ISSN 2320-9984
Volume 5 Issue 2 | June 2017 ISSN: 2320-9984 (Online)
International Journal of Modern Engineering & Management Research
Website: www.ijmemr.org

Massive Data Processing in Map Reduce Surroundings

Manoj Dwivedi
M. Tech. Research Scholar
Takshshila Institute of Engineering & Technology
Jabalpur (M.P.), [INDIA]
Email: manojdwivedi590@gmail.com

Deepak Agrawal
Head of the Department
Department of Computer Science and Engineering
Takshshila Institute of Engineering & Technology
Jabalpur (M.P.), [INDIA]
Email:deepakagrawal@takshshila.org

Abstract**—***Big data is a term that refers to data sets or combinations of data sets whose size (volume), complexity (variability), and rate of growth (velocity) make them difficult to be captured, managed, processed or analyzed by conventional technologies and tools, such as relational databases and desktop statistics or visualization packages, within the time necessary to make them useful. While the size used to determine whether a particular data set is considered big data is not firmly defined and continues to change over time, most analysts and practitioners currently refer to data sets from 30-50 terabytes(10 12 or 1000 gigabytes per terabyte) to multiple petabytes (1015 or 1000 terabytes per petabyte) as big data. To address the above mentioned issues, the Hadoop framework is designed to provide a reliable, shared storage and analysis infrastructure to the user community. The storage portion of the Hadoop framework is provided by a distributed file system solution such as HDFS, while the analysis functionality is presented by MapReduce. Big Data processing is performed through a programming paradigm known as MapReduce. Typically, implementation of the MapReduce paradigm requires networked attached storage and parallel processing. The computing needs of MapReduce programming are often beyond what small and medium sized business are able to commit.*

Keywords:— *Big Data, Hadoop, HDFS, Map/Reduce*

1. INTRODUCTION

Big data means really a big data, it is a collection of large datasets that cannot be processed using traditional computing techniques. Big data is not merely a data, rather it has become a complete subject, which involves various tools, techniques and frameworks. The complex nature of big data is primarily driven by the unstructured nature of much of the data that is generated by modern technologies, such as that from web logs, radio frequency Id (RFID), sensors embedded in devices, machinery, vehicles, Internet searches, social networks such as Facebook, portable computers, smart phones and other cell phones, GPS devices, and call center records. In most cases, in order to effectively utilize big data, it must be combined with structured data (typically from a relational database) from a more conventional business application, such as Enterprise Resource Planning (ERP) or Customer Relationship Management (CRM).

Before Internet and Web did not exist, we did not have enough data so that it was not easy to analyze people, society, and science etc with the limited volumes of data. Contradicting to the past, after Internet and web, it has been more difficult to analyze data because of its huge volumes, that is, tera-bytes or peta-bytes of data, which is called Big Data. Google faced to the issue when collecting Big Data as the

existing file systems were not sufficient to store Big Data efficiently. Besides, the legacy computing power and platforms were not enough to compute Big Data. Thus, Google implemented Google File Systems (GFS) and Map/Reduce parallel computing platform, which Apache Hadoop project is motivated from.

Hadoop is the parallel programming platform built on Hadoop Distributed File Systems (HDFS) for Map/Reduce computation that processes data as (key, value) pairs. Hadoop has been receiving highlights for the enterprise computing because business world always has the big data such as log files for web transactions. Hadoop is useful to process such big data for business intelligence so that it has been used in data mining for past few years. The era of Hadoop means that the legacy algorithms for sequential computing need to be redesigned or converted to Map/ Reduce algorithms. Therefore, in this paper, a Market Basket Analysis algorithm in data mining with Map/Reduce is proposed with its experimental result in Elastic Compute Cloud (EC2) and (Simple Storage Service) S3 of Amazon Web Service (AWS).

2. BIG DATA

We are awash in a flood of data today. In a broad range of application areas, data is being collected at unprecedented scale. Decisions that previously were based on guesswork, or on painstakingly constructed models of reality, can now be made based on the data itself. Such Big Data analysis now drives nearly every aspect of our modern society, including mobile services, retail, manufacturing, financial services, life sciences, and physical sciences. This definition is intentionally subjective and incorporates a moving definition of how big a dataset needs to be in order to be considered big data—i.e., we don't define big data in terms of being larger than a certain number of terabytes (thousands of gigabytes). We assume that, as technology advances over time, the size of datasets that qualify as big data will also increase. Also note that the definition can vary by sector, depending on what kinds of software tools are commonly available and what sizes of datasets are common in a particular industry.

