A CASE FOR REDUNDANT ARRAYS OF INEXPENSIVE DISKS (RAID)

D. A. Patterson, G. A. Gibson, R. H. Katz
University of California, Berkeley

Highlights

- The six RAID organizations
- Why RAID 1, 3, 5 and 6 are the most interesting
- The small write problem occurring with RAID 5 and 6

WARNING: *Skip the reliability and availability analyses: they are* ***not correct***

Original Motivation

- Replacing large and expensive mainframe hard drives (IBM 3310) by several cheaper Winchester disk drives
- Will work but introduce a data reliability problem:
  - Assume MTTF of a disk drive is 30,000 hours
  - MTTF for a set of *n* drives is 30,000/*n*
    - *n* = 10 means MTTF of 3,000 hours

Today's Motivation

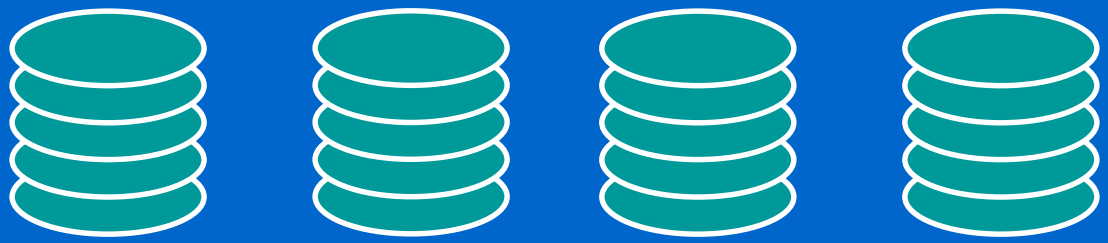
- "Cheap" SCSI hard drives are now big enough for most applications
- We use RAID today for
  - Increasing disk throughput by allowing parallel access
  - Eliminating the need to make disk backups
    - Disks are too big to be backed up in an efficient fashion

RAID LEVEL 0

- No replication
- ***Advantages:***
  - Simple to implement
  - No overhead
- ***Disadvantage:***
  - If array has *n* disks failure rate is *n* times the failure rate of a single disk

RAID levels 0 and 1

RAID level 0



RAID level 1

RAID LEVEL 1

- Mirroring
  - Two copies of each disk block
- Advantages:
  - Simple to implement
  - Fault-tolerant
- Disadvantage:
  - Requires twice the disk capacity of normal file systems

RAID LEVEL 2

- Instead of duplicating the data blocks we use an **error correction** code
- ***Very bad idea*** because disk drives either work correctly or do not work at all
  - Only possible errors are ***omission errors***
  - We need an ***omission correction code***
    - A parity bit is enough to correct a single omission

RAID levels 2 and 3

RAID level 2

RAID level 3

RAID LEVEL 3

- Requires N+1 disk drives
  - N drives contain data (1/N of each data block)
    - Block b[k] now partitioned into N fragments b[k,1], b[k,2], ... b[k,N]
  - Parity drive contains exclusive or of these N fragments

$$p[k] = b[k,1] \oplus b[k,2] \oplus \ldots \oplus b[k,N]$$

How parity works?

- Truth table for XOR (same as parity)

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Recovering from a disk failure

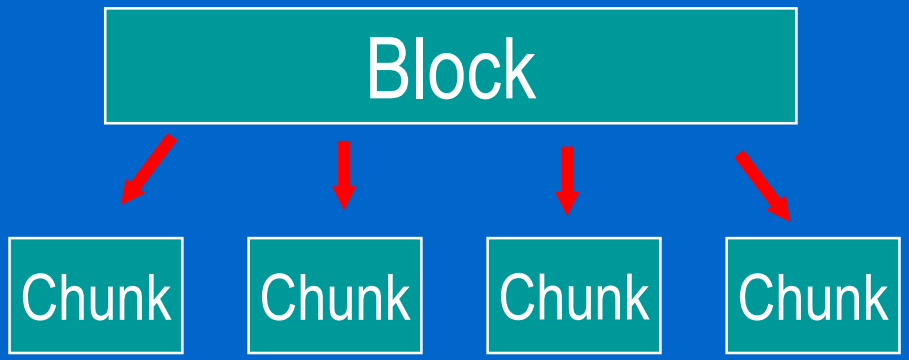
- Small RAID level 3 array with data disks D0 and D1 and parity disk P can tolerate failure of either D0 or D1

