Lecture.12.Storage.Management.txt

- Overview Of Mass Storage Structure
    - Bulk of secondary storage for modern computers is hard disk drives (HDDs) and non-volatile memory (NVM) devices
        - SSDs are non-volatile memory (NVM) devices
        - Hard disk drives (HDD) have mechanical parts that make them relatively slow
    - HDDs spin platters of magnetically coated material under moving read-write heads
    - The following parameters help us classify hard disk drives:
        - Transfer rate
            - Rate at which data flows between drive and computer
        - Positioning time
            - Also known as random access time
            - Is the time to move disk arm to desired cylinder
                - This is the seek time
            - And is the time for desired sector to rotate under the disk head
                - This is rotational latency
        - Head crash results from disk head making contact with the disk surface
            - Rarely happens in modern hard drives

- Moving Head Disk Mechanism
    - The arm assembly contains the read-write head
    - Platters are rotating around the spindle
        - In one direction
        - Rotates at 60 to 250 times per second
    - Each platter has a number of tracks
    - The vertical cross section of the tracks create a cylinder
        - Each cylinder has a section in which we can read and write

- Hard Disk Drives
    - Platters range from .85" to 14"
        - The common sizes are: 3.5", 2.5", & 1.8"
    - Capacity ranges from 30GB to 3TB per drive
    - Performance
        - Theoretical transfer rate: 6 Gb/sec
            - Gb = Gigabits
        - Effective transfer rate: 1 Gb/sec
            - Gb = Gigabits
    - Seek time ranges from 3ms to 12ms
        - On average, 9ms is common for desktop drives
    - Average seek time is 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
    - Answer:
        Controller overhead = 0.1ms
        Average seek time = 5ms
        Average latency = (latency / 2)
                          = ((60/7200) / 2)
                          = 4.17ms
        Amount to transfer
        ------------------ = (4 KB / (1 Gb/sec))
          Transfer rate
                           = (4 * 1024 * 8) bits / 1000000000 bits/sec
                           = 0.032ms
        Average IO time = 5ms + 4.17ms + 0.1ms + 0.032ms
                        = 9.302ms
    - Note: 'B' is for bytes, and 'b' is for bits

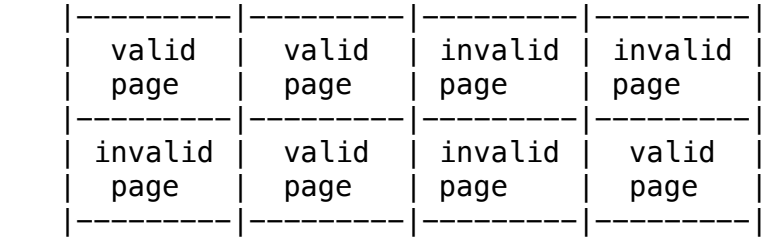
- The First Commercial Disk Drive
    - 1956 IBM RAMDAC computer included the IBM Model 350 disk storage
      system
    - The storage was only 5MB
    - Only supported 7-bit characters
    - 50x24" platters
    - Access time =< 1 second

- Nonvolatile Memory Devices (1)
    - If disk drive is a non-volatile memory device, then it is 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
    - Usually has less capacity, but much faster
        - However, modern SSDs are catching up to HDDs in storage size
    - Busses can be too slow -> Connect directly to PCI for example
    - No moving parts; so no seek time or rotational latency

- Nonvolatile Memory Devices (2)
    - 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

            ~ 100,000
        - Life span measured in drive writes per day (DWPD)
            - Hard disks do not have a DWPD
            - DWPD only applies to non-volatile storage devices
    - Even though SSDs are better than HDDs, they have their drawbacks
        - i.e. SSDs have limitations related to writing cycles
            - However, this is continually improving

- NAND Flash Controller Algorithms
    - With no overwrite, pages end up with mix of valid and invalid data
        - i.e. Figure of Pages/Blocks



    - 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 garbage collector
    - Each cell has lifespan, so wear levelling needed to write equally to all cells
        - If you read/write to particular cells, their lifespan will be lower than other cells
        - All cells should be equally written to

- Volatile Memory
    - Has quick read and write times
        - But the data is lost when the power is cut
            - In other words, volatile memory is not persistent
    - There is DRAM and SRAM
        - D = Dynamic
        - S = Static
    - DRAM frequently used as mass storage device
    - RAM drives present as raw blovk devices, commonly file system formatted
        - Also referred to as RAM disks
    - Computers have bufferring, 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
        - Useful for performing operations on a large database
    - 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)
    - Fibre channel (FC)
  - Secondary storage is becoming more popular via cloud and network

