MINI PROJECT – HADOOP

SPRING 2023

SPECIAL TOPICS DATA SCIENCE

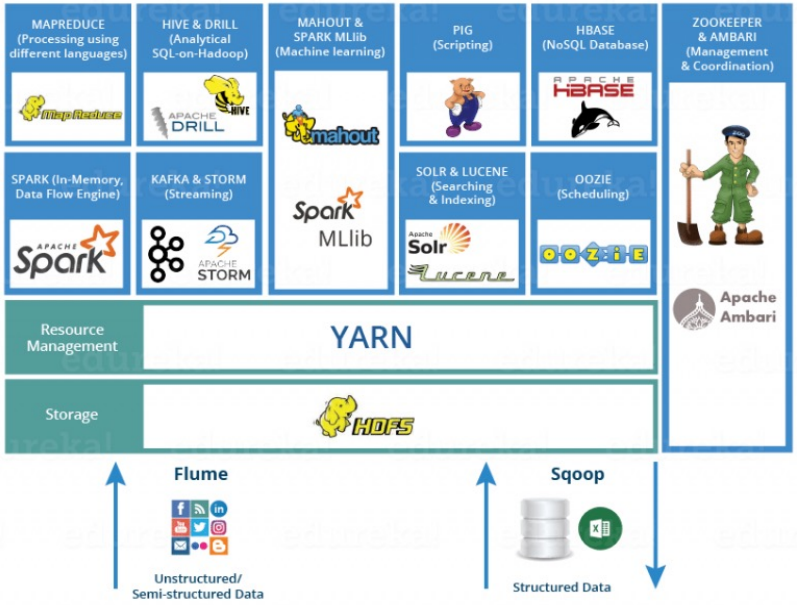
Jiechun Lin
Varshini Yanamandra
Lakshmi Kurapati
Vishal Reddy Mekala

What is Hadoop?

- “Big Data” revolutionized the way data is stored and retrieved over the years.
- The invention of Hadoop begun with the need to handle this Big Data efficiently.
- Challenges with Big Data :
  I. Storing enormous amount of data
  II. Different formats of data - structured, unstructured and semi-structured
  III. Processing speed
- Definition - Hadoop is an open-source software framework for storing and processing large datasets across clusters of computers.

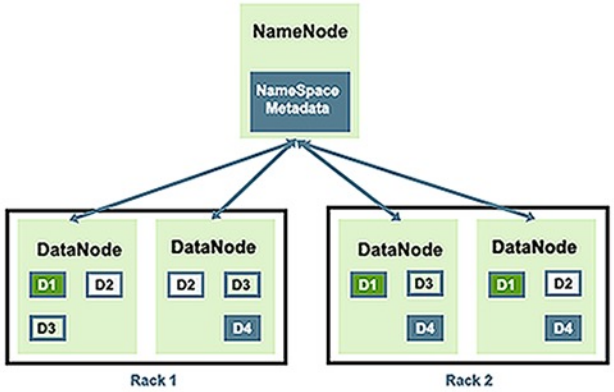
Hadoop Architecture

- Hadoop consists of two main components:
  - Hadoop Distributed File System (HDFS)
  - Yet Another Resource Negotiator (YARN)
- HDFS is a distributed file system that stores data across multiple nodes in the cluster.
  I. Name node or Master node
  II. Data node or Slave node
- YARN is a framework for job scheduling and cluster resource management.
  I. Resource manager
  II. Node manager



Hadoop Data Storage - HDFS

- HDFS spans across a cluster of nodes
- Files divided into smaller blocks of 128 MB or 256 MB – configurable
- Master–Slave architecture
  - Master node called the ‘NameNode’
  - Slave nodes called the ‘DataNodes’
- The NameNode manages the file system metadata
  - File directory structure
  - Block locations
- The DataNodes store the actual data blocks
- Read/write operations:
  - Client gets metadata from the NameNode first
  - Communicates directly with relevant DataNodes to read/write data blocks
  - Data is read and written in batches and parallelly



