Storage Technologies - 3

COMP 25212 - Lecture 10

Antoniu Pop

antoniu.pop@manchester.ac.uk

2 March 2018

Previous lectures

- Hard Disk Drives
  - Technology trends, evolution
  - Performance model (Seek, Search and Transfer time)
  - Limitations (Latency and Bandwidth, Reliability)
- Solution: RAID (Redundant Array of Independent/Inexpensive Disks)
  - Reliability: redundancy
  - Performance: increase transfer rate through parallelization
  - RAID types
  - Performance and reliability characteristics
- RAID reliability evaluation

RECAP: Array Failure Rates

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

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

$$\mathcal{R} = 1 - \left(\mathcal{P}(0) + \mathcal{P}(1) + \ldots + \mathcal{P}(k)\right)$$

where $\mathcal{P}(i)$ is the probability of precisely i disks failing:

$$\mathcal{P}(i) = \binom{n}{i} r^i (1-r)^{n-i}$$

RECAP: Failure Rates of RAID configurations

RAID 0 $1 - (1 - r)^n$ (**0** disks can safely fail)

RAID 1 r^n ($n - 1$ disks can safely fail)

RAID 2 *It's complicated*

RAID 3-5 (**1** disk can safely fail)

$$1 - (1 - r)^n - \binom{n}{1} r^1 (1 - r)^{n-1}$$

RAID 6 (**2** disks can safely fail)

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

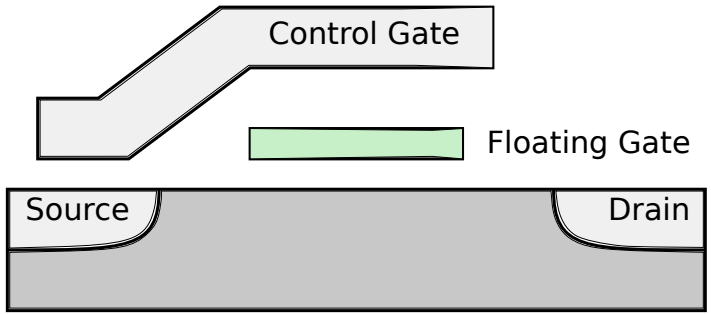
Learning Objectives - Storage 3

- Understand characteristics of Solid-State Drives (SSD)
- Compare SSD and Hard Disks
- Understand Logical Volume Management (LVM)
- Understand Storage Area Networks (SAN)
- Relate LVM and SAN to a modern File System implementation

Solid-State Drive/Disk (SSD)

Flash Memory:

- Floating Gate Field Effect Transistor
- Charge stored on the floating gate
- No electrical connection



- Conceptually like a switch: on (0) / off (1)
- Possibly multi-level (4 states - 2 bits)

Flash Controller

Issues:

