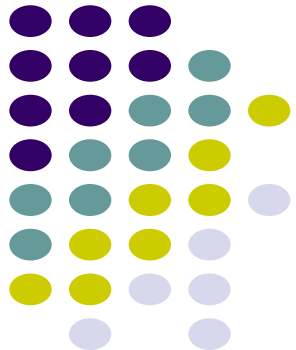
RAID

ECE 469, April 12

Aravind Machiry



RAID

- Two motivations
  - (in the past) Operating in parallel can increase disk throughput
    - RAID = Redundant Array of Inexpensive Disks
  - (today) Redundancy can increase reliability
    - RAID = Redundant Array of Independent Disks

RAID

- Parallel reading (striping) (for performance)
  - Splitting bits of a byte across multiple disks
  - 8 disks (bit-level striping)
    - Logically acts like single disk with sector size * 8 and access time / 8
    - Reduce the response time of large access (e.g. one 4K block)
  - Alternatively, block-level striping
    - Increases the throughput of multiple small accesses (e.g. eight 512-byte blocks)

RAID

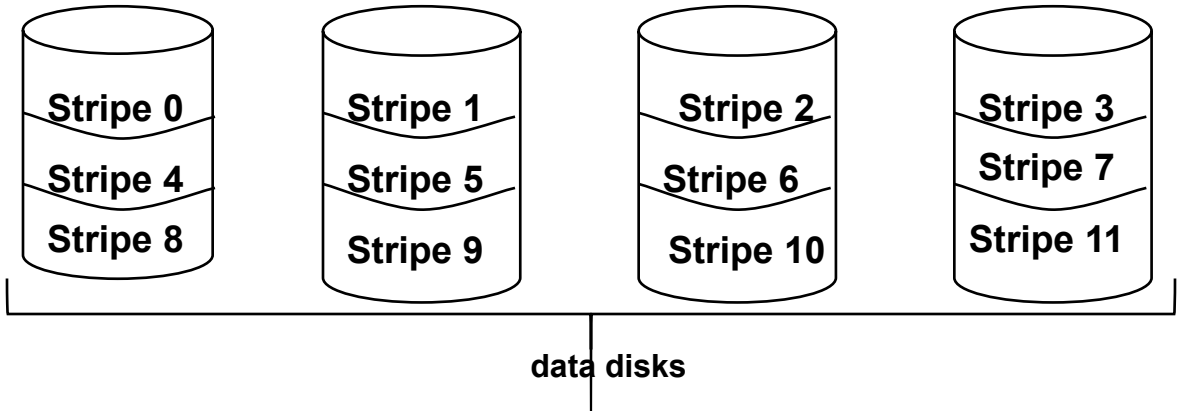
- Mirroring or shadowing (for reliability)
  - Local disk consists of 2 physical disks in parallel
  - Every write performed on both disks
    - Can read from either disk
    - Probability of both fail at the same time?

RAID - Combine the two ideas

- Mirroring gives reliability, but expensive
- Striping gives high data-transfer rate, but not reliability
- Challenge: can we provide redundancy at low cost?

