CS61C : Machine Structures

Lecture 40
Performance I

Lecturer PSOE Dan Garcia

www.cs.berkeley.edu/~ddgarcia

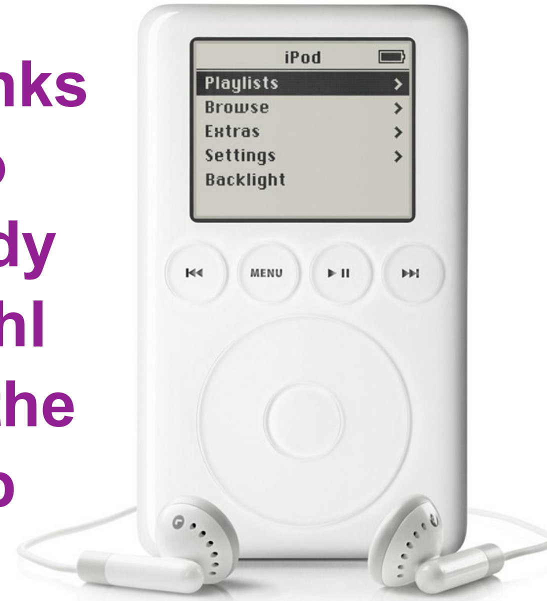
Hybrid Hard Drives (HHT) ⇒ Samsung & MS announced new drives, which would use flash memory to cache information on disk, so the drive could spin down & save power when on, as well as boot much faster.

www.samsung.com/PressCenter/PressRelease/PressRelease.asp?seq=20050425_0000116210

Cool addition to the last lecture

- **Drives inside the iPod and iPod Mini:**

Thanks to Andy Dahl for the tip



Hitachi 1 inch 4GB MicroDrive

Toshiba 1.8-inch 20/40/60GB (MK1504GAL)

Review

- Magnetic disks continue rapid advance: 2x/yr capacity, 2x/2-yr bandwidth, slow on seek, rotation improvements, MB/$ 2x/yr!
  - Designs to fit high volume form factor
- RAID
  - Motivation: In the 1980s, there were 2 classes of drives: expensive, big for enterprises and small for PCs. They thought "make one big out of many small!"
  - Higher performance with more disk arms per $
  - Adds option for small # of extra disks (the "R")
  - Started @ Cal by CS Profs Katz & Patterson


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

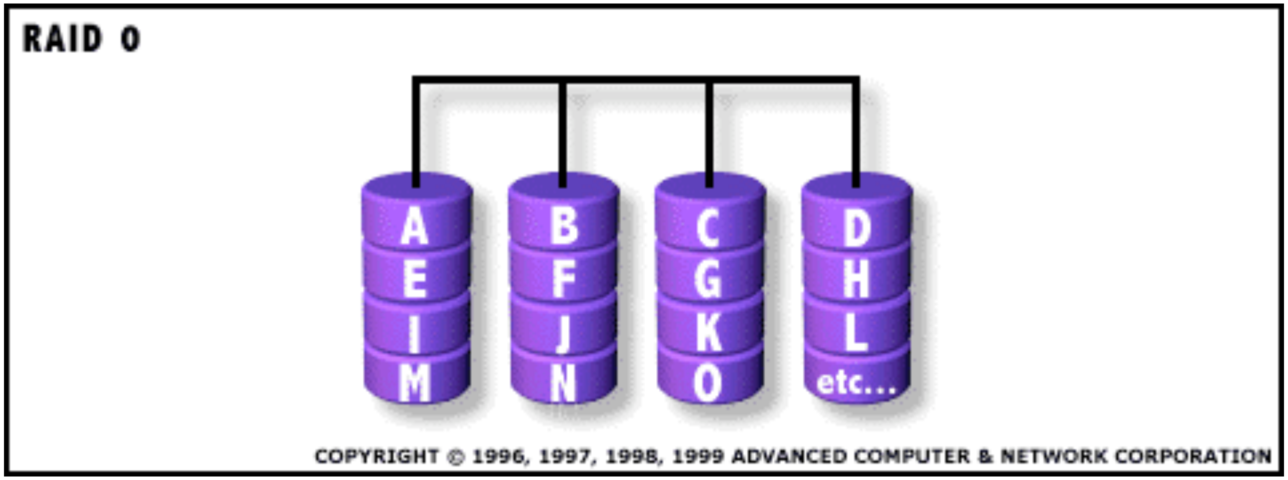
  ⇒ Capacity penalty to store redundant info

