Hadoop and its Ecosystem

IDS 521 Term Paper

Claudia Espinosa, Prajakta Iyer, Riddhi Jain & Snehal Thakur

Table of Content

Background

History

Hadoop was created by Doug Cutting who was also the creator of Apache Lucene. Cutting, started a project with Apache Nutch in 2002 as an open source web search engine which was still part of the Lucene Project. The purpose of creating Nutch was to sort and merge web search pages and in 2003 the project was able to successfully crawl 100 million pages on 4 nodes. Cutting later published a paper that explained how Google's Distributed Filesystem (GFS) would not allow for billions of pages to be scaled. This publication prompted Google to implement an open source system called Nutch Distributed Filesystem (NDFS) in 2004 and they also introduced MapReduce. With these implementations by the year 2005, Cutting and his partner Mike Cafarella had been working with MapReduce and the NDFS system but in early 2006 they branched off and went made its own project called, Hadoop. Cutting was inspired to call the project Hadoop after his son's toy elephant named, Hadoop. By the time Hadoop was independent from the earlier projects it was able to run 20 nodes at a time and crawl over 100 million pages.

In the same year, Cutting joined Yahoo! Which helped him make Hadoop the top project of the year and provided resources that would give Hadoop the ability to run at a web scale. After joining Yahoo! Hadoop evolved very quickly, by May 2006 it was able to run on 500 nodes in 42 hours and by the end of the year it was able to run on 20 nodes in less than 2 hours. Two years after joining Yahoo! Hadoop was able to sort 1 Terabyte in 3 minutes and 28 seconds using only

990 nodes. Since 2008 there have been many releases of Hadoop, but by July 2013 Hadoop had achieve a sorting rate of 1.42 Terabytes per minute.

The evolution of Hadoop has been very attractive to companies because at the sorting rate that Hadoop has achieved it helps process all the data that companies are generating on a daily basis.

Big Data

Big Data is large amount of data that consist of structured and unstructured data that cannot be stored or processed by traditional data storage techniques. The data can be categorized as operational or analytical Big Data. Large amounts of data are being generated daily which can help address business problems and needs that they would have not been able to address before. Big data prompted the 3 V's of data which are Volume, Velocity and Variety.

The Volume is the amount of data generated and stored from a company that adds the "big" in Big Data. The Volume of data has prompted a list of standard measurements used for data storage. Many people talk about gigabytes and terabytes as the common type of data storage measurement but with big data there are still 4 more types of measures greater than terabyte. Petabyte (PB) can consist of over millions of data records and can vary from different sources. Social media platforms like Facebook, Instagram and Twitter generate large amounts of data daily, from the number of messages sent, number of likes, number of posts this amount of data is stored in hundreds of petabytes and terabytes.

Velocity is the rate at which the data is generated. As explained earlier, Social Media platform generate huge amounts of data daily. Velocity measures data in 3 stages, Periodically, near Real Time and in Real Time. The amount of data that is generated is massive and can help companies make decisions if it is being valued in real-time. Analyzing data in real-time allows the companies to make strategic and competitive decisions when needed instead of waiting for the data to be gathered and then making decisions.

Variety consist of the type of data that is produced by Big Data. This can range from excel datasets, unstructured text, audio recordings, photos and videos. The different types of unstructured data can cause storing problems for the company and can delay the analysis of the data because an analyst would have to clean the data to a format that can be analyzed to make decisions.

Data comes in all shapes and sizes, but every company now has tons and tons of data. This data needs to be stored and cleaned and that is where Hadoop helps all the enterprises store and process large amounts of data.

NoSQL databases

NoSQL database refers to a non relational database which is used to store and access data. These databases are used in real time web applications and Big Data. NoSQL databases offer the concept of eventual consistency because of which database changes are propagated to all nodes so queries for data might not return updated data immediately or might result in reading data that is not accurate which is a problem known as stale reads.

Types of NoSQL databases and the name of the database system that falls in that category are:

1) MongoDB falls in the category of NoSQL document based database
2) Key Value pair: Redis, Coherence
3) Tabular: HBase, Big Table

4) Document based: MongoDB, CouchDB, Cloudant

For Hadoop, the type of NoSQL database is called as Hbase which is of tabular type. It is called as Hbase because it is a non relational database and runs on top of Hadoop as a distributed and scalable big data store. This means that HBase can leverage the distributed processing paradigm of the Hadoop Distributed File System (HDFS) and benefit from Hadoop's MapReduce programming model. It is meant to host large tables with billions of rows with potentially millions of columns and run across a cluster of commodity hardware. But beyond its Hadoop roots, HBase is a powerful database in its own right that blends real-time query capabilities with the speed of a key/value store and offline or batch processing via MapReduce. In short, HBase allows you to query for individual records as well as derive aggregate analytic reports across a massive amount of data.

What is Hadoop?

