Today's Objectives

- AWS/MR Review
- Exam Discussion
- Storage Systems
  - RAID

Project 3

- AWS Account Update?
  - Can get a non-student account but requires credit card
- Thursday
  - Set of documents
- Questions?

EXAM

Exam (not a midterm) – 20%

- Paragraphs/essays
- Sakai
  - Write answers in Word and then copy over to Sakai
- Two hours (out of class)
  - Open notes BUT that should just be a backup
- Plan: November 15-17

STORAGE SYSTEMS

Storage Systems

- Goals of storage systems:
  - Provide high *availability*
  - Provide high *reliability*
  - Provide high *performance* (fast reads and writes)
  - Provide high *capacity*

- Before thinking about a networked distributed system, let's ignore network problems.

How can we achieve these goals using multiple disks in a single computer?

(thanks to David Patterson for slide material)

RAID

Idea: 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 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) too unreliable to be useful!

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

Nov 1, 2017 Sprenkle - CSCI325 9

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

- Basic idea: files are "striped" across multiple disks
  - Can do reads in parallel on the multiple disks
- Redundancy yields high data availability
  - **Availability**: service still provided to user, even if some components failed

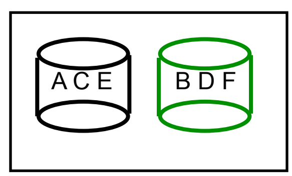
Nov 1, 2017 Sprenkle - CSCI325 10

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

- 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

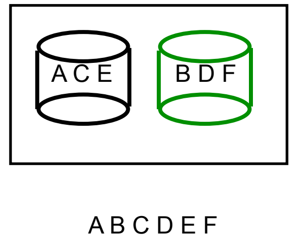


A B C D E F

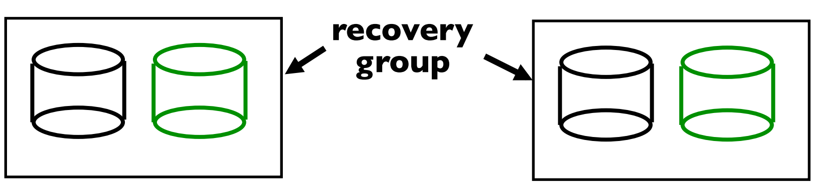
What are the outcomes?
- Expected behavior?
- Failure?

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



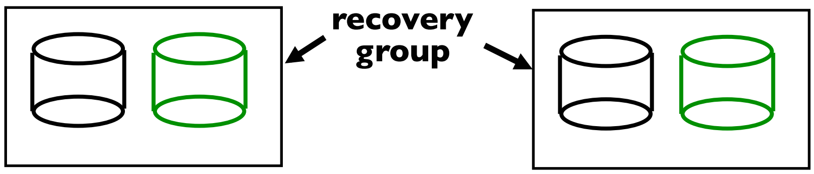
Redundant Arrays of Independent Disks
RAID 1: Disk Mirroring/Shadowing



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

What are the outcomes?
- Expected behavior?
- Failure?

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)

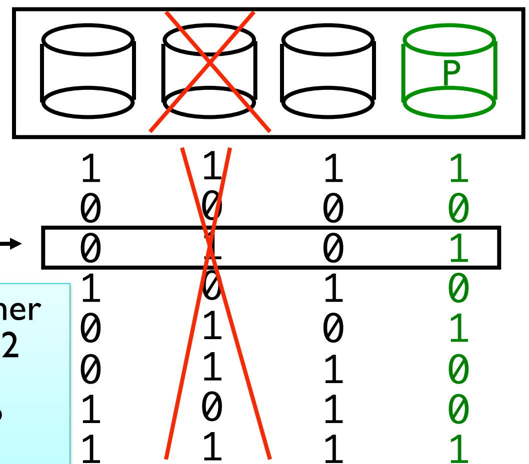
- Consisted of a Sun 4/280 workstation with
  - 128 MB of DRAM
  - 4 dual-string SCSI controllers
  - 28 5.25-inch SCSI disks
  - specialized disk striping software

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

Redundant Array of Independent Disks
RAID 3: Parity Disk

```
10010011
10101101
10010111
  . . .
```
logical record

