Semiconductor Science and Information Devices
https://ojs.bilpublishing.com/index.php/ssid

REVIEW

Apache Hadoop Architecture, Applications, and Hadoop Distributed File System

Pratit Raj Giri Gajendra Sharma*

Department of Computer Science and Engineering, Kathmandu University, Dhulikhel, Kavre, Nepal

ARTICLE INFO

Article history
Received: 29 March 2022
Revised: 08 May 2022
Accepted: 13 May 2022
Published Online: 16 May 2022

Keywords:
Hadoop
FsImage
HDFS
Apache Hadoop
MapReduce

ABSTRACT

The data and internet are highly growing which causes problems in management of the big-data. For these kinds of problems, there are many software frameworks used to increase the performance of the distributed system. This software is used for the availability of large data storage. One of the most beneficial software frameworks used to utilize data in distributed systems is Hadoop. This paper introduces Apache Hadoop architecture, components of Hadoop, their significance in managing vast volumes of data in a distributed system. Hadoop Distributed File System enables the storage of enormous chunks of data over a distributed network. Hadoop Framework maintains fsImage and edits files, which supports the availability and integrity of data. This paper includes cases of Hadoop implementation, such as monitoring weather, processing bioinformatics.

1. Introduction

Keeping large amounts of data in a single machine is tough. As a result, the data must be broken down into smaller parts and stored on different devices. Distributed file systems are filesystems that manage storage over a network of computers. Personal data are the new oil of the internet and the digital world's new currency [1]. Terabytes of data are being generated every minute via different web-based applications. The use of the internet has become wide and its trend has exponentially changed this decade. Therefore the problem in today's internet-based applications is storing data, managing them, and retrieving such significant streams of data. Such data volume is also known as Big Data, which consists of data characteristics such as volume, velocity, and variety. As the volume of data is increasing exponentially, storage space is not a problem for data storage. The cost of storage per gigabyte has been reduced to almost ten-cent. But processing and managing those data in terms of memory and performance costs more. To solve such issues, the Apache foundation has come up with Apache Hadoop. The two fundamental

*Corresponding Author:
Gajendra Sharma,
Department of Computer Science and Engineering, Kathmandu University, Dhulikhel, Kavre, Nepal;
Email: gajendra.sharma@ku.edu.np

DOI: https://doi.org/10.30564/ssid.v4i1.4619

issues of Big Data, storage and processing, are addressed by Apache Hadoop. HDFS is used to store data in Hadoop, while Map Reduce programming is used to process it [8]. Hadoop stores all data in a distributed fashion across a cluster of computers. Apache Hadoop is a collection of computers that work together to solve problems. The structure of a Hadoop cluster may be simply understood by looking at the Figure 1.

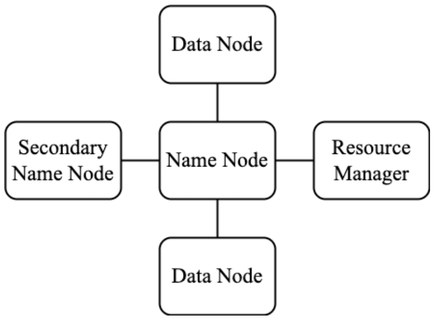

Figure 1. Typical Hadoop Cluster

2. Apache Hadoop

Apache Hadoop is a collection of open-source software utilities that facilitates using a group of computers interconnected in a network to solve problems regarding the massive amount of data and computation [2]. Mike Cafarella and Doug Cutting founded Hadoop. Hadoop is a software system that uses the MapReduce programming style to spread data storage and processing. Apache Hadoop was created with the intention of being used in computer clusters made up of commodity hardware. With a collection of commodity clusters, Apache Hadoop enables distributed computing. Apache Hadoop comprises MapReduce and Hadoop Distributed FileSystem.A file in the input directory is considered by MapReduce, and the desired output can be made to write to an output directory. Perhaps the architecture of HDFS is what makes it so popular now [10]. Google's GoogleFileSystem inspires Hadoop Distributed FileSystem. Apache Hadoop framework is developed by writing Java programming language. Native scripts were developed using the C programming language, and command-line scripts/utilities are written in a shell script. Apache Hadoop being a framework it supports various programming languages and applications such as Python, Perl, Ruby, SQL, PostgreSQL, HBase, etc. Hadoop stands as mostly used and easy because of the properties like the handling of Huge Volume of data, as a distributed file system, it is capable of holding petabytes of data without experiencing any problems. Data access is founded on the idea that "the most effective data processing pattern is write-once, read-many-times". HDFS is operated on commodity hardware in a cluster. These are low-cost machines that may be purchased from any retailer.