Hadoop is an open-source software framework under the Apache Software Foundation used for storing data and running applications on a cluster of commodity hardware. Hadoop provides the ability to store and process large amounts of any kind of data which makes it very attractive to all companies generating large amounts of data daily. One of the goals of Hadoop is to facilitate the processing of large amounts of data sets, structured and unstructured data as well as facilitate the storage process. Storing large amounts is expensive and Hadoop is a great option for companies who generate massive amounts of data daily. Hadoop is extremely accessible and allows for flexible use of hardware for the data, it is also low maintenance and very inexpensive to operate.

Real-life example

Companies that benefit from using ‘Big Data’ along with Hadoop are Financial Services companies as well as Retail Companies.

Financial Services Companies can use Hadoop and Big Data build investment models. The investment models can be trading algorithms that Hadoop can help build and run. Hadoop is extremely useful because it can work without any human interaction and can help with Financial Trading and Forecasting of any company. The high-frequency rating of the market can help financial services companies make decisions using any algorithm that was created.

Retail companies also benefit from using Hadoop in their everyday decision making. Retail companies can analyze prices set to any of the products in real time and giving them the opportunity to not lose any customers. To set the correct price retail companies would have to analyze loads of data to ensure that they know who their customers are, and which price is more likely to sell. Having this advantage can help retail companies know which items are the best sellers and what price, they will also have the ability to launch promotions to increase sales.

The use of Hadoop, Big Data and predictive analysis can help companies understand their customers needs and own need better to improve profits within the company.

About

Architecture

Apache Hadoop provides a scalable, flexible and reliable big data framework for a cluster of systems with storage capacity and local computing power by leveraging commodity hardware. The flexible nature of a Hadoop system enables organizations to add to or modify their data system as their needs change. Cost-effective and readily-available components from any IT vendor can be used. Another feature of Hadoop is that one can easily scale the cluster by adding more nodes. Lastly, Hadoop's architecture handles fault or system failure by the process of replica creation. These features are highlighted in the below figure number 1.

Fig no.1 Features of Hadoop Architecture

The architecture of hadoop follows a Master-Slave architecture for storage of data as well as distributed data processing using MapReduce and HDFS methods. The Master node takes care of the two functional parts of Hadoop - HDFS (for storage) and Mapreduce (parallel computation). 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. The connection between the Namenode and the Datanodes can be viewed in figure no. 2 below.

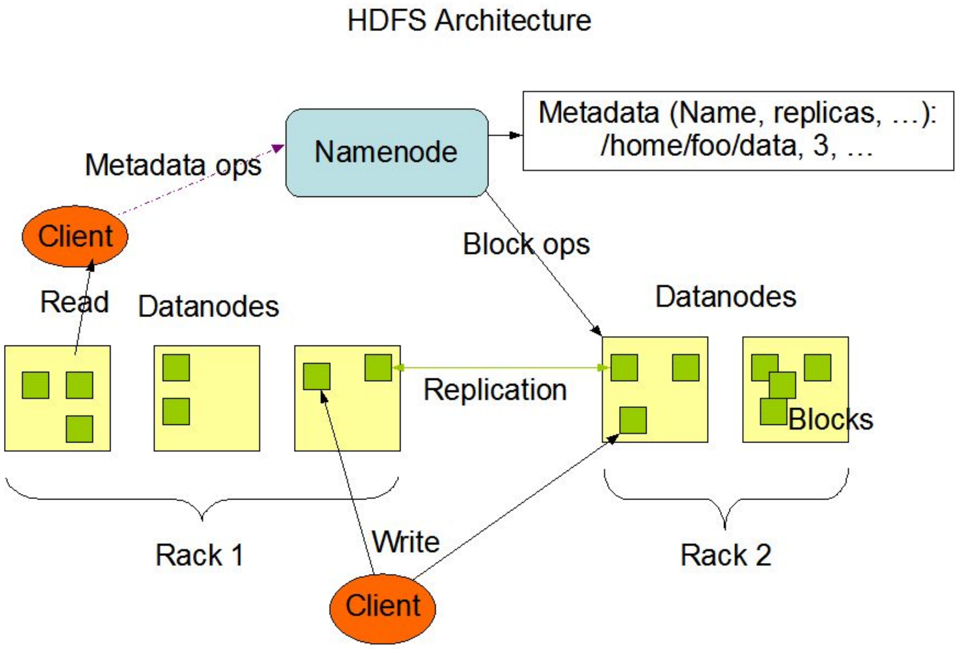


Fig no. 2 Master-Slave architecture

The three important Hadoop components in the Hadoop Architecture are -

i) HDFS (Hadoop Distributed File System)

ii) MapReduce

iii) YARN (Yet Another Resource Negotiator)

The master node for data storage in HDFS is called as the **Namenode**. The Master (NameNode) manages the file system namespace operations like opening, closing, and renaming files and directories and determines the mapping of blocks to DataNodes along with regulating access to files by client. On the other hand, Slaves (DataNodes) are responsible for serving read and write requests from the file system’s clients along with perform block creation, deletion, and replication upon instruction from the Master (NameNode).

