ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

HADOOP

K.Nagaraju
B.Tech Student,
Department of CSE,
Sphoorthy Engineering College,
Nadergul (Vill.), Sagar Road,
Saroonagar (Mdl), R.R Dist.T.S.

J.Deepthi
Associate Professor & HOD,
Department of CSE,
Sphoorthy Engineering College,
Nadergul (Vill.), Sagar Road,
Saroonagar (Mdl), R.R Dist.T.S.

Mr.T.Pavan Kumar
Assistant Professor,
Department of CSE,
Sphoorthy Engineering College,
Nadergul (Vill.), Sagar Road,
Saroonagar (Mdl), R.R Dist.T.S.

Apache Hadoop is an open-source software framework used for distributed storage and processing of very large data sets. It consists of computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are a common occurrence and should be automatically handled by the framework.[2]. The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part called MapReduce. Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel. This approach takes advantage of data locality[3] – nodes manipulating the data they have access to – to allow the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking.[4] The base Apache Hadoop framework is composed of the following modules:

Hadoop Common:
Contains libraries and utilities needed by other Hadoop modules;

Hadoop Distributed File System (HDFS):
A distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster;

Hadoop YARN:
A resource-management platform responsible for managing computing resources in clusters and using them for scheduling of users' applications;[5][6] and

Hadoop MapReduce:
An implementation of the MapReduce programming model for large scale data processing. The term Hadoop has come to refer not just to the base modules above, but also to the ecosystem,[7] or collection of additional software packages that can be installed on top of or alongside Hadoop, such as Apache Pig, Apache Hive, Apache HBase, Apache Phoenix, Apache Spark, Apache ZooKeeper, Cloudera Impala, Apache Flume, Apache Sqoop, Apache Oozie, Apache Storm.[8] Apache Hadoop's MapReduce and HDFS components were inspired by Google papers on their MapReduce and Google File System.[9]

The Hadoop framework itself is mostly written in the Java programming language, with some native code in C and command line utilities written as shell scripts. Though MapReduce Java code is common, any programming language can be used with "Hadoop Streaming" to implement the "map" and "reduce" parts of the user's program.[10] Other projects in the Hadoop ecosystem expose richer user interfaces.

History:
The genesis of Hadoop came from the Google File System paper[11] that was published in October 2003. This paper spawned another research paper from Google – MapReduce: Simplified Data Processing on Large Clusters.[12] Development started on the Apache Nutch project, but was moved to the new Hadoop subproject in January 2006.[13] Doug Cutting, who was working at Yahoo! at the time,[14] named it after his son's toy elephant.[15] The initial code that was factored out of Nutch consisted of 5k lines of code for NDFS and 6k lines of code for MapReduce.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The first committer added to the Hadoop project was Owen O'Malley in March 2006.[16] Hadoop 0.1.0 was released in April 2006[17] and continues to evolve by the many contributors[18] to the Apache Hadoop project.

File Systems:

Hadoop Distributed File System:

The Hadoop distributed file system (HDFS) is a distributed, scalable, and portable file system written in Java for the Hadoop framework. Some consider HDFS to instead be a data store due to its lack of POSIX compliance and inability to be mounted,[62] but it does provide shell commands and Java API methods that are similar to other file systems.[63] A Hadoop cluster has nominally a single namenode plus a cluster of datanodes, although redundancy options are available for the namenode due to its criticality. Each datanode serves up blocks of data over the network using a block protocol specific to HDFS. The file system uses TCP/IP sockets for communication. Clients use remote procedure call (RPC) to communicate between each other.

