CSE 451: Operating Systems
Spring 2010

Module 18
Redundant Arrays of Inexpensive Disks (RAID)

John Zahorjan
zahorjan@cs.washington.edu
Allen Center 534

Managing Physical Resources

- A single disk can be broken up into smaller pieces (partitions):
  - Limit the damage of failures
  - Unit of backup/migration
  - Run different kinds of file systems

- What about making use of multiple disks?
  - None of the file systems we've seen can span multiple disks
    - Why?
  - What about mounting multiple devices?
    - More storage?
    - Better performance?

Performance

1. Disk transfer rates are improving, but much less fast than CPU performance

2. We can use multiple disks to improve performance
   - by *striping* files across multiple disks (placing parts of each file on a different disk), we can use parallel I/O to improve access time

1. Striping reduces reliability
   - 10 disks have about 1/10th the MTBF (mean time between failures) of one disk

Reliability

- It's typically enough to be resilient to a single disk failure
  - In theory, the odds that another disk fails while you're replacing the first one are low
    - The first time CSE ran a RAID it happened to us…
- To improve reliability, add redundant data to the disks
  - We'll see how in a moment
- So:
  - Performance from striping
  - Resilience from redundancy, which steals back some of the performance gain

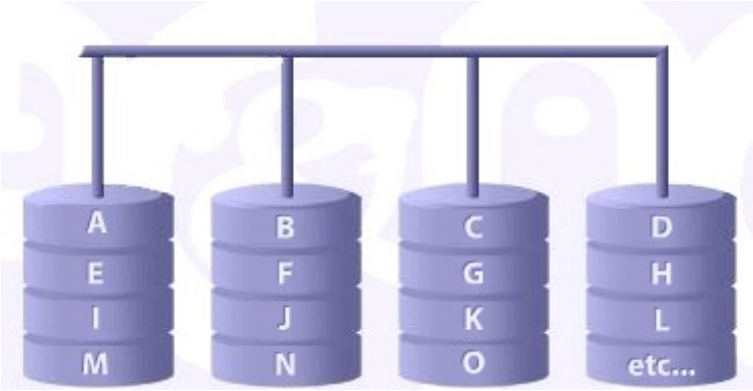
RAID

- A RAID is a Redundant Array of Inexpensive Disks
- Disks are small and cheap, so it's easy to put lots of disks (10s to 100s) in one box for increased storage, performance, and availability
- Data plus some redundant information is striped across the disks in some way
- How striping is done is key to performance and reliability

Some RAID tradeoffs

- Granularity
  - fine-grained: stripe each file over all disks
    - high throughput for the file
    - limits transfer to 1 file at a time
  - course-grained: stripe each file over only a few disks
    - limits throughput for 1 file
    - allows concurrent access to multiple files
- Redundancy
  - uniformly distribute redundancy information on disks
    - avoids load-balancing problems
  - concentrate redundancy information on a small number of disks
    - partition the disks into data disks and redundancy disks