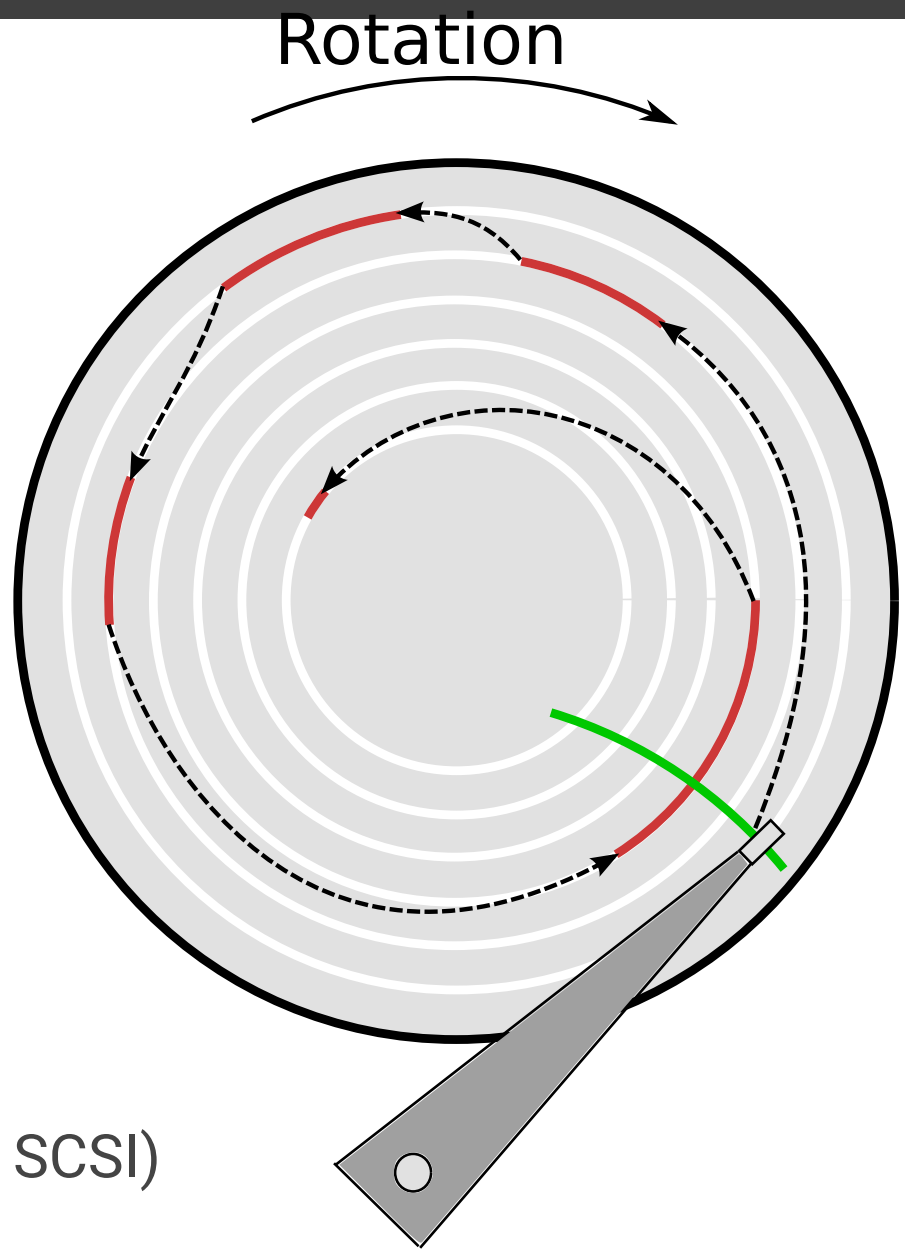
- Data retention (10 year?)
- Wear-out with write cycles (aka P/E cycles)
- Performance degradation with wearout

Implement:

- Error Correcting Codes
- (Bad) block remapping
- Wear-levelling:
  - remap Logical Block Addresses (LBA) to physical addresses
  - avoid wearing out specific blocks

Disk access example (recap)

- Host initiates read
  sends a list of blocks to read
- Block schedule requested...
- ... may not be optimal
- and leads to extra revolutions
- HDD internal processor
  optimizes the schedule
- No direct mapping from
  block numbers to the
  sector/track/cylinder position
  (high-level interfaces like ATA / SCSI)



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.

Hard Disks are too slow (recap)

Slow because of:

- High seek time
  - Reduce the number of times the head must move
  - Multiple platters $\Longrightarrow$ more tracks/sectors per cylinder
- High search time (aka. rotational latency)
  - Increase the rotation speed (e.g., server disks up to 15000 RPM)
- Low sustained transfer rate
  - Increase rotation speed (physical limitations)
  - Increase the recording density (physical limitations)
  - Apply cache and prefetch principles
  - **“Stripe” file system across multiple disks**

SSD vs. HDD

	HDD	SSD
Streaming Reads	**205** MB/s	**530** MB/s
Streaming Writes	**205** MB/s	**240** MB/s
Random 4kB Read	**15.5** ms	**11** μs
Random 4kB Write	**6.4** ms	**23** μs
Power	**4**/**6**/**8** W	**0.3**/**4.2** W
Capacity – Price	**4** TB – £**140**	**250** GB – £**125**
Price per GB	£**0.035**/GB	£**0.5**/GB

Hitachi 7k4000 Samsung SSD 840

Example: disk access time (1 - recap)

How long would it take **on average** to read / write a 512 byte sector on this disk?