HDFS stores large files (typically in the range of gigabytes to terabytes[64]) across multiple machines. It achieves reliability by replicating the data across multiple hosts, and hence theoretically does not require RAID storage on hosts (but to increase I/O performance some RAID configurations are still useful). With the default replication value, 3, data is stored on three nodes: two on the same rack, and one on a different rack. Data nodes can talk to each other to rebalance data, to move copies around, and to keep the replication of data high. HDFS is not fully POSIX-compliant, because the requirements for a POSIX file-system differ from the target goals for a Hadoop application. The trade-off of not having a fully POSIX-compliant file-system is increased performance for data throughput and support for non-POSIX operations such as Append.[65]

HDFS added the high-availability capabilities, as announced for release 2.0 in May 2012,[66] letting the main metadata server (the NameNode) fail over manually to a backup. The project has also started developing automatic fail-over. The HDFS file system includes a so-called secondary namenode, a misleading name that some might incorrectly interpret as a backup namenode for when the primary namenode goes offline. In fact, the secondary namenode regularly connects with the primary namenode and builds snapshots of the primary namenode's directory information, which the system then saves to local or remote directories. These checkpointed images can be used to restart a failed primary namenode without having to replay the entire journal of file-system actions, then to edit the log to create an up-to-date directory structure. Because the namenode is the single point for storage and management of metadata, it can become a bottleneck for supporting a huge number of files, especially a large number of small files.

HDFS Federation, a new addition, aims to tackle this problem to a certain extent by allowing multiple namespaces served by separate namenodes. Moreover, there are some issues in HDFS, namely, small file issue, scalability problem, Single Point of Failure (SPoF), and bottleneck in huge metadata request. An advantage of using HDFS is data awareness between the job tracker and task tracker. The job tracker schedules map or reduce jobs to task trackers with an awareness of the data location. For example: if node A contains data (x,y,z) and node B contains data (a,b,c), the job tracker schedules node B to perform map or reduce tasks on (a,b,c) and node A would be scheduled to perform map or reduce tasks on (x,y,z). This reduces the amount of traffic that goes over the network and prevents unnecessary data transfer. When Hadoop is used with other file systems, this advantage is not always available. This can have a significant impact on job-completion times, which has been demonstrated when running data-intensive jobs.[67]

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

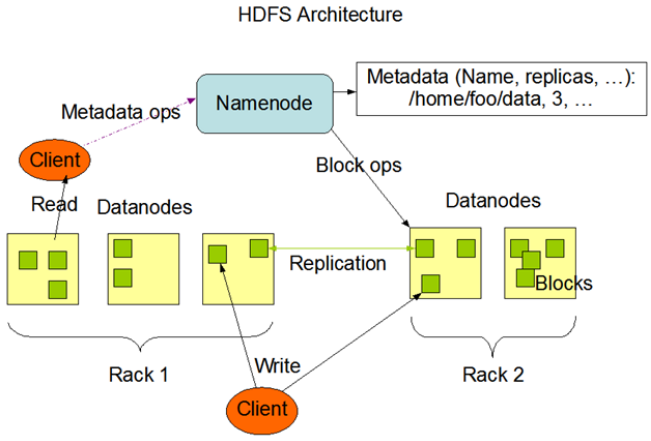
HDFS was designed for mostly immutable files[65] and may not be suitable for systems requiring concurrent write-operations. HDFS can be mounted directly with a Filesystem in Userspace (FUSE) virtual file system on Linux and some other Unix systems. File access can be achieved through the native Java application programming interface (API), the Thrift API to generate a client in the language of the users' choosing (C++, Java, Python, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, Smalltalk, and OCaml), the command-line interface, browsed through the HDFS-UI Web application (webapp) over HTTP, or via 3rd-party network client libraries.[68]. HDFS is designed for portability across various hardware platforms and compatibility with a variety of underlying operating systems.

The HDFS design introduces portability limitations that result in some performance bottlenecks, since the Java implementation can't use features that are exclusive to the platform on which HDFS is running.[69] Due to its widespread integration into enterprise-level infrastructures, monitoring HDFS performance at scale has become an increasingly important issue. Monitoring end-to-end performance requires tracking metrics from datanodes, namenodes, and the underlying operating system.[70] There are currently several monitoring platforms to track HDFS performance, including HortonWorks, Cloudera, and Datadog.

