no. tracks move arm, seek speed of disk
  - Rotation Time? depends on speed disk rotates, how far sector is from head
  - Transfer Time? depends on data rate (bandwidth) of disk (bit density), size of request

Data Rate: Inner vs. Outer Tracks

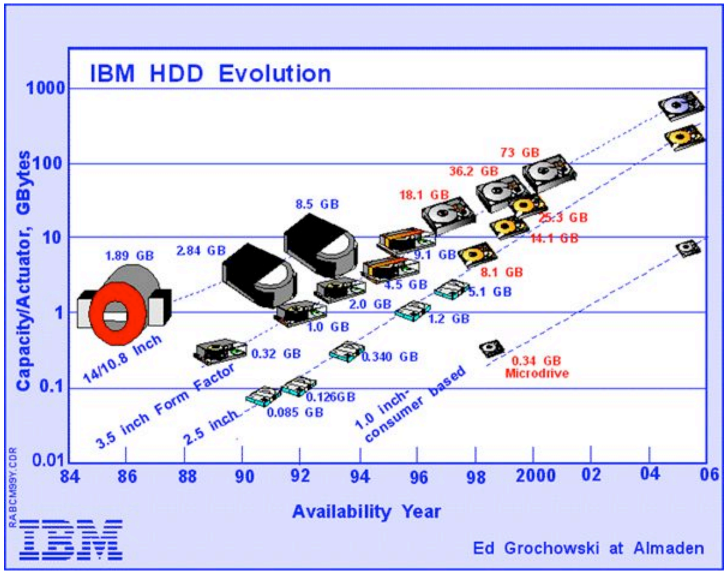
- To keep things simple, originally same # of sectors/track
  - Since outer track longer, lower bits per inch
- Competition decided to keep bits/inch (BPI) high for all tracks ("constant bit density")
  - More capacity per disk
  - More sectors per track towards edge
  - Since disk spins at constant speed, outer tracks have faster data rate
- Bandwidth outer track 1.7X inner track!

Disk Performance Model /Trends

- Capacity : + 100% / year (2X / 1.0 yrs)

  Over time, grown so fast that # of platters has reduced (some even use only 1 now!)

