International Journal of Database Theory and Application
Vol.8, No.5 (2015), pp.15-26
http://dx.doi.org/10.14257/ijdta.2015.8.5.02

Performance Evaluation of Hadoop and Oracle Platform for Distributed Parallel Processing in Big Data Environments

A. K. M. Mahbubul Hossen [1], A. B. M. Moniruzzaman, Member, IEEE, [2] and
Syed Akhter Hossain, Member, IEEE, ACM[3]

Department of Computer Science and Engineering
Daffodil International University
Dhaka, Bangladesh
[1]*akhossen@banglalinkgsm.com*, [2]*abm.moniruzzaman.eng@ieee.org*,
[3]*aktarhossain@daffodilvarsity.edu.bd*

Abstract

Hadoop–a popular open-source implementation of MapReduce is widely used for the analysis of large datasets. The current Hadoop implementation assumes that computing nodes in a cluster are homogeneous in nature. In this paper we evaluate performance of Hadoop Platform and Oracle for Distributed Parallel Processing in large datasets. For evaluation, we implement a prototype of a virtual datacenter using distributed and parallel computing technology. The purpose of this paper is to reduce datacenter implementation cost using commodity hardware and provide high performance. Hadoop is installed on a commodity Linux cluster the distributed processing of large data sets across clusters of computers using distributed and parallel computing architecture. This paper also helps to explain about some new technology and framework which are open source; that can easily utilize those technologies for our complex data analysis which resembling structured, semi structured and non-structured data. Here we tried to demonstrate a performance comparison by executing some queries between distributed parallel computing system and traditional single computing system. For the simulation of the infrastructure Hadoop cluster has been used for distributed parallel processing and Oracle 11g is used for traditional single processing system. We prepare three virtual host for Hadoop cluster and a high-end hardware for Oracle 11g.

Keywords: *In according to benchmarking standard we use TPC BENCHMARK ™ DS" by-Transaction Processing Performance Council (TPC)*

1. Introduction

Digital world is growing very fast and become more complex in the volume (terabyte to petabyte), variety (structured and un-structured and hybrid), velocity (high speed in growth) in nature. This refers to as 'Big Data' that is a global phenomenon [1-3].

The MapReduce [4-7] programming model has emerged as a scalable way to perform data-intensive computations on commodity cluster computers. Hadoop [8-11] is a popular open-source implementation of MapReduce for the analysis of large datasets. To manage storage resources across the cluster, Hadoop uses a distributed user-level filesystem. This filesystem — HDFS — is written in Java and designed for portability across heterogeneous hardware and software platforms. The efficiency of the MapReduce model has been questioned in recent research contrasting it with the parallel database paradigm for large-scale data analysis. Typically, Hadoop is used as representative of the MapReduce model because proprietary (*e.g.*, Google-developed) implementations with potentially higher performance are not publicly available. In one study, Hadoop applications performed poorly in experiments when compared to similar programs using

ISSN: 2005-4270 IJDTA

parallel databases [12-13]. However, this work did not perform the profiling necessary to distinguish the fundamental performance of the Hadoop platform for parallel processing of large datasets compared to conventional data platform *e.g.* Oracle. In this paper we target to evaluate performance of Hadoop Platform and Oracle for distributed parallel processing in large datasets. For evaluation, we design and implement a prototype of a virtual datacenter using distributed and parallel computing technology. Hadoop is installed on a commodity Linux cluster the distributed processing of large data sets across clusters of computers using distributed and parallel computing architecture. For the simulation of the infrastructure Hadoop cluster has been used for distributed parallel processing and Oracle 11g is used for traditional single processing system. We prepare three virtual host for Hadoop cluster and a high-end hardware for Oracle 11g. In according to benchmarking standard we use TPC BENCHMARK ™ DS" by-Transaction Processing Performance Council (TPC). This paper is organized as follows:

In this paper is presented as follows: Section 2. Related Technologies and Architectures. And finally Section 8, shows conclusion of this paper.

2. Related Technologies and Architectures

This Section describes technologies and architectures relating to big data environment. Major technologies combined with virtualization technologies, Hadoop framework, MapReduce, Hadoop distributed file systems and main components of hadoop.

2.1. Hardware Virtualization

Hardware virtualization or platform virtualization refers to the creation of a virtual machine that acts like a real computer with an operating system. Software executed on these virtual machines is separated from the underlying hardware resources. For example, a computer that is running Microsoft Windows may host a virtual machine that looks like a computer with the Ubuntu Linux operating system; Ubuntu-based software can be run on the virtual machine. In hardware virtualization, the host machine is the actual machine on which the virtualization takes place, and the guest machine is the virtual machine. The words host and guest are used to distinguish the software that runs on the physical machine from the software that runs on the virtual machine. The software or firmware that creates a virtual machine on the host hardware is called a hypervisor or Virtual Machine Manager [6].