3. Apache Hadoop Architecture

HDFS is Hadoop's dependable storage component. This is because every block in the filesystem is duplicated throughout the cluster's Data Nodes. As a result, HDFS is fault-tolerant. The replication factor in HDFS is set to 3 by default. This means that every block will be duplicated twice, each on its own DataNode in the cluster. This creates very less chances of data loss and failure in the system,but what could be the cost? Obviously the cost of machines won't be higher than data [11]. The earlier section covered the basic definition of what Hadoop is. This section briefly covers the architecture and components of Hadoop Architecture. Apache Hadoop Core comprises the following elements:

3.1 Storage Part

The storage Part of Hadoop Architecture consists of the Hadoop Distributed File System (HDFS). HDFS is the official file type that is defined by Apache Hadoop. The storage part is concerned with splitting large files into small chunks and indexing them. The indexed files are then stored in different nodes present in a cluster.

3.2 Processing Part

The processing Part of Hadoop Architecture plays an even more critical role. As we discussed in the earlier section, data storage is not a significant issue in big data. However, processing, managing, updating, and playing with data is an important aspect. Therefore, the processing part of Hadoop architecture is responsible for Hadoop's programming model, i.e., MapReduce [4].

4. Master-Slave Architecture Design

Apache Hadoop's architecture is based upon the Master-Slave architecture design. Apache follows the Master-Slave architecture design for data storage and distributed data processing using HDFS and MapReduce. The NameNode is the Hadoop HDFS MasterNode for data storage. The Job Tracker is the master node for Hadoop MapReduce data parallel processing. The Hadoop architecture's SlaveNodes are the Hadoop cluster's additional computers that store data and do sophisticated calculations. A Task Tracker daemon and a Data Node are present on every Slave node, which synchronize the process with the Job Tracker and Name Node, respectively. A master or slave system can be created on a cloud platform or on-premises. The figure below defines the master-slave architecture. It illustrates the distinction between HDFS and MapReduce layer, master node and slave node, Job Tracker, and Task Tracker.

Storage Part and Processing Part are the core concepts of Apache Hadoop. Figure 2 is the architecture diagram of a Hadoop, the architecture of Hadoop is distributed into four segments, which are listed below:

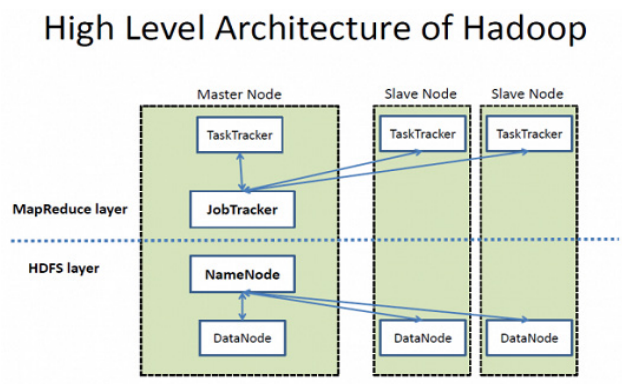

Figure 2. Apache Hadoop Master-Slave Architecture

4.1 Hadoop Common Package

Hadoop's common package is simply a bundle of software that provides different filesystems. Hadoop is a framework that runs on other operating systems. Therefore it requires the compatibility of running multiple filesystems. So the common package in Hadoop allows different filesystems to run. Besides that, it will enable operating-system-level abstractions. Herein this component of Hadoop framework software packages such as Apache Spark, Apache ZooKeeper, Cloudera, Cloudera Impala, Apache Flume, Pig, Apache Hive, etc. can be integrated.

4.2 Hadoop Distributed File System

Hadoop Distributed File System (HDFS) is the file system used in the Hadoop framework. This is the storage unit in Hadoop. A file on HDFS is split into multiple blocks. Each is replicated within the Hadoop Cluster A block of the Hadoop Distributed File System is a blob of data within the underlying file system with a default size of 64 MB. A single blob size, i.e., the block, can be extended up to 256 MB based on the requirements. The application data and the file system metadata are stored separately on different dedicated servers. The Figure 3 below shows the architecture of the components of the Hadoop distributed file system.