D0	D1	P
0	0	0
0	1	1
1	0	1
1	1	0

D1⊕P=D0	D0⊕P=D1
0	0
0	1
1	0
1	1

How RAID level 3 works (I)

- Assume we have $N + 1$ disks
- Each block is partitioned into N equal chunks

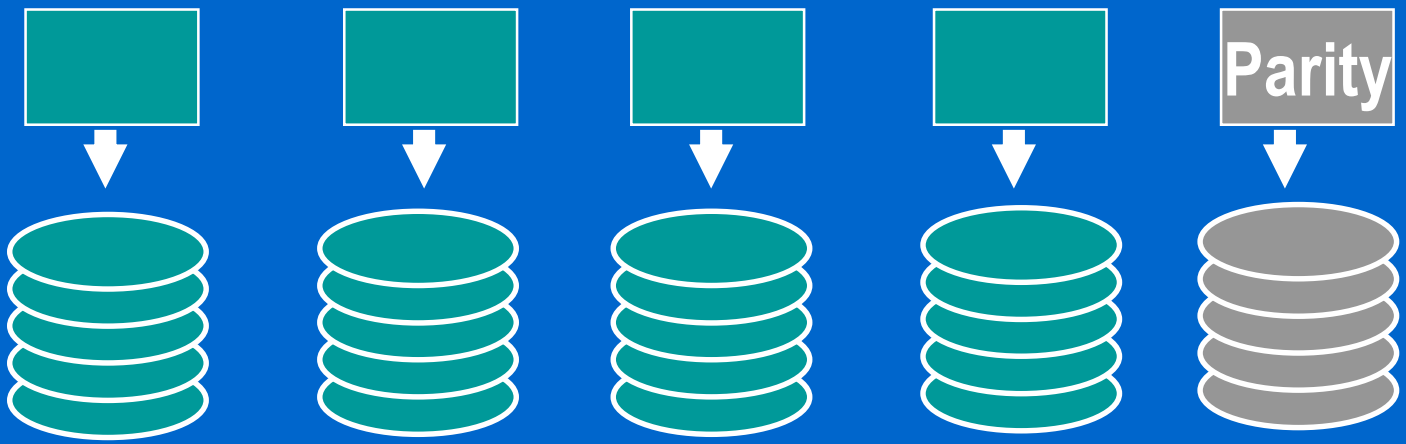


$N = 4$ in example

How RAID level 3 works (II)

- XOR data chunks to compute the parity chunk

- Each chunk is written into a ***separate disk***



How RAID level 3 works (III)

- Each read/write involves ***all disks*** in RAID array
  - Cannot do two or more reads/writes ***in parallel***
  - Performance of array not better than that of a ***single disk***

RAID LEVEL 4 (I)

- Requires N+1 disk drives
  - N drives contain data
    - Individual blocks, not chunks
  - Blocks with same disk address form a ***stripe***

RAID LEVEL 4 (II)

- Parity drive contains **exclusive or** of the *N* blocks in stripe

$$p[k] = b[k] \oplus b[k+1] \oplus \ldots \oplus b[k+N-1]$$

- Parity block now reflects contents of several blocks!
- Can now do parallel reads/writes

RAID levels 4 and 5

RAID level 4

RAID level 5

RAID LEVEL 5

- Single parity drive of RAID level 4 is involved in every write
  - ***Will limit parallelism***
- RAID-5 distribute the parity blocks among the N+1 drives
  - ***Much better***

The small write problem

- Specific to RAID 5
- Happens when we want to update a single block
  - Block belongs to a stripe
  - How can we compute the new value of the parity block

First solution

- Read values of N-1 other blocks in stripe
- Recompute

$$p[k] = b[k] \oplus b[k+1] \oplus \ldots \oplus b[k+N-1]$$

- Solution requires
  - N-1 reads
  - 2 writes (new block and new parity block)