Features of HDFS:

- It is suitable for the distributed storage and processing.
- Hadoop provides a command interface to interact with HDFS.
- The built-in servers of namenode and datanode help users to easily check the status of cluster.
- Streaming access to file system data.
- HDFS provides file permissions and authentication.

HDFS Architecture:



Namenode:

The namenode is the commodity hardware that contains the GNU/Linux operating system and the namenode software. It is a software that can be run on commodity hardware. The system having the namenode acts as the master server and it does the following tasks:

- Manages the file system namespace.
- Regulates client's access to files.
- It also executes file system operations such as renaming, closing, and opening files and directories.

Datanode:

The datanode is a commodity hardware having the GNU/Linux operating system and datanode software. For every node (Commodity hardware/System) in a cluster, there will be a datanode. These nodes manage the data storage of their system.

- Datanodes perform read-write operations on the file systems, as per client request.
- They also perform operations such as block creation, deletion, and replication according to the instructions of the namenode.

Block:

Generally the user data is stored in the files of HDFS. The file in a file system will be divided into one or more segments and/or stored in individual data nodes. These file segments are called as blocks.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In other words, the minimum amount of data that HDFS can read or write is called a Block. The default block size is 64MB, but it can be increased as per the need to change in HDFS configuration.

Goals of HDFS:

- **Fault Detection and Recovery:** Since HDFS includes a large number of commodity hardware, failure of components is frequent. Therefore HDFS should have mechanisms for quick and automatic fault detection and recovery.

- **Huge datasets:** HDFS should have hundreds of nodes per cluster to manage the applications having huge datasets.

- **Hardware at data:** A requested task can be done efficiently, when the computation takes place near the data. Especially where huge datasets are involved, it reduces the network traffic and increases the throughput.

Other file systems:

Hadoop works directly with any distributed file system that can be mounted by the underlying operating system simply by using a file:// URL; however, this comes at a price, the loss of locality. To reduce network traffic, Hadoop needs to know which servers are closest to the data; this is information that Hadoop-specific file system bridges can provide. In May 2011, the list of supported file systems bundled with Apache Hadoop were:

HDFS: Hadoop's own rack-aware file system.[71] This is designed to scale to tens of petabytes of storage and runs on top of the file systems of the underlying operating systems. FTP File system: this stores all its data on remotely accessible FTP servers.

Amazon S3 (Simple Storage Service) File System:

This is targeted at clusters hosted on the Amazon Elastic Compute Cloud server-on-demand infrastructure. There is no rack-awareness in this file system, as it is all remote.

Windows Azure Storage Blobs (WASB) File System:

WASB, an extension on top of HDFS, allows distributions of Hadoop to access data in Azure blob stores without moving the data permanently into the cluster. A number of third-party file system bridges have also been written, none of which are currently in Hadoop distributions. However, some commercial distributions of Hadoop ship with an alternative filesystem as the default – specifically IBM and MapR.

JobTracker and TaskTracker: the MapReduce engine Main article: MapReduce

Above the file systems comes the MapReduce Engine, which consists of one JobTracker, to which client applications submit MapReduce jobs. The JobTracker pushes work out to available TaskTracker nodes in the cluster, striving to keep the work as close to the data as possible. With a rack-aware file system, the JobTracker knows which node contains the data, and which other machines are nearby. If the work cannot be hosted on the actual node where the data resides, priority is given to nodes in the same rack. This reduces network traffic on the main backbone network. If a TaskTracker fails or times out, that part of the job is rescheduled. The TaskTracker on each node spawns a separate Java Virtual Machine process to prevent the TaskTracker itself from failing if the running job crashes its JVM. A heartbeat is sent from the TaskTracker to the JobTracker every few minutes to check its status. The Job Tracker and TaskTracker status and information is exposed by Jetty and can be viewed from a web browser.

