International Journal of Engineering & Technology, 7 (2.26) (2018) 80-84

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Applying compression algorithms on hadoop cluster implementing through apache tez and hadoop mapreduce

Dr. E. Laxmi Lydia [1] *, M. SrinivasaRao [2]

[1] *Associate Professor, Department of CSE, Vignan's Institute of Information and Technology, Andhra Pradesh, India*
[2] *Associate Professor, Department of CSE, HolyMary Institute of Technology & science, Telangana, India*
**Corresponding author E-mail:*

Abstract

The latest and famous subject all over the cloud research area is Big Data; its main appearances are volume, velocity and variety. The characteristics are difficult to manage through traditional software and their various available methodologies. To manage the data which is occurring from various domains of big data are handled through Hadoop, which is open framework software which is mainly developed to provide solutions. Handling of big data analytics is done through Hadoop Map Reduce framework and it is the key engine of hadoop cluster and it is extensively used in these days. It uses batch processing system.
Apache developed an engine named "Tez", which supports interactive query system and it won't writes any temporary data into the Hadoop Distributed File System(HDFS).The paper mainly focuses on performance juxtaposition of MapReduce and TeZ, performance of these two engines are examined through the compression of input files and map output files. To compare two engines we used Bzip compression algorithm for the input files and snappy for the map out files. Word Count and Terasort gauge are used on our experiments. For the Word Count gauge, the results shown that Tez engine has better execution time than Hadoop MapReduce engine for the both compressed and non-compressed data. It has reduced the execution time nearly 39% comparing to the execution time of the Hadoop MapReduce engine. Correspondingly for the terasort gauge, the Tez engine has higher execution time than Hadoop MapReduce engine.

Keywords: *Data; Mapreduce; Compression; Tez; Hadoop.*

1. Introduction

Hadoop is the more performance technologies used for the big data on present researches. It can be implemented on single or multi cluster. For example big data in a real world, social information's are used to predict and match personal's life style between people in Face book. For the large data sets business intelligent is the term used between social medial information and business information which are generally complex to analyze and predict the dynamic trend of customers need with their products. Hadoop is the solution for all big data problems. Its eco-components includes HBase, Hive R connectors, Mahout, Pig and OOZIE. They work on Hadoop distributed File system and MapReduce. HDFS is a logical disk over physical directories in each data node of Hadoop cluster. It communicates with TCP protocol port 22 such as secure shell on every node in the cluster. HDFS disk is a high fault-tolerant with various replicas in HDFS configuration, only YARN can access and process it. The replication number is a direct effect to HDFS storage. Cloudera and Hortonworks are the two major organizations which are providing a hadoop platform. They perform a data compression algorithm with hadoop, which can decrease disk storage and bandwidth network between each node on the cluster. A hadoop compression suite is including DEFLATE,GZIP,BZIP,LZ4 and Snappy. Yarn has two frameworks which can process data using the map reduce and Tez framework. MapReduce framework supports batch processing and it is a default framework of the hadoop cluster. on the other side, Tez supports interactive processing. It is complex to install and configure with the binary files from the Apache Tex website [1].Rupinder Singh's research [2] invested Tez framework with the Pig scripts and researchers purpose this framework which is better for pre structure data than MapReduce.
The paper mainly focuses on the performance evaluation of compression methods which are available for the hadoop cluster. We study to evaluate a comparison with those of frameworks map reduce and Tez. Tez can reduce the process of data stored in HDFS, it is significance for the research hypothesis. The results of this paper shows the execution time from Hadoop's benchmark and the best methods to implement compression in Hadoopcluster with the big data. We propose an alternative method for improving execution time which is the performance indicator of the Hadoopcluster.

