Lecture 11:
Cache Conclusion and I/O Introduction: Storage Devices, Metrics, & Productivity

Professor David A. Patterson

Computer Science 252

Fall 1996

Review: IRAM Challenges

- Chip
  - Speed, area, power, yield of logic in DRAM process?
  - Speed, area, power, yield of SRAM in DRAM process?
  - Good performance and reasonable power?
  - BW/Latency oriented DRAM tradeoffs?
- Architecture
  - How to turn high memory bandwidth into performance?
    - » Vector?
    - » Extensive Prefetching?
  - Extensible IRAM: Large pgm/data solution?
  - Redudancy in processor to match redundancy in DRAM?

Review: Doing Research in the Information Age

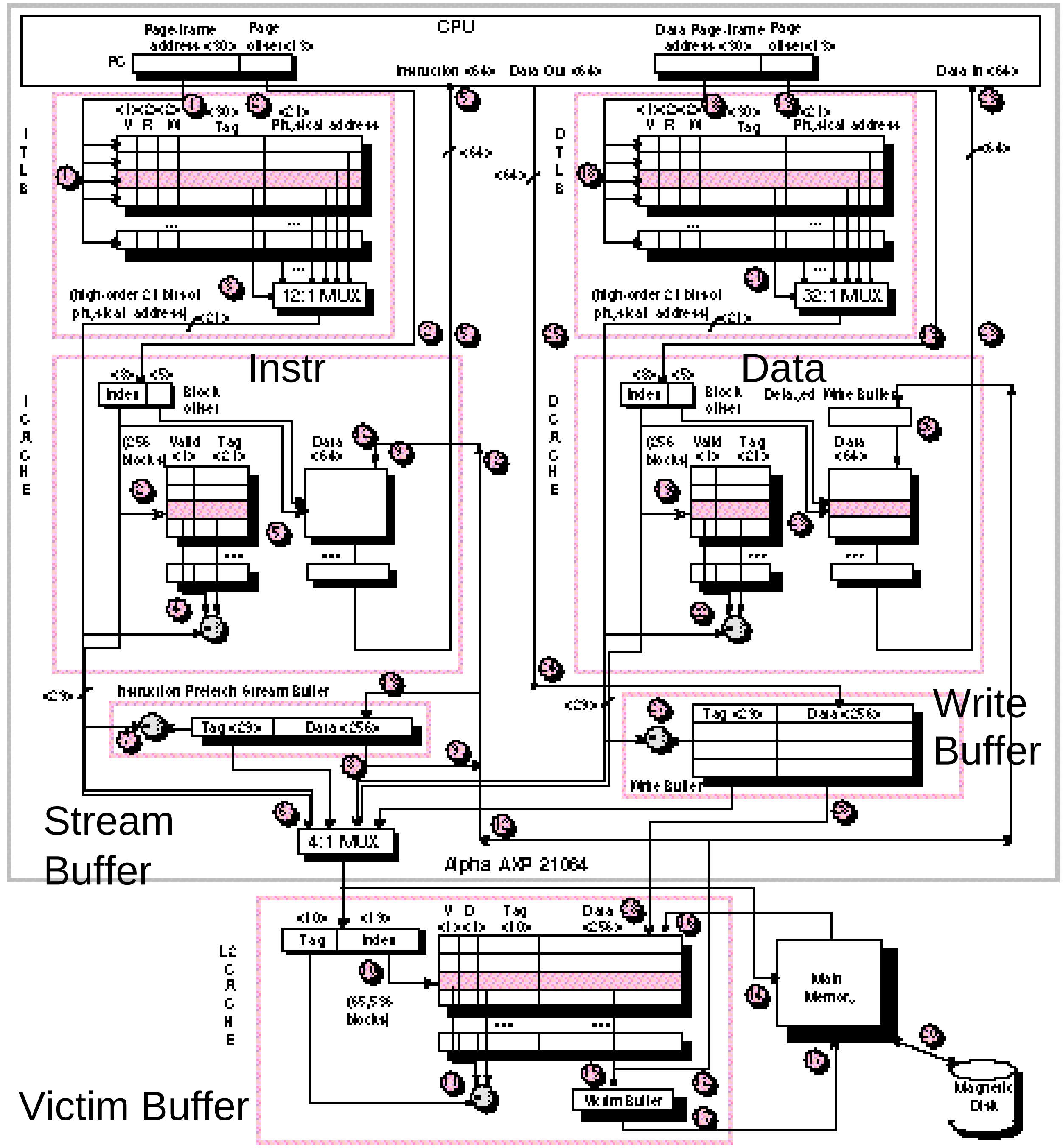
- **Online at UCB**
  - **Finding articles**
    - » **INSPECT database**
    - » **COMP database**
  - **Printing IEEE articles**
  - **Finding Books: MELVYL and GLADIS**
- **WWW Search Engines**
  - **Alta Vista, HotBot, Yahoo!**
- **Computer Architecture Resources**
  - **Architecture Homepage, Benchmark Database...**

Cache Cross Cutting Issues

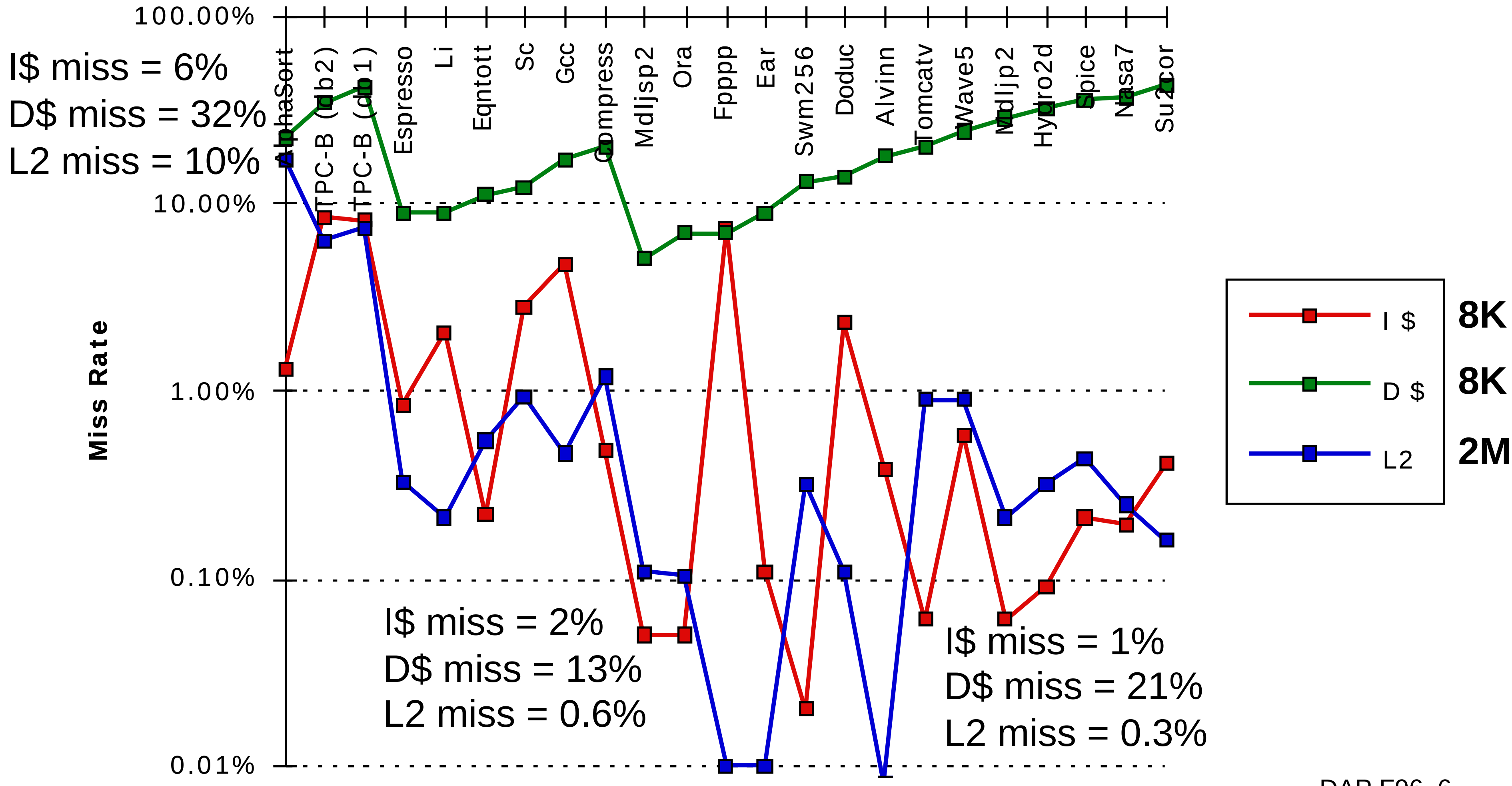
- **Superscalar CPU & Number Cache Ports**
- **Speculative Execution and non-faulting option on memory**
- **Parallel Execution vs. Cache locality**
  - **Want far separation to find independent operations vs. want reuse of data accesses to avoid misses**
- **I/O and consistency of data between cache and memory**
  - **Caches => multiple copies of data**
  - **Consistency by HW or by SW?**
  - **Where connect I/O to computer?**

Alpha 21064

- Separate Instr & Data TLB & Caches
- TLBs fully associative
- TLB updates in SW ("Priv Arch Libr")
- Caches 8KB direct mapped, write thru
- Critical 8 bytes first
- Prefetch instr. stream buffer
- 2 MB L2 cache, direct mapped, WB (off-chip)
- 256 bit path to main memory, 4 x 64-bit modules
- Victim Buffer: to give read priority over write
- 4 entry write buffer between D$ & L2$