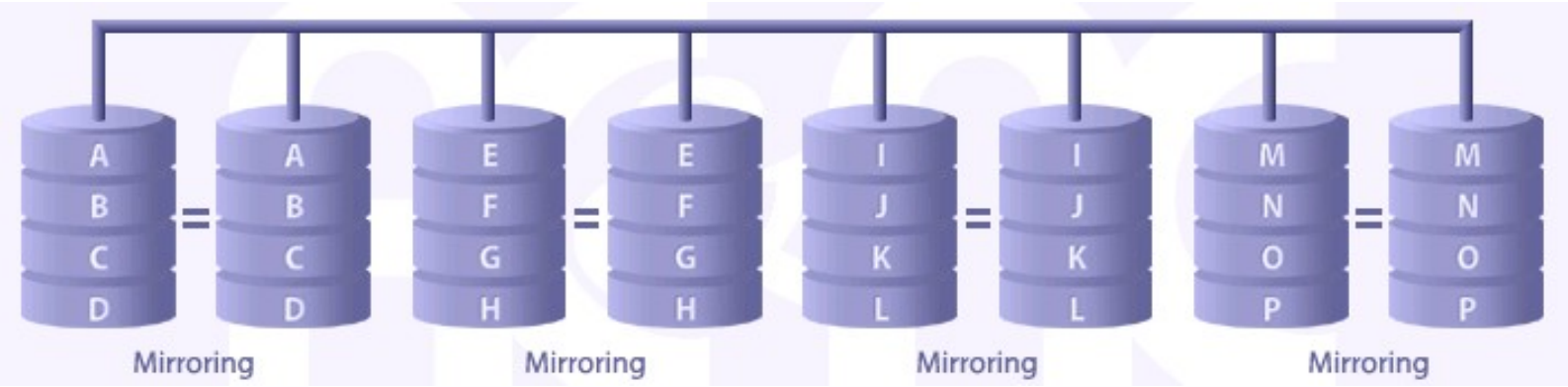
RAID Level 0: Non-Redundant Striping



- RAID Level 0 is a non-redundant disk array
- Files are striped across disks, no redundant info
- High (single file) read throughput
- Best write throughput (no redundant info to write)

- Any disk failure results in data loss
  - What is lost?

RAID Level 1: Mirrored Disks

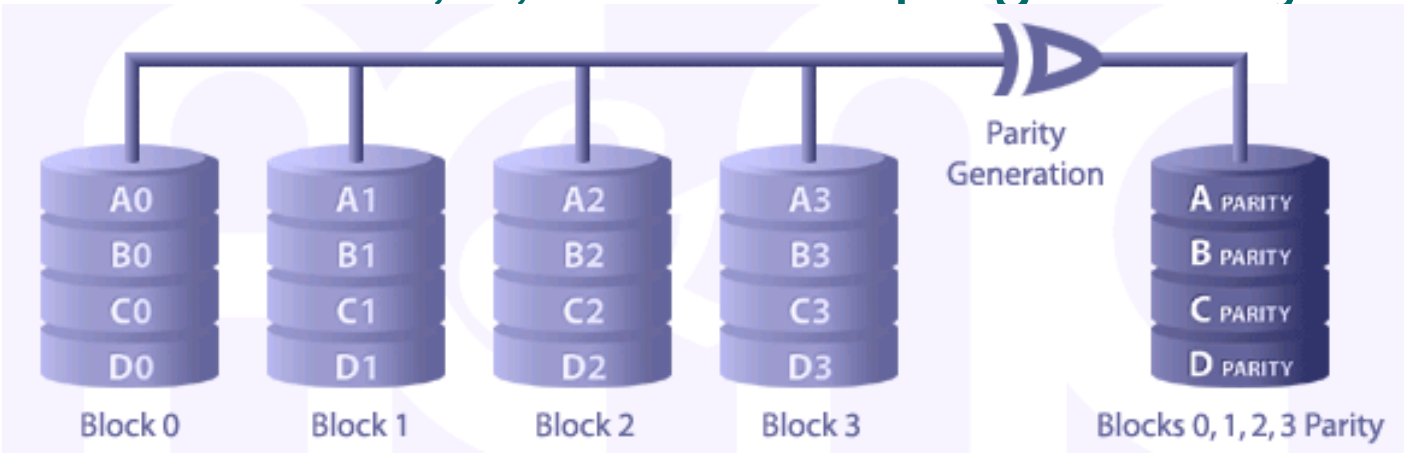


- Files are striped across half the disks, and mirrored to the other half
  - 2x space expansion
- Reads:
  - Read from either copy
- Writes:
  - Write both copies
- On failure, just use the surviving disk

What is the effect on performance?

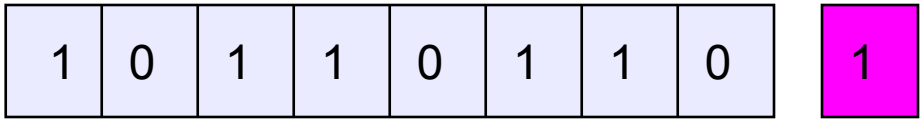
How many simultaneous disk failures can be tolerated?