2. Related work

Hadoop is the most popular issue in the high performance computing and big data now a days. In Rupindersingh's research, he focused on the performance comparison of Hadoop's framework MapReduce and Tex with pig scripts which are pre structure data. The results of his research shows that Tez has better performance than MapReduce. In the Hadoop compression algorithms, Andre Wenas [2] used compression as follow: GZIP, LZJB and ZLE for data warehouse and his results shows the best performance on ZLE. Yanpei Chen's research[4] tried to select compress or not compress map reduce output file for reducing power consumption. His results shows that it decreases energy consumption more than 50%.Bhavin

J. Mathiya [5] use more compression algorithm as follows: DEFLATE, LZ4, BZIP and GZIP with word-count benchmark both map output and reduce output. He presented Bzip which has higher compression ratio i.e 85% than other compressions. This paper demonstrates the performance on Tez framework with word-count benchmark on the hadoop cluster. The paper also focuses on the useable of snappy compression which has better performance than LZO and GZIP in the output file of map and reduce.

2.1. Hadoop

The popular tool used in big data is Hadoop, it is portable software which is widely used on commercial websites such as Google and yahoo. Its portable over the Java JDK and creates file system as HDFS which support with a various operating system. MapReduce, Tez and spark are the framework of the hadoop. Figure 1 demonstrates the hadoop eco-components.

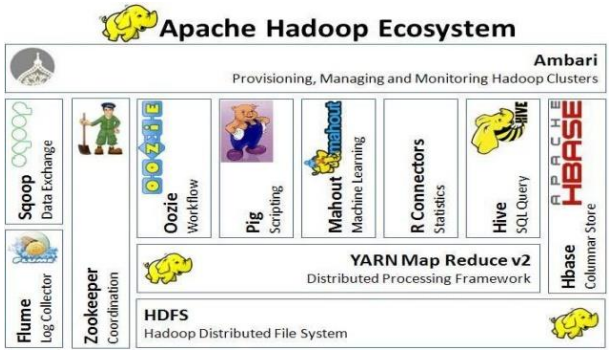

Fig. 1: Hadoop Ecosystem [6].

2.2. Hadoop coreeco-components

The Hadoop Ecosystem comprises of four core components

1) Hadoop Common

Apache foundation has pre-defined set of utilities and libraries which can used by various modules within the Hadoop ecosystem. For instance, if Hbase and Hive needed to access HDFS then they need t make use of Java achieves (JAR files) which are stored in Hadoop common.

2) Hadoop Distributed File System

The big data storage layer for the Apache Hadoop is HDFS, it can be used by the users to dump large datasets and later the user can use for analysis purpose. An HDFS component generates various replicas of the data block which is to be distributed across various clusters for reliable and quick data access. HDFS consists of three main components namely namenode,datanode and secondary namenode. HDFS functions Master Slave architecture where the namenode act as the master node in order to keep track of the storage cluster and datanode act as slave node within the hadoop cluster. Figure2 demonstrate the hadoop master/slave architecture.

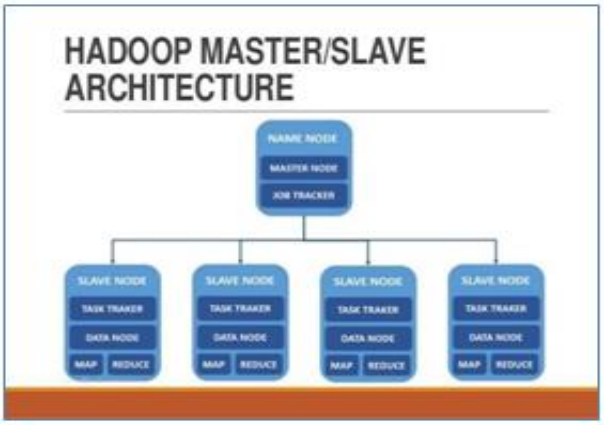

Fig. 2: Hadoop Master/Slave Architecture.

2.3. Mapreduce

MapReduce is a java based system which is created by Google where the actual data from the HDFS storage processed efficiently. MapReduce breaks down a big data processing job into smaller tasks. MapReduce is responsible for the analysing large datasets in parallel before reducing it to find the results. In the Hadoop ecosystem, Hadoop MapReduce is a framework based on YARN architecture. Yarn based on Hadoop architectures supports parallel processing of large data sets and MapReduce provides the framework for easily writing applications on thousands of nodes, considering fault and failure management.