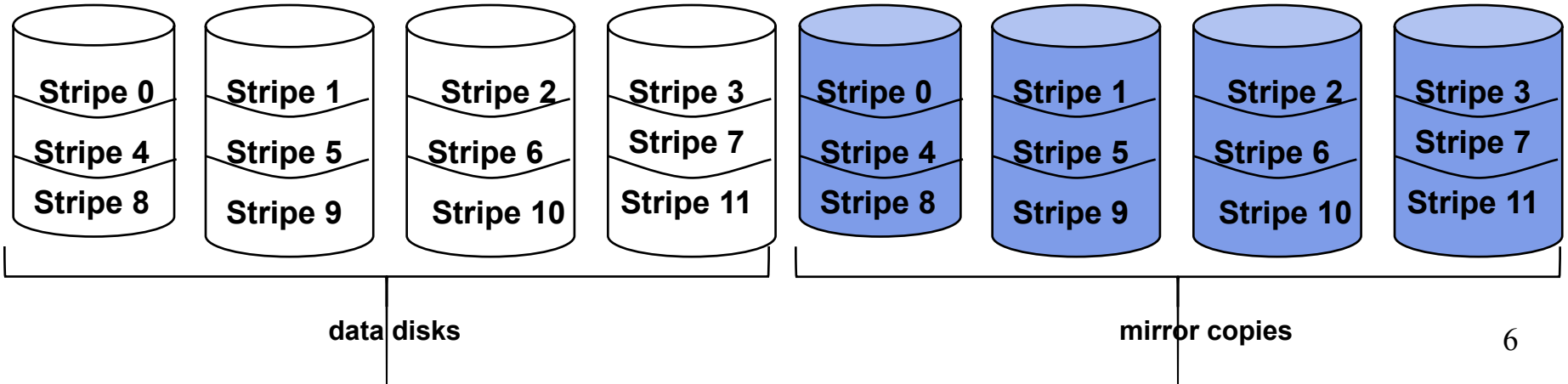
RAID Level 0

- Level 0 is non-redundant disk array
- Files are Striped across disks, no redundant info
- High read throughput
- Best write throughput among RAID levels (no redundant info to write)
- Any disk failure results in data loss
  - What's the MTTF (mean time to failure) of the whole system?



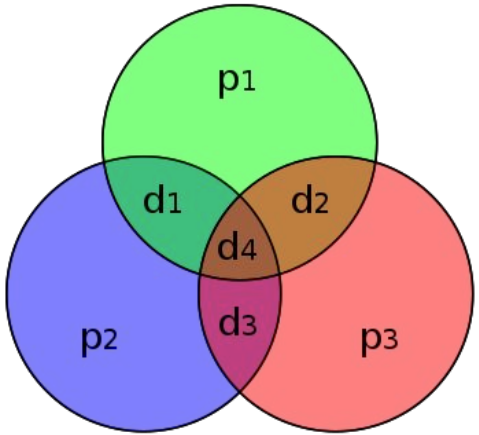
RAID Level 1

- Mirrored Disks
- Data is written to two places
  - On failure, just use surviving disk
- On read, choose fastest to read
  - Write performance is same as single drive, read performance is 2x better
- Expensive



Error Correcting Code : Hamming Code

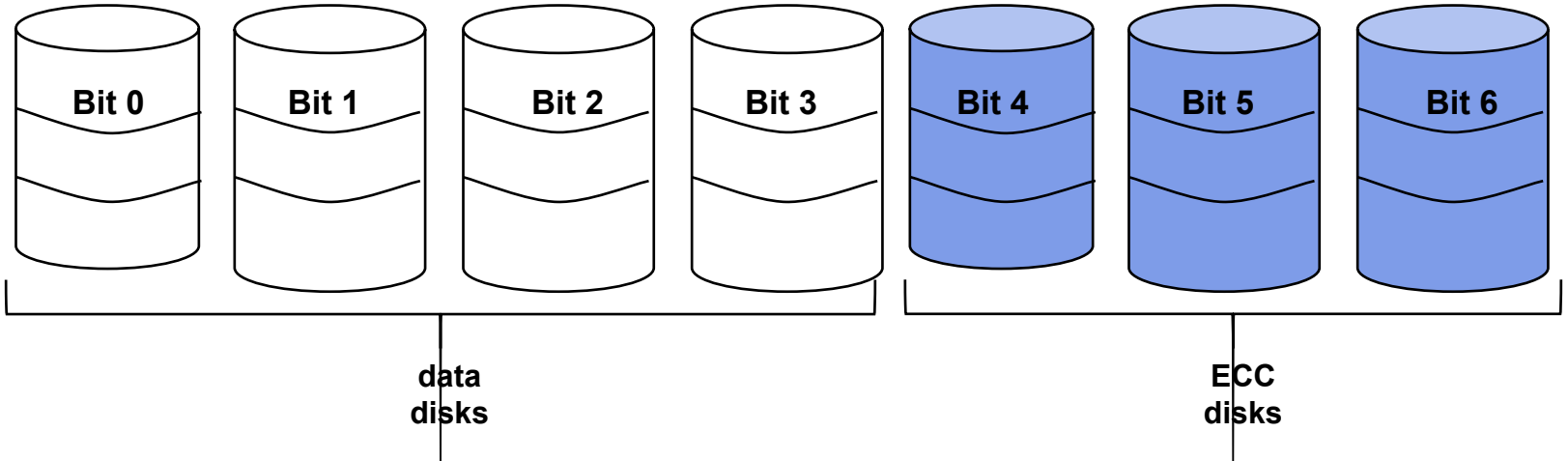
- Can we recover from errors without creating complete back up?
- Error Correcting Code:
  - E.g., Hamming Code
  - Can detect multiple bit-errors and can fix single bit error.



