Application of HADOOP Technologies on Text Processing

Prof.T.Venkat Narayana Rao[1], S. Manminder Kaur[2], Shaik Sohail Ali[3], G. Jahnavi Reddy[4]

[1] Professor, Department of Computer Science and Engineering, Sreenidhi Institute of Science and Technology
tvnrbobby@yahoo.com
[2]Student, Department of Computer Science and Engineering, Sreenidhi Institute of Science and Technology
manminder.1512@gmail.com

[3]Student, Department of Computer Science and Engineering, Sreenidhi Institute of Science and Technology
sohail.loyola@gmail.com

[4]Student, Department of Computer Science and Engineering, Sreenidhi Institute of Science and Technology
Yamnampet, Ghatkesar, Rangareddy-501301, India
Jahnavi12496@gmail.com

Abstract: The outbreak of massive data and the extent of databases used have become a considerable issue in present-day's business organizations as it is facing rapid change in its rate of growth. The amount of data to be handled and interpreted demands modern approach as it has turned to be quicker than the power of computing. "BigData" is used to develop applications of data that are immense and complicated. This paper deals with "Hadoop" which turned up as a prominent tool for performing BigData functionalities. The architecture of Hadoop, its significant approaches i.e. Hadoop Distributed File System (HDFS) and MapReduce and also the two programming languages, Pig and Hive which aids the MapReduce programming quickly within a short span are illustrated in this paper.

Keywords: BigData, HDFS, Hadoop, MapReduce, Pig, Hive.

1. Introduction

The universally developed range of data has an exceptional rise in both volume and variety. This upsurge in data is directed by the propagation of social media and expanding mobile devices [5] [6]. BigData specifies these data sets and the leading-edge technologies to seize, inspect, store and maintain petabytes of structured as well as unstructured data. Thus it is commonly cited as 3 V's i.e. BigData can be précised by volume of data, by variety of data and by velocity for processing the data [1] [2] [4] [5].

BigData grasps the data from any origin and can be integrated with extremely dynamic analytics, through which one can carry out work-relevant tasks like 1.deciding the origin of the breakdown and its weaknesses, 2.identifying forged practices sooner before spreading in the entire organization, 3.calculating integrated portfolios within minutes etc. and facilitates wise decision-making, agile product advancements and also rebate in both cost and time [3] [6].

The management of the data is one of the challenging issues. In order to handle the applications of big data efficiently, an open source framework, "Hadoop" came into presence [4] [7]. It is bedrock for grouping the big data and implements processing of massive data jointly across wide range of servers [1] [8].

2. Apache Hadoop

Hadoop is an open source project endowed by Apache Software Foundation and is established on java. Its prime motive is to enhance substantial amount of data [1] [8]. It is a framework which allows distributed handling of data. It was obtained from Google File System (GFS) and is established on Google's Map Reduce programming standard [4] [6] [7].

Apache Hadoop software is open source and is used for storing and handling immense data sets on extensive range of computers. It turned out as a solution and a prevailing technology for BigData processing. Its services include accession of data, storage of data, altering the data, control over data and data security [1] [4] [6].

The present Apache Hadoop environment encompasses the Hadoop Kernel, Hadoop Distributed File System (HDFS), MapReduce and distinct segments like Hive, Zookeeper and Base while HDFS and MapReduce stand out as the two major components of Hadoop for data storage and data processing respectively [4] [7].

The attributes of Hadoop can be specified as [6]:
1.Scalable i.e. new nodes can be combined independent of the data formats.
2. Profitable i.e. the output of the parallel computing of data makes it economical.
3. Flexible i.e. structured or unstructured data can be accumulated from numerous sources.
4. Fault tolerance i.e. continued operation even if the node is lost and redirecting the task to another point.

3. Hadoop Architecture

A) HDFS

Hadoop has initiated with a distributed file system commonly known as Hadoop Distributed File system (HDFS).This file system has been developed to run on clusters that are commodity hardware [1]. HDFS file system has very high throughput and is fault tolerant, hence is used to store large sets of data from applications. The major boon for HDFS when compared to other file systems is its write-once and read-many time's configuration [5] [6] [9].

