PROGRAMMING MODEL

1. Parallel and Distributed Programming Paradigms

Parallel and distributed programming paradigms enable multiple computing tasks to be processed simultaneously across different computing nodes or processors. This setup enhances processing speed, efficiency, and scalability, crucial for handling large datasets and complex computations, often used in cloud computing, big data analytics, and scientific simulations.

Key Paradigms:

- **Parallel Computing**: Multiple processors execute tasks simultaneously within a single system. It's efficient for tasks that can be divided into independent sub-tasks.
- **Distributed Computing**: Involves multiple systems connected over a network, where each system (node) processes part of a task. It's ideal for tasks that require resources beyond what a single system can provide.

	Parallel Systems	Distributed Systems
Memory	**Tightly coupled shared** memory UMA, NUMA	**Distributed** memory Message passing, RPC, and/or used of distributed shared memory
Control	**Global clock** control SIMD, MIMD	**No global clock** control Synchronization algorithms needed
Processor interconnection	Order of **Tbps** Bus, mesh, tree, mesh of tree, and hypercube (-related) network	Order of **Gbps** Ethernet(bus), token ring and SCI (ring), myrinet(switching network)
Main focus	**Performance** Scientific computing	Performance(**cost and scalability**) **Reliability/availability** **Information/resource sharing**

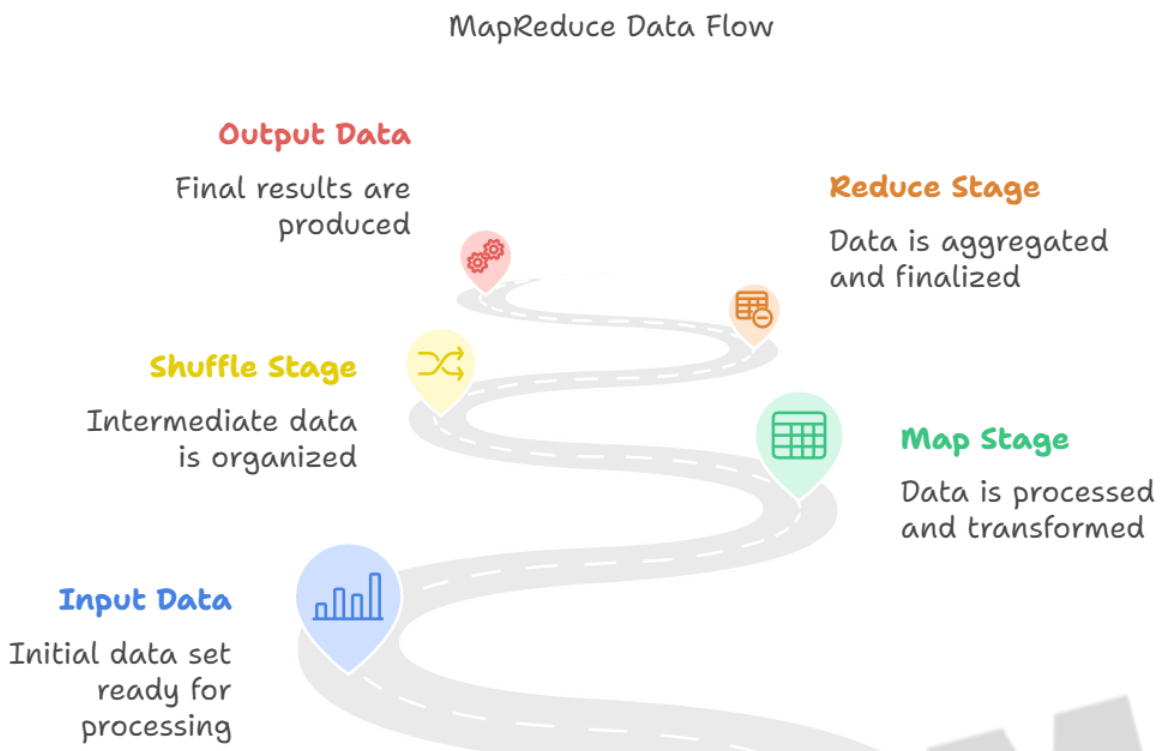
---

2. MapReduce

MapReduce is a programming model developed by Google to process large datasets in parallel across a distributed cluster of computers. It divides tasks into two main phases:

- **Map Phase**: Splits input data into smaller chunks and processes each chunk independently.
- **Reduce Phase**: Aggregates the results from the Map phase to produce the final output.

MapReduce is particularly useful for large-scale data processing applications like data mining and machine learning.



---

3. Twister and Iterative MapReduce

Twister is a distributed MapReduce framework optimized for iterative computations, which are common in scientific and machine learning applications. Unlike traditional MapReduce, Twister allows for:

- **Efficient Iterations**: Enables repeated MapReduce operations without repeatedly loading data, enhancing performance.
- **Caching Mechanisms**: Caches intermediate data, avoiding redundant computations.

