ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

Build Potential Usecase of Hadoop and Deploy On Cloud Vendor for Comparing Pipeline Execution Time and Cost of Cluster

Janvi Patel[1], Nidhi Gondalia[2]

PG Student, Department of C.E., Noble Group of Institutions, Junagadh, GTU, India[1]

Assistant Professor, Department of C.E., Noble Group of Institutions, Junagadh, GTU, India[2]

ABSTRACT: Big data is an emerging paradigm applied to datasets whose size or complexity is beyond the ability of commonly used computer software and hardware tools. Datasets are often from various sources (Variety) and are of large size (Volume) with fast data in/out (Velocity) that helps in business decision making which is hard to achieve using traditional system. Hadoop has become an effective and attractive solution for big data processing problems. It have been widely recognized for their high-throughput, elastic scalability, and fault tolerance. They focus more on these features than job execution efficiency. This results in relatively poor performance when using Hadoop MapReduce to execute jobs. There are different aspects to improve execution time. In this paper we present our approach on Hadoop to analyze its performance in terms of execution time with respect to different number of files on single-node and multi-node cluster on cloud. And also present an approach to improve execution time.

KEYWORDS: Hadoop, MapReduce, Execution Time Hive, Sparksql.

I. Introduction

Business Intelligence is useful in decision making. It is used at various levels of the business based on the needs. There is large number of tools containing modules to collect, enrich, analyze and finally provide visualization of the result. In the digital age, the previous day data is treated as archival data as the BI decisions are now based on data generated before minutes and seconds. There is a need to design a new platform or to restructure the existing one to analyze the huge amount of day generated per day. Big Data is the term which refers to the type (Variety), size (Volume), speed (Volume) of the data. Big data brings this technology change in terms of storing, retrieving, analysing and visualizing the data.

Google File System (GFS) was the first paper published by Google which described the way to store and process big data in a distributed manner. Based on this paper, Hadoop an open source java implementation of GFS was developed. The MapReduce parallel computing framework [1], proposed by Google in 2004, has become an effective and attractive solution for big data processing problems. Moreover, MapReduce offers other benefits, including load balancing, elastic scalability, and fault tolerance, which makes it a widely adopted parallel computing framework.

Though big data solves the problem of fast increase data and analysis of it, there is challenge at the hardware side which is unable to store the entire data on a single system leading to distributed storage. Handling of such cluster of hardware in terms of high availability, maintenance, security, fault tolerance is a huge challenge. Its downtime leads to unavailability of data to customer whichresults in loss of business. These challenges are addressed by cloud computing paradigm which serve scalability, on demand, orchestration and measured usage of resources.

II. Hadoop

Hadoop is a Programming framework used to support the processing of large data sets in a distributed computing environment. Hadoop was developed by Google's MapReduce that is a software framework where an application break

DOI: 10.15680/IJIRCCE.2016. 0404285

down into various parts. The Current Apache Hadoop ecosystem consists of the Hadoop Kernel, MapReduce, HDFS and numbers of various components like Apache Hive, Base and Zookeeper.
Two main component of Hadoop

A. Hadoop Distributed File System(HDFS)

Hadoop includes a fault-tolerant storage system called the Hadoop Distributed File System, or HDFS. HDFS is able to store huge amounts of information, scale up incrementally and survive the failure of significant parts of the storage infrastructure without losing data. Hadoop creates *clusters* of machines and coordinates work among them. Clusters can be built with inexpensive computers. If one fails, Hadoop continues to operate the cluster without losing data or interrupting work, by shifting work to the remaining machines in the cluster. HDFS manages storage on the cluster bybreaking incoming files into pieces, called “blocks,” and storing each of the blocks redundantly across the pool of servers. In the common case, HDFS stores three complete copies of each file by copying each piece to three different servers.

B. MapReduce

.

MapReduce is a distributed programming model is used for processing and generating large datasets. It is widely used for short jobs requiring low response time. Map and Reduce are the two programs used in MapReduce. Map ( ) function takes an input pair in the form of key and value. It then processes the input pair generating an intermediate set of <key, value> pairs. The Reduce( ) function merges the intermediate set of values associated with the same intermediate key. Generally one Reduce ( ) function produces typically just zero or one output values.

