Lecture

Operating System

38. RAID

Professor Dr. Michael Mächtel

38.RAID

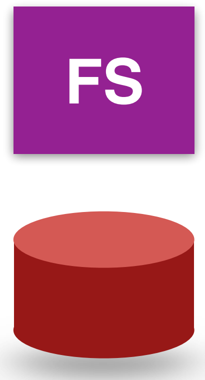
1. **Interface and RAID Internals**
2. **RAID Levels**
3. **Summary**

38.RAID

1. **Interface and RAID Internals**
2. RAID Levels
3. Summary

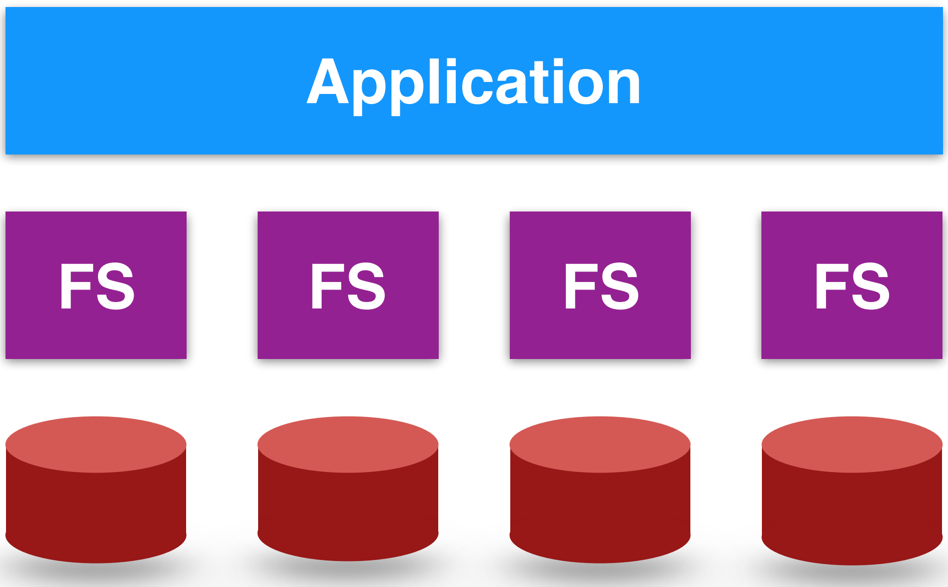
Only One Disk?

- Sometimes we want many disks — why?
  - capacity
  - reliability
  - performance
- Challenge: most file systems work on only one disk