- Address Mapping
  - Storage device are addressed as large 1-dimensional arrays of logical blocks
    - 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
      - The remaining sectors are mapped sequentially
    - Logical to physical address should be easy except for bad sectors
      - Bad sectors need to be mapped, and avoided during read-write cycles

- HDD Scheduling (1)
  - The scheduling of the hard disk drive should be done in a way that minimizes seek time
    - The seek distance between a location on a cylinder where data is located should be as short as possible
  - 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 ~ 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 (2)
  - There are many sources of disk I/O request:
    - Operating system
    - System processes
    - User processes
  - I/O request includes input or output mode, disk address, memory address, number of sectors to transfer, etc.
  - OS maintains queue of requests, per disk or device
    - The queue needs to be optimized to increase throughput and

    decrease seek time
  - 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 (3)
  - In the past, the operating system was responsible for queue management, disk drive head scheduling
    - Now, queue management is built into the storage devices' controller
      - Just provide logical block addresses (LBAs), and the controller handles 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
  - The FCFS algorithm goes in order of the queue
    - The first element is the first one to be served, and the last element is the last one to be served
    - This is not the most optimal way of servicing requests
  - 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
  - Order of service: 53 -> 98 -> 183 -> 37 -> 122 -> 14 -> 124 -> 65 -> 67
  - Total head movement = abs(53 - 98) + abs(98 - 183) + abs(183 - 37) + abs(37 - 122) + abs(122 - 14) + abs(14 - 124) + abs(124 - 65) + abs(65 - 67)
    = 640

- Scan (1)
  - The disk arm starts at one end of the disk, and move 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
    - In other words, move to one end and service all requests that are in the path. Then, move to the other end and service all requests that are in the path
  - SCAN algorithm is 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 to 0, and back to 14, which is 28
      - If we deduct 28 from the calculation we get:
        236 - 28 = 208

    - Note: If requests are uniformly dense, largest density at other end of disk, and those wait the longest
      - If most of the requests are on one side of the disk, then the requests at the other end of the disk will wait a long time to be serviced
        - Kind of like starvation

- Scan (2)
    - queue = 98, 183, 37, 122, 14, 124, 65, 67
      - Divide queue into two groups
        - First group contains all elements that are less than the head pointer
        - Second group contains all elements that are greater than the head pointer
      - Service all elements lower than head pointer before servicing elements that are greater than the head pointer
      - Order of service: 53 -> 37 -> 14 -> 65 -> 67 -> 98 -> 122 -> 124 -> 183
    - Head pointer = 53
    - Total head movement = abs(53 - 37) + abs(37 - 14) + abs(14 - 65) + abs(65 - 67) + abs(67 - 98) + abs(98 - 122) + abs(122 - 124) + abs(124 - 183) = 208
    - In this example, SCAN is almost 3 times as better than FCFS

- C:Scan (1)
    - 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, 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 (2)
    - queue = 98, 183, 37, 122, 14, 124, 65, 67
      - First, all requests greater than the head pointer are serviced. Then, head pointer resets to 0, and services all remaining requests from 0 to the head pointer
      - Order of service: 53 -> 65 -> 67 -> 98 -> 122 -> 124 -> 183 -> (199 -> 0)* -> 14 -> 37
        - Note: There is no service request at 199 or 0, but those positions are crossed when the pointer moves from 183 to 14
    - Head pointer = 53
    - Total head movement = abs(53 - 65) + abs(65 - 67) + abs(67 - 98) + abs(98 - 122) + abs(122 - 124) + abs(124 - 183) +

```
                          abs(183 - 199) + abs(0 - 14 + 1) +
                          abs(14 - 37)
                        = 182
```

- SCAN and C-SCAN perform much better than FCFS

- Selecting A Disk Scheduling Algorithm
  - Shortest seek time first (SSTF) is common and has a natural appeal
    - Logic is similar to shortest job first (SJF) process scheduling algorithm
    - Average serving time is lower when we service the shortest requests first
  - 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 4 queues; 2 for read & 2 for write
  - Linux completely fair queuing scheduler (CFQ)
  - There is dedicated software for handling deadlines for the disk scheduling or accessing sectors on the disk

