Storage Management

Bojan Nokovic

Based on: "Operating Systems Concepts", 10th Edition Silberschatz Et al.

Apr. 2021

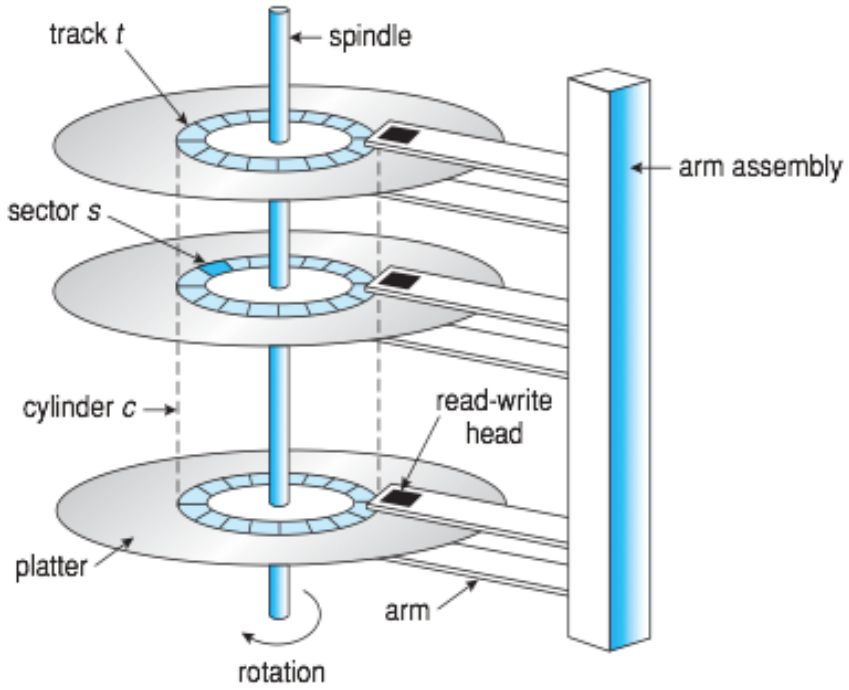
Overview of Mass Storage Structure

Bulk of secondary storage for modern computers is hard disk drives (HDDs) and nonvolatile memory (NVM) devices

HDDs spin platters of magnetically-coated material under moving read-write heads

- Transfer rate - rate at which data flow between drive and computer.
- Positioning time (random-access time) is time to move disk arm to desired cylinder (seek time) and time for desired sector to rotate under the disk head (rotational latency)
- Head crash results from disk head making contact with the disk surface.

Moving-head Disk Mechanism



Rotate at 60 to 250 times per second

Hard Disk Drives

Platters range from .85" to 14"; commonly 3.5", 2.5", and 1.8"

Range from 30GB to 3TB per drive

Performance: Transfer Rate, theoretical 6 Gb/sec, effective 1Gb/sec

Seek time from 3ms to 12ms - 9ms common for desktop drives

Average seek time measured or calculated based on 1/3 of tracks

Latency based on spindle speed: 1/ (RPM/60) = 60/RPM

Average latency = 1/2 latency

Hard Disk Performance

Access Latency = average seek time + average latency

Average I/O time = access latency + (amount to transfer / transfer rate) + controller overhead

Question

Calculate average I/O time to transfer 4KB block on a 7200 RPM disk with a 5ms average seek time, 1Gb/sec transfer rate with a 0.1ms controller overhead.

The First Commercial Disk Drive

1956 IBM RAMDAC computer included the IBM Model 350 disk storage system

5M 7bit characters 50x24" platters

Access time = < 1 second

Nonvolatile Memory Devices

If disk-drive like, then called solid-state disks (SSDs)

Other forms include USB drives (thumb drive, flash drive), DRAM disk replacements, surface-mounted on motherboards, and main storage in devices like smartphones.

Can be more reliable than HDDs, but more expensive per MB.

Less capacity, but much faster.

Busses can be too slow -> connect directly to PCI for example.

No moving parts, so no seek time or rotational latency!

