

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/289442593>

# A Cost Effective Virtual Cluster with Hadoop Framework for Big Data Analytics

Article · December 2015

DOI: 10.14257/ijdt.2015.8.6.18

---

CITATIONS

0

---

READS

2,267

2 authors, including:



Seraj Al Mahmud Mostafa  
Montana State University

8 PUBLICATIONS 20 CITATIONS

SEE PROFILE

# A Cost Effective Virtual Cluster with Hadoop Framework for Big Data Analytics

Seraj Al Mahmud Mostafa<sup>1</sup> and A.B.M Moniruzzaman

*Member, IEEE,* <sup>2</sup> *Department of Computer Science and Engineering*  
*Daffodil International University*  
*Dhaka, Bangladesh*  
*sammbd@gmail.com*<sup>1</sup>  
*abm.moniruzzaman.eng@ieee.org*<sup>2</sup>

## Abstract

*Big data processing is currently becoming increasingly important research field in computer technology professionals due to the continuous growth of the amount of data generated by various fields. However, the processing of large-scale research data requires cluster technology infrastructure which causes huge investments for educational institutions. Hadoop is an open-source framework that allows for distributed storage and processing of very large data sets on computer clusters built from commodity hardware. This technology is being widely used for the analysis of large datasets. This paper focuses on proposes a low cost scalable hadoop virtual cluster platform and the performance of hadoop irtual cluster. We first describe the design and implementation of a virtual datacenter using hadoop framework. Then we perform a set of experiments to investigate the performance of virtual datacenter with standard datasets. For experiment and evaluation, the Cloudera's distribution of Apache Hadoop (CDH) is installed and configured on the DIU Cloud and Big Data Lab as a prototype implementation of a virtual data center and tested with 40 workstations. In this paper, we use TPC BENCHMARK™ DS” by-Transaction Processing Performance Council (TPC) benchmarks for evaluation virtual data center performance. The contributions of this paper, is to design model and implement a cost effective elastic virtual data center with hadoop framework and resource utilizations for educational institutions to provide high performance for distributed and parallel processing; as well as, Identify the bottlenecks of this systems e.g. bandwidth of network connectivity with all nodes in the lab.*

**Keywords-** *Hadoop; MapReduce; Virtual Cluster; Cloud Computing; Big Data*

## 1. Introduction

Big data [1-3] has recently received considerable attention to research field due to the continuous growth of the amount of data generated by various fields such as particle physics, human genomics, earth observation, etc.. How to compute, transfer, and store these huge data is a prominent challenge which will bring great impact on the traditional architectures and methods of computation, networking and storage. Therefore big data becoming important and cutting edge research field in higher studies; and dig data research labs are being introducing in higher educational institutions in recent years. However, if we consider computer labs, in many educational institutes, we find so many desktop and low-end servers which are utilize only 15% to 25%. Therefore, we can detect such hardware resources and use them to design a low cost virtual datacenter of distributed and parallel processing architecture, where we can process big amount of research data like structured, unstructured and semi-structured data.

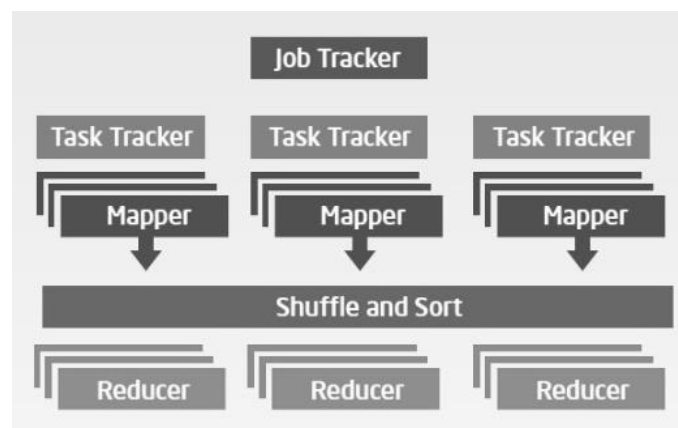
## 2. Related Technologies

### 2.1. Virtualization Technology

Virtualization technology is currently becoming increasingly popular as a core technology to implement the cloud computing paradigm. Many efforts have been made to study the performance characterization of virtualization, including performance evaluation, performance modeling, and performance optimization. Server consolidation is one of the most important application scenarios of virtualization to improve the resource utilization.