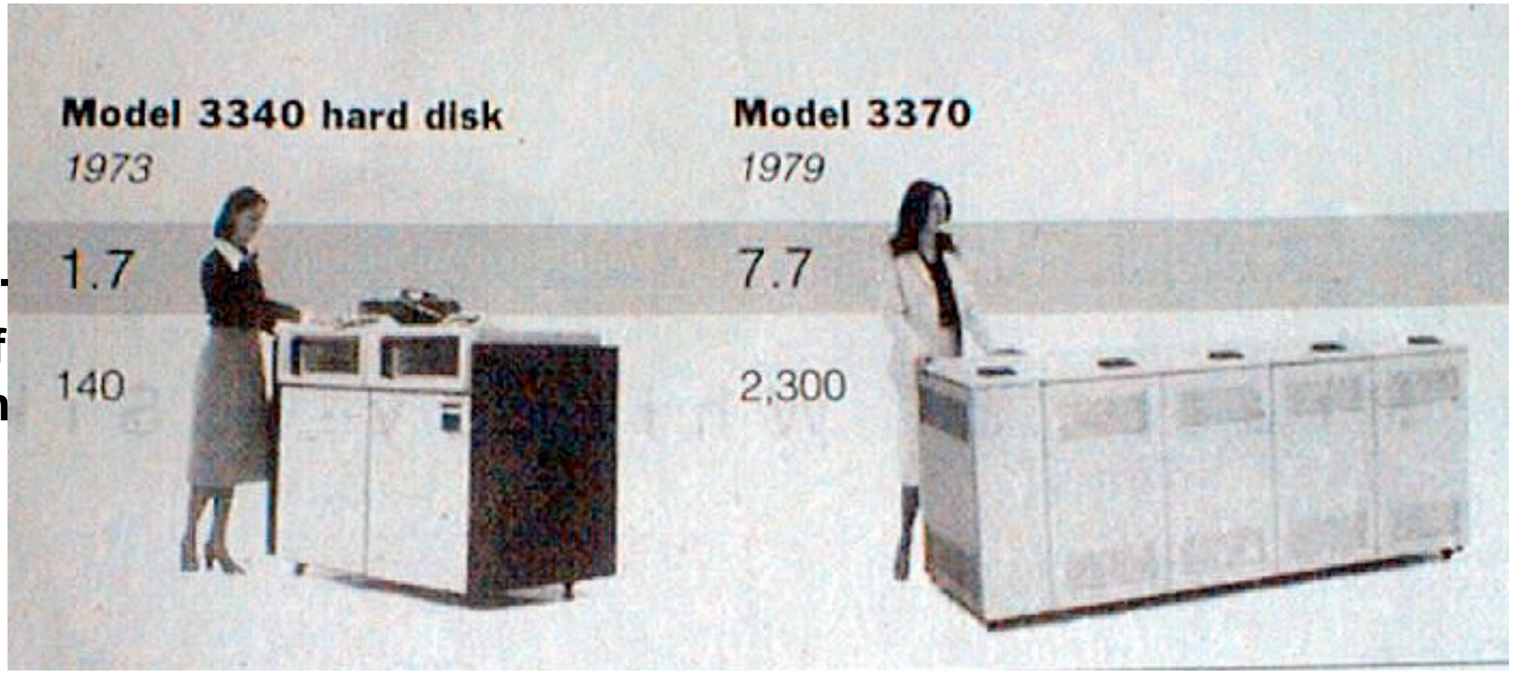
- Transfer rate (BW) : + 40%/yr (2X / 2 yrs)
- Rotation+Seek time : – 8%/yr (1/2 in 10 yrs)
- Areal Density
  - Bits recorded along a track: Bits/Inch (BPI)
  - # of tracks per surface: Tracks/Inch (TPI)
  - We care about bit density per unit area Bits/Inch^2
  - Called Areal Density = BPI x TPI
- MB/$: > 100%/year (2X / 1.0 yrs)
  - Fewer chips + areal density



Disk History (IBM)

Data density Mibit/sq. in.