Indexing In Hadoop

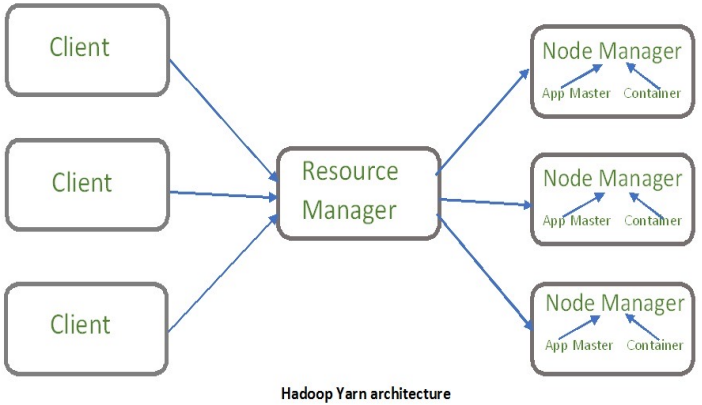
- No built-in indexing mechanisms in Hadoop:
  - Stores data in files without indexing
  - Run a MapReduce job through the whole data to find something
- Choice of index depends on the use-case and type of data
- Cost of regenerating indexes is high – unfeasible for changing data
- Can manually use indexing in HDFS in two ways:
  - File-based Indexing
  - InputSplit Indexing

Indexing In Hadoop

- Some popular indexing mechanisms include:
  - Apache HBase – a NoSQL database built on top of Hadoop
    - Distributed indexing mechanism based on MapReduce
  - Apache Solr
    - Popular for unstructured data like web pages and social media data
    - Hadoop-Solr integration
  - Some other mechanisms available include Apache Lucene, ElasticSearch and ClouderaSearch

