OPERATING SYSTEMS

Storage Management - 5

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

Storage Management - 5: Mass Storage Structure

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

Course Syllabus - Unit 3

Unit 4: Storage Management

Mass-Storage Structur - Mass-Storage overview, Disk Scheduling, Swap-Space Management, RAID structure. File System Interface - file organization/structure and access methods, directories, sharing File System Implementation/Internals: File control Block (inode), partitions & mounting, Allocation methods.

Case Study: Linux/Windows File Systems

OPERATING SYSTEMS

Course Outline

37	Mass-Storage Structure: Mass-Storage overview	12.1	82.1
38	Disk Scheduling – FCFS, SSTF, SCAN, C-SCAN, LOOK	12.4	
39	Swap-Space Management, RAID Structure	12.6,12.7	
40	File Concept, File Structure, Access Methods	10.1-10.2	
41	Directory and Disk Structure	10.3	
42	File-System Mounting, File Sharing, Protecting	10.4-10.6	
43	Implementing File-Systems: File control Block (inode), partitions & mounting	11.1,11.2	
44	Disk Space Allocation methods: Contiguous, Linked, Indexed	11.4	
45	Case Study: Unix/Linux File systems	11.8	
46	NFS	16.7	

Topic Outline

- Swap-Space Management
- Data Structures for Swapping on Linux Systems
- RAID Structure
- RAID Levels

Topic Outline

- RAID - Other Features
- RAID Extensions
- ZFS Checksums All Metadata and Data
- Traditional and Pooled Storage

Swap-Space Management

- Swapping is that setting which occurs when the amount of physical memory reaches a critically low point and processes are moved from memory to swap space to free available memory.

- In practice, very few modern operating systems implement swapping in this fashion. Rather, systems now combine swapping with virtual memory techniques and swap pages, not necessarily entire processes.

- In fact, some systems now use the terms “swapping” and “paging” interchangeably, reflecting the merging of these two concepts.

- Swap-space management is another low-level task of the operating system. Virtual memory uses disk space as an extension of main memory. Since disk access is much slower than memory access, using swap space significantly decreases system performance.

- The main goal for the design and implementation of swap space is to provide the best throughput for the virtual memory system. In this section, we discuss how swap space is used, where swap space is located on disk, and how swap space is managed.

Swap-Space Management

- Swap-space — Virtual memory uses disk space as an extension of main memory
- Less common now due to increase in memory capacity
- Swap-space can be carved out of the normal file system, or, more commonly, it can be in a separate disk partition (raw)
- Swap-space management
  - 4.3BSD allocates swap space when process starts; holds text segment (the program) and data segment
  - Kernel uses swap maps to track swap-space use

Swap-Space Use

- Swap space is used in various ways by different operating systems, depending on the memory-management algorithms in use.
- For instance, systems that implement swapping may use swap space to hold an entire process image, including the code and data segments
- Paging systems may simply store pages that have been pushed out of main memory.
- The amount of swap space needed on a system can therefore vary from a few megabytes of disk space to gigabytes, depending on the amount of physical memory, the amount of virtual memory it is backing, and the way in which the virtual memory is used.

Swap-Space Use

- Note that it may be safer to overestimate than to underestimate the amount of swap space required, because if a system runs out of swap space it may be forced to abort processes or may crash entirely.

- Overestimation wastes disk space that could otherwise be used for files, but it does no other harm.

Swap-Space Location

- A swap space can reside in one of two places: it can be carved out of the normal file system, or it can be in a separate disk partition.
- If the swap space is simply a large file within the file system, normal file-system routines can be used to create it, name it, and allocate its space. This approach, though easy to implement, is inefficient.
- Navigating the directory structure and the disk allocation data structures takes time and (possibly) extra disk accesses.
- External fragmentation can greatly increase swapping times by forcing multiple seeks during reading or writing of a process image.
- We can improve performance by caching the block location information in physical memory and by using special tools to allocate physically contiguous blocks for the swap file, but the cost of traversing the file-system data structures remains

Swap-Space Location

- Alternatively, swap space can be created in a separate raw partition.
- No file system or directory structure is placed in this space. Rather, a separate swap-space storage manager is used to allocate and deallocate the blocks from the raw partition.
- This manager uses algorithms optimized for speed rather than for storage efficiency, because swap space is accessed much more frequently than file systems (when it is used).
- Internal fragmentation may increase, but this trade-off is acceptable because the life of data in the swap space generally is much shorter than that of files in the file system. Since swap space is reinitialized at boot time, any fragmentation is short-lived.
- The raw-partition approach creates a fixed amount of swap space during disk partitioning. Adding more swap space requires either re-partitioning the disk (which involves moving the other file-system partitions or destroying them and restoring them from backup) or adding another swap space elsewhere.