Capacity of Unit Shown Mibytes

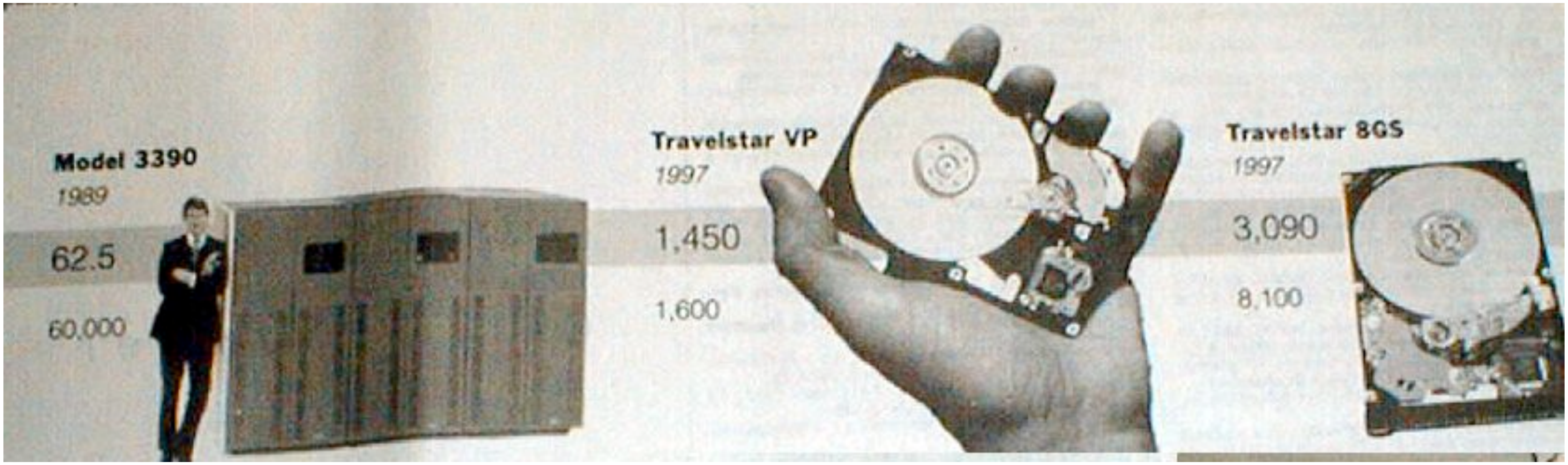


1973:
1. 7 Mibit/sq. in
0.14 GiBytes

1979:
7. 7 Mibit/sq. in
2.3 GiBytes

source: New York Times, 2/23/98, page C3,
"Makers of disk drives crowd even more data into even smaller spaces"

Disk History



1989:	1997:	1997:
63 Mibit/sq. in	1450 Mibit/sq. in	3090 Mibit/sq. in
60 GiBytes	2.3 GiBytes	8.1 GiBytes

source: New York Times, 2/23/98, page C3,
"Makers of disk drives crowd even more data into even smaller spaces"

Historical Perspective

- Form factor and capacity drives market, more than performance
- 1970s: Mainframes ⇒ 14" diam. disks
- 1980s: Minicomputers, Servers ⇒ 8", 5.25" diam. disks
- Late 1980s/Early 1990s:
  - Pizzabox PCs ⇒ 3.5 inch diameter disks
  - Laptops, notebooks ⇒ 2.5 inch disks
  - Palmtops didn't use disks, so 1.8 inch diameter disks didn't make it

The five most popular internal form factors for PC hard disks. Clockwise from the left: 5.25", 3.5", 2.5", PC Card and CompactFlash.

`www.pcguide.com/ref/hdd/op/form.htm`

State of the Art: Two camps (2005)

- Performance
  - Enterprise apps, servers
- E.g., Seagate Cheetah 15K.4
  - Serial-Attached SCSI, Ultra320 SCSI, 2Gbit Fibre Channel interface
  - 146 GB, 3.5-inch disk
  - 15,000 RPM
  - 4 discs, 8 heads
  - 13 watts (idle)
  - 3.5 ms avg. seek
  - 200 MB/s transfer rate
  - 1.4 Million hrs MTBF
  - 5 year warrantee
  - $1000 = $6.8 / GB

- Capacity
  - Mainstream, home uses
- E.g., Seagate Barracuda 7200.9
  - Serial ATA 3Gb/s, Ultra ATA/100
  - 500 GB, 3.5-inch disk
  - 7,200 RPM
  - ? discs, ? heads
  - 7 watts (idle)
  - 8.5 ms avg. seek
  - 300 MB/s transfer rate
  - ? Million hrs MTBF
  - 5 year warrantee
  - $330 = $0.66 / GB

source: www.seagate.com

1 inch disk drive!

- 2005 Hitachi Microdrive:
  - 40 x 30 x 5 mm, 13g
  - 8 GB, 3600 RPM, 1 disk, 10 MB/s, 12 ms seek
  - 400G operational shock, 2000G non-operational
  - Can detect a fall in 4” and retract heads to safety
  - For iPods, cameras, phones
- 2006 MicroDrive?
  - 16 GB, 12 MB/s!
  - Assuming past trends continue