Iterative Data Processing with Twister

Iteration

Repeating the process with updated data and cached results.

Data Retrieval

Accessing cached data for further processing.

Cache Data

Storing processed data in a cache for reuse.

Data Processing

Performing computations and transformations on the data.

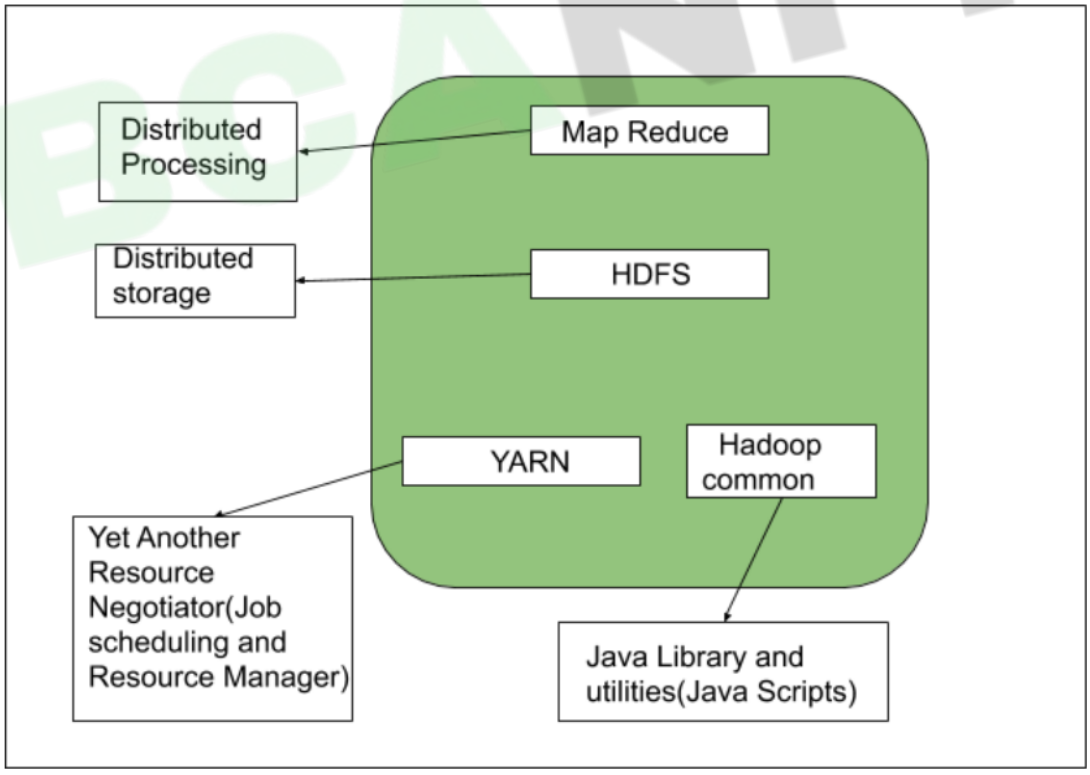
Initial Data Load

Loading the initial set of data into the system.

4. Hadoop Library from Apache

Apache Hadoop is an open-source framework that facilitates distributed storage and processing of large data across clusters of computers using simple programming models. Its core components include:

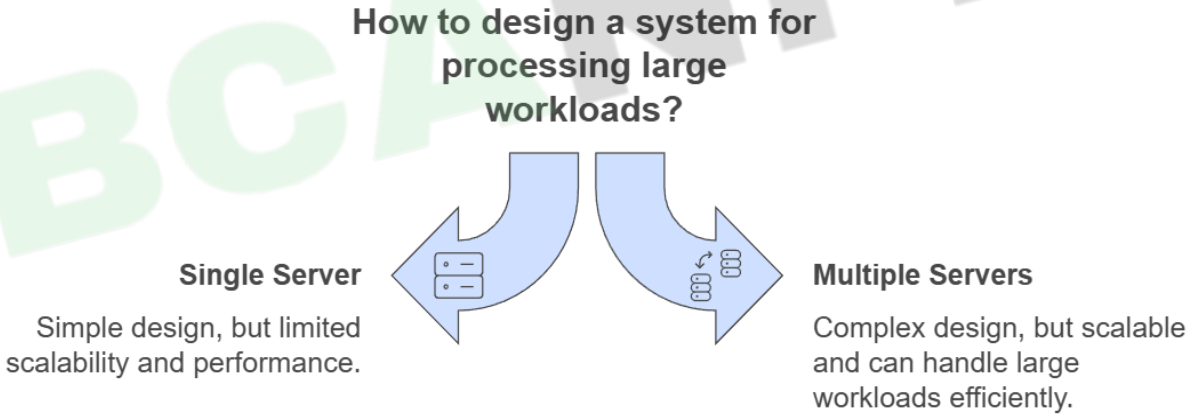
- **HDFS (Hadoop Distributed File System)**: A storage system that manages and replicates data across nodes.
- **MapReduce Engine**: Handles data processing tasks in a distributed environment.
- **YARN (Yet Another Resource Negotiator)**: Manages and allocates resources within the Hadoop cluster.



