© 2016 IJSRSET | Volume 2 | Issue 5 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Improvement in Performance of Hadoop using Hace Process and Word Count Result with Bigdata

Prof. Vivek Badhe, Shweta Verma

Department of Computer Science & Engineering, Gyan Ganga College of Technology Jabalpur, Madhya Pradesh, India

ABSTRACT

Figuring innovation has changed the way we work, concentrate on, and live. The appropriated information preparing innovation is one of the mainstream themes in the IT field. It gives a straightforward and concentrated registering stage by lessening the expense of the equipment. The attributes of circulated information preparing innovation have changed the entire business. Hadoop, as the open source undertaking of Apache establishment, is the most illustrative stage of circulated enormous information handling. The Hadoop conveyed structure has given a protected and quick huge information preparing engineering. The clients can outline the appropriated applications without knowing the points of interest in the base layer of the framework. This proposal gives a brief prologue to Hadoop. Because of the multifaceted nature of Hadoop stage, this proposal just focuses on the center advancements of the Hadoop, which are the HDFS, MapReduce, and HACE.

Keywords: Hadoop, Big Data, HDFS, MapReduce, HACE, Data Processing

I. INTRODUCTION

The IT industry is always developing new technologies, and big data is the one of them. With the developments of the cloud storage, big data has attracted more and more attention. Due to the emergence of the Internet, the big data technology will accelerate the innovation of the enterprises, lead the revolution of the business mode and create unlimited commercial opportunities.

A. Current situation of the big data

In recent years, we have been drowning in the ocean of data that was produced by the development of the Internet, the Mobile Internet, the Internet of Things (loT) and the Social Networks. A photo that uploaded to Instagram is about 1MB; a video that uploaded to YouTube is about dozens of Mega sizes. Chatting online, browsing websites, playing online games, and shopping online will also turn into data that may be stored in any corner in the world. Hence, how much data is there in our daily life? According to a report of IBM, there are 2.5 quintillion bytes of data that we create every day. Ninety percent of that data was created in the recent two years. That means, in one day, the data that appears on the Internet can fill 168 million DVDs; the 294 billion emails we sent equals to the numbers of printed newspaper in United States for recent two years. [4] By 2012, the volume of data has increased from Terabyte level to Petabyte level. The reducing price of computer hardware and the production of supercomputers make it possible to deal with large and complex data. All the data can be divided into four types: structured data (e.g., stock trading data), semi-structured data (e.g., blogs), unstructured data (e.g., text, audio, video), and multi-structured data.

B. The definition of Big Data

Finding a way to define the Big Data is not easy, but authors hold the same view with Ohlhosrt [12] that Big Data is the large and complex data that is difficult to use the traditional tools to store, manage, and analyze in an acceptable duration. Therefore, the Big Data needs a new processing model which has the better storage, decision-making, and analyzing

abilities. This is the reason why the Big Data technology was born. The Big Data technology provides a new way to extract, interact, integrate, and analyze of Big Data. The Big Data strategy is aiming at mining the significant valuable data information behind the Big Data by specialized processing. In other words, if comparing the Big Data to an industry, the key of the industry is to create the data value by increasing the processing capacity of the data[11].

C. The characteristics of Big Data

According to a research report from Gartner (Doug, 2001), the growth of the data is three-dimensional, which is volume, velocity and variety. So far, there are many industries still use the 3Vs model to describe the Big Data. However, the Big Data is not only 3Vs but also has some other characteristics [10]. The first one is the volume. As mentioned, the volume of Big Data has moved from Terabyte level to Petabyte level. The second one is the variety. Compared with the traditional easy to storage structured text data, there is now an increasing amount unstructured data that contains web logs, audio, video, pictures, and locations. Data no longer needs to be stored as traditional tables in databases or data warehouses but also stored as variable data types at the same time. To meet this requirement, it is urgent to improve the data storage abilities. Next is velocity. Velocity is the most significant feature to distinguish the Big Data and the traditional data mining. In the Age of Big Data, the volume of high concurrency access of users and submittion data are huge.

II. METHODS AND MATERIAL

A. Related Work

Distributed Data Processing (DDP) is not only a technical concept but also a logical structure. The concept of DDP is based on the principle that can achieve both centralized and decentralized information service[6].

