ASSIGNMENT 2

Ashna Karim

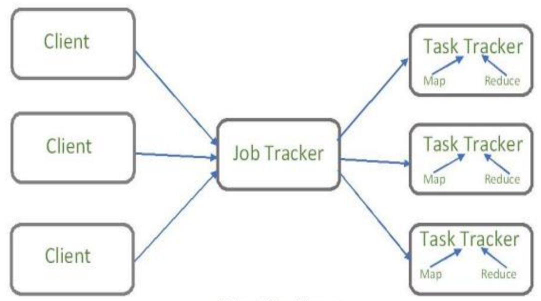
Roll no: 16

RLMCA 204

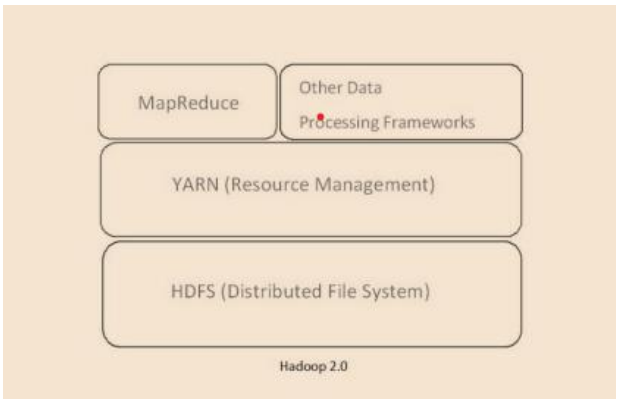
Importance of Hadoop YARN in HDFS

YARN is the main component of Hadoop v2.0. YARN helps to open up Hadoop by allowing to process and run data for batch processing, stream processing, interactive processing and graph processing which are stored in HDFS. In this way, It helps to run different types of distributed applications other than MapReduce.

MapReduce performs functions of Resource Management and Processing. Hadoop v1.0 is also known as MapReduce Version 1 (MRV1). There was only a single master for Job Tracker. In the previous version of Hadoop that is Hadoop version 1.0, which is also known as MapReduce version 1 (MRV1) use to perform both the task of process and resource management by itself. It has a job tracker module that is responsible for everything. Hence it is the single master that allocates resources for applications, performs scheduling for demand and also monitors the jobs of processing in the system. Hadoop version 1.0 reduces tasks & assigns maps on several sub-processes which is called Task Trackers. Task Tracker also reports the progress of processes in a periodical manner. But the main issue is not that, the problem is this design of a single master for all, resulting in bottlenecking issue. Also, the computational resource utilization was inefficient. Thus scalability became an issue with this version of Hadoop. But on the bright side, this issue is resolved by YARN, a vital core component in its successor Hadoop version 2.0 which was introduced in the year 2012 by Yahoo and Hortonworks. The basic idea behind this relief is separating MapReduce from Resource Management and Job scheduling instead of a single master. Thus, YARN is now responsible for Job scheduling and Resource Management.
In Hadoop 2.0, The concept of Application Master and Resource Manager was introduced by YARN. Across the cluster of Hadoop, the utilization of resources is monitored by the Resource Manager.



Hadoop 1.0 architecture


MapReduce
Other Data Processing Frameworks
YARN (Resource Management)
HDFS (Distributed File System)

Hadoop 2.0

YARN Features

1. **Multi-tenancy:** YARN has allowed access to multiple data processing engines such as batch processing engine, stream processing engine, interactive processing engine, graph processing engine and much more. This has given the benefit of multi-tenancy to the company.
2. **Cluster Utilization:** Clusters are utilized in an optimized way because clusters are used dynamically in Hadoop with the help of YARN.
3. **Compatibility**: YARN is also compatible with the first version of Hadoop, i.e. Hadoop 1.0, because it uses the existing map-reduce apps. So YARN can also be used with Hadoop 1.0.
4. **Scalability**: Thousands of clusters and nodes are allowed by the scheduler in Resource Manager of YARN to be managed and extended by Hadoop.

Components of YARN

- Container:

In the Container, one can find physical resources like a disk on a single node, CPU cores, RAM. Container Launch Context (CLC) is used to invoke containers. Data about the dependencies, security tokens, environment variables which are maintained as a record known as Container Launch Context (CLC).

1. On a specific host, an application can only use specified memory from the CPU and Memory. This specified amount of memory can only be used after the permission has been granted by the Container.
2. Container Launch Context is used to manage YARN Containers. It is also called Container LifeCycle (CLC). Necessary commands for the creation of the process is stored in this record. It also saves the payload for Node Manager services, security tokens, dependencies, map of environment variables.

- Application Master:

In a framework, when a single job is submitted, it is called an application. Monitoring the application progress, application status tracking, negotiation of resources with resource manager is the responsibility of the application manager. All the requirement of an application to run is done by sending the Container Launch Context (CLC). The application master posts container Launch Context (CLC) by requesting the Container from the node manager. From time to time, the resource manager receives a health report after the application has started.