As for MapReduce, the master node is called as the **JobTracker**. Master (Jobtracker) is the point of interaction between users and the map/reduce framework. When a map/reduce job is submitted, Jobtracker puts it in a queue of pending jobs and executes them on a first-come/first-served basis and then manages the assignment of map and reduce tasks to the tasktrackers. Slaves (tasktracker) execute tasks upon instruction from the Master {Jobtracker} and also handle data motion between the map and reduce phases.

Three Pillars

The flexibility, scalability and fault-tolerant nature of Hadoop is a result of the three pillars of Hadoop, namely MapReduce, HDFS and YARN. HDFS is responsible for storing the data while MapReduce is in charge of data processing. YARN is the most promising pillar of Hadoop. It has helped overcome Hadoop's many limitations and enabled it to be used by a pool of different applications while allowing real-time analysis. The pillars are explored in detail below-

MapReduce

MapReduce algorithm is a processing technique and a program model used for faster processing and processing large datasets. It is made up of two tasks- Map and Reduce. Map is the first task followed by Reduce. Map is responsible for taking the data and converting it to a dataset that comprises of tuples that can be processed. Reduce uses the output from Map and convert the tuples into even smaller sets of tuples. This process of breaking down the whole datasets into small tuples helps to decompress the dataset, thereby improves its scalability. Once an application is written in the MapReduce format, we can use it to run over thousands of machines in clusters.

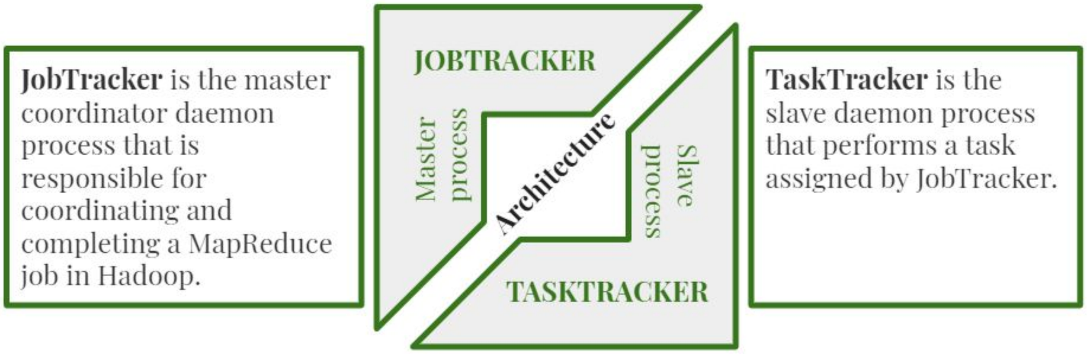
MapReduce Architecture

MapReduce sends a periodic notification to each worker, and if no response is received within the specified timeline, it is marked as a failure. MapReduce follows a master-slave process architecture. It has 2 daemon processes-

(1) **JobTracker -** This is the Master process. It is responsible for coordinating and completing the MapReduce job. It ensures resource management, tracking resource availability, and task process cycle. It is the single point of failure in this process. It assigns tasks to the TaskTracker.

(2) **TaskTracker -** This is the Slave Process. It follows the instructions received from the JobTracker. It sends a periodic status update about tasks and checks if there is any task in the queue for it to perform.

The Figure number 3 shows the MapReduce architecture divided into JobTracker and TaskTracker.



JobTracker is the master coordinator daemon process that is responsible for coordinating and completing a MapReduce job in Hadoop.

TaskTracker is the slave daemon process that performs a task assigned by JobTracker.

Fig no. 3 MapReduce architecture

MapReduce Process

MapReduce process involves multiple steps and is very complex. The following steps have to be performed in the order mentioned below-

(1) **Mapper-** This is the first step in the MapReduce process. It is a logical code which uses a Driver program to process or 'Map' the data. The output of this process is NOT saved in the HDFS and is in the form of <key, value> format.

(2) **Shuffler and sorter-** This is an intermediate step between the Mapper and Reducer and helps in processing the output achieved from the Mapper and append it in a list form.

(3) **Reducer-** This is the last step in the MapReduce process. It uses the key obtained by it to process the data from Mapper to consolidate the data and saves the final output in HDFS.

HDFS

HDFS, short for Hadoop Distributed File System, is a file system in Hadoop known for its understandable design and scalable, flexible and fault-tolerant capacity. It has a self- healing process which makes it more fault-tolerant. Like MapReduce, HDFS also has a master-slave architecture pattern which enables the filesystem to manage the slave nodes more efficiently. It is thereby moving computation to a cheaper era. HDFS is designed to work with MapReduce efficiently.

