 ISBN: 978-81-927230-0-9

Overview of Hadoop: An Open - Source Big Data Analytic Framework

Ms. Leena More (Deshmukh)

Research Scholar, JJT University, Rajasthan and Asst. Prof., JSPM's JIMS, Tathawade
linadeshmukh@gmail.com

Dr. Manik Kadam

Research Guide, Allana Institute of Management, Pune.

Abstract **- Now a days as size of data is going to be increasing there is a necessity of Big data technology which must support search, development, governance and analytics services for all data type; from transaction and application data to machine and sensor data to social, image and geospatial data, and more. Hadoop is a framework to store and process large amount of data quickly. Its distributed computing model processes data fast. It prevents data processing from hardware failure by making replicas on multiple processors. It can store unstructured data also like text, images and videos. Data lake support to store the data in its original format which helps to generate new queries without constraints. Apache Foundation included four modules in Hadoop basic framework. Namely; Hadoop Common, HDFS, YARN, MapReduce.**

Keywords - Data lake, SandBox, HADOOP, IoT, HDFS, YARN, MAPReduce, Sqoo, HCatalog, HBase, Hive, Pig, Solr, Ambari, Flume, Oozie, Cassandra, Spark, Zookeeper, distros, Cloudera, Hortonworks, MapR, IBM BigInsights and PivotalHD

I. INTRODUCTION

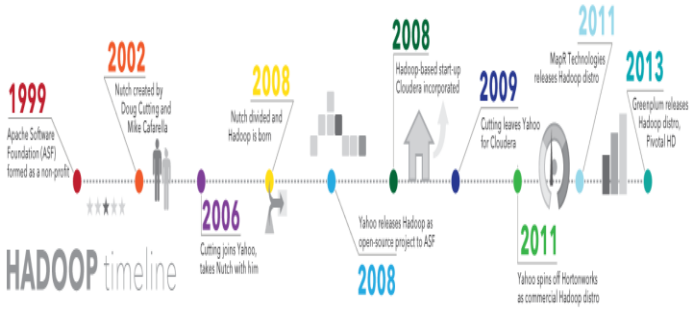
Big data is being generated by everything around us at all times. Every digital process and social media exchange produces it. Systems, sensors and mobile devices transmit it. Big data is arriving from multiple sources at an alarming velocity, volume and variety. To extract meaningful value from big data, you need optimal processing power, analytics capabilities and skills.

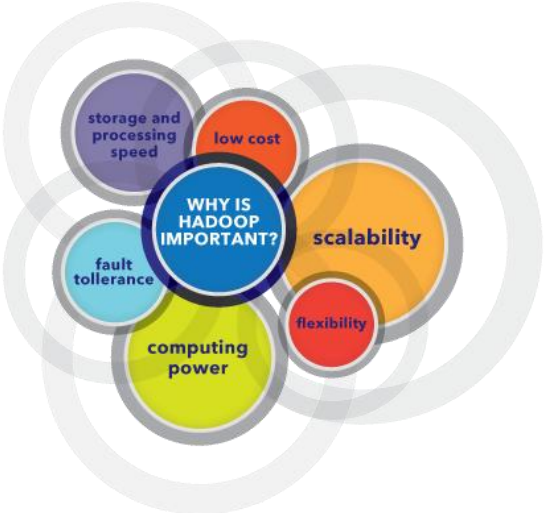
Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs.

Hadoop History

As the World Wide Web grew in the late 1900s and early 2000s, search engines and indexes were created to help locate relevant information amid the text-based content. In the early years, search results were returned by humans. But as the web grew from dozens to millions of pages, automation was needed. Web crawlers were created, many as university-led research projects, and search engine start-ups took off (Yahoo, AltaVista, etc.).

One such project was an open-source web search engine called Nutch – the brainchild of Doug Cutting and Mike Cafarella. They wanted to return web search results faster by distributing data and calculations across different computers so multiple tasks could be accomplished simultaneously. During this time, another search engine project called Google was in progress. It was based on the same concept – storing and processing data in a distributed, automated way so that relevant web search results could be returned faster.





In 2006, Cutting joined Yahoo and took with him the Nutch project as well as ideas based on Google's early work with

automating distributed data storage and processing. The Nutch project processing portion became Hadoop (named after Cutting's son's toy elephant). In 2008, Yahoo released Hadoop as an open-source project. Today, Hadoop's framework and ecosystem of technologies are managed and maintained by the non-profit Apache Software Foundation (ASF), a global community of software developers and contributors.

II. IMPORTANCE OF HADOOP

- **Ability to store and process huge amounts of any kind of data, quickly:** With data volumes and varieties constantly increasing, especially from social media and the Internet of Things (IoT), that's a key consideration.
- **Computing power:** Hadoop's distributed computing model processes big data fast. The more computing nodes you use the more processing power you have.
- **Fault tolerance:** Data and application processing are protected against hardware failure. If a node goes down, jobs are automatically redirected to other nodes to make sure the distributed computing does not fail. Multiple copies of all data are stored automatically.
- **Flexibility:** Unlike traditional relational databases, you don't have to preprocess data before storing it. You can store as much data as you want and decide how to use it later. That includes unstructured data like text, images and videos.
- **Low cost:** The open-source framework is free and uses commodity hardware to store large quantities of data.
- **Scalability:** You can easily grow your system to handle more data simply by adding nodes. Little administration is required.