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

seek time: **8.5 ms**

search time: the disk must, on average, complete a half rotation

$$\textbf{7200 RPM} \Longrightarrow \frac{\textbf{0.5 rotations} \cdot \textbf{60} \frac{\textbf{sec}}{\textbf{min}}}{\textbf{7200 RPM}} = \textbf{4.16 ms}$$

transfer time: $\frac{\textbf{512 B}}{\textbf{177}\cdot\textbf{10}^{\textbf{6}}\ \textbf{B/sec}} = \textbf{2.89}\ \mu\textbf{s}$

$$\textbf{access time} = 8.5 + 4.16 + 2.89 \cdot 10^{-3} = \textbf{12.66 ms}$$

Example: disk access time (2 - recap)

How long would it take **on average** to read / write 512 MB on this disk? (assuming sectors are “contiguous”)

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

seek time: **8.5 ms**

search time: the disk must, on average, complete a half rotation

$$\textbf{7200 RPM} \Longrightarrow \frac{\textbf{0.5 rotations} \cdot \textbf{60}\,\frac{\textbf{sec}}{\textbf{min}}}{\textbf{7200 RPM}} = \textbf{4.16 ms}$$

transfer time: $\frac{512 \cdot 10^6\ \text{B}}{177 \cdot 10^6\ \text{B/sec}} = 2.89\ \text{s}$

$$\textbf{access time} = 8.5 \cdot 10^{-3} + 4.16 \cdot 10^{-3} + 2.89 = 2.9\ \text{s}$$

Storage Virtualization

File System (classical)

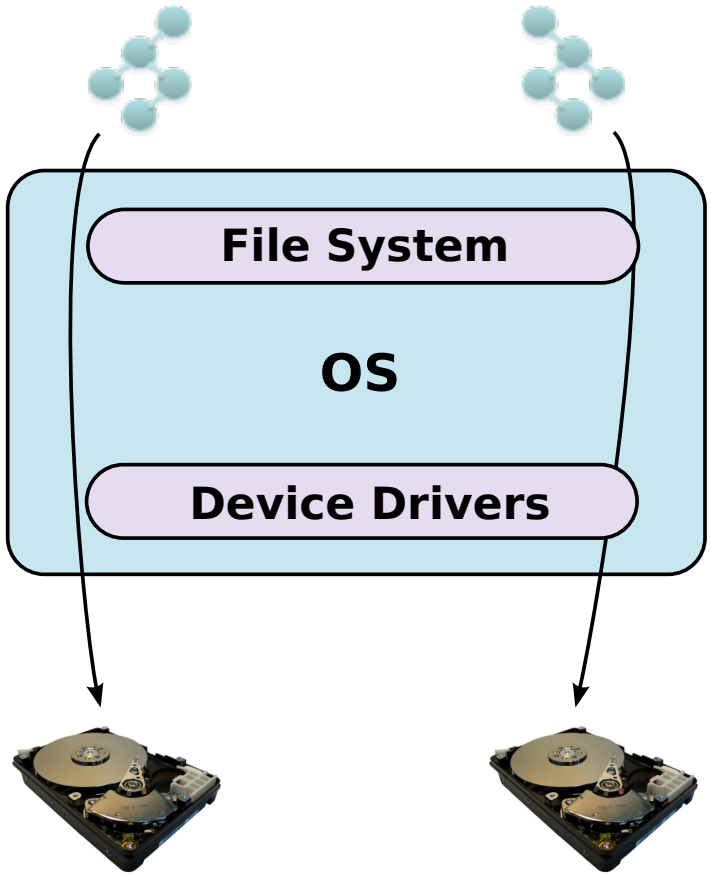
- FS to HDD partition mapping
- FS does not span multiple drives

RAID changes this

- E.g., Striping or Mirroring

Storage Virtualization:

- break the FS/HDD mapping



Logical Volume Management

- Virtual mapping between file system code and physical device
- Similar (but not identical!) to virtual memory addressing
  - FTSE: one more level of indirection
- “Volume Group”: set of drives in a pool
- Storage space in “Volume Group” divided into “Physical Extents”
  - usually all same size
- “Logical Volume” is a set of “Physical Extents”

