International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 07 | July-2016 www.irjet.net p-ISSN: 2395-0072

HadoopWeb: MapReduce Platform for Big Data Analysis

Saloni Minocha[1], Jitender Kumar[2],s Hari Singh[3], Seema Bawa[4]

[1]*Student, Computer Science Department, N.C. College of Engineering, Israna, Panipat, Haryana*
[2]*Associate Professor, CSE Department, N.C. College of Engineering, Israna, Panipat, Haryana*
[3]*Associate Professor, CSE Department, N.C. College of Engineering, Israna, Panipat, Haryana*
[4]*Professor, CSE Department, Thapar University, Patiala, Punjab*

Abstract: *MapReduce is one of the most popular programming model for Big Data Analysis in distributed and parallel computing environment. The Apache Hadoop project offers an open source MapReduce enabled parallel data processing ecosystem. In recent times, there has been a lot going of research going on in the area of Hadoop in Virtual Environment. This paper focuses on Hadoop in Virtual environment and presents the implementation and evaluation of HadoopWEB, a web service middleware platform between the user and Google Cloud based Virtual Hadoop Cluster. This can be used by users to perform their own MapReduce jobs. Experiments are conducted for different applications of MapReduce programs on real time data set of Chicago Crimes using HadoopWeb. Test results shows that HadoopWEB can be used to execute MapReduce jobs by users.*

Keywords: MapReduce, Virtual Hadoop, Web Service, Middleware, Cloud computing, Virtualization, Big Data.

1. INTRODUCTION

Every year due to the advent of new technologies, devices and communication means, the amount of data produced by mankind is growing rapidly. The amount of data generated is still increasing enormously. All this information produced is meaningful and can be useful when processed. This data is termed as "Big Data." Big data is a collection of large datasets that can't be processed using traditional database techniques. It includes huge volume, high velocity and extensible variety of data that can be structured, semi-structured and unstructured type. There are a lot of challenges associated with Big Data like capturing data, storage, searching, sharing, analysis, transfer and presentation. Google solved this problem using a parallel algorithm known as MapReduce. This algorithm divides the task into small tasks and assigns these small tasks to computers connected over the network and collects the results from computers to generate a final result dataset. However, in 2005 an open source project called Apache Hadoop was introduced by Doug Cutting and Mike Cafarella. Hadoop runs applications using MapReduce algorithms on clusters of computers and can perform statistical analysis for a huge amount of data. The four main components of Hadoop are as follow:

1) Hadoop Common: This includes java libraries and utilities required by Hadoop applications.
2) Hadoop Yarn: Framework for job scheduling and cluster resource management.
3) HDFS: Hadoop Distributed File System that supports high throughput access to application data.
4) Hadoop MapReduce: Yarn based system for parallel analysis and processing of large datasets.

Hadoop for a computation and storage uses a single server to thousands of machines. It provides a good performance in physical cluster but the huge amount of data launch in the system with less managing and deploying work is necessary. Efficient resource utilization and power saving is also necessary. Hence, in recent times the focus has been shifted to virtualization of Hadoop or Hadoop on cloud. Cloud Computing is a kind of internet based computing which enables on- demand access to a shared pool of configurable computing resources. There are three service models in cloud: Infrastructure as a Service (IaaS), Platform as a service (PaaS) and Software as a service (SaaS). Hadoop on cloud server allows less physical hardware requirements, less disscipation of heat; hence reduced electricity consumption and cost saving. There are also added advantages like easier maintenance, the effect of one virtual computer will not affect the other virtual computer, effective use of hardware, backups are made easier and also faster disaster recovery.

Amazon Elastic Compute Cloud, Microsoft Azure, IBM Smart Cloud or Google Cloud platform are few Cloud Computing Platforms that can be used to configure a Virtual Hadoop Cluster. Apart from these there are open source software solutions that can be used to make various cloud environments: Public, Private or Hybrid Clouds and can be further used to configure Hadoop in Virtual Environment. These are Eucalyptus, OpenNebula, OpenStack and CloudStack.

The paper is divided into several sections for the ease of understanding. Section 2, presents the literature survey and recent works in the field of Virtual Hadoop. Section 3, presents the proposed system and experimental setup details. Section 4 presents the results and observations. Section 5, states the conclusion drawn from the paper.