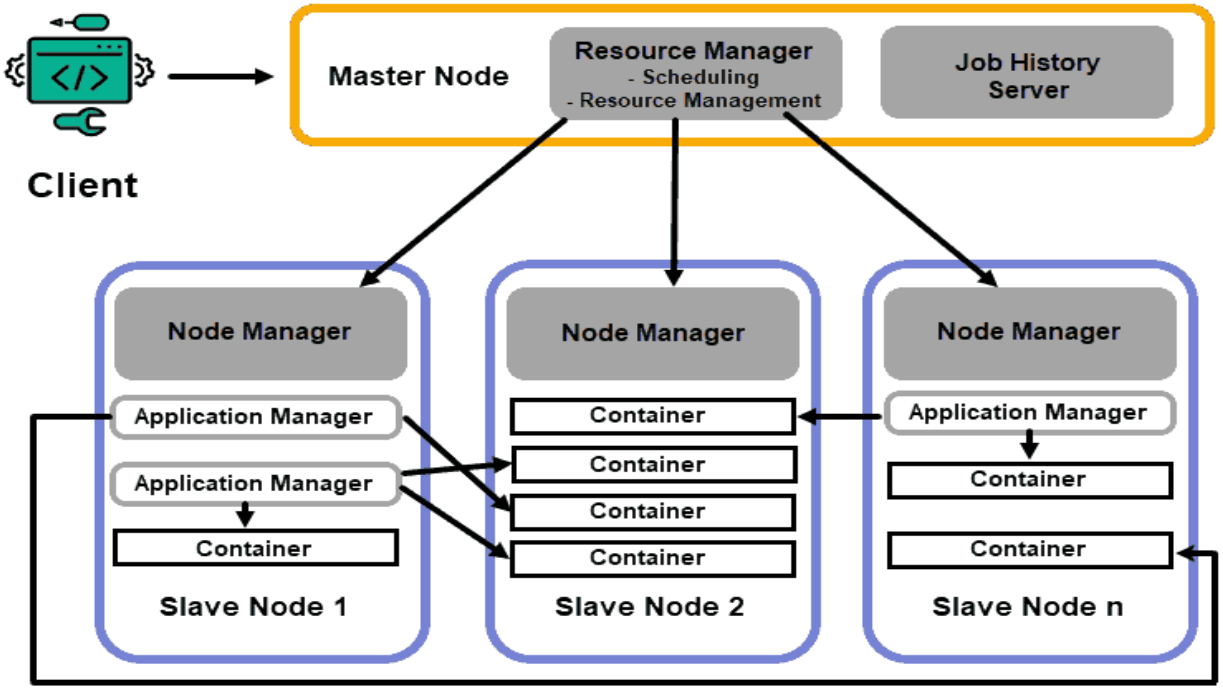
Yet Another Resource Negotiator - YARN

- YARN allows different data processing methods on top of HDFS
- It opens-up Hadoop to other distributed applications beyond MapReduce
- Consists of two main components:
  - I. Resource Manager
    - ➢ Runs on the master node
    - ➢ Scheduler – Responsible for allocating resources to various running applications
    - ➢ Application Manager - Responsible for accepting job submissions and executing the application
  - II. Node Manager
    - ➢ Runs on each data node
    - ➢ Monitors resources on the node
    - ➢ Tracks node health and continuously communicates with resource manager to remain up-to-date



Hadoop Yarn architecture

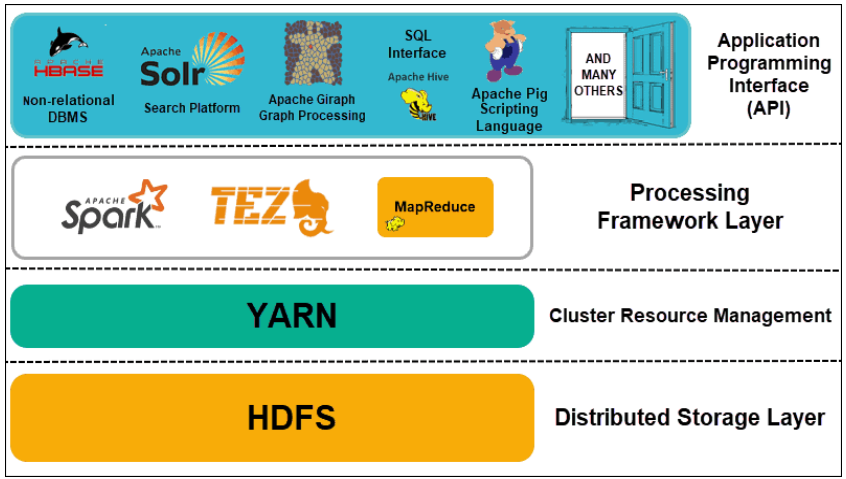
Working of HDFS & YARN



Processing Framework Layer

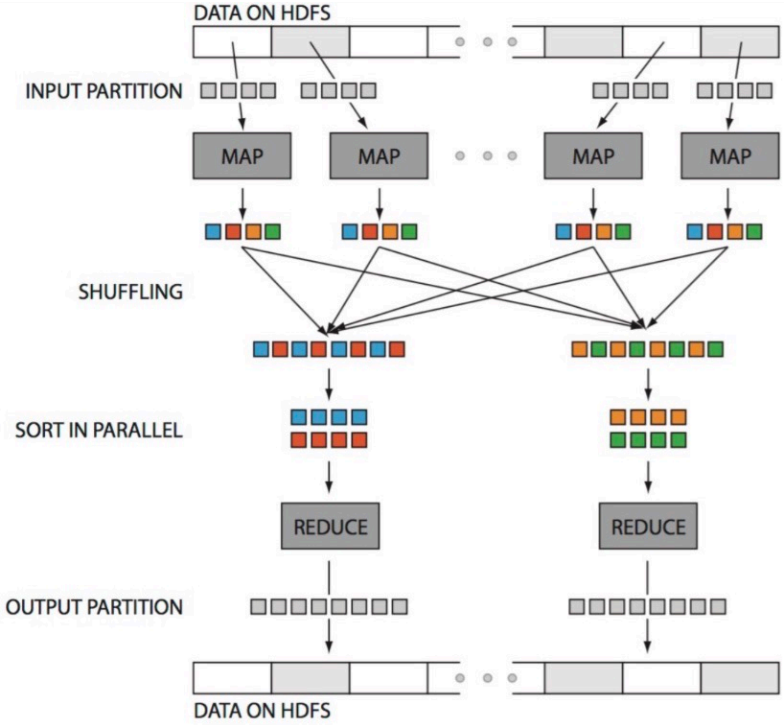
- Let's look at some of the most widely and essential analytics tools that Hadoop can use to improve its reliability and processing to generate new insight into data.