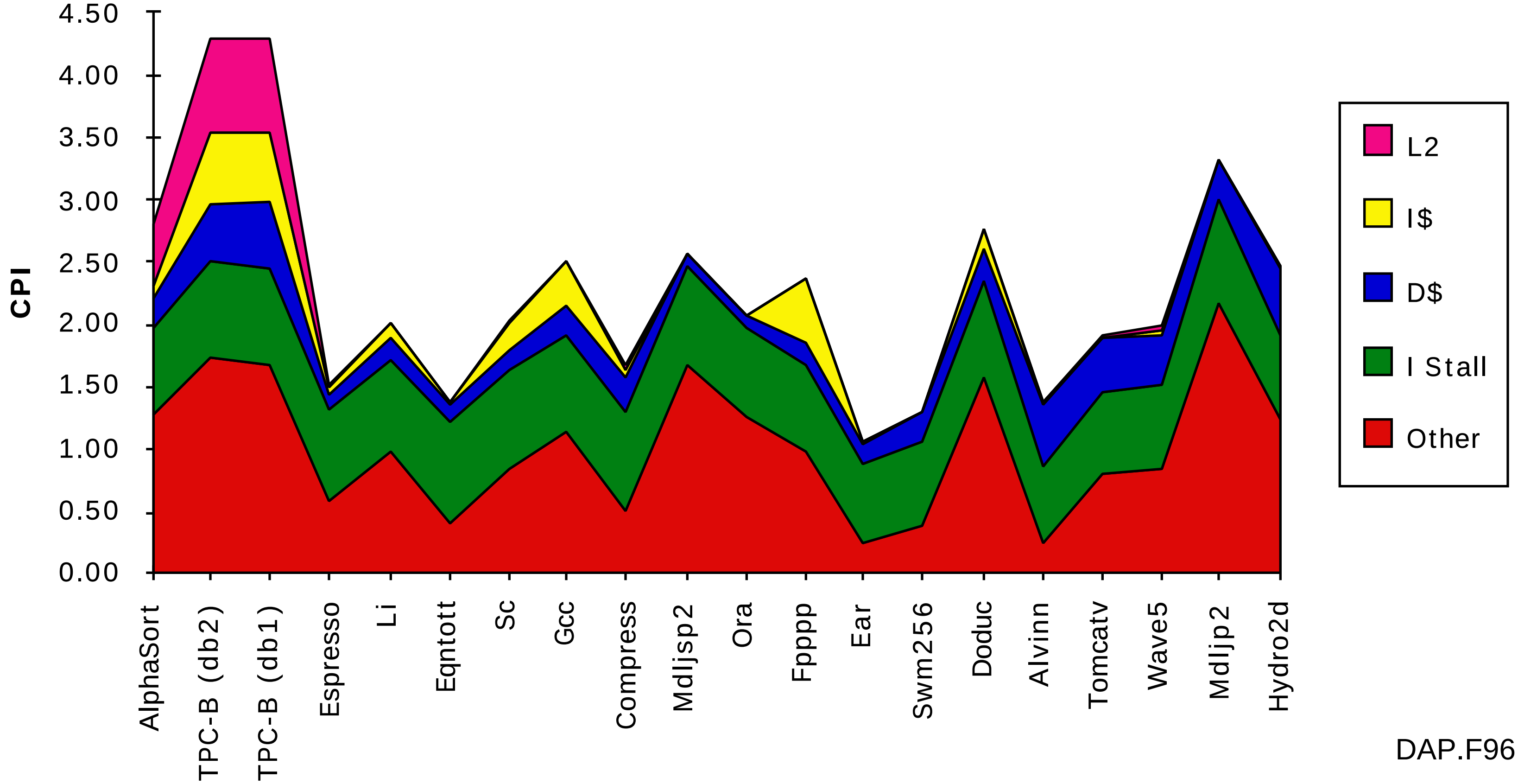


Alpha Memory Performance: Miss Rates of SPEC92



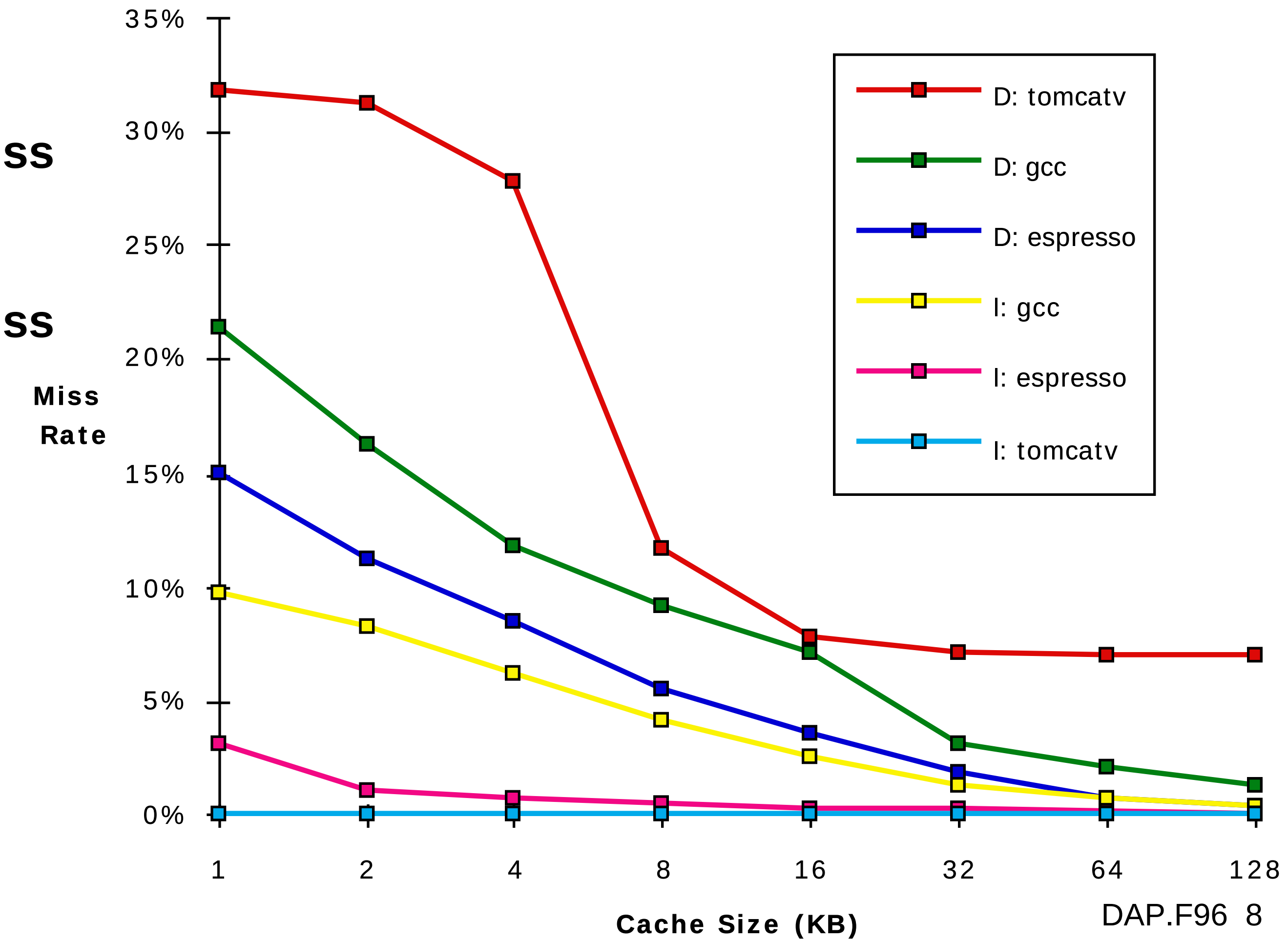
Alpha CPI Components

- **Instruction stall: branch mispredict;**
  Other: compute + reg conflicts, structural conflicts

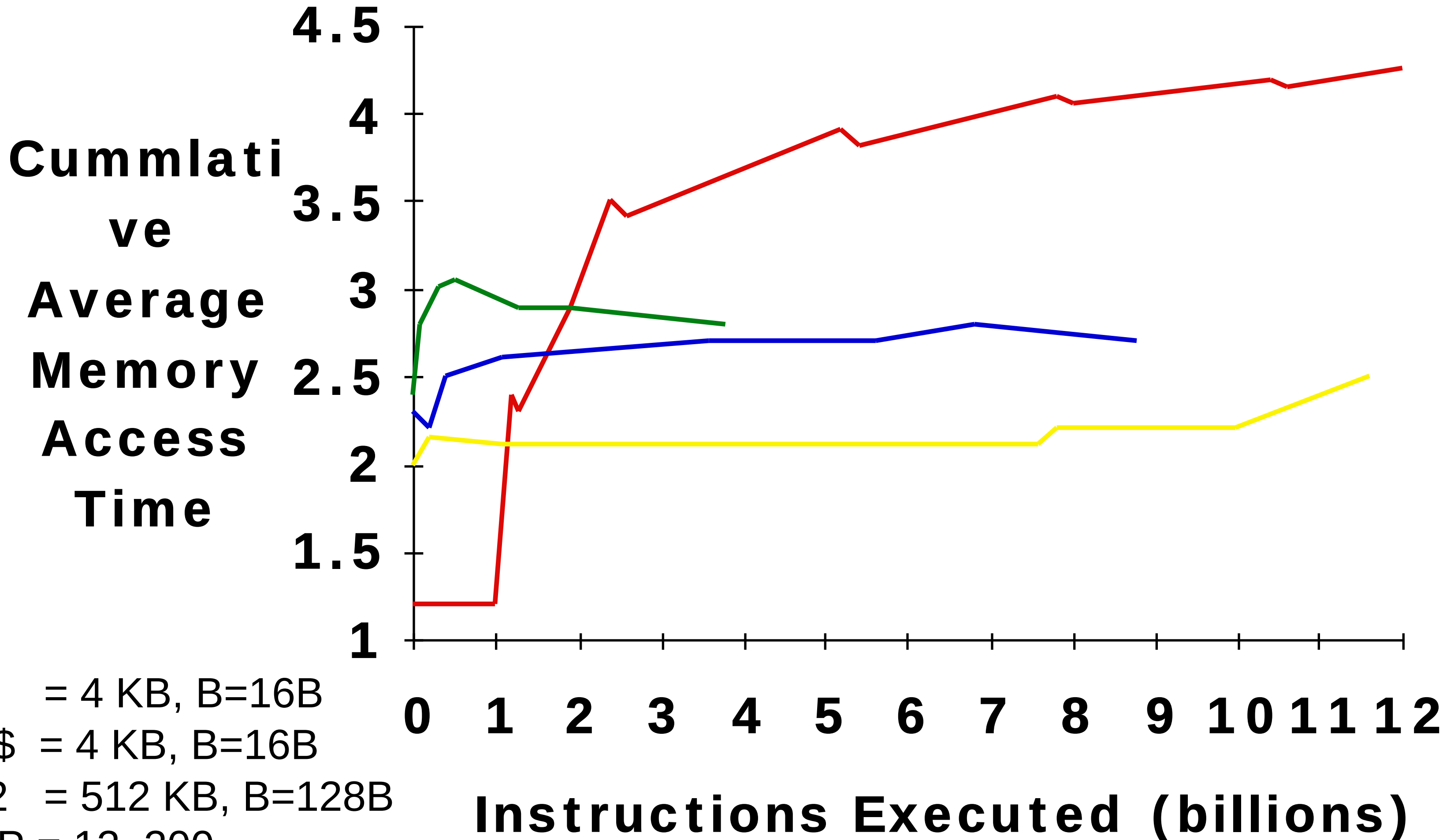


Pitfall: Predicting Cache Performance from Different Prog. (ISA, compiler, ...)