Deep down, every file that reaches HDFS is divided into one or more blocks depending on the size of the file as shown in Figure 1. By default the size of a block in HDFS is 64MB. HDFS is designed as a fault tolerant system; this can be achieved by making 3 replicas of all the blocks present in the HDFS, such that even if there is loss of data in one block, same block of data can be replaced [1] [5] [6][10].

HDFS has adopted the master and slave architecture. Every cluster has a single Name Node that serves as a master for that cluster, whose duty is to manage the name space of the file system and respond to the client request regarding blocks of data [9]. Data nodes serve as slaves in the system whose duty is to retrieve and store the information based on the request of the name node and the client. The consequent action after fulfilling the request is reporting the updated information back to the name node [5] [6].

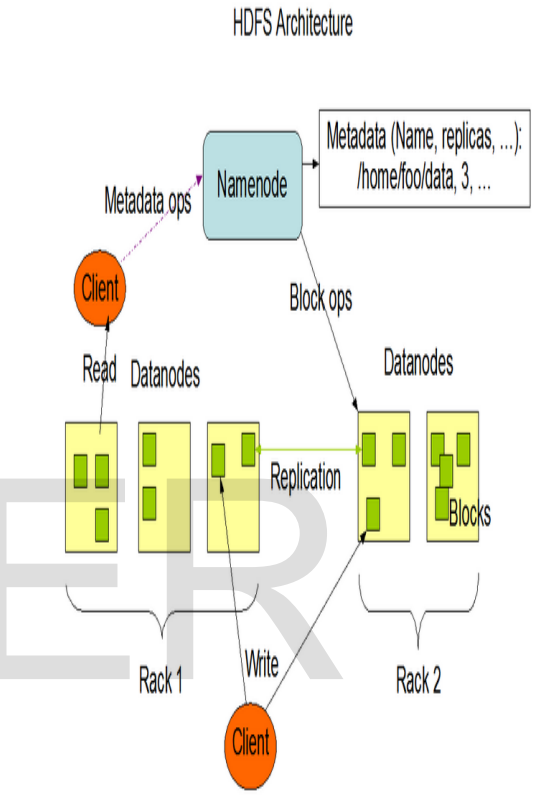

Figure 1: HDFS Architecture

Source: http://hadoop.apache.org/docs/r0.17.2/hdfs_design.html

Name Node lists the block by specific ID number and the data node in which the block is present, this data is commonly known as "Metadata" [9]. Generally the term metadata refers to as data about data. Data present in Name Node is persistent and of high priority, if data loss occurs here the loss is permanent. Hence to prevent this, Secondary Name Node has been introduced which contains the image copy of Name Node known as fs-image file. Secondary Name Node also consists of edit logs which is the log book of transactions in the distributed file system [5] [6] [9].

The transmission is enhanced by "Heart Beats" between the Data Nodes and Name Node. All the Data Nodes triggers the heart beat to Name Node

every three seconds, which consists of block report and the list of blocks present in that Data Node, based on which the Name Node confirms that a particular Data Node is active at this moment. If the Name Node fails to receive heart beat from a Data Node, the data present in that node is transferred into another Data Node based on the Meta Data [5].

B) MapReduce

MapReduce is a parallel processing model introduced by Google for processing distributed data of large volume. A distributed task from task tracker is assigned to multiple nodes, and their data is processed in parallel. The computation of data is followed by two functions namely Map and Reduce [8] [9]. The inputs and outputs to this MapReduce programs are key/value pairs. The key, value pair which is provided as input to the mapper is processed and an intermediate key/value pair is generated. This value is given as input to reducer function to get the processed output [4] [5].

As shown in Figure 2, the MapReduce program starts with the Input Reader phase where the data from HDFS file system is given as input and the output generated is key/value pairs [8] [10]. The input reader divides the data into splits and passed as input to the mapping phase. The next phase is the mapping phase where the input is generated key/value pairs. The map function processes these pairs and generates the intermediate key/value pairs and stores them in local disk [4] [5].

