# SNS COLLEGE OF TECHNOLOGY
# DEPARTMENT OF AIML
## 23CST202- OPERATING SYSTEMS
## II YEAR IV SEM AIML-B
UNIT 5 – I/O Systems

TOPIC – Swap Space management -RAID

Swapping is a memory management technique used in multi-programming to increase the number of processes sharing the CPU. It is a technique of removing a process from the main memory and storing it into secondary memory, and then bringing it back into the main memory for continued execution. This action of moving a process out from main memory to secondary memory is called Swap Out and the action of moving a process out from secondary memory to main memory is called Swap In.

Swap-space management is a technique used by operating systems to optimize memory usage and improve system performance. Here are some advantages and disadvantages of swap-space management:

Advantages:

1. Increased memory capacity: Swap-space management allows the operating system to use hard disk space as virtual memory, effectively increasing the available memory capacity.
2. Improved system performance: By using virtual memory, the operating system can swap out less frequently used data from physical memory to disk, freeing up space for more frequently used data and improving system performance.
3. Flexibility: Swap-space management allows the operating system to dynamically allocate and deallocate memory as needed, depending on the demands of running applications.

Disadvantages:

Slower access times: Accessing data from disk is slower than accessing data from physical memory, which can result in slower system performance if too much swapping is required.

Increased disk usage: Swap-space management requires disk space to be reserved for use as virtual memory, which can reduce the amount of available space for other data storage purposes.

Risk of data loss: In some cases, if there is a problem with the swap file, such as a disk error or corruption, data may be lost or corrupted.

Overall, swap-space management is a useful technique for optimizing memory usage and improving system performance. However, it is important to carefully manage swap space allocation and monitor system performance to ensure that excessive swapping does not negatively impact system performance.

Swap-Space :

The area on the disk where the swapped-out processes are stored is called swap space.

Swap-Space Management :
Swap-Space management is another low-level task of the operating system. Disk space is used as an extension of main memory by the virtual memory. As we know the fact that disk access is much slower than memory access, In the swap-space management we are using disk space, so it will significantly decreases system performance. Basically, in all our systems we require the best throughput, so the goal of this swap-space implementation is to provide the virtual memory the best throughput.
Swap-Space Use :
Swap-space is used by the different operating-system in various ways. The systems which are implementing swapping may use swap space to hold the entire process which may include image, code and data segments. Paging systems may simply store pages that have been pushed out of the main memory. The need of swap space on a system can vary from a megabytes to gigabytes but it also depends on the amount of physical memory, the virtual memory it is backing and the way in which it is using the virtual memory.

It is 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 the processes or may crash entirely. Overestimation wastes disk space that could otherwise be used for files, but it does not harm other.

Following table shows different system using amount of swap space:

| System | Swap-Space |
|---|---|
| 1. Solaris | Equal amount of physical memory |
| 2. Linux | Double the amount of physical memory |

Figure - Different systems using amount of swap-space

Explanation of above table :
Solaris, setting swap space equal to the amount by which virtual memory exceeds page-able physical memory. In the past Linux has suggested setting swap space to double the amount of physical memory. Today, this limitation is gone, and most Linux systems use considerably less swap space.

Including Linux, some operating systems; allow the use of multiple swap spaces, including both files and dedicated swap partitions. The swap spaces are placed on the disk so the load which is on the I/O by the paging and swapping will spread over the system's bandwidth.

Swap-Space Location :