2. LITERATURE SURVEY AND RECENT WORKS

Pratiksha D. Mandal *et al*, 2016[1] in their paper proposed a system with the goal of adding the features of Elasric MapReduce to open source private cloud. They also stated limitations of Amazon EMR that it works on private cloud and does not allow to use Amazon EMR service on the private cloud, as a result user is not able to use their own infrastructure for task execution. Another limitation is the client needs for the service. They concluded in their paper that current Hadoop environments are manually deployed on physical servers and lack flexibility. They are presenting a framework capable of supporting a cluster to be dynamically elastic and also providing the same functionality as as Amazon EMR but on the top of private cloud build on OpenStack platform. This will enable users having access to private cloud to run their MapReduce computation tasks without having to worry about cloud resources management, failure handling that too free of cost. Hiren Parmar and Tuskar Champaneria, 2014[2] in their paper compared and analyzed OpenNebula, Eucalyptus, OpenStack and CloudStack platforms. These all are Cloud management platforms providing Infrastructure as a service. They concluded in their paper that OpenNebula and CloudStack are more towards the datacenter virtualization whereas Eucalyptus and OpenStack more towards the infrastructure provisioning. Among all the four, OpenNebula is more flexible they concluded. Arun S. Devadiga *et al,* 2014[3] in their paper integrated cloudstack, Hadoop and KVM. This integration resulted in Virtual Hadoop which will allow user to process huge amounts of data concurrently in virtual environment with efficient use of resources. In their paper they concluded that the virtual hadoop has slightly higher but almost similar execution time to execute the MapReduce program than the physical Hadoop and the advantages of virtual hadoop are that the management is easier, full utilization of computing resources, making Hadoop more reliable and save power. Parth Gohil *et al*, 2014[4] in their paper observed and analyzed the results of various MapReduce Applications on Big Data in cloud based Hadoop cluster and concluded that the results of MapReduce Applications are dependent on the size of Hadoop Cluster. Dweepna Garg and Bahul Panchal, 2014[5] in their paper discussed various MapReduce Applications like Word Count, Pi, Terasort and Grep in cloud based Hadoop using Amazon EMR. They concluded that as the number of nodes increases the execution time decreases and performance increases. Jobby P. Jacob and Akirban Basu, 2013[6] in their paper analyzed the performance of K-means clustering algorithm using Hadoop Mapreduce on Eucalyptus. They concluded that increase in the number of nodes boosted performance significantly. Also creating and running an OS is much faster as prebuild images which can be launched very easily and customized according to the user requirements. Cloud also gives the user the ability to supply more machines when needed as long as it is not reaching the physical upper limits of the underlying host machines. Anca Iordache *et al*, 2013[7] in their paper presented the design, implementation and evaluation of RESLIN, a novel EMR-API compatible system to perform distributed MapReduce Computations. RESLIN goes one step beyond Amazon EMR solution and allows users to leverage resources from one or multiple public/private clouds. In future, they plan to extend the system by adding features allowing to automatically scaling the execution platform based on MapReduce job profiles and high level objectives. They will also conduct more experiments with different cloud managers, applications and parameters. Ruchi Mittal and Ruhi Bagga, 2013[8] in their paper evaluated and analyzed the performance of word count MapReduce application using Hadoop on Amazon EC2 using different Ubuntu instances and concluded that a threshold exists below which adding more nodes does not result in performance enhancement in the cluster but after that value with increasing number of data nodes, the Hadoop Cluster performance can be enhanced. Javier Conejaro *et al*, 2013[9] in their paper present the COSMOS platform supporting sentimental and tension analysis on Twitter data and demonstrate how this platform can be scaled using OpenNebula cloud environment with MapReduce based analysis using Hadoop. They concluded in their paper that the performance benefits can be achieved by using multiple virtual nodes, compared to running a sequential version of the application. However, the greater the worker nodes deployed, the better the performance achieved, but they are limited by the amount of VMs that can run simultaneously without affecting another. Hence, there is a limit on the number of VMs that can be hosted per physical machine to ensure consistent and reliable performance. Samira Daneshyar and Majid Razmjoo, 2012[10] proposed a framework where MapReduce application can be processed on Amazon cloud environment and validated the proposed framework by running experiments of MapReduce applications in cloud environment. From their analysis they concluded that using Cloud computing and MapReduce together improves the speed of processing and decreases the response time and cost of processing of large datasets.