Hadoop [8] is an open source framework that implements the MapReduce parallel programming model [4]. Hadoop is composed of a MapReduce engine and a user-level filesystem that manages storage resources across the cluster [5]. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. Hadoop can be installed on a commodity Linux cluster to permit large scale distributed data analysis. It is designed to scale up from a single server to thousands of machines, with very high degree of fault tolerance. Rather than relying on high-end hardware, the resiliency of these clusters comes from the software's ability to detect and handle failures at the application layer. No hardware modification is needed other than possible changes to meet minimum recommended RAM, disk space, *etc.* requirements per node. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly available service on top of a cluster of computers, each of which may be prone to failures. Basically, it's a way of storing enormous data sets across distributed clusters of servers and then running "distributed" analysis applications in each cluster. It's designed to be robust, in that our Big Data applications will continue to run even when individual servers

— or clusters — fail. And it's also designed to be efficient, because it doesn't require applications to shuttle huge volumes of data across your network.

Hadoop has two main parts - a data processing framework (MapReduce) and a distributed filesystem for data storage (HDFS).

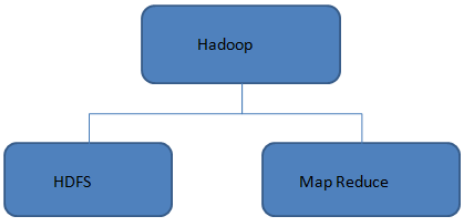


Figure. Main Parts of Hadoop

The hadoop distributed file system [8, 14-15] is like the bucket of the Hadoop system. We can dump in our data and it sits there all nice and cozy until we want to do something with it, whether that's running an analysis on it within Hadoop or capturing and exporting a set of data to another tool and performing the analysis there.

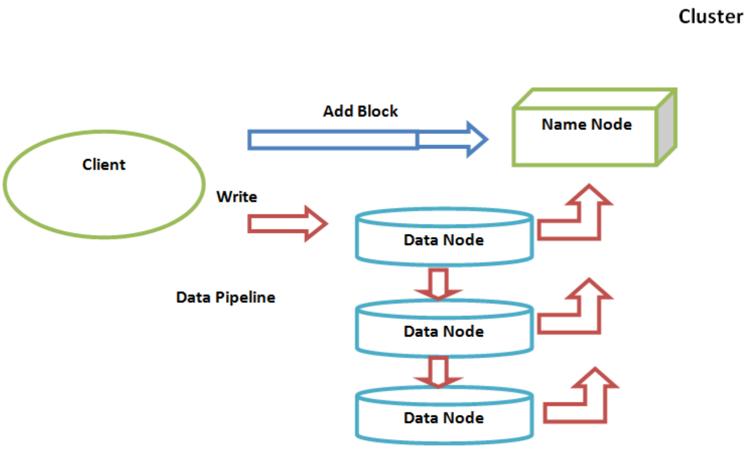


Figure. The Hadoop Distributed File System

MapReduce is a programming model for processing and generating large data sets implementations. A map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key [4]. Programs written with MapReduce are automatically parallelized and executed on a large cluster of commodity hardware. The run-time system deals of partitioning the input data, scheduling the program's execution across a set of machines, handling machine failures, and managing the required inter-machine communication. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system [5]. The data processing framework is the tool used to work with the data itself. By default, this is the Java-based system known as MapReduce. In a "normal" relational database, data is found and analyzed using queries, based on the industry-standard Structured Query Language (SQL). Non-relational databases use queries, too; they're just not constrained to use only SQL, but can use other query languages to pull information out of data stores. Hence, the term NoSQL. But Hadoop is not really a database. It stores data and we can pull data out of it, but there are no queries involved - SQL or otherwise. Hadoop is more of a data warehousing system - so it needs a system like MapReduce to actually process the data. MapReduce runs as a series of jobs, with each job essentially a separate Java application that goes out into the data and starts pulling out information as needed [5].

Hadoop is composed of four core components—Hadoop Common, Hadoop Distributed File System (HDFS), MapReduce and YARN.

Master/NameNode and Slave/DataNode

The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. It does not store the data of these files itself.Client applications talk to the NameNode whenever they wish to locate a file, or when they want to add/copy/move/delete a file. The NameNode responds the successful requests by returning a list of relevant DataNode servers where the data lives. [8]

A DataNode stores data in the [HadoopFileSystem]. A functional filesystem has more than one DataNode, with data replicated across them.On startup, a DataNode connects to the NameNode; spinning until that service comes up. It then responds to requests from the NameNode for filesystem operations.Client applications can talk directly to a DataNode, once the NameNode has provided the location of the data. Similarly, MapReduce operations farmed out to TaskTrackerinstances near a DataNode, talk directly to the DataNode to access the files. TaskTracker instances can, indeed should, be deployed on the same servers that host DataNode instances, so that MapReduce operations are performed close to the data. [9]

An architecture of Hadoop framework has been provide in Figure 1.4 where itself explains how NameNode and DataNodes are connect with each other.