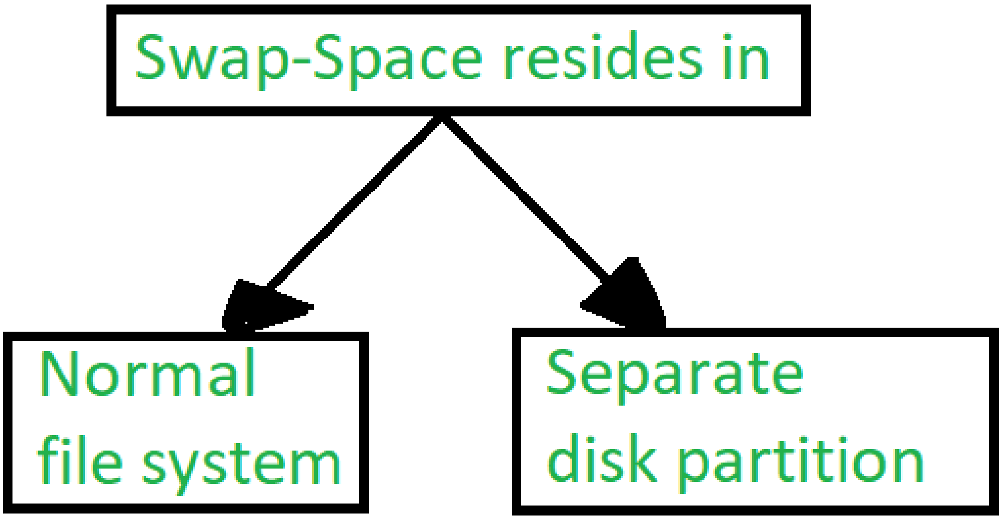



Figure - Location of swap-space

A swap space can reside in one of the two places -

1. Normal file system
2. Separate disk partition

Let, if the swap-space is simply a large file within the file system. To create it, name it and allocate its space normal file-system routines can be used. This approach, through easy to implement, is inefficient. Navigating the directory structures and the disk-allocation data structures takes time and extra disk access. During reading or writing of a process image, external fragmentation can greatly increase swapping times by forcing multiple seeks.

There is also an alternate to create the swap space which is in a separate raw partition. There is no presence of any file system in this place. Rather, a swap space storage manager is used to allocate and de-allocate the blocks. from the raw partition. It uses the algorithms for speed rather than storage efficiency, because we know the access time of swap space is shorter than the file system. By this Internal fragmentation increases, but it is acceptable, because the life span of the swap space is shorter than the files in the file system. Raw partition approach creates fixed amount of swap space in case of the disk partitioning.

Some operating systems are flexible and can swap both in raw partitions and in the file system space, example: Linux.

Swap-Space Management: An Example -

The traditional UNIX kernel started with an implementation of swapping that copied entire process between contiguous disk regions and memory. UNIX later evolve to a combination of swapping and paging as paging hardware became available. In Solaris, the designers changed standard UNIX methods to improve efficiency. More changes were made in later versions of Solaris, to improve the efficiency.

Linux is almost similar to Solaris system. In both the systems the swap space is used only for anonymous memory, it is that kind of memory which is not backed by any file. In the Linux system, one or more swap areas are allowed to be established. A swap area may be in either in a swap file on a regular file system or a dedicated file partition.

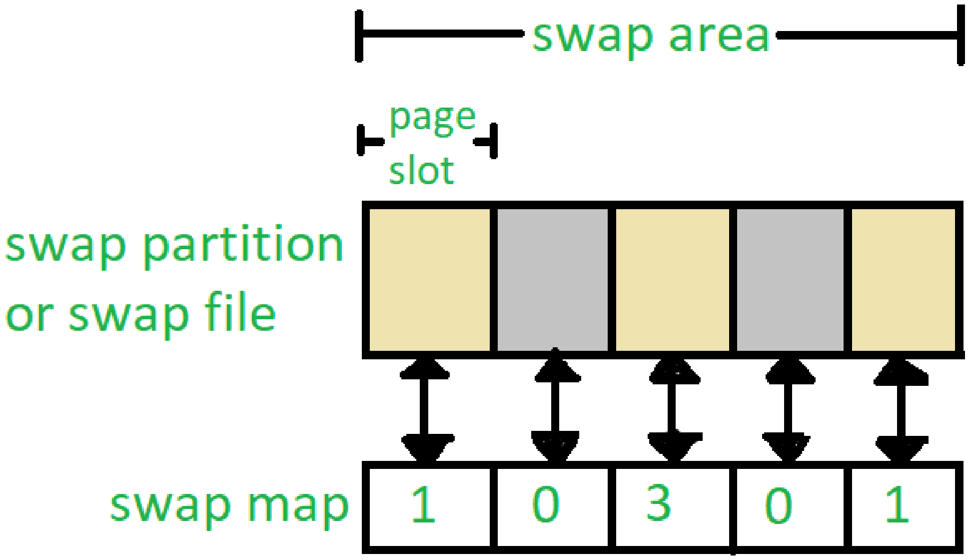



Figure - Data structure for swapping on Linux system

Each swap area consists of 4-KB page slots, which are used to hold the swapped pages. Associated with each swap area is a swap-map- an array of integers counters, each corresponding to a page slot in the swap area. If the value of the counter is 0 it means page slot is occupied by swapped page. The value of counter indicates the number of mappings to the swapped page. For example, a value 3 indicates that the swapped page is mapped to the 3 different processes.

