Hadoop MapReduce Framework: Performance Analysis in Business Intelligence

Blessy Trencia Lincy. S. S.
M. Tech, CSE Department,
SRM University

Archana. G.K.
M. Tech, CSE Department,
SRM University

ABSTRACT
Organizations generate large amount of data each day which involves storage of data, processing of data and retrieval of data for other purpose and usage. But an individual organization or an enterprise finds this difficult i.e. they cannot handle this data and thus the useful data remains useless for a longer duration resulting in waste of storage area. Thus I propose an approach for processing and handling the data from various sources in an efficient manner. This approach can be used for Business Intelligence where the data can be processed and can provide ideas about the popularity, cost and feedback about the product released by the enterprise. The Hadoop technology is used for this purpose.

Keywords
Big-Data, Hadoop, MapReduce, Hadoop Distributed Filesystem, Business Intelligence.

1. INTRODUCTION
The organization or an enterprise obtains data from various sources and this will be processed by the MapReduce function of the Hadoop technology. The extraction of the useful information from the incoming large amount of data is a complex task. The MapReduce framework consists of many patterns where the input file will be split and they can be processed through the patterns provided by the MapReduce and they can be sorted accordingly and the reduce phase will merge the processed data and the output can be delivered specifying the required result[1].

For a product being developed in an enterprise the data from other sources can be used to determine the factors like similar products release, the success rate, sales details, cost structure, customer satisfaction and other quality issues[4]. Thus the large amount of data needs to be processed for the organization to develop a product. Even to improve the quality of the product review has to be done and the same product can be produced rectifying the defects.

The SQL queries can be used for processing, retrieval and other operations. But this cannot be done for a huge volume of data[6]. Thus the MapReduce framework can be used and the processing can be done.

In this approach once the data is obtained and processed by the Hadoop MapReduce framework the results can be stored in the HDFS so that these results can be further used by the enterprise for improving the performance. i.e. a product is developed and released by an enterprise and data about the products release, area of sales, maximum sale, minimum sale, cost analysis, defects, customer satisfaction, expectations, etc. can becollected and processed by the Hadoop MapReduce framework and the results can be used for the Business Intelligence(i.e. report generation) and they can be stored in the HDFS so that the up-to-date reports can be stored[5]. The enterprise can use the recent report for later usage and for improving the business.

This approach can be used efficiently by the organizations, enterprises or any other projects, since they data can be made clear and easy for use. In certain cases duplicate computations can be avoided. i.e. analysis can be made for a product and the results can be stored in the HDFS. Thus before performing a different process the file system can be reviewed for the data or report being available already.

2. HADOOP AND MAPREDUCE
Big data can be processed and handled by using the Hadoop and MapReduce. The Hadoop cluster can scale from single nodes in which all the Hadoop entities operate on the same node to thousands of nodes. The functionality can be distributed across the nodes to increase parallel processing activities[1]. When a client makes a request of a Hadoop cluster this request is managed by the job tracker. The job tracker is the namenode that distributes the work among the data nodes. The namenode is the master of the file system providing meta-data services for data distribution and replication.

The job tracker schedules map and reduce tasks into available slots at one or more task trackers. The task tracker is the data nodes referred to as the slave nodes[3]. They execute map and reduce tasks on the data. When the map and reduce tasks are complete the task tracker notifies the job tracker, which identifies when all tasks are complete and eventually notifies client of job completion.

Hadoop MapReduce is a software that provides a framework for applications which can process huge volume of data in parallel on large clusters of commodity machines easily[2]. This can be accomplished in a reliable, fault tolerant manner. A MapReduce job usually splits the input data into independent chunks which are processed by the map tasks in a parallel manner. The MapReduce framework does the sorting of the outputs of the map tasks which are input to reduce tasks. The inputs and outputs will be stored in a local filesystem[1]. This framework schedules the tasks, monitor them and re-executes them on failure.

The architecture of the MapReduce framework represented in Figure 1 can explain the overview of execution of the processing[1]. The MapReduce library function split the input file into some number of pieces. Then many copies of the program will be made on the cluster of machines. Of these copies one will be special that is the master node. The remaining nodes will be the slave nodes. The master node assigns and idle worker nodes with one map task or one reduce task.

The worker nodes creates intermediate results as a key/value pairs and they will be buffered in a memory and given as input to the reduce function[1]. After the processing the location of these buffered pairs will be notified to the master node. When the reducer worker reads the intermediate data it sorts the keys so that all the occurrences of the same key are grouped together. The reduce workers produces the results

and they will be given as the final output of this reduce function. When all map and reduce tasks are completed the master wakes up the user program. After the completion of the processing the output can be in a single file or N number of files and they can be made as input for other process.