  ⇒ 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

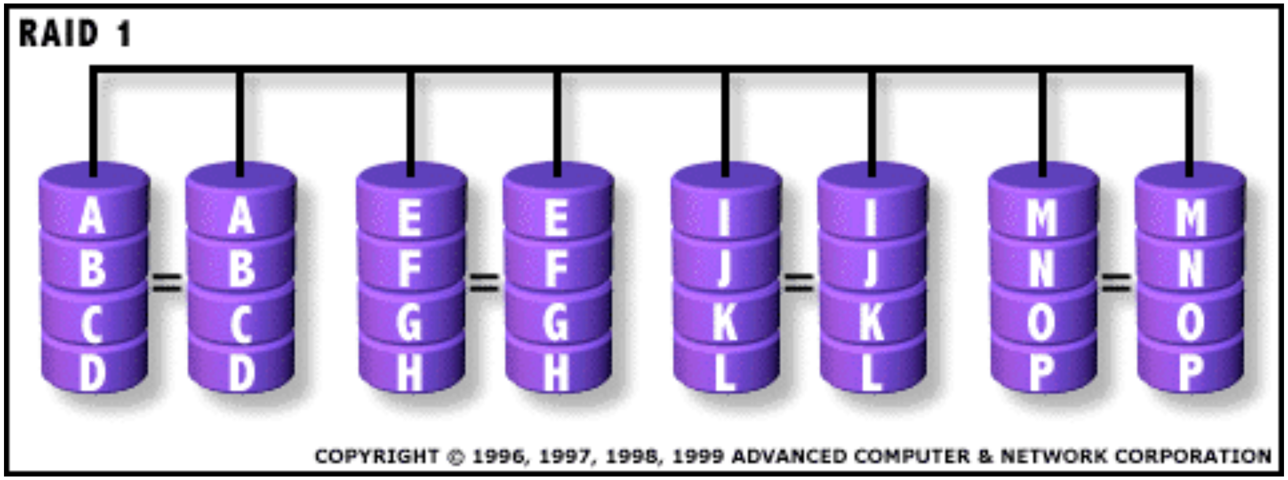
“RAID 0”: No redundancy = “AID”



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

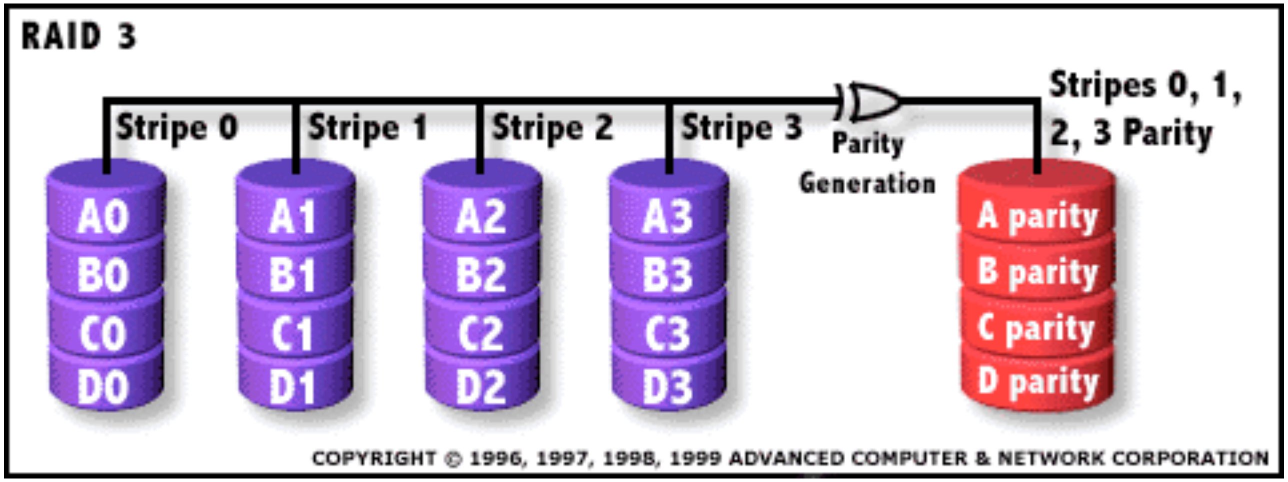
This and next 5 slides from RAID.edu, http://www.acnc.com/04_01_00.html

