Contents

[MapReduce 2](#_Toc140063794)

[YARN 3](#_Toc140063795)

[Kubernets 4](#_Toc140063796)

[Standalone 5](#_Toc140063797)

# MapReduce

MapReduce is a programming model and a way of processing large amounts of data in a distributed and parallel manner. It is designed to simplify the processing of big data across a cluster of computers.

1. Map Phase: During the map phase, the input data is divided into smaller chunks and distributed to different computers in a cluster. Each computer independently processes its assigned data and generates intermediate key-value pairs. This processing is done in parallel, meaning multiple computers are working on different chunks of data simultaneously.
2. Shuffle and Sort: The intermediate key-value pairs generated in the map phase are then grouped and sorted based on their keys. This step is called shuffle and sort. It ensures that all the values associated with the same key are brought together.
3. Reduce Phase: In the reduce phase, the sorted key-value pairs are passed to another set of computers, where they are combined, aggregated, or analyzed to produce the final output. Each computer receives a subset of the sorted key-value pairs and performs its own computation on that subset. The results from all the computers are combined to generate the final result.

# YARN

Certainly! YARN stands for Yet Another Resource Negotiator. It is a central component of Apache Hadoop, a popular framework for processing big data. YARN is responsible for managing and allocating resources in a Hadoop cluster.

Here's a simplified explanation of how YARN works:

1. Resource Management: YARN is responsible for managing the resources available in a Hadoop cluster. It keeps track of the available computing resources such as CPU and memory across all the nodes in the cluster. It ensures that the resources are efficiently utilized and allocated to different applications running on the cluster.

2. Application Coordination: YARN coordinates the execution of different applications or tasks running on the cluster. It acts as a mediator between the applications and the cluster resources. When an application is submitted to the cluster, YARN assigns necessary resources to it and monitors its progress.

3. Application Master: YARN creates an Application Master for each application running on the cluster. The Application Master is responsible for requesting and managing resources specific to that application. It communicates with the YARN Resource Manager to negotiate and acquire the necessary resources for its application.

4. Task Execution: YARN divides the submitted application into smaller tasks, which are then scheduled and executed on individual nodes in the cluster. Each task is assigned to a container, which is a specific allocation of resources (CPU, memory) on a node. YARN ensures that the tasks are executed in parallel across multiple nodes to speed up the processing.

YARN provides a flexible and scalable framework for running various distributed applications on a Hadoop cluster. It enables efficient resource management, workload coordination, and parallel execution of tasks, making it easier to process large amounts of data.

# Kubernets

Kubernetes, often referred to as "K8s" (as there are eight letters between 'K' and 's' in "Kubernetes"), is an open-source container orchestration platform. It simplifies the deployment, scaling, and management of applications and services across a cluster of computers.

Here's a simplified explanation of how Kubernetes works:

1. Containers: Kubernetes leverages the concept of containers, which are lightweight and isolated environments that package an application and its dependencies. Containers allow applications to run consistently across different computing environments.

2. Cluster: Kubernetes operates on a cluster of computers, called nodes, which can be physical or virtual machines. Each node runs multiple containers.

3. Master Node: Kubernetes has a control plane that consists of a master node. The master node is responsible for managing the cluster and making decisions about scheduling containers, scaling applications, and ensuring the desired state of the cluster.

4. Worker Nodes: Worker nodes are the computers where the containers are actually deployed and run. They receive instructions from the master node and execute them accordingly.

5. Pods: Kubernetes organizes containers into logical units called pods. A pod represents one or more containers that are tightly coupled and share resources, such as network and storage. It's the smallest deployable unit in Kubernetes.

6. Scaling and Load Balancing: Kubernetes allows you to scale your applications easily. You can increase or decrease the number of replicas of a pod or scale the number of worker nodes based on the demand. Kubernetes also handles load balancing, distributing network traffic across multiple pods to ensure optimal performance.

7. Service Discovery and Networking: Kubernetes provides a mechanism for service discovery, which enables different parts of an application to find and communicate with each other. It assigns a unique IP address to each pod and manages the networking between them.

8. Self-Healing: Kubernetes constantly monitors the health of containers and nodes. If a container or node fails, Kubernetes automatically restarts or reschedules the affected containers on other healthy nodes to ensure high availability and fault tolerance.

Kubernetes simplifies the management and scaling of containerized applications, making it easier to deploy and maintain complex applications across a cluster of machines. It provides features for automation, scalability, resilience, and service discovery, enabling efficient application deployment and management.

# Standalone

In the context of software or systems, "standalone" refers to something that is capable of functioning independently without relying on or being connected to other components or systems.

Imagine you have a computer program or a device that is described as "standalone." This means that it can operate by itself without needing any other programs, devices, or external connections to function properly.