RAID

RAID (Redundant Arrays of Independent Disks) is a technique that makes use of a combination of multiple disks for storing the data instead of using a single disk for increased performance, data redundancy, or to protect data in the case of a drive failure. The term was defined by David Patterson, Garth A. Gibson, and Randy Katz at the University of California, Berkeley in 1987.

What is RAID?

RAID (Redundant Array of Independent Disks) is like having backup copies of your important files stored in different places on several hard drives or solid-state drives (SSDs). If one drive stops working, your data is still safe because you have other copies stored on the other drives. It's like having a safety net to protect your files from being lost if one of your drives breaks down.

RAID (Redundant Array of Independent Disks) in a Database Management System (DBMS) is a technology that combines multiple physical disk drives into a single logical unit for data storage. The main purpose of RAID is to improve data reliability, availability, and performance. There are different levels of RAID, each offering a balance of these benefits.

How RAID Works?

Let's understand how RAID (Redundant Array of Independent Disks) works through a relatable example:

Imagine you have a favourite book that you want to keep safe. Instead of giving the entire book to just one friend for safekeeping, you take a smart approach:

1. Splitting the Book: You divide the book into smaller pieces (like chapters or sections) and give each piece to a different friend.
2. Making Copies: For extra security, you might also create duplicate pieces and give them to multiple friends.

Now, if one friend misplaces their piece, you can still recreate the entire book using the pieces held by the others. This way, your book is safe even if someone loses their portion.

This is exactly how RAID works with hard drives! RAID splits your data across multiple drives (similar to dividing the book into pieces). Depending on the RAID configuration, it may also create duplicates (like making extra copies). If one drive fails, the remaining drives can help reconstruct the lost data.

What is a RAID Controller?

A RAID controller is like a boss for your hard drives in a big storage system. It works between your computer's operating system and the actual hard drives, organizing them into groups to make them easier to manage. This helps speed up how fast your computer can read and write data, and it also adds a layer of protection in case one of your hard drives breaks down. So, it's like having a smart helper that makes your hard drives work better and keeps your important data safer.

Types of RAID Controller

There are three types of RAID controller:

Hardware Based: In hardware-based RAID, there's a physical controller that manages the whole array. This controller can handle the whole group of hard drives together. It's designed to work with different types of hard drives, like SATA (Serial Advanced Technology Attachment) or SCSI (Small Computer System Interface). Sometimes, this controller is built right into the computer's main board, making it easier to set up and manage your RAID system. It's like having a captain for your team of hard drives, making sure they work together smoothly.

Software Based: In software-based RAID, the controller doesn't have its own special hardware. So it use computer's main processor and memory to do its job. It perform the same function as a hardware-based RAID controller, like managing the hard drives and keeping your data safe. But because it's sharing resources with other programs on your computer, it might not make things run as fast. So, while it's still helpful, it might not give you as big of a speed boost as a hardware-based RAID system

Firmware Based: Firmware-based RAID controllers are like helpers built into the computer's main board. They work with the main processor, just like software-based RAID. But they only implement when the computer starts up. Once the operating system is running, a special driver takes over the RAID job. These controllers aren't as expensive as hardware ones, but they make the computer's main processor work harder. People also call them hardware-assisted software RAID, hybrid model RAID, or fake RAID.

Why Data Redundancy?

Data redundancy, although taking up extra space, adds to disk reliability. This means, that in case of disk failure, if the same data is also backed up onto another disk, we can retrieve the data and go on with the operation. On the other hand, if the data is spread across multiple disks without the RAID technique, the loss of a single disk can affect the entire data.

Key Evaluation Points for a RAID System

When evaluating a RAID system, the following critical aspects should be considered:

1. Reliability

Definition: Refers to the system's ability to tolerate disk faults and prevent data loss.

Key Questions:

- How many disk failures can the RAID configuration sustain without losing data?
- Is there redundancy in the system, and how is it implemented (e.g., parity, mirroring)?

Example:

- RAID 0 offers no fault tolerance; if one disk fails all data is lost.
- RAID 5 can tolerate one disk failure due to parity data.
- RAID 6 can handle two simultaneous disk failures.

2. Availability

Definition: The fraction of time the RAID system is operational and available for use.

Key Questions:

- What is the system's uptime, and how quickly can it recover from failures?
- Can data be accessed during disk rebuilds or replacements?

Example:

- RAID 1 (Mirroring) allows immediate data access even during a single disk failure.
- RAID 5 and 6 may degrade performance during a rebuild, but data remains available.

3. Performance

Definition: Measures how efficiently the RAID system handles data processing tasks. This includes:

- Response Time: How quickly the system responds to data requests.
- Throughput: The rate at which the system processes data (e.g., MB/s or IOPS).

