Hadoop Architecture Explained-What it is and why it matters

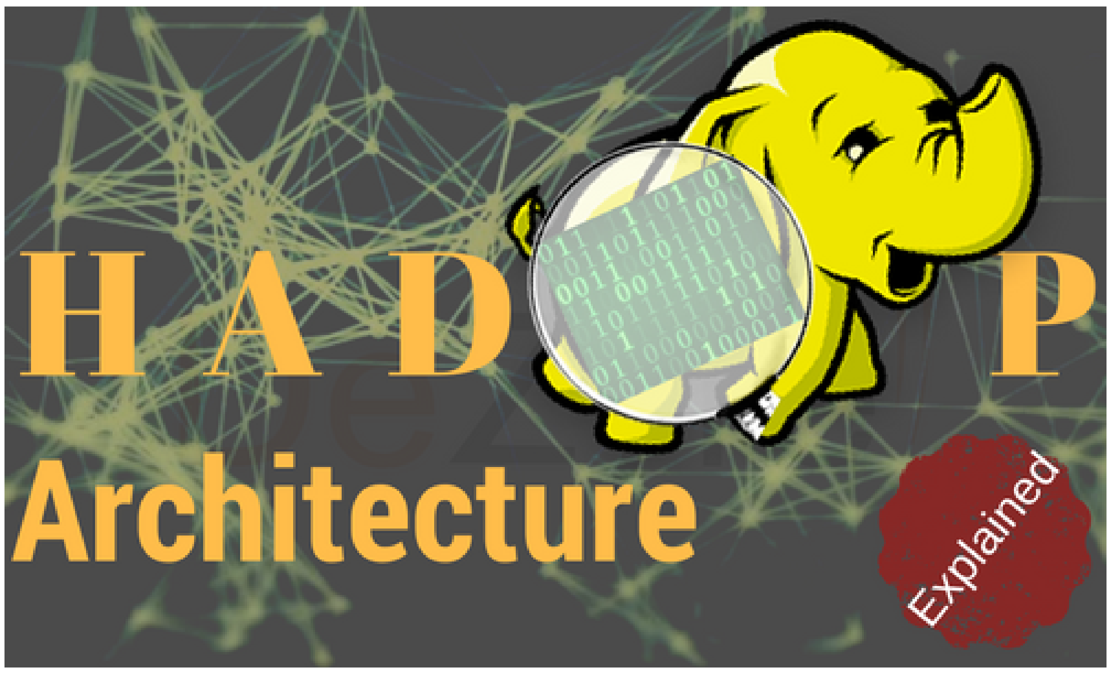
07 Nov 2016

Last Update made on March 20, 2018

Hortonworks founder predicted that by end of 2020, 75% of Fortune 2000 companies will be running 1000 node hadoop clusters in production. The tiny toy elephant in the big data room has become the most popular big data solution across the globe. However, implementation of Hadoop in production is still accompanied by deployment and management challenges like scalability, flexibility and cost effectiveness.

Many organizations that venture into enterprise adoption of Hadoop by business users or by an analytics group within the company do not have any knowledge on how a good hadoop architecture design should be and how actually a hadoop cluster works in production. This lack of knowledge leads to design of a hadoop cluster that is more complex than is necessary for a particular big data application making it a pricey implementation. Apache Hadoop was developed with the purpose of having a low–cost, redundant data store that would allow organizations to leverage big data analytics at economical cost and maximize profitability of the business.

A good hadoop architectural design requires various design considerations in terms of computing power, networking and storage. This blog post gives an in-depth explanation of the Hadoop architecture and the factors to be considered when designing and building a Hadoop cluster for production success.



Hadoop Architecture Overview

Apache Hadoop offers a scalable, flexible and reliable distributed computing big data framework for a cluster of systems with storage capacity and local computing power by leveraging commodity hardware. Hadoop follows a Master Slave architecture for the transformation and analysis of large datasets using Hadoop MapReduce paradigm. The 3 important hadoop components that play a vital role in the Hadoop architecture are -

i. Hadoop Distributed File System (HDFS) – Patterned after the UNIX file system
ii. Hadoop MapReduce
iii. Yet Another Resource Negotiator (YARN)

Hadoop Architecture Explained

Hadoop skillset requires thoughtful knowledge of every layer in the hadoop stack right from understanding about the various components in the hadoop architecture, designing a hadoop cluster, performance tuning it and setting up the top chain responsible for data processing.

Hadoop follows a master slave architecture design for data storage and distributed data processing using HDFS and MapReduce respectively. The master node for data storage is hadoop HDFS is the NameNode and the master node for parallel processing of data using Hadoop MapReduce is the Job Tracker. The slave nodes in the hadoop architecture are the other machines in the Hadoop cluster which store data and perform complex computations. Every slave node has a Task Tracker daemon and a DataNode that synchronizes the processes with the Job Tracker and NameNode respectively. In Hadoop architectural implementation the master or slave systems can be setup in the cloud or on-premise.