- **4KB Data cache miss rate 8%,12%, or 28%?**
- **1KB Instr cache miss rate 0%,3%, or 10%?**
- **Alpha vs. MIPS for 8KB Data: 17% vs. 10%**



Pitfall: Simulating Too Small an Address Trace



I$ = 4 KB, B=16B
D$ = 4 KB, B=16B
L2 = 512 KB, B=128B
MP = 12, 200

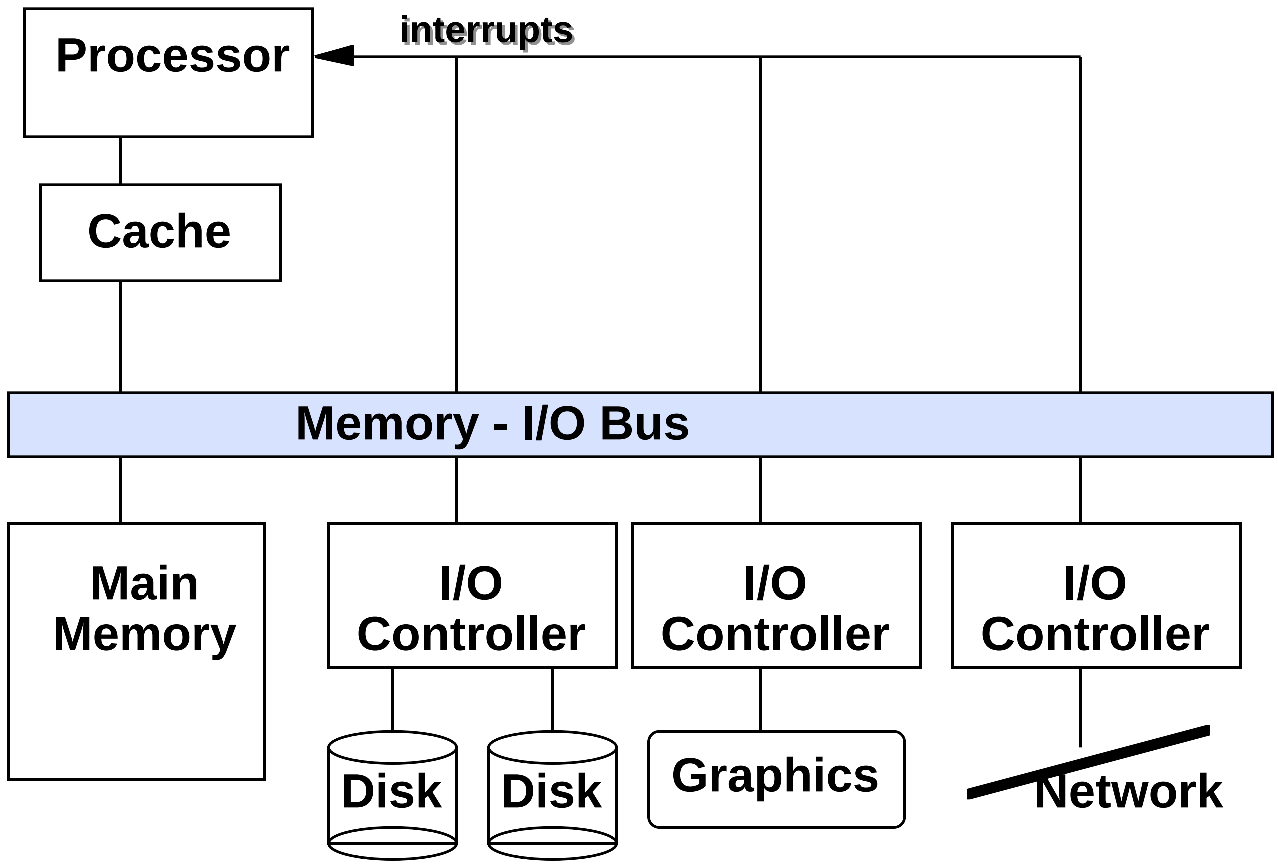
Motivation: Who Cares About I/O?

- **CPU Performance: 50% to 100% per year**
- **Multiprocessor supercomputers 150% per year**
- **I/O system performance limited by *mechanical* delays**
  < 10% per year (IO per sec or MB per sec)
- **Amdahl's Law: system speed-up limited by the slowest part!**
  10% IO & 10x CPU => 5x Performance (lose 50%)
  10% IO & 100x CPU => 10x Performance (lose 90%)
- **I/O bottleneck:**
  Diminishing fraction of time in CPU
  Diminishing value of faster CPUs

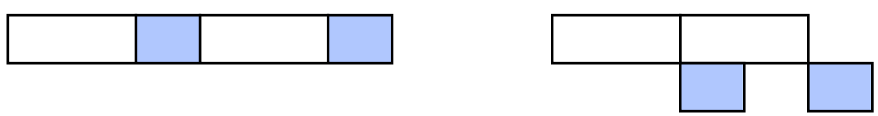
Storage System Issues

- Historical Context of Storage I/O
- Secondary and Tertiary Storage Devices
- Storage I/O Performance Measures
- A Little Queuing Theory
- **Processor Interface Issues**
- **I/O Buses**
- **Redundant Arrarys of Inexpensive Disks (RAID)**
- **ABCs of UNIX File Systems**
- **I/O Benchmarks**
- **Comparing UNIX File System Performance**

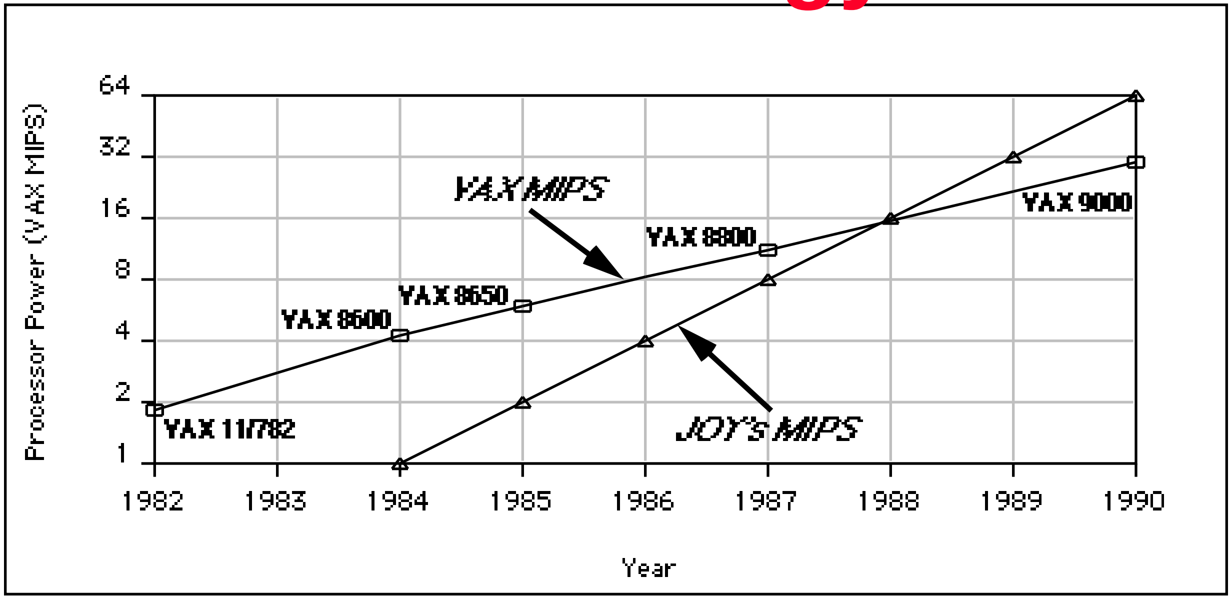
I/O Systems



Time(workload) = Time(CPU) + Time(I/O) - Time(Overlap)