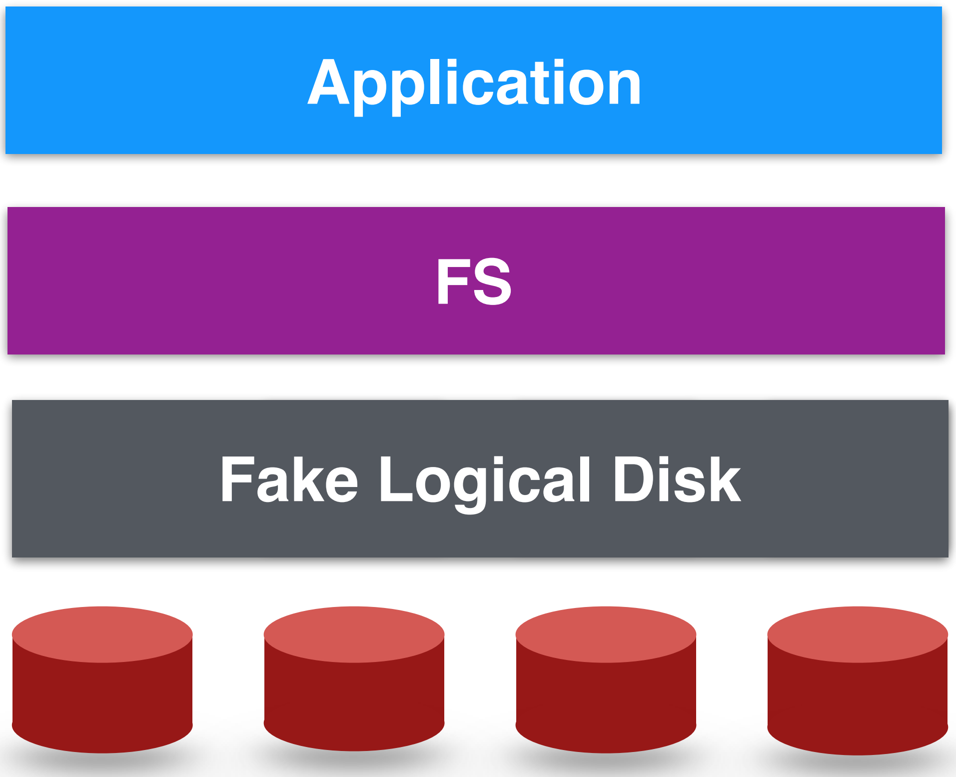
Solution 1: JBOD

- Application is smart, stores different files on different file systems.
- JBOD: **J**ust a **B**unch **O**f **D**isks



Solution 2: RAID

- Build logical disk from many physical disks
- RAID: **R**edundant **A**rray of **I**nexpensive **D**isks



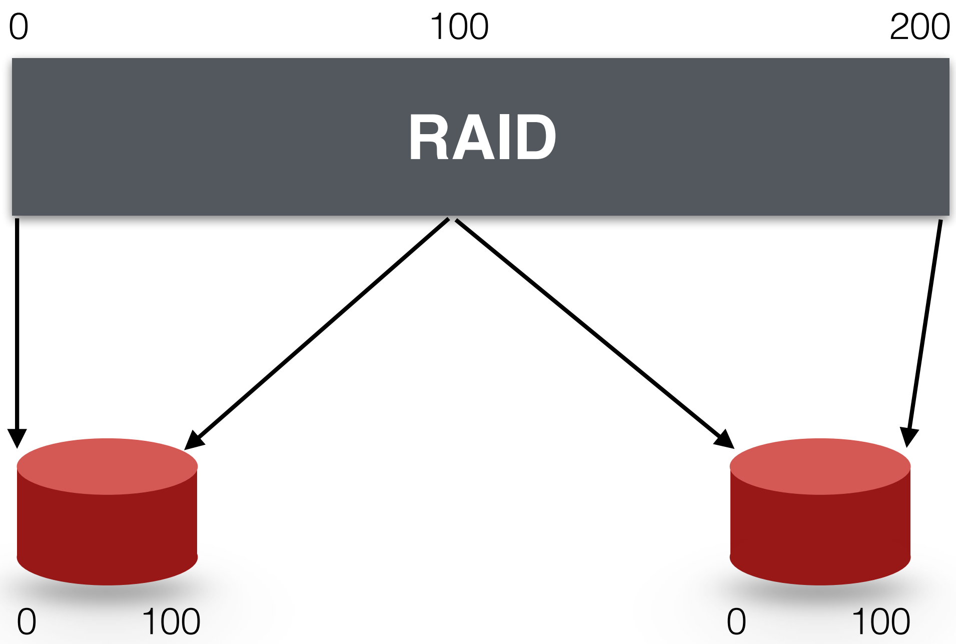
- RAID is:
  - transparent
  - deployable
- Logical disk gives
  - capacity
  - performance
  - reliability

Why Inexpensive Disks?

- Economies of scale! Commodity disks cost less
    - Can buy **many** commodity H/W components for the same price as few high-end components
    - **Strategy: write S/W** to build high-quality logical devices from many cheap devices
- **Alternative** to RAID: buy an **expensive**, **high-end disk**
- Use multiple disks in concert to build a faster, bigger, and more reliable disk system.
    - RAID just looks like a big disk to the host system.
- **Advantage**
    - **Performance** & **Capacity**: Using multiple disks in parallel
    - **Reliability**: RAID can tolerate the loss of a disk.

