ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 5, May 2015

Analysis of Research Data using MapReduce Word Count Algorithm

Manisha Sahane[1], Sanjay Sirsat[2], Razaullah Khan[3]

Department of Management Science, Dr. B.A.M. University, University Campus, Near Soneri Mahal, Jaisingpura, Aurangabad (M.S.), India[1,2]

Department of Commerce & Management Science, Maulana Azad College, Dr. Rafiq Zakaria Marg, Rauza Baug, Aurangabad (M.S.), India[3]

Abstract: We are living in the data world where the data continuously grows exponentially from different data generator factors like Sensors, Healthcare, Telecom, Online Shopping, Social Media, Digital Images and Videos, Retail, Hospitality, Finance, Buy and sell transaction, Airlines, Security Surveillance etc. Big data is about collection of datasets that cannot be handled by the traditional database management system and such data which are beyond to storage capacity and processing power. Hadoop is the recent tool to analyze the volume of data which is rapidly increasing by Giga bytes, Tera Bytes, Peta Bytes, Zeta Bytes and so on. The research objective of paper is to study Hadoop and associated technologies with glance focus on MapReduce and analysis of university research data set to know the focused area of research in Zoology and Botany department.

Key Words: BigData, Hadoop Technology, Hadoop Distributed File System (HDFS), MapReduce.

I. INTRODUCTION

Now a day, data analytics is one of the booming markets. There are different data centers where one can store voluminous data, such as IBM Server, EMC Server etc. and Amazon web services provide a host of services to store and crunch the data at scale in a cost effective manner. Big data doesn't only refer to data sets that are large in size, but also covers data sets that are complex in structures, high dimensional, distributed, and heterogeneous [14]. Unstructured data consist of text messages, images, audios and videos.

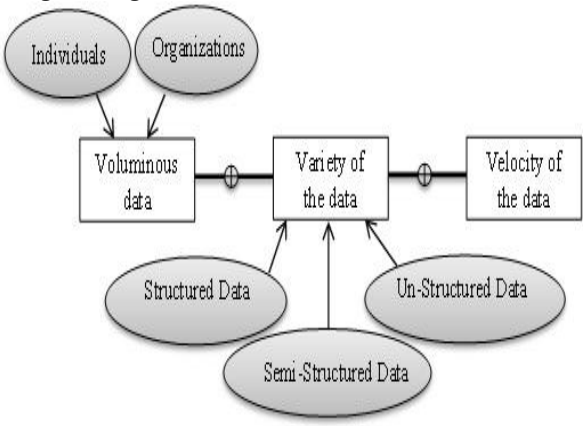


Figure 1 Concept of BigData defined by IBM

Social networks generate unstructured data and application log files are type of semi-structured data whereas RDBMS (Relational Database Management System) stores structured data. Figure 1 show varieties of data are in the form of unstructured, semi-structured and structured data. The big data occupies much attention in some extent for his volume, velocity, and variety [8].The identified challenges are grouped into four main categories corresponding to Big Data tasks types: data storage (relational databases and NoSQL stores), Big Data analytics (machine learning and interactive analytics), online processing, and security and privacy [7].

A. Hadoop

The rate at which unstructured or semi-structured data has increased exponentially. To consume such exponential growth of data, the data ingestion and processing techniques should be sophisticated and handle the fast pace, in that sense Hadoop has been in focused. It is one of the most popular platforms, an open-source software framework for BigData processing used by leading companies like Yahoo and Facebook [11]. Hadoop is Google's Map Reduce and Google File System. The Apache Hadoop is used for parallel computing of large data sets across distributed nodes. Figure 2 shows Hadoop is well suited for Storage and Data Processing. Hadoop can access the data in interactive, batch and real time manner. It provides security with respect to the data. The Hadoop model of distributing the data with the computations presents a paradigm shift for the data-intensive scientific computing community [1]. PIG is one of the components of Hadoop built on top of the HDFS. It is an abstract and high level language on top of MapReduce. Apache pig is used to process the huge amount of data by the means of multiple transformations. Apache pig is used for getting a summarization, query and for advanced query. The process workflow of data that undergoes these multiple transformations and hence we can call Apache pig as "Data Flow Language or Transformation Language". Pig user defined functions can be implemented using Java, and Python to perform transformations. Pig is designed for batch processing of

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 5, May 2015

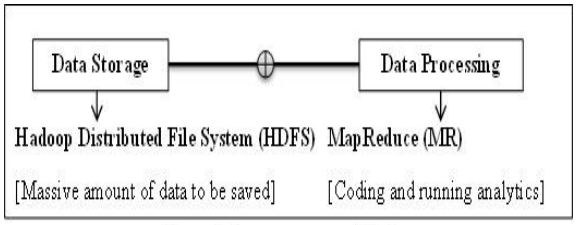


Figure 2 Core component of Hadoop

data. Pig's infrastructure layer consists of a compiler that turns (relatively short) Pig Latin programs into sequences of MapReduce programs [18].