- Node Manager:

The node manager takes care of individual nodes in the Hadoop cluster and also manages containers related to each specific node. It is registered with the Resource Manager and sends each node's health status to the Resource Manager, stating if the node process has finished working with the resource. As its primary goal is to manage each specific node container that is assigned by the resource manager. The node manager also creates a container process when requested by the Application master. When the application master sends and asks the attached Container from the node manager by a CLC(Container Launch Context) which includes everything an application needs to execute. Then the node manager creates the requested process container and runs it. Node manager is also responsible for monitoring resource usage by individual Container and reporting it to the Resource manager. Thus node manager and resource manager collaborate to communicate between nodes and manage resource usage by each node in the cluster. It can also kill containers if directed by the Resources manager. Finally, node managers log everything by the log management system in it.
A particular node is taken care of by the Node Manager. The Node Manager manages the workflow and application of the node. Log management is performed, and the Node Manager monitors resource usage. The resource manager gives directions to kill a container to the Node Manager. The application master requests the Node manager to start the container process. The creation of a container process is the responsibility of the Node Manager.

- Resource Manager:

Resource management and assignment of all the apps is the responsibility of Resource Manager and is the master daemon of YARN. Requests received by the resource manager are forwarded to the corresponding node manager. According to the application, resources are allocated by the resource manager for completion.

1. Utilization of Cluster is optimized, such as keeping the usage of all the resources active against different kinds of limitations such as SLAs, fairness and capacity guarantees.
2. The Resource Manager does the allocation of available resources.
3. The Resource Manager arbitrates cluster resources.
4. The actual processing of requests takes place in nodes, and the node managers manage it. Whenever any request for processing is received, it transfers the requests in parts to its corresponding node managers.
5. Resource Manager is the highest authority for the allocation of resources.

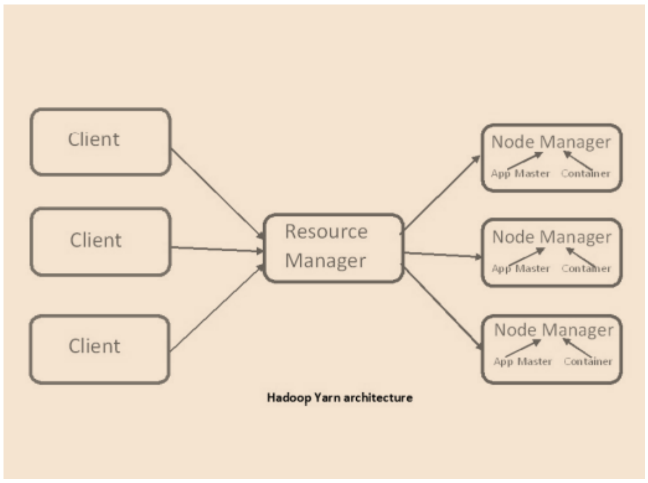
There are two primary components of the Resource Manager, which are: –

- Application Manager –

The application manager is responsible for managing a set of submitted tasks or applications. It first verifies and validates the submitted application's specifications and may reject the applications if there are not enough resources available. It also ensures no other application exists with the same ID which is already submitted that can be caused by an erroneous or a malicious client. Then it forwards the submitted application after validation to the scheduler. Finally, it also observes the states of applications and manages finished applications to save some Resource Manager's memory. The application manager keeps a cache of finished applications and moves out old, finished applications to accommodate space for freshly submitted applications.

- Scheduler –

Based on Resource Availability and Application Allocation, Scheduler schedules the tasks. There is no other task performed by scheduler like no restart of the job after failing, tracking or monitoring of tasks. The different types of scheduler plugins are Fair Scheduler and Capacity Scheduler, which are supported by the YARN scheduler for the partition of cluster resources.



Steps of Workflow of Application in Hadoop YARN

An application is submitted by the client.

1. Application Manager is started by the allocation of the Container by the Resource Manager.
2. Resource Manager and Application Manager register with each other.
3. The Application Manager does the negotiation of the Container to the Resource Manager.
4. The Node Manager launches the Container after being notified by the Application Manager.
5. Execution of Application code is done in the Container.
6. The Application Manager or Resource Manager monitors the status of the application after being contacted by the client.
7. Un-Registration of Application Manager is done with Resource Manager after the process is complete.

Analyze the role of PIG

What is Apache Pig?

Apache Pig is an abstraction over MapReduce. It is a tool/platform which is used to analyze larger sets of data representing them as data flows. Pig is generally used with Hadoop; we can perform all the data manipulation operations in Hadoop using Apache Pig.

Why Do We Need Apache Pig?

Programmers who are not so good at Java normally used to struggle working with Hadoop, especially while performing any MapReduce tasks. Apache Pig is a boon for all such programmers. Using Pig Latin, programmers can perform MapReduce tasks easily without having to type complex codes in Java.

Apache Pig uses multi-query approach, thereby reducing the length of codes. For example, an operation that would require you to type 200 lines of code (LoC) in Java can be easily done by typing as less as just 10 LoC in Apache Pig. Ultimately Apache Pig reduces the development time by almost 16 times. Pig Latin is SQL-like language and it is easy to learn Apache Pig when you are familiar with SQL. Apache Pig provides many built-in operators to support data operations like joins, filters, ordering, etc. In addition, it also provides nested data types like tuples, bags, and maps that are missing from MapReduce.

Features of Pig

- Rich set of operators − It provides many operators to perform operations like join, sort, filer, etc.
- Ease of programming − Pig Latin is similar to SQL and it is easy to write a Pig script if you are good at SQL.
- Optimization opportunities − The tasks in Apache Pig optimize their execution automatically, so the programmers need to focus only on semantics of the language.
- Extensibility − Using the existing operators, users can develop their own functions to read, process, and write data.
- UDF's − Pig provides the facility to create User-defined Functions in other programming languages such as Java and invoke or embed them in Pig Scripts.
- Handles all kinds of data − Apache Pig analyzes all kinds of data, both structured as well as unstructured. It stores the results in HDFS.