HADOOP

INTRODUCTION TO HADOOP

Hadoop is an open source framework overseen by Apache Software Foundation which is written in Java for storing and processing of huge datasets with the cluster of commodity hardware. While Hadoop is sometimes referred to as an acronym for High Availability Distributed Object Oriented Platform, it was originally named after Cutting's son's toy elephant.

There are mainly two problems with the big data. First one is to store such a huge amount of data and the second one is to process that stored data. The traditional approach like RDBMS is not sufficient due to the heterogeneity of the data. So Hadoop comes as the solution to the problem of big data i.e. storing and processing the big data with some extra capabilities. There are mainly two components of Hadoop which are Hadoop Distributed File System (HDFS) and Yet Another Resource Negotiator(YARN).

HISTORY OF HADOOP

Hadoop, an open-source framework for distributed data processing, originated from the Nutch project, an open-source web search engine created by Doug Cutting and Mike Cafarella. The need to handle the massive scale of the internet led them to draw inspiration from Google’s 2003-2004 publications on the Google File System (GFS) and MapReduce.

In 2005, the Hadoop project officially began, named after Cutting’s son’s toy elephant. By 2006, Hadoop had become a subproject of Apache Lucene, focusing on the development of the Hadoop Distributed File System (HDFS) and MapReduce. With significant contributions from Yahoo!, Hadoop rapidly grew, and by 2008, Yahoo! operated the largest Hadoop cluster in the world. The framework gained widespread adoption in the tech industry, with companies like Facebook, Twitter, and LinkedIn using it to manage their data. As Hadoop matured, the ecosystem expanded with tools like Pig, Hive, and later, HBase, YARN, and Spark. While newer technologies have emerged, Hadoop remains a cornerstone in big data analytics, enabling organizations to efficiently store, process, and analyze vast amounts of data.

VERSIONS OF HADOOP

Hadoop 0.x Series

- **0.1.0** (April 2006): The first release of Hadoop, primarily focused on the basic implementation of the Hadoop Distributed File System (HDFS) and the MapReduce programming model.
- **0.20.0** (April 2009): Introduced several significant features, including the new API for MapReduce, improved job scheduling, and more robust HDFS.

Hadoop 1.x Series

- **1.0.0** (December 2011): Marked the first stable release of Hadoop. Key features included:
  - HDFS: Reliable and scalable storage.
  - MapReduce: The core data processing engine.
  - JobTracker and TaskTracker: Components for managing and tracking jobs and tasks.
- **1.2.1** (August 2013): Provided stability and bug fixes, improving the reliability of the Hadoop 1.x series.

Hadoop 2.x Series

- **2.0.0-alpha** (October 2012): Introduced YARN (Yet Another Resource Negotiator), which decoupled resource management and job scheduling/monitoring functions from the MapReduce layer. This allowed Hadoop to support other data processing models beyond MapReduce.
- **2.2.0** (October 2013): Marked the first stable release of Hadoop 2.x with YARN.
- **2.7.0** (April 2015): Introduced several improvements in HDFS, YARN, and MapReduce, including enhanced security, performance optimizations, and new APIs.
- **2.9.0** (November 2017): Added features like erasure coding for HDFS to reduce storage overhead, improved support for containerized applications in YARN, and other enhancements.

Hadoop 3.x Series

- **3.0.0** (December 2017): Brought significant advancements, including:
  - Erasure Coding: Improved storage efficiency for HDFS.
  - Docker Support: Native support for Docker containers in YARN.

    - Improved HDFS: Enhanced scalability and support for multiple NameNodes.
    - More efficient resource management and scheduling in YARN.
- **3.1.0** (April 2018): Included enhancements like in-place upgrades for HDFS, improved scalability, and performance improvements for YARN.
- **3.2.0** (January 2019): Added features like HDFS federation improvements, better container management in YARN, and enhanced support for cloud storage.
- **3.3.0** (July 2020): Introduced features such as Hadoop Ozone, a new distributed storage system designed for cloud-native environments, and further improvements in HDFS and YARN.

Hardware Requirements

Minimum Requirements (For Development/Small Scale Testing)

- **CPU**: Dual-core processor
- **RAM**: 8 GB
- **Storage**: 100 GB
- **Network**: Gigabit Ethernet

Recommended Requirements (For Production/Cluster Deployment)

- **Master Node (NameNode, ResourceManager)**
    - **CPU**: Quad-core processor or better
    - **RAM**: 16-64 GB (depends on cluster size)
    - **Storage**: SSDs recommended for faster access, 1 TB or more
    - **Network**: High-speed network (Gigabit Ethernet or higher)
- **Slave Nodes (DataNodes, NodeManagers)**
    - **CPU**: Quad-core processor or better
    - **RAM**: 16-64 GB (depends on workload)
    - **Storage**: Multiple high-capacity disks (HDDs or SSDs), 1 TB or more
    - **Network**: High-speed network (Gigabit Ethernet or higher)

Software Requirements

Operating System

- **Linux** (Preferred distributions: CentOS, Ubuntu, Debian, Red Hat)
- **Windows** (Supported but less common in production environments)

Java

- **Java Development Kit (JDK)**: Version 8 or higher (Hadoop is primarily developed in Java)
    - Make sure JAVA_HOME environment variable is set correctly.

SSH

- **Secure Shell (SSH)**: Password-less SSH access must be configured between nodes in the cluster.

Hadoop Distribution

- **Hadoop Version**: Ensure compatibility with other software components.
  - Download the appropriate Hadoop version from Apache Hadoop.

Configuration Recommendations

- **Memory Allocation**: Allocate enough memory to HDFS and YARN based on the available RAM and expected workload.
- **Storage Configuration**: Use RAID configurations for data redundancy and performance improvements. SSDs are preferred for critical components like the NameNode.
- **Network Setup**: Ensure a high-speed, reliable network setup. In a large cluster, consider using multiple network interfaces for better performance.

Example Deployment Sizes

Small Deployment

- **Nodes**: 1-10
- **RAM per Node**: 16 GB
- **Storage per Node**: 1 TB
- **Use Case**: Development, testing, small-scale data processing

Medium Deployment

- **Nodes**: 10-50
- **RAM per Node**: 32 GB
- **Storage per Node**: 2-4 TB
- **Use Case**: Medium-scale data processing, business analytics

Large Deployment

- **Nodes**: 50-100+
- **RAM per Node**: 64 GB or more
- **Storage per Node**: 4-10 TB
- **Use Case**: Large-scale data processing, enterprise-level data analytics, real-time data processing