Second solution

- Assume we want to update block b[m]
- Read old values of b[m] and parity block p[k]
- Compute

$$p[k] = \textbf{new}\ b[m] \oplus \textbf{old}\ b[m] \oplus \textbf{old}\ p[k]$$

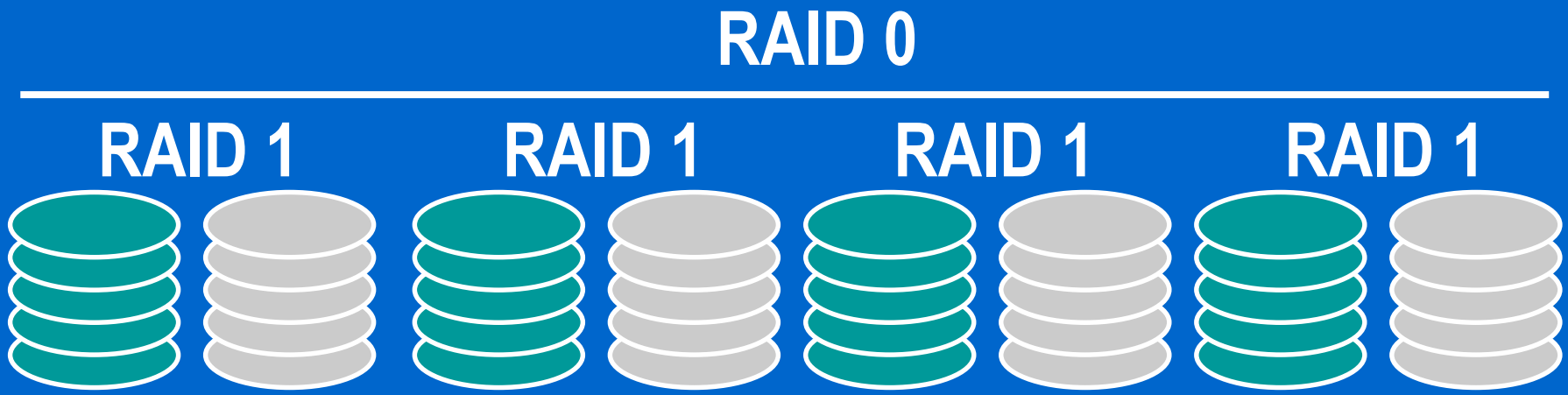
- Solution requires
  - 2 reads (old values of block and parity block)
  - 2 writes (new block and new parity block)

Other RAID organizations (I)

- **RAID 6:**
  - Two check disks
  - Tolerates two disk failures
  - More complex updates

Other RAID organizations (II)

- **RAID 10**:
  - Also known as **RAID 1 + 0**
  - Data are striped (as in RAID 0 or RAID 5) over pairs of mirrored disks (RAID 1)



What about flash drives?

- Having no moving parts should mean ***fewer failures***?
  - Failures still happen
  - Flash drives age as they are written to
  - Irrecoverable red errors occur (at least as frequently as in magnetic disks?)
- Pure Storage uses a proprietary 3D-RAID organization for their SSD stores

CONCLUSION (I)

- RAID original purpose ***was*** to take advantage of Winchester drives that were smaller and cheaper than conventional disk drives
  - Replace a single drive by an array of smaller drives
- ***Current purpose*** is to build fault-tolerant file systems that do not need backups

CONCLUSION (II)

- Low cost of disk drives made RAID level 1 attractive for small installations
- Otherwise pick
  - RAID level 6 for ***higher protection***
    - Can tolerate ***one disk failure and irrecoverable read errors***

A review question

- Consider an array consisting of four 750 GB disks
- What is the storage capacity of the array if we organize it
  - As a RAID level 0 array?
  - As a RAID level 1 array?
  - As a RAID level 5 array?

The answers

- Consider an array consisting of four 750 GB disks
- What is the storage capacity of the array if we organize it
  - As a RAID level 0 array? 3 TB
  - As a RAID level 1 array? 1.5 TB
  - As a RAID level 5 array? 2.25 TB