Nonvolatile Memory Devices

Read and written in "page" increments (think sector) but can't overwrite in place

Must first be erased, and erases happen in larger "block" increments

- Can only be erased/written a limited number of times before worn out $\sim$ 100,000
- Life span measured in drive writes per day (DWPD)

NAND Flash Controller Algorithms

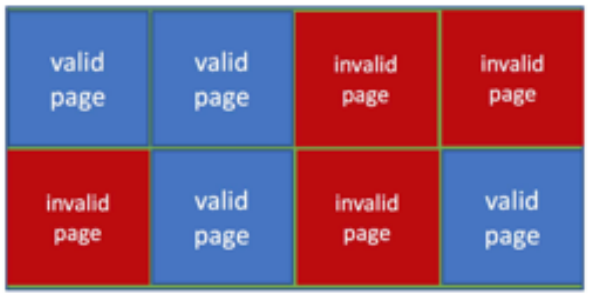
With no overwrite, pages end up with mix of valid and invalid data

To track which logical blocks are valid, controller maintains flash translation layer (FTL) table

Also implements garbage collection to free invalid page space

Allocates overprovisioning to provide working space for GC

Each cell has lifespan, so wear leveling needed to write equally to all cells



Volatile Memory

DRAM frequently used as mass-storage device

RAM drives (with many names, including RAM disks) present as raw block devices, commonly file system formatted

Computers have buffering, caching via RAM, so why RAM drives?

- Caches and buffers are allocated by programmer or operating system
- RAM drives allow user to place data in memory for temporary safekeeping using standard file operations.

Used as high speed temporary storage

Example: Create 1M RAM Disk on macOS

```
diskutil erasevolume HFS+ "RAMDisk" 'hdiutil
attach -nomount ram://2048'
```

Disk Attachment

A secondary storage device is attached to a computer by the system bus or an I/O bus.

Several kinds of buses are available

- advanced technology attachment (ATA)
- serial ATA (SATA) and eSATA
- serial attached SCSI (SAS)
- universal serial bus (USB)
- fibr channel (FC).

Address Mapping

Storage device are addressed as large 1-dimensional arrays of logical blocks, where the logical block is the smallest unit of transfer

The 1-dimensional array of logical blocks is mapped into the sectors of the disk sequentially

- Sector 0 is the first sector of the first track on the outermost cylinder
- Logical to physical address should be easy except for bad sectors

HDD Scheduling

One of the responsibilities of the operating system is to use the hardware efficiently - for the disk drives, this means having a fast access time and disk bandwidth.

Minimize seek time $\approx$ seek distance.

Disk bandwidth is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer.

HDD Scheduling

There are many sources of disk I/O request: OS, system processes, users processes

I/O request includes input or output mode, disk address, memory address, number of sectors to transfer

OS maintains queue of requests, per disk or device

Idle disk can immediately work on I/O request, busy disk means work must queue

- Optimization algorithms only make sense when a queue exists

HDD Scheduling

In the past, operating system responsible for queue management, disk drive head scheduling

- Now, built into the storage devices, controllers
- Just provide (Logical Block Addresses) LBAs, handle sorting of requests

Drive controllers have small buffers and can manage a queue of I/O requests (of varying depth)