- p => Parity Bits
- d => Data bits

RAID Level 2

- Bit-level Striping with Hamming (ECC) codes for error correction
- All 7 disk arms are synchronized and move in unison
- Complicated controller
- Single access at a time
- Tolerates only one error, but with no performance degradation
- Not used in real world

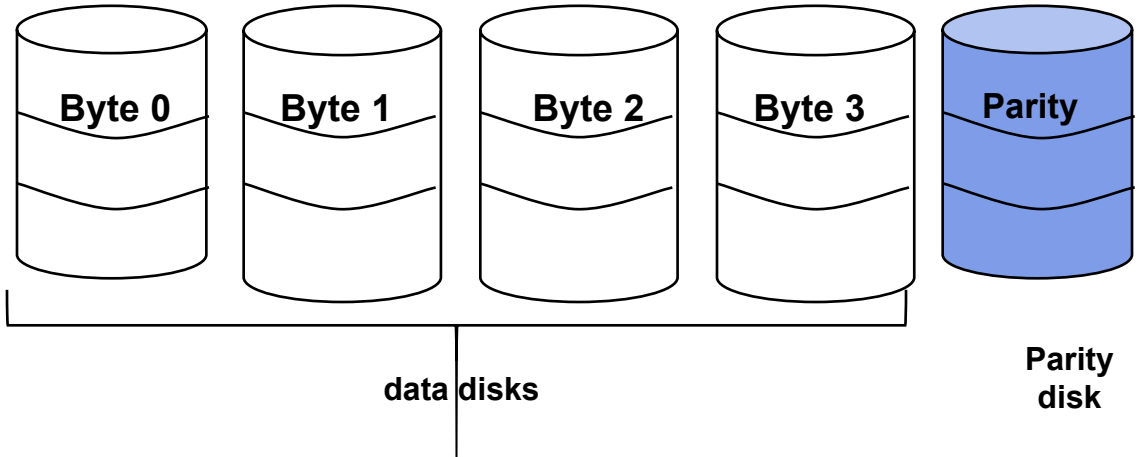


Parity Bits

- What do you need to do in order to detect and correct a one-bit error ?
  - Suppose you have a binary number, represented as a collection of bits: <b3, b2, b1, b0>, e.g. 0110
- Detection is easy
- Parity:
  - Count the number of bits that are on, see if it's odd or even
    - EVEN parity is 0 if the number of 1 bits is even
  - Parity(<b3, b2, b1, b0 >) = P0 = b0 ⊗ b1 ⊗ b2 ⊗ b3
  - Parity(<b3, b2, b1, b0, p0>) = 0 if all bits are intact
  - Parity(0110) = 0, Parity(01100) = 0
  - Parity(11100) = 1 => ERROR!
  - Parity can detect a single bit error, but can't tell you which of the bits got flipped