General Strategy: MAPPING

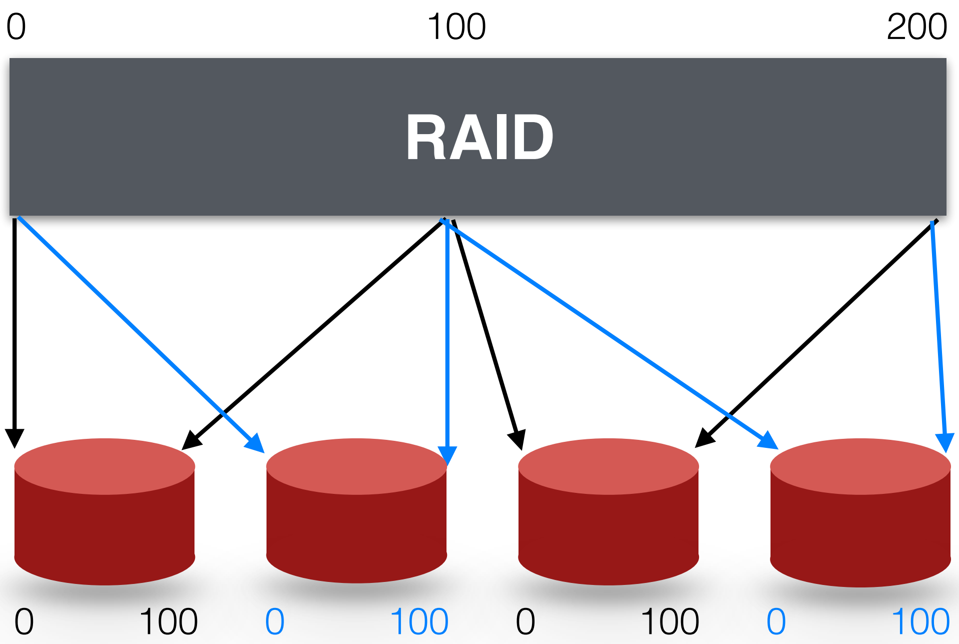
- Build fast, large disk from smaller ones.



RAIDs provide these advantages transparently to systems that use them.

General Strategy: REDUNDANCY

- Add even more disks for reliability.



RAIDs provide these advantages transparently to systems that use them.

RAID Interface

- When a RAID receives I/O request,
  - The RAID **calculates** which disk to access.
  - The RAID **issue** one or more **physical I/Os** to do so.
- RAID example: A mirrored RAID system
  - Keep two copies of each block (each one on a separate disk)
  - Perform two physical I/Os for every one logical I/O it is issued.

RAID Internals

- A **microcontroller**
  - Run firmware to direct the operation of the RAID
- **Volatile memory** (such as DRAM)
  - Buffer data blocks
- **Non-volatile memory** (perhaps)
  - Buffer writes safely
- **Specialized logic**
  - to perform parity calculation

Fault Model

- RAIDs are designed to **detect** and **recover** from certain kinds of disk faults.
- **Fail-stop** fault model
  - A disk can be in one of two states: *Working* or *Failed*.
    - Working: all blocks can be read or written.
    - Failed: the disk is permanently lost.
  - RAID controller can immediately observe when a disk has failed.

Mapping

- How should we map logical block addresses to physical block addresses?
  - Some similarity to virtual memory
- **Dynamic** mapping: use data structure (hash table, tree)
  - page tables
- **Static** mapping: use simple math
  - RAID

Redundancy

- Trade-offs to amount of redundancy
- Increase number of copies:
  - improves **reliability** (and maybe **performance**)
- Decrease number of copies (deduplication)
  - improves **space efficiency**

Reasoning About RAID

- **RAID**: system for mapping logical to physical blocks
  - Which logical blocks map to which physical blocks?
  - How do we use extra physical blocks (if any)?
  - **Different RAID levels** make different trade-offs
- **Workload**: types of reads/writes issued by applications (sequential vs. random)
- **Metric**: capacity, reliability, performance

How to evaluate a RAID?

- **Capacity**:
  - How much useful capacity is available to systems?
- **Reliability**:
  - How many disk faults can the given design tolerate?
- **Performance**:
  - How long does each workload take?
- **Normalize** each to characteristics of one disk

38.RAID

1. Interface and RAID Internals
2. **RAID Levels**
3. Summary

RAID Level 0: Striping

- RAID Level 0 is the simplest form as striping blocks.
  - Spread the blocks across the disks in a round-robin fashion.
  - No redundancy
  - Excellent performance and capacity

Disk 0	Disk 1	Disk 2	Disk 3
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Stripe
(The blocks in the same row)

RAID-0: Simple Striping
(Assume here a 4-disk array)

RAID Level 0 (Cont.)