HBASE is one of the components of Hadoop, which is built on HDFS and It is made for real time random reads and random writes (In against to the sequential file of HDFS). HBase is open source, distributed, fault tolerant, scalable, multi-dimensional, version and column oriented database which is built in google big table (built on the origins of the Google File System (GFS)). All the data of HBase is organized by means of tables. When values insert into HBase table, a unique and random time stamp get an auto generated for each value. HBase has its own Java client API, and tables in HBase can be used both as an input source and as an output target for MapReduce jobs through Table Input/TableOutputFormat [18].

HIVE (SQL on HADOOP) is warehouse kind of the system in Hadoop. It is used for data summarization, querying and advanced querying. All the hive data are organized by means of tables only. Hive is designed for batch processing, not online transaction processing – unlike HBase, Hive does not offer real-time queries [18]. Hive does not follow any primary key, foreign key and ACID properties concept. It is one of the components of Hadoop built on top of HDFS.

SQOOP is one of the components of Hadoop built on the top of MapReduce and made for interacting with the RDBMS i.e. to import the data of RDBMS tables into Hadoop HDFS or to export the data from HDFS to any RDBMS table.

OOZIE is one of the components of Hadoop built on top of HDFS and its workflow creation and scheduling component in Hadoop stack. It is one of the open-source Java based application and is Graphical User Interface component. Apache OOZIE is a Direct Acyclic Graph (DAG) component, i.e. execution of current task depends on completion of previous task.

FLUME is one of the components of Hadoop built on the top of HDFS. Flume architecture contains flume master, flume source, flume channels, and flume sink. It is made for live streaming data collection and distribution of the same data over HDFS. Apache flume captures the online data, but it never ever involves in any kind of processing of Hadoop.
MONGO DB is one of the NO SQL databases which store data by default in the form [Key, Value] record architecture. Specifically known as binary JSON i.e. BSON. It is one of the famous document storage databases where each value we insert, get stored as document only.
IMPALA (SQL ON HDFS) is advanced warehouse component over Hive to process the data in a real time manner. Impala avoids MapReduce and access the data directly using a specialized distributed query engine similar to commercial RDBMS. Impala is distributed component and supports Massively Parallel Processing (MPP).

Data Management	HDFS (Storage), YARN (Resource Management)
Data Access	Map Reduce, Pig (Scripting), Hive (Data Access), tez, Hbase, Accumulo (Cell level access control), Storm (Real-time processing), Hcatalog
Data Governance & Integration	Falcon, Flume, SQOOP, WebHDFS, NFS
Security	Knox, Security
Operations – 1. Provision, Manage and Monitor 2. Scheduling	Ambari, Zookeeper OOZIE

Table 1 some associated technologies in Hadoop

MAHOUT is a Big Data analytics component. It basically provides two tasks feed the application and train the program. Mahout supports four main data science use cases such as classification, clustering, collaborative filtering or recommendation, and frequent item set mining. Machine learning and recognition are provided by Mahout and Hama. Mahout also provides a number of distributed clustering algorithms, including k-means, dirichlet, mean-shift, and canopy [19].

Hadoop distributed file system (HDFS) supports a high throughput mechanism after handling large data and high availability of the data file by storing it in a sequential redundant manner over multiple clusters. HDFS, we can import huge amounts of data from heterogeneous sources, providing data cleaning engine with data input and the resulting output [6]. HDFS handles continuous updates (write many) less well than a traditional relational database management system [18].