4.2.1 NameNode and DataNode

The Hadoop HDFS architecture includes two key components: NameNode and DataNode. DataNodes are where application data is kept on the server. On the server known as NameNode, file system metadata is saved. To maintain data dependability, HDFS duplicates file information across several DataNodes based on the replication factor. The NameNode and DataNode communicate with each other using Transmission Control Protocol-based network protocols. For the architecture to be performance efficient, HDFS must satisfy specific properties like the throughput of hard disk drives set up on each node. Also, a good network speed should be managed for intermediate data transfer and block replications.

On the Namenode, Inodes represent all of the files and directories in the HDFS namespace, carrying different features such as permissions, modification timestamps, disk space quotas, and access timings. During restarts, two files, fsImage and edits, are utilized for persistence.

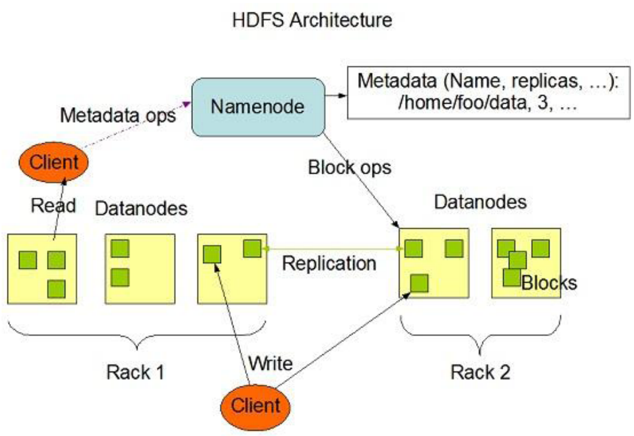

Figure 3. Hadoop Architecture

4.2.2 FsImage and Edits

The Inodes and list of blocks that define the Metadata are stored in the fsImage file. At any moment in time, fsImage has a full snapshot of the file system's metadata. Any changes made to the content of the fsImage file are saved in the modifications file. Instead of producing a new fsImage snapshot every time the namespace is changed, an incremental modification like leaving or adding data to the file is saved in the edit log to maintain persistence. When the NameNode starts, the fsImage file is loaded, and then the contents of the edits file are applied to recover the latest state of the file system. Over time the edits file grows and consumes all the disk space resulting in slowing down the restart process. This is an issue in the Hadoop Distributed File System created by an increase in the number and the size of edit files. If the Hadoop cluster has not been restarted for months together, there will be an extended downtime as the edits file size will increase. Secondary NameNode gets the fsImage edits to log from the primary NameNome at regular intervals and loads both the fsImage and edit logs file to the main memory.

Datanode manages the state of an HDFS node and interacts with the blocks. Datanode handles CPU-intensive operations like semantic and linguistic analysis statistics, as well as I/O-intensive processes like clustering, data input, data export, decompression, and indexing. Every DataNode connects to the NameNode during startup and performs a handshake to validate the DataNode's namespace ID and software version.

If one of these does not match, the DataNode will instantly shut down. A data node independently validates block replicas and sends a block report to the NameNode. The initial block report is delivered as soon as the data node registers. Every three seconds, the DataNode transmits a heartbeat to the NameNode to ensure that the data node is up and running, as well as the block replicas if hosts are available.

4.2.3 Job Tracker

Job Tracker is useful in processing the data. Job Tracker receives the requests for MapReduce execution from the client, talks to Namenode to know about the location of the data, and gives Metadata to the JobTracker. The JobTracker process runs on a separate node and not usually on a DataNode [9]. JobTracker is a master which creates and runs the job.

4.2.4 Task Tracker

The task tracker is the Job tracker's Slave Node, and it will take the job from the Job tracker. The code from the Job Tracker is received and applied to the file. Mapper is the term for the process of using code on a file. Failure of TaskTracker is not deemed fatal. When a TaskTracker stops responding, JobTracker will reassign the TaskTracker's task to another node. It has the function of following the orders of the job tracker and updating the job tracker with its progress status periodically.

