Big Data: New Trend to Handle Big Data

Manish Kumar Gupta
Assistant Professor, Amity University, Greater Noida Campus
manish.testing09@gmail.com

Abstract:
In current scenario, enormous amounts of data are available and grow day by day. Bid data refers that, they are not only in volume, but also in variety, velocity and veracity, which make difficult to handle using traditional approach and techniques. So, overcome to this, a new approach has been introduced, called Hadoop and Hadoop2(YARN). With help of these we can analyze and produce result not only structured data but also semi structured and unstructured data. Aim of this paper to analyze some tools which can be applied to big data.

Keywords: Varirty, Velocity, Volume, YARN, Data, Structured Data, Semi Structured, Unstructured Data.

1. Introduction to BIG DATA

Big data is defined as the voluminous amount of structured, semi- structured and unstructured data that has huge potential for mining but is so large that it cannot be processed using traditional database systems. Big data can be defined by its high velocity, volume and variety and veracity that require cost effective and innovative methods for information processing to draw meaningful business insights [1]. More than the volume of the data – it is the nature of the data that defines whether it is considered as Big Data or not [2].

A. What do the four V's of Big Data denote

IBM has a nice, simple explanation for the four critical features of big data:

a) Volume [1] –Scale of data
b) Velocity [1] –Analysis of streaming data
c) Variety [1] – Different forms of data
d) Veracity [1]–Uncertainty of data

2. Concept of HADOOP Framework

Hadoop Framework works on the following two core components-data [3].

BackupNode: Backup Node also provides check pointing functionality like that of the checkpoint node, but it also maintains its up to date in-memory copy of the file system namespace that is in sync with the active NameNode [1].

3. Commodity Hardware

Commodity Hardware refers to inexpensive systems that do not have high availability or high quality [3]. Commodity Hardware consists of RAM because there are specific services that need to be executed on RAM. Hadoop can be run on any commodity hardware and does not require any supercomputer s or high end [2].

4. HIVE

As we know Hadoop support any type of data (Structured, Un Structured and Semi Structured), HIVE is used for processing Structured Data [3]. HIVE is like by Non- JAVA programmer, if any one happy with Query language then they can choose HIVE. HIVE is given by FACEBOOK (HiveQL). HiveQL is a Case Insensitive Language. The Apache Hive ™ data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL. Structure can be projected onto data already in storage. A command line tool and JDBC driver are provided to connect users to Hive [7].

5. PIG

Pig has given its own language called PIG latin Scripting. It is a data flow language [3]. YAHOO introduce pig. PIG is used for both Structured and Un Structured data [3].

Apache Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets [8].

A. HDFS

Hadoop Distributed File System is the java-based file system used for store large set of data in streamline access pattern. Data in HDFS is stored in the form of blocks and it operates on the Master Slave Architecture

B. Hadoop MapReduce

This is a java-based programming paradigm of Hadoop framework that is used for processing the data across various Hadoop clusters [2]. It distributes the workload into various tasks that can run in parallel. Map Reduce can be divided in to two parts. Map & Reduce. Process of Accessing data through Data Node from Task Tracker is known as a Map function and reducing the output in a single Data Node is known as Reduce [3].

6. Main Concept of A HADOOP Application

Hadoop applications have wide range of technologies that provide great advantage in solving complex business problems [1].

Core components of a Hadoop application are:
1) Hadoop Common
2) HDFS
3) Hadoop MapReduce
4) YARN

7. Name Node, Backup Node

NameNode: It is at the heart of the HDFS file system which manages the hardware configuration to execute jobs. Name Node store only schema of

8. FLUME

Apache FLUME aids in transferring large amount of data from Distributed resource to a single centralized repository. It is used for real time data capturing in Hadoop. A common case where flume is used to collecting the log data from different system and collect the aggregate data into HDFS for later analysis [3].

Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows. It is robust and fault tolerant with tunable reliability mechanisms and many failover and recovery mechanisms. It uses a simple extensible data model that allows for online analytic application [9].