- Example: RAID-0 with a bigger chunk size
  - Chunk size : 2 blocks (8 KB)
  - A Stripe: 4 chunks (32 KB)

Disk 0	Disk 1	Disk 2	Disk 3
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

chunk size:
2blocks

Striping with a Bigger Chunk Size

Chunk Sizes

- Chunk size mostly affects performance of the array
  - **Small chunk size**
    - Increasing the parallelism
    - Increasing positioning time to access blocks
  - **Big chunk size**
    - Reducing intra-file parallelism
    - Reducing positioning time

Determining the “best” chunk size is hard to do.
Most arrays use larger chunk sizes (e.g., 64 KB)

RAID Level 0 Analysis

- **Capacity**
  - RAID-0 is perfect.
  - Striping delivers N disks worth of useful capacity.
- **Performance** of striping
  - RAID-0 is excellent.
  - All disks are utilized often in parallel.
- **Reliability**
  - RAID-0 is bad.
  - Any disk failure will lead to data loss.

Evaluating RAID Performance

- Consider two performance metrics
  - Single request latency
  - Steady-state throughput
- Workload
  - **Sequential**:
    - access 1MB of data (block (B) → ... → block (B + 1MB))
  - **Random**:
    - access 4KB at random logical address
- A disk can transfer data at
  - `S` MB/s under a sequential workload
  - `R` MB/s under a random workload

Evaluating Performance Example

- sequential (`S`) vs random (`R`)
  - **Sequential**: transfer 10 MB on average as cont. data
  - **Random**: transfer 10 kB on average
  - Average seek time: 7 ms
  - Average rotational delay: 3 ms
  - Transfer rate of disk: 50 MB/s
- Results
  - `S` = $\frac{Amount\ of\ Data}{Time\ to\ access} = \frac{10\ MB}{210\ ms} = 47{,}62\ MB/s$
  - `R` = $\frac{Amount\ of\ Data}{Time\ to\ access} = \frac{10\ kB}{10{,}195\ ms} = 0{,}981\ MB/s$

Evaluating RAID-0 Performance

$N: \text{the number of disks}$

- Single request latency
  - Identical to that of a single disk.
- Steady-state throughput
  - Sequential workload: $N * S \; MB/s$
  - Random workload: $N * R \; MB/s$

RAID Level 1: Mirroring

- RAID Level 1 tolerates **disk failures**.
  - **Copy** more than one of **each block** in the system.
  - Copy block places on a separate disk.
    - RAID-10 (RAID 1+0) : mirrored pairs and then stripe
    - RAID-01 (RAID 0+1) : contain two large striping arrays, and then mirrors

Disk 0	Disk 1	Disk 2	Disk 3
0	0	1	1
2	2	3	3
4	4	5	5
6	6	7	7

Simple RAID-1: Mirroring (Keep two physical copies)

RAID-1 Analysis ...

- **Capacity: RAID-1 is Expensive**

  N : the number of disks

  - The useful capacity of RAID-1 is N/2.
- **Reliability**: RAID-1 does well.
  - It can tolerate the failure of any one disk (up to N/2 failures depending on which disk fail).

... Performance of RAID-1

- Two physical writes to complete
  - It suffers the worst-case seek and rotational delay of the two requests
  - Steady-state throughput

    $N:$ *the number of disks*

    - Sequential Write: $\frac{N}{2} * S\ MB/s$
      - Each logical write must result in two physical writes.
    - Sequential Read: $\frac{N}{2} * S\ MB/s$; $Others: N * S\ MB/s$
      - Each disk will only deliver halt its peak bandwidth
    - Ramdom Write: $\frac{N}{2} * R\ MB/s$
      - Each logical write must turn into two physical writes
    - Random Read: $N * R\ MB/s$
      - Distribute the reads across all the disks

RAID Level 4 : Saving Space With Parity

- Add a **single parity block**
  - A Parity block stores the redundant information for that stripe of blocks.

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	0	1	1	P0
2	2	3	3	P1
4	4	5	5	P2
6	6	7	7	P3

* P: Parity

Five-disk RAID-4 system layout

RAID Level 4 (Cont.)

- **Compute parity**: the XOR of all of bits

C0	C1	C2	C3	P
0	0	1	1	XOR(0,0,1,1)=0
0	1	0	0	XOR(0,1,0,0)=1

- **Recover from parity**
  - Imagine the bit of the C2 in the first row is lost.
    - Reading the other values in that row : 0, 0, 1
    - The parity bit is 0 → even number of 1's in the row
    - What the missing data must be: a 1.