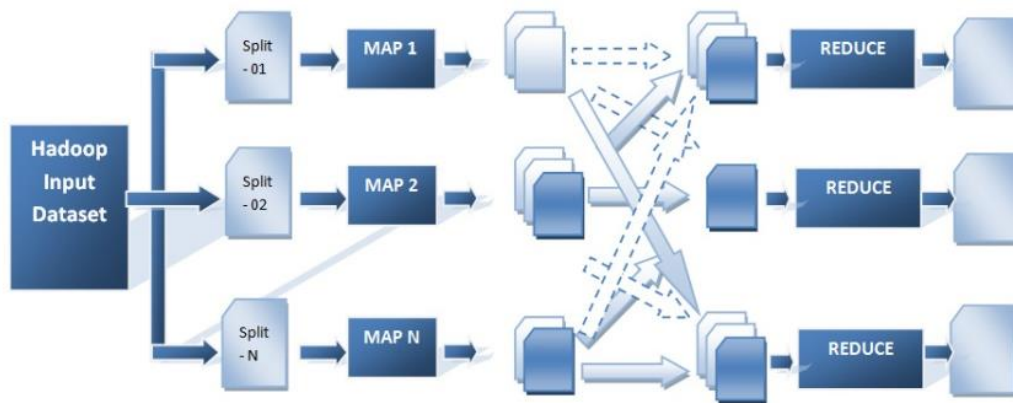
### 2.2. Hadoop

Apache Hadoop [4-6] is an open-source software framework written in Java for distributed storage and distributed processing of very large data sets on computer clusters. The system splits the input data-set into multiple chunks, each of which is assigned a map task that can process the data in parallel. Each map task reads the input as a set of (key, value) pairs and produces a transformed set of (key, value) pairs as the output. The framework shuffles and sorts outputs of the map tasks, sending the intermediate (key, value) pairs to the reduce tasks, which group them into final results. MapReduce uses JobTracker and TaskTracker mechanisms to schedule tasks, monitor them, and restart any that fail.



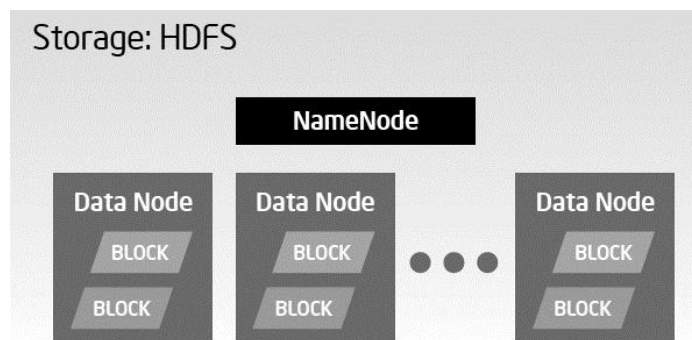
**Figure 1. Logical Architecture of Hadoop**

The Apache Hadoop platform also includes the Hadoop Distributed File System (HDFS) [7-8], which is designed for scalability and fault tolerance. HDFS stores large files by dividing them into blocks (usually 64 or 128 MB) and replicating the blocks on three or more servers. HDFS provides APIs for MapReduce [9-10] applications to read and write data in parallel. Capacity and performance can be scaled by adding Data Nodes, and a single NameNode mechanism manages data placement and monitors server availability.