www.hitachigst.com

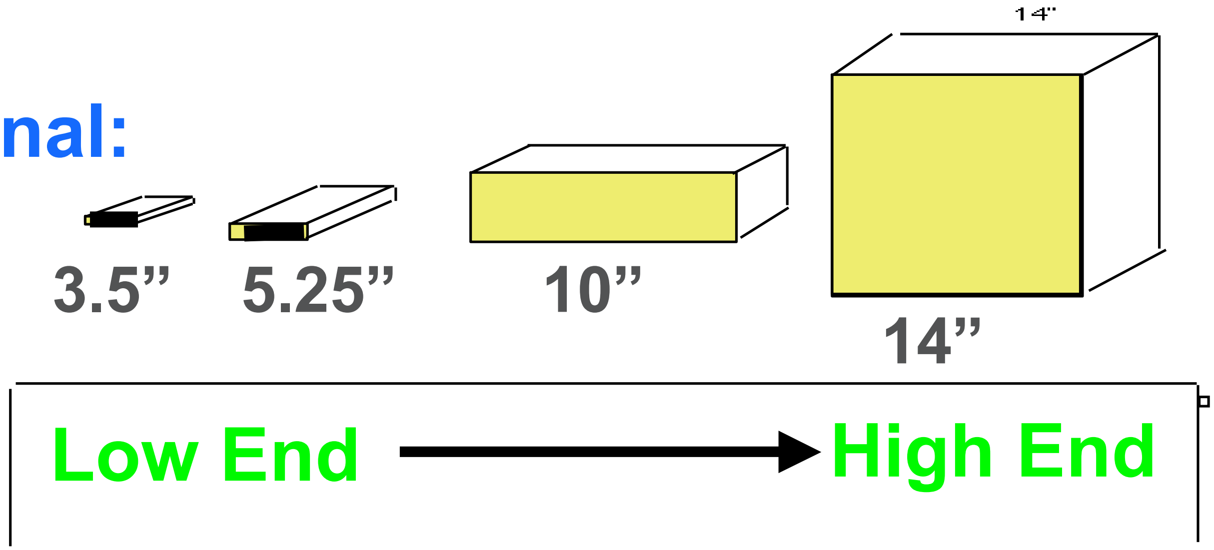
Where does Flash memory come in?

- Microdrives and Flash memory (e.g., CompactFlash) are going head-to-head
  - Both non-volatile (no power, data ok)
  - Flash benefits: durable & lower power (no moving parts)
  - Flash limitations: finite number of write cycles (wear on the insulating oxide layer around the charge storage mechanism)
    - OEMs work around by spreading writes out
- How does Flash memory work?
  - NMOS transistor with an additional conductor between gate and source/drain which “traps” electrons. The presence/absence is a 1 or 0.
  - `wikipedia.org/wiki/Flash_memory`

Use Arrays of Small Disks…

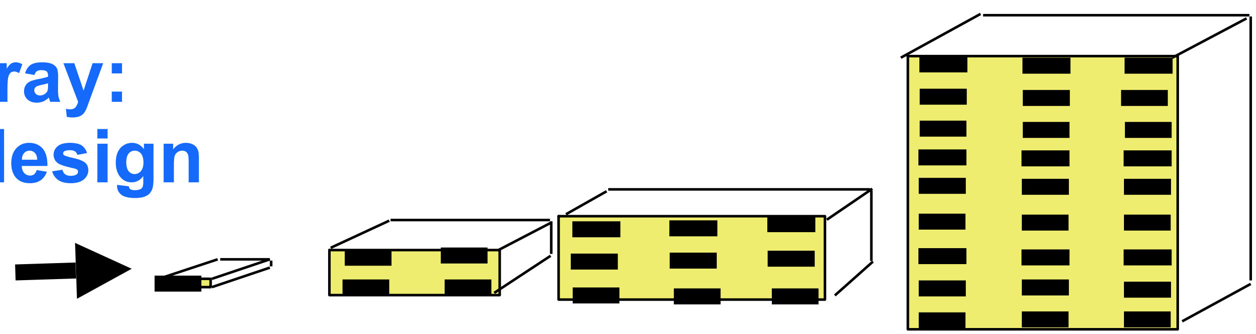
- Katz and Patterson asked in 1987:
  - Can smaller disks be used to close gap in performance between disks and CPUs?