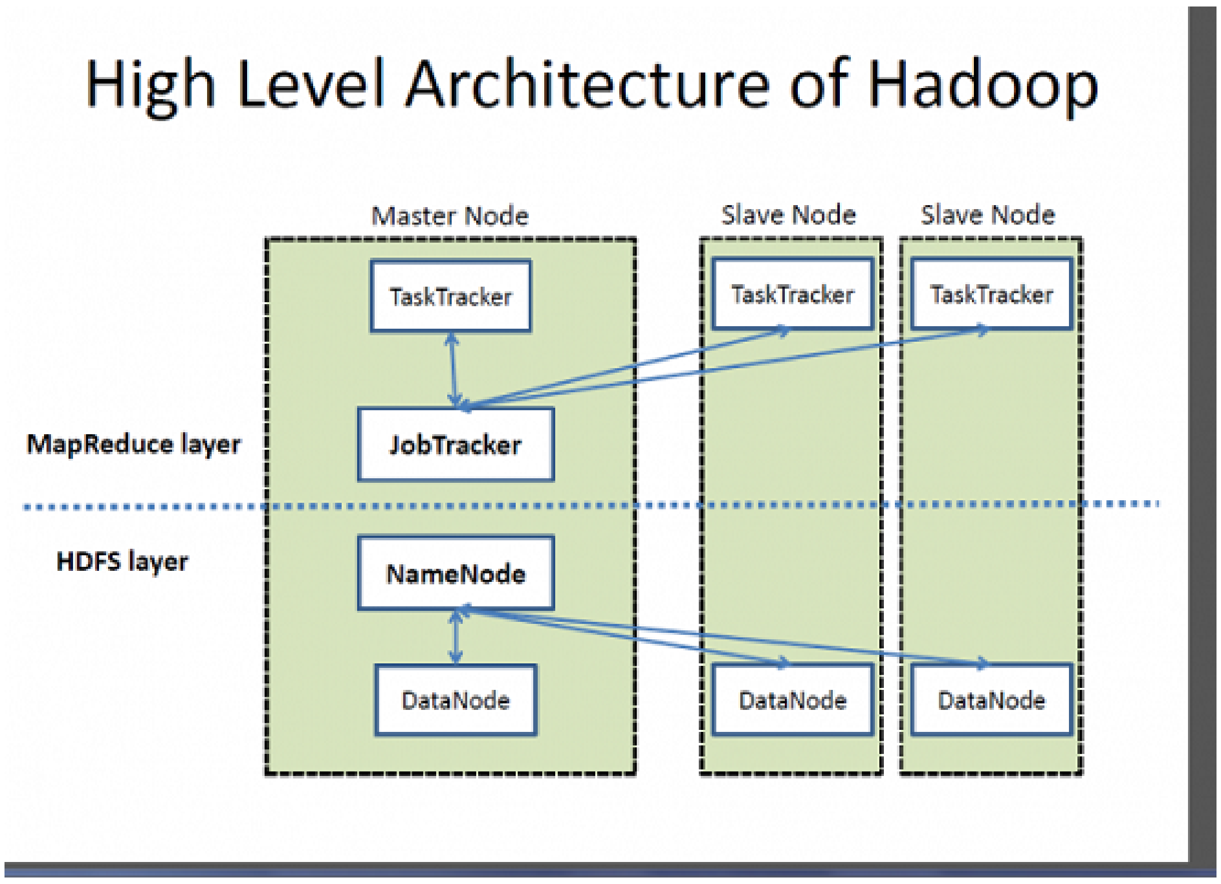


Image Credit : OpenSource.com

If you would like more information about Big Data and Hadoop Certification training, please click the orange "Request Info" button on top of this page.

Role of Distributed Storage - HDFS in Hadoop Application Architecture Implementation

A file on HDFS is split into multiple bocks and each is replicated within the Hadoop cluster. A block on HDFS is a blob of data within the underlying file system with a default size of 64MB.The size of a block can be extended up to 256 MB based on the requirements.

