International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

Distinctive Advancements in Bigdata Persistence and Processing

Krishnaraj Rajagopal, Kumar Narayanan

Abstract: ***The Goal of this paper aims to detail the exploration of Big Data and different technology stack to handle and process Big Data to make more insights. In recent years the digit data is growing more and more. There are so many sources of these Big Data like mobiles, videos, social media like facebook/twitter, images, application logs, data generated from the system, etc. We need a special storage and processing system to analyze these Big Data.***

Keywords: Big Data, Distributed Data Storage, Distributed Parallel Processing.

I.INTRODUCTION

In the digital era, we are generating a huge amount of digital data in different ways like social media, videos, images, system logs and etc. According to recent statistics, nighty percentages of the digital data in the world has been created in the last two years. The impact of AI in marketing is growing and it's predicted to reach nearly 40 billion dollars by 2025. Big Data is the field that is used to extract information from a huge volume of data. On a major level data will be considered as Big Data in terms of three characteristics namely Volume, Velocity and Variety. Volume level the data will be considered as Big Data based on its sizes like MB, GB, TB, and PB. Velocity is nothing but how fast the data is coming from the source. Variety as the name implies it is different kinds of data formats like database, documents, pictures, videos, etc.

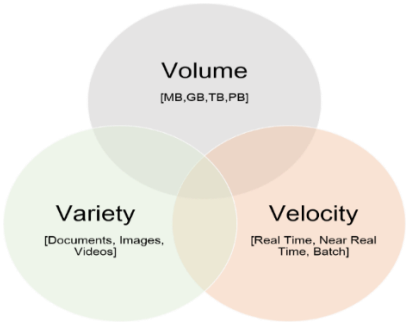

Figure 1. Three V's of Big Data

II.BIG DATA TECHNOLOGY STACK

There are so many big data technologies are available in the market. Different technologies are used in a different layer like the ingestion layer, processing layer and modelled layer with different purposes. Ingestion is nothing but ingesting data from the source system to the Hadoop environment.

A. Sqoop

There are different methods and technologies are used to ingest data from a source system to Hadoop. Sqoop is used to ingest data from relational databases (RDBMS) like Oracle, DB2, etc. to Hadoop distributed file system (HDFS). Sqoop has two main functionalities called Import and Export. Import is nothing but moving data from relational databases to Hadoop. Export is another way around, moving data from Hadoop to relation databases.

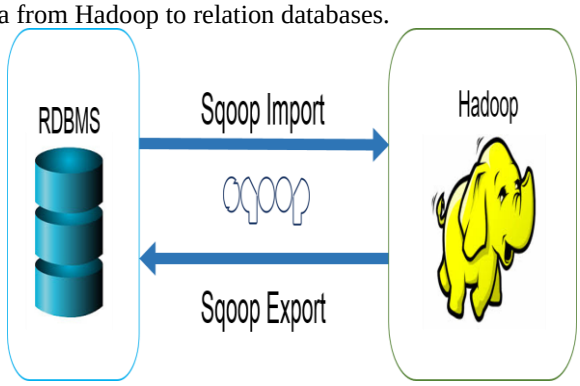

Figure 2. Sqoop Import and Export

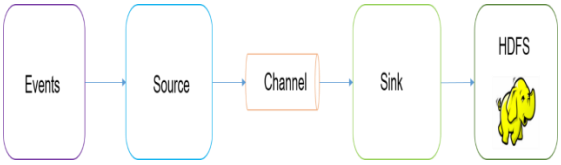

Figure 3. Apache Flume Components

B. Flume

It is another efficient technology used for ingesting data to Hadoop. This is mainly used to collect, aggregate and move the streaming data or event to Hadoop [1]. It is one of the best suitable technology to ingest data to Hadoop in near real-time. For near real-time analytics flume is one of the best ingestion methodologies. Flume has mainly three blocks called source, channel and sink. Source – Source of the data. Channel – This is an intermittent layer which is used to transfer the data from source to sick. There are different types of flume channels like memory channel and file channel. Sink – This is the target layer where we want to send the data. There are different types of sink supported by a flume, one among them is HDFS.

C. HDFS

HDFS Stands for Hadoop Distributed File System [2]. It is a distributed data storage. Many nodes can be integrated called cluster is used to store the data. Big Data will be sliced into multiple small chunks and those chunks will be stored in the data nodes.

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

Krishnaraj Rajagopal, Research Scholar, Department of Computer science and Engineering Vels Institute of Science, Technology and Advanced Studies, Chennai, India krishnaraj.rajagopal@gmail.com

Kumar Narayanan, Associate Professor, Department of Computer science and Engineering Vels Institute of Science, Technology and Advanced Studies, Chennai, India kumar.se@velsuniv.ac.in

There is one more daemon called name node used to maintain the metadata information. We can easily increase the cluster size by simply adding many nodes to cluster called commissioning node. Remove the faulty node from the cluster is generally called as decommissioning node. We don't need a special configuration node to be part of Hadoop cluster. Commodity machines can be part of HDFS distributed data storage. HDFS has many advantages in its architecture. Fault-Tolerant is one of the main advantage. In a distributed environment, the possibility the any node goes down is high. Even any one of the node contains part of the data, goes down, then there will be a data loss. To avoid this, It is been designed with a concept called replication. Replication – Replicate the data to a different node to avoid data loss in case of node failure. Default replication factor is three.