RecordReader reads the input from single input split and divides them into key/value pairs which are directed to map function. Then comes the shuffle phase where the intermediate results of different map functions are combined together [9] [10]. This is the third phase of data flow. The sort phase is next phase where the shuffled data from preceding phase is sorted based on the key, in this phase all the contents having same key value are combined together. The result from this sorting phase is fed as input to the reducer phase [4] [5].

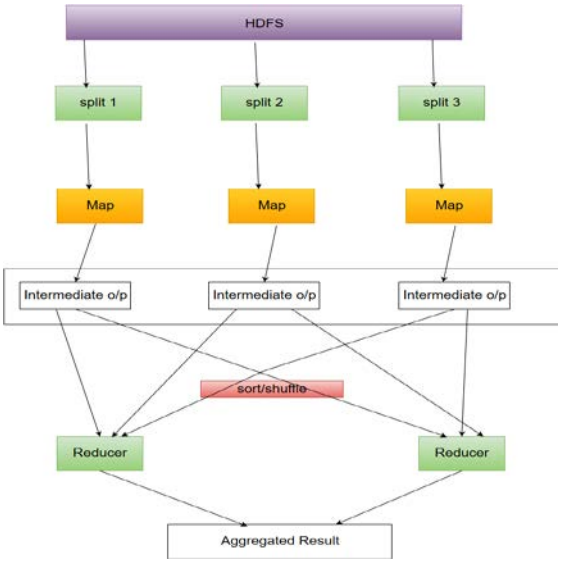

Figure 2: MapReduce Data Flow

The final phase of the MapReduce program is reducer phase. In this phase the reducer function does the aggregation operation also known as summation operation. The reducer then reads the entire buffer using remote procedure calls. The reducer passes each key and according set of values to the user defined reducer function which gives the aggregate result and this result is stored in the HDFS file system [4] [5] [10].

Processing through MapReduce

Input: (Word Count)
Dear Car Bear
Car Car River
River Car Bear
Algorithm:

1. The input data for word count above, stored in the HDFS in the form of blocks is processed using MapReduce.
2. These blocks of data is sent to mapper class which consists of map function, here the data is split by the given delimiter and for the word count for each split 1 is appended so that the count is calculated.
3. Then the output from the map function is set as input to the reducer function which consists of reducer class.
4. The data here undergoes calculation, which means according to the desired use case the code is modified in this reducer class.
5. For the word count program the number of 1 in each split is calculated and then presented as the list of words and the count of the words. The result is as shown below.

Result:

Figure 3: Map Reduce output

4. APACHE PIG

Apache Pig was first evolved by yahoo and its main goal is to perform the data manipulation. Pig is comparatively very easy language and can be easily understood. It is the one of the implementations of Map Reduce. Generally Map Reduce subsists of two tasks, Map and Reduce tasks independently. Map Reduce is used to convert the data from one form into another form of data [4].

Pig can be established with local mode or cluster mode. Local mode starts if we designate option 'xlocal' whereas if we do not acknowledge any option it automatically initiates the cluster mode. Local mode retrieves all data or files from the local system whereas in cluster it gains access from HDFS [4] [11].

Map reduce assignments can be done easily through Pig Latin than using a java or a complicated language [4]. It reduces the number of lines in the code as compared to a java program. For example, a particular java program consists of 50 lines whereas a corresponding program in Pig can be written in 10 lines on rough approximation [11].

The Pig programming language was initially created with two constituents typically Pig Latin and the execution of Pig Latin [11]. Pig Latin is similar to Structured Query Language (SQL) and it is effortless if we are perfect with SQL [4].

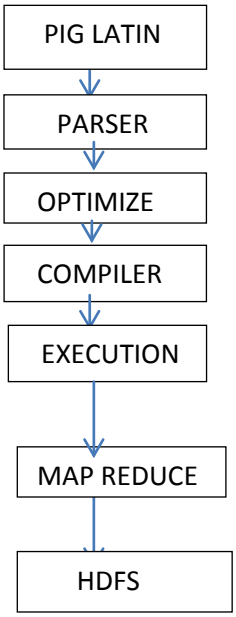

Figure 4: Apache Pig Architecture

As shown in Figure 4, to begin any task in pig it must be first written in scripting language i.e. Pig Latin and it is the first phase in Apache Pig Architecture [11].