4.2.5 Rack

A Rack is a group of computers (40-50 in Hadoop) that are physically stored together. A Hadoop cluster is made up of numerous Racks that are all connected via switches. Hadoop implements the Rack Awareness algorithm [11].

5. Hadoop MapReduce

MapReduce is the processing unit in Hadoop. MapReduce Paradigm is the most efficient programming model that handles the management, processing, and execution of big data.MapReduce is a computer technique for processing data that is distributed over a Hadoop cluster. Once a MapReduce task starts, the ResourceManager requisitions an Application Master to manage and monitor the MapReduce job lifecycle, just like any other Hadoop activity. Hadoop MapReduce is a java-based programming paradigm that is the core component of the distributed computation platform. Map or Reduce is a particular type of directed acyclic graph applied to a wide range of business use cases [3]. A directed graph is acyclic if and only if it has a topological ordering. The map function converts a piece of data into a set of key-value pairs. A reduction function is used to the values depending on the key to blend them into a single output. The Figures 4 and 5 below show a Map and smaller code snippets used in Big Data.

```
In [33]: # Crimes mapped acc to days
         crimes.map(lambda x:x.Day_of_Week).countByValue()

Out[33]: defaultdict(int,
                     {'Friday': 178230,
                      'Saturday': 160324,
                      'Tuesday': 158015,
                      'Thursday': 159360,
                      'Sunday': 145995,
                      'Monday': 161229,
                      'Wednesday': 160138,
                      '': 173,
                      'NA': 1})
```

Figure 4. Mapping a data processing query

```
In [44]: #  Checking anamolies in location
         def extractCoords(location):
             location_lat = float(location[1:location.index(",")])
             location_lon = float(location[location.index(",")+1:-1])
             return (location_lat, location_lon)

In [45]: crimesFiltered.map(lambda x:extractCoords(x.Location_1))\
                      .reduce(lambda x,y:(min(x[0],y[0]),min(x[1],y[1])))

Out[45]: (40.494426993, -77.0567227029999)
```

Figure 5. Reducing results

6. Yet Another Resource Negotiator

YARN is the resource management unit. For a Hadoop Application to execute big data processing, it should go through the following steps:

- Application submission
- Bootstrapping the ApplicationMaster instance for the application
- Application execution managed by the Application Master instance

In YARN, the ApplicationMaster is an instance of a framework-specific library that is responsible for negotiating resources from ResourceManger and collaborating with the NodeManager to execute and monitor containers and their resource usage. Figure 6 illustrates the YARN architecture.

ResourceManager is a pure scheduler which schedules competing applications arbitrating available resources.

YARN is a system for managing distributed applications, which consists of a central ResourceManager and a NodeManager. The fundamental idea is to split up resource management and job scheduling and monitoring on separate daemons.The storage layer, represented by HDFS, and the MapReduce processing engine are separated by YARN's resource allocation function. YARN also has a general interface for creating new processing engines for a variety of data kinds.

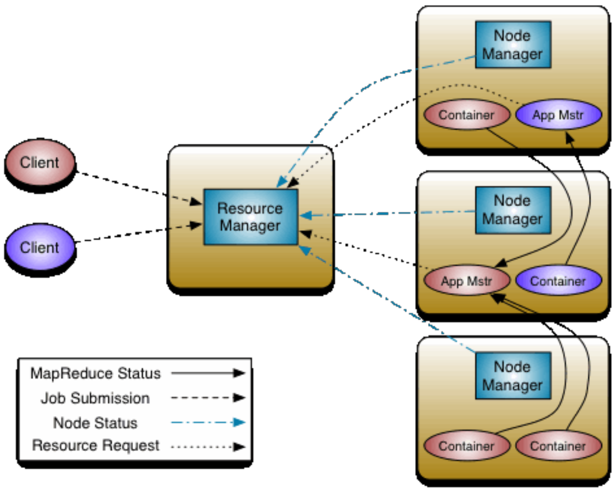

Figure 6. YARN Architecture

7. Apache Hadoop Implementation

Distributed computation with the help of the Hadoop Framework can be easily implemented in commodity hardware. Hadoop Framework is used in weather data analysis in research conducted by authors M. A. Fakhreldin, K. Adam, N. A. Bakar, and M. A. Majid [7]. The weather data processed by the Hadoop framework ranged 20 GB, and 1 TB disk space was occupied in weather data execution.