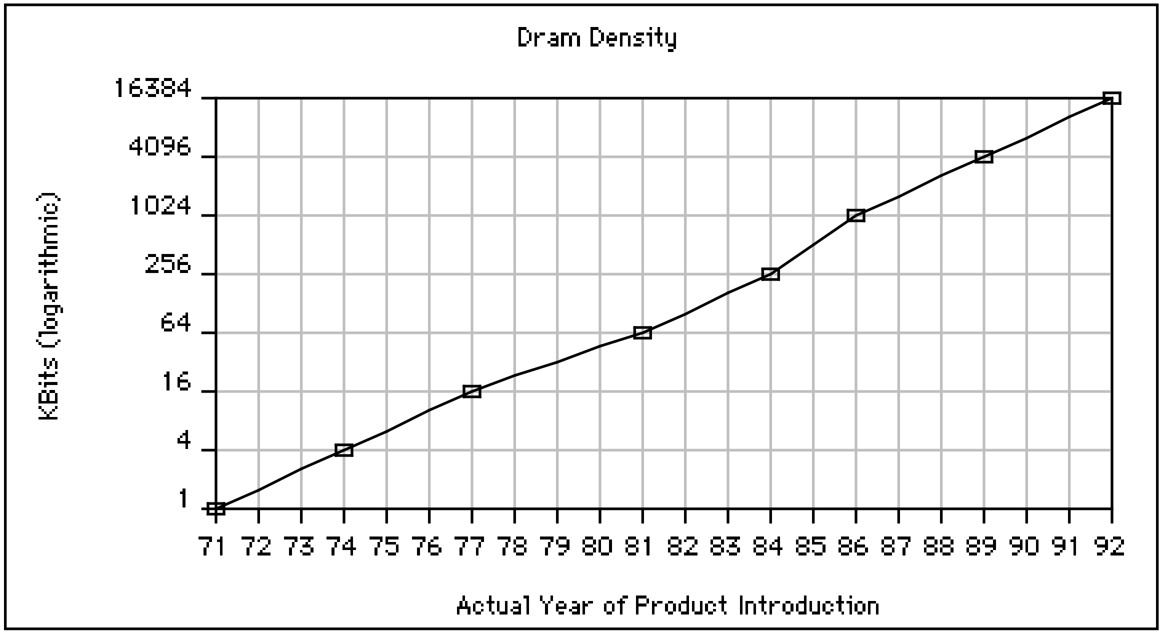


Technology Trends



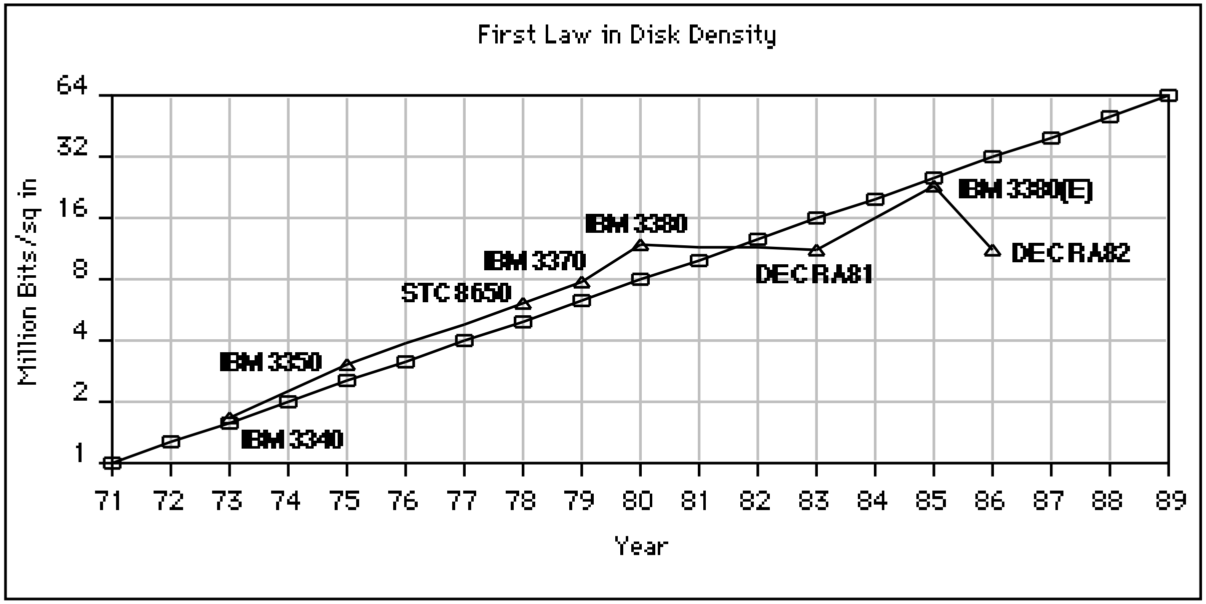
CPU Performance

- **Mini:**
  40% increase per year
- **RISC:**
  100% increase per year



DRAM Capacity
doubles every 3 years

Technology Trends



Disk Capacity **doubles every 3 years**

- **Today: Processing Power Doubles Every 18 months**
- **Today: Memory Size Doubles Every 18 months(?)**
- **Today: Disk Capacity Doubles Every 18 months**
- ***Disk Positioning Rate (Seek + Rotate) Doubles Every Ten Years!***

The I/O GAP

Storage Technology Drivers

- Driven by the prevailing computing paradigm
  - 1950s: migration from batch to on-line processing
  - 1990s: migration to ubiquitous computing
    - » computers in phones, books, cars, video cameras, …
    - » nationwide fiber optical network with wireless tails
- Effects on storage industry:
  - Embedded storage
    - » smaller, cheaper, more reliable, lower power
  - Data utilities
    - » high capacity, hierarchically managed storage

Historical Perspectives

- **1956 IBM Ramac — early 1970s Winchester**
  - **Developed for mainframe computers**
    - » **proprietary interfaces**
  - **Steady shrink in form factor: 27 in. to 14 in.**
    - » **driven by performance demands**

      higher rotation rate

      more actuators in the machine room

Historical Perspective

- 1970s developments
  - 5.25 inch floppy disk formfactor
    - » download microcode into mainframe
  - semiconductor memory and microprocessors
  - early emergence of industry standard disk interfaces
    - » ST506, SASI, SMD, ESDI

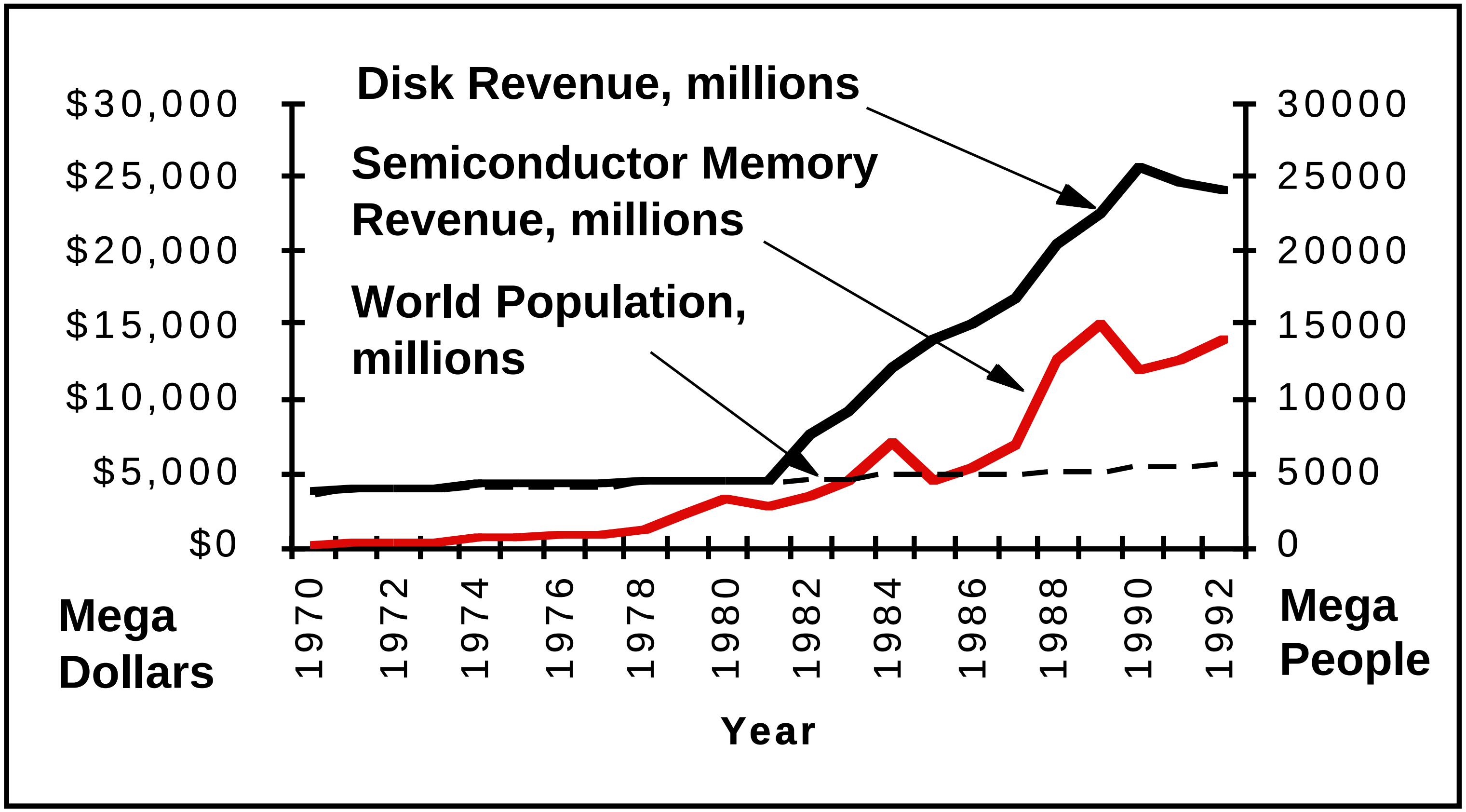
Historical Perspective

- **Early 1980s**
  - **PCs and first generation workstations**
- **Mid 1980s**
  - **Client/server computing**
  - **Centralized storage on file server**
    - » **accelerates disk downsizing**
    - » **8 inch to 5.25 inch**
  - **Mass market disk drives become a reality**
    - » **industry standards: SCSI, IPI, IDE**
    - » **5.25 inch drives for standalone PCs**
    - » **End of proprietary disk interfaces**