Key Factors:

- RAID levels affect performance differently:
- RAID 0 offers high throughput but no redundancy.
- RAID 1 improves read performance by serving data from either mirrored disk but may not improve write performance significantly.
- RAID 5/6 introduces overhead for parity calculations, affecting write speeds.
- Workload type (e.g., sequential vs. random read/write operations).

Performance Trade-offs: Higher redundancy often comes at the cost of slower writes (due to parity calculations).

4. Capacity

Definition: The amount of usable storage available to the user after accounting for redundancy mechanisms.

Key Calculation: For a set of N disks, each with B blocks, the available capacity depends on the RAID level:

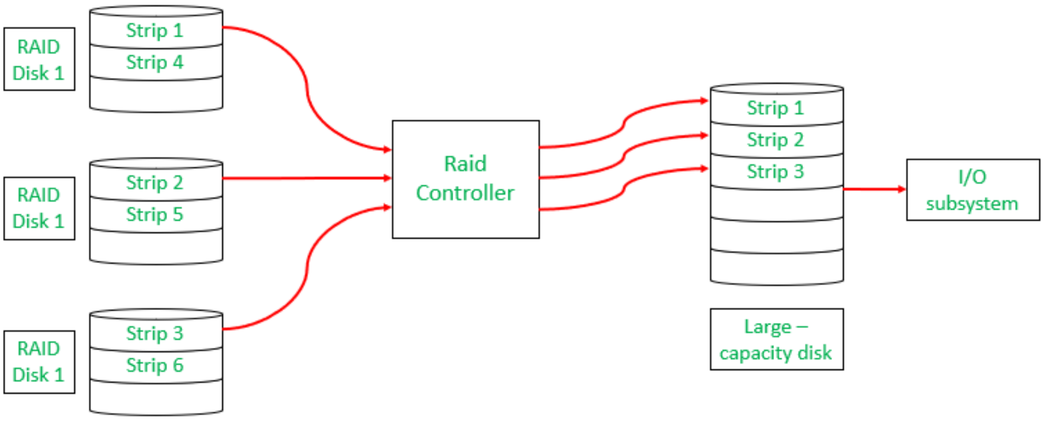
- RAID 0: All $N \times B$ blocks are usable (no redundancy).
- RAID 1: Usable capacity is B (only one disk's capacity due to mirroring).
- RAID 5: Usable capacity is $(N-1) \times B$ (one disk's worth of capacity used for parity).
- RAID 6: Usable capacity is $(N-2) \times B$ (two disks’ worth used for parity).

Trade-offs: Higher redundancy (RAID 5/6) reduces available capacity compared to non-redundant setups (RAID 0).

RAID is very transparent to the underlying system. This means, that to the host system, it appears as a single big disk presenting itself as a linear array of blocks. This allows older technologies to be replaced by RAID without making too many changes to the existing code.

Different RAID Levels

- RAID-0 (Stripping)
- RAID-1 (Mirroring)
- RAID-2 (Bit-Level Stripping with Dedicated Parity)
- RAID-3 (Byte-Level Stripping with Dedicated Parity)
- RAID-4 (Block-Level Stripping with Dedicated Parity)
- RAID-5 (Block-Level Stripping with Distributed Parity)
- RAID-6 (Block-Level Stripping with two Parity Bits)




Raid Controller

1. RAID-0 (Stripping)

- RAID-0 improves system performance by splitting data into smaller "blocks" and spreading them across multiple disks. This process is called "striping." It enhances data access speed by enabling parallel read/write operations but provides no redundancy or fault tolerance.

**RAID 0**

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

RAID-0

- A set of blocks distributed across the disks forms a "stripe." For instance, "0, 1, 2, 3" is one stripe.
- Instead of placing just one block into a disk at a time, we can work with two (or more) blocks placed into a disk before moving on to the next one.

Evaluation

- Reliability: 0
  There is no duplication of data. Hence, a block once lost cannot be recovered.
- Capacity: N*B
  The entire space is being used to store data. Since there is no duplication, N disks each having B blocks are fully utilized.

Advantages

- It is easy to implement.
- It utilizes the storage capacity in a better way.

Disadvantages

- A single drive loss can result in the complete failure of the system.
- It's not a good choice for a critical system.

2. RAID-1 (Mirroring)

- RAID-1 enhances reliability by creating an identical copy (mirror) of each data block on separate disks. This ensures that even if one disk fails, the data remains accessible from its duplicate. While this configuration is highly reliable, it requires significant storage overhead.