III. RELATED WORK

Following on Google’s MapReduce paradigm [1], proposed by Google in 2004, has become an effective and attractive solution for big data processing problems. MapReduce offers other benefits, including load balancing, elastic scalability, and fault tolerance, which makes it a widely adopted parallel computing framework. Despite of above advantages of MapReduce there is Performance issues – in terms of query execution time, Programming Model Issues – requires advanced programming skills Configuration & automation issues – Configuration of parameter, misconfiguration might leads to inefficient execution time. For performance analysis some work has been carried out, [8] describe the performance of K-Means algorithm on Eucalyptus platform and [9] perform experiment by executing MapReduce application on EC2, both show that performance scale with number of nodes. [10] Analyze the performance of word count on single node and multi node cluster(homogenous & heterogeneous) with respect to file size, the result show that in homogenous cluster perform with increasing file size performance i.e. execution time increase that can be improved by adding data nodes. While in heterogeneous cluster performance depends on right combination of heterogeneous nodes. And [11] they had implemented Hadoop on multi node cluster on on-premises to analyze vehicle diagnostics data.[1] analyze performance of Hadoop on single node with respect to number of files.All above had only performance analysis, we perform analysis as well also improve execution time. To improve the performance of Hadoop MapReduce framework, from different aspects there are many works have been done. They categories as follows: 1) to optimize the execution order of jobs or tasks more intelligently [16, 17, and 18] focuses on designing scheduling algorithms. 2) [24, 25, 26] show that with the aid of special hardware or supporting software to improve the efficiency of MapReduce. [12, 13, 14, and 15] is shows the various ways to improve the execution time. Another way to improve execution time is optimizing job configuration settings or parameters [27].In order to improve execution time, we first perform analysis of Hadoop by executing MapReduce application on multi node cluster created on cloud with respect to number of files and then in next step In order to improve execution time we will use proposed pipeline [MapReduce, Hive & Spark] to analyze data on same criteria and also compare result of both scenario.

IV. PROPOSED ALGORITHM

1. Create Instances
2. Deploy Hadoop services on created instances
3. Get data files and transfer it on hdfs
4. Create an external hive table from data stored in hdfsfrom the external hive table create managed tables for session and events. For the storage efficiency, data of managed tables is in parquet file format.
5. Connect to the parquet file using sparksql; the resultant processed data is stored in form of a file in hdfs.
6. Evaluate execution time.

V. PROPOSED FLOWCHART

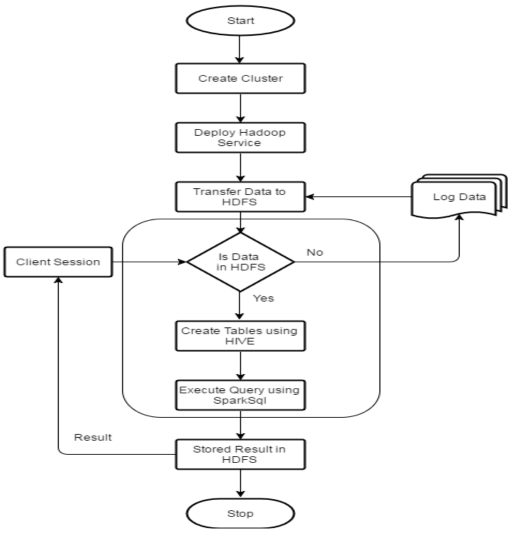


Fig.1. Proposed Flowchart

A. *Cluster Creation:*

Cluster of compute nodes with proposed configuration is instantiated along with thePersistent Hard Disk attached to the nodes. One of the nodes of the cluster is markedas Name/Master Node where the server will be launched and other nodes of thecluster are marked as Compute Nodes.

B. *Deploy Hadoop Service:*

After the master is launched it provides a web based UI to setup the Hadoop servicesover the compute nodes of the cluster. If we failed to deploy Hadoop service then weremove the existing setup and again start to deploy Hadoop service on theCompute/Data node.

C. *Data Transfer:*

Next step is to transfer the data from on premise to the cloud. For this experiment thedata was initially transferred to the bucket of a cloud object storage service where thecluster is to be formed. The dataset was copied to HDFS.The next step is to create External table from Enriched Data stored in the HDFSdirectory. External Table only point to the data in directory it does not copy it. Nownext is to create a Hive Managed Table from the External Table where it copy thedata and stored the data in parquet file format for storage efficiency.