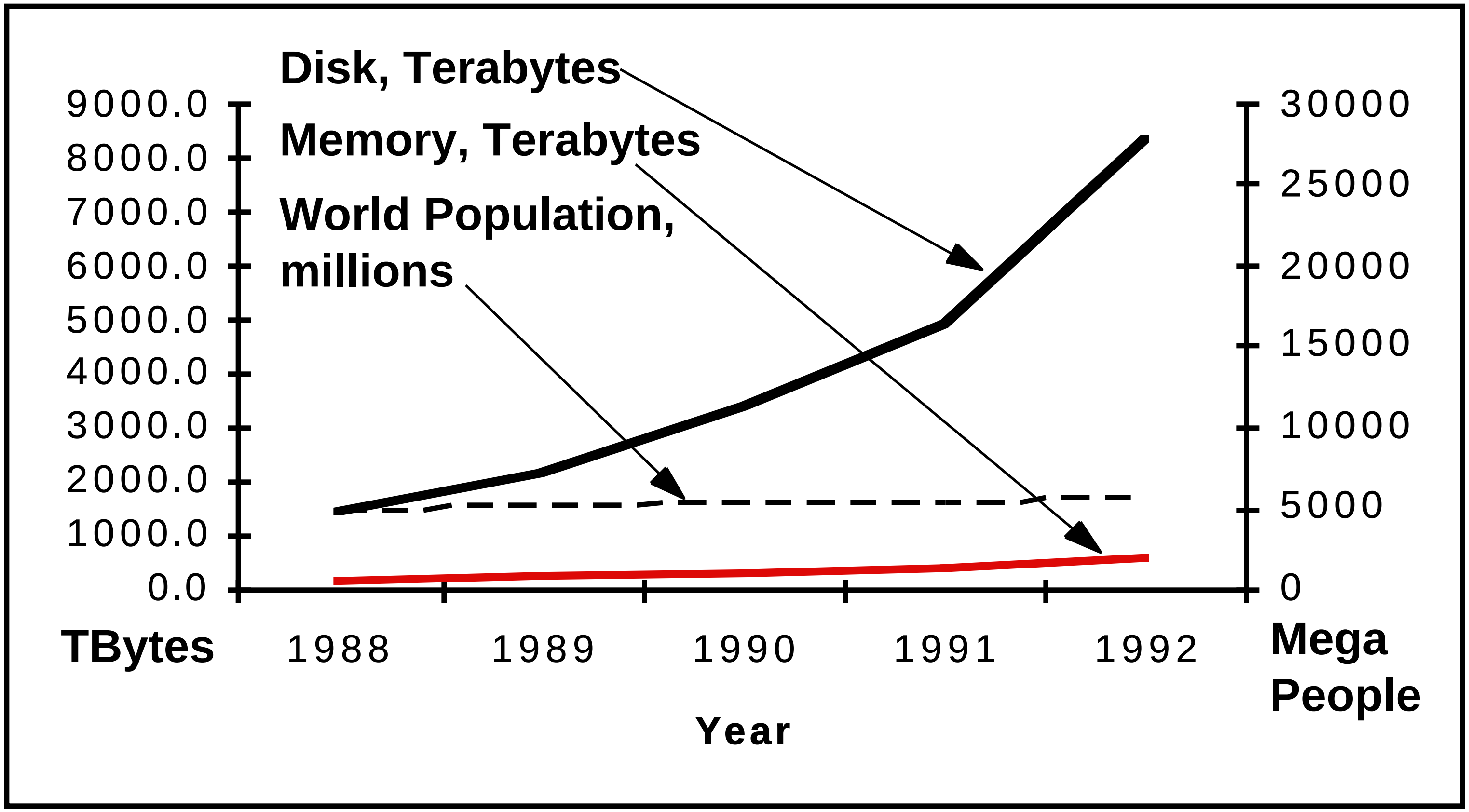
Historical Perspective

- Late 1980s/Early 1990s:
  - Laptops, notebooks, palmtops
  - 3.5 inch, 2.5 inch, 1.8 inch formfactors
  - Formfactor plus capacity drives market, not performance
  - Challenged by DRAM, flash RAM in PCMCIA cards
    - » still expensive, Intel promises but doesn't deliver
    - » unattractive MBytes per cubic inch
  - Optical disk fails on performace (e.g., NEXT) but finds niche (CD ROM)

Historical Perspective



Historical Perspectives



1.5 MBytes Disk per person on the earth sold in 1992
0.1 MBytes Memory per person on the earth sold in 1992

CS 252 Administrivia

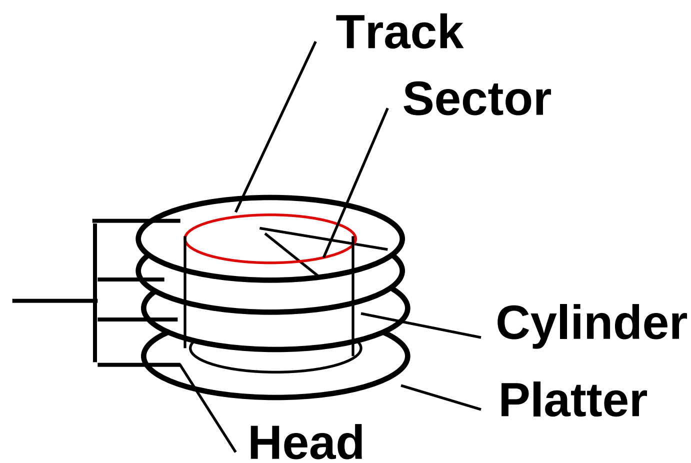
- **Midterm Quiz Wednesday October 8**
  5:45 - 8:45 PM in 306 Soda
  - 2 sheets with notes
  - Chapters 4, 5, and Ap B + Lectures
- **Answer questions during lecture time Wednesday**
- **Pizza at LaVal's after quiz; how many?**
- **8 minute project meetings for Friday October 4 (11-12:30, 2:10-3:10) in 635 Soda**
- **Email URL of initial project home page to TA**

Alternative Data Storage Technologies

Technology	Cap (MB)	BPI	TPI	BPI*TPI (Million)	Data Xfer (KByte/s)	Access Time
Conventional Tape:						
Cartridge (.25")	150	12000	104	1.2	92	minutes
IBM 3490 (.5")	800	22860	38	0.9	3000	seconds
Helical Scan Tape:						
Video (8mm)	4600	43200	1638	71	492	45 secs
DAT (4mm)	1300	61000	1870	114	183	20 secs
D-3 (1/2")	20,000					15 secs?
Magnetic & Optical Disk:						
Hard Disk (5.25")	1200	33528	1880	63	3000	18 ms
IBM 3390 (10.5")	3800	27940	2235	62	4250	20 ms
Sony MO (5.25")	640	24130	18796	454	88	100 ms

Devices: Magnetic Disks

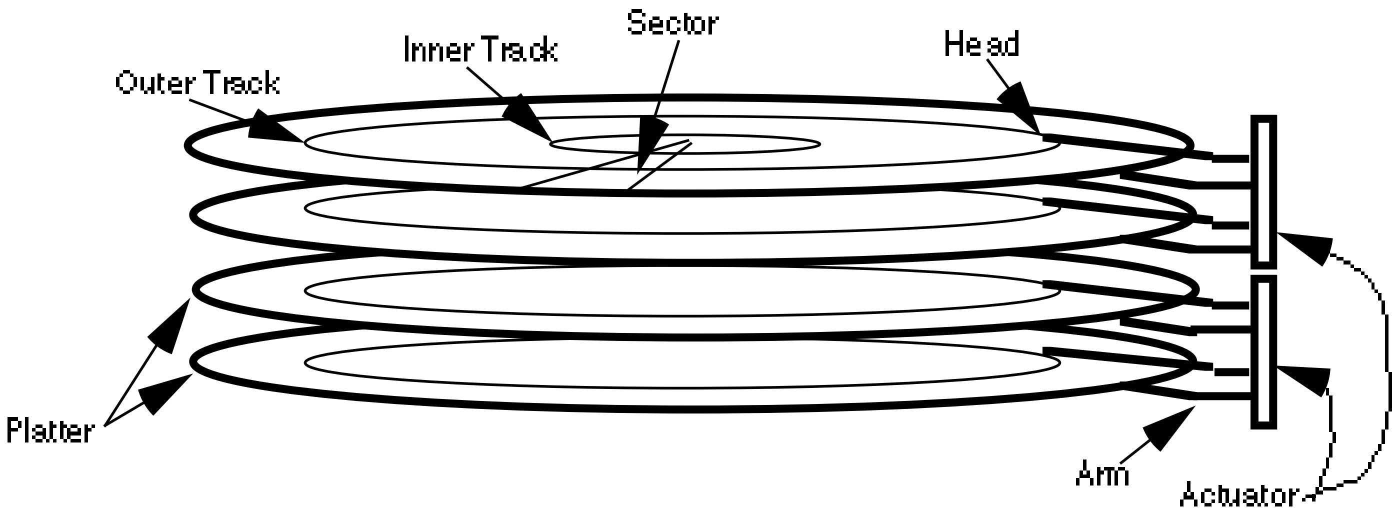
- **Purpose:**
  - Long-term, nonvolatile storage
  - Large, inexpensive, slow level in the storage hierarchy
- **Characteristics:**
  - Seek Time (~15 ms avg, 1M cyc at 50MHz)
    - » positional latency
    - » rotational latency
- **Transfer rate**
  - About a sector per ms (1-10 MB/s)
  - Blocks
- **Capacity**
  - Gigabytes
  - Quadruples every 3 years (aerodynamics)



3600 RPM = 60 RPS => 16 ms per rev
ave rot. latency = 8 ms
32 sectors per track => 0.5 ms per sector
1 KB per sector => 2 MB / s
32 KB per track
20 tracks per cyl => 640 KB per cyl
2000 cyl => 1.2 GB

Response time
$$= \underbrace{\text{Queue} + \text{Controller} + \text{Seek} + \text{Rot} + \text{Xfer}}_{\text{Service time}}$$

Disk Device Terminology



Disk Latency = Queuing Time + Seek Time + Rotation Time + Xfer Time

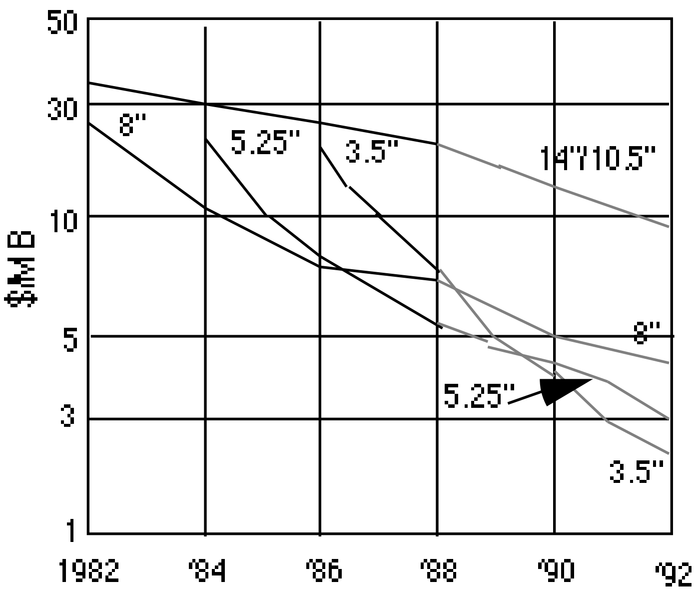
Order of magnitude times for 4K byte transfers:

Seek: 12 ms or less

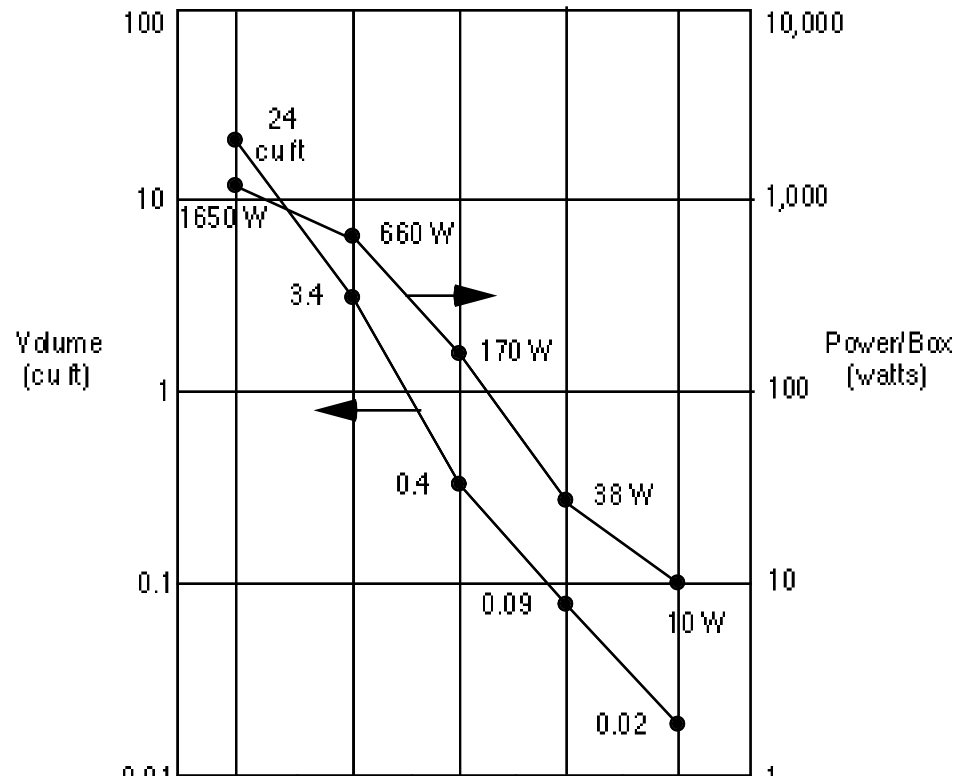
Rotate: 4.2 ms @ 7200 rpm (8.3 ms @ 3600 rpm)

Xfer: 1 ms @ 7200 rpm (2 ms @ 3600 rpm)

Advantages of Small Formfactor Disk Drives



Low cost/MB
High MB/volume
High MB/watt
Low cost/Actuator



	14"	10.5"	8"	5.25"	3.5"
MB	7,500 F	1,320 F	1,000	765	200
MB/watt	4.54	2.0	5.88	20.13	20.0
MB/cu ft	313	388	2,500	8,500	10,000

Cost and Environmental Efficiencies

Tape vs. Disk

- Longitudinal tape uses same technology as hard disk; tracks its density improvements

- Inherent cost-performance based on geometries: fixed rotating platters with gaps

  (random access, limited area, 1 media / reader)

vs.

  removable long strips wound on spool

  (sequential access, "unlimited" length, multiple / reader)

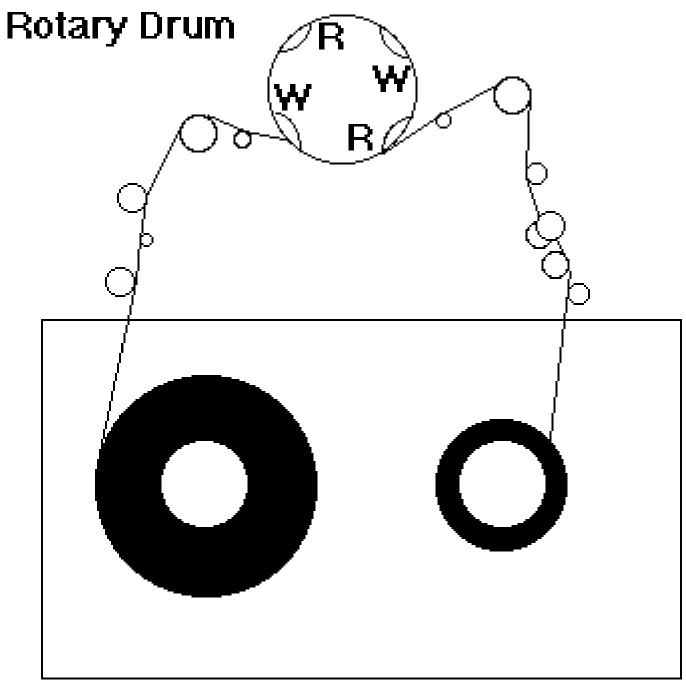
- New technology trend:
  Helical Scan (VCR, Camcoder, DAT)
  Spins head at angle to tape to improve density

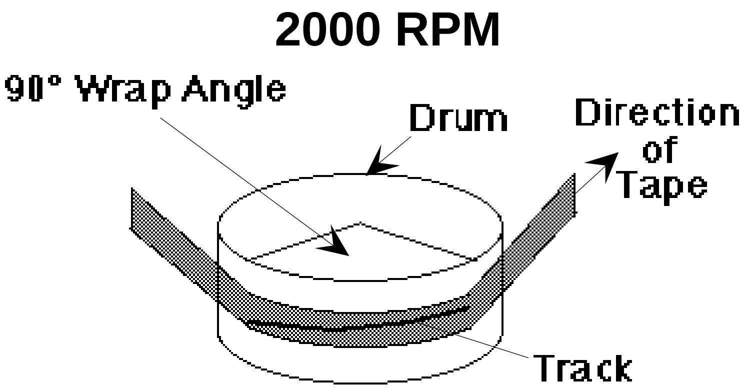
Example: R-DAT Technology

Rotating (vs. Stationary) head Digital Audio Tape

- **Highest areal recording density commercially available**
- **High density due to:**
  - **high coercivity metal tape**
  - **helical scan recording method**
  - **narrow, gapless (overlapping) recording tracks**
- **10X improvement capacity & xfer rate by 1999**
  - **faster tape and drum speeds**
  - **greater track overlap**

R-DAT Technology





Four Head Recording

Helical Recording Scheme

Tracks Recorded ±20° w/o guard band

Read After Write Verify

Optical Disk vs. Tape

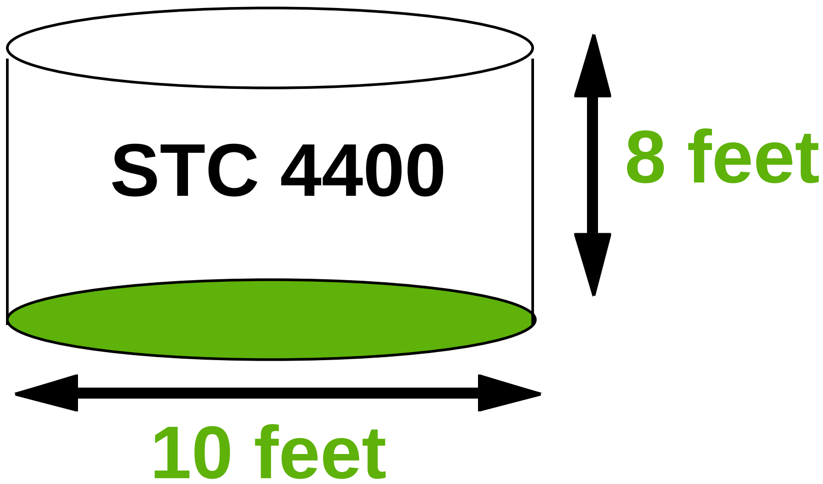
	Optical Disk	Helical Scan Tape
Type	5.25"	8mm
Capacity	0.75 GB	5 GB
Media Cost	$90 - $175	$8
Drive Cost	$3,000	$3,000
Access	Write Once	Read/Write
Robot Time	10 - 20 s	10 - 20 s

Media cost ratio optical disk vs. helical tape
=   75 : 1 to 150 : 1

Current Drawbacks to Tape

- **Tape wear out:**
  - **Helical 100s of passes to 1000s for longitudinal**
- **Head wear out:**
  - **2000 hours for helical**
- **Both must be accounted for in economic / reliability model**
- **Long rewind, eject, load, spin-up times; not inherent, just no need in marketplace (so far)**

Automated Cartridge System



6000 x 0.8 GB 3490 tapes = 5 TBytes in 1992
$500,000 O.E.M. Price

6000 x 20 GB D3 tapes = 120 TBytes in 1994
1 Petabyte (1024 TBytes) in 2000

Relative Cost of Storage Technology—Late 1995/Early 1996

Magnetic Disks

5.25”	9.1 GB	$2129	$0.23/MB
		$1985	$0.22/MB
3.5”	4.3 GB	$1199	$0.27/MB
		$999	$0.23/MB
2.5”	514 MB	$299	$0.58/MB
	1.1 GB	$345	$0.33/MB

Optical Disks

5.25”	4.6 GB	$1695+199	$0.41/MB
		$1499+189	$0.39/MB

PCMCIA Cards

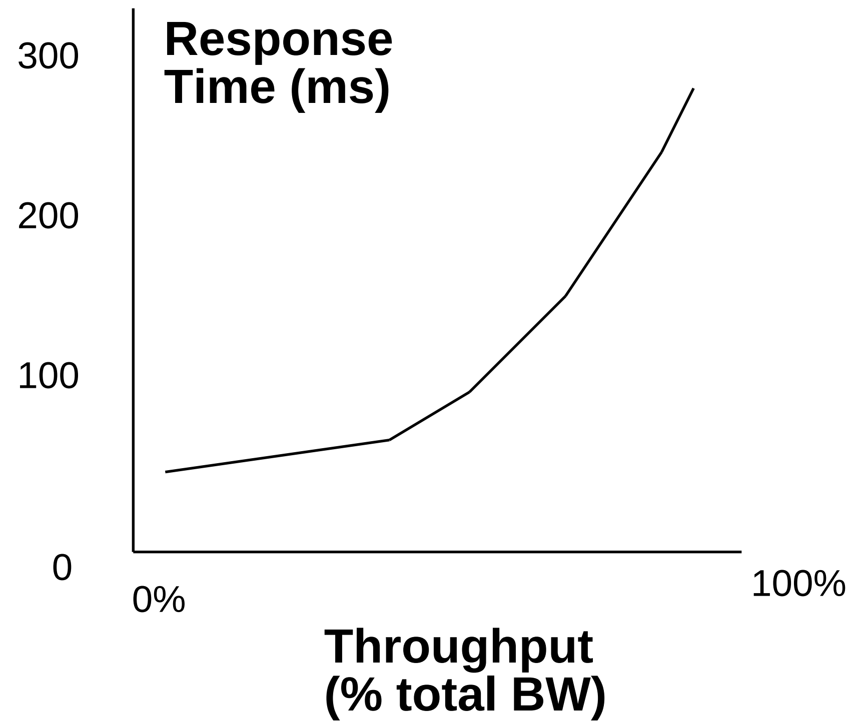
Static RAM	4.0 MB	$700	$175/MB
Flash RAM	40.0 MB	$1300	$32/MB
	175 MB	$3600	$20.50/MB

