Antoniu Pop

antoniu.pop@manchester.ac.uk

28 February 2018
Previous lecture

- Characterisation of storage technologies
  - Write once/many/not-too-many and Read many times
- Performance model
  - Seek, Search and Transfer time
  - Latency and Bandwidth
- Limitations
  - Mechanical constraints - latency and reliability issues
- RAID (Redundant Array of Independent/Inexpensive Disks)
Hard Disk Performance (recap)

Seek time  Time for the head to reach the target track.

Search time  Time for the target sector to arrive under the head. Also called rotational latency.

Transfer rate  Amount of data that can be read / written per unit of time. Dependent on access patterns.
Aka. “sustained transfer rate” in contrast to “interface transfer rate”

Disk access time = seek time + search time + transfer time

Note: all values are average as they depend on many factors.
Learning Objectives - Storage 2

- Motivate RAID
- Understand the principles of RAID configurations
- Understand how RAID impacts performance and reliability
- Understand failure & recovery constraints
1956 first HDD IBM 350: \(\sim 3.5\) MB (enough to store one selfie!)
2015 first 10 TB disk: 1000s of times smaller, \(3 \cdot 10^6 \times\) capacity

**1000s of times cheaper!**

Source: [https://www-03.ibm.com/ibm/history/exhibits/storage/storage_350.html](https://www-03.ibm.com/ibm/history/exhibits/storage/storage_350.html)
Technology Trends and Drivers

1956 - 1980s
- Mainframe/Server Disk Drives
  - High capacity, large formats (e.g., 14”)
  - Expensive, low volume market
  - Somewhat slow evolution
- PCs used mostly floppy disks

(Early) 1990s
- Still two markets: Server & PC drives
- Most PCs use hard disks
- PC hard disk sales explode – high volume market
  - Drives costs lower
  - Drives disk technology faster

How to use PC disks to build server-class storage?
**RAID**

**Redundant Array of Independent Disks**

- Compensate for loss of reliability, capacity, performance
- Use lots of cheap(er), (disposable ?) disks
Disks Problems and Solutions (recap)

- Disks are too small
  - Fixed: use multiple disks

- Disks are too slow
  - Fixed: disk striping (RAID 0)

- Disks are unreliable
  - Fixed: disk mirroring (RAID 1)
  - Data redundancy
RAID 0 — Striping

Number of disks | $n$ | 4
---|---|---
Read (short ... long) | $1 \times \cdots n \times$ | $1 \times \cdots 4 \times$
Write (short ... long) | $1 \times \cdots n \times$ | $1 \times \cdots 4 \times$
Failure tolerance | 0 disks | 0 disks
Capacity efficiency | 1 | 100%
RAID 1 — Mirroring

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of disks</td>
<td></td>
<td>$n$</td>
</tr>
<tr>
<td>Read (short ... long)</td>
<td></td>
<td>$1 \times ... n \times$</td>
</tr>
<tr>
<td>Write (short ... long)</td>
<td></td>
<td>$1 \times ... 1 \times$</td>
</tr>
<tr>
<td>Failure tolerance</td>
<td></td>
<td>$n - 1$ disks</td>
</tr>
<tr>
<td>Capacity efficiency</td>
<td></td>
<td>$1/n$</td>
</tr>
</tbody>
</table>
Parity

- Old idea: first tape drive (1951) had a parity track
- Transverse redundancy check

- How does it really work?
Assume we have three blocs $A_0, A_1, A_2$:

<p>| | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_0$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$A_1$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$A_2$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$A_p$ (parity) = 1 0 1 1 1 1 0 0 1 0 1 0

Where $A_p = A_0 \oplus A_1 \oplus A_2$.

And, importantly:

$A_0 = A_p \oplus A_1 \oplus A_2$

$A_1 = A_0 \oplus A_p \oplus A_2$

$A_2 = A_p \oplus A_1 \oplus A_0$
### RAID 3 — Byte-Striping + Parity

#### RAID 3 Diagram

```
Disk 0
   \--- A0
   \--- B0
   \--- C0

Disk 1
   \--- A1
   \--- B1
   \--- C1

Disk 2
   \--- A2
   \--- B2
   \--- C2

Disk 3
   \--- Ap
   \--- Bp
   \--- Cp
```

#### Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>RAID 3 Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of disks</td>
<td>( n + 1 )</td>
</tr>
<tr>
<td>Read (short ... long)</td>
<td>( 1 \times \ldots n \times )</td>
</tr>
<tr>
<td>Write (short ... long)</td>
<td>( 1 \times \ldots n \times )</td>
</tr>
<tr>
<td>Failure tolerance</td>
<td>1 disks</td>
</tr>
<tr>
<td>Capacity efficiency</td>
<td>( n / (n + 1) )</td>
</tr>
<tr>
<td></td>
<td>75%</td>
</tr>
</tbody>
</table>