Swap-Space Location

- Some operating systems are flexible and can swap both in raw partitions and in file-system space.

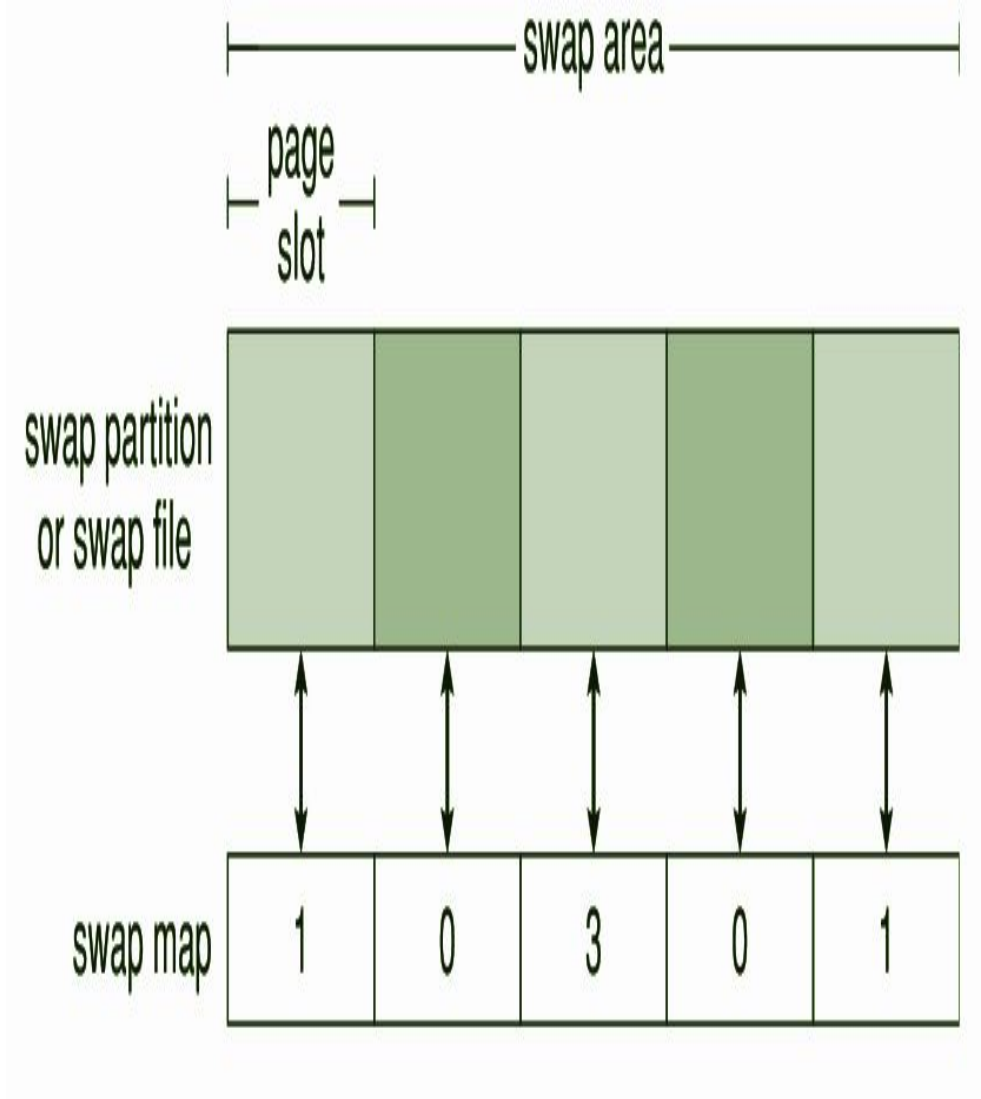
- Linux is an example: the policy and implementation are separate, allowing the machine's administrator to decide which type of swapping to use.