The basic principle of operation behind MapReduce is that the "Map" job sends a query for processing to various nodes in a Hadoop cluster ad the "Reduce" job collects all the results to output into a single value. Map task in the hadoop ecosystem takes input data and splits into independent chunks and output of this task will be the input for Reduce task.

MapReduce framework forms the compute node while the HDFS file system forms the data node. Typically in the Hadoop ecosystem architecture both data node and compute node are considered to be the same.

The delegation task of the MapReduce component are tackled by two daemons-job tracker and task tracker as show in the figure3.

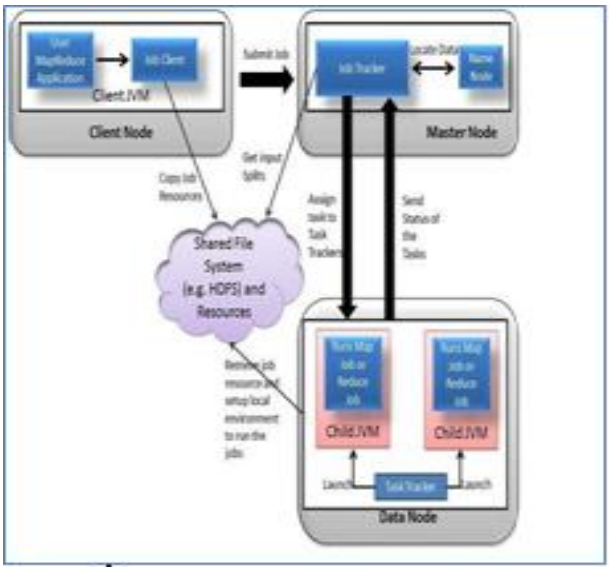

Fig. 3: Task Delegation of MapReduce.

Map Reduce is the distributed processing framework in the hadoop, its working is on batch processing mechanism. The processing is divided into two tasks which are map and reduce. Map task is followed by sort and shuffle task and reduce task is performed at the process. It distributes the data parallel over thousands of machines. MapReduce processing model always needs map phase first than reduce phase and stores the data in the temporary HDFS disk after every map and reduce phase, which is demonstrated in figure 4. Storing of data in the HDFS leads to low in the performance with respect to the map reduce mechanism.

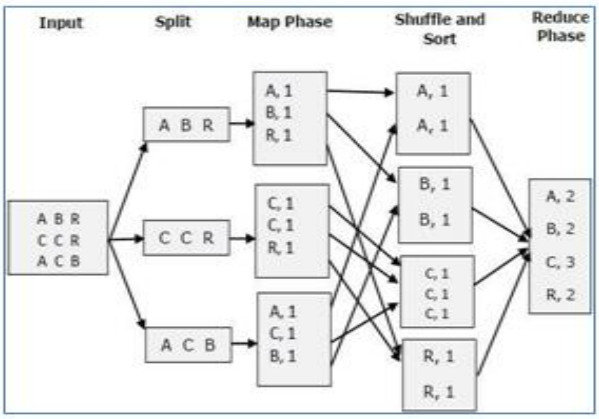

Fig. 4: Map And Reduce Task in A Processing Model [2].

2.4. Tez

Tez is next generation distributed processing framework, as like as map reduce in the hadoop framework. MapReduce is not suitable for interactive query, whereas Tez can be used for interactive query. It is an alternative for query processing, figure 5 demonstrates the difference between "TeZ" processing model and MapReduce Programming model which has only a single map phase and has multiple temporary storing data in HDFS during process that it is significant with respect to the performance.

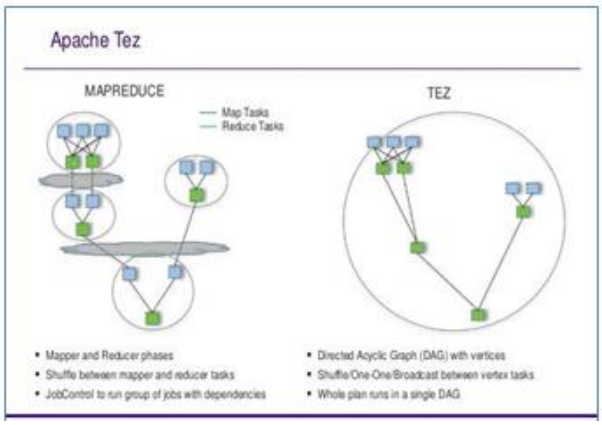

Fig. 5: Apache Tez.