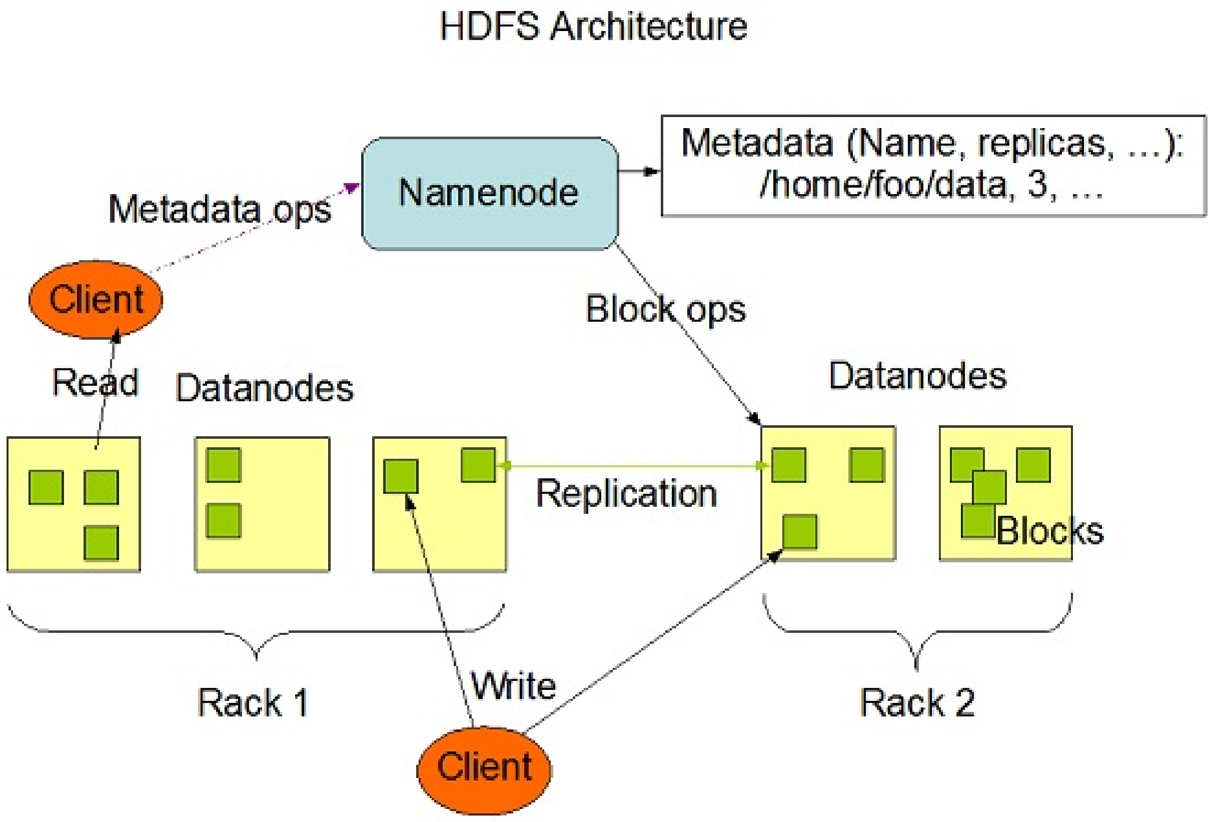


Image Credit :Apache.org

Hadoop Distributed File System (HDFS) stores the application data and file system metadata separately on dedicated servers. NameNode and DataNode are the two critical components of the Hadoop HDFS architecture. Application data is stored on servers referred to as DataNodes and file system metadata is stored on servers referred to as NameNode. HDFS replicates the file content on multiple DataNodes based on the replication factor to ensure reliability of data. The NameNode and DataNode communicate with each other using TCP based protocols. For the Hadoop architecture to be performance efficient, HDFS must satisfy certain pre-requisites –

- All the hard drives should have a high throughput.
- Good network speed to manage intermediate data transfer and block replications.

NameNode

All the files and directories in the HDFS namespace are represented on the NameNode by Inodes that contain various attributes like permissions, modification timestamp, disk space quota, namespace quota and access times. NameNode maps the entire file system structure into memory. Two files fsimage and edits are used for persistence during restarts.

- Fsimage file contains the Inodes and the list of blocks which define the metadata.It has a complete snapshot of the file systems metadata at any given point of time.
- The edits file contains any modifications that have been performed on the content of the fsimage file.Incremental changes like renaming or appending data to the file are stored in the edit log to ensure durability instead of creating a new fsimage snapshot everytime the namespace is being altered.

When the NameNode starts, fsimage file is loaded and then the contents of the edits file are applied to recover the latest state of the file system. The only problem with this is that over the time the edits file grows and consumes all the disk space resulting in slowing down the restart process. If the hadoop cluster has not been restarted for months together then there will be a huge downtime as the size of the edits file will be increase. This is when Secondary NameNode comes to the rescue. Secondary NameNode gets the fsimage and edits log from the primary NameNode at regular intervals and loads both the fsimage and edit logs file to the main memory by applying each operation from edits log file to fsimage. Secondary NameNode copies the new fsimage file to the primary NameNode and also will update the modified time of the fsimage file to fstime file to track when then fsimage file has been updated.