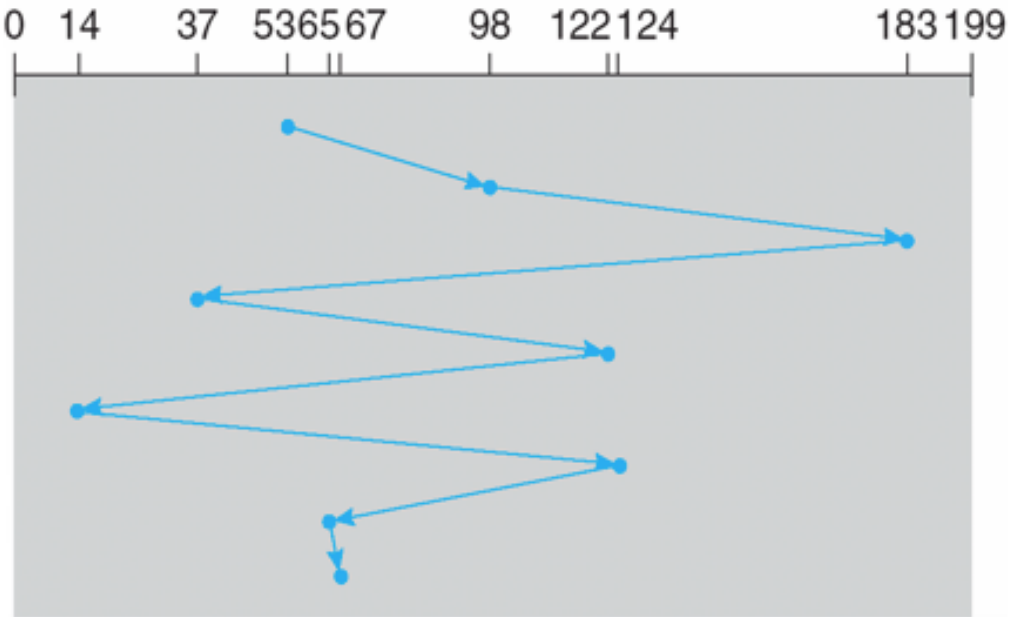
Several algorithms exist to schedule the servicing of disk I/O requests

FCFS

We illustrate scheduling algorithms with a request queue (0-199), total head movement of 640 cylinders

queue = `98, 183, 37, 122, 14, 124, 65, 67`

Head pointer `53`



SCAN

The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues.

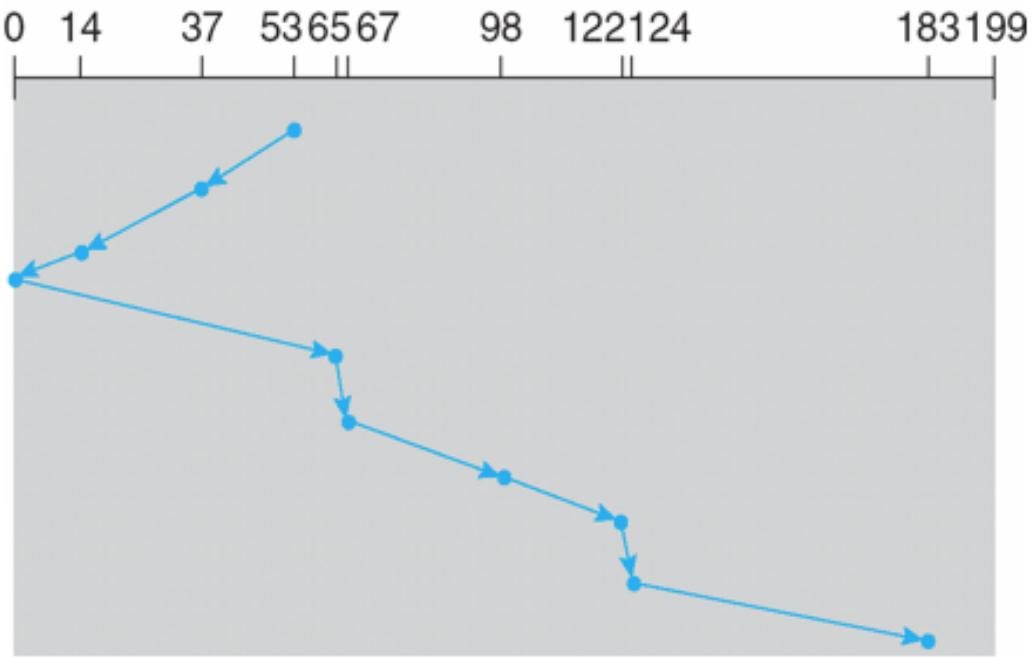
SCAN algorithm Sometimes called the elevator algorithm

Illustration shows total head movement of 208 cylinders. It is calculated assuming that the head does not need to go below position 14, so instead of going back to 0 and then to 65, it went from 14 to 65 and saved trips from 14-0 and back to 14, which is 28. If we deduct 28 from your calculation we get 236-28 = 208.