Logical Volume Management

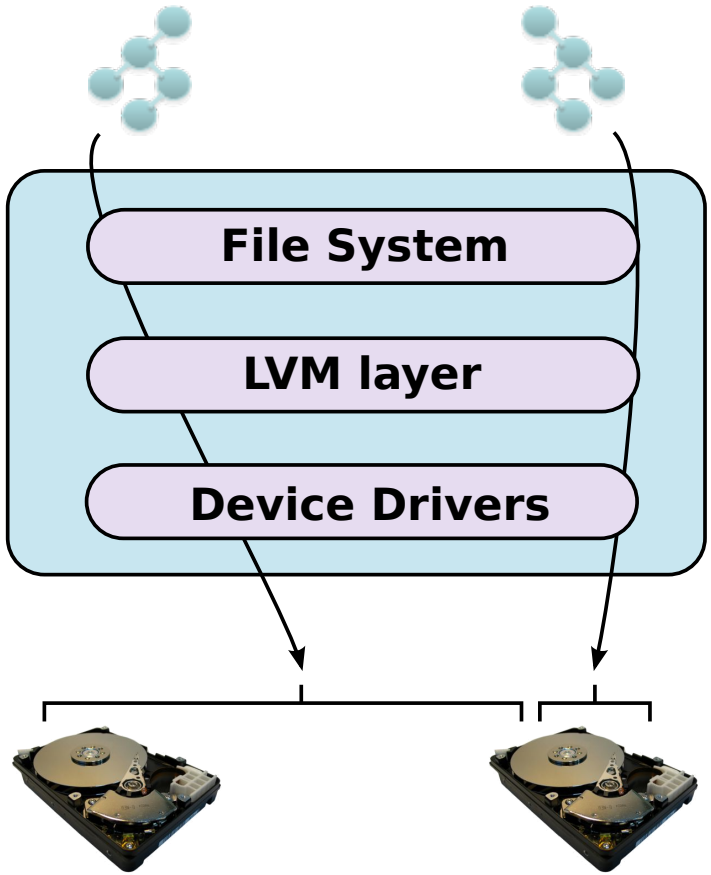
Mirror/Stripe/RAID

- within the LVM layer

Resize the File System

- add physical extents
- extend the partition

“Snapshot” a live filesystem



Example: the Linux File System

- / – mostly read: want fast seeks, high read transfer rates
- swap – read / write: want high bandwidth, data loss (?)
- /opt – infrequent access
- /var – huge, infrequent access

- Mirror /
- Stripe swap
- spare space to /opt and /var

LVM Example

Efficient / flexible resource management

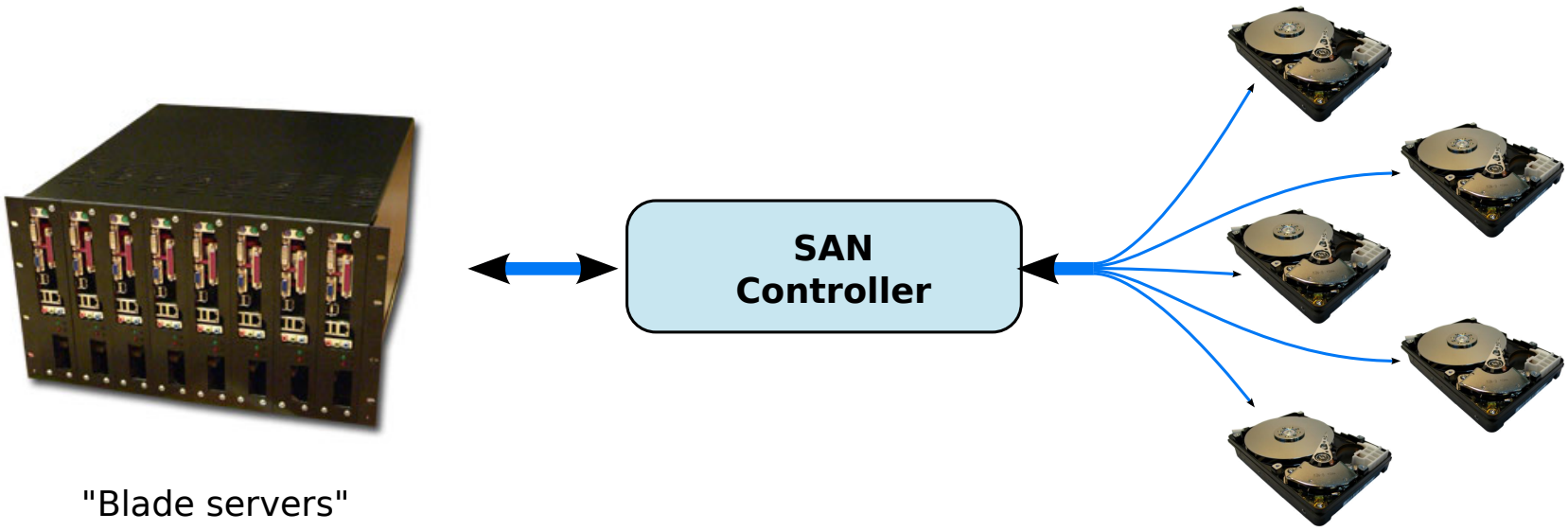
- / is mirrored across two disks
- swap is striped across two other disks