- NVM Scheduling
  - No disk heads or rotational latency but still room for optimization
    - NVM best at random I/O
      - HDD is best at sequential I/O
    - Throughput, between SSD and HDD, can be similar
    - Input/Output operations per second (IOPS) much higher for NVM
      - Hundreds of thousands VS. Hundreds
    - But write amplification (one write, causing garbage collection and many read/writes) can decrease the performance advantage
  - NVM storage devices have improved a lot since they were first released
    - i.e. Erasing is now done on the fly

- Error Detection & Correction
  - For each sector on the hard drive, we compute a checksum that helps us detect error on data
    - Checksum can detect if data content has changed due to corruption or other reasons
  - Fundamental aspect of many parts of computing
    - i.e. Memory, networking, storage, etc.
  - Error detection determines if a problem has occurred
    - i.e. A bit flipping
      - If detected, can halt the operation
      - Detection frequently done via parity bit

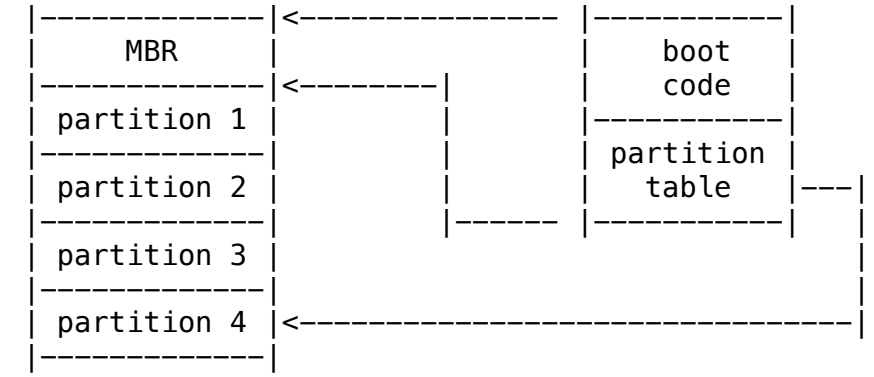
    - Parity is one form of checksum
        - Uses modular arithmetic to compute, store, compare values of fixed-length words
        - Parity checking has some limitations
            - i.e. If there is an error on two bits, it may or may not be detected
                - In other words, parity bit-checking does not detect errors all the time
        - Checksum is better than parity bit-checking
    - Another error detection method common in networking is cyclic redundancy check (CRC)
        - 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
        - Limitations of ECC:
            - Can only correct up to a certain percentage of errors
            - Requires extra storage space to work
    - These algorithms are very effective and used across the computer science field to verify data integrity
        - i.e. Telecommunications, networking, etc.

- Storage Device Management (1)
    - Low level formatting, or physical formatting
        - Divide 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
        - It's important to have the data structure on the same 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 (2)
    - Root partition contains the operating system
        - Other partitions can hold:
            - Other operating systems
            - Other file systems
            - Be raw
                - i.e. Contain nothing (or are empty)
            - Misc. data
    - There is only one master boot record (MBR)
        - The MBR is used to mount other operating systems

    - The root partition is mounted at boot time
        - Other partitions can mount automatically or manually
    - At mount time, file system consistency checked
        - i.e. Is all metadata correct?
            - If not, fix it, and try again
            - If yes, add to mount table, and allow access