B. MapReduce

It is one of the core components of Hadoop, which is made for the processing of huge amount of data in parallel fashion on commodity machines. Map makes the traditional inquire task, disassembling task and data analysis task into distributed processing, handling and allocating task to different nodes. The MapReduce algorithm parallelizes the performance of a number of problems [13]. Reduce combines different information coming from the Map, computing the result sets and achieving the reduced answer [6]. New buzz word and Hadoop MapReduce is the best tool available for processing data and its distributed, column-oriented database, HBase which uses HDFS for its underlying storage, and support provides more efficiency to the system [15]. MapReduce is a programming model of Hadoop system. MapReduce model is used to process intensive data within less time. Figure 3 shows [20] each one of the map and reduce phase has key-value pairs as input and output. The shuffle phase shuffles the outputs of map phase to the input of the reduce phase evenly using

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 5, May 2015

the MapReduce library. The map phase runs a user defined mapper function on a set of key-value pairs [kj, vj] taken as input, and generates a set of intermediate key-value pairs.

Each major (reduce) step reduces the number of data objects (key-value pairs) by an order of magnitude or more [10]. "org.apache.hadoop.mapreduce.InputFormat" is responsible for dividing given input data block into multiple input splits. A Hadoop application's execution time is greatly affected by the shuffling phase, where an amount of data is transferred from map tasks to reduce tasks [21]."org.apache. hadoop. mapreduce.recordreader" is responsible for dividing split into (key, value) phase to be accepted by mapper phase. Below are steps that happen as part of the MapReduce life cycle.

Step 1. Inputrequest should in the form of .JAR file which contains Driver code, Mapper code and Reducer code.
Step 2. Job Tracker assigns the mapper tasks by tracking the business logic from the .JAR file on the all the available task trackers.
Step 3. Once all the task trackers are done with mapperprocesses, they send the same status back to Job Tracker.
Step 4. All the task trackers do with mapper phase, then job tracker initiates sort and shuffle phase on all the mapper outputs.

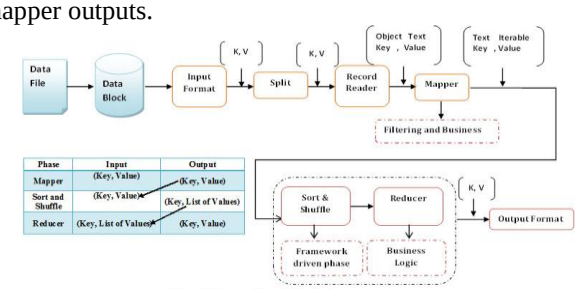


Figure 3 Process flow of MapReduce

Step 5. When the sort and shuffle is done, job tracker initiates reducer phase on all available task trackers.
Step 6. Once all task trackers do with reducer phase, they update the same status back to the job tracker.

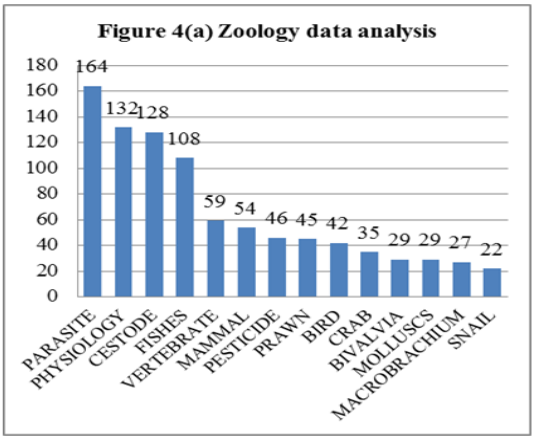
Mapper and Reducer are user driven phases. Mapper class output filename is "part-m-00000" and Reducer class output filename is "part-r-00000". Job Tracker and Task Tracker are the two daemons which are entirely responsible for MapReduce processing.

II. RELATED WORK

MareMareprovides programmers with Map and Reduce algorithmic skeletons, as well as with an optimized composition of the two, i.e. a map+reduce skeleton [2]. [17] developed a set of tools that integrate data from log files and machine metrics to examine job performance. The goal is to provide insight for improving the performance of complex MapReduce algorithms. The MapReduce based algorithms are mentioned with one of their performance on Amazon cloud. The recognized activities can be used in mobile applications like Zompopo that utilizes the information in creating an intelligent calendar [16]. Distributed_Arabic_handwriting OCR system based on a parallel FastDTW algorithm confirms that MapReduce, Hadoop and cascading technologies provide an adequate platform to accelerate the Arabic handwritten recognition process [9]. WordCount, the Data Placement Policy can improve by up to 24.7%, with an average improvement of 14.5% for map tasks to allocate data blocks [12]. Multiple query optimization (MQO) framework, SharedHive, for improving the performance of MapReduce-based data warehouse Hadoop Hive queries [5].In order to increase the overall performance of MapReduce applications, the computing power must be scaled up proportionally along the network and storage system [3].