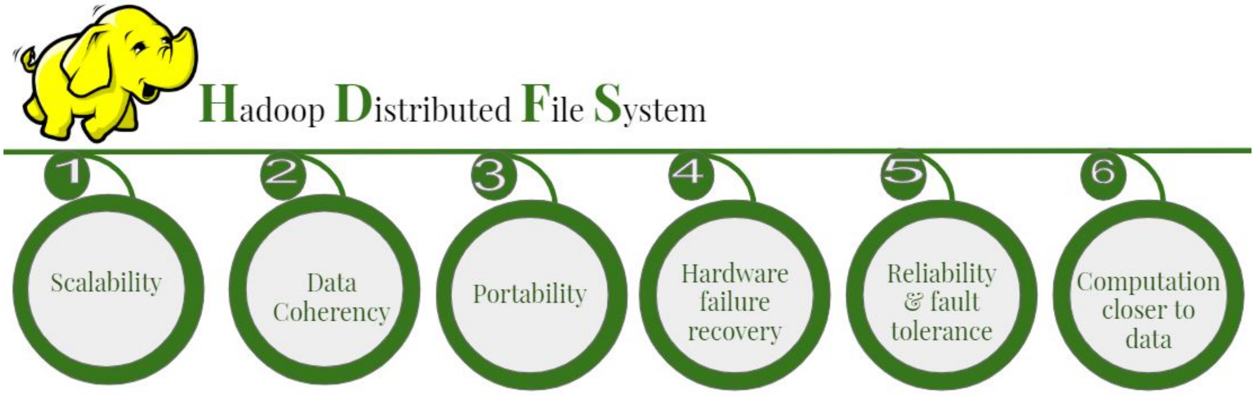

Hadoop Distributed File System
1
Scalability
2
Data Coherency
3
Portability
4
Hardware failure recovery
5
Reliability & fault tolerance
6
Computation closer to data

Fig no. 4 Features of HDFS

Features of HDFS

The Figure no. 4 shows the features of HDFS--

(1) **Scalability-** HDFS can compute data in petabytes or even more. You also have the flexibility to add or delete nodes which helps make the data more scalable.

(2) **Data Coherency**- HDFC has the functionality of allowing a program once written to be run again. This makes the data more Coherent and is known as its WORM (Write Once, Read Many) functionality.

(3) **Portability-** Works on different hardware and software across the board.

(4) **Hardware Failure Recovery-** HDFS allows the nodes to fail and has a good failure recovery process. It can also recover lost data.

(5) **Reliability & Fault Tolerance-** HDFS makes copies of the data to give the user flexibility of getting higher reliability and increase the fault tolerance of the system. As the data is broken down and stored in multiple nodes, the data can be accessed from more than one available node.

(6) **Computation closer to Data**- It moves the process closer to the data instead of moving the data closer to the computation process. This enables faster computation and works best with MapReduce process.

HDFS Architecture

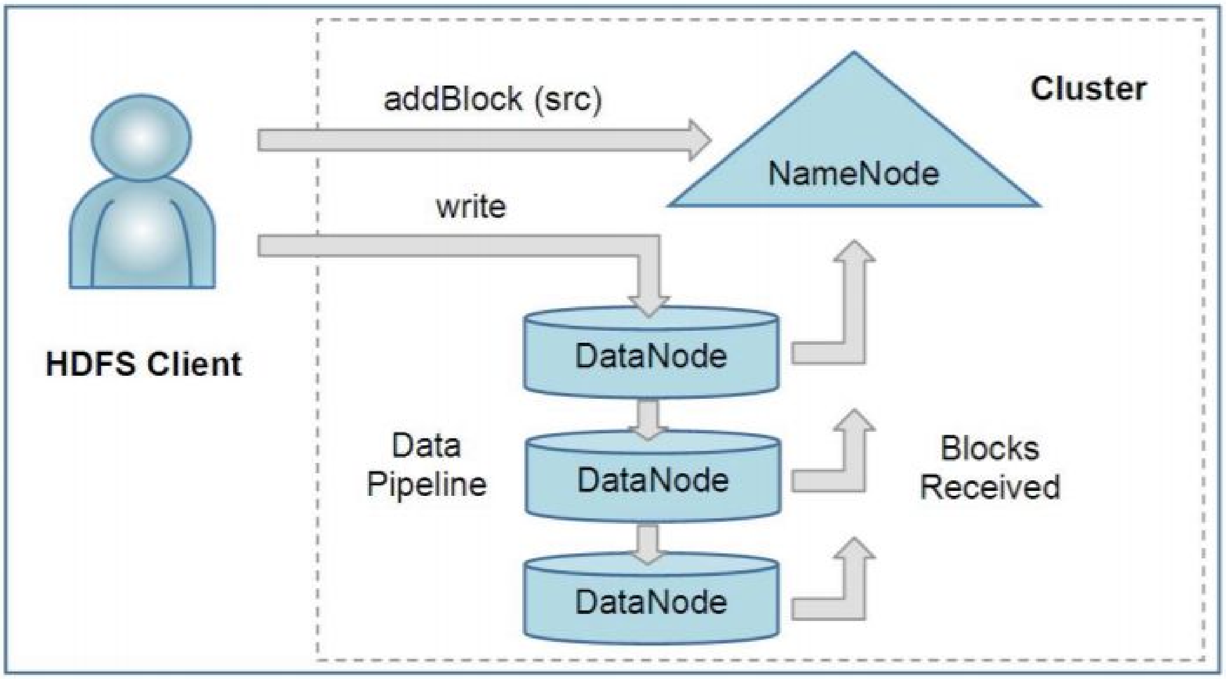


Fig no. 5 HDFS architecture

The figure number 5 shows that HDFS architecture. The four daemon process in HDFC architecture are as follows-