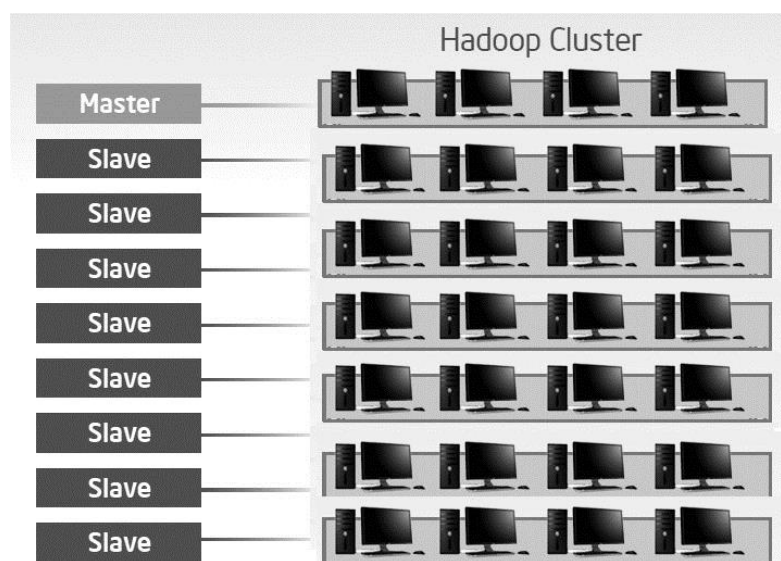


**Figure 2. Hadoop Process Flow**

HDFS clusters in production use today reliably hold petabytes of data on thousands of nodes. MapReduce, the programming paradigm implemented by Apache Hadoop, breaks-up a batch job into many smaller tasks for parallel processing on a distributed system HDFS, the distributed file system stores the data reliably shown in the Figure 1 and Figure 2.



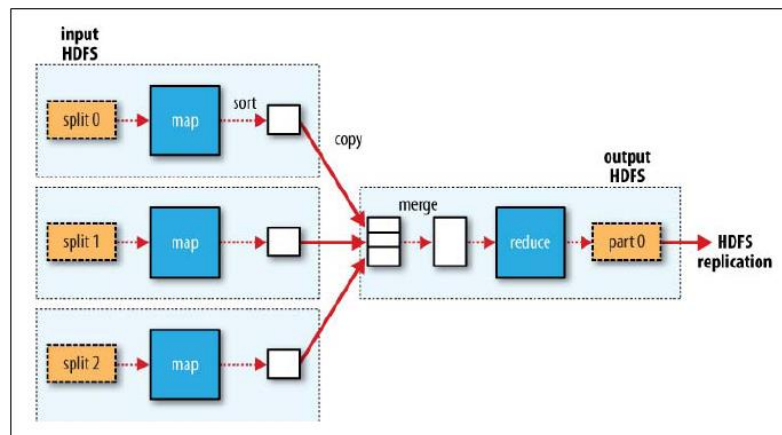
**Figure 3. HDFS Storage**



**Figure 4. Physical Architecture of Hadoop Cluster**

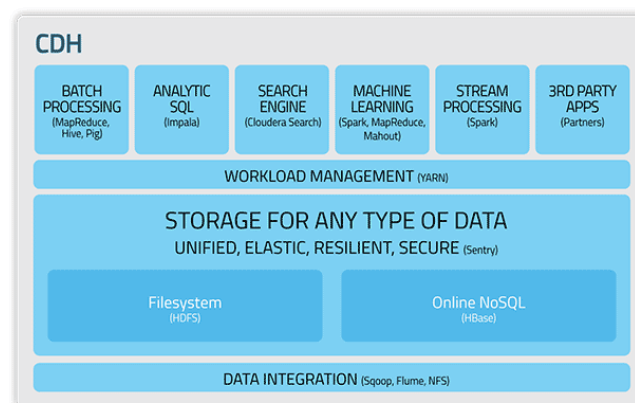
## 2.3. MapReduce

MapReduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster. A MapReduce program is composed of a Map() procedure that performs filtering and sorting (such as sorting students by first name into queues, one queue for each name) and a Reduce() procedure that performs a summary operation (such as counting the number of students in each queue, yielding name frequencies).



**Figure 5. MapReduce Data Flow with a Single Reduce Task [6]**

MapReduce optimized the hadoop provisioning in the cloud to reduce the cost and improve the performance, and compared the performance of hadoop cluster on virtual machines and physical machines and found that running MapReduce application on virtual machines incurs additional performance degradation compared to the case that running on physical machines.