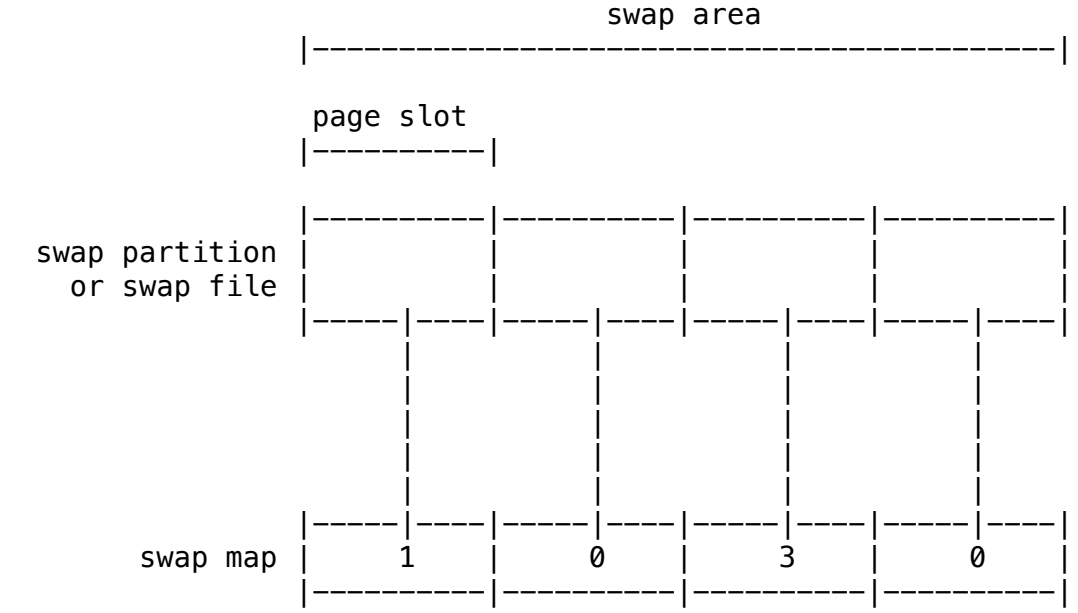
- Storage Device Management (3)
    - Raw disk access for apps that want to do their own block management, keep OS out of the way
        - i.e. Databases
    - Boot-block initializes system
        - The bootstrap is stored in ROM, firmware
        - Bootstrap loader program stored in boot blocks of boot partition
    - 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
    - i.e. Diagram of Partitions



            - Booting from secondary storage in Windows
            - MBR = Master Boot Record
                - The MBR is usually written in ROM
                    - ROM = Read-Only Memory
                    - The MBR cannot be modified/updated in any way
                - Contains boot code and partition table
                    - Tells you which partition to go to, to boot the system/OS

- 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
        - Swaps blocks of memory from main memory to hard disk drive, and vice versa
    - OS provides swap space management
        - Usually multiple swap spaces possible
            - Decreases I/O load on any given device
            - The OS keeps track of which page slot is used, and by how many processes

        - 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
                - Anonymous memory is data that is produced by a process during its operation
                    - Must be stored on a hard disk (HDD), to be able to continue a process where it was left off
                - Anonymous memory is stored in an anonymous page
            - If there is already something in the hard disk (HDD), it is not swapped
    - Ideally, there should be as less swapping as possible
        - This is because swapping is time consuming
            - Interfacing with the hard disk (HDD) is a bottle-neck
    - i.e. Diagram

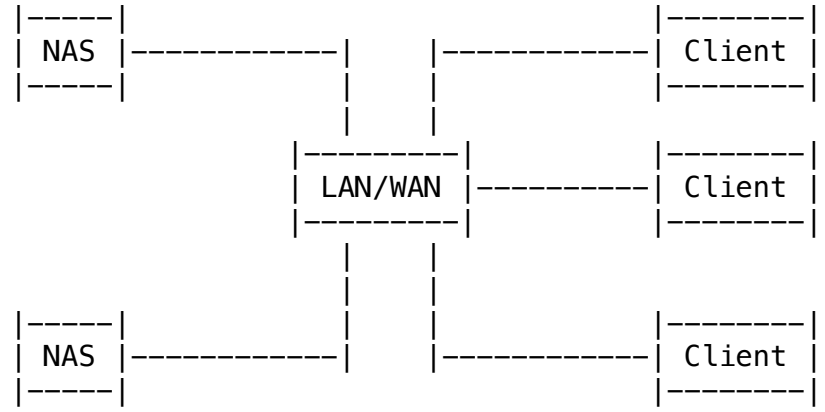


        - The numbers in the swap map indicate how many processes are using the corresponding page slot
            - i.e. The 3rd page slot is being used by 3 different processes
            - i.e. The 1st page slot is only being used by 1 process
            - 2nd and 4th page slots are empty because the swap maps are 0

- Storage Attachment
    - Computers access storage in 3 ways:
        - Host attached
            - Device is directly attached to the computer
        - Network attached
            - Device is attached to LAN/WAN
        - Cloud
            - Need internet to access cloud
            - Gaining more popularity and traction

            - Especially with data centers
    - 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, etc.
        - High end systems use fibre channel (FC)

- Network Attached Storage
    - Network attached storage (NAS) is a 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
    - i.e. Diagram of NAS



- 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
        - i.e. Dropbox, Amazon S3, Microsoft's OneDrive, Apple's iCloud
        - Use APIs because of latency and failure scenarios
            - NAS protocols would not work well

- Storage Array
    - Can just attach disks, or array of disks
        - Avoids the NAS drawback of using network bandwidth
    - Storage array has controller(s)
        - Provides features attached to host(s), such as:
            - Ports to connect hosts to array
            - Memory controlling software
                - Sometimes NVRAM, etc.