- **Antoniu Pop**
- **Storage Technologies**

---

(Translation)

### RAID 3 — Byte-Striping + Parity

#### RAID 3 Diagram

```
Disk 0
   \--- A0
   \--- B0
   \--- C0

Disk 1
   \--- A1
   \--- B1
   \--- C1

Disk 2
   \--- A2
   \--- B2
   \--- C2

Disk 3
   \--- Ap
   \--- Bp
   \--- Cp
```

#### Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>RAID 3 Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of disks</td>
<td>( n + 1 )</td>
</tr>
<tr>
<td>Read (short ... long)</td>
<td>( 1 \times \ldots n \times )</td>
</tr>
<tr>
<td>Write (short ... long)</td>
<td>( 1 \times \ldots n \times )</td>
</tr>
<tr>
<td>Failure tolerance</td>
<td>1 disks</td>
</tr>
<tr>
<td>Capacity efficiency</td>
<td>( n / (n + 1) )</td>
</tr>
</tbody>
</table>

- **Antoniu Pop**
- **Storage Technologies**

---

(Translation)

### RAID 3 — Byte-Striping + Parity

#### RAID 3 Diagram

```
Disk 0
   \--- A0
   \--- B0
   \--- C0

Disk 1
   \--- A1
   \--- B1
   \--- C1

Disk 2
   \--- A2
   \--- B2
   \--- C2

Disk 3
   \--- Ap
   \--- Bp
   \--- Cp
```

#### Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>RAID 3 Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of disks</td>
<td>( n + 1 )</td>
</tr>
<tr>
<td>Read (short ... long)</td>
<td>( 1 \times \ldots n \times )</td>
</tr>
<tr>
<td>Write (short ... long)</td>
<td>( 1 \times \ldots n \times )</td>
</tr>
<tr>
<td>Failure tolerance</td>
<td>1 disks</td>
</tr>
<tr>
<td>Capacity efficiency</td>
<td>( n / (n + 1) )</td>
</tr>
</tbody>
</table>

- **Antoniu Pop**
- **Storage Technologies**

---

(Translation)
RAID 4 — Block-Striping + Parity

<table>
<thead>
<tr>
<th></th>
<th>Disk 0</th>
<th>Disk 1</th>
<th>Disk 2</th>
<th>Disk 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ap</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Number of disks: \( n + 1 \)

- **Read (short ... long)**: \( 1 \times \ldots n \times 4 \)
- **Write (short ... long)**: \( 0.5 \times \text{(RMW)} \ldots n \times 3 \times \)
- **Failure tolerance**: 1 disks
- **Capacity efficiency**: \( n/(n + 1) 75\% \)
RAID 5 — Block-Striping + Distrib. Parity

**Number of disks**

<table>
<thead>
<tr>
<th>Disk</th>
<th>Number of Disks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Read (short ... long)</th>
<th>1× ... n + 1×</th>
<th>1× ... 4×</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write (short ... long)</td>
<td>0.5× (RMW) ... n×</td>
<td>0.5× ... 3×</td>
</tr>
</tbody>
</table>

**Failure tolerance**

| Failure tolerance | 1 disks | 1 disks |

**Capacity efficiency**

| Capacity efficiency | n/(n + 1) | 75%     |
### RAID 6 — Double Distributed Parity

**Number of disks**

<table>
<thead>
<tr>
<th>Disk 0</th>
<th>Disk 1</th>
<th>Disk 2</th>
<th>Disk 3</th>
<th>Disk 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>A1</td>
<td>A2</td>
<td>A_q</td>
<td>A_q</td>
</tr>
<tr>
<td>B_q</td>
<td>B_0</td>
<td>B_1</td>
<td>B_p</td>
<td>B_p</td>
</tr>
<tr>
<td>C_p</td>
<td>C_q</td>
<td>C_0</td>
<td>C_1</td>
<td>C_2</td>
</tr>
<tr>
<td>D_2</td>
<td>D_p</td>
<td>D_q</td>
<td>D_0</td>
<td>D_1</td>
</tr>
<tr>
<td>E_1</td>
<td>E_2</td>
<td>E_p</td>
<td>E_q</td>
<td>E_0</td>
</tr>
</tbody>
</table>

**RAID 6**

- **Number of disks**: $n + 2$
- **Read (short ... long)**: $1 \times \ldots n + 2 \times$
- **Write (short ... long)**: $0.5 \times \text{(RMW)} \ldots n \times$
- **Failure tolerance**: 2 disks
- **Capacity efficiency**: $n / (n + 2)$