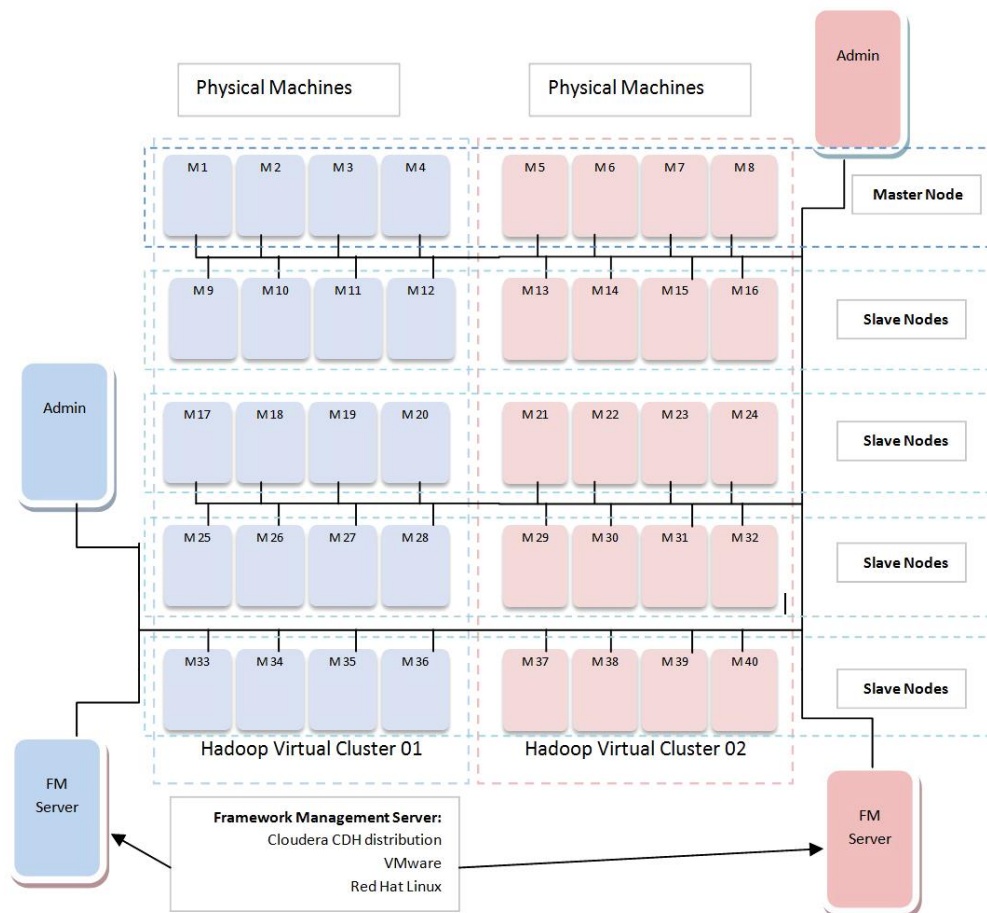
**Figure 6. CDH (Cloudera Distribution Including Apache Hadoop) Key Elements**

Cloudera's open-source Apache Hadoop distribution [11], CDH (Cloudera Distribution Including Apache Hadoop) [12-13] is the only Hadoop solution to offer unified batch processing, interactive SQL, and interactive search, and role-based access controls. CDH delivers the core elements of Hadoop – scalable storage and distributed computing – along with additional components such as a user interface.

### 3. Implementation of Hadoop Virtual Cluster

#### 3.1. Infrastructure Design

We design and with a small lab with 40 workstations and we separate logically two group and they have 20 workstations in each site. Now design a virtual datacenter using these 40 workstations and additional two high-end server. 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 a virtual datacenter infrastructure with distributed and parallel processing power.



**Figure 7. Infrastructure Design of a Virtual Cluster With Hadoop Framework**

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.

In the infrastructure design in Figure 7 where two virtual cluster are connected with each other through private network. Each cluster has LAN connection and all the

workstations are connected to LAN. A high-end server has been placed to each cluster connected to LAN.

### 3.2. Environment Setup

**3.2.1. Installation and Configuration:** 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. Please find the installation steps and command as below.