But note that if requests are uniformly dense, largest density at other end of disk and those wait the longest

SCAN (Cont.)

queue = `98, 183, 37, 122, 14, 124, 65, 67`
Head pointer `53`



C-SCAN

Provides a more uniform wait time than SCAN

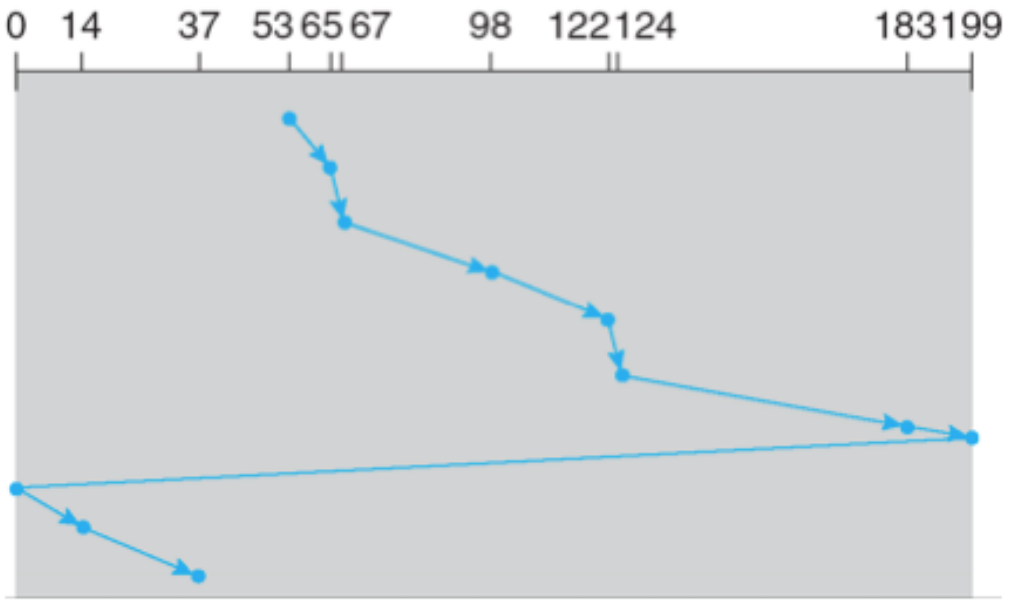
The head moves from one end of the disk to the other, servicing requests as it goes

- When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip.

Treats the cylinders as a circular list that wraps around from the last cylinder to the first one

C-SCAN (Cont.)

queue = `98, 183, 37, 122, 14, 124, 65, 67`
Head pointer `53`



Selecting a Disk-Scheduling Algorithm

Shortest Seek Time First (SSTF) is common and has a natural appeal

SCAN and C-SCAN perform better for systems that place a heavy load on the disk

- Less starvation, but still possible

To avoid starvation Linux implements deadline scheduler

- Maintains separate read and write queues, gives read priority
- Implements four queues: 2 x read and 2 x write

Linux Completely Fair Queueing scheduler (CFQ)

NVM Scheduling

No disk heads or rotational latency but still room for optimization

- NVM best at random I/O, HDD at sequential
- Throughput can be similar
- Input/Output operations per second (IOPS) much higher with NVM (hundreds of thousands vs hundreds)
- But write amplification (one write, causing garbage collection and many read/writes) can decrease the performance advantage

Error Detection and Correction

Fundamental aspect of many parts of computing (memory, networking, storage)

Error detection determines if there a problem has occurred (for example a bit flipping)

- If detected, can halt the operation
- Detection frequently done via parity bit

Parity one form of checksum - uses modular arithmetic to compute, store, compare values of fixed-length words

- Another error-detection method common in networking is cyclic redundancy check (CRC) which uses hash function to detect multiple-bit errors

Error-correction code (ECC) not only detects, but can correct some errors (i.e. Reed-Solomon error correction).

Storage Device Management

Low-level formatting, or physical formatting - Dividing a disk into sectors that the disk controller can read and write

- Each sector can hold header information, plus data, plus error correction code (ECC)
- Usually 512 bytes of data but can be selectable

To use a disk to hold files, the OS needs to record its own data structures on the disk

- Partition the disk into one or more groups of cylinders, each treated as a logical disk
- Logical formatting or "making a file system"
- To increase efficiency most file systems group blocks into clusters; Disk I/O done in blocks, File I/O done in clusters

Storage Device Management (cont.)