(1) **NameNode**- Master Process

(2) **DataNode**- Slave Process

(3) **CheckpointNode**- Checkpoint Process

(4) **BackupNode**- Backup NameNode

Once an HDFS client creates a new file by giving its path to the NameNode, each block of the file, the NameNode returns a list of DataNodes to host its copies. The client then sends the data to DataNodes, which confirms the creation of the block replicas to the NameNode.

YARN

YARN, short for Yet Another Resource Negotiator, was released in 2012 and took about 5 years to develop. YARN was adopted by Yahoo! in 2006 to replaces its old infrastructure called the WebMap application. It is a next-generation technology used to compute and cluster management technology. It divides and splits up the responsibility into separates daemons to run more efficiently. YARN helps Hadoop to be able to work on a real-time basis and be more interactive. This feature is very important as it allows Hadoop to process huge amounts of data on a continuous basis.

YARN Architecture

Just like MapReduce, YARN's architecture is known for being scalable and fault-tolerant but it is faster than the MapReduce process. Its architecture comprises of the following components-

(1) **ResourceManager-** This is a Master Process which uses a scheduler to allocate resources pending the resource availability and need. It has two interfaces-

  (a) Client Submitting applications

  (b) Application Masters

(2) **NodeManager-** It is responsible for authentication and monitoring the usage of the resources and sends a periodic update to the ResourceManager about its presence. It is present in every application that is run with YARN.

(3) **ApplicationMaster-** It is responsible for coordinating with the ResourceManager to create a plan and execute the plan from the resources it receives. It is present in each application that is run with YARN and sends a periodic update to ResourceManager about its projects and presence.

Applications Powered by YARN

Some of the applications powered by YARN are as follows-

(1) **Apache Giraph** - This tool helps to process graphs and big data.

(2) **Apache Spark** - This is used for big data processing and comes with SQL and machine learning capabilities, amongst other tools.

(3) **Apache Samza** -This is used to stream and process real-time data.

(4) **Apache Storm** - This is a realtime computational system.

(5) **Apache S4** - S4 stands for Simple Scalable Streaming System and is a general application which helps programmers stream continuous data as well as process it.

(6) **Apache Tez** - This helps in processing complex directed-acyclic-graph of tasks.

(7) **Apache Hama** - This is used for Big Data Analytics and uses a BSP computational model.

(8) **Apache Hadoop MapReduce -** This is a Big Data Analytics tool used to process large amounts of data.

Data Access Components

While MapReduce is a very powerful framework, mastering and optimizing a MapReduce job has a huge learning curve and involves a lot of analysis and coding. Since users in big data come from different backgrounds, abstractions are built on top of Hadoop for the ease of data

Analysis. Two of such abstractions are Pig and Hive where Hive provides SQL like interface on top of Hadoop and Pig uses Pig-Latin procedural language interface.

Hive

Hive was developed by Facebook for summarizing , querying and analyzing data and later on taken up by Apache Software Foundation and further made open source under the name Apache Hive. Hive is an abstraction over Map-Reduce. It is an SQL like interface to Hadoop and is called as Hive-QL(HQL). It processes structured data, stores schema in database and processed data into HDFS. It is designed for OLAP and used for analytical queries. It is familiar, fast, scalable and extensible. Hive converts SQL like queries into Map-reduce jobs for easy execution and processing of large amounts of data.

Pig

Developed at Yahoo, Pig was used for creating programs for Hadoop by using procedural language. Pig is an Open-source high-level data flow system. Pig is a scripting language and high level language is known as Pig Latin. Large data sets present in the Hadoop cluster are processed with the help of Pig. Using Pig as an alternative to Java for creating MapReduce programs, developers spend less time creating mapper and reducer programs. Analyzing data sets can thus be their primary focus. Pig is built on the philosophy that it can almost eat anything, can be controlled and can store anything like an actual pig. Pig can handle any kind of data. Pig processes structured and semi-structured data. 10 lines of Pig is equivalent to 200 lines of Java. Thus, the biggest advantage of Pig is reduction of overall development and testing time.

Data Ingestion in HADOOP

Volume of data in Hadoop is usually in terabytes or petabytes. With changing data formats , the key issue is to manage data consistency and how to leverage resources available. Since data in Hadoop is being processed in batch, stream and real time,data ingestion can become a bottleneck or can break a system if not designed according to the requirements. We have Data Ingestion tools like Flume and Sqoop to enable transferring and processing of streaming data .