RAID 1: Mirror data



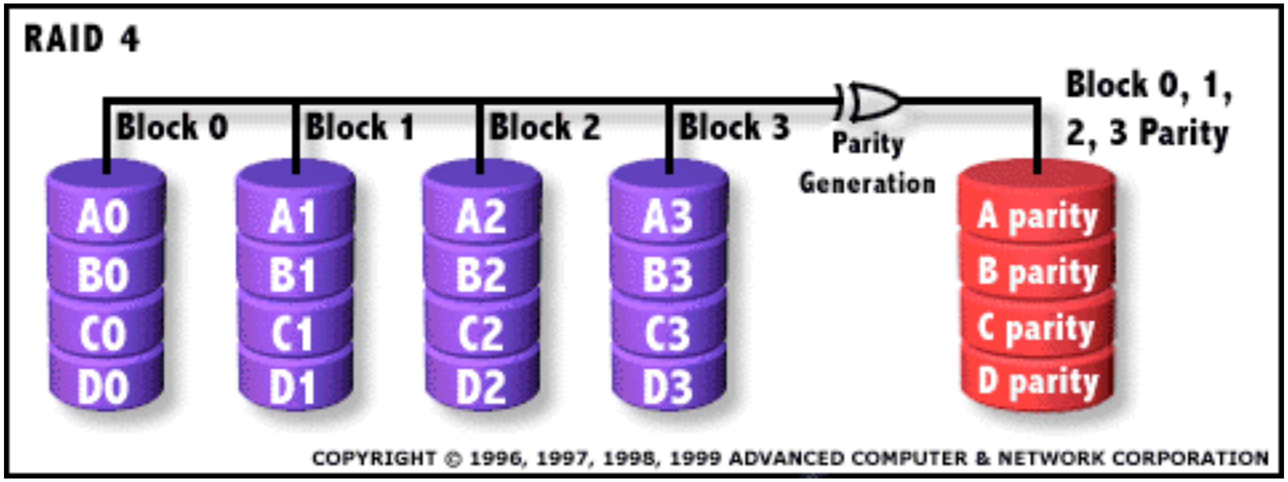
- Each disk is fully duplicated onto its "mirror"
  - Very high availability can be achieved
- Bandwidth reduced on write:
  - 1 Logical write = 2 physical writes
- Most expensive solution: 100% capacity overhead

RAID 3: Parity



- **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)**

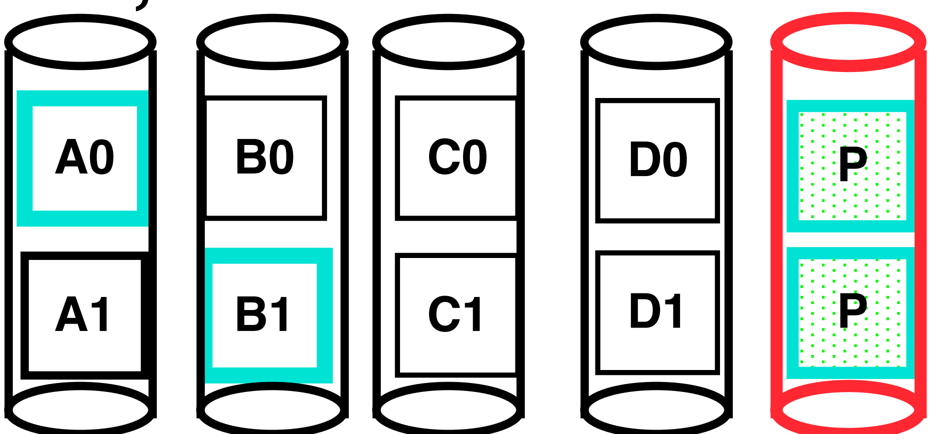
RAID 4: parity plus small sized accesses