Root partition contains the OS, other partitions can hold other OSes, other file systems, or be raw

- Mounted at boot time
- Other partitions can mount automatically or manually

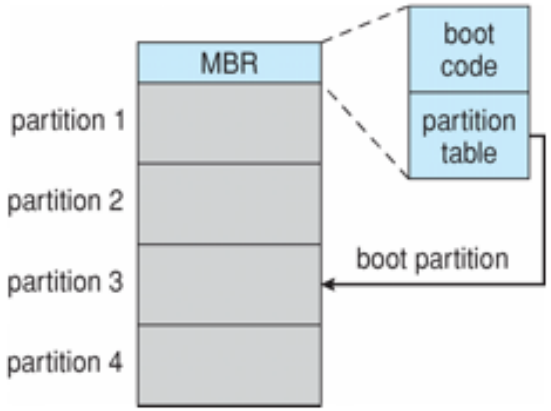
At mount time, file system consistency checked

- Is all metadata correct?
  - If not, fix it, try again
  - If yes, add to mount table, allow access

Device Storage Management (Cont.)

Raw disk access for apps that want to do their own block management, keep OS out of the way (databases for example)

Boot-block initializes sys.
- The bootstrap is stored in ROM, firmware
- Bootstrap loader program stored in boot blocks of boot partition



Booting from secondary storage in Windows

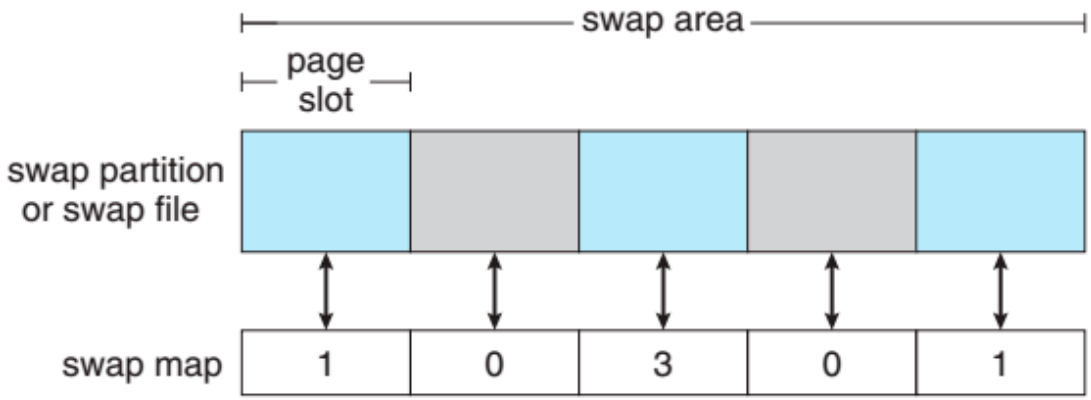
Once the system identifies the boot partition, it reads the first sector/page from that partition (called the boot sector), which directs it to the kernel.

Swap-Space Management

Used for moving entire processes (swapping), or pages (paging), from DRAM to secondary storage when DRAM not large enough for all processes

OS provides swap space management

- Usually multiple swap spaces possible - decreasing I/O load on any given device
- Best to have dedicated devices
- Can be in raw partition or a file within a file system
- Data structures for swapping on Linux systems - used only for anonymous memory.



Storage Attachment

Computers access storage in three ways

- Host-attached
- Network-attached
- Cloud

Host attached access through local I/O ports, using one of several technologies

- To attach many devices, use storage busses such as USB, firewire, thunderbolt
- High-end systems use fibre channel (FC)

Network-Attached Storage

Network-attached storage (NAS) is storage made available over a network rather than over a local connection (such as a bus)