**RAID 1**

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

Raid-1

Mirroring:

Each block of data is written to two (or more) disks.

For example:

- Block 0 is stored on Disk 0 and its duplicate on Disk 1.
- Block 1 is stored on Disk 2 and its duplicate on Disk 3.

Mirroring Level 2:

In the figure, "mirroring level 2" indicates that there are two copies of each block, distributed across different disks.

RAID 0 was unable to tolerate any disk failure. But RAID 1 is capable of reliability.

Read about Difference Between RAID 0 and RAID 1.

Evaluation

Assume a RAID system with mirroring level 2.

- Reliability: 1 to N/2
  1 disk failure can be handled for certain because blocks of that disk would have duplicates on some other disk. If we are lucky enough and disks 0 and 2 fail, then again this can be handled as the blocks of these disks have duplicates on disks 1 and 3. So, in the best case, N/2 disk failures can be handled.
- Capacity: N*B/2
  Only half the space is being used to store data. The other half is just a mirror of the already stored data.

Advantages

- It covers complete redundancy.
- It can increase data security and speed.

Disadvantages

- It is highly expensive.
- Storage capacity is less.

3. RAID-2 (Bit-Level Stripping with Dedicated Parity)

- RAID-2 is a specialized RAID level that uses bit-level striping combined with error correction using Hamming Code. In this configuration, data is distributed at the bit level across multiple drives, and a dedicated parity drive is used for error detection and correction. While it offers strong fault tolerance, its complexity and cost make it rarely used in practice.

Advantages

- In case of Error Correction, it uses hamming code.
- It Uses one designated drive to store parity.

Disadvantages

- It has a complex structure and high cost due to extra drive.
- It requires an extra drive for error detection.

4. RAID-3 (Byte-Level Stripping with Dedicated Parity)

- RAID-3 enhances fault tolerance by employing byte-level striping across multiple drives and storing parity information on a dedicated parity drive. The dedicated parity drive allows for the reconstruction of lost data if a single drive fails. This configuration is suitable for workloads requiring high throughput for sequential data but is less efficient for random I/O operations.

**RAID 3**

| Disk 0 | Disk 1 | Disk 2 | Disk 3 |
|---|---|---|---|
| 15 | 16 | 17 | P(15, 16, 17) |
| 18 | 19 | 20 | P(18, 19, 20) |
| 21 | 22 | 23 | P(21, 22, 23) |
| 24 | 25 | 26 | P(24, 25, 26) |

Raid-3

- Here Disk 3 contains the Parity bits for Disk 0, Disk 1, and Disk 2. If data loss occurs, we can construct it with Disk 3.

Evaluation

Reliability: RAID-3 can tolerate the failure of one disk. The lost data can be reconstructed using the parity drive and the remaining data drives.

Capacity: Usable Capacity=(N−1)×B where N is the total number of drives, and B is the number of blocks per drive. The capacity of one drive is reserved for storing parity information.

Advantages

- Data can be transferred in bulk.
- Data can be accessed in parallel.

Disadvantages

- It requires an additional drive for parity.
- In the case of small-size files, it performs slowly.

Read about Difference Between RAID 2 and RAID 3.

5. RAID-4 (Block-Level Stripping with Dedicated Parity)

- RAID-4 introduces block-level striping across multiple disks, combined with a dedicated parity disk to provide fault tolerance. Data is written in blocks, and a separate disk stores parity information calculated using the XOR function. This setup allows for data recovery in case of a single disk failure, making RAID-4 more reliable than RAID-0 but less efficient in write-intensive scenarios due to reliance on a dedicated parity disk.

**RAID 4**

| 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 |

Raid-4

- In the figure, we can observe one column (disk) dedicated to parity.
- Parity is calculated using a simple XOR function. If the data bits are 0,0,0,1 the parity bit is XOR(0,0,0,1) = 1. If the data bits are 0,1,1,0 the parity bit is XOR(0,1,1,0) = 0. A simple approach is that an even number of ones results in parity 0, and an odd number of ones results in parity 1.

| C1 | C2 | C3 | C4 | Parity |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |

Raid-4

- Assume that in the above figure, C3 is lost due to some disk failure. Then, we can recompute the data bit stored in C3 by looking at the values of all the other columns and the parity bit. This allows us to recover lost data.

Read about Difference Between RAID 3 and RAID 4.