#### Prepare Master / NameNode

- Install Red Hat Enterprise Linux 64 bit operating system
- Configure IP
- Disable firewall
- Disable selinux
- Configure hosts file
- Configure NTP
- Configure SSH (Password less authentication)
- Install Java and other prerequisite
- Install Postgresql Database
- Install Cloudera Manager CDH5.2.0 (Hadoop Framework)
- Login to Cloudera Manager console
- Configure cluster in Cloudera Manager Configure services as required

#### Prepare Slave / Data Node

- Install Red Hat Enterprise Linux 64 bit operating system
- Configure IP
- Disable firewall
- Disable selinux
- Configure hosts file
- Configure NTP
- Configure SSH (Password less authentication)
- Install Java and other prerequisite
- Login to Cloudera Manager console
- Configure data node to the cluster
- Configure services as required

1	ConFigure IP Address (Master and Slave) setup	vim/etc./sysconfig/network-scripts/ifcfg-eth, ONBOOT=yes, service network restart
2	Stop Firewall (Master and Slave)	chkconfig --list iptables chkconfig --level 0123456 iptables off service iptables stop service iptables status
3	Disableselinux (Master and Slave)	sestatus vim /etc./sysconfig/selinux SELINUX=disabled Init 0
4	ConFigure Hostname	vim /etc./sysconfig/network

		<pre> NETWORKING=yes HOSTNAME= XXXXXXXXX GATEWAY= XXX.XXX.XXX.XXX vim /etc./hosts IP&lt;XXX.XXX.XXX.XXX&gt; FQDN&lt;XXXX.XXXXXX.XXX&gt; </pre>
5	<b>ConFigure NTP Service (Master and Slave)</b>	<pre> cd /etc. catntp.conf  more vintp.conf  more cd /etc./init.d service ntpd status service ntpd start ntpq -p chkconfig --list --level 345  grepntp chkconfigntpd on --level 345 chkconfig --list --level 345  grepntpd </pre>
6	<b>ConFigure SSH for Password less login (Master and Slave)</b>	<pre> login to slave01: ssh-keygen -t rsa cd /root/.ssh ls Login to master01: ssh-keygen -t rsa cd /root/.ssh ls cat id_rsa.pub &gt;&gt;authorized_keys scpauthorized_keys root@slave01:/root/.ssh/ login to slave01: cd /root/.ssh ls cat id_rsa.pub &gt;&gt;authorized_keys scpauthorized_keys root@master01:/root/.ssh/ Test: ssh slave01 (able to login without password) ssh master02 (able to login without password) </pre>
7	<b>Install Postgresql8.4 Database (Master Server)</b>	<pre> Create folder and copy jre and postgresql software to that folder: chmod 777 jre-7u1-linux-x64.tar.gz tar -zxvf jre-7u1-linux-x64.tar.gz cd / host to export java path: export PATH=\$PATH:/soft/jre1.7.0_01/bin/ chmod 777 postgresql84-8.4.22-1PGDG.rhel6.x86_64.rpm chmod 777 postgresql84-libs-8.4.22-1PGDG.rhel6.x86_64.rpm chmod 777 postgresql84-server-8.4.22-1PGDG.rhel6.x86_64.rpm rpm -Uvh postgresql84-libs-8.4.22-1PGDG.rhel6.x86_64.rpm rpm -Uvh postgresql84-8.4.22-1PGDG.rhel6.x86_64.rpm rpm -Uvh postgresql84-server-8.4.22-1PGDG.rhel6.x86_64.rpm </pre>
8	<b>Check services and logs after installation (Master Server)</b>	<pre> service postgresql-8.4 status service postgresql-8.4 initdb service postgresql-8.4 start chkconfig --level 345 postgresql-8.4 on chkconfig --list  greppostgre* </pre>
9	<b>ConFigure yum.conf file if proxy used for internet (Master and Slave)</b>	<pre> vim /etc./yum.conf proxy=http://XXX.XXX.XXX.XXX:XXXX </pre>
10	<b>DownloadCloudera manager (Master Server)</b>	<pre> cd /root wget http://archive.cloudera.com/cm5/installer/latest/cloudera- manager-installer.bin ls-l </pre>
11	<b>Install Cloudera Manager (Master Server)</b>	<pre> Cd /root ls -l -rw-r--r-- 1 root root 501703 Jul 29 10:51 cloudera-manager- installer.bin chmod 777 cloudera-manager-installer.bin ./cloudera-manager-installer.bin </pre>



After this step cloudera manager will be installed on the Master server and go the following link (<http://XXX.XXX.XXX.XXX <IP>:7180/>) to install and conFigure Hadoop cluster.