2.5. Compression

In Linux operating system, it has native compression algorithm such as gzip, deflate and bzip2.gzip and deflate are the same general compression using the zlib format. LZ777 is an original concept of snappy. It uses Google oriented technology and designed for the Hadoop Ecosystem, on the other side bzip2 is only compression codec which can split file format.

This is implemented on using native compression of a Hadoop tool which are as follows:

1) bzip2: bzip2 is a high quality data compression algorithm and based on libbzip2, which is open source. It uses a similar PPM family technique. It offers performance at the double time of compression and six times at the decompression which is faster than older version 1.0.6. It is useful for disk drives, distribution CDs, backup taps and USB sticks. It can reduce download times in network. In recovery mode, it can restore the compression data and decompress those parts of the file which are not damaged. It can use libbzip2 to directly read and write bz2 files with compressed data inmemory.
2) Snappy: The older name of zippy is Snappy, it is developed using c++ by the google based on the LZ77 concept in the year 2011. its aim is to increase the speed of the compression which the most high. The benchmarks for snappy is used by core i7 with only a single core which 64-bits mode that results have compression ratio at 20-100% lower than gzip. In hadoop cluster, snappy is the most popular native compression codec that it has widely used in the cassandra, Hadoop, LevelDB, MongoDB, RocksDB,Lucene.

3. Methodology

3.1. Experimental setup

To perform the experiment, we established Hadoop cluster with 5 virtual machines on full virtualization of KVM hypervisor on 3 computers. Each computer is i7-2600 Quad core, 3.40 GHz which supports Intel-VT, 4 GB DDR3, 1.0 TB SATA disks which is shown in figure 6.The established Hadoop cluster architecture is described as follows:

- Master node is created on one big virtual machine in Host01 with 4 vCPU and 4 GB Ram.
- Data Node has 2 virtual machines, each physical host with 4vCPUS and 2GBRAM

In the physical host machine, Centos with KVM virtualization software is installed. The image of virtual machines stored on an extended XFS partition in the home directory. All of the virtual machine installed centos, JDK, Hadoop and Tez.

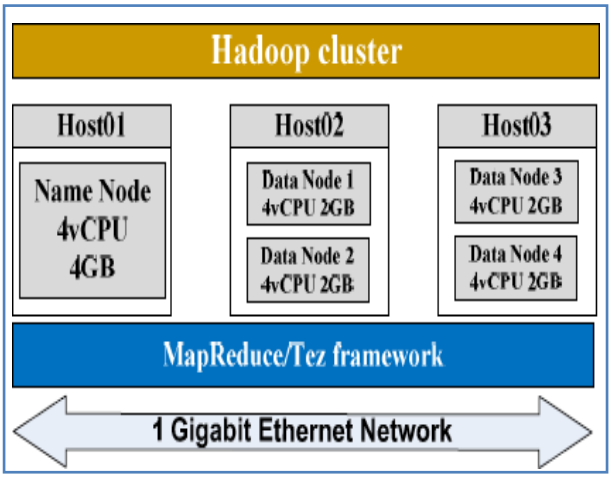

Fig. 6: Hadoop Cluster Architecture.

Map and Reduce has to use compression algorithm with the output files in map and reduce process. it must configure in mapred-site.xml or yarn-site.xml. Figure 7 demonstrating the data compression processing in Hadoop MapReduce [8].