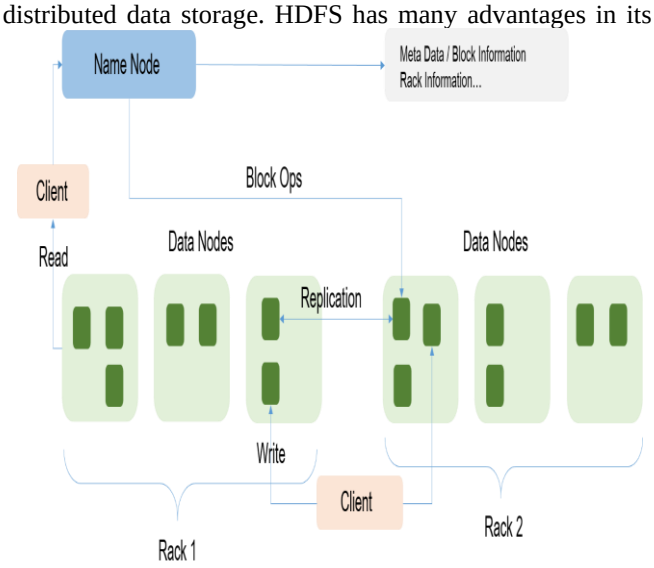

Figure 4. HDFS Architecture

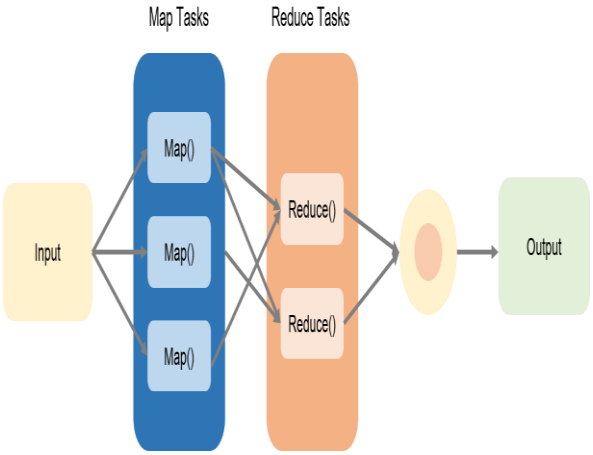

Figure 5. Map Reduce Flow

D. Map Reduce

Map Reduce is a distributed parallel processing framework. It is mainly used in batch processing [3]. It has two phases called Map and Reduce Phase. Filtering, Sorting will be performed in Map method. Summary operation like count and etc. will be performed in reduce method. Map reduce frame works on <Key, Value> pairs. Input to the framework is <Key, Value> pairs and output of the framework is <Key, Value> pairs.

Map: Big Data will be sliced into multiple small size data called split. These splits send to Map method to process. Work nodes applies the map function to local data and create intermediate output to a temporary storage. Sorting: Sorting the keys based on the lexicographic algorithm.

Reduce: Shuffle the data and ensure same set of keys are in the same partitions to minimize the work for reducer. Intermediate output generated by Map phase will be the input for reduce phase in this phase mostly summary operation will be performed.

E. Hive

Hive is a data warehouse platform build on top of the Hadoop. It supports SQL like query to process and analyze the data resides in the Hadoop environment [5]. It converts the SQL like query into map reduce program and submit the map reduce job to get the desired output. It will be ease to write compared to write a map reduce programs. Hive architecture has many components in its like Meta Store, Driver, CLI etc. Hive Metadata contains the Metadata information's like tables location, table name. It generally persist in relation database. Driver contains many blocks namely Compiler to parse the Hive Query language (HQL) to Map reduce programs and Optimizer to optimize to HQL to optimized Map Reduce program.

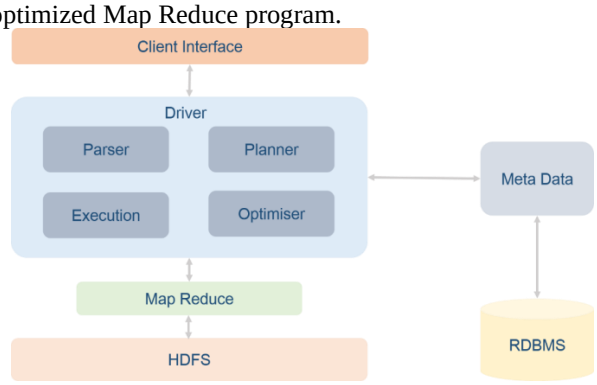

Figure 6. Hive Architecture

Elements	Spark	Map Reduce
Speed	100 x times faster than Map Reduce	Faster than Traditional System
Language	Scala	Java
Data Processing	Batch Processing/Real Time Processing/Interactive/Graph	Batch Processing
Usage	Compact and Easier	Complex and More codes
Caching	Caching data in memory improves the performance	Does not support Caching

Table 1. Comparison between Spark and Map Reduce

F. Spark

It one of the open source in cluster computing system. It has significant supports low latency queries [4]. Performance wise it is much faster in Batch and Streaming application because of its DAG scheduler, Query Optimizer, Physical Execution Engine. It supports many languages like Java, Python, Scala, R, and SQL. Spark is much faster compared to map reduce because of its in memory caching methodology.

III.CONCLUSION

This paper assesses the complete education on big data and various characterizes of big data like volume, velocity and variety. Survey about Big Data technologies along with their architecture and usage. Comparison between different technologies based on its performance has been explained in a tabular format.

REFERENCE

1. "Apache Flume." [Online]. Available: http://flume.apache.org/
2. K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop distributed file system," in 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies, MSST2010, 2010
3. J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Commun. ACM, vol. 51, no. 1, pp. 1–13, 2008
4. "Spark."[Online].Available: https://spark.incubator.apache.org/.
5. A. Thusoo, J. Sen Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang, S. Antony, H. Liu, and R. Murthy, "Hive - a petabyte scale data warehouse using Hadoop," in 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010), 2010, pp. 996–1005