III. CHALLENGES OF USING HADOOP

- **MapReduce programming is not a good match for all problems:** It's good for simple information requests and problems that can be divided into independent units, but it's not efficient for iterative and interactive analytic tasks. MapReduce is file-intensive. Because the nodes don't intercommunicate except through sorts and shuffles, iterative algorithms require multiple map-shuffle/sort-reduce phases to complete. This creates multiple files between MapReduce phases and is inefficient for advanced analytic computing.
- **There's a widely acknowledged talent gap:** It can be difficult to find entry-level programmers who have sufficient Java skills to be productive with MapReduce. That's one reason distribution providers are racing to put relational (SQL) technology on top of Hadoop. It is much easier to find programmers with SQL skills than MapReduce skills. And, Hadoop administration seems part art and part science, requiring low-level knowledge of operating systems, hardware and Hadoop kernel settings.
- **Data security:** Another challenge center around the fragmented data security issues, though new tools and technologies are surfacing. The Kerberos authentication protocol is a great step toward making Hadoop environments secure.
- **Full-fledged data management and governance:** Hadoop does not have easy-to-use, full-feature tools for data management, data cleansing, governance and metadata. Especially lacking are tools for data quality and standardization.

Data management for Hadoop Big data skills are in high demand. Now business users can profile, transform and cleanse data – on Hadoop or anywhere else it may reside – using an intuitive user interface.

How Is Hadoop Being Used?

Going beyond its original goal of searching millions (or billions) of web pages and returning relevant results, many organizations are looking to Hadoop as their next big data platform. Popular uses today include:

- **Low-cost storage and data archive**

The modest cost of commodity hardware makes Hadoop useful for storing and combining data such as transactional, social media, sensor, machine, scientific, click streams, etc. The low-cost storage lets you keep information that is not deemed currently critical but that you might want to analyze later.

- **Sandbox for discovery and analysis**

Because Hadoop was designed to deal with volumes of data in a variety of shapes and forms, it can run analytical algorithms. Big data analytics on Hadoop can help your organization operate more efficiently, uncover new opportunities and derive next-level competitive advantage. The sandbox approach provides an opportunity to innovate with minimal investment.

- **Data lake**

Data lakes support storing data in its original or exact format. The goal is to offer a raw or unrefined view of data to data scientists and analysts for discovery and analytics. It helps them ask new or difficult questions without constraints. Data lakes are not a replacement for data warehouses. In fact, how to secure and govern data lakes is a huge topic for IT. They may rely on data federation techniques to create a logical data structures.

- **Complement your data warehouse**

Now Hadoop beginning to sit beside data warehouse environments, as well as certain data sets being offloaded from the data warehouse into Hadoop or new types of data going directly to Hadoop. The end goal for every organization is to have a right platform for storing and processing data of different schema, formats, etc. to support different use cases that can be integrated at different levels.

- **IoT and Hadoop**

Things in the IoT need to know what to communicate and when to act. At the core of the IoT is a streaming, always on torrent of data. Hadoop is often used as the data store for millions or billions of transactions. Massive storage and processing capabilities also allow you to use Hadoop as a sandbox for discovery and definition of patterns to be monitored for prescriptive instruction. You can then continuously improve

 ISBN: 978-81-927230-0-9

these instructions, because Hadoop is constantly being updated with new data that doesn't match previously defined patterns.

Building a recommendation engine in Hadoop

One of the most popular analytical uses by some of Hadoop's largest adopters is for web-based recommendation systems. Facebook, LinkedIn, Netflix, eBay and Hulu; these systems analyze huge amounts of data in real time to quickly predict preferences before customers leave the web page.

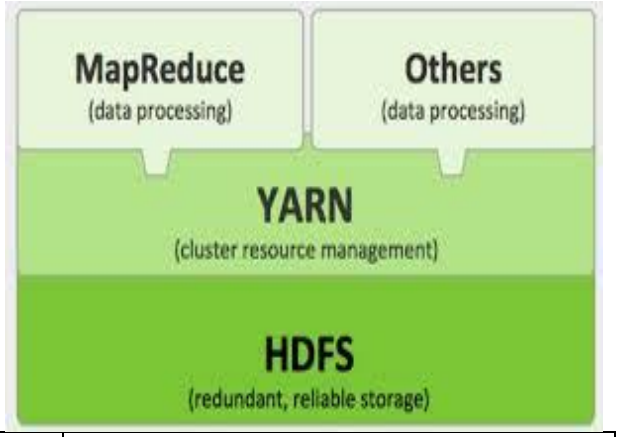
How: A recommender system can generate a user profile explicitly (by querying the user) and implicitly (by observing the user's behavior) – then compares this profile to reference characteristics (observations from an entire community of users) to provide relevant recommendations. SAS provides a number of techniques and algorithms for creating a recommendation system, ranging from basic distance measures to matrix factorization and collaborative filtering – all of which can be done within Hadoop.