- The trade-off is between the convenience of allocation and management in the file system and the performance of swapping in raw partitions

Swap-Space Management

- Solaris 2 allocates swap space only when a dirty page is forced out of physical memory, not when the virtual memory page is first created
    - File data written to swap space until write to file system requested
    - Other dirty pages go to swap space due to no other home
    - Text segment pages thrown out and reread from the file system as needed
- What if a system runs out of swap space ?
- Some systems allow multiple swap spaces

Data Structures for Swapping on Linux Systems



- Associated with each swap area is a swap map—an array of integer counters, each corresponding to a page slot in the swap area.
- If the value of a counter is 0, the corresponding page slot is available.
- Values greater than 0 indicate that the page slot is occupied by a swapped page.
- The value of the counter indicates the number of mappings to the swapped page.
- For example, a value of 3 indicates that the swapped page is mapped to three different processes (which can occur if the swapped page is storing a region of memory shared by three processes).

Redundant Array of Inexpensive Disks (RAID) - About

- It can also be called appropriately as **R**edundant **A**rray of **I**ndependent **D**isks

- 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

Redundant Array of Inexpensive Disks (RAID) - About

To create an optimal cost-effective RAID configuration, we need to simultaneously achieve the following goals:

- Maximize the number of disks being accessed in parallel.
- Minimize the amount of disk space being used for redundant data.
- Minimize the overhead required to achieve the above goals.

Redundant Array of Inexpensive Disks (RAID) - About

- Redundancy – Redundancy refers to having numerous components which offer the same function, so that the system functioning can continue in the event of partial system failure.

- Fault Tolerance – Fault Tolerance means, in the event of a component failure, the system is designed in such a way that a backup component will be available, in order to prevent loss of service.

- The distribution of data is done in two ways, which are two generalized concepts.

- **Mirroring** is one of the ways, which offers replication of data on another disk and the second concept is **striping**, where splitting of reproduced data takes place across the available disk drives.

- **Parity** is a concept, which is also streamlined into RAID technology as another way of storage method. It involves saving of information across the disk arrays, so that, the same information can be used to recreate or reconstruct the affected data, which is otherwise filled with errors or data loss, when disk drive fails.

Redundant Array of Inexpensive Disks (RAID) - About

- Redundant Array of Independent/Inexpensive Disks (RAID) is a technology that allows storing data across multiple hard drives.
- The purpose of RAID is to achieve data redundancy to reduce data loss and, in a lot of cases, improve performance. The best way to get in on the RAID action is with a Network Attached Storage - NAS
- RAID was created in 1988 in order to deal with costs of high-performing disk drives.
- The inventors argued that an array of inexpensive disks could outperform a single expensive disk, which was no doubt a huge problem when 10MB of storage cost upwards of $100.
- RAID allows for data to be stored on multiple disk drives at once, though a RAID setup will appear to, say, Windows 10, as a single disk.

Redundant Array of Inexpensive Disks (RAID) - About

- A RAID controller is the thing that directs data in and out of storage drives that can be based on either software or hardware.

- In the software case, if you have a few drives set up in a RAID array inside your home PC, it's likely handled on a software level with your PC's processor (CPU) handling the work.

- In the hardware case, say with an external Network Attached Storage (NAS) enclosure, there will be a dedicated controller card to deal with input and output (I/O) operations of the RAID array.

Redundant Array of Inexpensive Disks (RAID) - About

- RAID 0 was the first version created, and it provides users with fast read and write speeds for improved performance.
- Despite its name, there is no data redundancy.
- Data is striped across drives, meaning each drive holds a piece of the overall information.
- Striping means quicker data access, but if one drive fails they will all fail, resulting in a loss of data.
- Speeds are measured by the number of drives in the RAID 0 array, so consider a RAID 0 array with four drives to be four times faster than a single drive.
- Instead of having a one-lane highway for all data to travel, there are now multiple lanes, all shipping data back and forth.

