Parallel Computing

Parallel computing is a type of computation where many calculations or processes are carried out simultaneously. Large problems can often be divided into smaller ones, which can then be solved at the same time. It relies heavily on the principle that large problems can be divided into smaller ones that are then solved concurrently or "in parallel".

Hardware Components

- Processors (CPUs/GPUs): In parallel computing, multiple processors are used to perform multiple tasks simultaneously. Processors, whether general-purpose CPUs or specialised GPUs, are essential for executing parallel tasks.
- Memory/RAM: Memory is required to store and quickly access the data being processed. In a parallel system, memory may be shared (as in Shared Memory Multiprocessing), or distributed where each processor has its own (as in Distributed Memory Multiprocessing).
- Interconnection Network devices: These are the network devices that connect all the processors and memory units in a parallel system. They are responsible for facilitating communication between processors and memory units.
- Storage Systems: Similar to cloud computing, parallel computing also requires substantial storage systems to hold the data that's being processed.
- Load Balancers: In parallel computing, load balancers are responsible for distributing tasks across multiple computing resources. This helps ensure that no single node is overwhelmed with tasks, improving system efficiency and responsiveness.

Software Components

- Parallel Programming Languages and Libraries: Languages like C/C++ with MPI or OpenMP, Fortran, or more recently Python, are used to write

parallel algorithms. These languages and their libraries contain constructs for common parallel computing patterns.

- Operating System: Parallel computers require a special operating system capable of managing and scheduling parallel tasks, handling communication between tasks, and managing memory and other resources.
- Parallel Compilers: These are needed to compile the parallel code written in the programming languages. They transform the source code into machine code that can be executed in parallel on the hardware.
- Middleware: This software provides an additional level of abstraction to enable interaction between distributed systems in a network, providing essential services for communication, authentication, and more.
- Parallel File Systems: These are essential to allow for simultaneous access to and operations on files. Examples include IBM's General Parallel File System (GPFS) and the open-source Lustre.

SAN and NAS in Parallel Computing:

In parallel computing, both SAN and NAS have their own roles. SAN, with its block-level access, provides high-speed data transfers and can handle large data sets, making it suitable for applications that require high-performance computing and large-scale data processing. In contrast, NAS, with its file-level storage, is excellent for smaller scale applications where data sharing between nodes is necessary.

- SAN: SANs in parallel computing can be used to store and retrieve large blocks of data. It provides high-speed connections allowing for real-time or near real-time data processing. It is especially suitable for situations where the organisation needs to process massive amounts of data quickly and efficiently, such as big data analytics or complex scientific simulations.
- NAS: NAS in parallel computing is often used for storing shared files and providing a central location for users and/or processes to access and store data. NAS could be suitable for an organisation that requires multiple nodes (computers, servers, etc.) to access the same data for

read and write operations, like collaborative project work, content serving, or certain types of data analysis work where data sharing is necessary.

- In parallel computing, Mass Storage and RAID (Redundant Array of Independent Disks) are concepts related to data storage and fault tolerance. In parallel computing, Mass Storage is essential for storing and retrieving data used by parallel applications. It allows for the persistence and durability of data, enabling parallel programs to read and write large volumes of data efficiently.