DataNode

DataNode manages the state of an HDFS node and interacts with the blocks .A DataNode can perform CPU intensive jobs like semantic and language analysis, statistics and machine learning tasks, and I/O intensive jobs like clustering, data import, data export, search, decompression, and indexing. A DataNode needs lot of I/O for data processing and transfer.

On startup every DataNode connects to the NameNode and performs a handshake to verify the namespace ID and the software version of the DataNode. If either of them does not match then the DataNode shuts down automatically. A DataNode verifies the block replicas in its ownership by sending a block report to the NameNode. As soon as the DataNode registers, the first block report is sent. DataNode sends heartbeat to the NameNode every 3 seconds to confirm that the DataNode is operating and the block replicas it hosts are available.

Role of Distributed Computation - MapReduce in Hadoop Application Architecture Implementation

The heart of the distributed computation platform Hadoop is its java-based programming paradigm Hadoop MapReduce. Map or Reduce is a special type of directed acyclic graph that can be applied to a wide range of business use cases. Map function transforms the piece of data into key-value pairs and then the keys are sorted where a reduce function is applied to merge the values based on the key into a single output.

How does the Hadoop MapReduce architecture work?

The execution of a MapReduce job begins when the client submits the job configuration to the Job Tracker that specifies the map, combine and reduce functions along with the location for input and output data. On receiving the job configuration, the job tracker identifies the number of splits based on the input path and select Task Trackers based on their network vicinity to the data sources. Job Tracker sends a request to the selected Task Trackers.

The processing of the Map phase begins where the Task Tracker extracts the input data from the splits. Map function is invoked for each record parsed by the “InputFormat” which produces key-value pairs in the memory buffer. The memory buffer is then sorted to different reducer nodes by invoking the combine function. On completion of the map task, Task Tracker notifies the Job Tracker. When all Task Trackers are done, the Job Tracker notifies the selected Task Trackers to begin the reduce phase. Task Tracker reads the region files and sorts the key-value pairs for each key. The reduce function is then invoked which collects the aggregated values into the output file.

Hadoop Architecture Design – Best Practices to Follow

- Use good-quality commodity servers to make it cost efficient and flexible to scale out for complex business use cases. One of the best configurations for Hadoop architecture is to begin with 6 core processors, 96 GB of memory and 1 0 4 TB of local hard drives. This is just a good configuration but not an absolute one.
- For faster and efficient processing of data, move the processing in close proximity to data instead of separating the two.
- Hadoop scales and performs better with local drives so use Just a Bunch of Disks (JBOD) with replication instead of redundant array of independent disks (RAID).
- Design the Hadoop architecture for multi-tenancy by sharing the compute capacity with capacity scheduler and share HDFS storage.
- Do not edit the metadata files as it can corrupt the state of the Hadoop cluster.

Facebook Hadoop Architecture

With 1.59 billion accounts (approximately 1/5th of worlds total population) , 30 million FB users updating their status at least once each day, 10+ million videos uploaded every month, 1+ billion content pieces shared every week and more than 1 billion photos uploaded every month – Facebook uses hadoop to interact with petabytes of data. Facebook runs world's largest Hadoop Cluster with more than 4000 machine storing hundreds of millions of gigabytes of data. The biggest hadoop cluster at Facebook has about 2500 CPU cores and 1 PB of disk space and the engineers at Facebook load more than 250 GB of compressed data (is greater than 2 TB of uncompressed data) into HDFS daily and there are 100's of hadoop jobs running daily on these datasets.

Where is the data stored at Facebook?

135 TB of compressed data is scanned daily and 4 TB compressed data is added daily. Wondering where is all this data stored? Facebook has a Hadoop/Hive warehouse with two level network topology having 4800 cores, 5.5 PB storing up to 12TB per node. 7500+ hadoop hive jobs run in production cluster per day with an average of 80K compute hours. Non-engineers i.e. analysts at Facebook use Hadoop through hive and aprroximately 200 people/month run jobs on Apache Hadoop.

Hadoop/Hive warehouse at Facebook uses a two level network topology -

- 4 Gbit/sec to top level rack switch
- 1 Gbit/sec from node to rack switch

Yahoo Hadoop Architecture

Hadoop at Yahoo has 36 different hadoop clusters spread across Apache HBase, Storm and YARN, totalling 60,000 servers made from 100's of different hardware configurations built up over generations.Yahoo runs the largest multi-tenant hadoop installation in the world withh broad set of use cases. Yahoo runs 850,000 hadoop jobs daily.

For organizations planning to implement hadoop architecture in production, the best way to determine whether Hadoop is right for their company is - to determine the cost of storing and processing data using Hadoop. Compare the determined cost to the cost of legacy approach for managing data.