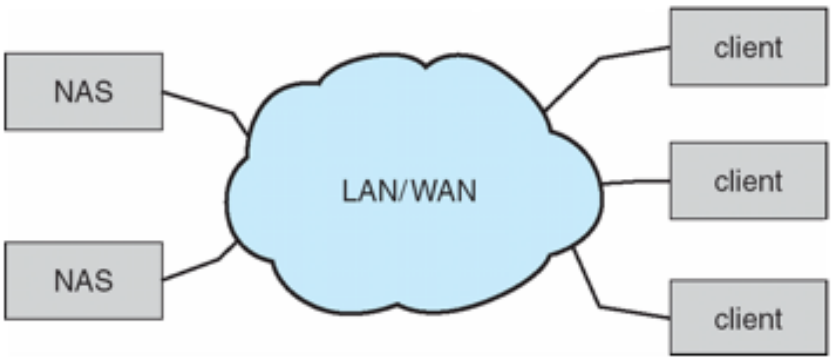
- Remotely attaching to file systems

Network File System (NFS) and Common Internet File System (CIFS) are common protocols

Implemented via remote procedure calls (RPCs) between host and storage over typically TCP or UDP on IP network

iSCSI protocol uses IP network to carry the SCSI protocol

- Remotely attaching to devices (blocks)



Cloud Storage

Similar to NAS, provides access to storage across a network

- Unlike NAS, accessed over the Internet or a WAN to remote data center

NAS presented as just another file system, while cloud storage is API based, with programs using the APIs to provide access

- Examples include Dropbox, Amazon S3, Microsoft OneDrive, Apple iCloud
- Use APIs because of latency and failure scenarios (NAS protocols wouldn't work well)

Storage Array

Can just attach disks, or arrays of disks

Avoids the NAS drawback of using network bandwidth

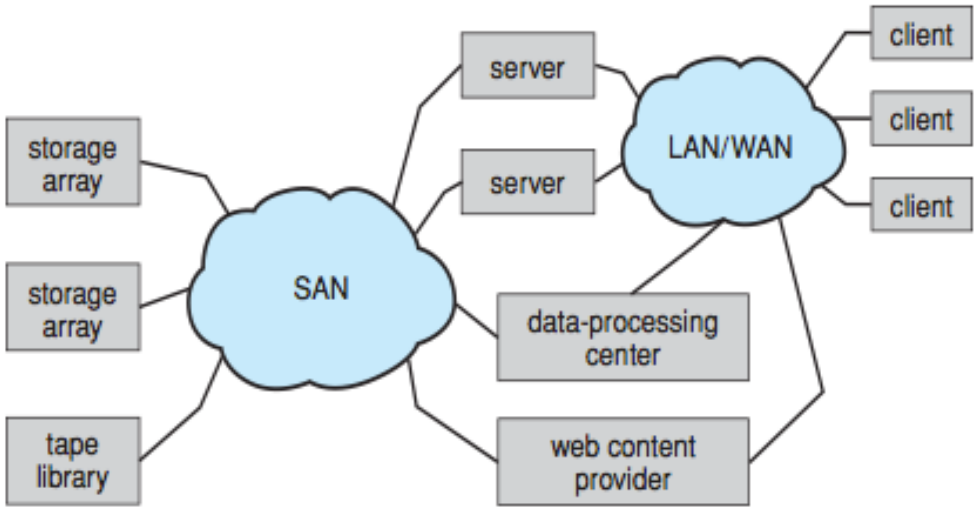
Storage Array has controller(s), provides features to attached host(s)

- Ports to connect hosts to array
- Memory, controlling software (sometimes NVRAM, etc)
- A few to thousands of disks
- RAID, hot spares, hot swap
- Shared storage $\rightarrow$ more efficiency
- Features found in some file systems

Storage Area Network (SAN)

Common in large storage environments

Multiple hosts attached to multiple storage arrays - flexible



Storage Area Network (Cont.)

SAN is one or more storage arrays. Connected to one or more Fibre Channel switches or InfiniBand (IB) network

Hosts also attach to the switches

Storage made available via Logical Unit Number (LUN) masking from specific arrays to specific servers

Easy to add or remove storage, add new host and allocate it storage

A Storage Array

RAID Structure

RAID - redundant array of inexpensive disks (multiple disk drives provides reliability via redundancy)

Increases the mean time to failure

Mean time to repair - exposure time when another failure could cause data loss

Mean time to data loss based on above factors

Combined with NVRAM to improve write performance

Several improvements in disk-use techniques involve the use of multiple disks working cooperatively

Example

No redundancy

MTBF of a single disk is 100,000 hours.
MTBF of some disk in an array of 100 is 100000h/100 = 1000h = 41.6 days

Mirroring

MTBF of a single drive is 100,000 hours
The mean time to repair is 10 hours.
Mean time to data loss is $(100,000\text{h})^2/(2 * 10\text{h}) = 500 * 10^6\text{h}$ = 57,000 years!