Striped physical records



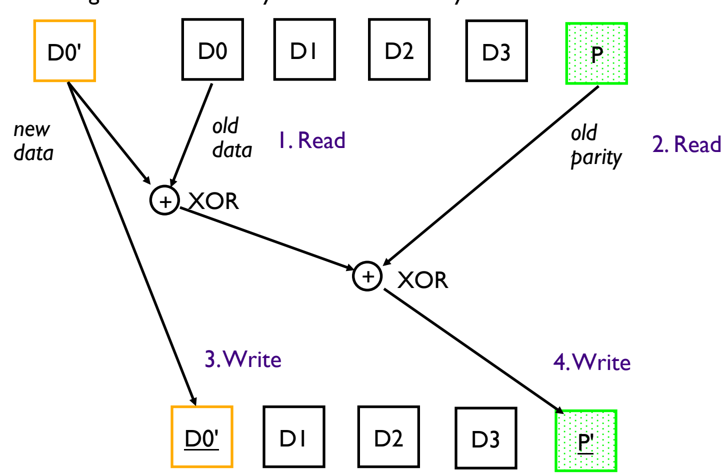
- P contains sum of other disks per stripe mod 2 (***parity***)
- If disk fails, subtract P from sum of other disks to find missing information

Problems of Disk Arrays: Small Writes

Update to bytes
(just changing the D's)

RAID-5: Small Write Algorithm

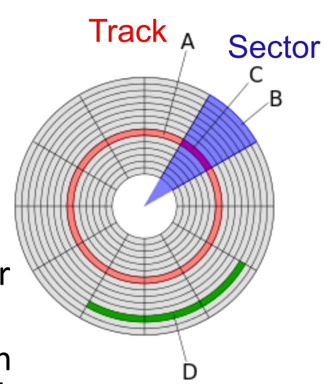
1 Logical Write = 2 Physical Reads + 2 Physical Writes



RAID 3

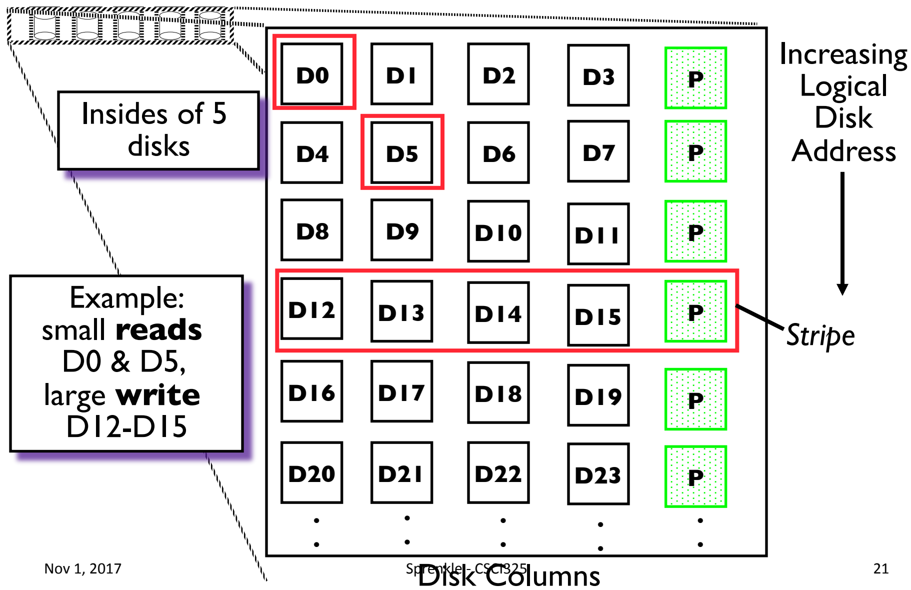
- Sum computed across recovery group to protect against hard disk failures, stored in P disk
- Logically, a single high-capacity, high-transfer-rate disk: good for large transfers
- But byte-level striping is bad for small files (all disks involved)
- Parity disk is still a bottleneck

Inspiration for RAID 4



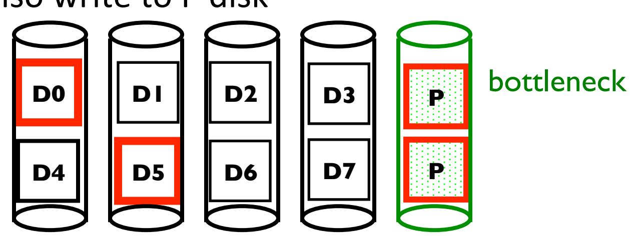
- RAID 3 stripes data at the *byte* level
- RAID 3 relies on parity disk to discover errors on read
- But every sector on disk has an error detection field
- Rely on error detection field to catch errors on read, not on the parity disk
- Allows independent reads to different disks simultaneously
- Increases 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