D. *Execute Query using Sparksql:*

The next step is to connect created parquet file to Sparksql. Sparksql is the corecomponent of Apache spark that execute the query and the resultant processed data isstored in form of a file in HDFS. And finally result can be read by user from HDFS.

VI. Experimental setup

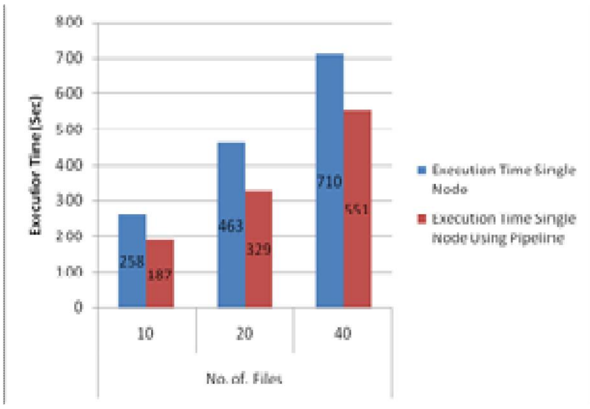
The experiments were setup using Google Cloud Platform (GCP). For our series of experiments, we have used different Resources their specification is shown in table II.

Table 1: ResourcesDescription

Operating system	CentOS 6.6
Deployment Model	IAAS
Compute	n1-highmem-2 [13GB Ram ,2 vCPU]
Disks	500 GB
Object Storage	Google Cloud Storage
Network	Default
Hadoop	HDP 2.2
Deployment utility	gcloud, bdutil ,gsutil,ApacheAmbari , SSH

VII. Result Analysis

Firstly we have performed experiment to analyze performance in terms of execution time on single node with respect to different no. of files and then perform to experiment with same scenario on multi-node cluster on Google Cloud Platform.



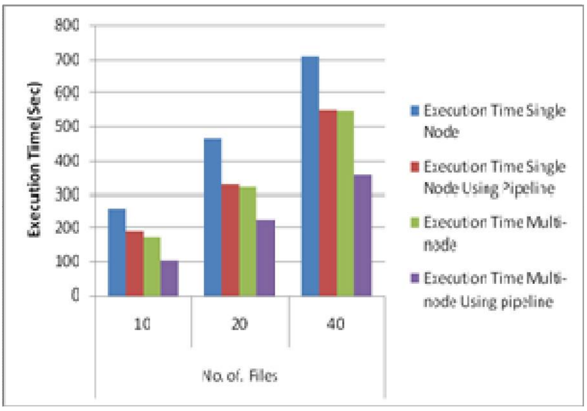


Fig.2.Single-node without using pipeline Vs single-node using pipelineFig. 3. Single& multi-node both using &without using pipeline

VIII. Conclusion and Future Work

In this paper, we have analysed and observed the results of MapReduce application on Single-node and Multi-node (homogeneous) Hadoop clusters on Google cloud platform with different numbers of files. It has been concluded that in both single-node and multi-node Hadoop setup execution time is improved using proposed pipeline. As a future work this approach can also be extended for the heterogeneous Hadoop cluster setup.