Redundant Array of Inexpensive Disks (RAID) - About

- The performance gains are offset by the lack of data redundancy and the risk of losing all drives in the event of one failing.

- Other than RAID 0, you'll also often hear of RAID 1, RAID 5, RAID 6, and RAID 10 setups.

Redundant Array of Inexpensive Disks (RAID) - About

- A RAID 1 setup consists of at least two drives that are mirrored to contain the exact same information.
- This RAID 1 setup includes fault tolerance, as one drive failing will not result in the other drives failing as well.
- There's no striping involved, so as long as one drive works, the array will continue to operate, making this a favorite of those who require high reliability.
- In most cases, read performance should be about the same as with a single disk, though there can be a detrimental effect to write speed and storage capacity
- When data is written to the array, it must be written to each drive independently.
- Write speeds will therefore only be as fast as the slowest drive in the array.
- Likewise, storage capacity is dependent on the size of the smallest disk, so having a 256GB and a 512GB drive means you'll lose half the storage space of the latter hardware.
- RAID 1 is generally the most expensive choice since its efficiency can basically be measured as the number of drives divided by its own number.

Redundant Array of Inexpensive Disks (RAID) - About

- If you have three drives, RAID 5 is likely your best choice.
- RAID 5 uses a combination of striping and parity that is distributed across drives. In the event of a drive failure, an Exclusive Or (XOR) logic gate is used to piece together the lost drive using parity information from the other drives.
- This can be accomplished even while the other drives continue their regular function (though at a slower speed), meaning there will be minimal downtime if you lose a drive.

Redundant Array of Inexpensive Disks (RAID) - About

- RAID 6 is similar to RAID 5, though it uses at least four drives due to a dual parity setup.
- This means that in a four-drive setup you lose half of your storage space for parity, though your data will still be intact in the event of simultaneously losing two drives. Like RAID 5, read speeds are excellent due to striping, though write speeds are generally slowed down due to parity.
- Though it requires four drives to function, a RAID 6 setup is likely best suited for arrays with five or more drives.
- RAID 10 (also known as RAID 1+0) is best suited for four drives.
- Instead of relying on either mirroring or striping, it actually involves both, hence the 1+0.
- Data is essentially striped between two sets of mirrored disks, creating a system that can handle a single drive failure in either set of drives or a simultaneous drive failure in both sets. What it cannot deal with is a simultaneous drive failure of one set.

Redundant Array of Inexpensive Disks (RAID) - About

- RAID was originally invented to cut costs of '80s storage prices while simultaneously accounting for high fail-rates of some early technology.