- /opt uses space on one disk
- /var takes the remaining space across other 3 disks

Storage Area Networks

- Implement LVM features in a separate storage controller
- Connect multiple servers to storage controller
  - via SCSI, or FibreChannel, or Infiniband, or...
  - SAN over Ethernet, aka Networked Attached Storage (NAS)
- Share disk resources across multiple servers
- Rapid migration of disk images

SAN Controller



"Blade servers"

Decouple **compute servers** from **storage servers**.

Connect through network:

- Bandwidth ?
- Latency ?

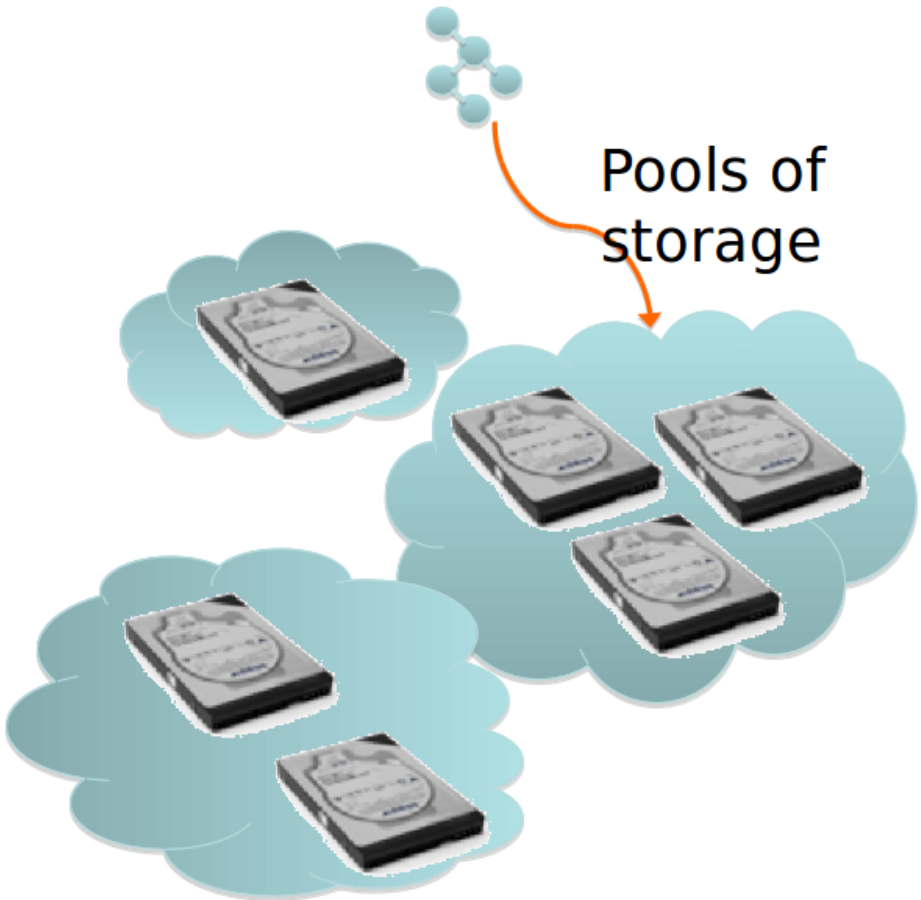
SAN Key Features

- Functionality
  - Key element of **System Virtualization**
  - Migrating virtual machines
  - “De-duping” – share common subsets of file systems (think Virtual Machine images!)
- Management:
  - Manage storage separately from server physical resources
  - Maximize flexibility of storage provisioning

ZFS – Volume Aware File System

Marketing claims:

- Lost a file?
- Run out of space?
- Difficult disk upgrade?
- Want to grow/shrink?
- Data Corruption?



ZFS Solutions

- Lost a file?
  - Copy-on-Write (CoW)
  - simple rollback/recovery
  - (indirect wear-leveling)
- Run out of space / difficult disk upgrade?
  - Add new storage to live systems
  - Self-checking, self-healing
- Want to grow / shrink?
- Data Corruption?
  - end-to-end sumchecking

ZFS combines File System and Logical Volume Management