III. RESEARCH METHODOLOGY

The research activity is largely based on application of MapReduce concept on the university research dataset. Hence the study is demanded an appropriate technique of data analysis with the word count algorithm. Data collection is done mainly from secondary sources. The published data set has been collected from website of Dr. BabasahebAmbedkarMarathwada University [11]. The dataset contains records from 1980 to 2010. A sample size of Zoology dataset is577 and Botany dataset contains 335samples. Research dataset is stored on local file system. Text cleaning has been done by removing corrupted, erroneous, misleading and empty fields. It has been then copied from local file system to HDFS. The WordCount module is developed in the Java programming language and then the .JAR file is uploaded to single node storage. The data are tokenized using the MapReduce algorithm in order to establish the interest area of research.



Afterwards only such tokens or data that are most frequent and relevant to the task has been chosen. Top fourteen keywords are selected from each dataset which are having maximum occurrences.After processing the data, the output is stored on HDFS and copied back to local file system.The WordCount algorithm has been successfully applied to the datasets.

IV. EXPERIMENTALRESULTS

MapReduce implementation for WordCountalgorithm computation.The outcome is the list of words with the

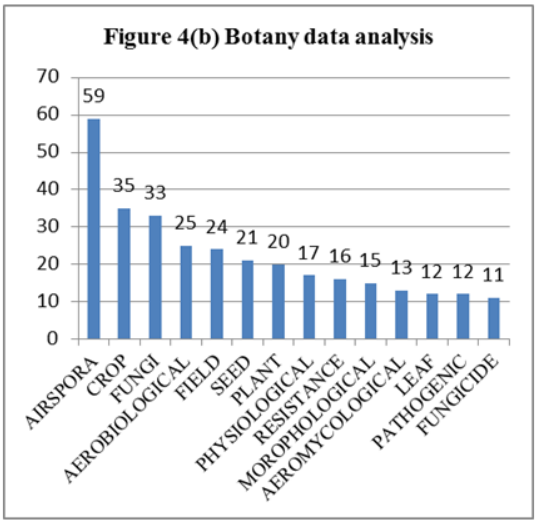
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 5, May 2015

count of appearance of each word.Figure 4 illustrates the trend of researchdata on the native Apache Hadoop. The research trend indicates,the most of the research studies are done in particular domains. Figure 4(a) shows that in Zoology domain, the majority of focused research areas are parasites, Physiology, Cestode, and Fishes. Figure 4(b) proves that in the Botany domain, mainly research concentration is on Airospora, Crop, Fungi and Aerobiological. Results are possible using a MapReduce WordCount algorithm.

V. CONCLUSIONS

Core parts of Hadoop technology are HDFS and MapReduce where HDFS provides voluminous, variety of data storage and MapReduce provides fast data processing.The goal of paper is to summarize the potential use of MapReduce in the processing of research data.MapReduce typically supports batch-based high scale parallelization and reliable processing.The Hadoop MapReduce framework utilizes a distributed file system to read and write its data. Therefore, the I/O performance of a Hadoop MapReduce job strongly depends on HDFS [4]. In this study authors are analyzed most researched area and least researched area of corresponding domains. Outcomes of study can be useful for researchers for their study.



REFERENCES

[1]. Addair, T., Dodge, D., Walter, W., and Ruppert, S. (2014). Large-scale seismic signal analysis with Hadoop. Computers & Geosciences, 66, 145–154

[2]. Buono, D.,Danelutto, M., andLametti, S. (2012). Map, Reduce and MapReduce, the skeleton way. In the proceedings of International Conference on Computational Science, ICCS 2010,Procedia Computer Science 1, 2095–2103

[3]. Castane, G., Nunez, A.,Filgueira, R.,and Carretero, J. (2012). Dimensioning Scientific Computing systems to improve performance of Map-Reduce based applications. In the proceedings of International Conference on Computational Science, ICCS 2012, Procedia Computer Science 9, 226 – 235

[4]. Dittrich, J., and Quiane-Ruiz, J-A. (2012). Efficient Big Data Processing in Hadoop MapReduce. In the Proceedings of the VLDB Endowment, 5(12)

[5]. Dokeroglu, T., Ozal, S., Bayir, M., Cinar, M., and Cosar, A. (2014). Improving the performance of Hadoop Hive by sharing scan and computation tasks. Journal of Cloud Computing: Advances, Systems and Applications, 3:12, doi:10.1186/s13677-014-0012-6

