CSE 451: Operating Systems
Spring 2009

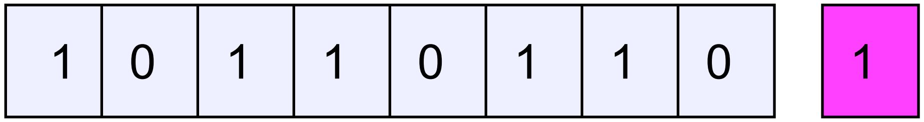
Redundant Arrays of Inexpensive Disks (RAID)

Steve Gribble

The challenge

- Disk transfer rates are improving
  - but much less fast than CPU performance
- We can use multiple disks to improve performance
  - *stripe* files across multiple disks
    - place parts of each file on a different disk
    - with this, we can use parallel I/O to improve performance
- But, striping reduces reliability!
  - 100 disks have 1/100th the MTBF (mean time between failures) of one disk
- So, we need striping for performance, but we need something to help with reliability / availability
  - to improve reliability, we can add redundant data to the disks, in addition to striping

Refresher: What’s parity?



- To each byte, add a bit whose value is set so that the total number of 1’s is even
- Any single missing bit can be reconstructed
  - *more complex schemes (e.g., based on Hamming codes) can detect multiple bit errors and correct single bit errors. Called ECC (error correcting code) memory.*

RAID

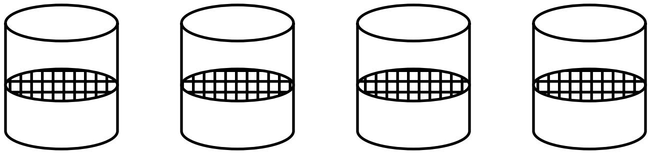
- A RAID is a Redundant Array of Inexpensive Disks
- Disks are small and cheap, so it’s easy to put lots of disks (10s to 100s) in one box for increased storage, performance, and availability
- Data plus some redundant information is striped across the disks in some way
- How striping is done is key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

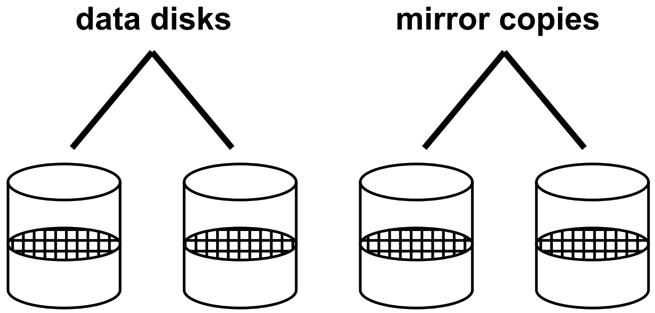
RAID Level 0

- RAID Level 0 is a non-redundant disk array
- Files are striped across disks, no redundant info
- High read throughput
- Best write throughput (no redundant info to write)
- Any disk failure results in data loss



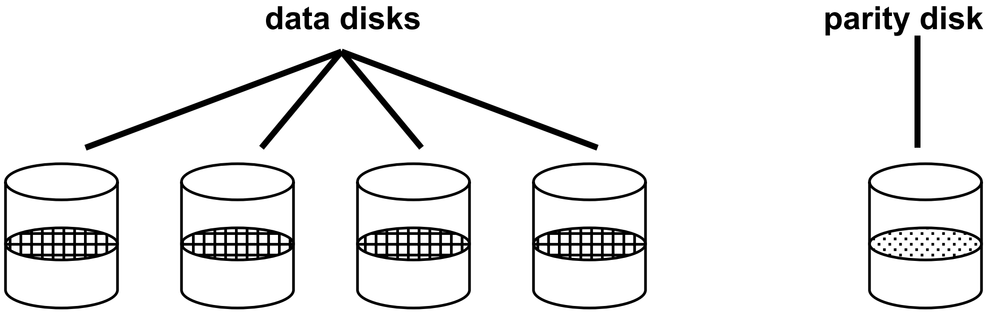
RAID Level 1

- RAID Level 1: mirrored disks
- Files are striped across half the disks
- Data is written to two places
  - a data disk and a mirror disk
- On failure, just use the surviving disk
- 2x space expansion



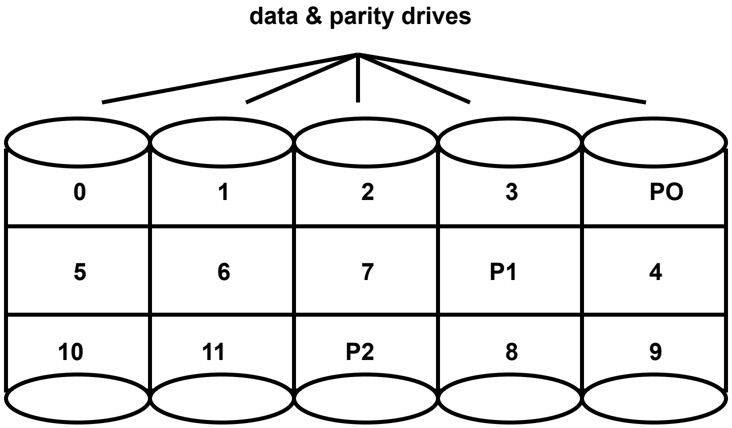
RAID Levels 2, 3, and 4

- RAID levels 2, 3, and 4 use ECC or parity disks
  - e.g., each byte on the parity disk is a parity function of the corresponding bytes on all the other disks
  - details between the different levels have to do with kind of ECC used, and whether it is bit-level or block-level
- A read accesses all the data disks, a write accesses all the data disks plus the parity disk
- On disk failure, read the remaining disks plus the parity disk to compute the missing data



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



data & parity drives

0	1	2	3	PO
5	6	7	P1	4
10	11	P2	8	9

File Block Numbers

RAID Level 6

- Basically like RAID 5 but with replicated parity blocks so that it can survive two disk failures.
- Useful for larger disk arrays where multiple failures are more likely.

Example RAID Storage

Promise 3U rack-mountable 16-disk RAID Storage System

Hot swappable drives

Dual controllers with 4 host interface ports for reliability

Can be ganged together into larger units