IV. How It Works and a Hadoop Lexicon

Currently, four core modules are included in the basic framework from the Apache Foundation:

- **Hadoop Common** – the libraries and utilities used by other Hadoop modules.
- **Hadoop Distributed File System (HDFS)** – the Java-based scalable system that stores data across multiple machines without prior organization.
- **YARN** – (Yet Another Resource Negotiator) provides resource management for the processes running on Hadoop.
- **MapReduce** – a parallel processing software framework. It is comprised of two steps. Map step is a master node that takes inputs and partitions them into smaller subproblems and then distributes them to worker nodes. After the map step has taken place, the master node takes the answers to all of the subproblems and combines them to produce output.

Other software components that can run on top of or alongside Hadoop and have achieved top-level Apache project status include:



Ambari	A web interface for managing, configuring and testing Hadoop services and components.
Cassandra	A distributed database system.
Flume	Software that collects, aggregates and moves large amounts of streaming data into HDFS.
HBase	A nonrelational, distributed database that runs on top of Hadoop. HBase tables can serve as input and output for MapReduce jobs.
HCatalog	A table and storage management layer that helps users share and access data.
Hive	A data warehousing and SQL-like query language that presents data in the form of tables. Hive programming is similar to database programming.
Oozie	A Hadoop job scheduler.
Pig	A platform for manipulating data stored in HDFS that includes a compiler for MapReduce programs and a high-level language called Pig Latin. It provides a way to perform data extractions, transformations and loading, and basic analysis without having to write MapReduce programs.
Solr	A scalable search tool that includes indexing, reliability, central configuration, failover and recovery.

 ISBN: 978-81-927230-0-9

Spark	An open-source cluster computing framework with in-memory analytics.
Sqoop	A connection and transfer mechanism that moves data between Hadoop and relational databases.
Zookeeper	An application that coordinates distributed processing.

V. GETTING DATA INTO HADOOP

Here are just a few ways to get your data into Hadoop.

- Use third-party vendor connectors (like SAS/ACCESS® or SAS Data Loader for Hadoop).
- Use Sqoop to import structured data from a relational database to HDFS, Hive and HBase. It can also extract data from Hadoop and export it to relational databases and data warehouses.
- Use Flume to continuously load data from logs into Hadoop.
- Load files to the system using simple Java commands.
- Create a cron job to scan a directory for new files and "put" them in HDFS as they show up. This is useful for things like downloading email at regular intervals.
- Mount HDFS as a file system and copy or write files there.

VI. CONCLUSION

Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs.

Open-source software is created and maintained by a network of developers from around the world. More and more commercial versions of Hadoop are becoming available (these are often called "distros.") With distributions from software vendors, you pay for their version of the Hadoop framework and receive additional capabilities related to security, governance, SQL and management/administration consoles, as well as training, documentation and other services. Popular distros include Cloudera, Hortonworks, MapR, IBM BigInsights and PivotalHD.

REFERENCES

1. *Lam, Chuck (July 28, 2010). Hadoop in Action (1st ed.). Manning Publications. p. 325. ISBN 1-935-18219-6.*
2. *Venner, Jason (June 22, 2009). Pro Hadoop (1st ed.). Apress. p. 440. ISBN 1-430-21942-4.*
3. *White, Tom (June 16, 2009). Hadoop: The Definitive Guide (1st ed.). O'Reilly Media. p. 524. ISBN 0-596-52197-9.*
4. *Evans, Chris (Oct 2013). "Big data storage: Hadoop storage basics". computerweekly.com. Computer Weekly.*
5. *J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In Sixth Symposium on Operating System Design and Implementation, pages 137–150, December 2004.*
6. *S. Ghemawat, H. Gobioff, and S. Leung. The Google file system. In ACM SOSP, October 2003., 2003.*
7. *F. Schmuck and R. Haskin. GPFS: A shared-disk file system for large computing clusters. In Proc. of the First Conference on File and Storage Technologies (FAST), pages 231–244, Jan. 2002.*
8. *W. Tantisiriroj, S. Patil, and G. Gibson. The crossing the chasm: Sneaking a parallel file system into hadoop. In SC08 Petascale Data Stroage Workshop, 2008.*
9. *Jeffrey Dean, Sanjay Ghemawat (2004) MapReduce: Simplified Data Processing on Large Clusters, Google.*
10. *Michael Franklin, Alon Halevy, David Maier (2005) From Databases to Dataspaces: A New Abstraction for Information Management.*
11. *Fay Chang et al. (2006) Bigtable: A Distributed Storage System for Structured Data, Google.*
12. *Robert Kallman et al. (2008) H-store: a high-performance, distributed main memory transaction processing system*
13. *http://www.sas.com/en_us/insights/big-data/hadoop.html*
14. *Hadoop Distributed Filesystem. http://hadoop.apache.org. ss*