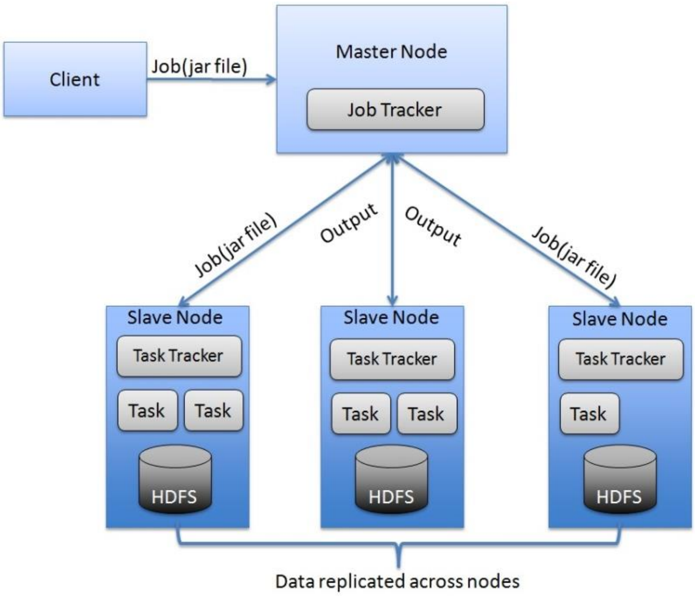


Figure 1.4. Architecture of Hadoop Framework [10]

Implementation Details

If we consider a small organization (profitable or non-profitable) we can find so many desktop and low-end servers which are utilize only 5% to 10%. So we can find such hardware and use them to design a Virtual Datacenter of distributed and parallel processing architecture, where we can process big amount of research data like structured, unstructured and semi-structured data. For this project we need Hypervisor software, Linux Operating System and Hadoop framework.

Infrastructure Design

Let’s assume a small organization have two remote site and they have 20 workstations in each site. Now design a virtual datacenter using these 40 workstations and additional two high-end servers. User can use these workstations to perform their usual purpose and in background we can install hypervisor to create another virtual machine on each physical machine. Combining those virtual machine (Slave/DataNode) and two high-end machine (Master/NameNode) with Hadoop cluster and design avirtual datacenter infrastructure with distributed and parallel processing power.

First of all place a high-end server in each site which is called as cluster Master/NameNode and other 20 workstation will work as Slave/DataNode.Hadoop framework will be install on Master/NameNode. Need to install Hypervisor software in each workstation to create a virtual server and add the virtual server to the Hadoop cluster as Slave/DataNode.Same task need to complete on both site and finally conFigure Hadoop cluster services and other related services to Master and Slave nodes.

Please find the below infrastructure design in Figure 2.1 where two remote site is connected with each other through privet network. Each site have LAN connection and all the workstations are connected to LAN. A high-end server has been placed to each site connected to LAN.