RAID Level 3

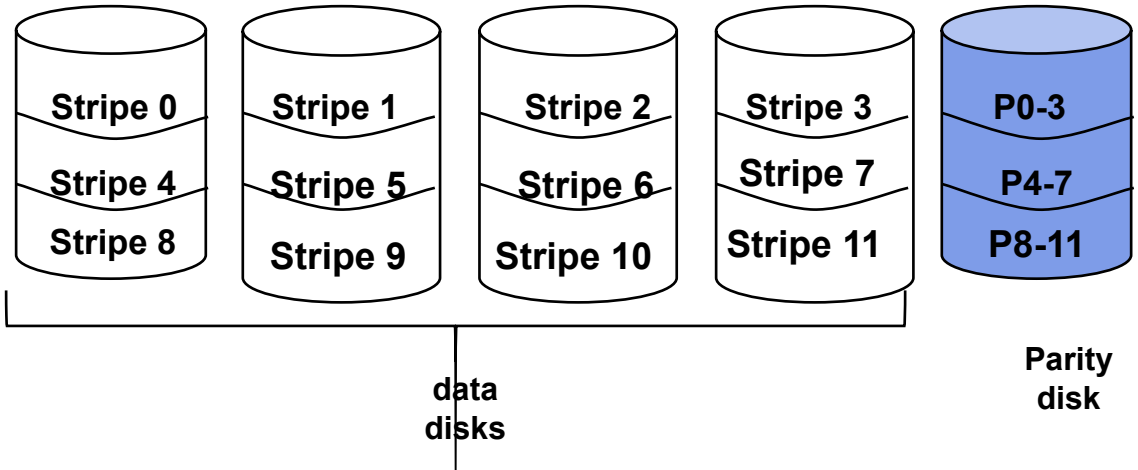
- Use a parity disk
  - Each byte on the parity disk is a parity function of the corresponding bytes on all the other disks
- A read accesses all the data disks
- A write accesses all data disks plus the parity disk
- **On disk failure, read remaining disks plus parity disk to compute the missing data**



Single parity disk can be used to detect and recover errors

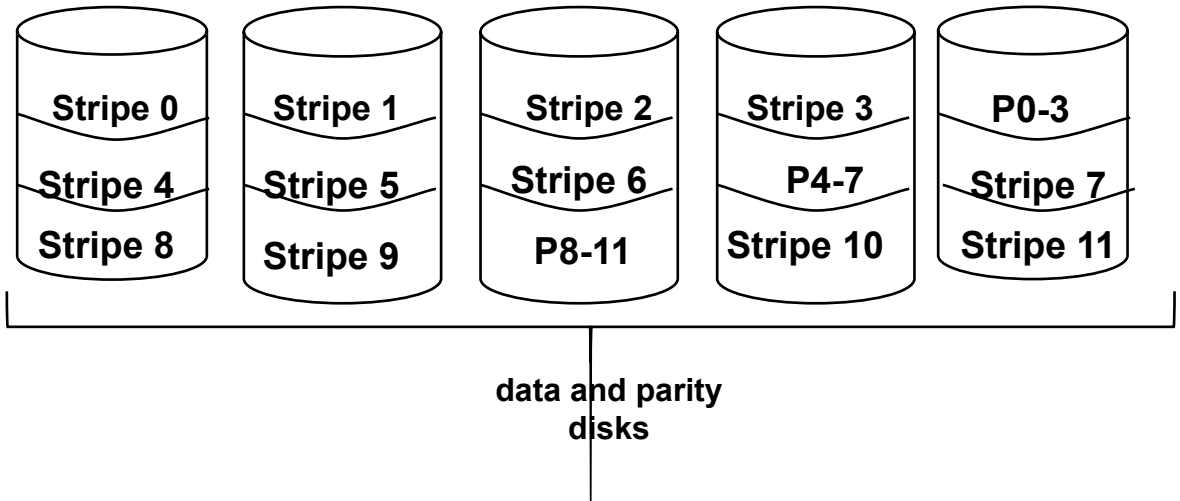
RAID Level 4

- Combines Level 0 and 3 – block-level parity with Stripes
- Lower transfer rate for each block (by single disk)
- Higher overall rate (many small files, or a large file)
- Large writes □ parity bits can be written in parallel
- Small writes □ 2 reads + 2 writes !
- Heavy load on the parity disk