RAID (Cont.)

Disk striping uses a group of disks as one storage unit

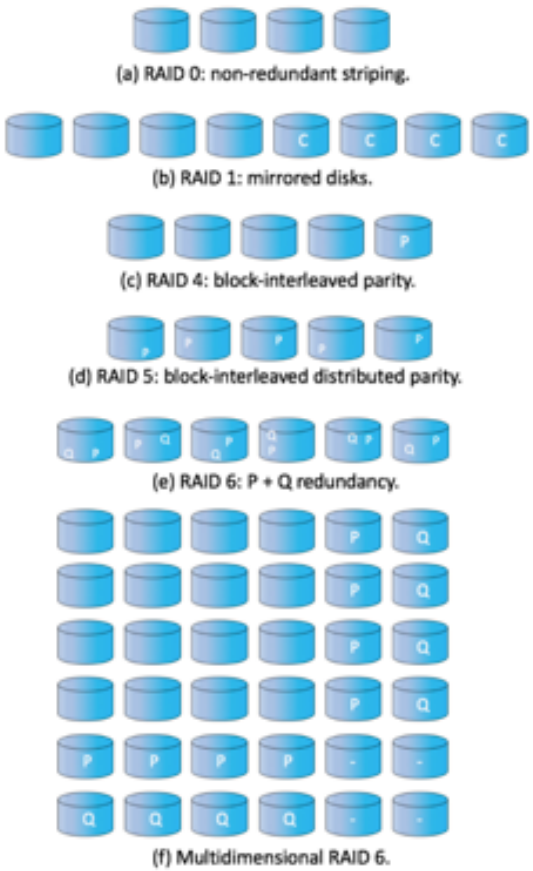
RAID is arranged into six different levels

RAID schemes improve performance and improve the reliability of the storage system by storing redundant data

- Mirroring or shadowing (RAID 1) keeps duplicate of each disk
- Striped mirrors (RAID 1+0) or mirrored stripes (RAID 0+1) provides high performance and high reliability
- Block interleaved parity (RAID 4, 5, 6) uses much less redundancy

RAID within a storage array can still fail if the array fails, so automatic replication of the data between arrays is common

RAID Levels



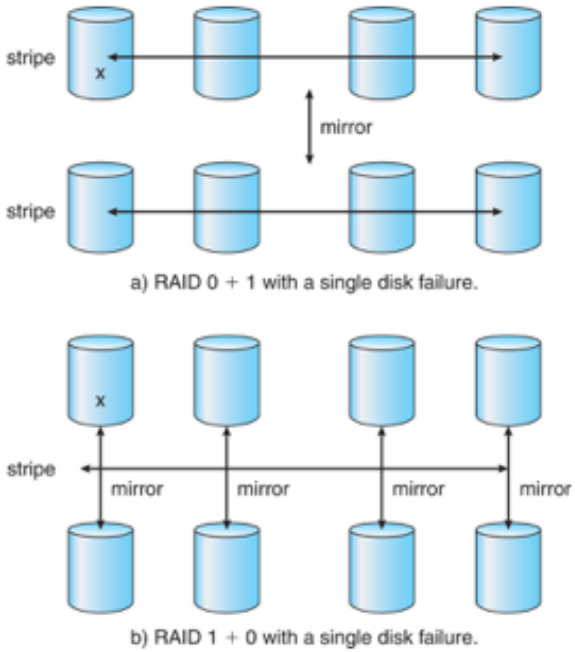
RAID levels

C - a second copy of the data

P - error-correcting bits (XOR parity)

Q - error-correcting bits calculated by Galois field math

RAID (0 + 1) and (1 + 0)



RAID 0 provides the performance, while RAID 1 provides the reliability.