REFERENCES

1. J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Commun. ACM*, vol. 51, pp. 107–113, January 2008.
2. Amrit Pal ,PinkiAgrawal ,Kunal Jain , Sanjay Agrawal , "A Performance Analysis of MapReduce Task with Large Number of Files Dataset in Big Data Using Hadoop ", IEEE,2014
3. Stephen Kaisler, Frank Armour, J. Alberto Espinosa, William Money, "Big Data: Issues and Challenges Moving Forward", Hawaii International Conference on System Sciences, IEEE, 2012
4. JyotiNandimath, AnkurPatil, Ekata Banerjee, PratimaKakade, SaumitraVaidya, "Big Data Analysis Using Apache Hadoop", IEEE, 2013.
5. Harshawardhan S. Bhosale, Prof. Devendra P. Gadekar, "A Review Paper on Big Data & Hadoop", International Journal of Scientific and Research Publications (IJSRP), 4 (10) , 2014
6. Rainer Schmidt, Michael Möhring, "Strategic alignment of Cloud-based Architectures for Big Data", IEEE , 2013
7. Marcos D. Assunção , Rodrigo N. Calheiros , Silvia Bianchi , Marco A.S. Netto ,RajkumarBuyya, "Big Data computing and clouds: Trends and future directions", Elsevier, 2014
8. Jobby P Jacob, AnirbanBasu, "Performance Analysis of Hadoop Map Reduce on Eucalyptus Private Cloud", IJCA, 2013
9. ParthGohil, DweepnaGarg, Prof. BakulPanchal, "A Performance Analysis of MapReduce Applications on Big Data in Cloud based Hadoop" IEEE, 2014
10. Ruchi Mittal1, RuhiBagga, "Performance Analysis of Multi-Node Hadoop Clusters using Amazon EC2 Instances",IJSR,2014
11. Lionel Nkenyereye, Jong-Wook Jang, "A study of Big Data solution using Hadoop to process connected Vehicle's Diagnostics data", Springer, 2015
12. Qi Chen, Cheng Liu, and Zhen Xiao, Senior Member, "Improving MapReduce Performance Using Smart Speculative Execution Strategy",IEEE, 2013
13. Jia-Chun Lin, Fang-YieLeu, and Ying-ping Chen, "Impact of MapReduce Policies on JobCompletion Reliability and Job Energy Consumption",IEEE, 2013
14. R. Gu, X. Yang, J. Yan, Y. Sun, B. Wang, C. Yuan, Y. Huang, "SHadoop: Improving MapReduce performance by optimizing job execution mechanism in Hadoop clusters J. Parallel Distrib. Comput., 2013
15. Muhammad Idris , ShujaatHussain , Sungyoung Lee ,"In-Map/In-Reduce: Concurrent Job Execution in MapReduce", IEEE, 2014
16. Mao, H. and Hu, S. and Zhang, Z. and Xiao, L. and Ruan, L. A Load-Driven Task Scheduler with Adaptive DSC for MapReduce, in: 2011 IEEE/ACM International Conference on Green Computing and Communications (GreenCom), 2011, pp 28-33.
17. You, H.H. and Yang, C.C. and Huang, J.L, A load-aware scheduler for MapReduce framework in heterogeneous cloud environments, in: Proceedings of the 2011 ACM Symposium on Applied Computing, 2011, pp. 127-132.
18. Nanduri, R. and Maheshwari, N. and Reddyraja, A. and Varma, V., Job Aware Scheduling Algorithm for MapReduce Framework, in: 3rd IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 2011, pp. 724-729.
19. VibhaSarjolta, Dr. A.J Singh," A Study of Hadoop: Structure and performance Issues", International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), 2015.
20. VasilikiKalavri, Vladimir Vlassov, "MapReduce: Limitations, Optimizations and Open Issues", IEEE, 2013.
21. Seo, S. et al. , HPMR: Prefetching and Pre-shuffling in Shared MapReduce Computation Environment in: International Conference on Cluster Computing and Workshops (CLUSTER), 2009, pp. 1-8.
22. Wang, Y. and Que, X. and Yu, W. and Goldenberg, D. and Sehgal, D., Hadoop Acceleration Through Network Levitated Merge, in: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, 2011, pp. 57-67.
23. Li, B. and Mazur, E. and Diao, Y. and McGregor, A. and Shenoy, P., A Platform for Scalable One-Pass Analytics using MapReduce, in: Proceedings of the 2011 ACM SIGMOD international conference on Management of data, 2011, pp. 985-996.
24. Xin, M. and Li, H. An Implementation of GPU Accelerated MapReduce: Using Hadoop with OpenCL for Data-and Compute-Intensive Jobs, in: 2012 International Joint Conference on Service Sciences (IJCSS), 2012, pp. 6-11.
25. Becerra Fontal, Y. and Beltran Querol, V. and Carrera P, D. and others., Speeding up distributed MapReduce applications using hardware accelerators, in: International Conference on Parallel Processing(ICPP), 2009, pp. 42-49.
26. Zhang, S. and Han, J. and Liu, Z. and Wang, K. and Feng, S. Accelerating MapReduce with Distributed Memory Cache, in: 15th International Conference on Parallel and Distributed Systems (ICPADS), 2009,pp. 472-478.
27. Babu,S. , Towards automatic optimization of mapreduce programs, in: Proceedings of the 1st ACM symposium on Cloud computing (SoCC), 2011, pp.137-142.