5. Mapping Applications

In cloud environments, mapping applications help users distribute tasks across resources efficiently. With tools like Hadoop and MapReduce, mapping applications become simpler, allowing developers to:

- **Divide Workloads**: Split applications into smaller, manageable parts to run on multiple servers.
- **Optimize Resource Usage**: Allocate resources dynamically based on demand.
- **Parallelize Tasks**: Execute tasks simultaneously to save time and improve efficiency.



---

6. Programming Support

Programming support in cloud computing simplifies the development, deployment, and management of applications

across distributed systems. Some popular cloud-based programming platforms include:

- **Google App Engine**: A PaaS offering that lets developers build applications on Google's infrastructure, supporting several programming languages and automatic scaling.
- **Amazon AWS (Amazon Web Services)**: A comprehensive cloud platform that provides compute power, storage, and databases on demand, widely used for its flexibility and range of services.

---

7. Cloud Software Environments

Cloud software environments provide infrastructure and tools for building, deploying, and managing cloud applications. Some notable platforms include:

Eucalyptus

An open-source platform that provides IaaS, compatible with AWS, enabling private cloud infrastructure creation. It allows users to manage cloud resources, storage, and networks locally.

OpenNebula

A versatile cloud management platform designed for data centers and virtualized infrastructures, offering features like resource allocation, virtual machine (VM) management, and network configuration.

OpenStack

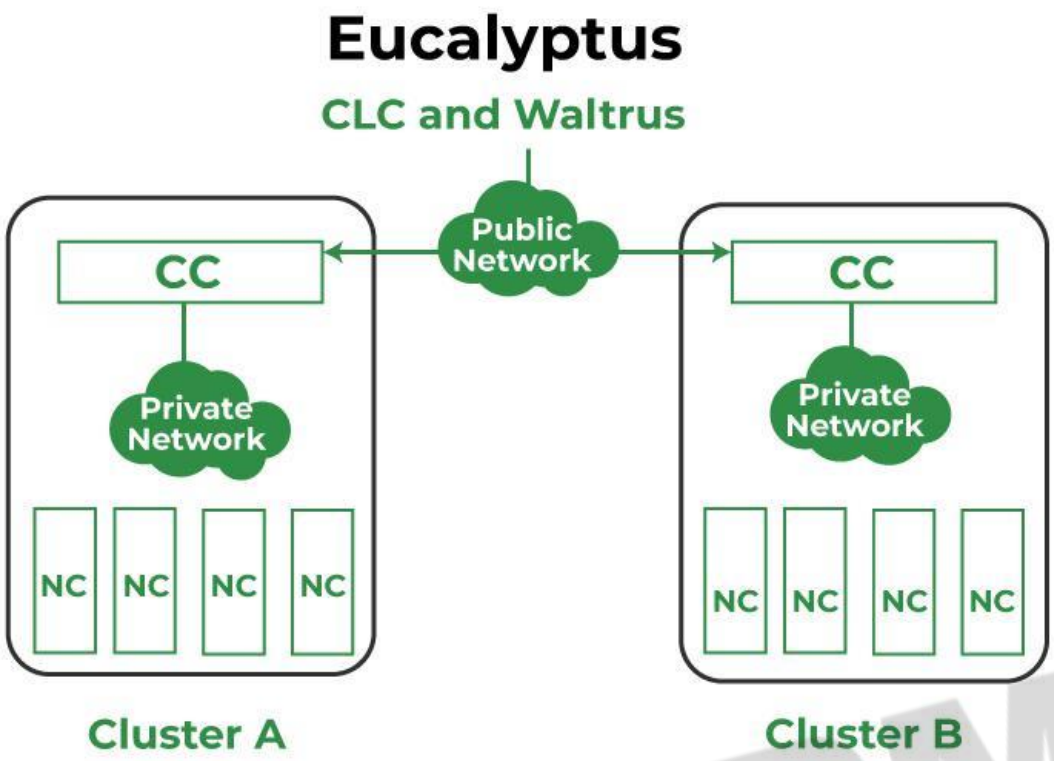
An open-source cloud operating system supporting public and private clouds, OpenStack provides components for compute, storage, and networking, allowing users to control large pools of computing resources.

Aneka

A cloud platform for building .NET applications in distributed environments. Aneka supports task distribution, resource scheduling, and load balancing.

CloudSim

A simulation framework for modeling and simulating cloud environments, helping researchers test cloud algorithms and applications without needing a physical cloud environment.


Eucalyptus
CLC and Waltrus
Public Network
CC
CC
Private Network
Private Network
NC
NC
NC
NC
NC
NC
NC
NC
Cluster A
Cluster B