Conventional:
4 disk
designs



Disk Array:
1 disk design

3.5”



Replace Small Number of Large Disks with Large Number of Small Disks! (1988 Disks)

	IBM 3390K	IBM 3.5" 0061	x70	
Capacity	20 GBytes	320 MBytes	23 GBytes	
Volume	97 cu. ft.	0.1 cu. ft.	11 cu. ft.	9X
Power	3 KW	11 W	1 KW	3X
Data Rate	15 MB/s	1.5 MB/s	120 MB/s	8X
I/O Rate	600 I/Os/s	55 I/Os/s	3900 IOs/s	6X
MTTF	250 KHrs	50 KHrs	??? Hrs	
Cost	$250K	$2K	$150K	

Disk Arrays potentially high performance, high MB per cu. ft., high MB per KW,

but what about reliability?

Array Reliability

- **Reliability** - whether or not a component has failed
  - measured as Mean Time To Failure (MTTF)
- Reliability of N disks = Reliability of 1 Disk ÷ N (assuming failures independent)
  - 50,000 Hours ÷ 70 disks = 700 hour
- Disk system MTTF: Drops from 6 years to 1 month!
- Disk arrays too unreliable to be useful!

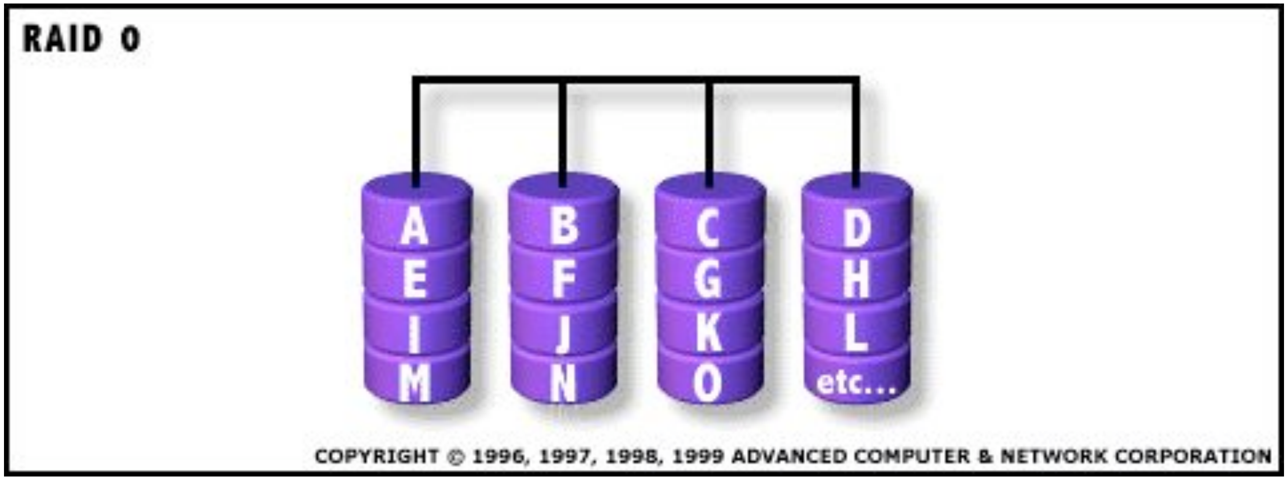
Redundant Arrays of (Inexpensive) Disks

- Files are "striped" across multiple disks
- Redundancy yields high data availability
  - **Availability**: service still provided to user, even if some components failed
- Disks will still fail
- Contents reconstructed from data redundantly stored in the array
  - ⇒ Capacity penalty to store redundant info
  - ⇒ Bandwidth penalty to update redundant info

Berkeley History, RAID-I

- RAID-I (1989)
  - Consisted of a Sun 4/280 workstation with 128 MB of DRAM, four dual-string SCSI controllers, 28 5.25-inch SCSI disks and specialized disk striping software
- Today RAID is > $27 billion dollar industry, 80% nonPC disks sold in RAIDs