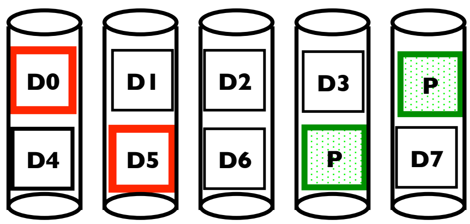
Inspiration for RAID 5

- 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



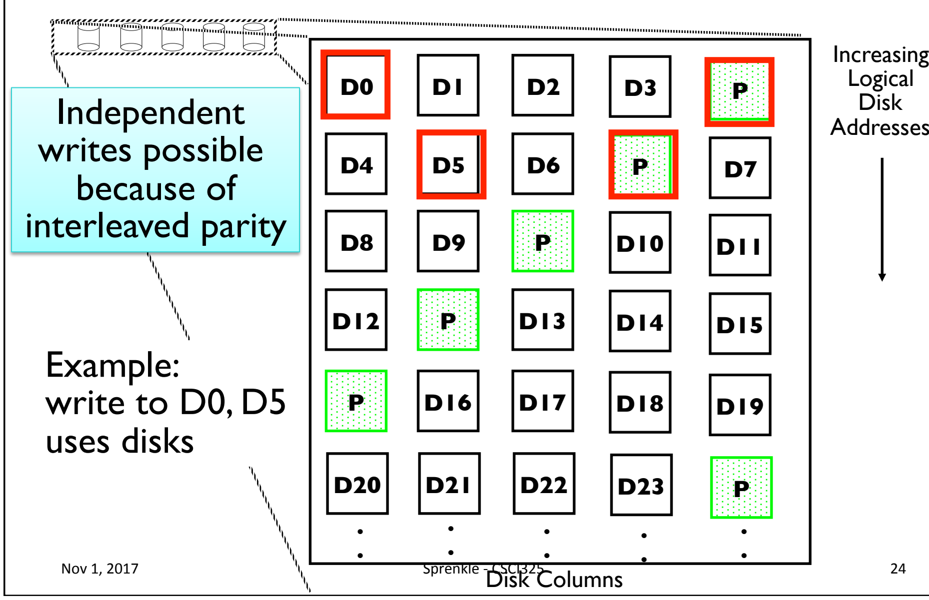
Inspiration for RAID 5

- 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



Result: same disk isn't a bottleneck for *all writes*

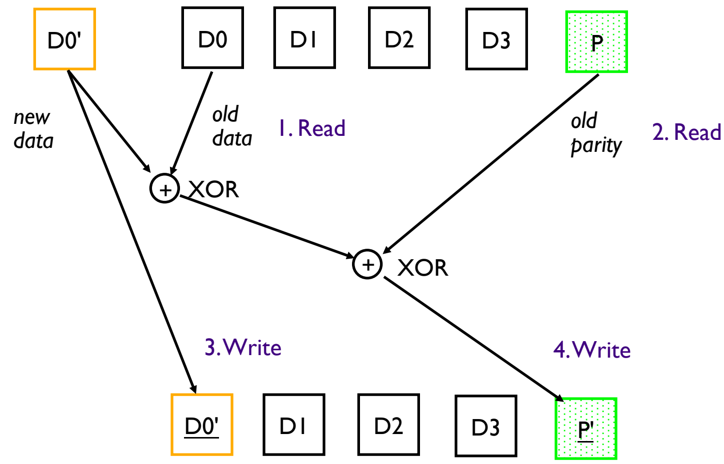
Redundant Arrays of Independent Disks RAID 5: High I/O Rate Interleaved Parity



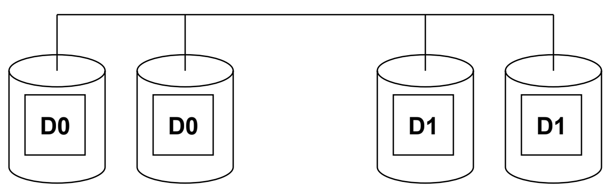
Problems of Disk Arrays: Small Writes

RAID-5: Small Write Algorithm

1 Logical Write = 2 Physical Reads + 2 Physical Writes



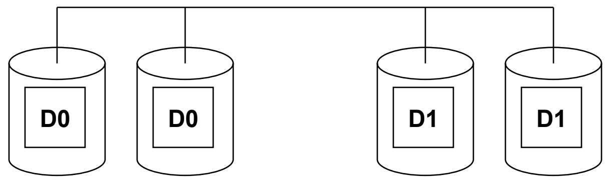
RAID-10 (0+1)



- Striping + mirroring
- High storage overhead/cost

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

Weaknesses

- Disks tend to be the same age
  - Similar failure times
- Disk capacity has increased
  - Transfer speed hasn't
  - Error rates haven't decreased

But what about the network?

- How does the network complicate things?
- What can we do about it?

- What new challenges are introduced by a distributed file system in addition to scalable storage?
  - FRIDAY!

Nov 1, 2017 Sprenkle - CSCI325 29

Looking Ahead

- AWS Project
- Networked File Systems

Nov 1, 2017 Sprenkle - CSCI325 30