Capability components of DDP Platform

DDP platforms have different capability components to help it to complete the whole process. Different capability components are responsible for different jobs and aspects. The following sections will introduce the most important capability components of a DDP platform.

a) File Storage

The file storage capability component is the basic unit of data management in the data processing architecture. It aims to provide a fast and reliable access ability to meet the needs of large amount of data computing.

b) Data Storage

The data storage capability component is an advanced unit of data management in the data processing architecture. It aims to store the data according to an organized data model and to provide an independent ability of deleting and modifying data. IBM DB2 is a good example of a data storage capability component.

c) Data Integration

The data integration capability component integrates the different data which has different sources, formats, and characters into units to support the data
input and output between multiple data sources and databases. Oracle Data Integrator is an example of data integration component.

d) Data Computing

The data computing capability component is the core component of the whole platform. It aims to solve the specific problem by using the computing resources of the processing platform. Taking MPI (Message Passing Interface) which is commonly used in parallel computing as an example, it is a typical data computing component. In the Big Data environment, the core problem is how to split the task that needs huge computing ability to calculate into a number of small tasks and assign them into specified computing resources to processing.

e) Data analysis

The data analysis capability component is the closest component to the users in the data processing platform. It aims to provide an easy way to support the user to extract the data related to their purpose from the complex information. For instance, as a data analysis component, SQL (Structured Query Language)

provides a good analysis method for the relational databases. Data analysis aims at blocking the complex technical details in the bottom layer of the processing platform for the users by abstract data access and analysis. Through the coordinates of data analysis components, the users can do the analysis by using the friendly interfaces rather than concentrate on data storage format, data streaming and file storage.

f) Platform Management

The platform management capability component is the managing component of the data processing. It aims to guarantee the safety and stability for the data processing. In the Big Data processing platform, it may consist of a large amount of servers that may be distributed in different locations. On this occasion, how to manage these servers' work efficiently to ensure the entire system running is a tremendous challenge.

III. RESULTS AND DISCUSSION

A. Proposed Work and Result

The MapReduce operation architecture includes the following three basic components [11]:

- ✓ Client: Every job in the Client will be packaged into a JAR file which is stored in HDFS and the client submits the path to the Job Tracker.
- ✓ Job Tracker: The Job Tracker is a master service which is responsible for coordinating all the jobs that are executed on the MapReduce. When the software is on, the Job Tracker is starting to receive the jobs and monitor them. The functions of MapReduce include designing the job execution plan, assigning the jobs to the Task Tracker, monitoring the tasks, and redistributing the failed tasks.
- ✓ Task Tracker: The Task Tracker is a slave service which runs on the multiple nodes. It is in charge of executing the jobs which are assigned by the Job Tracker. The Task Tracker receives the tasks through actively communicating with the Job Tracker.

B. MapReduce Procedure

The MapReduce procedure is complex and smart. This thesis will not discuss the MapReduce procedure in detail but will introduce it briefly based on author's own thoughts.

Usually, MapReduce and HDFS are running in the same group of nodes. This means that the computing nodes and storage nodes are working together.

This kind of design allows the framework to schedule the tasks quickly so that the whole cluster will be used efficiently. In brief, the process of MapReduce can be divided into the following six steps [8]:

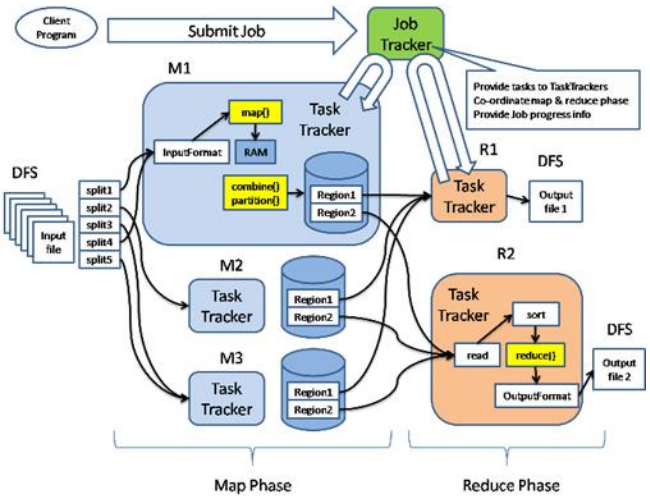

Figure 4. MapReduce Procedure

(1) Job Submission

When the user writes a program to create a new Job Client, the Job Client will send the request to Job Tracker to obtain a new Jobbed. Then, the Job Client will check if the input and output directories are correct. After this check, Job Client will store the related resources which contain the configuration files, the number of the input data fragmentations, and Mapper/Reducer JAR files to HDFS. In particular, the JAR files will be stored as multiple backups. After all the preparations have been completed, the Job Client will submit a job request to the Job Tracker.

(2) Job Initialization

As the master node of the system, Job Tracker will receive several Job Client requests so that Job Tracker implements a queue mechanism to deal with these

problems. All the requests will be in a queue that is managed by the job scheduler. When the Job Tracker starts to initialize, its job is to create a JobInProgress instance to represent the job. The Job Tracker needs to retrieve the input data from HDFS and to decide on the number of the Map tasks. The Reduce tasks and the TaskInProgress are determined by the parameters in the configuration files.

(3) Task Allocation

The task allocation mechanism in the MapReduce is to pull the whole process. Before the task allocation, the Task Tracker which is responsible for Map tasks and Reduce tasks has been already launched. The Task Tracker will send the heartbeat message to the Job Tracker to ask if there are any tasks that can be done any time. When the Job Tracker job queue is not empty, the Task Tracker will receive the tasks to do. Due to the lack of the Task Tracker computing capability, the tasks that can be done on the Task Tracker are also limited. Each Task Tracker has two fixed task slots which correspond to the Map tasks and Reduce tasks. During the tasks allocation, the Job Tracker will use the Map task slot first. Once the Map task slot is empty, it will be assigned to the next Map task. After the Map task slot is full, then the Reduce task slot revives the tasks to do.

(4) Map Tasks Execution

After the Map Task Tracker has received the Map tasks, there is a series of operations to finish the tasks. Firstly, the Map Task Tracker will create a Task In Progress object to schedule and monitor the tasks. Secondly, the Map Task Tracker will take out and copy the JAR files and the related parameter configuration files from HDFS to the local working directory. Finally, when all the preparations have been completed, the Task Tracker will create a new Task Runner to run the Map task. The Task Runner will launch a separate JVM and will start the Map Task inside to execute the map () function in case the abnormal Map Task affects the normal Task Tracker works. During the process,

The Map Task will communicate with Task Tracker to report the task progress until all the tasks are completed. At that time, all the computing results will be stored in the local disk.

(5) Reduce Tasks Execution

When the part of the Map Tasks completed, the Job Tracker will follow a similar mechanism to allocate the tasks to the Reduce Task Tracker. Similar to the process of Map tasks, the Reduce Task Tracker will also execute the reduce () function in the separate JVM. At the same time, the Reduce Task will download the results data files from the Map Task Tracker. Until now, the real Reduce process has not started yet. Only when all the Map tasks have been completed, the Job Tracker will inform the Reduce Task Tracker to start to work. Similarly, the Reduce Task will communicate with the Task Tracker about the progress until the tasks are finished.

(6) Job Completion

In the each Reduce execution stage, every Reduce Task will send the result to the temporary files in HDFS. When all the Reduce Tasks are competed, all these temporary files will be combined into a final output file. After the Job Tracker has received the completion message, it will set the state to show that jobs done. After that, the Job Client will receive the completion message, then notify the user and display the necessary information.

(7) Limitations of MapReduce

Although MapReduce is popular all over the world, most people still have realized the limits of the MapReduce. There are following the four main limitations of the MapReduce [10]:

- ✓ The bottleneck of Job Tracker

From the previous chapters, the Job Tracker should be responsible for jobs allocation, management, and scheduling. In addition, it should also communicate with all the nodes to know the processing status. It is obvious that the Job Tracker which is unique in the MapReduce, takes too many tasks. If the number of clusters and the submission jobs increase rapidly, it will cause network bandwidth consumption. As a result, the Job Tracker will reach bottleneck and this is the core risk of MapReduce.

