Today

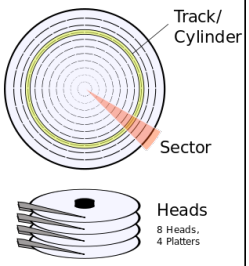
- Storage
  - Disk Management
  - RAID

Review

- How should we schedule reads/writes to the disk?

Disk Management: Formatting

- Low-level formatting or physical formatting
- Start with a blank disk
- Divide disk into sectors that the disk controller can read and write
- Sector
  - Header, Trailer
    - Sector number
    - error-correcting code
  - Data area, usually 512 bytes



Disk Management: Formatting

OS needs to record data structures on disk

1. Partition disk into one or more groups of cylinders
   - each partition treated as a logical disk
2. Logical formatting or making a file system
   - Maps of free and allocated space
   - Empty directory

To increase efficiency, most file systems group *blocks* into *clusters*

- Disk I/O done in blocks
- File I/O done in clusters – sequential access

Disk Management

- Allow ***raw*** disk access for apps that want to do their own block management
  - Bypass OS
  - For example: databases

Boot block

- Recall: bootstrap initializes system, starts OS
- Bootstrap ***loader*** is stored in ROM
  - Doesn't change
- Bootstrap stored in ***boot blocks*** of boot partition
  - Boot partition: boot disk or system disk

Handling Bad Blocks

- Disks are prone to failure
- Sectors are or become defective
- Basic handling:
  1. Scan disk for bad blocks
  2. File system does not allocate bad blocks
- Improvements
  - Keep list of bad blocks
  - Keep ***spare*** sectors not visible to the OS
  - Replace bad sectors with spares, logically
    - Logical block 87 goes to updated physical location

David Patterson

Garth Gibson

Randy Katz

(thanks to David Patterson for slide material)

RAID

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

	Big, Expensive IBM 3390K	Small, Cheap IBM 3.5" 0061	Small, Cheap 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 have potential for large data and I/O rates, high MB per cu. ft., high MB per KW

But what about reliability?

Array Reliability

- Reliability of N disks = Reliability of 1 Disk÷N
  - 50,000 Hours ÷ 70 disks = 700 hours
  - Disk system MTTF: drops from 6 years→1 month!
- Arrays (without redundancy) are too unreliable to be useful!

Hot spares: unallocated disks, automatically replace a failed disk and have data rebuilt onto them

→ support reconstruction in parallel with access: very high media availability can be achieved

Redundant Arrays of (Inexpensive →Independent) Disks (RAID)

- Basic idea: 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
- Multiple schemes
  - Provide different balance between data reliability and input/output performance

Redundant Arrays of Independent Disks RAID 0: Striping

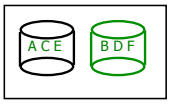
- Stripe data at the block level across multiple disks

A B C D E F

What are the effects of having such an arrangement?

Redundant Arrays of Independent Disks
RAID 0: Striping

- Stripe data at the block level across multiple disks
- High read and write bandwidth
- Not a true **R**AID since no redundancy
- Failure of any one drive will cause the entire array to become unavailable



A B C D E F

Redundant Arrays of Independent Disks
RAID 1: Disk Mirroring/Shadowing

- Each disk is fully duplicated onto its **mirror**

Impact of this arrangement?

Redundant Arrays of Independent Disks
RAID 1: Disk Mirroring/Shadowing

- Each disk is fully duplicated onto its **mirror**
  - Very high availability can be achieved
- Bandwidth sacrifice on write:
  - Logical write = two physical writes
  - Reads may be optimized
- Most expensive solution: 100% capacity overhead

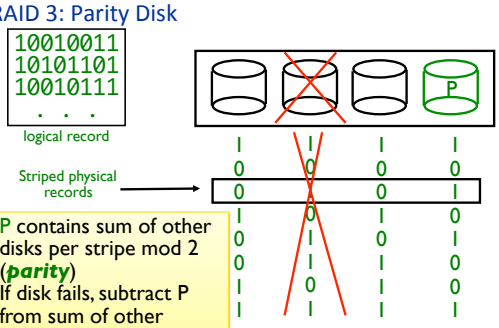
Prefer reliability & performance over low data storage

RAID-I (1989)

- Consisted of a Sun 4/280 workstation with
  - 128 MB of DRAM
  - 4 dual-string SCSI controllers
  - 28 5.25-inch SCSI disks
  - specialized disk striping software

(RAID 2 not interesting, so skip… involves Hamming codes)