RAID Levels 2, 3, and 4: Striping + Parity Disk



- RAID levels 2, 3, and 4 use ECC (error correcting code) or parity disks
  - E.g., each byte on the parity disk is a parity function of the corresponding bytes on all the other disks
- A large read accesses all the data disks
  - A single block read accesses only one disk (RAID 4)
- A write updates one or more data disks plus the parity disk
- Resilient to single disk failures (How?)
- Better ECC ⇒ higher failure resilience ⇒ more parity disks

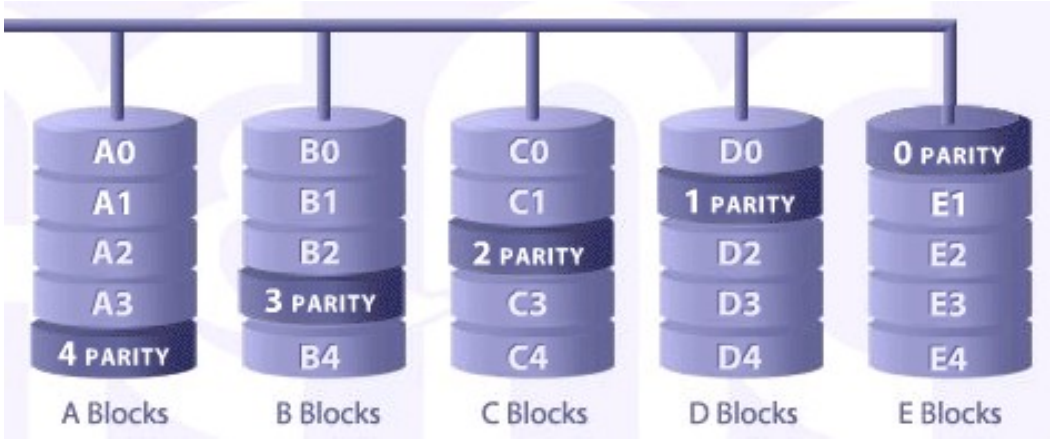
Refresher: What's parity?



- To each byte, add a bit set so that the total number of 1's is even
- Any single missing bit can be reconstructed
- (Why does memory parity not work quite this way?)