**Example**

- **Number of disks**: 5
- **Read (short ... long)**: $1 \times \ldots 5 \times$
- **Write (short ... long)**: $0.5 \times \ldots 3 \times$
Nested RAID

- Each raid configuration comes with tradeoffs
- Combine RAID configurations to alleviate shortcomings
- Multiple RAID layers
RAID 1+0 (aka. RAID 10)

Number of disks: $m_{[\text{RAID 1}]} \cdot n_{[\text{RAID 0}]} = 2 \cdot 2 = 4$

Read (short ... long): $1 \times \ldots n \cdot m \times$
Write (short ... long): $1 \times \ldots n \times$
Failure tolerance: $m - 1$ disks
Capacity efficiency: $n / (m \cdot n) = 1 / m$
Number of disks: $m_{[\text{RAID 1}]} \cdot n_{[\text{RAID 0}]} = 2 \cdot 2 = 4$

Read (short ... long): $1 \times \ldots n \cdot m \times$
Write (short ... long): $1 \times \ldots n \times$
Failure tolerance: $m - 1$ disks
Capacity efficiency: $n/(m \cdot n) = 1/m$
RAID 10 vs. 01 – different?
RAID Failure Mode Operation

What happens when a disk fails?

RAID 0  Lose all data (hope there’s more than one RAID layer)
RAID 1  Business as usual, hot-swap the failed disk
RAID 2-6  Operate in degraded mode

- If data drive failed, every read must be reconstructed
- If parity drive failed, low performance impact
- Replace drive (hot-swapping: the system continues running)
- Rebuild the array (re-constitute the state of the lost drive)
RAID Recovery Limitations

Rebuilding a degraded array

- Sequentially
- How long?
- On live system?

Risk of failure during recovery

- Statistical distortion:
  higher risk of multiple failures within a narrow time frame

- RAID 5 risk advisory notice:
  Do not use for business-critical data! [Dell]
No performance degradation on disk failure!
Where to Implement RAID?

- **Software – Operating System**
  - Most OS now provide software RAID
  - E.g. Linux `md` (multiple devices) supports RAID 0, 1, 4, 5, 6 plus nestings

- **Software – File System**
  - E.g., ZFS

- **Dedicated hardware (RAID controller)**
Array Failure Rates (full data loss)

Failure rate of a disk drive: $r$ (with some assumptions!)

Failure rate $R$ of an array of $n$ disks (RAID) where $k$ disks can safely fail:

$$R = 1 - (P(0) + P(1) + ... + P(k))$$

where $P(i)$ is the probability of precisely $i$ disks failing:

$$P(i) = \binom{n}{i} r^i (1 - r)^{n-i}$$
Array Failure Rates for RAID configurations

RAID 0 \(1 - (1 - r)^n\) \(\text{(0 disks can safely fail)}\)

RAID 1 \(r^n\) \(\text{(}n - 1\text{ disks can safely fail)}\)

RAID 2 It’s complicated

RAID 3-5 \(1 - (1 - r)^n - \binom{n}{1} r^1 (1 - r)^{n-1}\) \(\text{(1 disk can safely fail)}\)

RAID 6 \(1 - (1 - r)^n - \binom{n}{1} r^1 (1 - r)^{n-1} - \binom{n}{2} r^2 (1 - r)^{n-2}\) \(\text{(2 disks can safely fail)}\)
Array Failure Rate (Raid 6 example)

Failure rate:

\[ 1 - (1 - r)^n - \binom{n}{1} r^1 (1 - r)^{n-1} - \binom{n}{2} r^2 (1 - r)^{n-2} \]

Example drive 1%/year failure rate with RAID 6 (3+2 drives):

\[
1 - 0.99^5 - 5 \cdot 0.01 \cdot 0.99^4 - \frac{4 \cdot 5}{2} \cdot 0.01^2 \cdot 0.99^3 = 0.0000098511
\]

Rounding up, that’s a 1% failure rate of the RAID in 1000 years!
RAID 2 — Bit-Striping + Hamming Code

Number of disks: $2^k - 1$  
$k = 3 \Rightarrow 7$

Read (short ... long): $1 \times \ldots 2^k - k - 1 \times$  
$1 \times \ldots 4 \times$

Write (short ... long): $1 \times \ldots 2^k - k - 1 \times$  
$1 \times \ldots 4 \times$

Failure tolerance: $1 \ldots 2^* \text{ disks}$  
$1 \ldots 2^* \text{ disks}$

Capacity efficiency: $\frac{2^k - k - 1}{2^k - 1}$  
$\frac{4}{7} \Rightarrow 57\%$
Hamming Codes

RAID 2 no longer used, but...
- Hamming code error correction
- ECC