2.1 MapReduce Processing Architecture

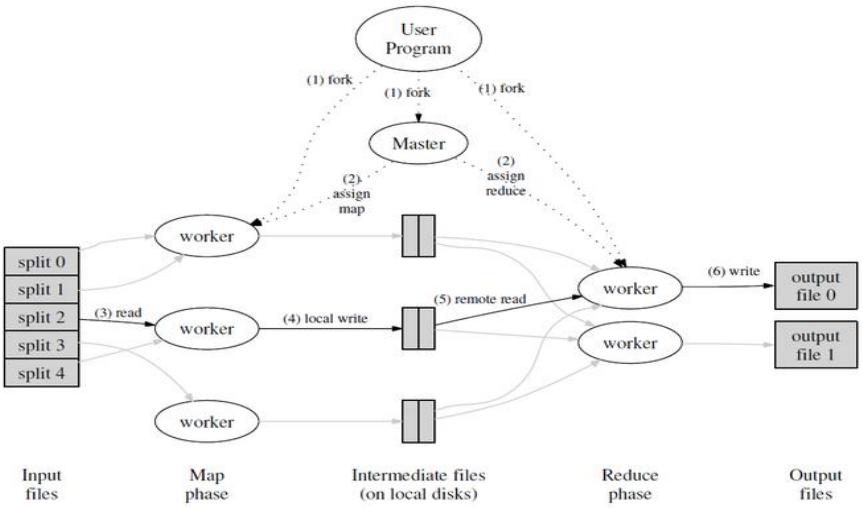


Fig 1: MapReduce job processing

3. PROPOSED SYSTEM

Data from various sources can be obtained and collected to an organization file system i.e. data will be imported into the HDFS. From here the useful information can be obtained by processing the large volume of data by the Hadoop MapReduce and the results can be obtained. From these results the organization can generate report and they can be used by the organization for further improvement of the product being developed or can be used as a feedback for already developed product i.e.to meet the quality issues like customer satisfaction, cost analysis, sales analysis, defects in the product quality, product success , etc[4].

If the same data is given as input file to the system the framework can refer the file system whether this data has been already processed if it is then the data can be directly made available from the file system storage of the enterprise. Thus the entire file need not be processed again. The file system can be referenced before processing of the incoming files. This saves the computation time and the overall performance of the operation done.

Since, the results are being stored in this approach duplicate computations can be avoided and the enterprise need not process huge volume of data for the analysis of already developed product. The recent information will be made available and can be retrieved when needed. The Figure 2 represents the basic overview of the processing of this approach. The report generation is the major task of an organization and thus this approach make it simple and efficient task.

3.1 Architecture

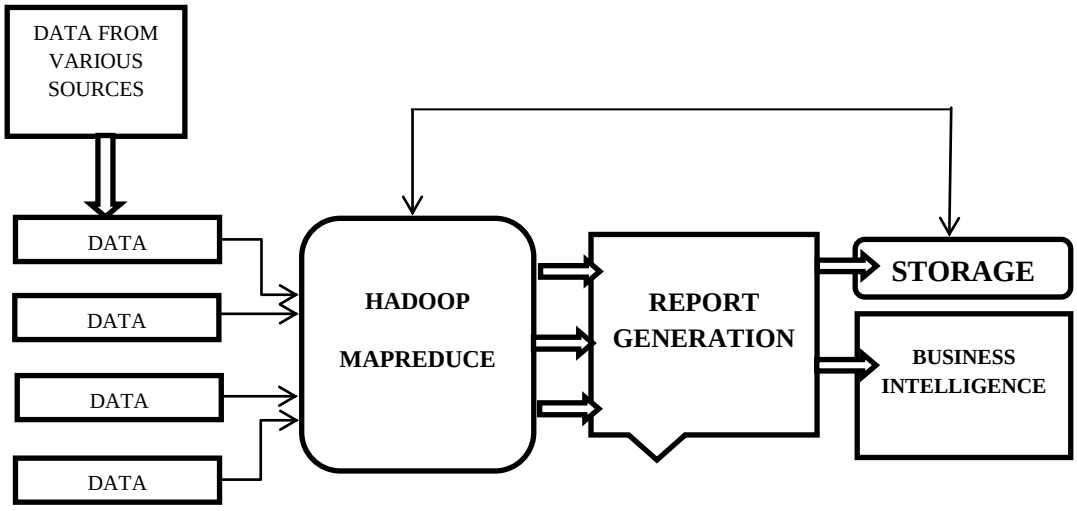
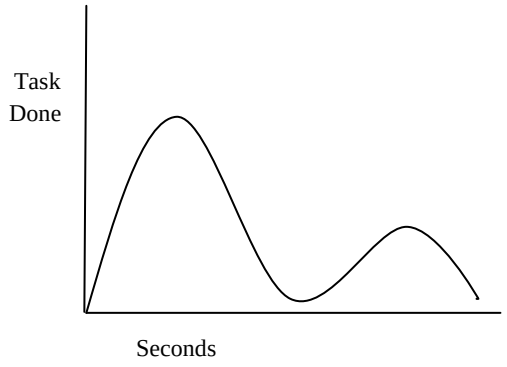