3. PROPOSED SYSTEM

HadoopWEB: It is a middleware web service between the users and Google Cloud based Hadoop Cluster. The users will be able to do data analysis without having to configure physical Hadoop and also without having to go on the Cloud Platform. For a user it will be a fast and an easy platform to do data analysis, the user just has to upload the data and MapReduce Jar file for the execution. Thus, it will increase efficiency on the user end.

3.1Experimental Setup

The backend framework of Virtual Hadoop Cluster of 8 nodes was set up using Google Cloud Platform. For our series of experiments one instance of n1-standard-1-GNU/Linux8 (jessie) 1vCPU 3.75GB memory, 500GB SSD persistent Disk was used as Master Node and other 7 instances of n1-standard-1-GNU/Linux8 (jessie) 1vCPU, 3.75GB memory and 10GB SSD persistent Disks were used as Data Nodes. Hadoop-2.6.4 and 64 bit Java jdk 1.8.0_91 were installed in all the instances.

The frontend of HadoopWeb service was developed in HTML, CSS and PHP5.6 and configured in Master Node.

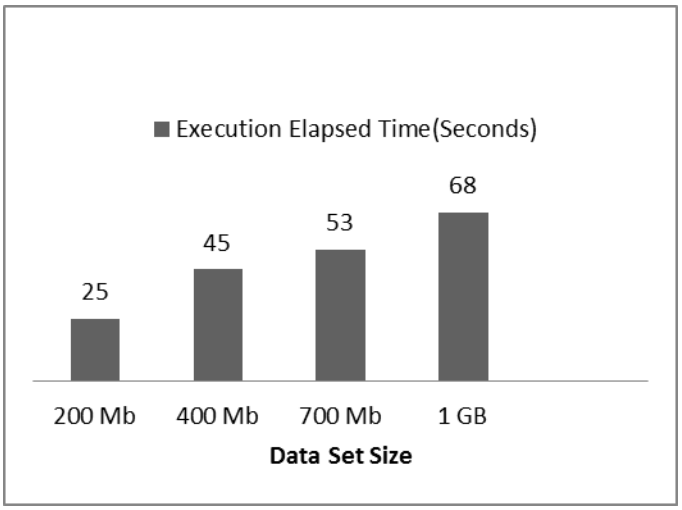
4. RESULTS AND OBSERVATIONS

In this section we have analyzed the data set picked from City of Chicago data portal known as Crimes-2001 to present, using MapReduce and HadoopWeb. This dataset reflects reported incidents of crimes that occurred in the City of Chicago from 2001-2008. The row count of this dataset was 6084910. It had 21 columns in total and was of 1.33 GB in size.

The results are observed in two parts:

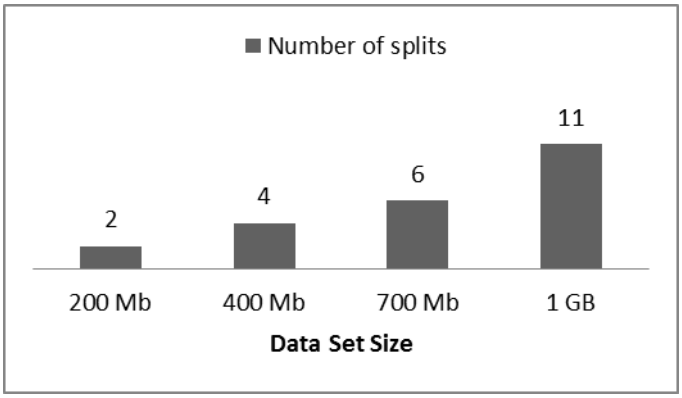
a) The execution time, Number of splits, Launched Map Tasks and CPU Time is observed by varying the size of dataset. The MapReduce program which is applied on the dataset was to list out unique crime types and their location description. The following graphs depict the results:

Chart-1: Execution Elapsed Time: Data Set Size vs Elapsed Time



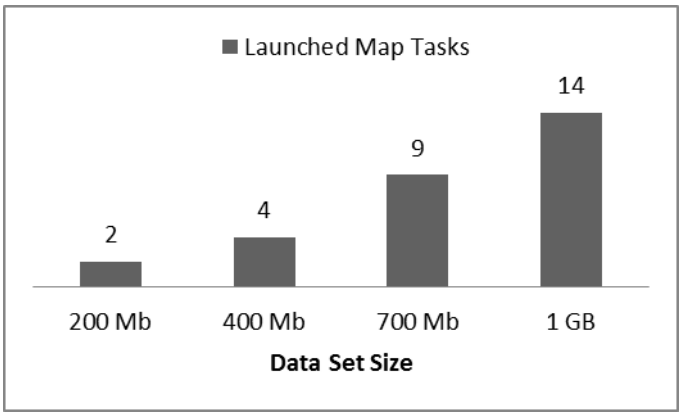
Job duration is the elapsed time for the job. It is observed that as the Data Set size increases, the Elapsed Time increases.