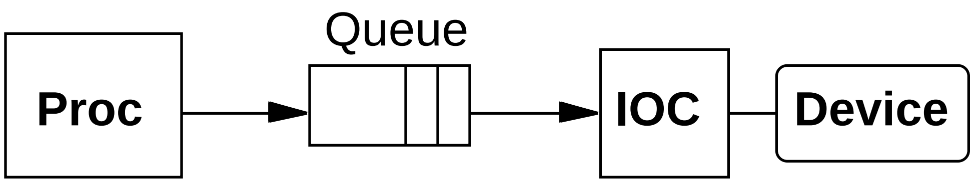
5 minute Class Break

- **Lecture Format:**
  - ≈ 1 minute: review last time & motivate this lecture
  - ≈ 20 minute lecture
  - ≈ 3 minutes: discuss class manangement
  - ≈ 25 minutes: lecture
  - 5 minutes: break
  - ≈25 minutes: lecture
  - ≈1 minute: summary of today's important topics

Disk I/O Performance

Metrics:
- **Response Time**
- **Throughput**





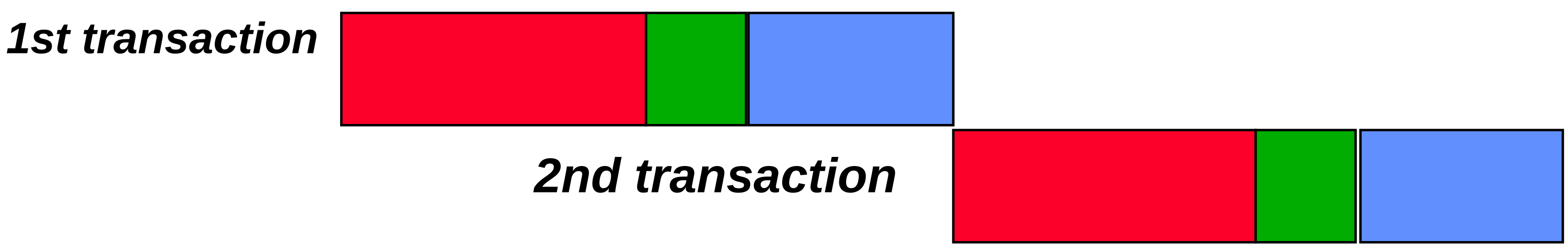
Response time = Queue + Device Service time

Response Time vs. Productivity

- **Interactive environments:**

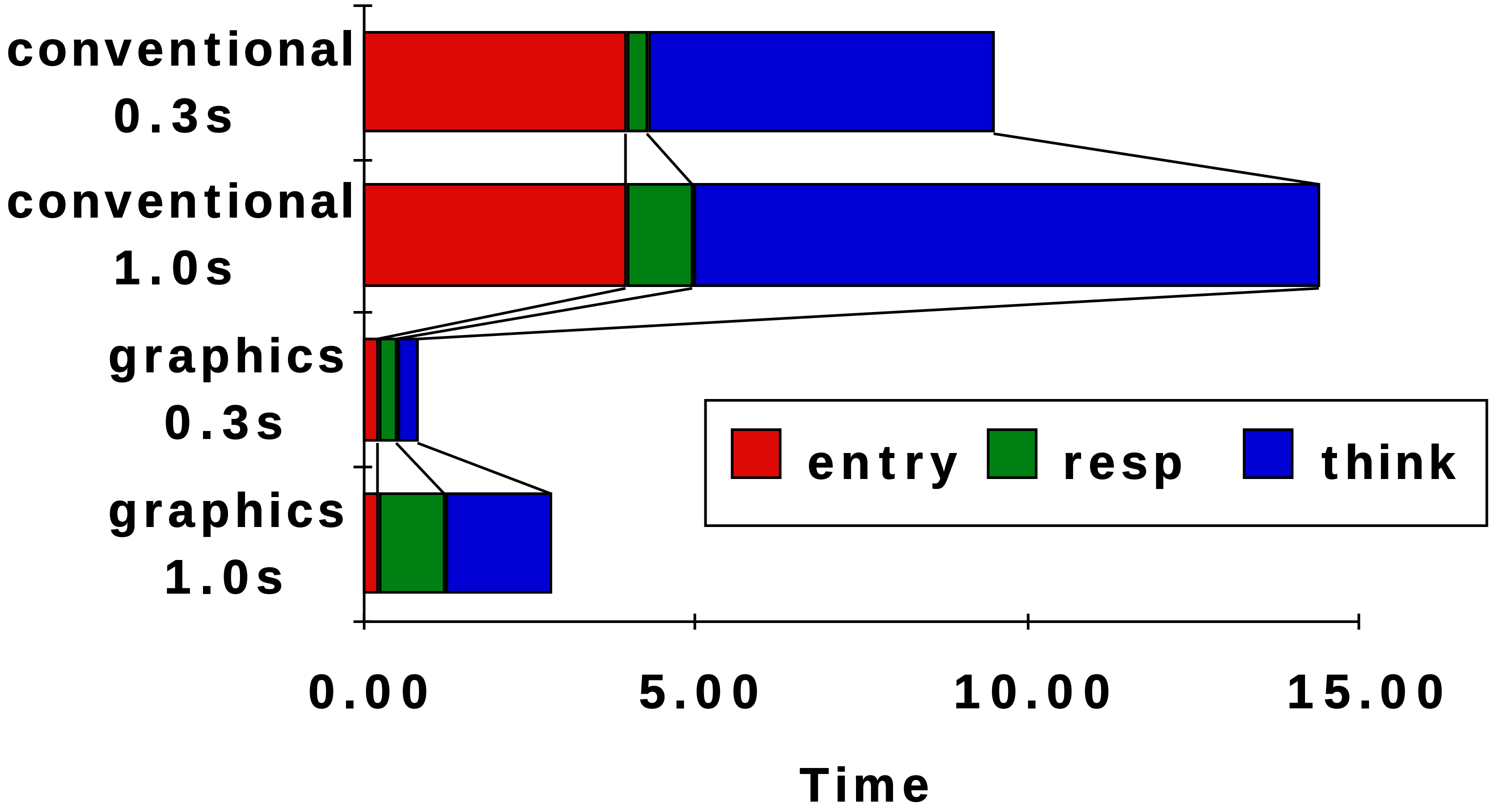
  Each interaction or *transaction* has 3 parts:

  - ***Entry Time*: time for user to enter command**
  - ***System Response Time*: time between user entry & system replies**
  - ***Think Time*: Time from response until user begins next command**



- **What happens to transaction time as shrink system response time from 1.0 sec to 0.3 sec?**
  - **With Keyboard: 4.0 sec entry, 9.4 sec think time**
  - **With Graphics: 0.25 sec entry, 1.6 sec think time**

Response Time & Productivity



- 0.7sec off response saves 4.9 sec (34%) and 2.0 sec (70%) total time per transaction => greater productivity
- Another study: everyone gets more done with faster response, but novice with fast response = expert with slow

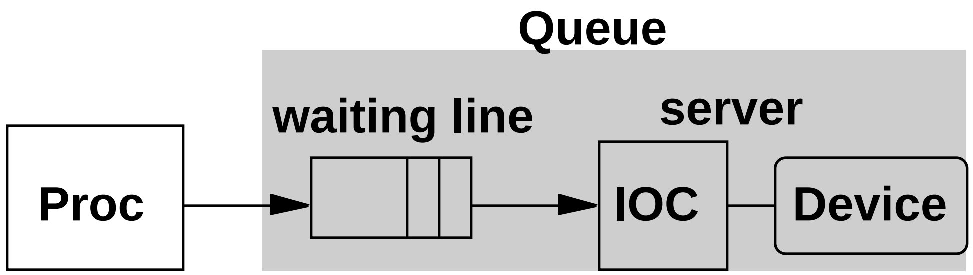
Disk Time Example

- **Disk Parameters:**
  - **Transfer size is 8K bytes**
  - **Advertised average seek is 12 ms**
  - **Disk spins at 7200 RPM**
  - **Transfer rate is 4 MB/sec**
- **Controller overhead is 2 ms**
- **Assume that disk is idle so no queuing delay**
- **What is Average Disk Access Time for a Sector?**
  - **Ave seek + ave rot delay + transfer time + controller overhead**
  - **12 ms + 0.5/(7200 RPM/60) + 8 KB/4 MB/s + 2 ms**
  - **12 + 4.15 + 2 + 2 = 20 ms**
- **Advertised seek time assumes no locality: typically 1/4 to 1/3 advertised seek time: 20 ms => 12 ms**

INtroduction To Queueing Theory

- More interested in long term, steady state than in startup => Arrivals = Departures
- <u>Little's Law</u>: Mean number tasks in system = arrival rate x mean reponse time
- Applies to any system in equilibrium, as long as nothing in black box is creating or destroying tasks