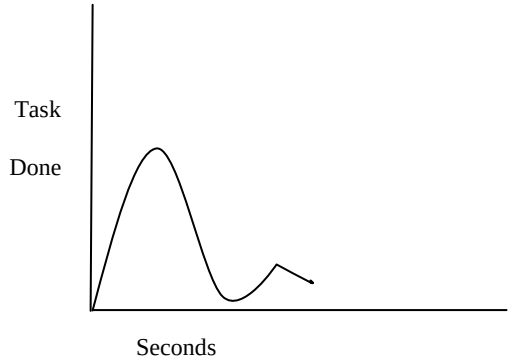
Fig 2 Execution overview

4. RELATED WORK

The jobs can be carried out using the MapReduce framework and thus the performance can be compared by using a simple graph.



a) Normal MapReduce Job Computation time



b) Using the proposed approach

Fig 3 Comparison of performance

The performance of the normal MapReduce processing and the proposed approach is compared and shown in these two graphs shown in Fig 3. The time taken for the MapReduce job can be greatly reduce compared to the processing of the job using the proposed approach.

5. EXPERIMENTAL SETUP

Hadoop is running on a distributed mode with a CPU core of 2 GHz and 4GB memory. This approach is implemented by using a simple word count application. The performance of the operation can be determined from the results of the operation. Hadoop 2.2.0 version is installed and configuration is made accordingly setting up a node in the cluster on the network.

6. CONCLUSION

This approach can improve the overall performance of an operation byproviding therequired results appropriately and also improves the speed of the processing since the results can be stored and referred for later use thus avoiding duplicate computation, saving time of execution and avoid wastage of storage. The enterprise does not need to generate report from the scratch since the up-to-date data can be made available. The future work may include the processing of the MapReduce jobs by performing the computation that improves the overall performance in terms of latency, bandwidth and the storage issues.

7. REFERENCES

[1] "MapReduce: Simplified Data Processing on Large Clusters" Jeffrey Dean and Sanjay Ghemawat.

[2] Hadoop.http://Hadoop.apache.org/.

[3] "A Performance Analysis of MapReduce Task with Large Number of Files Dataset in Big Data Using Hadoop" Amrit pal, Kunal Jain, oinkiAgrawal, Sanjay Agrawal.

[4] "The BTWorld Use Case for Big Data Analytics: Description, MapReduce Logical Workflow, and Empirical Evaluation", Tim Hegeman, BogdanGhit,, MihaiCapot˘a, Jan Hidders, Dick Epema, and AlexandruIosup Parallel and Distributed Systems Group, Delft University of Technology, the Netherlands T.M.{B.I.Ghit, M.Capota, A.J.H.Hidders, D.H.J.Epema, A.Iosup}@tudelft.nl.

[5] Big Data for Business Managers - Bridging the gapbetween Potential and Value, AnmolRajpurohit, Department of Computer Science, The LNM Institute of Information Technology, Jaipur, India, anmol@lnmiit.ac.in.

[6] "Reducing the Search Space for Big Data Miningfor Interesting Patterns from Uncertain Data" Carson Kai-Sang Leung, Richard Kyle MacKinnon, Fan Jiang, Department of Computer Science, University of Manitoba, Winnipeg, MB, Canada, kleung@cs.umanitoba.ca.

[7] "Store, Schedule and Switch – A New Data Delivery Modelin the Big Data Era" Weiqiang Sun, Fengqin Li, Wei Guo, Yaohui Jin and Weisheng, Hu, State Key Laboratory of Advanced Optical Communications Systems and NetworksShanghai Jiao Tong University, Shanghai, 200240, China, sunwq@sjtu.edu.cn.