Evaluation

- Reliability: 1
  RAID-4 allows recovery of at most 1 disk failure (because of the way parity works). If more than one disk fails, there is no way to recover the data.
- Capacity: (N-1)*B
  One disk in the system is reserved for storing the parity. Hence, (N-1) disks are made available for data storage, each disk having B blocks.

Advantages

- It helps in reconstructing the data if at most one data is lost.

Disadvantages

- It can't help reconstructing data when more than one is lost.

6. RAID-5 (Block-Level Stripping with Distributed Parity)

- RAID-5 builds on RAID-4 by distributing parity information across all disks instead of storing it on a dedicated parity drive. This distributed parity significantly improves write performance, especially for random write operations, while maintaining fault tolerance for single disk failures. RAID-5 is one of the most commonly used RAID configurations due to its balance between reliability, performance, and storage efficiency.

# RAID 5

| 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

How RAID-5 Works?

Block-Level Striping: Data is divided into blocks and striped across all drives in the array.

Distributed Parity: Parity bits, calculated using the XOR function, are distributed across all drives in a rotating pattern.

For example:

- Disk 0: Data (D1, D2), Parity (P1)
- Disk 1: Data (D3, D4), Parity (P2)
- Disk 2: Data (D5, D6), Parity (P3)

This rotation ensures no single disk is burdened with all parity operations, reducing bottlenecks.

Data Recovery: In case of a single disk failure, the missing data can be reconstructed by XOR-ing the remaining data blocks and parity information.

Evaluation

- Reliability: 1
  RAID-5 allows recovery of at most 1 disk failure (because of the way parity works). If more than one disk fails, there is no way to recover the data. This is identical to RAID-4.
- Capacity: (N-1)*B
  Overall, space equivalent to one disk is utilized in storing the parity. Hence, (N-1) disks are made available for data storage, each disk having B blocks.

Advantages

- Data can be reconstructed using parity bits.
- It makes the performance better.

Disadvantages

- Its technology is complex and extra space is required.
- If both discs get damaged, data will be lost forever.

7. RAID-6 (Block-Level Stripping with two Parity Bits)

- RAID-6 is an advanced version of RAID-5 that provides enhanced fault tolerance by introducing double distributed parity. This allows RAID-6 to recover from the failure of up to two disks simultaneously, making it more reliable for critical systems with larger arrays. However, the added parity calculations can impact write performance.

# RAID 6

| Disk 1 | Disk 2 | Disk 3 | Disk 4 |
|---|---|---|---|
| A1 | B1 | P(B1) | P(B1) |
| A2 | P(B2) | P(B2) | B2 |
| P(B3) | P(B3) | A3 | B3 |
| P(B4) | A4 | A4 | P(B4) |

Raid-6

How RAID-6 Works ?

Block-Level Striping: Data is divided into blocks and striped across all disks in the array.

Double Distributed Parity: Two sets of parity information are calculated for every block and distributed across all disks in the array in a rotating pattern.

Example:

- Disk 0: Data (D1, D2), Parity (P1)
- Disk 1: Data (D3, D4), Parity (P2)
- Disk 2: Data (D5, P1), Parity (P3)
- Disk 3: Parity (P2, P3), Data (D6)

Data Recovery: If one or two disks fail, the missing data can be reconstructed using the remaining data and parity information.

Evaluation

1. Reliability: RAID-6 can tolerate the simultaneous failure of two disks, providing greater fault tolerance than RAID-5.

2. Capacity: Usable Capacity=$(N-2)\times B$ where N is the total number of disks and B is the number of blocks per disk.

Advantages

- Very high data Accessibility.
- Fast read data transactions.

Disadvantages

- Due to double parity, it has slow write data transactions.
- Extra space is required.

Advantages of RAID

- Data redundancy: By keeping numerous copies of the data on many disks, RAID can shield data from disk failures.
- Performance enhancement: RAID can enhance performance by distributing data over several drives, enabling the simultaneous execution of several read/write operations.
- Scalability: RAID is scalable, therefore by adding more disks to the array, the storage capacity may be expanded.
- Versatility: RAID is applicable to a wide range of devices, such as workstations, servers, and personal PCs

Disadvantages of RAID

- Cost: RAID implementation can be costly, particularly for arrays with large capacities.
- Complexity: The setup and management of RAID might be challenging.
- Decreased performance: The parity calculations necessary for some RAID configurations, including RAID 5 and RAID 6, may result in a decrease in speed.
- Single point of failure: RAID is not a comprehensive backup solution while offering data redundancy. The array's whole contents could be lost if the RAID controller malfunctions.