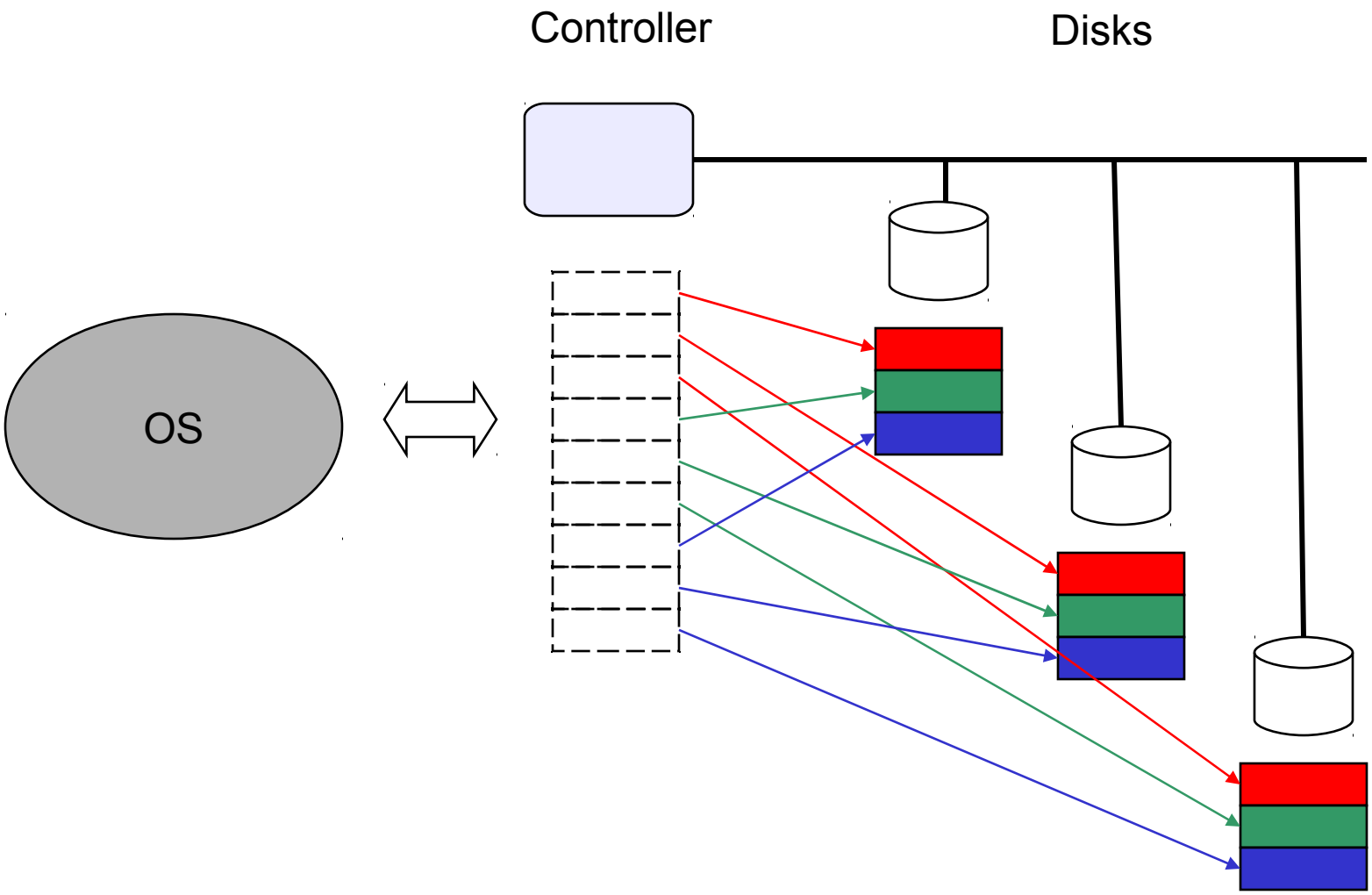
- Think of ECC as just being similar but fancier (more capable)

RAID Level 5

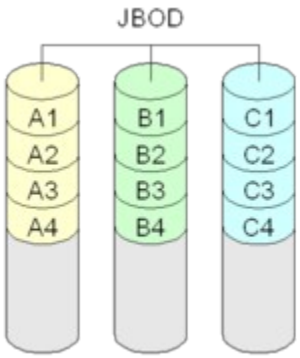


- RAID Level 5 uses block interleaved distributed parity
- Like parity scheme, but distribute the parity info (as well as data) over all disks
  - for each block, one disk holds the parity, and the other disks hold the data
- Significantly better performance
  - parity disk is not a hot spot

Typical Implementation



JBOD (Just a Bunch Of Disks)



- Not really a RAID – no striping, no redundancy
- Blocks from disks are concatenated to form a single, logical device
- Allows use of disks of differing sizes

Final Issues

- If you're running a RAID level with sufficient redundancy, do you need backup?
  - What's the difference between RAID and backup?
- Does RAID on its own provide "sufficient" reliability?
  - If you're Amazon?

Tier I
Single path for power and cooling distribution, no redundant components, 99.671% availability.

Tier II
Single path for power and cooling distribution, redundant components, 99.741% availability.

Tier III
Multiple power and cooling distribution paths, but only one path active, redundant components, concurrently maintainable, 99.982% availability.

Tier IV
Multiple active power and cooling distribution paths, redundant components, fault tolerant, 99.995% availability.