Disk Arrays

Dave Eckhardt
de0u@andrew.cmu.edu

Synchronization

- Who read Effective Java over break?
- Survey questions
  - CVS, PRCS?
  - Hard disk crash?
  - Lecture frequency reduction? Day?
- Project 3
  - You've read the handout, right?

Synchronization

- Today: Disk Arrays
  - Text: 14.5 (far from exhaustive)
    - Please read remainder of chapter
  - www.acnc.com 's “RAID.edu” pages
  - www.uni-mainz.de/~neuffer/scsi/what_is_raid.html
  - Papers (@ end)

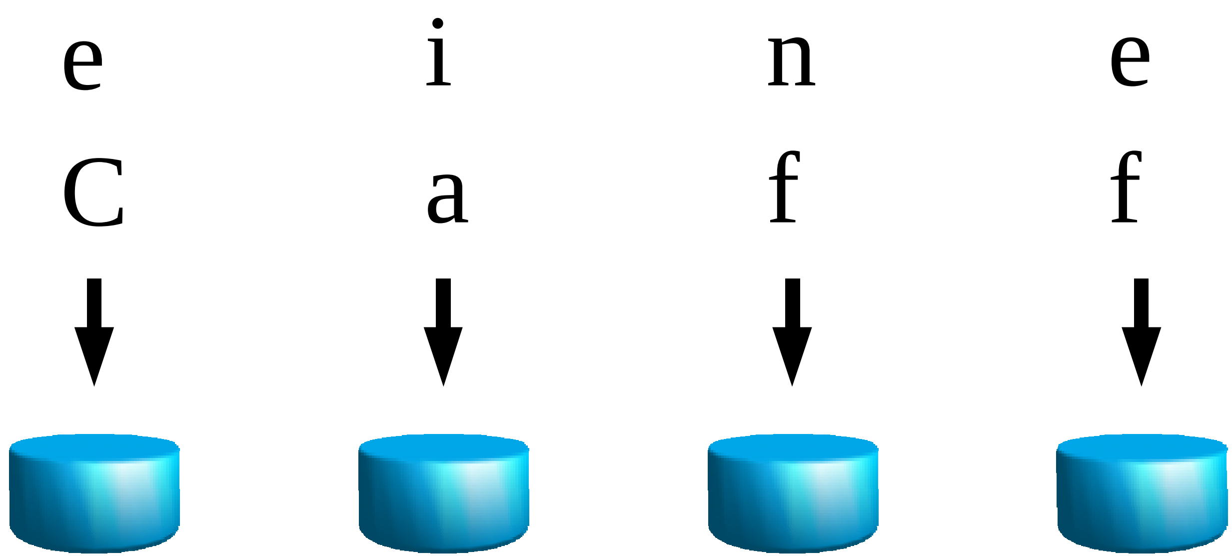
Overview

- Historical practices
  - Striping, mirroring
- The reliability problem
- Parity, ECC, why parity is enough
- RAID “levels” (really: flavors)
- Applications
- Papers

Striping

- Goal
  - High-performance I/O for databases, supercomputers
- Issues
  - Can't spin a disk infinitely fast
  - 100-platter disks would be a niche market
- Solution: parallelism
  - Gang multiple disks together

Striping



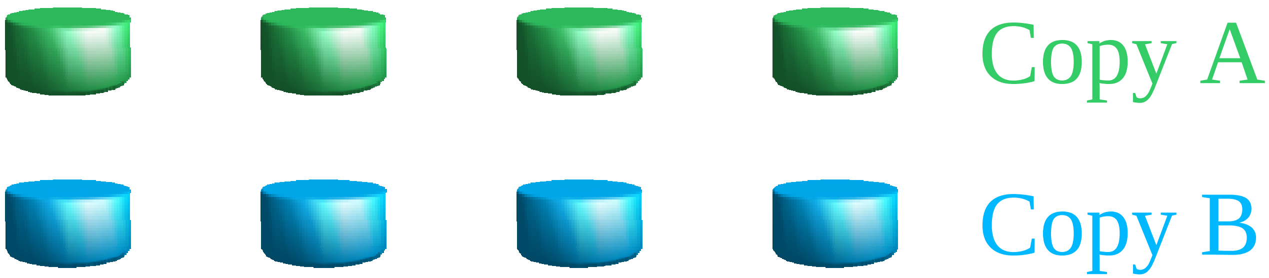
Striping

- Stripe ***size*** can vary
  - Byte
  - Bit
  - Sector
- Results
  - Latency (time to get first byte): unchanged
  - Throughput (bytes per second): linear increase