  - MapReduce
  - Spark
  - Hive / Impala / Pig and many more



Processing Framework Layer

- MapReduce
  - Data is processed in two phases – Map and Reduce
  - Mapper reads data from HDFS and converts input into < key , value > pairs
  - The < key , value > pairs are organized and sorted(by key) before reaching the Reducer
  - Reducer accepts these pairs, does the necessary computation and outputs it



Processing Framework Layer

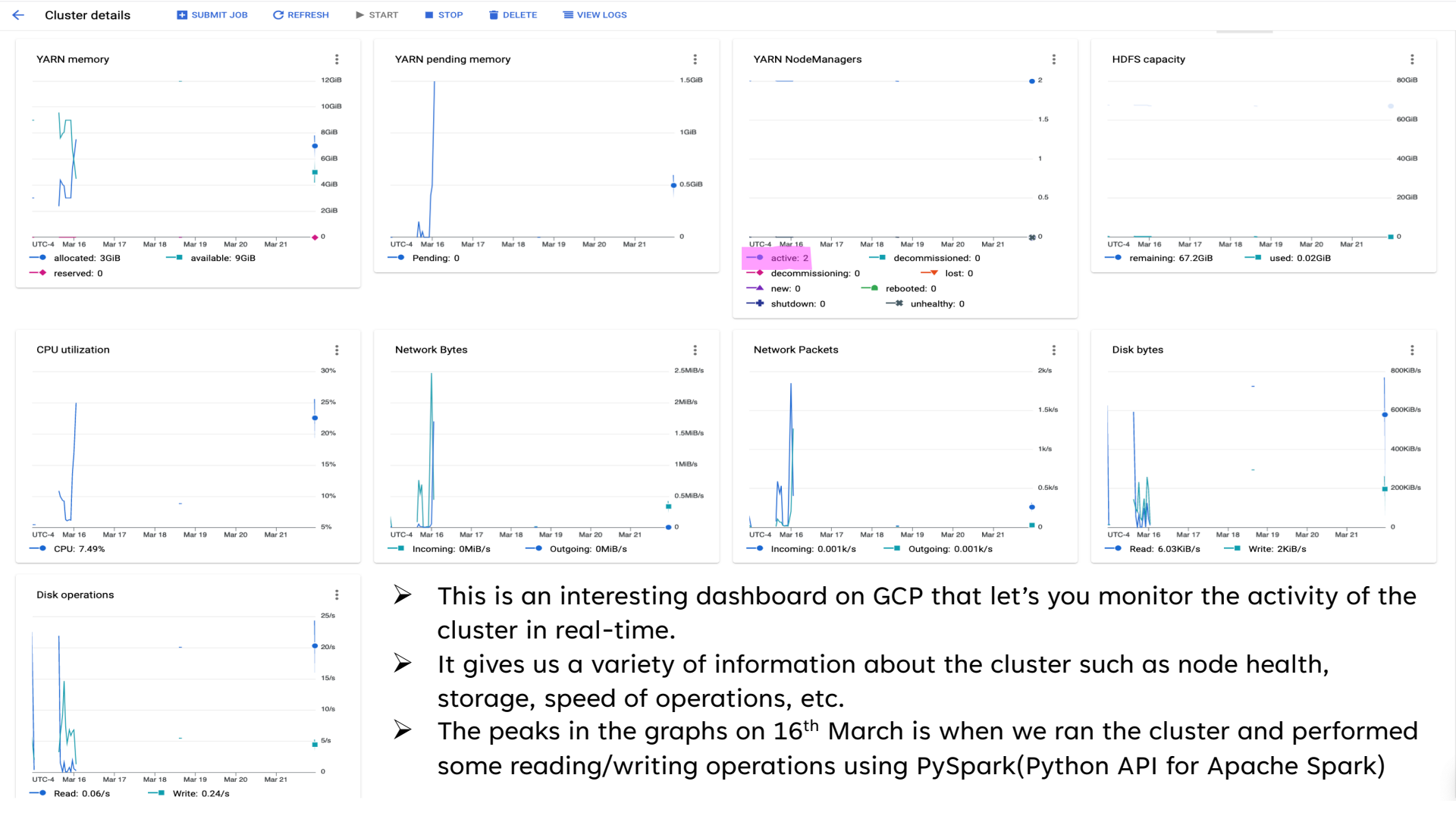
- Spark
  - Each step of MapReduce requires a disk read and write, making it slower due to the latency of disk I/O.
  - Spark addresses the limitations to MapReduce.
  - This is achieved by doing in-memory processing, which reduces the number of steps in a job, and by reusing data across multiple parallel operations.
  - This framework makes real-time analytics possible.
  - For our project, we utilized Hadoop Distributed File System as a storage layer and processed data using Spark, as given below.