        - A few to thousands of disks
        - RAID, hot spares, hot swaps
        - Shared storage -> More efficiency
        - Features found in some file systems

- Storage Area Network (SAN) (1)
    - Common in large storage environments
    - Multiple hosts attached to multiple storage arrays
        - Very flexible
    - Storage arrays are accessed through the storage area network (SAN)
        - i.e. Client -> LAN/WAN -> Server -> SAN -> Storage array
        - i.e. Client -> LAN/WAN -> Data processing center -> SAN -> Storage array

- Storage Area Network (SAN) (2)
    - 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
        - i.e. Add new host and allocate it storage

- RAID Structure
    - RAID
        - Stands for Redundant Array of Inexpensive Disks
            - Multiple disk drives provide reliability via redundancy
    - Increases the mean time to failure
        - Makes things more reliable
    - Quality of RAID structure is measured via:
        - 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

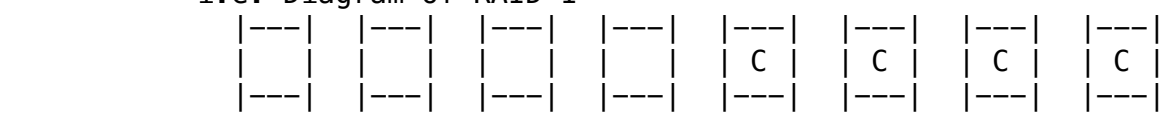
- Example
    - No redundancy
        - MTBF of a single disk is 100,000 hours
        - MTBF of some disk in an array of 100 is:
          (100000 hours / 100 drives) = 1000 hours
                                      = 41.6 days
    - Mirroring
        - MTBF of a single drive is 100,000 hours
        - The mean time to repair is 10 hours
        - Mean time to data loss is:

((100,000 hours) ^ 2) / (2 * 10 hours) =
(500 * (10^6)) hours =
57,000 hours
- It's almost impossible to lose data in a mirrored structure
  - However, this is mean time; data loss can occur sooner

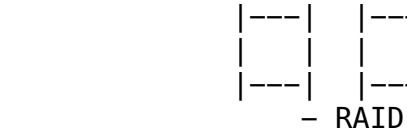
- RAID
  - Disk striping uses a group of disks as one storage unit
  - RAID is arranged into 6 different levels
  - RAID schemes improve performance and improve the reliability of the storage system by storing redundant data
    - Mirroring or shadowing (RAID 1) keeps duplicate of each disk
    - Striped 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
  - i.e. Diagram of RAID 0

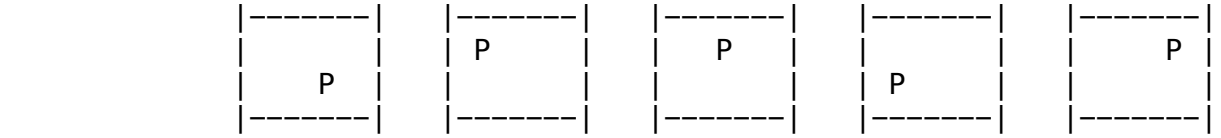
      - RAID 0: Non-redundant striping
  - i.e. Diagram of RAID 1



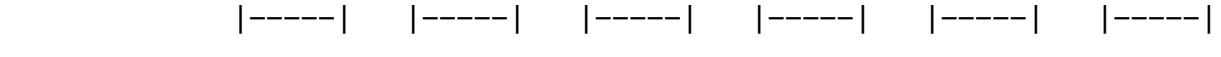
      - RAID 1: Mirrored disks
        - There are 4 disks and 4 copies
  - i.e. Diagram of RAID 4

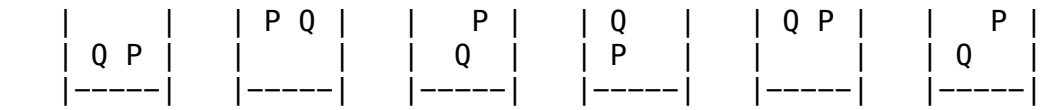


      - RAID 4: Block-interleaved parity
        - The 5th disk gives parity of all 4 disks
  - i.e. Diagram of RAID 5