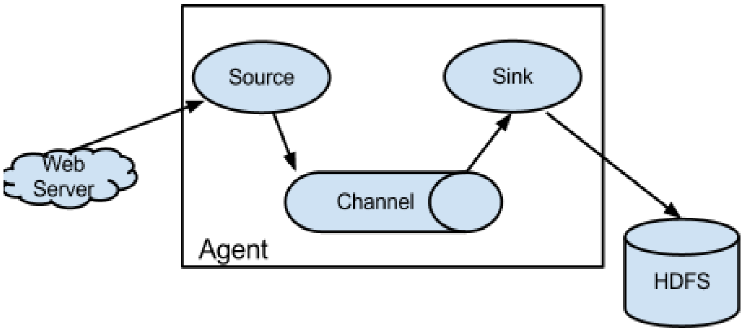

Fig. 1. Flume workflow

9. OOZIE

OOZIE is a server. OOZIE is coordinate one job to another job [3]. OOZIE is also called workflow manager. OOZIE has two nodes [1] Control Node- Where the job will be stored.

Action Node: How to execute a Job.

Oozie is integrated with the rest of the Hadoop stack supporting several types of Hadoop jobs out of the box (such as Java map-reduce, Streaming map-reduce, Pig, Hive, Sqoop and Distcp) as well as system specific jobs (such as Java programs and shell scripts) [10].

10. Map Reduce

Map reduce is a programming model design for processing large volume of data in parallel by dividing the work into a set of independent tasks [1]. In MapReduce, during the map phase it counts the words in each document, while in the reduce phase it aggregates the data as per the document spanning the entire collection. During the map phase the input data is divided into splits for analysis by map tasks running in parallel across Hadoop framework [2].

The process by which the system performs the sort and transfers the map outputs to the reducer as inputs is known as the shuffle [3].

11. What's New in HDFS 2.0

As you learned in "Introduction to Hadoop, Its Architecture, Ecosystem, and Microsoft Offerings," HDFS in Hadoop 1.0 had some limitations and lacked support for providing a highly available distributed storage system. Consider the following limitations of Hadoop 1.0, related to HDFS; Hadoop 2.0 and HDFS 2.0 have resolved them [4].

Single point of failure: Hadoop Name Node is a single point of failure. Although you can have a secondary name node in Hadoop 1.0, it's not a standby node, so it does not provide failover capabilities. The name node still is a single point of failure [5].

Horizontal scaling performance issue: As the number of data nodes grows beyond 4,000 the performance of the name node degrades. This sets a kind of upper limit to the number of nodes in a cluster [6].

12. HDFS High Availability

In the Hadoop 1.0 cluster, the name node was a single point of failure. Name node failure gravely impacted the complete cluster availability [4]. Taking down the name node for maintenance or upgrades meant that the entire cluster was unavailable during that time [3]. The HDFS High Availability (HA) feature introduced with Hadoop 2.0 addresses this problem [1]. Now you can have two name nodes in a cluster in an active-passive configuration: One node is active at a time, and the other node is in standby mode. The active and standby name nodes remain synchronized. If the active name node fails, the standby name node takes over and promotes itself to the active state. In other words, the active name node is responsible for serving all client requests, whereas the standby name node simply acts as a passive name node—it maintains enough state to provide a fast failover to act as the active name node if the current active name node fails [5]. This allows a fast failover to the standby name node if an active name node crashes, or a graceful failing over to the standby name node by the Hadoop administrator for any planned maintenance [6].