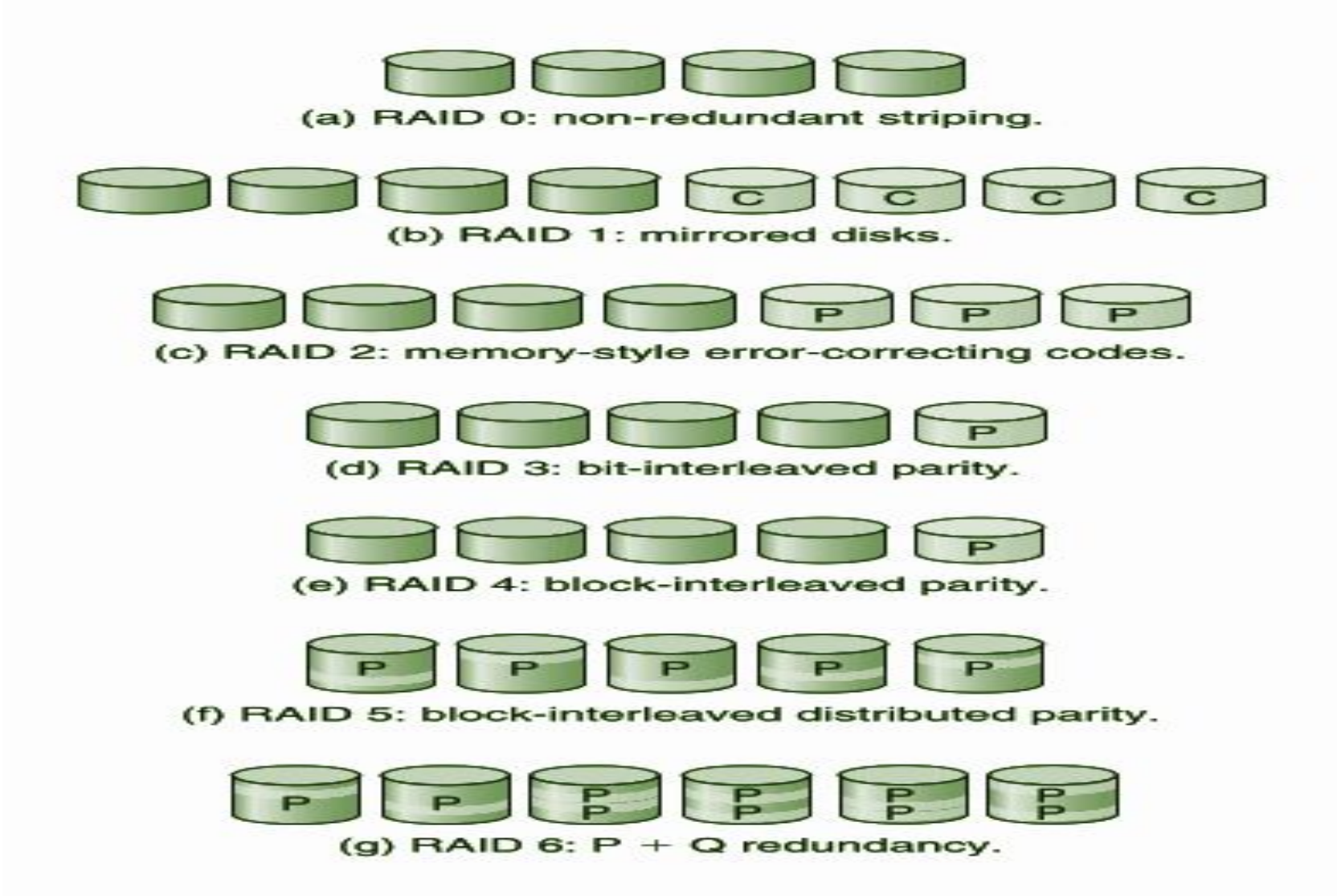
- RAID setups are still primarily used in large servers; think of corporations that require 24/7 access to their critical data.

- Now though, in an age of cheap bulk storage, RAID setups are often used privately in conjunction with specialized tasks and to avoid monthly fees and privacy concerns surrounding cloud storage services.

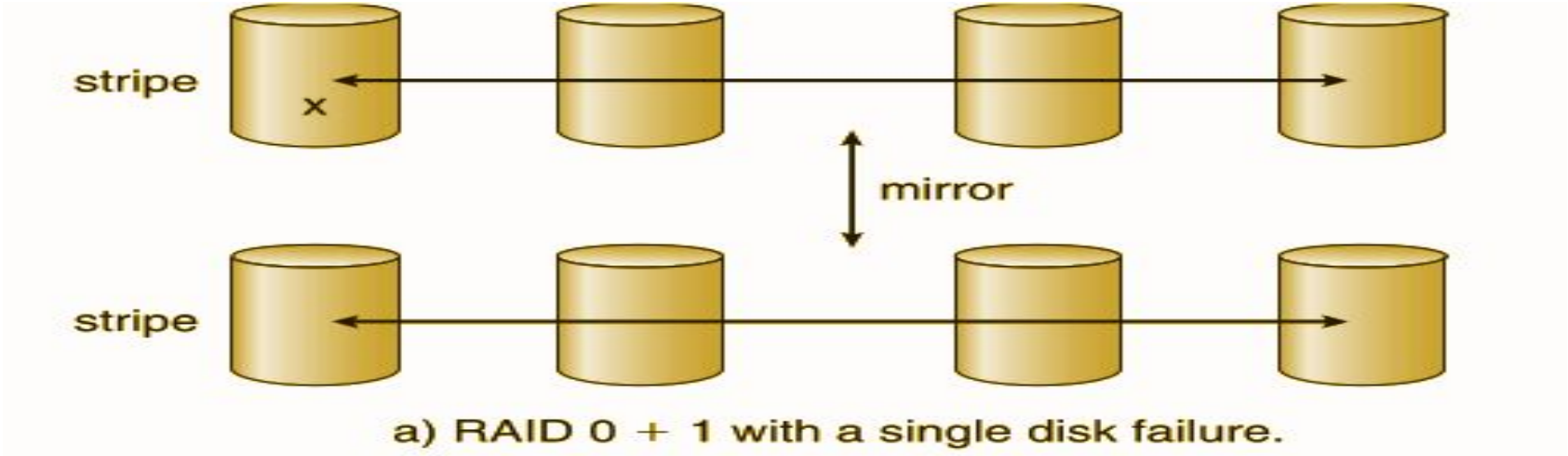
Redundant Array of Inexpensive Disks (RAID) - Structure