Flume

Apache Flume is a framework for transferring huge amounts of data to and from HDFS. It captures a stream of moving data continuously flowing into the system. Flume is extremely efficient in handling scenarios with changing data formats and has greater control over each data source and processing layer. Flume deals with streaming data such as geological data, machine data, sensor, social media data and mobile data. Flume is used because it is extremely fault tolerant, reliable ,distributed , scalable and customizable to ingest data and process it from multiple sources to multiple destinations. It helps to overcome the challenges in data ingestion like parallel processing and scalability.

Sqoop

Sqoop is used for transfer of data between traditional databases, Hadoop and NoSQL databases. It helps to process bulk data transfer on HDFS, Hive or HBase. It provides utility to import and export data in Hadoop. Sqoop executes the process in parallel and hence is fast and efficient. Data in Sqoop is sliced up into different partitions and uses connectors and drivers to connect with the underlying database source. It executes import and export in multiple mapper processes executing the data in parallel and faster.

DATA MANAGEMENT IN HADOOP

To manage the huge volume of data flowing and being processed in Hadoop, data management tools like Zookeeper and Oozie are used. These tools manage the coordination, resource availability and scheduling of data efficiently.

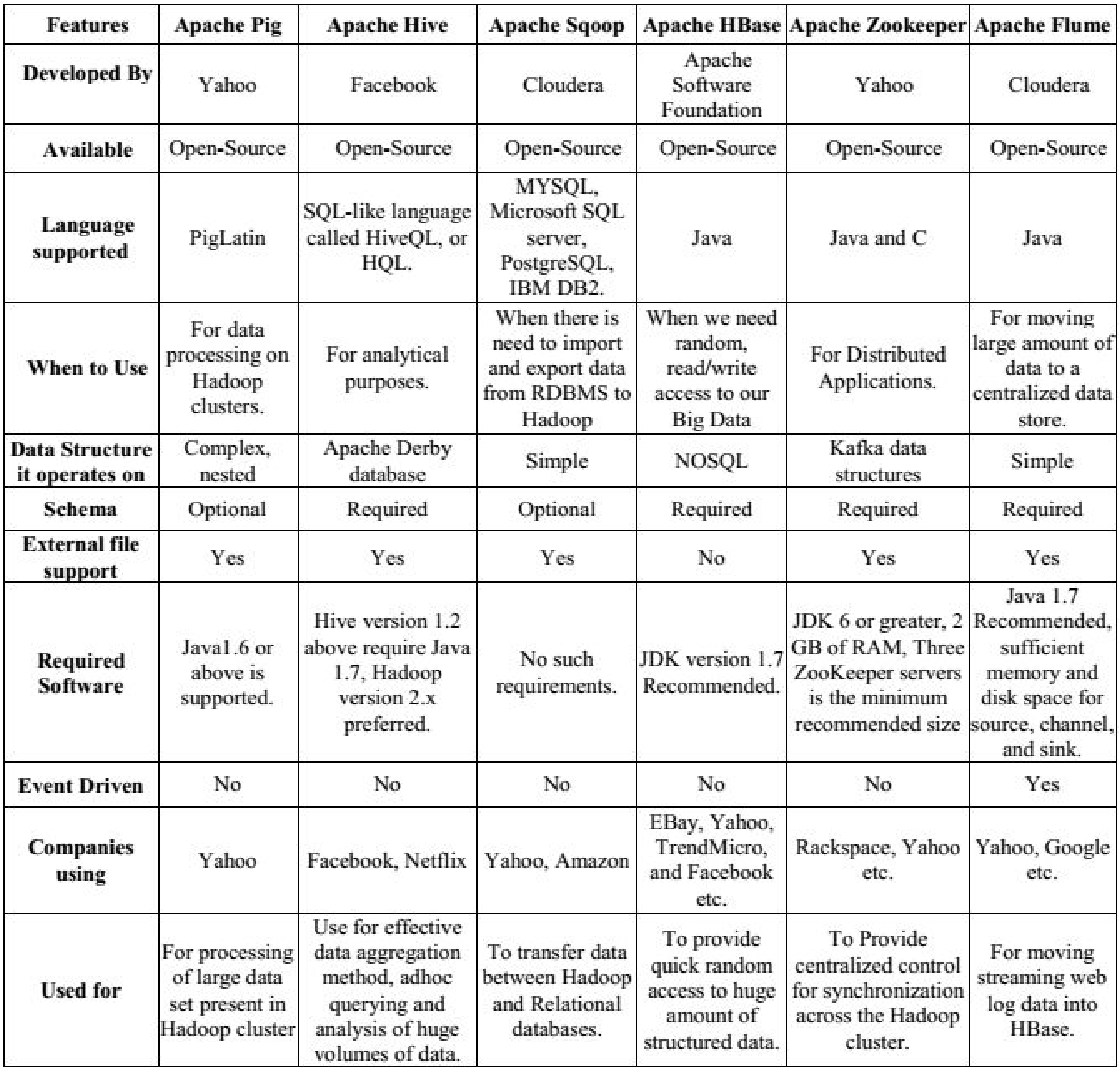
Zookeeper

Zookeeper is a distributed ,open source coordination service enabling synchronization across a cluster. Zookeeper runs in Java and contains bindings for Java and C. It is basically a tool for managing jobs in the cluster. Because of the difficulty of distributed applications to coordinate, manage race conditions and deadlocks, zookeeper is an important part of Hadoop that handles these issues with its properties such as serialization, synchronization and atomicity. With these properties, zookeeper achieves co-ordination and high availability.