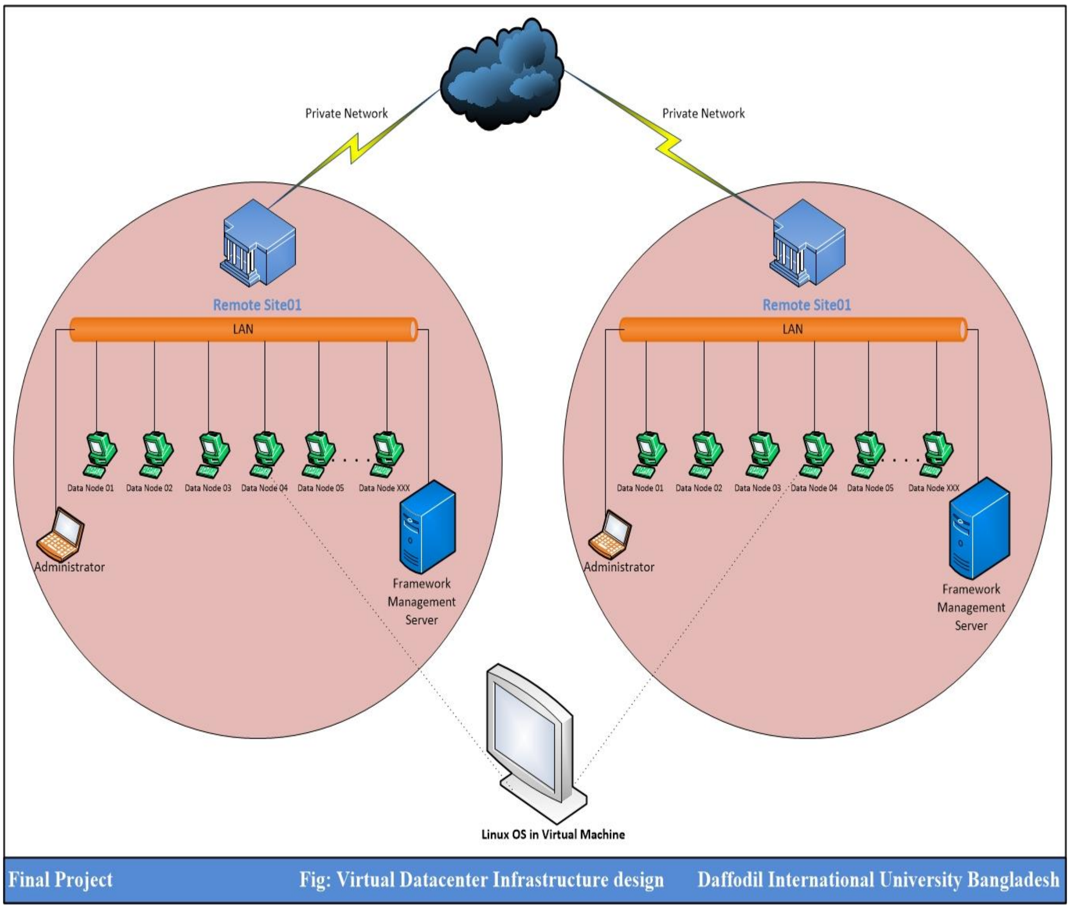


Figure 2.1. Infrastructure Design of a Virtual Datacenter

Installation Steps Summery

To simulate the infrastructure 64 bit Red Hat Linux6 has been used as operating system on Master and Slave servers in this project. Cloudera Manager CDH5.2.0 distribution has been used for Hadoop cluster framework and VMware has been used as hypervisor to prepare this virtual datacenter as a simulation.

Here we tried to demonstrate a performance comparison by executing some queries between distributed parallel computing system and traditional single computing system. For the simulation of the infrastructure Hadoop cluster has been used for distributed parallel processing and Oracle 11g is used for traditional single processing system. We prepare three virtual host for Hadoop cluster and a high-end hardware for Oracle 11g.

In according to benchmarking standard for Logical database design, Scaling and Database population, Query selection to execute and compare the performance by following an standard benchmarking guideline document from "TPC BENCHMARK ™ DS" by-Transaction Processing Performance Council (TPC), Version 1.3.0, publish on November 2014 [11-12].

Hardware Details

Oracle Database Server configuration:

System Model: HP Compaq 6000 Pro MT PC
System Type: x64-based PC
Processor: Intel(R) Core 2 Quad CPU @ 2.66GHz
OS: Red Hat Enterprise Linux 64 bit
RAM: 24 GB

Hadoop Cluster Server configuration:

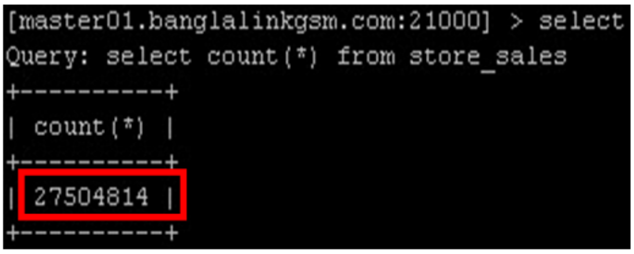
Name Node / Master Server
System Type: VMware
Processor: 4 vCPU
OS: Red Hat Enterprise Linux 64 bit
RAM: 16 GB

Data Node / Slave Server
System Type: VMware
Processor: 1 vCPU
OS: Red Hat Enterprise Linux 64 bit
RAM: 8 GB

Performance Comparison

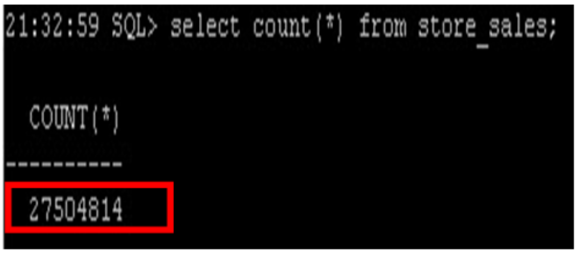
With the help of TPC guideline we design Database on both systems. Generate same set of data and table on each system. We uses TPC-DS provided Data Generator tools to generate benchmarking standard dataset.