- Frequently combined with NVRAM to improve write performance
- Several improvements in disk-use techniques involve the use of multiple disks working cooperatively
- 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

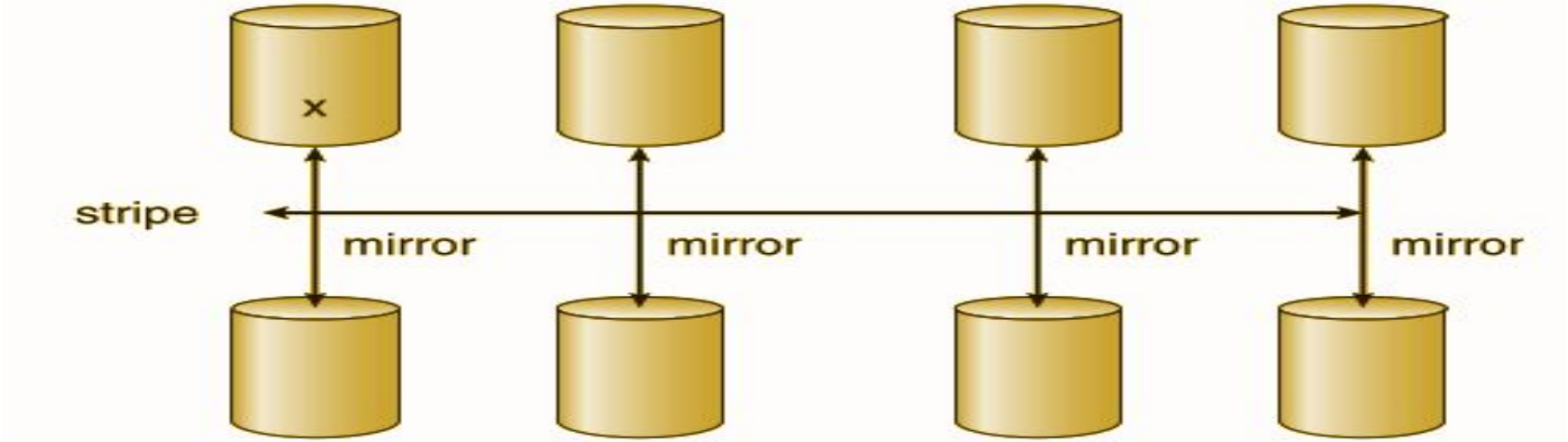
RAID Levels


(a) RAID 0: non-redundant striping.
C
(b) RAID 1: mirrored disks.
P
(c) RAID 2: memory-style error-correcting codes.
(d) RAID 3: bit-interleaved parity.
(e) RAID 4: block-interleaved parity.
(f) RAID 5: block-interleaved distributed parity.
(g) RAID 6: P + Q redundancy.

RAID (0 + 1) (1 + 0)



a) RAID 0 + 1 with a single disk failure.



b) RAID 1 + 0 with a single disk failure.