[6]. Du, D., Li, A., Zhang, L., and Li, H. (2015). Review on the Applications and the Handling Techniques of Big Data in Chinese Realty Enterprises. Ann. Data. Sci., DOI 10.1007/s40745-014-0025-5

[7]. Grolinger, K., Hayes, M., Higashino, W., L'Heureux, A., and Allison, D. (2014). Challenges for MapReduce in Big Data. In the proceedings of IEEE 10th 2014 World Congress on Services (SERVICES 2014), Alaska, USA.http://ir.lib.uwo.ca/electricalpub/44

[8]. Gu, J., and Zhang, L. (2015). Some Comments on Big Data and Data Science. Ann. Data. Sci., DOI 10.1007/s40745-014-0021-9

[9]. Hassen, H. and Khemakhem, M. (2013). Large Distributed Arabic Handwriting Recognition System Based on the Combination of FastDTW Algorithm Map Reduce Programming Model Via Cloud Computing Technologies, In Proceeding of AASRI, Procedia 5, 156 – 163

[10]. Highland, F., and Stephenson, J. (2012). Fitting the Problem to the Paradigm: Algorithm Characteristics Required for Effective Use of MapReduce. In the proceedings of Conference Organized by Missouri University of Science and Technology 2012- Washington D.C., Procedia Computer Science 12, 212 – 217

[11]. https://www.bamu.net/library/theses%20database.html

[12]. Jeong, Y-S., and Kim, Y-T. (2015). A token-based authentication security scheme for Hadoop distributed file system using elliptic curve cryptography. J ComputVirol Hack Tech, DOI 10.1007/s11416-014-0236-5

[13]. Lee, C-W., Hsieh, K-Y., Hsieha, S –Y., and Hsiao, H-C. (2014). A Dynamic Data Placement Strategy for Hadoop in Heterogeneous Environments. Big Data Research 1, 14–22, http://dx.doi.org/10.1016/j.bdr.2014.07.002

[14]. Lewis, S., Csordas, A., Killcoyne, S., Hermjakob, H., Hoopmann, M., Moritz, R., Deutsch, E., and Boyle, J. (2012). Hydra: a scalable proteomic search engine which utilizes the Hadoop distributed computing framework. BMC Bioinformatics, 13:324, doi:10.1186/1471-2105-13-324

[15]. Li, F., and Nath, S. (2014). Scalable data summarization on big data. Distrib Parallel Databases 32, 313–314, DOI 10.1007/s10619-014-7145-y

[16]. Mridul, M., Khajuria, A., Dutta, S., Kumar, N., and Prasad, R. (2014). Analysis of Bidgata using Apache Hadoop and Map Reduce. International Journal of Advanced Research in Computer Science and Software Engineering, 4(5), 555-560

[17]. Paniagua, C., Flores, H., and Srirama, S. (2012). Mobile Sensor Data Classification for Human Activity Recognition using MapReduce on Cloud. In the proceedings ofThe 9th International Conference on Mobile Web Information Systems (MobiWIS 2012), Procedia Computer Science 10, 585 – 592

[18]. Plantenga, T., Choe, Y., and Yoshimura, A. (2012). Using Performance Measurements to Improve MapReduce Algorithms. Procedia Computer Science 9, 1920 – 1929

[19]. Taylor, R. (2010). An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. In the proceedings of 11th Annual Bioinformatics Open Source Conference (BOSC) Boston, MA, USA,BMC Bioinformatics 2010, 11(Suppl 12):S1, doi:10.1186/1471-2105-11-S12-S1

[20]. Venner, J. (2009). Pro Hadoop. Apress publisher, ISBN13: 978-1-4302-1942-2

[21]. Wang, Y., Goh, W., Wong, L., and Montana, G. (2013). Random forests on Hadoop for genome-wide association studies of multivariate neuroimaging phenotypes. In the proceedings of Asia Pacific Bioinformatics Network (APBioNet) Twelfth International Conference on Bioinformatics (InCoB2013) Taicang, China,BMC Bioinformatics, 14(Suppl 16):S6, doi:10.1186/1471-2105-14-S16-S6

[22]. Xie, J., Tian, Y., Yin, S., Zhang, J., Ruan, X., and Qin, X. (2013). Adaptive Preshuffling in Hadoop clusters. Procedia Computer Science 18, 2458 – 2467