Here Figure 3.1 and Figure 3.2 show the same dataset on a table "store_sales" on both of the database (Hadoop and Oracle). Some queries will be execute on the table "store_sales" to compare performance of distributed parallel processing and single processing system.



```
[master01.banglalinkgsm.com:21000] > select
Query: select count(*) from store_sales
+----------+
| count(*) |
+----------+
| 27504814 |
+----------+
```

Figure 3.1. Hadoop Dataset

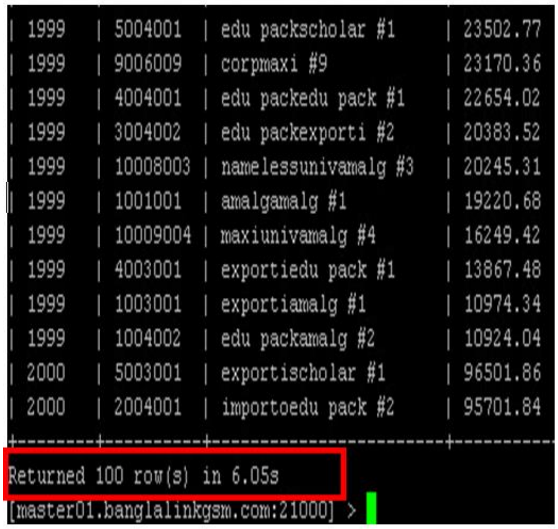


```
21:32:59 SQL> select count(*) from store_sales;

  COUNT(*)
----------
  27504814
```

Figure 3.2. Oracle Dataset

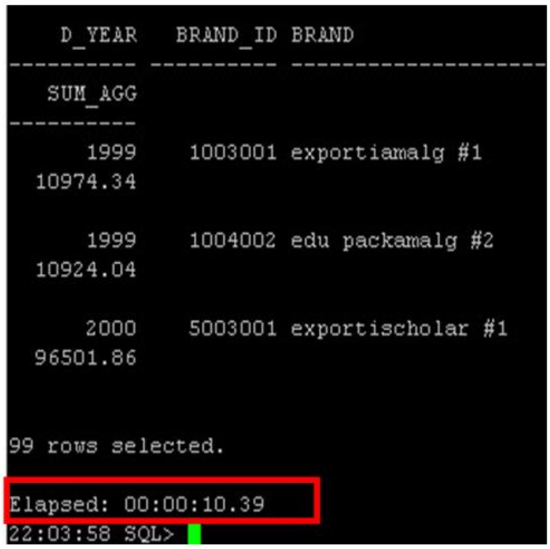
Query Example One

This query is collected from TPC benchmarking document query number Q3. This query report the total extended sales price per item brand of a specific manufacturer for all sales in a specific month of the year. Figure 3.3 and Figure 3.4 shows the time consumed to run the query on both of the systems.



```
| 1999   | 5004001  | edu packscholar #1     | 23502.77
| 1999   | 9006009  | corpmaxi #9            | 23170.36
| 1999   | 4004001  | edu packedu pack #1    | 22654.02
| 1999   | 3004002  | edu packexporti #2     | 20383.52
| 1999   | 10008003 | namelessunivamalg #3   | 20245.31
| 1999   | 1001001  | amalgamalg #1          | 19220.68
| 1999   | 10009004 | maxiunivamalg #4       | 16249.42
| 1999   | 4003001  | exportiedu pack #1     | 13867.48
| 1999   | 1003001  | exportiamalg #1        | 10974.34
| 1999   | 1004002  | edu packamalg #2       | 10924.04
| 2000   | 5003001  | exportischolar #1      | 96501.86
| 2000   | 2004001  | importoedu pack #2     | 95701.84
+--------+----------+------------------------+----------
Returned 100 row(s) in 6.05s
[master01.banglalinkgsm.com:21000] >
```