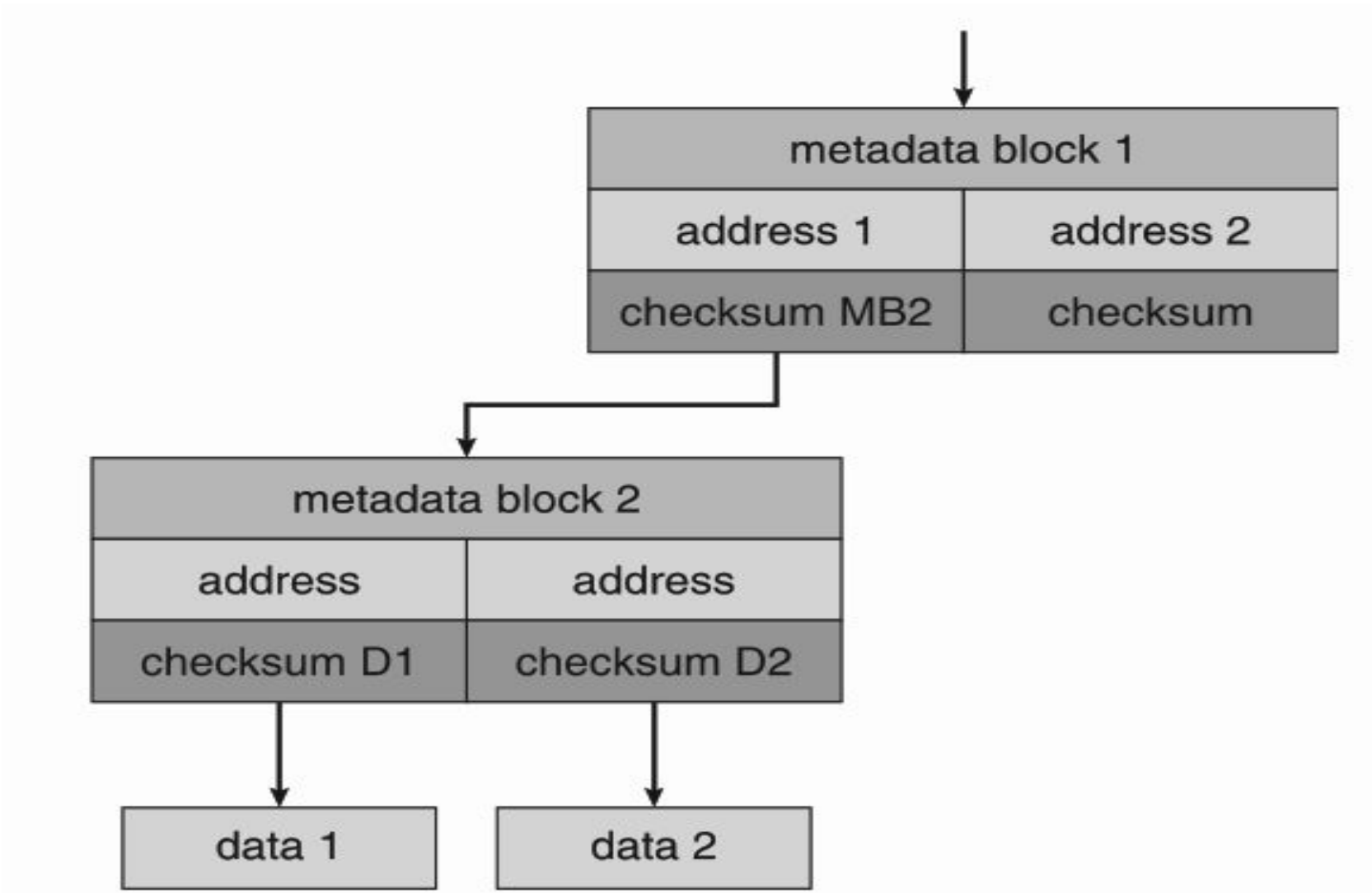
RAID - Other Features

- 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

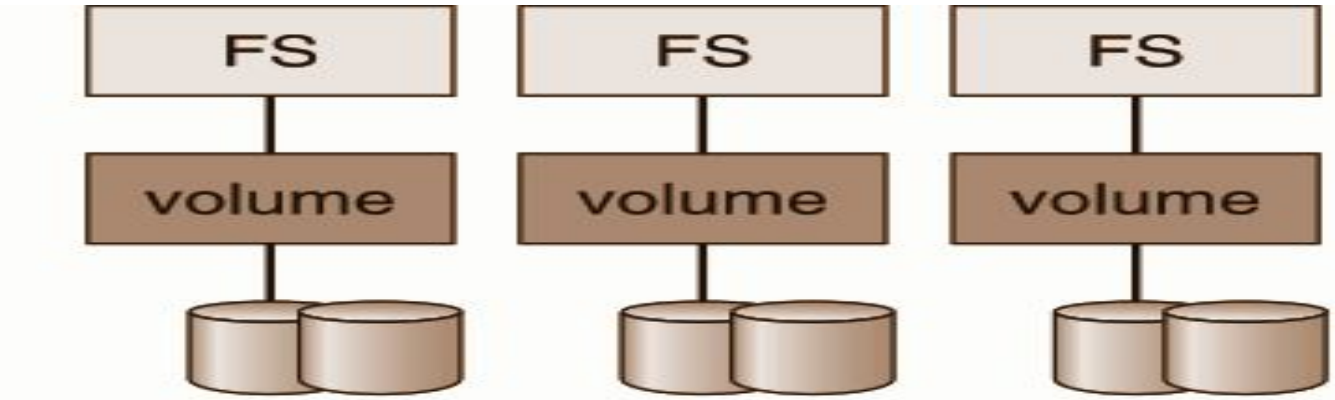
RAID - Extensions

- RAID alone does not prevent or detect data corruption or other errors, just disk failures
- Solaris ZFS adds checksums of all data and metadata
- Checksums kept with pointer to object, to detect if object is the right one and whether it changed
- Can detect and correct data and metadata corruption
- ZFS also removes volumes, partitions
  - Disks allocated in pools
  - Filesystems with a pool share that pool, use and release