HDFS on Google Cloud Platform

- Setting up a Hadoop cluster on actual computers or virtual machines can get cumbersome.
- Dataproc is a service offered by Google to run a Hadoop cluster on the cloud.
- For this project, we used HDFS as a storage layer and performed operations using Spark.
- We've configured our Hadoop cluster to have 2 slave nodes along with the master node.
- The number of nodes and disk size of each is completely configurable.

Master node	Standard (1 master, N workers)
Machine type	n1-standard-2
Number of GPUs	0
Primary disk type	pd-standard
Primary disk size	50GB
Local SSDs	0
Worker nodes	2
Machine type	n1-standard-2
Number of GPUs	0
Primary disk type	pd-standard
Primary disk size	50GB
Local SSDs	0

Fun Fact: Google offers up to $400 free credit on sign-up that you can use for a variety of services on their cloud platform.



- This is an interesting dashboard on GCP that let's you monitor the activity of the cluster in real-time.
- It gives us a variety of information about the cluster such as node health, storage, speed of operations, etc.
- The peaks in the graphs on 16th March is when we ran the cluster and performed some reading/writing operations using PySpark(Python API for Apache Spark)

Features

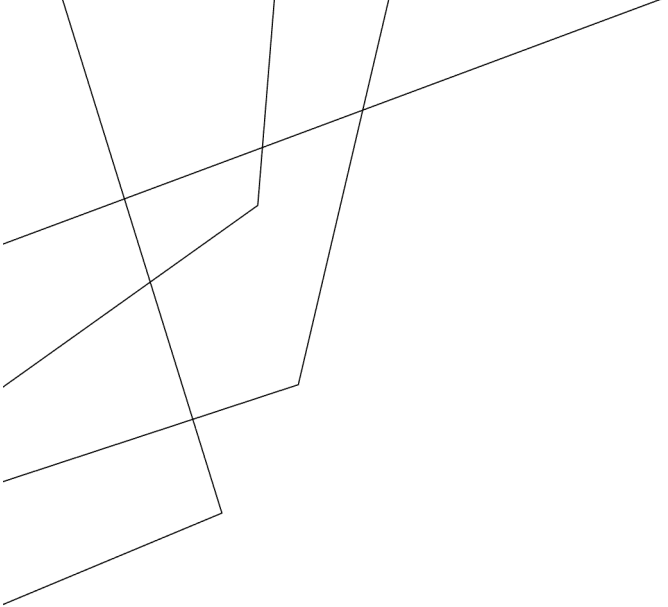
- Reliability - If one machine fails, another machine will take over the responsibility. Hadoop has in-built fault tolerance features making it highly reliable.
- Economical – Hadoop is free and does not have any licensing cost. Also, it's easier to maintain a Hadoop cluster.
- Flexibility – It can store and process a variety of data making it extremely flexible.
- Scalability – Clusters can be scaled both horizontally and vertically without stopping the system. Hence, there's no downtime.

DRAWBACKS

- Support for batch – processing only
- Issue with small files
- Latency
- No Caching
- High complexity and steep learning curve

REFERENCES

1. A Comprehensive Guide to Hadoop
2. MapReduce VS Spark
3. Apache Spark and Hadoop : Working Together
4. A video guide to process data on a Hadoop Cluster using PySpark



THANK YOU!