The Second phase of model is the Parser. The main purpose of parser is to scrutinize the semantic rules of the code and also to perform type checking. The output of the parser phase is in the form of Direct Acyclic Graph (DAG). It reduces the number of repeated nodes [4] [11].

The next Phase is Optimization. It optimizes the Direct Acyclic Graph (DAG) and then compiles the optimized code in the fourth phase which is the Compiler [11].

The last phase is Execution. In execution engine, Map Reduce operations are performed and postulated to the Hadoop [11].

Pig Latin consists of datatypes, arithmetic operators, relational operators and comparison operators. It reduces the time of development because it combines the minute operations in pipeline and converts the unstructured data to structured data, as it absorbs both formats of data [1] [4] [11].

Processing through Pig

Input:

Custs:

4000001, Kristina, Chung, 55, Pilot
4000002, Paige, Chen, 74, Teacher
4000003, Sherri, Melton, 34, Fire-fighter
4000004, Gretchen, Hill, 66, Computer hardware engineer
4000005, Karen, Puckett, 74, Lawyer
4000006, Patrick, Song, 42, Veterinarian
4000007, Elsie, Hamilton, 43, Pilot
4000008, Hazel, Bender, 63, Carpenter
4000009, Malcolm, Wagner, 39, Artist
4000010, Dolores, McLaughlin, 60, Writer

Txns:

00000000, 06-26-2011, 4000001, 040.33, Exercise & Fitness, Cardio Machine Accessories, Clarksville, Tennessee, credit
00000001,05-26-2011,4000002,198.44,Exercise & Fitness, Weightlifting Gloves, Long Beach, California, credit

Figure 5: Example of input data

Algorithm:

1. Firstly we need to load the input data using the load statement. The 'PigStorage' function does the loading and comma is passed as the data delimiter.
2. As per the required use case, the next line of code can be any of the operations like 'foreach', 'generate', 'filter' etc. and the result produced will be as shown below in Figure 6.

Result:

```
Reno  2
Omaha 1
Plano 1
Boston      1
Denton      1
Newark      1
Orange      1
Anaheim     1
Atlanta     1
Chicago     1
Everett     2
Fremont     1
```

Figure 6: Example of Pig output

6. APACHE HIVE

Jen Hammerbacher who formerly worked with Facebook endowed Apache Hive. It is been overlooked by the Apache Software Foundation after its invention in Facebook. It is a kind of mechanism which handles huge amount of data and is effectively used in Facebook and Amazon [4].

Hive is handed over for querying and evaluating the data and it is an open source system [4]. It handles the structured data and repose on top of the Hadoop to manage the Big Data. It is swift, accustomed and easy to understand [12].

Hive is identical to SQL. It allows us to analyse the data through SQL queries which is known as the HiveQL (Hive Query Language).SQL type queries are used to process data in HDFS [12].

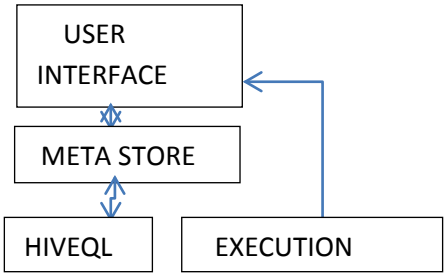

Figure 7: Apache Hive Architecture

In the architecture of Hive, the User Interface (UI) acts as a communication medium between the user and the HDFS as shown in Figure 7. Through these, the user can intercommunicate with the other systems [12]. Hive can brace only few User Interfaces such as Hive web UI, Hive command Line and Hive HD Insight [4].

The Meta Store is used to store all the metadata of tables which is commonly defined as the "data about data". It stores even the metadata of HDFS mapping and its data types [4].

HiveQL is used to acquire data from the Meta Store through Queries. It is used to perform Map Reduce work through queries instead of writing Map Reduce work through a java program. Then the Execution phase will execute the query that has been written in HiveQL and generates the results [12].