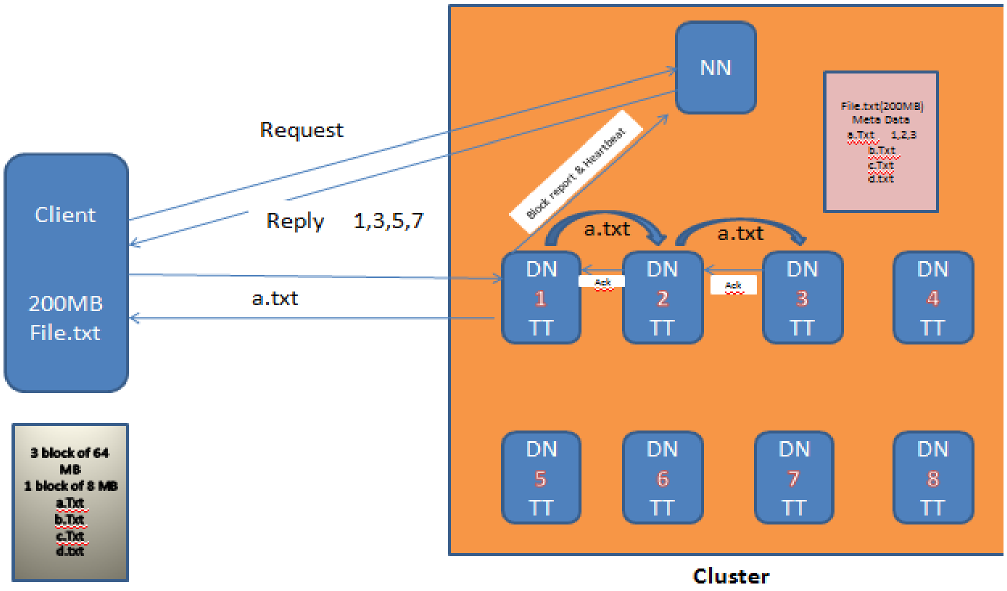

Fig. 2. Architecture of HDFS

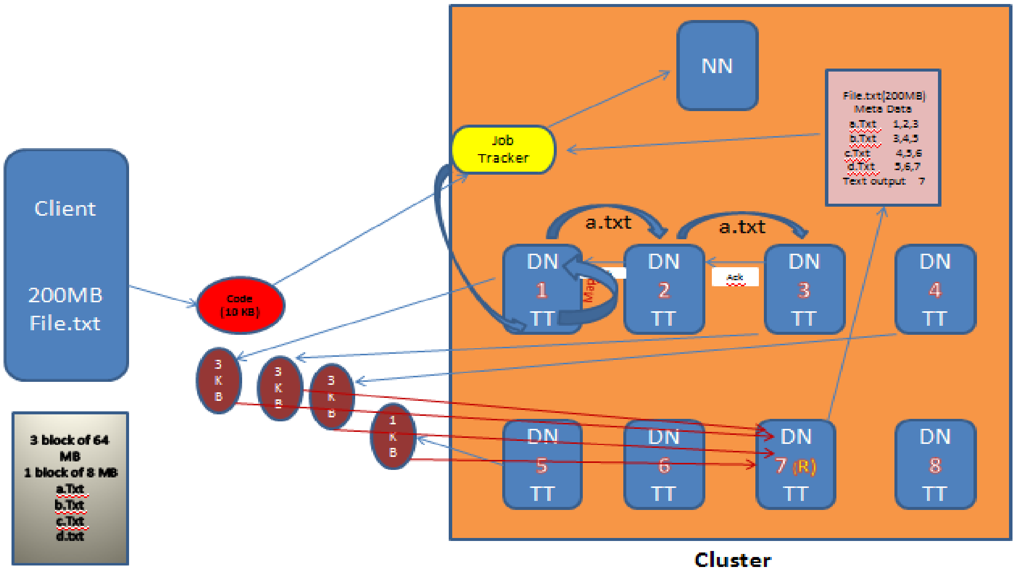

Fig. 3. Architecture of Map Reduce

REFERENCES

[1] DT Editorial Services "Big Data" Black Book.
[2] www.wileyindia.com/big-data-black-book-covers-hadoop-2
[3] Video Lecture of Rama Krishna from Drga Soft.
[4] https://www.tutorialspoint.com.
[5] https://www.data-flair.training.
[6] http://www.informit.com
[7] https://hive.apache.org
[8] https://pig.apache.org/
[9] https://flume.apache.org/
[10] oozie.apache.org/