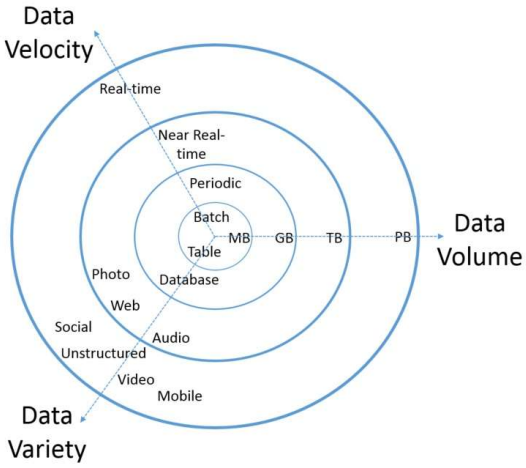

Figure 1. 3V's of Big Data

3. HADOOP

The Hadoop framework is designed to provide a reliable, shared storage and analysis infrastructure to the user community. The storage portion of the Hadoop framework is provided by a distributed file system solution such as HDFS, while the analysis functionality is presented by MapReduce. Several other components (discussed later in this report) are part of the overall Hadoop solution suite. The MapReduce functionality is designed as a tool for deep data analysis and the transformation of very large data sets. Hadoop enables the users to explore/analyze complex data sets by utilizing customized analysis scripts/ commands. In other words, via the customized MapReduce routines, unstructured data sets can be distributed, analyzed, and explored across thousands of shared-nothing processing systems/clusters/nodes. Hadoop's HDFS replicates the data onto multiple nodes to safeguard the environment from any potential data-loss.

Hadoop Data Distribution

In a Hadoop cluster environment (commodity HW is normally used to setup the cluster), the data is distributed among all the nodes during the data load phase. The HDFS splits large data files into chunks that are

managed by different nodes in the cluster. Each chunk is replicated across several nodes to address single node outage or fencing scenarios. An active monitoring system re-replicates the data during node failure events. Despite the fact that the file chunks are replicated and distributed across several nodes, Hadoop operates in a single namespace and hence, the cluster content is collectively accessible. In the Hadoop programming framework, data is conceptually record-oriented. The individual input files are carved up into lines or other (application logic) specific formats. Hence, each thread executing on a cluster node processes a subset of the records. The Hadoop framework schedules these threads in proximity to the location of the data/records by utilizing knowledge obtained from the distributed file system. Which data chunk is operated on by a node is chosen based on the data chunks locality to a node. The design goal is that most data is read from a local disk straight into the CPU subsystem to economize on the number of network transfers necessary to complete the processing cycle. The design strategy of moving the actual computation to the data (instead of moving the data to the computation) allows Hadoop to achieve high data locality reference values that result in increased performance scenarios.

4. MAP/REDUCE IN HADOOP

Map/Reduce is an algorithm used in Artificial Intelligence as functional programming. It has been received the highlight since re-introduced by Google to solve the problems to analyze huge volumes of data set in distributed computing environment. It is composed of two functions to specify, "Map" and "Reduce". They are both defined to process data structured in (key, value) pairs.

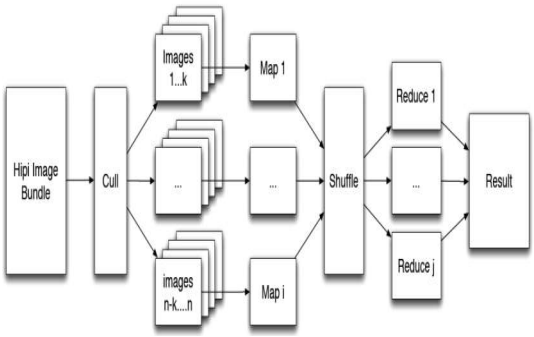

Figure 2. MapReduce Process

A. Map/Reduce in Parallel Computing

Map/Reduce programming platform is implemented in the Apache Hadoop project that develops open-source software for reliable, scalable, and distributed computing. Hadoop can compose hundreds of nodes that process and compute peta- or tera-bytes of data working together. Hadoop was inspired by Google's MapReduce and GFS as Google has had needs to process huge data set for information retrieval and analysis [1]. It is used by a global community of contributors such as Yahoo, Facebook, and Twitters. Hadoop's subprojects include Hadoop Common, HDFS, MapReduce, Avro, Chukwa, HBase, Hive, Mahout, Pig, and ZooKeeper etc. [2].