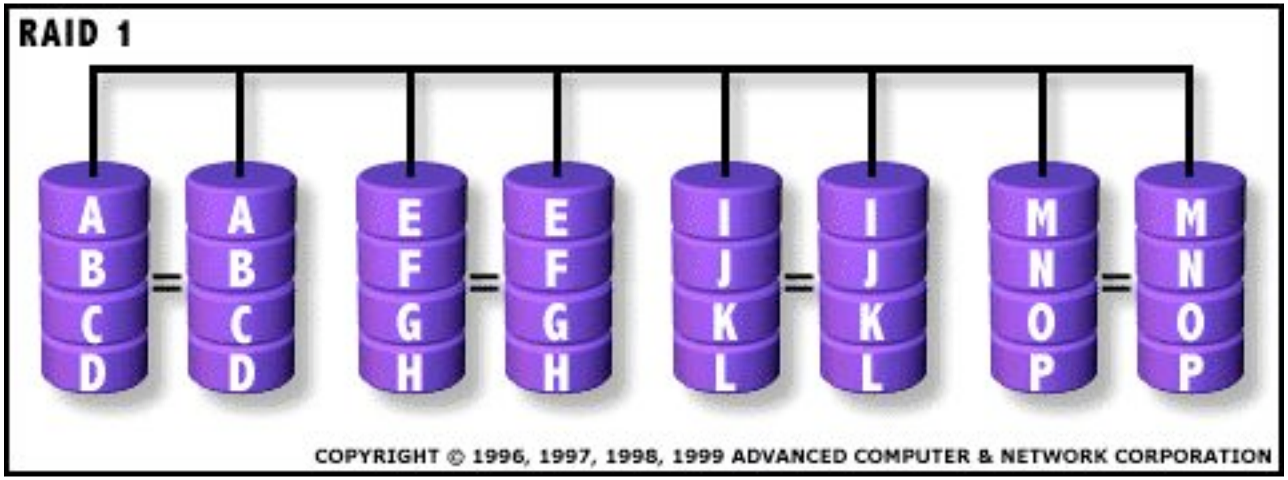
“RAID 0”: No redundancy = “AID”



- **Assume have 4 disks of data for this example, organized in blocks**
- **Large accesses faster since transfer from several disks at once**

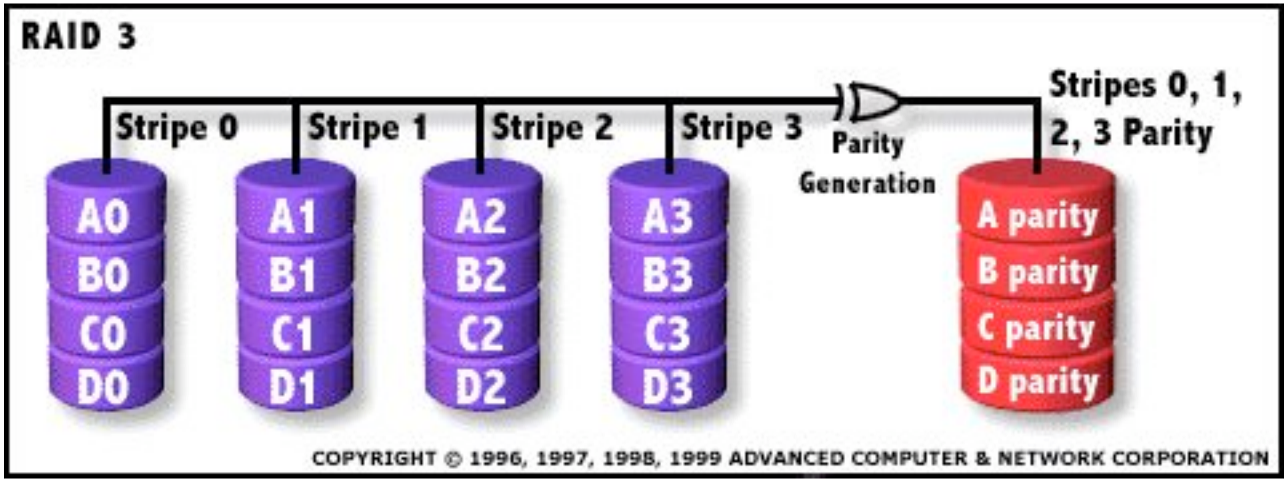
This and next 5 slides from RAID.edu, http://www.acnc.com/04_01_00.html