Figure 3.3. Q3 Executed on Hadoop Cluster

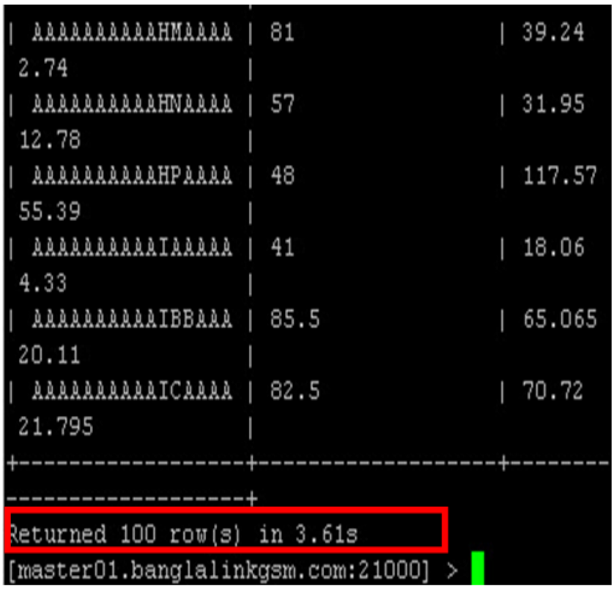


```
    D_YEAR   BRAND_ID BRAND
---------- ---------- --------------------
   SUM_AGG
----------
      1999    1003001 exportiamalg #1
  10974.34

      1999    1004002 edu packamalg #2
  10924.04

      2000    5003001 exportischolar #1
  96501.86


99 rows selected.

Elapsed: 00:00:10.39
22:03:58 SQL>
```

Figure 3.4. Q3 Executed on OracleDB

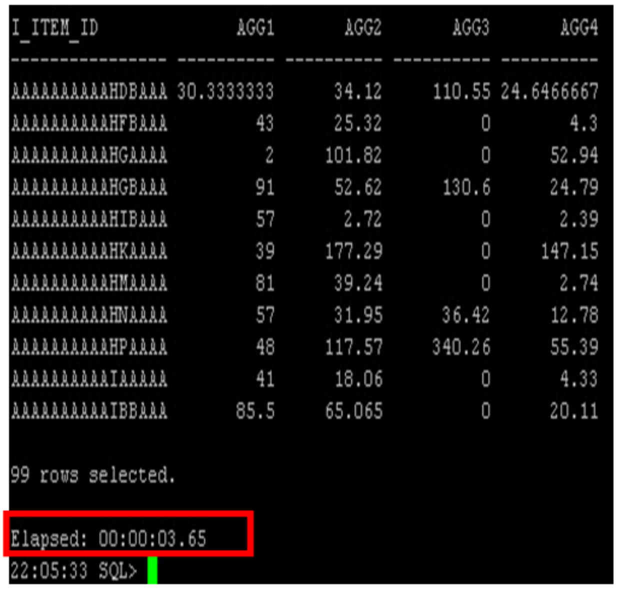
Query Example Two

This query is collected from TPC benchmarking document query number Q7. This query report to compute the average quantity, list price, discount, and sales price for promotional items sold in stores where the promotion is not offered by mail or a special event. Restrict the results to a specific gender, marital and educational status. Figure 3.5 and Figure 3.6 shows the elapsed time to process same query on same data.



```
| AAAAAAAAAAHMAAAA | 81                | 39.24
 2.74              |
| AAAAAAAAAAHNAAAA | 57                | 31.95
 12.78             |
| AAAAAAAAAAHPAAAA | 48                | 117.57
 55.39             |
| AAAAAAAAAAIAAAAA | 41                | 18.06
 4.33              |
| AAAAAAAAAAIBBAAA | 85.5              | 65.065
 20.11             |
| AAAAAAAAAAICAAAA | 82.5              | 70.72
 21.795            |
+------------------+-------------------+--------
-------------------+
Returned 100 row(s) in 3.61s
[master01.banglalinkgsm.com:21000] >
```

Figure 3.5. Q7 Executed on Hadoop Cluster

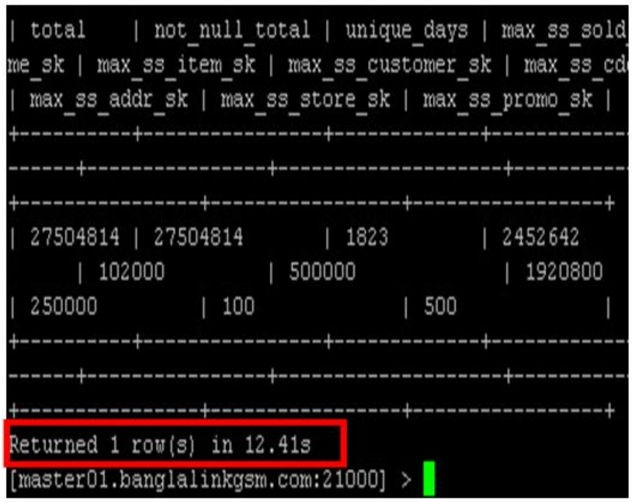


```
I_ITEM_ID              AGG1       AGG2       AGG3       AGG4
---------------- ---------- ---------- ---------- ----------
AAAAAAAAAAHDBAAA 30.3333333      34.12     110.55 24.6466667
AAAAAAAAAAHFBAAA         43      25.32          0        4.3
AAAAAAAAAAHGAAAA          2     101.82          0      52.94
AAAAAAAAAAHGBAAA         91      52.62      130.6      24.79
AAAAAAAAAAHIBAAA         57       2.72          0       2.39
AAAAAAAAAAHKAAAA         39     177.29          0     147.15
AAAAAAAAAAHMAAAA         81      39.24          0       2.74
AAAAAAAAAAHNAAAA         57      31.95      36.42      12.78
AAAAAAAAAAHPAAAA         48     117.57     340.26      55.39
AAAAAAAAAAIAAAAA         41      18.06          0       4.33
AAAAAAAAAAIBBAAA       85.5     65.065          0      20.11

99 rows selected.

Elapsed: 00:00:03.65
22:05:33 SQL>
```