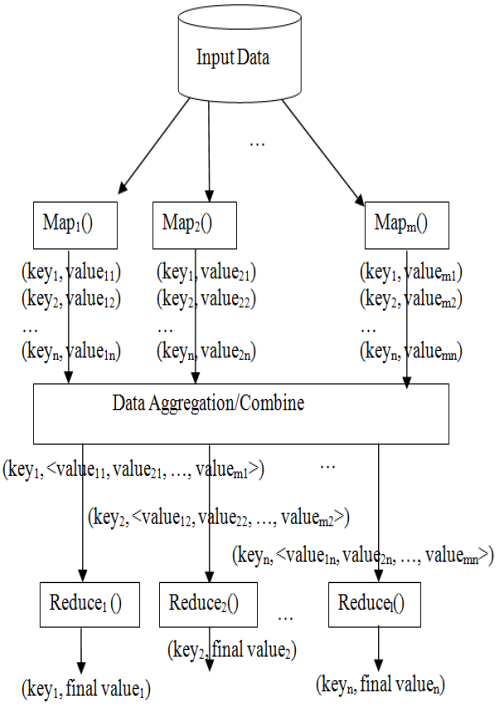

Figure 3. Map/Reduce Work Flows

The map and reduce functions run on distributed nodes in parallel. Each map operation can be processed independently on each node and all the operations can be performed in parallel. But in practice, it is limited by the data source and/or the number of CPUs near that data. The reduce functions are in the similar situation because they are from all the output of the map operations. However, Map/Reduce can handle significantly huge data sets since data are distributed on HDFS and operations move close to data for better performance [5]. Hadoop is restricted or partial parallel programming platform because it needs to collect data of (key, value) pairs as input and parallel computes and generates the list of (key, value) as output on map/reduce functions. In map function, the master node parts the input into smaller sub-problems, and distributes those to worker nodes. Those worker nodes process smaller problems, and pass the answers back to their master node. That is, map function takes inputs (k1, v1) and generates <k2, v2> where < > represents list or set. Between map and reduce, there is a combiner that resides on map node, which takes inputs (k2, <v2>) and generates <k2, v2>. In Figure 3.2, a list of values is collected for a key as (keyn, <value1n, value2n, …, valuemn>) from mappers.

In reduce function, the master node takes the answers to all the sub-problems and combines them in some way to get the output, the answer to the problem [1, 2]. That is, reduce function takes inputs (k2, <v2>) and generates <k3, v3>. Figure 3.2 illustrates Map/ Reduce control flow where each valuemn is simply 1 and gets accumulated for the occurrence of items together in the proposed Market Basket Analysis Algorithm. Thus, for each key, the final value is the total number of values, that is the sum of 1s, as (keyn, final valuen)

B. The Issues of Map/Reduce

Although there are advantages of Map/ Reduce, for some researchers and educators, it is:

1. Need tens-, hundreds-, or thousands-of-nodes to compose Hadoop Map/Reduce platform.
2. If using services of cloud computing, for example, AWS EC2, the overheads mainly come from I/O. That is, it takes long to upload big data to AWS EC2 platform or AWS S3, which is more than computing time.
3. A giant step backward in the programming paradigm for large-scale data intensive applications
4. Not new at all - it represents a specific implementation of well known techniques developed tens of years ago, especially in Artificial Intelligence
5. Data should be converted to the format of (key, value) pair for Map/Reduce, which misses most of the features that are routinely included in current DBMS
6. Incompatible with all of the tools or algorithms that have been built [4].

However, the issues clearly show us not only the problems but also the opportunity where we can implement algorithms with Map/ Reduce approach, especially for big data set. It will give us the chance to develop new systems and evolve IT in parallel computing environment. It started a few years ago and many IT departments of companies have been moving to Map/Reduce approach in the states.

5. CONCLUSION

Hadoop with Map/Reduce motivates the needs to propose new algorithms for the existing applications that have had algorithms for sequential computation. Besides, it is (key, value) based restricted parallel computing so that the legacy parallel algorithms need to be redesigned with Map/Reduce.

In the paper, the Market Basket Analysis Algorithm on Map/Reduce is presented, which is association based data mining analysis to find the most frequently occurred pair of products in baskets at a store. The data set of the experimental result shows that associated items can be paired in orders 2 and 3 with Map/Reduce approach. Once we have the associated items, it can be used for more studies by statically analyzing them even sequentially, which is beyond this paper.

REFERENCES:

[1] J. Dean and S. Ghemawa, "MapReduce: Simplified Data Processing on Large Clusters", Google Labs, OSDI 2004, **(**2004**)**, pp. 137–150.

[2] Apache Hadoop Project, http:// hadoop.apache.org/.

[3] B. Stephens, "Building a business on an open source distributed computing", Oreilly Open Source Convention (OSCON) 2009, (2009**)** July 20-24, San Jose, CA

[4] W. Kim, "MapReduce Debates and Schema-Free", Coord, **(**2010**)** March 3.

[5] J. Lin and C. Dyer, "Data-Intensive Text Processing with MapReduce", Tutorial at the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010), **(**2010**)** June, Los Angeles, California

[6] J. Woo, "Introduction to Cloud Computing", the 10th KOCSEA 2009 Symposium, UNLV, (2009) December 18-19.

[7] J. Woo, "The Technical Demand of Cloud Computing", Korean Technical Report of KISTI (Korea Institute of Science and Technical Information), **(**2011**)** February.

[8] J. Woo, "Market Basket Analysis Example in Hadoop", http://dal-cloudcomputing.blogspot.com/2011/03/ market-basket-analysis-example-in.html, **(**2011**)** March.

[9] Aster Data, "SQL MapReduce framework", http://www.asterdata.com/product/advanced-analytics.php.

[10] Apache HBase, http:// hbase.apache.org/.

[11] J. Lin and C. Dyer, "Data-Intensive Text Processing with MapReduce", Morgan & Claypool Publishers, **(**2010**)**.

[12] GNU Coord, http://www.coordguru.com/.

[13] J. Woo, D. -Y. Kim, W. Cho and M. Jang, "Integrated Information Systems Architecture in e-Business", The 2007 international Conference on e-Learning, e-Business, Enterprise Information Systems, e-Government, and Outsourcing, Las Vegas, **(**2007**)** June 26-29.