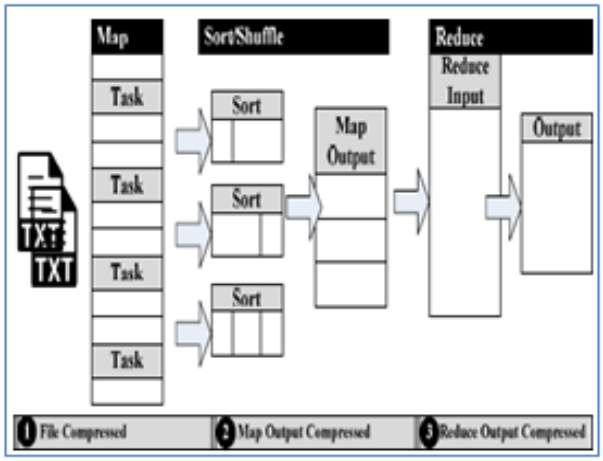

Fig. 7: Data Compression Processing in Hadoop Mapreduce [8].

3.2. Benchmark tools

We conducted experiments to test and compare the execution time between map reduce and Tez. We have executed on two benchmarks to measure various solution of a dataset which is demonstrated in figure 8, details of the benchmark as follows:

- Word count benchmark with workload of e-book text from "project gutenberg"[9] in website is 14.4GB.

Terasort benchmark with workload of text from hadoop teragen benchmark is 10GB.

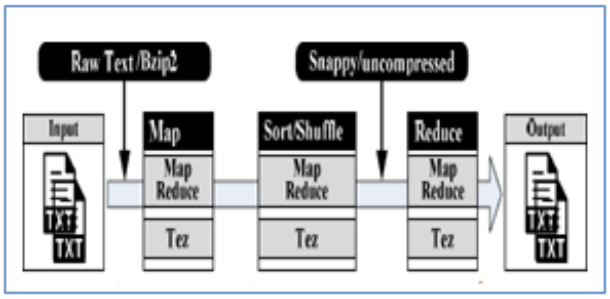

Fig. 8: Test Bed Configuration of Research.

4. Result analysis

We conducted experiments to investigate the computing performance of compression using the compressed methods map output and input file on two benchmark of hadoop. in out experiment, we set an environment of hadoop cluster in full virtualization of KVM to a similar as cloud computing. All the results showed parallel computing performance in term of execution time both framework MapReduce and Tez. In the word count benchmark, the experiment uses word count with ebook text of 14.4GB in four ways; first using raw text without compression map output, second using Bzip2 compressed text without compression map output, third using raw text with snappy compression for map output, last using Bzip2 compressed text with snappy compression for map output. In terasort benchmark, the experiment uses only raw text of 10GB, created by teragen in two ways, first using sort without compression for map output, second using sort with snappy compression for map output. Benchmark MapReduce and Tex evaluate the average execution time of both the benchmarks which are demonstrated in the figure 9 and 10.

Figure6 demonstrating that snappy compression with Tex framework has best execution time at 565.75 seconds as lower than others in a raw text.

4.1. Word count benchmark

The average execution time of word count benchmark with the ebook files of size 14.4 GB and Bzip2 compression file size 3.75GB on hadoop cluster is shown in the figure 6, it compares between MapReduce and Tex frameworks. Figure6 demonstrating that snappy compression with Tex framework has best execution time at 565.75 seconds as lower than others in a raw text.

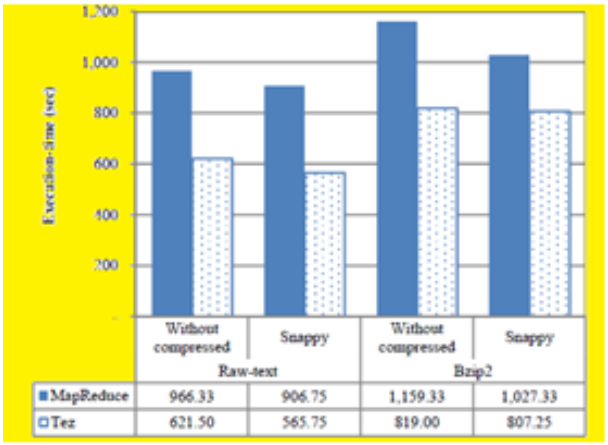

Fig. 9: Execution Time Comparison between Mapreduce and Text in Word Count Benchmark.