- ✓ The Task Tracker

Because the jobs allocation information is too simple, the TaskTracker might assign a few tasks that need more sources or need a long execution time to the same node. In this situation, it will cause node failure or slow down the processing speed.

- ✓ Jobs Delay

Before the MapReduce starts to work, the TaskTracker will report its own resources and operation situation. According to the report, the JobTracker will assign the jobs and then the TaskTracker starts to run. As a consequence, the communication delay may make the JobTracker to wait too long so that the jobs cannot be completed in time.

- ✓ Inflexible Framework

Although the MapReduce currently allows the users to define its own functions for different processing stages, the MapReduce framework still limits the programming model and the resources allocation.

IV. CONCLUSION

This paper has introduced the core technology of Hadoop but there are still many applications and projects developed on Hadoop. In conclusion, the Hadoop, which is based on the Hadoop HDFS and MapReduce, has provided a distributed data processing platform. The high fault tolerance and high scalability allow its users to apply Hadoop on cheap hardware. The MapReduce distributed programming mode allows the users to develop their own applications without the users having to know the bottom layer of the MapReduce. Because of the advantages of Hadoop, the users can easily manage the computer resources and build their own distributed data processing platform.

Above all, it is obvious to notice the convenience that the Hadoop has brought in Big Data processing. It also should be pointed out that since Google published the first paper on the distributed file system till now, the history of Hadoop is only 10-year old. With the advancement of the computer science and the Internet technology, Hadoop has rapidly solved key problems and been widely used in real life. In spite of this, there are still some problems in facing the rapid changes and the ever increasing demand of analysis. To solve these problems, Internet companies, such as Google also introduced the newer technologies. It is predictable that with the key problems being solved, Big Data processing based on Hadoop will have a wider application prospect.

V. REFERENCES

[1] Apache Hadoop Org. (2013). HDFS architecture guide, available at http://hadoop.apache.org/ docs/r1.2.1/hdfs_design.html#Introduction Accessed: 15 May 2015].

[2] Baldeschsieler, E. (2010), How Yahoo Spawned Hadoop, the Future of Big Data. Available at http://www.wired.com/2011/10/how-yahoo-spawned-hadoop/ Accessed: 21 May 2015].

[3] Bloor, B. (2003). The failure of relational database, the rise of object technology and the need for the hybrid database. Arlington: Baroudi Bloor International Inc.

[4] Borthakur, D. (2007). The hadoop distributed file system: Architecture and design. Hadoop Project Research. Apache Orgnization.

[5] Borthakur, D. (2008). HDFS architecture guide. Available at: http://hadoop.apache.org/common/docs/current/hdfs design.pdf. Accessed: 14 May 2015].

[6] Boulon, J., Konwinski, A., Qi, R., Rabkin, A., Yang, E., and Yang, M. (2008). Chukwa, a large-scale monitoring system. Company Report. Yahoo!, inc.

[7] Dean, J. and Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), 113.

[8] Enslow, P. (1978). 'What is a "Distributed" Data Processing System?' Computer, 11(1), pp.13-21.

[9] Gang, L. (2014). Applications and development of Hadoop. Beijing: Zhangtu Information Technology, Inc.

[10] Geczy, P. (2014). Big Data Characteristics. Bachelor. National Institute of Advanced Industrial Science and Technology (AIST), Japan.

[11] George, L. (2011). HBase: the definitive guide. Sebastopol, CA: O'Reilly. Google Developers, (2015). Google Cloud Computing, Hosting Services & Cloud Support. OnlineAvailable at: https://cloud.google.com/ Accessed: 13 May 2015].

[12] Hunt, P., Konar, M., Junqueira, F. P., and Reed, B. (2010). ZooKeeper: Wait-free Coordination for Internet-scale Systems. In USENIX Annual Technical Conference (Vol. 8, p. 9).

[13] He, B., Fang, W., Luo, Q., Govindaraju, N. K., & Wang, T. (2008, October). Mars: a MapReduce framework on graphics processors. In Proceedings of the 17th international conference on Parallel architectures and compilation techniques (pp. 260-269). ACM.