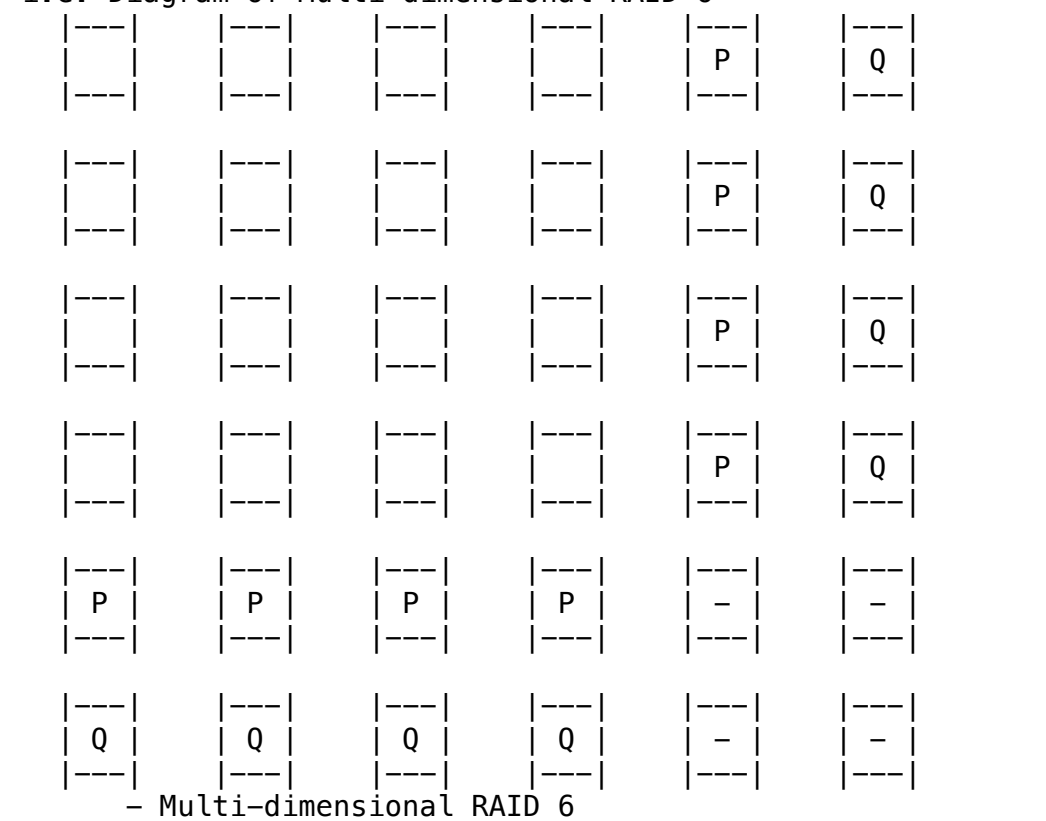
      - RAID 5: Block-interleaved distributed parity
        - Each disk has its own parity covered
  - i.e. Diagram of RAID 6





- RAID 6: P + Q redundancy

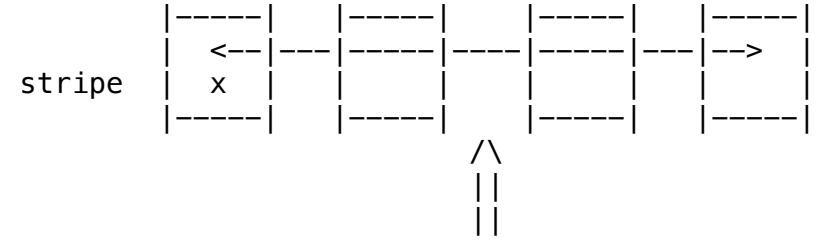
- i.e. Diagram of Multi-dimensional RAID 6

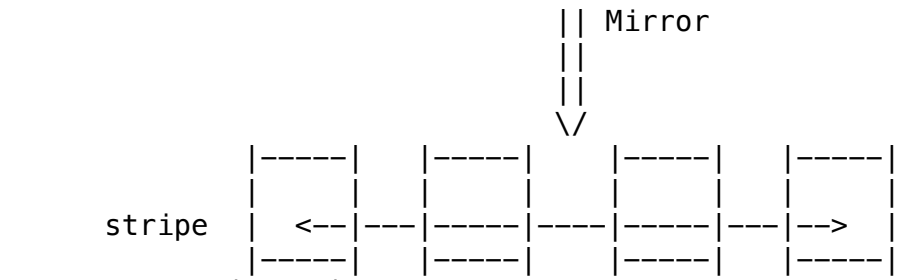


- Multi-dimensional RAID 6
  - Lots of disks are dedicated for error correcting bits