4.2. Terasortbenchmark

We created a 10GB raw text file by teragencommand; we evaluate an average execution time of terasort benchmark with snappy compression in map output. it shows compression between map reduce and Tez frameworks. In Figure 7, map reduce framework has execution time of 1,886.50 seconds which is better than Tez framework which is having execution time at 2212.00 seconds. In Snappy compression, it has same best performance in map reduce framework i.e, 115025 second and increases performance up to 39% without compressed case. Tez framework has 1256.00 second and increase performance up to 43% of a without compressed case. However, the compressed map output file with snappy compression that it can increase a computing performance more than39% on both the frameworks. On the other hand Tez framework has a performance which is lower than map reduce framework around 13% in hadoop cluster.

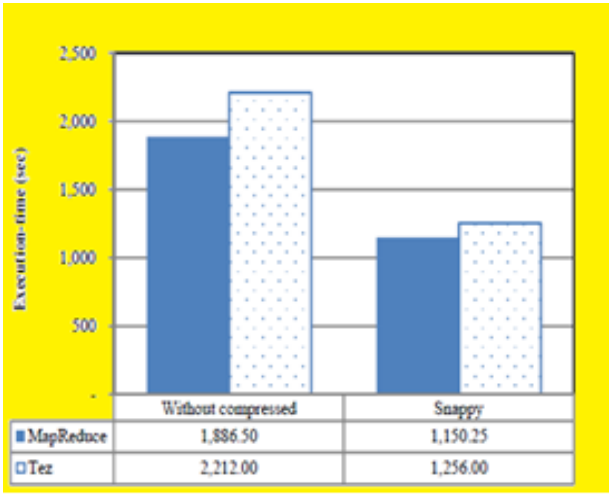

Fig. 10: Execution Time Comparison between Map Reduce and Tez in Terasort Benchmark.

5. Conclusion

Our experiment and results of Hadoop benchmarks, we installed hadoop cluster in figure 3 and evaluate an execution time of hadoop benchmark with a compressed or not compressed of map output by snappy compression. The best performance found in a compressed map output by snappy with both of map reduce and Tez frameworks in figure 6 and figure 7.In conclusion Hadoop cluster has become widely used which challenges to improve the performance of big data solution with compression algorithm. Tez is interactive query which is designed for data structure of Hadoop cluster that the new

challenge to compare a performance between Hadoop-hive with Tez and spark SQL by data compression.

References

[1] Apache software foundation, "welcome to apache Tez",[online] available:http://tez.apache.org,march 16,2017.

[2] R. Singh and P.J. Karu,"Analyzing performance of apache tez and Map Reduce with hadoop multinode cluster on Amazon cloud", J Big Data vol.3, no1,p.19,dec 2016.

[3] A.wenas, S.Suharjito,"Improving data technology warehouse performance using filesystem with gzip,lzjb and zle compression", journal informtikadans isteminformasi, vol2, no.2, pp 41-51,feb 2017.

[4] Y.Chen,A.Ganapthi,R.H.Katz,"To compress or not to compress compute vs. IO tradeoffs for MapReduce energy efficiency ", green networking 2010, new delhi, India,pp 23-28, August 30,2010.

[5] B.J Mathiya, V.I.Desai,"Hadoop map reduce performance evaluation and improvement using compression algorithms on single cluster", techrepublic, available: http://www//tecchrepublic.com/rescource-library/whitepapers/hadoop-MapReduce-performance-evaluation and improvement using compression algorithms on single cluster [Accessed 01-may-2017].

[6] The big Data blog,"Hadoop ecosystem overview", [online] available:http://the big data blog.weebly.com/blog/the-hadoop-ecossytem-overview,2016.

[7] Cve-2010-0405,"bzip2 and lib bzip2", [online] available:http://www/bzip.org/index.html,2017.

[8] G.Kamat,S.Singh,"Compression options in hadoop- a tale of tradeoffs",[online]available:http://www/slideshare.net/hadoop_summit/Kamat-singh-june27425pmroom210cv2,2013.

[9] Project gutenberg,"free ebooks by project gutenberg", [online] available:http://www/gutenberg.org,2017.