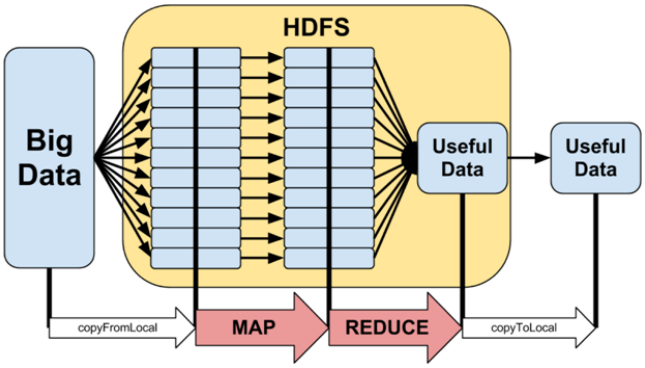
Known limitations of this approach are:

The allocation of work to TaskTrackers is very simple. Every TaskTracker has a number of available slots (such as "4 slots"). Every active map or reduce task takes up one slot. The Job Tracker allocates work to the tracker nearest to the data with an available slot.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

There is no consideration of the current system load of the allocated machine, and hence its actual availability. If one TaskTracker is very slow, it can delay the entire MapReduce job – especially towards the end of a job, where everything can end up waiting for the slowest task. With speculative execution enabled, however, a single task can be executed on multiple slave nodes.

Map Reduce Workflow:



Scheduling:

By default Hadoop uses FIFO scheduling, and optionally 5 scheduling priorities to schedule jobs from a work queue.[77] In version 0.19 the job scheduler was refactored out of the JobTracker, while adding the ability to use an alternate scheduler (such as the Fair scheduler or the Capacity scheduler, described next).[78]

Fair Scheduler:

The fair scheduler was developed by Facebook.[79] The goal of the fair scheduler is to provide fast response times for small jobs and QoS for production jobs. The fair scheduler has three basic concepts.[80]

Jobs are grouped into pools.
Each pool is assigned a guaranteed minimum share.
Excess capacity is split between jobs.

By default, jobs that are uncategorized go into a default pool. Pools have to specify the minimum number of map slots, reduce slots, and a limit on the number of running jobs.

Capacity Scheduler:

The capacity scheduler was developed by Yahoo. The capacity scheduler supports several features that are similar to the fair scheduler.[81] Queues are allocated a fraction of the total resource capacity. Free resources are allocated to queues beyond their total capacity. Within a queue a job with a high level of priority has access to the queue's resources. There is no pre-emption once a job is running.

Other applications:

The HDFS file system is not restricted to MapReduce jobs. It can be used for other applications, many of which are under development at Apache. The list includes the HBase database, the Apache Mahout machine learning system, and the Apache Hive Data Warehouse system. Hadoop can in theory be used for any sort of work that is batch-oriented rather than real-time, is very data-intensive, and benefits from parallel processing of data. It can also be used to complement a real-time system, such as lambda architecture, Apache Storm, Flink and Spark Streaming.[82]

As of October 2009, commercial applications of Hadoop[83] included:

log and/or clickstream analysis of various kinds marketing analytics machine learning and/or sophisticated data mining image processing processing of XML messages web crawling and/ or text processing general archiving, including of relational/ tabular data, e.g. for compliance

Prominent users:

On February 19, 2008, Yahoo! Inc. launched what it claimed was the world's largest Hadoop production application. The Yahoo! Search Webmap is a Hadoop application that runs on a Linux cluster with more than 10,000 cores and produced data that was used in every Yahoo! web search query.[84] There are multiple Hadoop clusters at Yahoo! and no HDFS file systems or MapReduce jobs are split across multiple datacenters. Every Hadoop cluster node bootstraps the

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Linux image, including the Hadoop distribution. Work that the clusters perform is known to include the index calculations for the Yahoo! search engine. In June 2009, Yahoo! made the source code of the Hadoop version it runs available to the public via the open-source community.[85] In 2010, Facebook claimed that they had the largest Hadoop cluster in the world with 21 PB of storage.[86] In June 2012, they announced the data had grown to 100 PB[87] and later that year they announced that the data was growing by roughly half a PB per day.[88] As of 2013, Hadoop adoption had become widespread: more than half of the Fortune 50 used Hadoop.[89]