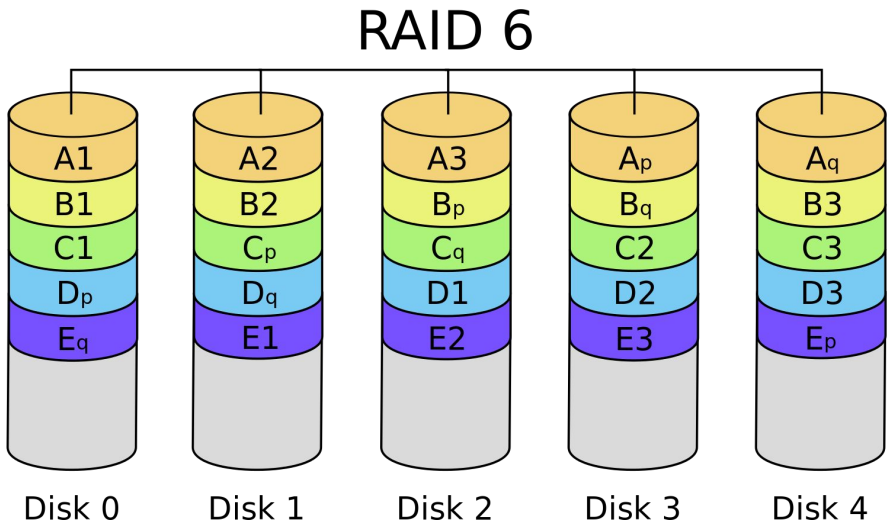
RAID Level 5

- Block Interleaved Distributed Parity
- Like parity scheme, but distribute the parity info over all disks (as well as data over all disks)
- Better (large) write performance
  - **No single disk as performance bottleneck**



RAID Level 6

- Level 5 with an extra parity bit
- **Can tolerate two failures**
  - What are the odds of having two concurrent failures ?
- No performance penalty on reads, slower on writes (compared to RAID5)

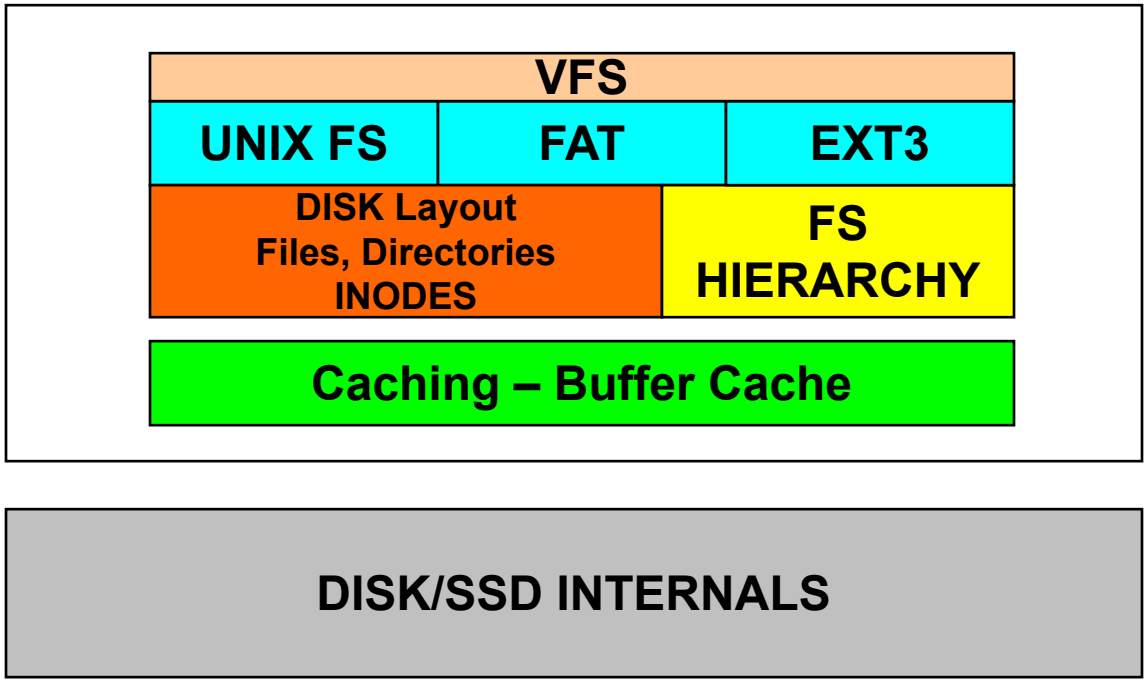


RAID Implementation

- Typically in hardware
  - Special-purpose RAID controller (PCI card)
  - Manages disks
  - Performs parity calculation

- Can be in software (by OS)
  - Can be fast
  - At the cost of CPU time

FS Topics Covered Till now!



FS Reliability

Crash Recovery

Journaling

RAID