RAID-4 Analysis ...

N : the number of disks

- **Capacity**
  - The useful capacity is (N-1).
- **Reliability**
  - RAID-4 tolerates <u>1 disk failure</u> and no more.

... Performance of RAID-4

- Steady-state throughput

$N:$ *the number of disks*

  - Sequential read: $(N-1)*S\ MB/s$
  - Sequential write: $(N-1)*S\ MB/s$

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P0
4	5	6	7	P1
8	9	10	11	P2
12	13	14	15	P3

  - Random read: $(N-1)*R\ MB/s$

Random write performance for RAID-4

Overwrite a block + update the parity

- **Method 1**: *additive parity*
    - Read in all of the other data blocks in the stripe
    - XOR those blocks with the new block (1)
    - **Problem**: the performance scales with the number of disks
- **Method 2**: *subtractive parity*

C0	C1	C2	C3	P
0	0	1	1	XOR(0,0,1,1)=0

- Update C2(old) ➝ C2(new)
    - Read in the old data at C2 (C2(old)=1) and the old parity (P(old)=0)
    - Calculate P(new): $P(new) = \left(C2(old)\ \ XOR\ \ C2(new)\right)\ \ XOR\ \ P(old)$
        - If C2(new) == C2(old) ➝ P(new)==P(old)
        - If C2(new) != C2(old) ➝ Flip the old parity bit

Small-write problem

- The parity disk can be a bottleneck.
  - Example: update blocks 4 and 13 (marked with *)

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P0
*4	5	6	7	+P1
8	9	10	11	P2
12	*13	14	15	+P3

- Disk 0 and Disk 1 can be accessed in parallel.
- Disk 4 prevents any parallelism.

RAID-4 throughput under random small writes is $\frac{R}{2}$ MB/s (*terrible*)

I/O latency in RAID-4

- **A single read**
  - Equivalent to the latency of a single disk request.
- **A single write**
  - Two reads and then two writes
    - Data block + Parity block
    - The reads and writes can happen in parallel.
  - Total latency *is about twice* that of a single disk.

RAID Level 5: Rotating Parity

- RAID-5 **is solution of** small write problem.
  - Rotate the parity blocks across drives.
  - Remove the parity-disk bottleneck for RAID-4

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P0
5	6	7	P1	4
10	11	P2	8	9
15	P3	12	13	14
P4	16	17	18	19

RAID-5 With Rotated Parity

RAID-5 Analysis ...

- **Capacity**

  $N:$ *the number of disks*

  - The useful capacity for a RAID group is (N-1)
- **Reliability**
  - RAID-5 tolerates 1 disk failure and no more

… Performance of RAID-5

$N: \text{ the number of disks}$

- **Sequential** read and write
- A singe read and write request

Same as RAID-4

- **Random read**: a litte better than RAID-4
  - RAID-5 can utilize all of the disks
- **Random write**:
  - Bandwith small writes: $\frac{N}{4} * R \ MB/s$
  - The factor of four loss is cost of using parity-based RAID

38.RAID

1. Interface and RAID Internals
2. RAID Levels
3. **Summary**

RAID Comparison: A Summary

N : the number of disks

D : the time that a request to a single disk take

	RAID-0	RAID-1	RAID-4	RAID-5
Capacity	N	N/1	N-1	N-1
Reliability	0	1 (for sure) N/2 (if lucky)	1	1
Throughput				
Sequential	N · S	½N · S	(N-1) · S	(N-1) · S
Sequential	N · S	½N · S	(N-1) · S	(N-1) · S
Random Read	N · R	N · R	(N-1) · R	N · R
Random Write	N · R	½N · R	½R	¼N · R
Latency				
Read	D	D	D	D
Write	D	D	2D	2D

RAID Capacity, Reliability, and Performance

RAID Comparison: A Summary

- **Performance** and do not care about reliability
  - RAID-0 (Striping)
- **Random I/O** performance and **Reliability**
  - RAID-1 (Mirroring)
- **Capacity** and **Reliability**
  - RAID-5
- **Sequential I/O** and Maximize **Capacity**
  - RAID-5

Thanks

Questions?