A research work organized by D. Papakyriakou in August 2019 used the Raspberry PI 2 to prepare a Hadoop Cluster. Fifteen Raspberry PI 2 was used to prepare the cluster with either 16 MB or 32 MB microSD for the boot process, and an external 320 GB of Hard disk was connected to the Namenode and 32 GB to boot it [5]. Ethernet switches of 10/100 Mbps were used to provide maximum throughput. On a test load for speed during the data transmission that ranges from 1000 MBytes-12000 MBytes of data, the Write Speed and Read Speed average was 2.152 MB/sec and 4.749 MB/sec, respectively. This promises the data execution can be performed over distributed clusters of commodity hardware in Apache Hadoop.

Research work conducted by J. Alnasir, H. P. Shanahan, has implemented a remote 68 node Hadoop cluster for docking 2367 compounds against Human estrogen receptor alpha agonist protein [6]. Ten mappers per node on the 57 nodes were dedicated to running 570 mappers MapReduce tasks in parallel, which resulted in 450× speed-up, and 95.59% of CPU time was utilized. The same research work was extended to a parallel multithreading scheme at a large scale, i.e., 15,408 CPUs, which reflected only a 3.94% scale, which was 5% in an earlier experiment. For running 17 million flexible compound docking compounds, 70% of CPU time was utilized.

8. Advantage and Challenges of Hadoop

Hadoop offers surprising benefits for big data. Some of the benefits are:

- We can perform sophisticated queries in just a few seconds because of Hadoop's concurrent processing, MapReduce paradigm, and HDFS.
- HDFS can hold a variety of data forms, including structured, semi-structured, and unstructured data.
- Hadoop is an open-source data architecture that is cost-effective.
- Data saved in one node are duplicated on other nodes in the cluster, ensuring fault tolerance.
- Hadoop is scalable because it works in a distributed setting, so you can quickly add more servers.

Besides the advantages there are few challenges in implementing the Hadoop, there is a significant learning curve. Not all datasets can be treated in the same way and MapReduce has its limitations. Apart from the challenges, Hadoop has more benefits of use [12].

9. Comparison with Other Frameworks

Hadoop's constraints necessitated the use of Spark, while Spark's shortcomings necessitated the use of Flink. Apache Hadoop, Spark and Flink are the major big data frameworks. The only processing framework that integrates data with artificial intelligence is Apache Spark, the greatest open-source project in data processing (AI) [13]. Spark is a MapReduce improvement in Hadoop. Spark and MapReduce vary in that Spark processes and keeps data in memory for following stages, whereas MapReduce processes data on disk. Hadoop's scalability ranges from a single server to thousands of computers, and it may be utilized in real-time analytics for historical studies and decision-making [13]. Here is a short comparison between Apache Hadoop and Spark in terms of the following indicators.

Performance and Processing: Spark is quicker because it stores intermediate data in Random Access Memory (RAM) rather than reading and storing it to drives.

Hadoop collects data from a variety of sources and uses MapReduce to process it in batches. Despite the fact that both technologies process data in a distributed setting, Hadoop is better for batch and linear data processing. Spark is suited for real-time processing of unstructured data streams in real time [14].

Cost and Scalability: Hadoop is less expensive to run since it processes data on any form of disk storage. Spark is more expensive to run as it relies on in-memory calculations for real-time data processing, necessitating the utilization of large amounts of RAM to spin up nodes. Hadoop's Hadoop Distributed File System efficiently expands to meet demand when data volume rises fast (HDFS). For big amounts of data, Spark relies on the fault-tolerant HDFS [13,14].

Security: Spark improves security by using a shared secret or event logging for authentication, whereas Hadoop combines several authentication and access control techniques. Though Hadoop is safer in general, Spark may be integrated with Hadoop to increase security.