Oozie

Apache Oozie works as a scheduler for Hadoop.Directed Acyclic Graphs are used by the users to specify dependencies between jobs. This information is then consumed by Oozie and is executed in the correct order of the specified workflow. Using the Web Service APIs of Oozie, one can control jobs from anywhere. Oozie aids in scheduling jobs periodically and notifies the user at flagged stages. Oozie is extensible and enables scalability and data awareness.

Below is a comparison of all the data access and storage components in Hadoop.



Features	Apache Pig	Apache Hive	Apache Sqoop	Apache HBase	Apache Zookeeper	Apache Flume
Developed By	Yahoo	Facebook	Cloudera	Apache Software Foundation	Yahoo	Cloudera
Available	Open-Source	Open-Source	Open-Source	Open-Source	Open-Source	Open-Source
Language supported	PigLatin	SQL-like language called HiveQL, or HQL.	MYSQL, Microsoft SQL server, PostgreSQL, IBM DB2.	Java	Java and C	Java
When to Use	For data processing on Hadoop clusters.	For analytical purposes.	When there is need to import and export data from RDBMS to Hadoop	When we need random, read/write access to our Big Data	For Distributed Applications.	For moving large amount of data to a centralized data store.
Data Structure it operates on	Complex, nested	Apache Derby database	Simple	NOSQL	Kafka data structures	Simple
Schema	Optional	Required	Optional	Required	Required	Required
External file support	Yes	Yes	Yes	No	Yes	Yes
Required Software	Java1.6 or above is supported.	Hive version 1.2 above require Java 1.7, Hadoop version 2.x preferred.	No such requirements.	JDK version 1.7 Recommended.	JDK 6 or greater, 2 GB of RAM, Three ZooKeeper servers is the minimum recommended size	Java 1.7 Recommended, sufficient memory and disk space for source, channel, and sink.
Event Driven	No	No	No	No	No	Yes
Companies using	Yahoo	Facebook, Netflix	Yahoo, Amazon	EBay, Yahoo, TrendMicro, and Facebook etc.	Rackspace, Yahoo etc.	Yahoo, Google etc.
Used for	For processing of large data set present in Hadoop cluster	Use for effective data aggregation method, adhoc querying and analysis of huge volumes of data.	To transfer data between Hadoop and Relational databases.	To provide quick random access to huge amount of structured data.	To Provide centralized control for synchronization across the Hadoop cluster.	For moving streaming web log data into HBase.

Table no.1 Comparative analysis of Hadoop components

Comparison

RDBMS vs Hadoop

A huge change in the world of storage and processing came with the introduction of Big Data and Hadoop, almost on the verge of overthrowing the traditional relational database management system. Main differences are listed below:

Volume of data

The quantity of data that is being stored and processed is huge in Hadoop (in Terabytes and Petabytes) as compared to the RDBMS system (in Gigabytes). Hadoop can easily process and store large amount of data quite effectively as compared to the traditional RDBMS.

Architecture

Hadoop architecture consists of the following core components: HDFS(Hadoop Distributed File System), Hadoop MapReduce(a programming model to process large data sets) and Hadoop YARN(used to manage computing resources in computer clusters). On the other hand RDBMS has ACID properties (Atomicity, Consistency, Isolation and Durability) which are responsible to maintain and ensure data integrity and accuracy when a transaction takes place in database.

Throughput

Throughput is defined as the ability to maximise the amount of data being processed in a particular time period. RDBMS fails to achieve high throughput as compared to Hadoop framework.

Data Variety

Traditional RDBMS allowed only structured data to be entered into the systems which was a time consuming task in itself. With Hadoop the ability to process and store all variety of data, i.e. structured, semi-structured and unstructured, is made possible.

Latency

Although Hadoop has higher throughput but it takes more time than RDBMS systems when it comes to fetching or accessing a particular record from the data. RDBMS takes very little time to access data provided there is a small amount of data.

Scalability

RDBMS provides vertical scaling, which means that you can add more resources or hardwares such as memory, CPU to a machine in the cluster. On the other hand, Hadoop provides horizontal scalability known as ‘Scaling Out’ a machine. It means that it allows to add more machines to the existing computer clusters as a result of which Hadoop becomes a fault tolerant. Due to the presence of these “back up” machines you can easily recover data irrespective of failure of one of the machines.