The reliability problem

- MTTF = Mean time to failure
- MTTF(array) = MTTF(disk) / #disks
- Example from original 1988 RAID paper
  - Connors CP3100 (100 megabytes!)
  - MTTF = 30,000 hours = 3.4 years
  - Array of 100 CP3100's: MTTF = 300 hours = ***12.5 days***

Mirroring



Mirroring

- Operation
  - Write: write to ***both*** mirrors
  - Read: read from ***either*** mirror
- Cost per byte ***doubles***
- Performance
  - Writes: a little slower
  - Reads: maybe 2X faster
- Reliability ***vastly*** increased

Mirroring

- When a disk breaks
  - Identify it to system administrator
    - Beep, blink a light
  - System administrator provides blank disk
  - Copy contents from surviving mirror

Parity

- Parity = XOR "sum" of bits
  - $0 \oplus 1 \oplus 1 = 0$
- Parity provides ***single error detection***
  - Sender provides ***code word*** and ***parity bit***
  - Correct: 011,0
  - Incorrect: 011,1
    - Something is wrong with this picture – ***but what?***
- ***Cannot*** detect multiple-bit errors

ECC

- ECC = error correcting code
- “Super parity”
  - Code word, ***multiple*** “parity” bits
  - Mysterious math computes parity from data
    - Hamming code, Reed-Solomon code
  - Can detect N *multiple-bit* errors
  - Can ***correct*** M < N bit errors!
- Arazi, Commonsense Approach to the Theory of Error Correcting Codes

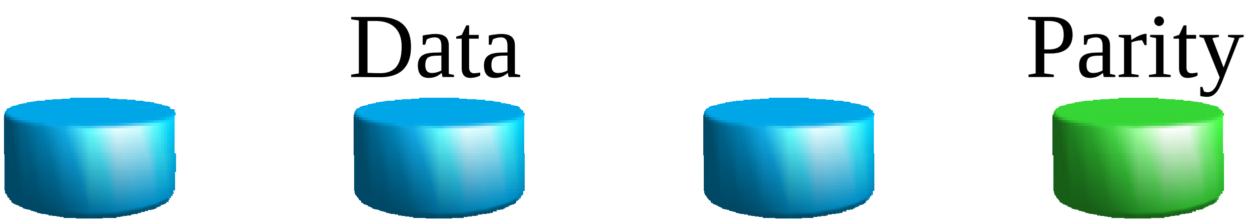
Parity revisited

- Parity provides single ***erasure*** correction!
- Erasure channel
  - Knows when it doesn't know something
  - Each bit is 0 or 1 or “don't know”
- Sender provides code word, parity bit: (0 1 1 , 0)
- Channel provides corrupted message: (0 ? 1 , 0)
- $? = 0 \oplus 1 \oplus 0 = 1$

Erasure channel???

- Are erasure channels real?
- Radio
  - signal strength during reception of bit
- Disk drives!
  - Each sector is stored with CRC
    - Read sector 42 from 4 disks
    - Receive 0..4 good sectors, 4..0 errors
  - “Drive not ready” = “erasure” of all sectors

“Fractional mirroring”



“Fractional mirroring”

- Operation
  - Read: read data disks
    - Error? Read parity disk, compute lost value
  - Write: write data disks ***and parity disk***
- Cost
  - ***Fractional*** increase (50%, 33%, ...)
  - Better than mirroring: 100%

“Fractional mirroring”

- Performance
  - Writes: slower (see below)
  - Reads: unaffected
- Reliability ***vastly*** increased
  - Not as good as mirroring
    - Why not?

RAID “levels”

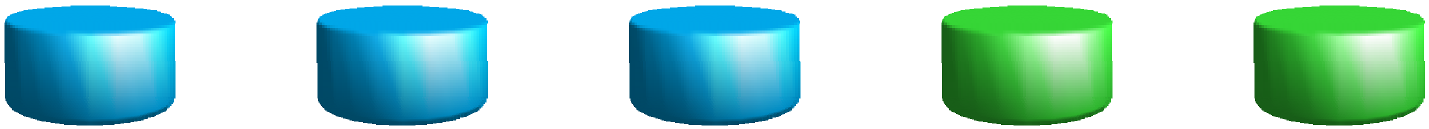
- They're not really levels
  - RAID 2 isn't “more advanced than” RAID 1
    - People really do RAID 1
    - People basically never do RAID 2
- People invent new ones randomly
  - RAID 0+1 ???
  - JBOD ???