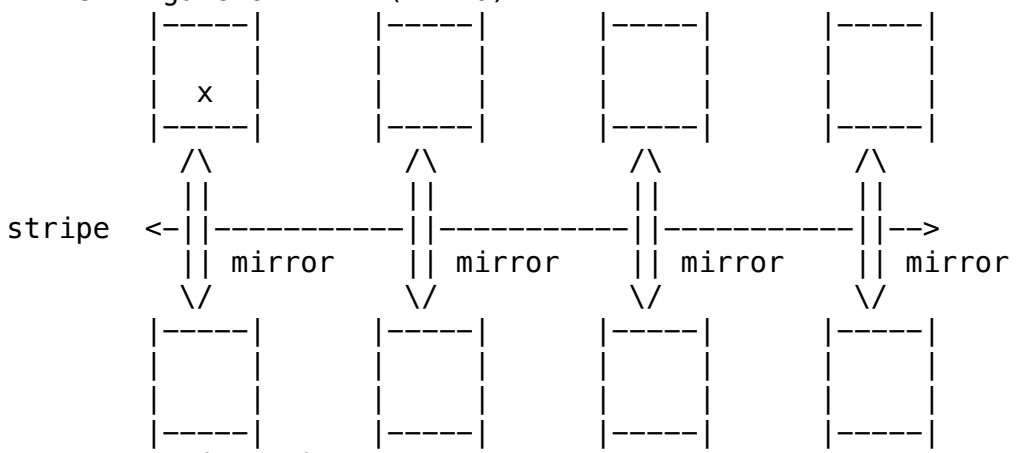
- By combining number of inexpensive disks we can increase reliability via copying disks or storing error-correcting bits

- RAID (0 + 1) & (1 + 0)
  - RAID 0 provides the performance, while RAID 1 provides the reliability
    - RAID 0 is faster, but RAID 1 is better in the event of a failure
  - RAID level 0 + 1
    - A set of drives are striped, and then the stripe is mirrored to another equivalent stripe
    - i.e. Figure of RAID (0 + 1)





- RAID (0 + 1) with a single disk failure
    - This is a vertical mirror
- RAID level 1 + 0
    - Drives are mirrored in pairs, and then the resulting mirrored pairs are striped
    - i.e. Figure of RAID (1 + 0)



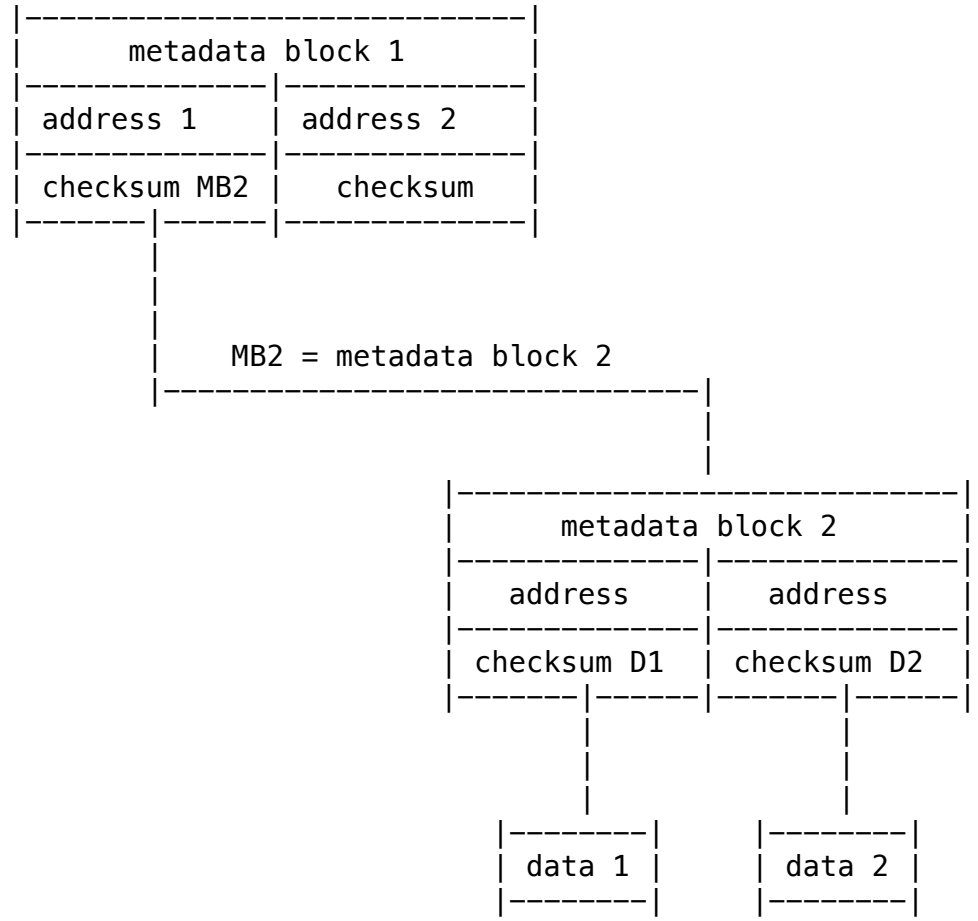
    - RAID (1 + 0) with a single disk failure
        - Mirrors horizontally, one by one