RAID level 0 + 1 - a set of drives are striped, and then the stripe is mirrored to another, equivalent stripe.

RAID level 1 + 0 - drives are mirrored in pairs and then the resulting mirrored pairs are striped.

Other Features

Regardless of where RAID implemented, other useful features can be added

Snapshot is a view of file system before a set of changes take place (i.e. at a point in time)

Replication is automatic duplication of writes between separate sites

- For redundancy and disaster recovery
- Can be synchronous or asynchronous

Hot spare disk is unused, automatically used by RAID production if a disk fails to replace the failed disk and rebuild the RAID set if possible
- Decreases mean time to repair

Comparison

Observation

Compare the performance of write operations achieved by a RAID level 5 organization with that achieved by a RAID level 1 organization.

RAID level 1 organization can perform writes by simply issuing the writes to mirrored data concurrently.
RAID level 5, on the other hand, requires the old contents of the parity block to be read before it is updated based on the new contents of the target block.

This results in more overhead for the write operations on a RAID level 5 system.

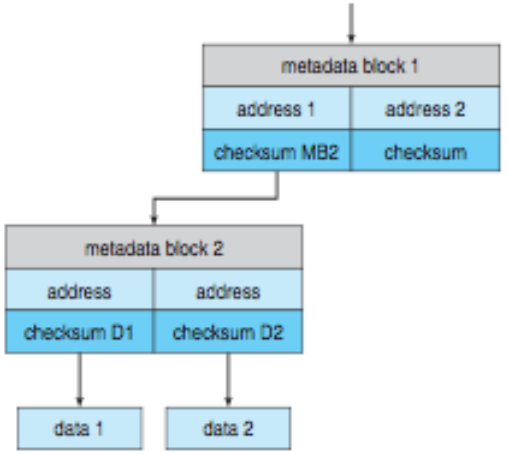
Extensions

RAID alone does not prevent or detect data corruption or other errors, just disk failures

Checksums kept with pointer to object, to detect if object is the right one and whether it is changed

Can detect and correct data and metadata corruption

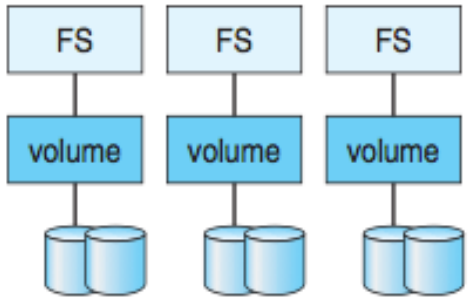
ZFS also removes volumes, partitions



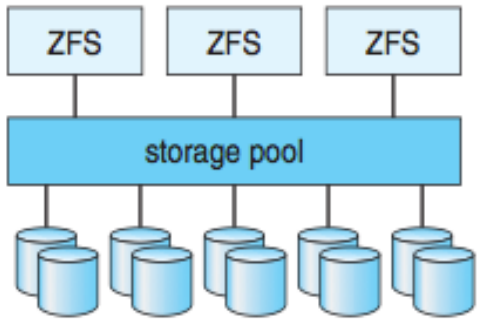
Solaris ZFS checksums all metadata and data

Traditional and Pooled Storage

ZFS combines file-system management and volume management into a unit providing greater functionality than the traditional separation of those functions allows.



(a) Traditional volumes and file systems.



(b) ZFS and pooled storage.

Object Storage

General-purpose computing, file systems not sufficient for very large scale

Another approach - start with a storage pool and place objects in it

- Object just a container of data
- No way to navigate the pool to find objects (no directory structures, few services
- Computer-oriented, not user-oriented

Object Storage (Cont.)

Typical sequence

- Create an object within the pool, receive an object ID
- Access object via that ID
- Delete object via that ID

Object storage management software like Hadoop file system (HDFS) and Ceph determine where to store objects, manages protection

- Typically by storing N copies, across N systems, in the object storage cluster
- Horizontally scalable
- Content addressable, unstructured

Object Stores

Google's Internet search contents

Dropbox contents

Spotify's songs

Facebook photos

Amazon AWS file systems and data objects

Thank you !

Operating Systems are among the most complex pieces of software ever developed !