Figure 3.6. Q7 Executed on Oracle DB

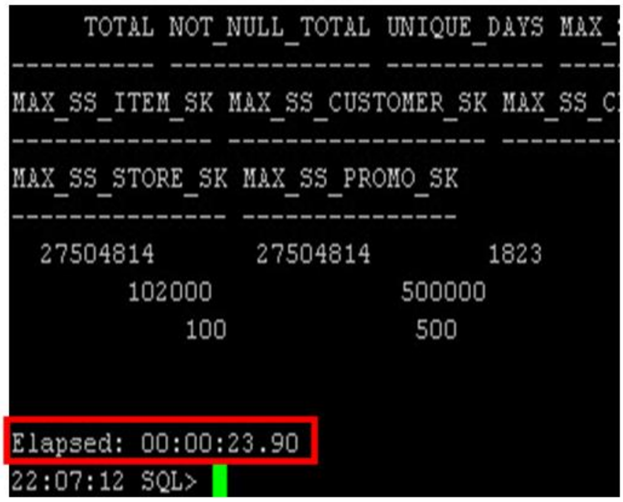
Query Example Three:

This query is collected from TPC benchmarking document which will stress the database system by accessing all table and will apply aggregation for in-depth analysis of the database. This type of query used to stress the database to test the performance. Figure 3.7 and Figure 3.8 shows the stress test timing of the database.



```
| total    | not_null_total | unique_days | max_ss_sold
me_sk | max_ss_item_sk | max_ss_customer_sk | max_ss_cd
| max_ss_addr_sk | max_ss_store_sk | max_ss_promo_sk |
+----------+----------------+-------------+-----------
------+---------------+--------------------+----------
+---------------+-----------------+----------------+
| 27504814 | 27504814       | 1823        | 2452642
      | 102000         | 500000             | 1920800
| 250000        | 100             | 500            |
+----------+----------------+-------------+-----------
------+---------------+--------------------+----------
+---------------+-----------------+----------------+
Returned 1 row(s) in 12.41s
[master01.banglalinkgsm.com:21000] >
```

Figure 3.7. Query Executed on Hadoop Cluster



```
     TOTAL NOT_NULL_TOTAL UNIQUE_DAYS MAX_
---------- -------------- ----------- ----
MAX_SS_ITEM_SK MAX_SS_CUSTOMER_SK MAX_SS_C
-------------- ------------------ --------
MAX_SS_STORE_SK MAX_SS_PROMO_SK
--------------- ---------------
  27504814       27504814          1823
        102000               500000
            100              500


Elapsed: 00:00:23.90
22:07:12 SQL>
```

Figure 3.8. Query Executed on Oracle

3. Observations

Analyzing given examples and query execution time, it has been observed that Hadoop cluster provide better performance than Oracle because of distributed and parallel processing architecture.

4. Conclusion

The purpose of this paper is to reduce datacenter implementation cost using commodity hardware and provide high performance. This paper also helps to explain about some new technology and framework which are open source; that can easily utilize those technologies for our complex data analysis which resembling structured, semi structured and non-structured data. Here we tried to demonstrate a performance comparison by executing some queries between distributed parallel computing system and traditional single computing system.

References

[1] A. B. M. Moniruzzaman and S. A. Hossain, “Nosql database: New era of databases for big data analytics-classification, characteristics and comparison”, International Journal of Database Theory and Application, vol. 6, no. 4, **(2013)**, pp. 199-214.

[2] H. Hu, Y. Wen, T. Chua and X. Li, “Towards Scalable Systems for Big Data Analytics”, A Technology Tutorial, **(2014)**.

[3] A. B. M. Moniruzzaman, “NewSQL: Towards Next-Generation Scalable RDBMS for Online Transaction Processing (OLTP) for Big Data Management. International”, Journal of Database Theory & Application, vol. 7, no. 6, **(2014)**.

[4] J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters”, Communications of the ACM, vol. 51, no. 1, **(2008)**, pp. 107-113.