RAID 1: Mirror data



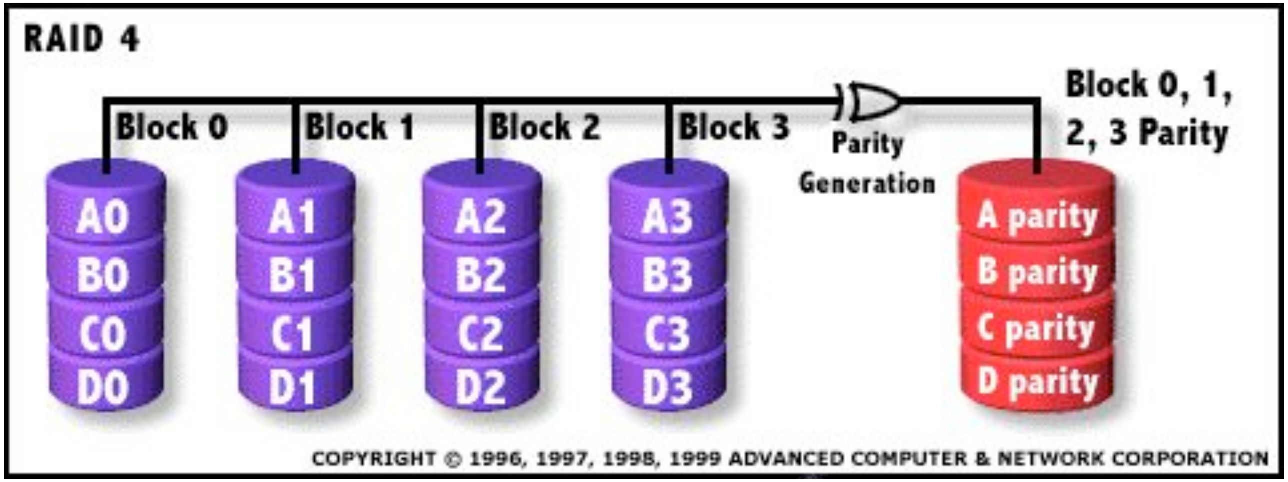
- Each disk is fully duplicated onto its "mirror"
  - Very high availability can be achieved
- Bandwidth reduced on write:
  - 1 Logical write = 2 physical writes
- Most expensive solution: 100% capacity overhead

RAID 3: Parity (RAID 2 has bit-level striping)



- Parity computed across group to protect against hard disk failures, stored in P disk
- Logically, a single high capacity, high transfer rate disk
- 25% capacity cost for parity in this example vs. 100% for RAID 1 (5 disks vs. 8 disks)

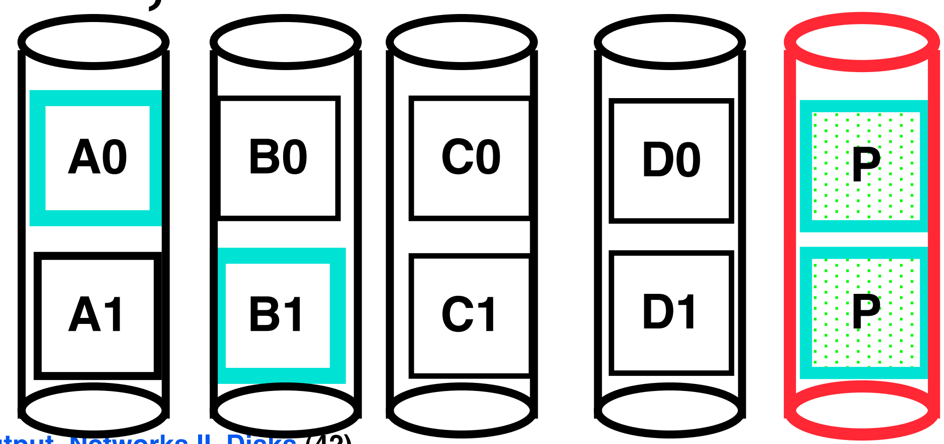
RAID 4: parity plus small sized accesses