A Little Queuing Theory: Litttle's Theorem

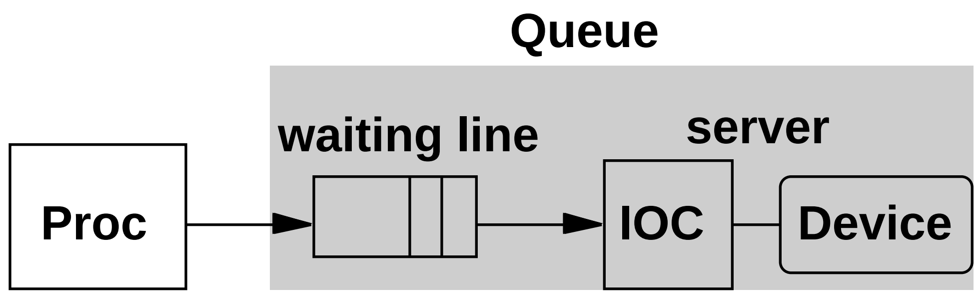


- **Queuing models assume state of equilibrium: input rate = output rate**
- **Notation:**

r	average number of arriving customers/second
T_s	average time to service a customer ($\mu = 1/ T_s$)
u	server utilization (0..1): $u = r \times T_s$
T_w	average time/customer in waiting line
T_q	average time/customer in queue: $T_q = T_w + T_s$
L_w	average length of waiting line: $L_w = r \times T_w$
L_q	average length of queue: $L_q = r \times T_q$

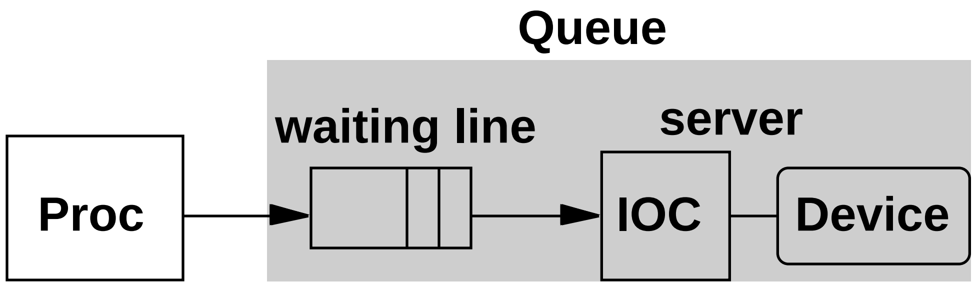
- **Little's Law: $L_q = r \times T_q$**
  Mean number customers = arrival rate x mean service time

A Little Queuing Theory



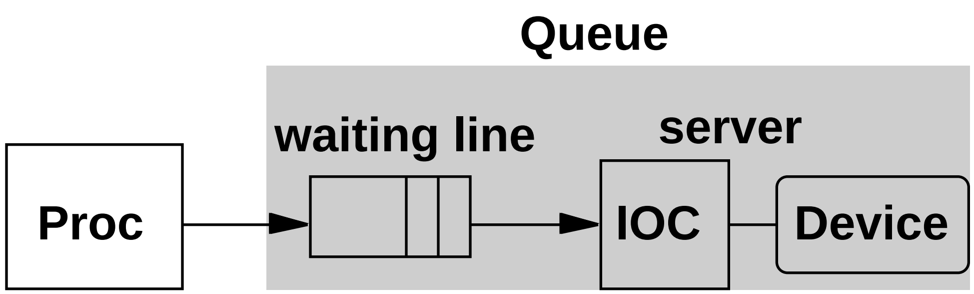
- Service time completions vs. waiting time for a busy server when randomly arriving event joins a waiting line of arbitrary length when server is busy, otherwise serviced immediately
- A *single server queue*: combination of a servicing facility that accomodates 1 customer at a time (*server*) + waiting area (*waiting line*): together called a *queue*
- Server spends a variable amount of time with customers; *how do you characterize variability?*
  – Distribution of a random variable: histogram? curve?

A Little Queuing Theory



- **Server spends a variable amount of time with customers**
  - **Weighted mean *m1* = (f1 x T1 + f2 x T2 +...+ fn x Tn)/F (F=f1 + f2...)**
  - ***variance* = (f1 x T1^2 + f2 x T2^2 +...+ fn x Tn^2)/F – m1^2**
    - » **Changes depending on unit of measure (100 ms vs. 0.1 s)**
  - ***Squared coefficient of variance*: *C = variance/m1^2***
- **Exponential distribution *C = 1* : most short relative to average, few others long; 90% < 2.3 x average, 63% < average**
- **Hypoexponential distribution *C < 1* : most close to average, C=0.5 => 90% < 2.0 x average, only 57% < average**
- **Hyperexponential distribution *C > 1* : further from average C=2.0 => 90% < 2.8 x average, 69% < average**

A Little Queuing Theory: Variable Service Time



- Server spends a variable amount of time with customers
  - Weighted mean *m1* = (f1xT1 + f2xT2 +...+ fnXTn)/F (F=f1+f2+...)
  - Squared coefficient of variance *C*
- Disk response times *C ≈ 1.5* (majority seeks < average)
- Yet usually pick *C = 1.0* for simplicity
- Another useful value is average time must wait for server to complete task: *m1(z)*
  - Not just 1/2 x m1 because doesn't capture variance
  - Can derive *m1(z) = 1/2 x m1 x (1 + C)*
  - *No variance => C= 0 => m1(z) = 1/2 x m1*

A Little Queuing Theory: Average Wait Time

- **Calculating average wait time T_w**
  - **If something at server, it takes to complete on average $m1(z)$**
  - **Chance server is busy = u; average delay is u x $m1(z)$**
  - **All customers in line must complete; each avg T_s**

$$T_w = u \times \underline{m1(z)} + L_w \times T_s = 1/2 \times u \times T_s \times (1 + C) + \underline{L_w} \times T_s$$
$$T_w = 1/2 \times u \times T_s \times (1 + C) + \underline{r} \times T_w \times \underline{T_s}$$
$$T_w = 1/2 \times u \times T_s \times (1 + C) + \underline{u \times T_w}$$
$$T_w \times \underline{(1 - u)} = T_s \times u \times (1 + C) /2$$
$$T_w = T_s \times u \times (1 + C) / (2 \times (1 - u))$$

- **Notation:**

r	**average number of arriving customers/second**
T_s	**average time to service a customer**
u	**server utilization (0..1): $u = r \times T_s$**
T_w	**average time/customer in waiting line**
L_w	**average length of waiting line: $L_w = r \times T_w$**

A Little Queuing Theory: M/G/1 and M/M/1

- **Assumptions so far:**
  - System in equilibrium
  - Time between two successive arrivals in line are random
  - Server can start on next customer immediately after prior finishes
  - No limit to the waiting line: works First-In-First-Out
  - Afterward, all customers in line must complete; each avg T_s
- **Described "memoryless" Markovian request arrival (M for C=1 exponentially random), General service distribution (no restrictions), 1 server: *M/G/1 queue***
- **When Service times have C = 1, *M/M/1 queue***

  $T_w = T_s \times u \times (1 + C) / (2 \times (1 - u)) = T_s \times u / (1 - u)$

  T_s average time to service a customer

  u server utilization (0..1): $u = r \times T_s$

  T_w average time/customer in waiting line
- **Note distinction between waiting time and queue delay**

A Little Queuing Theory: An Example

- **Suppose processor sends 10 x 8KB disk I/Os per second, requests exponentially distrib., disk service time = 20 ms**
- **On average, how utilized is the disk?**
  - **What is the number of requests in the waiting line?**
  - **What is the average time spent in the waiting line?**
  - **What is the average response time for a disk request?**
- **Notation:**

r **average number of arriving customers/second = 10**
T_s **average time to service a customer = 20 ms**
u **server utilization (0..1):** $u = r \times T_s$ **= 10/s x .02s = 0.2**
T_w **average time/customer in waiting line** $= T_s \times u / (1 - u)$
= 20 x 0.2/(1-0.2) = 20 x 0.25 = 5 ms
T_q **average time/customer in queue:** $T_q = T_w + T_s$ **= 25 ms**
L_w **average length of waiting line:** $L_w = r \times T_w$
= 10/s x .005s = 0.05 requests in wait line
L_q **average length of "queue":** $L_q = r \times T_q$ **= 10/s x .025s = 0.25**

A Little Queuing Theory: Another Example

- **Suppose processor sends 20 x 8KB disk I/Os per sec, requests exponentially distrib., disk service time = 12 ms**
- **On average, how utilized is the disk?**
  - **What is the number of requests in the waiting line?**
  - **What is the average time a spent in the waiting line?**
  - **What is the average response time for a disk request?**
- **Notation:**

r average number of arriving customers/second= 20
T_s average time to service a customer= 12 ms
u server utilization (0..1): $u = r \times T_s$= 20/s x .012s = 0.24
T_w average time/customer in waiting line $= T_s \times u / (1 - u)$
= 12 x 0.24/(1-0.24) = 12 x 0.32 = 3.8 ms
T_q average time/customer in queue: $T_q = T_w + T_s$= 16 ms
L_w average length of waiting line: $L_w = r \times T_w$
= 20/s x .0038s = 0.016 requests in wait line
L_q average length of "queue": $L_q = r \times T_q$= 20/s x .016s = 0.32

A Little Queuing Theory: Yet Another Example

- **Suppose processor sends 10 x 8KB disk I/Os per second, req. squared coef. var. = 1.5, disk service time = 20 ms**
- **On average, how utilized is the disk?**
  - **What is the number of requests in the waiting line?**
  - **What is the average time a spent in the waiting line?**
  - **What is the average response time for a disk request?**
- **Notation:**

r **average number of arriving customers/second= 10**
T_s **average time to service a customer= 20 ms**
u **server utilization (0..1):** $u = r \times T_s$ **= 10/s x .02s = 0.2**
T_w **average time/customer in waiting line** $= T_s \times u \times (1 + C) / (2 \times (1 - u))$
= 20 x 0.2(2.5)/2(1 – 0.2) = 20 x 0.32 = 6.25 ms
T_q **average time/customer in queue:** $T_q = T_w + T_s$ **= 26 ms**
L_w **average length of waiting line:** $L_w = r \times T_w$
= 10/s x .006s = 0.06 requests in wait line
L_q **average length of “queue”:** $L_q = r \times T_q$ **= 10/s x .026s = 0.26**

Summary: Storage System Issues

- Historical Context of Storage I/O
- Secondary and Tertiary Storage Devices
- Storage I/O Performance Measures
- A Little Queuing Theory
- Processor Interface Issues
- I/O Buses
- Redundant Arrarys of Inexpensive Disks (RAID)
- ABCs of UNIX File Systems
- I/O Benchmarks
- Comparing UNIX File System Performance