Redundant Array of Independent Disks
RAID 3: Parity Disk



- P contains sum of other disks per stripe mod 2 (***parity***)
- If disk fails, subtract P from sum of other disks to find missing information

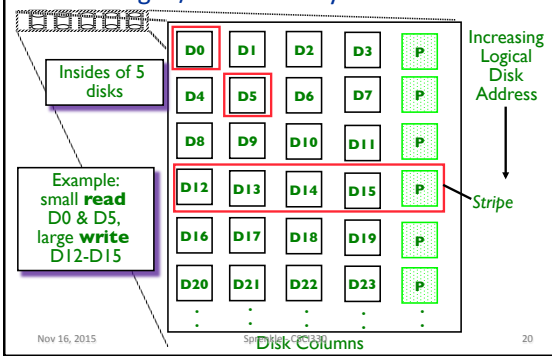
RAID 3

- Sum computed across recovery group to protect against hard disk failures, stored in P disk
- Logically, a single high-capacity, high-transfer-rate disk: good for large transfers
- But
  - byte-level striping is bad for small files
    - all disks involved
  - Parity disk is still a bottleneck

Inspiration for RAID 4

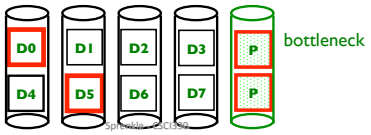
- RAID 3 stripes data at the *byte* level
  - RAID 3 relies on parity disk to discover errors on read
  - But every sector on disk has an error detection field
- Block-level striping
- Rely on error detection field to catch errors on read
  - not on the parity disk
- Allows independent reads to different disks simultaneously
- Increases read I/O rate since only one disk is accessed rather than all disks for a small read

Redundant Arrays of Independent Disks
RAID 4: High I/O Rate Parity

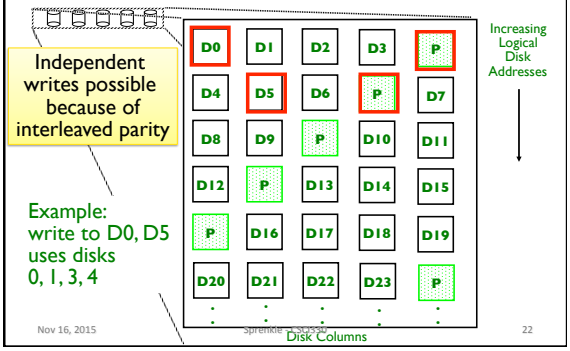


Inspiration for RAID 5

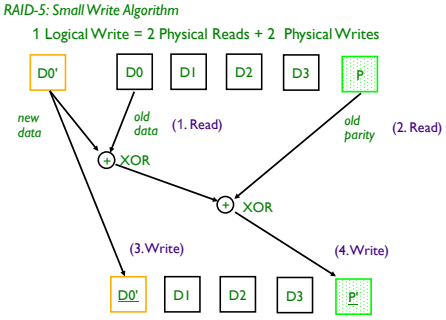
- RAID 4 works well for small reads
- Small writes (write to one disk):
  - Option 1: read other data disks, create new sum and write to Parity Disk
  - Option 2: since P has old sum, compare old data to new data, add the difference to P
- Small writes are still limited by Parity Disk:
  - Write to D0, D5, both also write to P disk



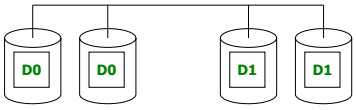
Redundant Arrays of Independent Disks
RAID 5: High I/O Rate Interleaved Parity



Problems of Disk Arrays: Small Writes



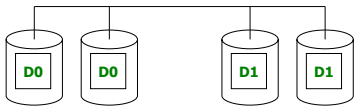
RAID-10 (0+1)



- Striping + mirroring

What's the impact?

RAID-10 (0+1)



- Striping + mirroring
- High storage overhead/cost
- For small write-intensive apps, may be better than RAID-5
  - Write data twice but no reads or XORs required

Other Features

- **Snapshot** is a view of file system before a set of changes take place (i.e. at a point in time)
  - See Ch 11
- Replication is automatic duplication of writes between separate sites
  - For redundancy and disaster recovery
  - Can be synchronous or asynchronous
  - Tangent on TACT

Choosing a RAID Level

How should you choose a RAID level?

What are your considerations?

Choosing a RAID Level

- Tradeoffs in availability, performance, recoverability, resource consumption
- Typically: RAID 5 or 10
  - Consideration: small writes

Looking Ahead

- Project 4: Due Sunday after Thanksgiving
  - BUT, hopefully working on it a bit every day
- Wed: File Systems