  - space like malloc() and free() memory allocate /release calls

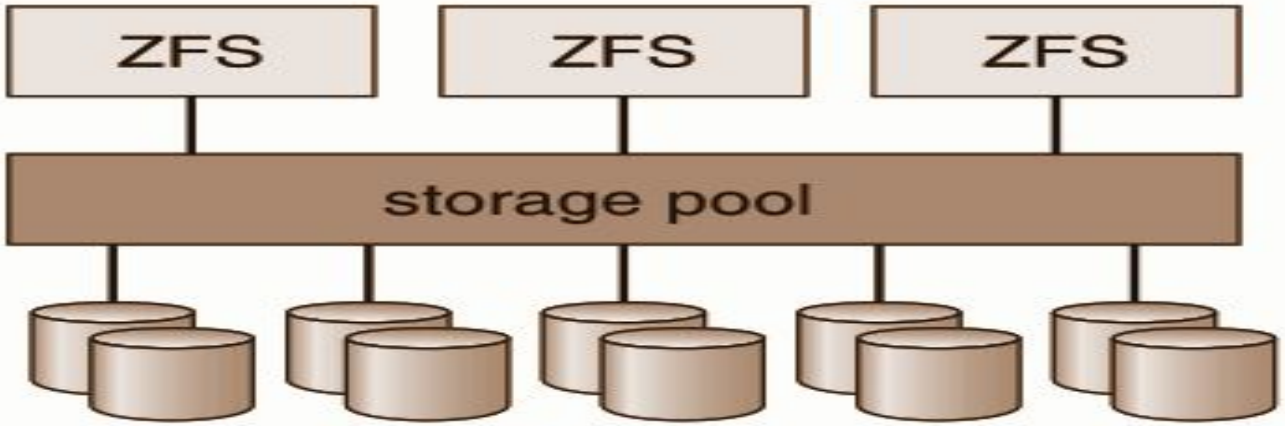
ZFS Checksums All Metadata and Data



Traditional and Pooled Storage



(a) Traditional volumes and file systems.



(b) ZFS and pooled storage.

Zettabyte File System

Topics Uncovered in this session

- Swap-Space Management
- Data Structures for Swapping on Linux Systems
- RAID Structure
- RAID Levels

Topics Uncovered in this session

- RAID - Other Features
- RAID Extensions
- ZFS Checksums All Metadata and Data
- Traditional and Pooled Storage

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com