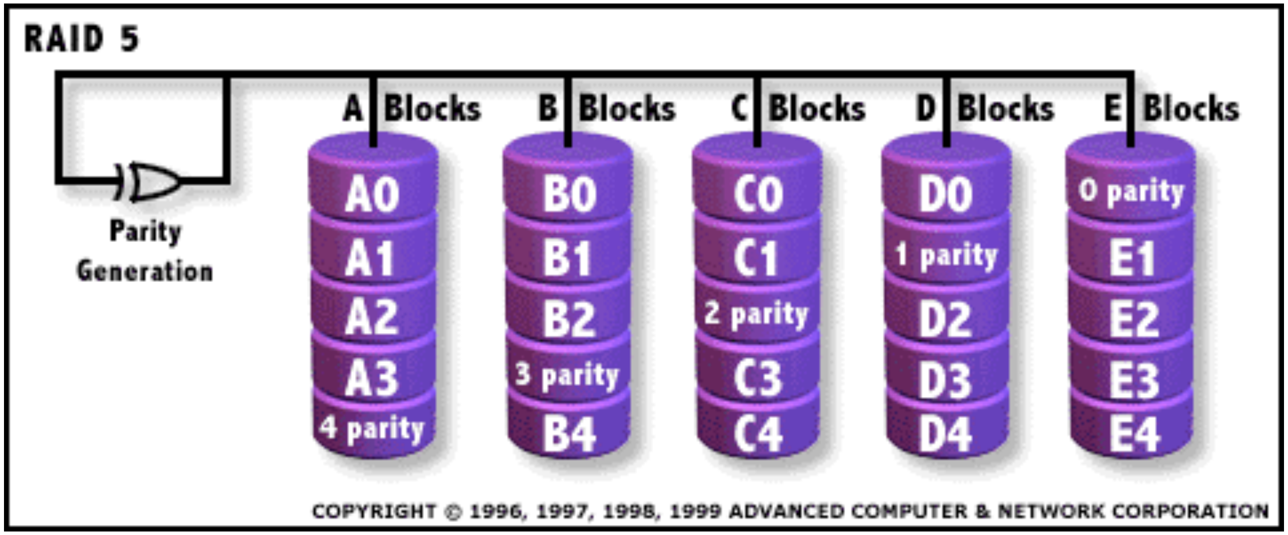
- RAID 3 relies on parity disk to discover errors on Read
- But every sector has an error detection field
- Rely on error detection field to catch errors on read, not on the parity disk
- Allows small independent reads to different disks simultaneously

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

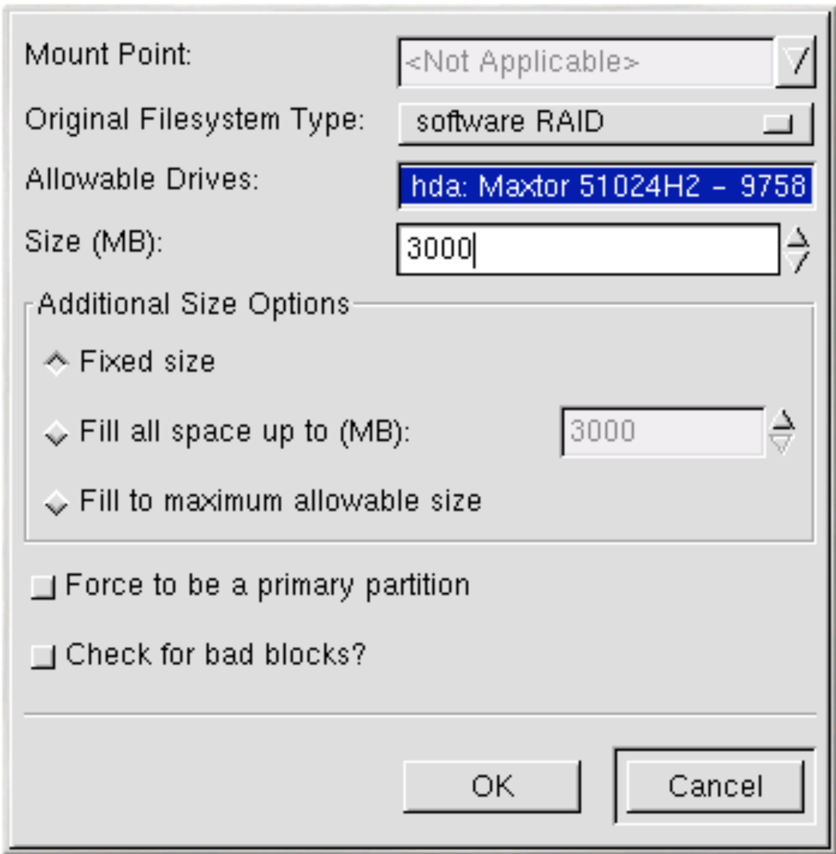


RAID 5: Rotated Parity, faster small writes



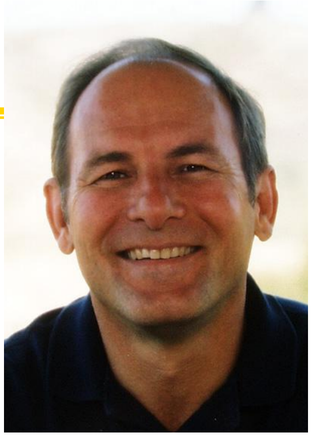
- Independent writes possible because of interleaved parity
  - Example: write to A0, B1 uses disks 0, 1, 4, 5, so can proceed in parallel
  - Still 1 small write = 4 physical disk accesses

RAID products: Software, Chips, Systems



RAID is $32 B industry in 2002, 80% nonPC disks sold in RAIDs

Margin of Safety in CS&E?