Easy cases

- JBOD = “just a bunch of disks”
  - What you get if you lobotomize your RAID controller
- RAID 0 = striping
- RAID 1 = mirroring

RAID 2

- Distribute ***bits*** across disks, with ECC
- N data disks, M parity disks
- Multiple-error correction
- Very rarely used



RAID 3

- Distribute ***bits*** across disks, with parity
- Rely on disks to announce erasures
- N data disks, 1 parity disk
- Used in some high-performance applications

RAID 4

- RAID 3, distribute ***sectors*** instead of ***bits***
- Single-sector reads involve only 1 disk: parallel!
- Single-sector writes: read, read, write, write!
- Rarely used: parity disk is a ***hot spot***

RAID 5

- RAID 4, distribute parity among disks
- No more “parity disk hot spot”
- Frequently used

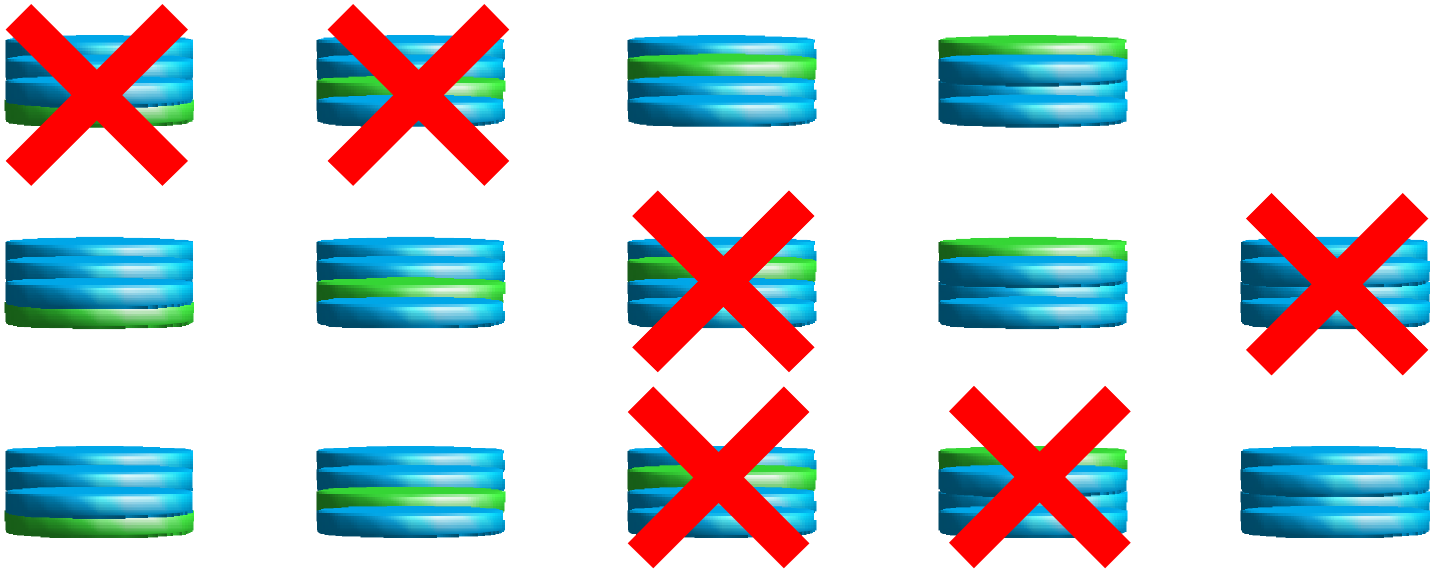
Other fun flavors

- RAID 6, 7, 10, 53
  - Esoteric, single-vendor, non-standard terminology
- RAID 0+1
  - Stripe data across half of your disks
  - Use the other half to mirror the first half
  - Sensible if you like mirroring but need lots of space

Applications

- RAID 0
  - Supercomputer temporary storage / swapping
- RAID 1
  - Simple to explain, reasonable performance, expensive
  - Traditional high-reliability applications (banking)
- RAID 5
  - Cheap reliability for large on-line storage
  - AFS servers

Are failures independent?



Papers

- 1988: Patterson, Gibson, Katz: A Case for Redundant Arrays of Inexpensive Disks (RAID), www.cs.cmu.edu/~garth/RAIDpaper/Patterson88.pdf
- 1990: Chervenak, Performance Measurements of the First RAID Prototype, isi.edu/~annc/papers/masters.ps
- Countless others

Summary

- Need more disks!
  - More space, lower latency, more throughput
- ***Cannot*** tolerate 1/N reliability
- Store information carefully and redundantly
- Lots of variations on a common theme
- You should understand RAID 0, 1, 5