Chart-2: Number of Splits: Data Set Size vs Number of Splits



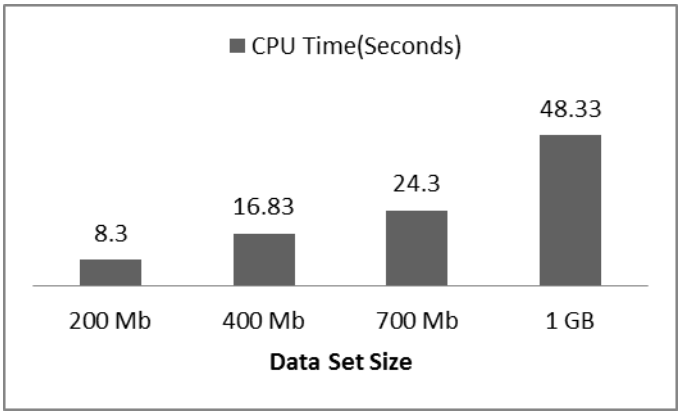
When you input data into Hadoop Distributed File System (HDFS), Hadoop splits your data depending on the block size (default 64/128 MB) and distributes the blocks across the cluster. It is observed that as the Data Size increases the number of splits increases accordingly.

Chart-3: Launched Map Tasks: Data Set size vs Launched Map Tasks.



The Application Master in Hadoop launches one Map Task for each map split. Typically, there is a map split for each input file. If the input file is too big (bigger than the HDFS block size) then we have two or more map splits associated to the same input file. It is observed that the launched map tasks increases as the Data Size is increased.

Chart-4: CPU Time: Data Set size vs CPU time



CPU time is the total of CPU cycles spent for the job across the cluster. So, if a job is run for 1h, it could comprise of thousands of map tasks. As a result it might be possible that the cumulative CPU time will amount to CPU time higher than the elapsed time for the job. In our experiment we observed that as the Data Set Size increases the CPU time spent increases, but is less than the total elapsed time.

b) HadoopWeb is tested with 9 MapReduce programs on the same dataset and the following counter tables were observed:

Table 1 presents the Number of launched Map Tasks and Killed Map Tasks per Application. Different values are observed for every application. The dataset size and the cluster size are same throughout.

Table-1: Number of Launched Map Tasks and Number of Killed Map Tasks per Application

Number of Launched Map Tasks and Number of Killed Map Tasks per Application		
Application Program of MapReduce	Number of Launched Map Tasks	Number of Killed Map Tasks
List unique Crime Types and Location Descriptions	13	3
Total Count of Number of Arrests	12	1
Total Count of Number of Domestic Crimes	13	2
List unique Domestic Crimes	14	4
List unique Crimes where arrests was made	14	4
List unique Crimes where arrest was not made	14	3
List unique Domestic Crimes where arrest was made	12	1
List unique Domestic Crimes where arrest was not made	14	3
Join query Crime Primary Type, Location Description and Arrest Status	14	4

Table 2 presents the total time taken by all the Map Tasks and Reduce Tasks per Application. Different values are observed per Application.

Table-2: Total Time By all Map and Reduce Tasks per Application

Total Time By all Map and Reduce Tasks per Application		
Application Program of MapReduce	Total Time by all Map Tasks(Seconds)	Total Time by all Reduce Tasks(Seconds)
List unique Crime Types and Location Descriptions	429.630	34.492

Total Count of Number of Arrests	565.564	9.083
Total count of Number of Domestic Crimes	567.398	25.763
List of unique Domestic Crimes	474.001	50.696
List of unique Crimes where Arrest was made	538.855	54.772
List of unique Crimes where Arrest was not made	585.874	42.754
List of unique Domestic Crimes where Arrest was made	509.871	14.313
List of unique Domestic Crimes where Arrest was not made	563.702	42.346
Join query on Crime Primary Type, Location Description and Arrest Status	531.566	58.392

Table 3 presents the Garbage Collection Time and CPU Time per Application. Garbage Collection Time is a JVM counter which is observed during MapReduce job. Different values are observed per Application.