Hive categorized the datatypes into four types namely column types, literals, null values and complex types. Column types are used for creating columns. Literals used in hive are Floating literals and decimal literals .Floating literals consist of decimal numbers whereas decimal literals consist of high range numbers. Null value is a value whose value is zero and Complex types consists of arrays, maps and structs [4].

Processing through Hive

Input:

	Athlete	Country	Year	Sport	Gold	Silver	Bronze	Total
2	Yang Yilin	China	2008	Gymnastic	1	0	2	3
3	Leisel Jon	Australia	2000	Swimming	0	2	0	2
4	Go Gi-Hye	South Kor	2002	Short-Trac	1	1	0	2
5	Chen Ruol	China	2008	Diving	2	0	0	2
6	Katie Lede	United Sta	2012	Swimming	1	0	0	1
7	Ruta Meil	Lithuania	2012	Swimming	1	0	0	1
8	DÃ¡niel G	Hungary	2004	Swimming	0	1	0	1
9	Arianna F	Italy	2006	Short-Trac	0	0	1	1
10	Olga Glats	Russia	2004	Rhythmic	1	0	0	1
11	Kharikleia	Greece	2000	Rhythmic	0	0	1	1
12	Kim Marti	Sweden	2002	Ice Hocke	0	0	1	1

Figure 8: Example of input data

Algorithm:

1. First task is to create a table for holding the data by providing the 'create table' query.
2. The next line of code would be loading the data into the above table using the 'load data inpath' query.
3. After loading the data, create a query as per the use case to filter the data and the result will be as shown below in Figure 9.

Result:

Afghanistan 2
Algeria 8
Argentina 139
Armenia 10
Australia 524
Austria 70
Azerbaijan 25

Figure 9: Example of Hive output

7. Conclusion

The escalating usage of internet generates immense data sets that are increasing the value of present day's business and Big Data already acquired a subtle response in a way of doing business. Hadoop is an open source mechanism of managing and transforming big data. This paper precisely defines the architecture of Hadoop and how the data can be processed by implementing the programming models of MapReduce, Pig and Hive within a short span.

8. References

[1] Rabi Prasad Padhy "Big Data Processing with Hadoop-MapReduce in Cloud Systems", International Journal of Cloud Computing and Services Science (IJ-CLOSER), Vol.2, No.1, February 2013, pp. 16~27.
[2] Zan Mo, Yanfei Li "Research of Big Data Based on the Views of Technology and Application", American Journal of Industrial and Business Management, 2015, 5, 192-197.
[3] Kai Ren, YongChul Kwon, Magdalena Balazinska, Bill Howe "Hadoop's Adolescence: A Comparative Workload Analysis from Three Research Clusters", Carnegie Mellon University, CMU-PDL-12-106, June 2012.
[4] Navya Francis, Sheena Kurian K "Data Processing for Big Data Applications using Hadoop Framework", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 3, March 2015.
[5] Harshawardhan S. Bhosale, Prof. Devendra P. Gadekar "A Review Paper on Big Data and Hadoop", International Journal of Scientific and Research Publications, Volume 4, Issue 10, October 2014.
[6] D.Usha and Aslin Jenil A.P.S "A Survey of Big Data Processing in Perspective of Hadoop and Mapreduce", International Journal of Current Engineering and Technology.
[7] Bijesh Dhyani , Anurag Barthwal "Big Data Analytics using Hadoop", International Journal of Computer Applications Volume 108 – No 12, December 2014.
[8] Mrigank Mridul, Akashdeep Khajuria, Snehasish Dutta, Kumar N "Analysis of Bidgata using Apache Hadoop and Map Reduce", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 5, May 2014.
[9]Garry Turkington "Hadoop Beginner's Guide, February 2013", Birmingham: Packt Publishing Ltd, pp.25-50.
[10] Tom White "Hadoop:The Definitive Guide" April 2009, O'Reilly Media, Inc. CA, pp.45-65.
[11] Alan Gates "Programming Pig", October 2011, O'Reilly Media, Inc. CA, pp.33-57.
[12] Edward Capriolo, Dean Wampler, and Jason Rutherglen "Programming Hive", October 2012, O'Reilly Media, Inc. CA, pp.49-71.