- Other Features
    - Regardless of where RAID is implemented, other useful features can be added
    - Snapshot is a view of file system before a set of changes take place
        - i.e. Snapshot of file system when OS is freshly installed
    - Replication is automatic duplication of writes between separate sites
        - For redundancy and disaster recovery
        - Can be synchronous or asynchronous
        - Increases reliability
    - 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

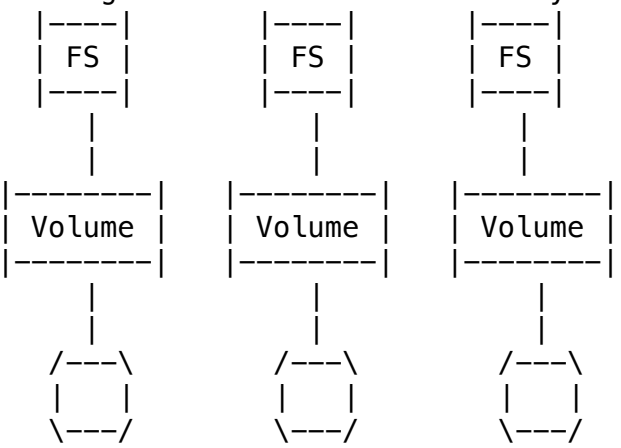
- Extensions
  - RAID alone does not prevent or detect data corruption or other errors, just disk failures
  - Checksums kept with pointer to object, to detect if object is the right one and whether it is changed
  - Can detect and correct data and metadata corruption
    - Metadata includes all possible data
  - ZFS also removes volumes, partitions
  - i.e. Figure of Solaris ZFS



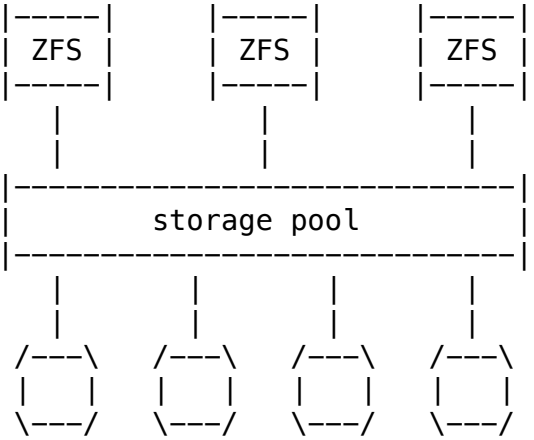
  - Solaris ZFS checksums all metadata and data

- Traditional & Pooled Storage
  - ZFS combines file system management and volume management into a unit providing greater functionality than the traditional

  separation of those functions allows
- i.e. Diagram of Traditional File Systems



  - Traditional volumes and file systems (FS)
- i.e. Diagram of ZFS



  - ZFS and pooled storage

- Object Storage (1)
  - 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 (2)
  - 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, and manage protection

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

- Object Stores
    - Examples of object stores are:
        - Google’s Internet search contents
        - Dropbox contents
        - Spotify’s songs
        - Facebook photos
        - Amazon AWS file systems and data objects

- Questions Answered
    - How does a snapshot differ from a backup?
        - They are very similar
        - A backup is frequently taken
            - i.e. Once a week, month, etc.
        - A snapshot is only done when it is needed
            - i.e. Right after a fresh install of the OS
            - i.e. After installing all essential software

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