Data Processing

Hadoop uses OLAP (Online Analytical Processing) which involves complex queries and aggregations. The database design is denormalized having fewer tables. Whereas, RDBMS supports OLTP (Online Transaction Processing). The database design is normalized having a large number of tables.

Cost

Hadoop is a free and open source software framework whereas RDBMS is a licensed software and you have to pay in order to buy the complete software license.

Analysis

Advantages

Hadoop is very scalable when storing very large datasets. Hadoop allows companies to run application on thousands of nodes that can now handles thousands of terabytes of data. Having the ability to run thousands of terabytes of data also provides means that it is very fast. Hadoop can efficiently and effectively process large amounts of terabytes of data both structured and unstructured data in minutes and petabytes in hours. Lastly, Hadoop is also extremely cost effective. Storing massive amounts of data can be expensive but Hadoop gives the option to reduce cost opting out of traditional database storing.

Disadvantages

Although Hadoop may have great intentions and with analyzing large amounts of data, it also has its drawbacks. Hadoop unfortunately is not fit for small data, because it has the capacity to store massive dataset that a small file will not be supported properly, and it is not recommended. Hadoop may also be exposed to potential stability issues. Some organizations use a third-party vendor to ensure they are responsible for running the latest stable version of Hadoop. The most critical drawback of Hadoop is that there are a lot of security concerns and how it is missing the encryption portion for storing the data. Companies can bypass some of these critical concerns can opt to using third party vendors to help secure their data and handle any stability issues.

Summary

Hadoop provides us with a framework which is easily scalable, highly flexible and resistant to failure. The architecture follows a Master-Slave architecture where the functional part is handled by the Master and most of the clusters make up the slave nodes. Hadoop's architecture consists of three components- HDFS, MapReduce, and YARN. HDFS is scalable, flexible, and reliable and can be used with a wide variety of software and hardware. Similarly, MapReduce is also scalable and flexible like HDFS and can also be used on commodity hardware. YARN is a resource manager which provides the much-needed management layer to Hadoop and make it more available and integrated. For the ease of data access and management, abstractions such as Hive and Pig are built on top of Hadoop. Hive is used by users who are comfortable with SQL like environment as it has HiveQL. Pig highly reduces development time, with the significantly low number of lines of code of Pig-Latin as compared to Java. We also have tools such as Zookeeper and Oozie for managing the high velocity of data with parallel processing and job scheduling. Tools such as Sqoop and flume are reliable, customizable and flexible to ingest data and overcome the challenges of Data Ingestion in Hadoop.

Citation

- Dean, J. and Ghemawat, S. (2004). *MapReduce: Simplified Data Processing on Large Clusters*. Google, Inc.
- Shvachko, K., Kuang, H., Radia, S. and Chansler, R. (2010). *The Hadoop Distributed File System*. Yahoo!
- Vavilapalli, V., Murthy, A., Douglas, C., Agarwal, S., Konar, M., Evans, R., Graves, T., Lowes, J., Shah, H., Seth, S., Saha, B., Curino, C., O'Malley, O., Radia, S., Reed, B. and Baldeschweiler, E. (n.d.). *Apache Hadoop YARN: Yet Another Resource Negotiator*.
- Bhardwaj, A., Kumar, A., Narayan, Y., & Kumar, P. (2015, December). Big data emerging technologies: A Case Study with analyzing twitter data using apache hive. In *2015 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS)* (pp. 1-6). IEEE.
- “What Is Big Data?” *Oracle*, https://www.oracle.com/big-data/guide/what-is-big-data.html.
- Soubra, Diya, et al. “The 3Vs That Define Big Data.” *Data Science Central*, https://www.datasciencecentral.com/forum/topics/the-3vs-that-define-big-data.
- “What Is Hadoop?” *SAS*, https://www.sas.com/en_us/insights/big-data/hadoop.html.
- “Big Data Volume, Variety, Velocity and Veracity.” *InsideBIGDATA*, 19 July 2019, https://insidebigdata.com/2013/09/12/beyond-volume-variety-velocity-issue-big-data-veracity/.
- “Why Big Data and Hadoop Are so Often Spoken in the Same Breath, Explained.” *Tableau Software*, https://www.tableau.com/learn/articles/big-data-hadoop-explained.
- “9 Ways Retailers Are Using Big Data and Hadoop.” *Datanami*, 27 July 2016, https://www.datanami.com/2016/07/20/9-ways-retailers-using-big-data-hadoop/.

- Team, DataFlair. “13 Big Limitations of Hadoop & Solution To Hadoop Drawbacks.” *DataFlair*, 7 Mar. 2019, https://data-flair.training/blogs/13-limitations-of-hadoop/.
- Achari, Shiva. *Hadoop Essentials Delve into the Key Concepts of Hadoop and Get a Thorough Understanding of the Hadoop Ecosystem*. Packt Publishing, 2015.
- Raphael, Roopa, and Raj Kumar. *Big Data, RDBMS and HADOOP- A Comparative Study* . College of Engineering Kallooppara, Kerala .