- Patterson reflects…
  - Operator removing good disk vs. bad disk
  - Temperature, vibration causing failure before repair
  - In retrospect, suggested RAID 5 for what we anticipated, but should have suggested RAID 6 (double failure OK) for unanticipated/safety margin…

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

Peer Instruction Answer

1. All RAID (0-5) helps with performance, only RAID 0 doesn't help availability. TRUE
2. Surely! Must buy 2x disks rather than 1.25x (from diagram, in practice even less) FALSE
3. RAID5 (2R,2W) vs. RAID1 (2W). Latency worse, throughput (ll writes) better. TRUE

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

Administrivia

- **Last semester's final + answers online soon**
- **HKN evaluations next Monday**
- **Final survey in lab this week**
- **Final exam review**
  - **Sunday, 2005-05-08 in the aft (location TBA)**
- **Final exam**
  - **Saturday, 2005-05-14 @ 12:30-3:30pm (loc TBA)**
  - **Same rules as Midterm, except you get 2 double-sided handwritten review sheets (1 from your midterm, 1 new one) + green sheet**

Upcoming Calendar

Week #	Mon	Wed	Thu Lab	Fri
#15 This week	Perfor-mance I	Perfor-mance II	I/O Networks	TA Andy TBD
#16 Next Week Sun aft Review	LAST CLASS Summary Review & HKN Evals			FINAL EXAM SAT 05-14 @ 12:30pm

Performance

- **Purchasing Perspective**: given a collection of machines (or upgrade options), which has the
  - best performance ?
  - least cost ?
  - best performance / cost ?
- **Computer Designer Perspective**: faced with design options, which has the
  - best performance improvement ?
  - least cost ?
  - best performance / cost ?
- All require basis for comparison and metric for evaluation
  - Solid metrics lead to solid progress!

Two Notions of "Performance"

Plane	DC to Paris	Top Speed	Passen-gers	Throughput (pmph)
Boeing 747	6.5 hours	610 mph	470	286,700
BAD/Sud Concorde	3 hours	1350 mph	132	178,200

- Which has higher performance?
  - Time to deliver 1 passenger?
  - Time to deliver 400 passengers?
- In a computer, time for 1 job called
  Response Time or Execution Time
- In a computer, jobs per day called
  Throughput or Bandwidth

Definitions

- **Performance is in units of things per sec**
  - **bigger is better**
- **If we are primarily concerned with response time**
  - **performance(x) =** $\dfrac{1}{\text{execution\_time}(x)}$

" F(ast) is *n* times faster than S(low) " means…

$$n = \frac{\text{performance}(F)}{\text{performance}(S)} = \frac{\text{execution\_time}(S)}{\text{execution\_time}(F)}$$

Example of Response Time v. Throughput

- Time of Concorde vs. Boeing 747?
  - Concord is 6.5 hours / 3 hours = 2.2 times faster
- Throughput of Boeing vs. Concorde?
  - Boeing 747: 286,700 pmph / 178,200 pmph = 1.6 times faster
- Boeing is 1.6 times ("60%") faster in terms of throughput
- Concord is 2.2 times ("120%") faster in terms of flying time (response time)

We will focus primarily on execution time for a single job

Confusing Wording on Performance

- Will (try to) stick to "n times faster"; its less confusing than "m % faster"
- As faster means both increased performance and decreased execution time, to reduce confusion we will (and you should) use "improve performance" or "improve execution time"

What is Time?

- Straightforward definition of time:
  - Total time to complete a task, including disk accesses, memory accesses, I/O activities, operating system overhead, ...
  - "real time", "response time" or "elapsed time"
- Alternative: just time processor (CPU) is working only on your program (since multiple processes running at same time)
  - "CPU execution time" or "CPU time"
  - Often divided into system CPU time (in OS) and user CPU time (in user program)

How to Measure Time?

- User Time $\Rightarrow$ seconds
- CPU Time: Computers constructed using a clock that runs at a constant rate and determines when events take place in the hardware
  - These discrete time intervals called clock cycles (or informally clocks or cycles)
  - Length of clock period: clock cycle time (e.g., 2 nanoseconds or 2 ns) and clock rate (e.g., 500 megahertz, or 500 MHz), which is the inverse of the clock period; use these!

"And in conclusion…"

- RAID
  - Motivation: In the 1980s, there were 2 classes of drives: expensive, big for enterprises and small for PCs. They thought "make one big out of many small!"
  - Higher performance with more disk arms per $
  - Adds option for small # of extra disks (the "R")
  - Started @ Cal by CS Profs Katz & Patterson
- Latency v. Throughput
- Measure time as User time vs CPU time