[5] J. Dean and S. Ghemawat, “MapReduce: a flexible data processing tool”, Communications of the ACM, vol. 53, no. 1, **(2010)**, pp. 72-77.

[6] M. Zaharia, A. Konwinski, A. D. Joseph, R. H. Katz and I. Stoica, “Improving MapReduce Performance in Heterogeneous Environments”, OSDI, vol. 8, no. 4, **(2008)** December, pp. 7.

[7] H. C. Yang, A. Dasdan, R. L. Hsiao and D. S. Parker, “Map-reduce-merge: simplified relational data processing on large clusters”, Proceedings of the 2007 ACM SIGMOD international conference on Management of data, ACM, **(2007)** June, pp. 1029-1040.

[8] “Hadoop - Apache Software Foundation project home page”. [http://hadoop.apache.org/].

[9] C. Lam and J. Warren, “Hadoop in Action. Manning Publications”, **(2010)**.

[10] J. Venner, “Pro Hadoop”, New York: A Press, **(2009)**.

[11] T. White, “Hadoop: The Definitive Guide”, Sebastopol: O’Reilly Media, **(2009)**.

[12] A. Pavlo, E. Paulson, A. Rasin, D. J. Abadi, D. J. DeWitt, S. Madden and M. Stonebraker, “A comparison of approaches to large-scale data analysis. In SIGMOD ’09”, Proceedings of the 35th SIGMOD International Conference on Management of Data, New York, NY, USA, ACM, **(2009)**, pp. 165-178.

[13] M. Stonebraker, D. Abadi, D. J. DeWitt, S. Madden, E. Paulson, A. Pavlo and A. Rasin, “Mapreduce and parallel DBMSs: friends or foes”, Commun. ACM, vol. 53, no. 1, **(2010)**, pp. 64-71.

[14] S. Konstantin, “The hadoop distributed file system”, Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on IEEE, **(2010)**.

[15] B. Dhruba, “The hadoop distributed file system: Architecture and design”, Hadoop Project Website, vol. 11.2007, **(2007)**, pp. 21.

Authors

A K M Mahbubul Hossen, received his MSc degree from Daffodil International University and BSc degree from State University of Bangladesh in Computer Science and Information Technology Dhaka, Bangladesh 2007 and 2015 respectively. Currently he is working as an Assistant Manager of Banglalink Digital Communications Ltd. Technology Department, Bangladesh. Before that we was an IT officer at BRAC Bank Limited, Technology Department, Bangladesh. He is Microsoft Certified Technology and IT Specialist in Certifications *e.g*. MCSA, MCITP, MCSA, MCP, MCTS. Besides, He has completed many training programs in especially in networking and servers administrations areas. He has been experienced in planning, designing and implementing many network solutions in many different organizations. His research interest includes Cloud Computing, Cloud Applications, Big Data Management, Hadoop, MapReduce, Parallel and Distributed Computing, Clustering, High performance computing.

A B M Moniruzzaman Received his B.Sc (Hon’s) degree in Computing and Information System (CIS) from London Metropolitan University, London, UK and M.Sc degree in Computer Science and Engineering (CSE) from Daffodil International University, Dhaka, Bangladesh in 2005 and 2013, respectively. Currently he is working as a Lecturer of the department of Computer Science and Engineering ad Daffodil International University. His voluntarily works as reviewer of many international journals including IEEE, Elsevier, Springer, IGI-Global, InderScience. Besides, he has been serving as Technical Program Committee Members in many (over seventeen) different international Conferences. He has 12 (twelve) international publications including journals and conference proceedings. He is a member of IEEE since 2012. His research interests include Cloud Computing, Cloud Applications, Open-source Clouds, Cloud Management Platforms, Big Data Management, Agile Software Development, Hadoop, MapReduce, Parallel and Distributed Computing, Clustering, High performance computing, Distributed Databases, NoSQL Databases.

Prof. Dr. Syed Akhter Hossain is Post Doc-toral Fellow, Informatics and Systems Engineering, LIESP Laboratory, University Lyon2, Lyon, France. He received PhD degree, in Computer Science and Engineering from University Lumiere Lyon 2, France and M.Sc degree in Applied Physics and Electronics, First Class (First), and B.Sc (Hons) in Applied Physics and Electronics, First Class (First), Gold Medalist from Rajshahi University, Rajshahi, Bangladesh. Currently he is working as Professor and Head, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. He is a member of ACM, and member of IEEE. He has good numbers of publications in international referred journals and conference proceedings. He served organizing committee chair in the International Conference on Computer and Information Technology (ICCIT). His research areas includes simulation and modeling distributed system design and implementation signal and image processing internet and web engineering net-work planning and management database and data warehouse modeling.