10. Conclusions

The volume of data on the internet increases every minute in the number of terabytes, and the computation, management, and storage of big data is a challenging problem. Hadoop by Apache foundation is a prominent solution for executing, storing and managing vast amounts of data in a distributed system. The Storage and Processing part are significant components of Hadoop Architecture. Data is partitioned and indexed over a distributed cluster in the storage part. The calculation is done by mapping the Metadata and reducing the data as per the expression or query provided in the Processing part. MapReduce paradigm enables computation of a big chunk of data over distributed nodes, retrieving results quickly. Hadoop architecture was designed considering that the distributed nodes could be made with commodity hardware. Hadoop architecture supports thousands of nodes for complex computation. Apache Hadoop architecture resembles Master-Slave Architecture. The Inodes and list of blocks that constitute the Information are stored in the fsImage file, which includes a complete snapshot of the file system's metadata at any one moment. Similarly, the edits file contains any changes made to the content of the fsImage file. Job Tracker keeps a record of jobs that the task tracker should be executing. Likewise, NameNode in HDFS contains information about data nodes. HDFS is one of the best techniques that maintain, analyze, process and manage large data. BlockHDFS, a new way for distributed file systems, uses the enterprise-level Hyperledger Fabric platform to capitalize on files' metadata for building trusted data security and traceability in HDFS [10]. There are many frameworks similar to Hadoop but Hadoop is the most widely used. In large data research frameworks like Hadoop, HDFS is one of the most extensively used distributed file systems.

Conflict of Interest

There is no conflict of interest.

References

[1] Personal Data: The Emergence of a New Asset Class. Meglena Kuneva, European Consumer Commissioner, March 2009 (Online). Available: http://www3.weforum.org/PersonalDataNewAsset_Report_2011.pdf (Accessed Dec.10, 2021).

[2] Apache Hadoop Documentation (Online). Available: https://hadoop.apache.org/old/ (Accessed Dec.10, 2021).

[3] Directed Acyclic Graph (Online). Available: https://en.wikipedia.org/wiki/Directed_acyclic_graph (Accessed Dec.10, 2021).

[4] Vasuja, R., Bhandaralia, A., Chuchra, K., 2018. Daemons of Hadoop: An Overview. International Journal of Engineering Research & Technology(IJERT). 07.

[5] Papakyriakou, D., 2018. Benchmarking Raspberry Pi 2 Hadoop Cluster. International Journal of Computer Applications (0975-8887). 178(42), 37-47.

[6] Alnasir, J., Shanahan, H.P., 2018. The application of Hadoop in Structural Bioinformatics. Oxford University Press. pp. 16-35.
DOI: https://doi.org/10.1093/bioinformatics

[7] Fakhreldin, M.A., 2019. Weather Data Analysis Using Hadoop: Applications and Challenges. Joint Conference on Green Engineering Technology & Applied Computing.
DOI: https://doi.org/10.1088/1757-899X/551/1/012044

[8] Begum, G., Huq, S.Z.U., Kumar, A.P.S., 2020. Sandbox security model for Hadoop file system. Journal of Big Data. 7, 82.
DOI: https://doi.org/10.1186/s40537-020-00356-z

[9] World, H., 2022. JobTracker and TaskTracker. Hadoop In Real World (Online). Available: https://www.hadoopinrealworld.com/jobtracker-and-tasktracker/. (Accessed 18 Mar 2022).

[10] Mothukuri, V., Cheerla, S., Parizi, R., et al., 2021. BlockHDFS: Blockchain-integrated Hadoop distributed file system for secure provenance traceability. Blockchain: Research and Applications. 2(4), 100032. (Accessed 15 March 2022)
DOI: https://doi.org/10.1016/j.bcra.2021.100032

[11] HDFS | HDFS Architecture | Components Of HDFS.

Analytics Vidhya, 2022. (Online). Available: https://www.analyticsvidhya.com/blog/2020/10/hadoop-distributed-file-system-hdfs-architecture-a-guide-to-hdfs-for-every-data-engineer/. (Accessed 08 Mar 2022).

[12] Hadoop and How Does It Work, simplilearn, 2022. (Online). Available: https://www.simplilearn.com/tutorials/hadoop-tutorial/what-is-hadoop. (Accessed 10 Mar 2022).

[13] Hadoop vs Spark vs Flink – Big Data Frameworks Comparison, Data Flair, 2022. (Online). Available: https://data-flair.training/blogs/hadoop-vs-spark-vs-flink/. (Accessed 01 Mar 2022).

[14] Hadoop vs. Spark: What's the Difference? 2022. (Online). Available: https://www.ibm.com/cloud/blog/hadoop-vs-spark. (Accessed 08 Mar. 2022).