Hadoop hosting in the cloud:

Hadoop can be deployed in a traditional onsite datacenter as well as in the cloud.[90] The cloud allows organizations to deploy Hadoop without hardware to acquire or specific setup expertise.[91] Vendors who currently have an offer for the cloud include Microsoft, Amazon, IBM,[92] Google and Oracle.[93]

On Microsoft Azure:

Azure HDInsight[94] is a service that deploys Hadoop on Microsoft Azure. HDInsight uses Hortonworks HDP and was jointly developed for HDI with Hortonworks. HDI allows programming extensions with .NET (in addition to Java). HDInsight also supports creation of Hadoop clusters using Linux with Ubuntu.[94] By deploying HDInsight in the cloud, organizations can spin up the number of nodes they want and only get charged for the compute and storage that is used.[94] Hortonworks implementations can also move data from the on-premises datacenter to the cloud for backup, development/test, and bursting scenarios.[94] It is also possible to run Cloudera or Hortonworks Hadoop clusters on Azure Virtual Machines.

On Amazon EC2/S3 services:

It is possible to run Hadoop on Amazon Elastic Compute Cloud (EC2) and Amazon Simple Storage Service (S3).[95] As an example, The New York Times used 100 Amazon EC2 instances and a Hadoop application to process 4 TB of raw image TIFF data (stored in S3) into 11 million finished PDFs in the space of 24 hours at a computation cost of about $240 (not including bandwidth).[96] There is support for the S3 object store in the Apache Hadoop releases, though this is below what one expects from a traditional POSIX filesystem. Specifically, operations such as rename() and delete() on directories are not atomic, and can take time proportional to the number of entries and the amount of data in them.

Amazon Elastic MapReduce:

Elastic MapReduce (EMR)[97] was introduced by Amazon.com in April 2009. Provisioning of the Hadoop cluster, running and terminating jobs, and handling data transfer between EC2(VM) and S3(Object Storage) are automated by Elastic MapReduce. Apache Hive, which is built on top of Hadoop for providing data warehouse services, is also offered in Elastic MapReduce.[98] Support for using Spot Instances[99] was later added in August 2011.[100] Elastic MapReduce is fault-tolerant for slave failures,[101] and it is recommended to only run the Task Instance Group on spot instances to take advantage of the lower cost while maintaining availability.[102]

Google Cloud Platform

There are multiple ways to run the Hadoop ecosystem on Google Cloud Platform ranging from self-managed to Google-managed.[106]

Google Cloud Dataproc – A managed Spark and Hadoop service[107]

Command line tools (bdutil) – A collection of shell scripts to manually create and manage Spark and Hadoop clusters[108]

Third Party Hadoop Distributions:

Cloudera – Using the Cloudera Director Plugin for

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Google Cloud Platform[109]

Hortonworks – Using bdutil support for Hortonworks HDP[110]

MapR – Using bdutil support for MapR[111]
Google also offers connectors for using other Google Cloud Platform products with Hadoop, such as a Google Cloud Storage connector for using Google Cloud Storage and a Google BigQuery connector for using Google BigQuery.

Author's Details:

K.Nagaraju
B.Tech Student,
Department of CSE,
Sphoorthy Engineering College,
Nadergul (Vill.), Sagar Road,
Saroonagar (Mdl), R.R Dist..T.S.

J.Deepthi
Associate Professor & HOD,
Department of CSE,
Sphoorthy Engineering College,
Nadergul (Vill.), Sagar Road,
Saroonagar (Mdl), R.R Dist.T.S.

Mr.T.Pavan Kumar
Assistant Professor,
Department of CSE,
Sphoorthy Engineering College,
Nadergul (Vill.), Sagar Road,
Saroonagar (Mdl), R.R Dist.T.S.