Table-3: Garbage Collection Time and CPU Time per application

Garbage Collection Time and CPU Time per application		
Application Program of MapReduce	GC Time (Seconds)	CPU Time (Seconds)
List unique Crimes and Location Descriptions	6.012	46.770
Total Count of Number of Arrests	7.749	46.030
Total Count of Number of Domestic Arrests	7.871	40.580
List unique Domestic Crimes	6.138	37.520
List unique Crimes where Arrest was made	7.246	43.930
List unique Crimes where Arrest was not made	7.893	45.870
List unique Domestic Crimes where Arrest was made	7.275	40.710
List unique Domestic Crimes where Arrest was not made	7.041	40.750
Join query on Crime Primary Type, Location Description and Arrest Status	5.705	48.530

5. CONCLUSION AND FUTURE SCOPE

It is found that Virtual Hadoop has advantages over Physical Hadoop like management is easier, full utilization of computer resources making Hadoop more reliable and it also saves power. Along with this there is cost saving, less physical hardware and less dissipation of heat. It is observed that proposed system HadoopWeb can be used by users to perform MapReduce Applications. The two unique features of HadoopWeb are: The users can upload data directly from the server link and the users can set their own replication

factor for the data they upload unlike other Web Services like Amazon EMR. In future, the direction will be to tune the number of Map and Reduce tasks appropriately according to the input data. It can be done in one of the three ways: Reducing the number of tasks if each task completes in less than 30-40 seconds, can increase the block size of data or can increase the mapper tasks to some multiple number of the mapper slots in the cluster. Furthermore, the backend of Hadoop Cluster can be made using CloudStack, which is an open source software designed to deploy and manage large networks of VM's, as an available and scalable Infrastructure as a Service cloud platform.

ACKOWLEDGEMENT

This paper is a part of the study and experimental work accomplished towards the sponsored project. We thank All India Council for Technical Education (AICTE)-India for sponsoring the project titled "Developing a Grid-GIS Framework for Spatial Data" with file no.: 20/AICTE/RIFD/RPS (POLICY-1)6/2013-14 for duration of three years under the Research Promotion Scheme (RPS).

REFERENCES

[1] Mandal D. Pratiksha, " Study of Elastic Hadoop On Private Cloud." International Journal of Scientific and Research Publications, Volume 6, Issue 1, January 2016 321 ISSN 2250-3153.

[2] Parmar, Hiren, and Tushar Champaneria. "Comparative Study of Open Nebula, Eucalyptus, Open Stack and Cloud Stack." *International Journal of Advanced Research in Computer Science and Software Engineering* 4.2 (2014).

[3] Devadiga, Arun S., P. R. Shalini, and Aditya Kumar Sinha. "Virtual Hadoop: The Study and Implementation of Hadoop in Virtual Environment using CloudStack KVM." *International Journal of Engineering Development and Research*. Vol. 2. No. 2 (June 2014). IJEDR, 2014.

[4] Gohil, Parth, Dweepna Garg, and Bakul Panchal. "A performance analysis of MapReduce applications on big data in cloud based Hadoop." *Information Communication and Embedded Systems (ICICES), 2014 International Conference on*. IEEE, 2014.

[5] Garg,Dweepna and Bakul Panchal. "A performance analysis of MapReduce applications on big data in cloud based Hadoop." *Information Communication and Embedded Systems (ICICES), 2014 International Conference on*. IEEE, 2014.

[6] Jacob, Jobby P., and Anirban Basu. "Performance Analysis of Hadoop Map Reduce on Eucalyptus Private Cloud." *International Journal of Computer Applications* 79.17 (2013).

[7] Iordache, Anca, et al. "Resilin: Elastic MapReduce over multiple clouds." *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*. IEEE, 2013.

[8] Mittal, Ruchi, and Ruhi Bagga. "Performance Analysis of Multi-Node Hadoop Clusters using Amazon EC2 Instances." International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064 Index Copernicus Value (2013).

[9] Conejero, Javier, et al. "Scaling archived social media data analysis using a hadoop cloud." *IEEE 6th International Conference on Cloud Computing (CLOUD)*. IEEE, 2013.

[10] Daneshyar, Samira, and Majid Razmjoo. "Large-scale data processing using Mapreduce in cloud computing Environment." *International Journal on Web Service Computing* 3.4 (2012): 1.