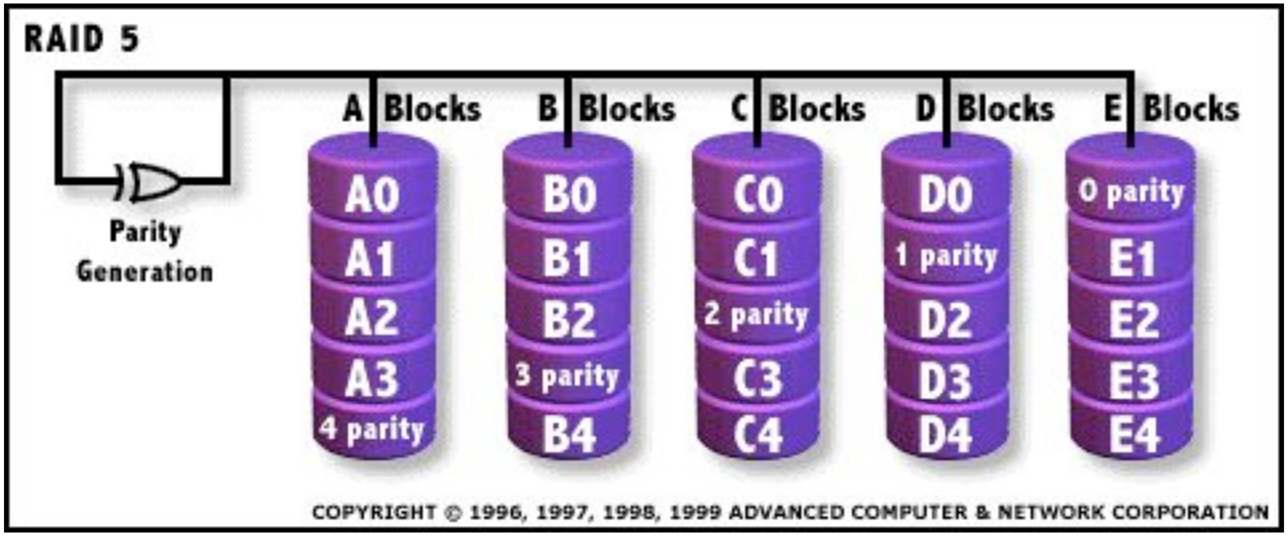
- RAID 3 relies on parity disk to discover errors on Read
- But every sector has an error detection field
- Rely on error detection field to catch errors on read, not on the parity disk
- Allows small independent reads to different disks simultaneously

Inspiration for RAID 5

- **Small writes (write to one disk):**
  - Option 1: read other data disks, create new sum and write to Parity Disk (access all disks)
  - Option 2: since P has old sum, compare old data to new data, add the difference to P: 1 logical write = 2 physical reads + 2 physical writes to 2 disks
- **Parity Disk is bottleneck for Small writes: Write to A0, B1 => both write to P disk**



RAID 5: Rotated Parity, faster small writes



- Independent writes possible because of interleaved parity
  - Example: write to A0, B1 uses disks 0, 1, 4, 5, so can proceed in parallel
  - Still 1 small write = 4 physical disk accesses

Peer Instruction

1. RAID 1 (mirror) and 5 (rotated parity) help with performance <u>and</u> availability
2. RAID 1 has higher cost than RAID 5
3. Small writes on RAID 5 are slower than on RAID 1

	ABC
1:	FFF
2:	FFT
3:	FTF
4:	FTT
5:	TFF
6:	TFT
7:	TTF
8:	TTT

"And In conclusion…"

- Protocol suites allow heterogeneous networking
  - Another form of principle of abstraction
  - Protocols ⇒ operation in presence of failures
  - Standardization key for LAN, WAN
- 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
- RAID
  - Higher performance with more disk arms per $
  - Adds option for small # of extra disks
  - Today RAID is > $27 billion dollar industry, 80% nonPC disks sold in RAIDs; started at Cal