1	<b>Check services after installation (Master Server)</b>	cd /var/log/cloudera-manager-installer less 6.start-scm-server.log (check OK) less 5.start-embedded-db.log (check OK) chkconfig --level 345 cloudera-scm-server-db on chkconfig --level 345 cloudera-scm-server on
2	<b>Yum configuration (All Slave Servers)</b>	cd /etc./yum.repos.d/ vim server_rhel6.repo  ----- [local_mirror_server_rhel6] name=patch_mirror_server_rhel6 baseurl=ftp://XXX.XXX.XXX.XXX/repo/rhel-x86_64-server-6/getPackage enabled=1 gpgcheck=0 -----  yum clean all yumrepolist yum list all
3	<b>Install dependent services (All Slave Servers)</b>	yum install cyrus-sasl-gssapi yum install redhat-lsb yum install portmap servicercbind status servicercbind start
4	<b>ConFigure Hadoop Cluster (Cloudera Manager Console)</b>	Go the following link ( <a href="http://XXX.XXX.XXX.XXX &lt;IP&gt;:7180/">http://XXX.XXX.XXX.XXX &lt;IP&gt;:7180/</a> ) to install and conFigure Hadoop cluster that is showing in the Figure 2.6 Default User ID and Password is admin, admin

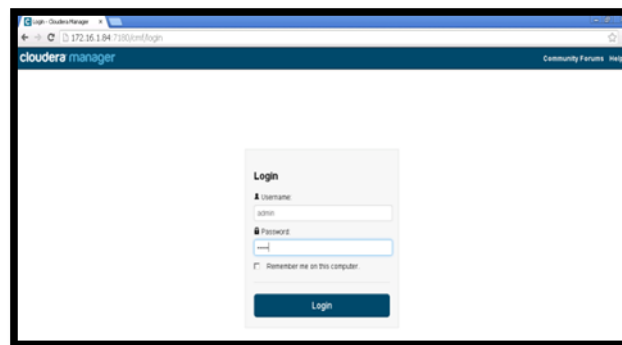


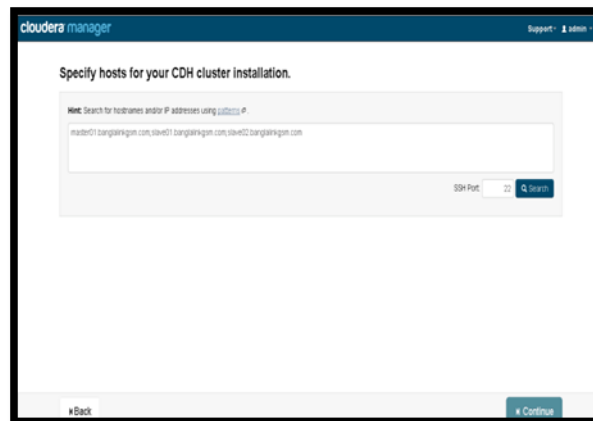
Figure 8. Cloudera Manager Console to Login



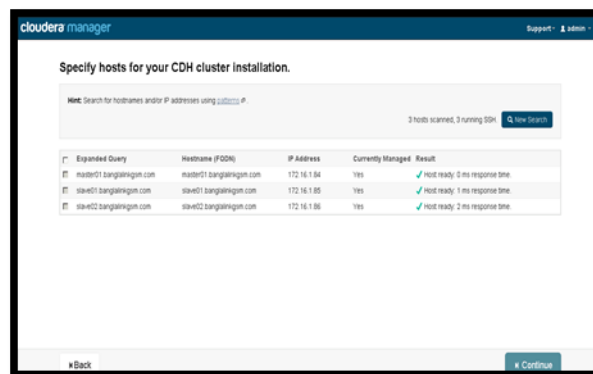
Figure 9. Cloudera Manager Version Selection

Figure 8 showing to click on just free installation to choose the Cloudera Manager version.

Figure 10 and Figure 11 showing the how to add host to create a cluster on Cloudera manager. Add hostnames as defined in on `/etc./hosts` file of all servers.



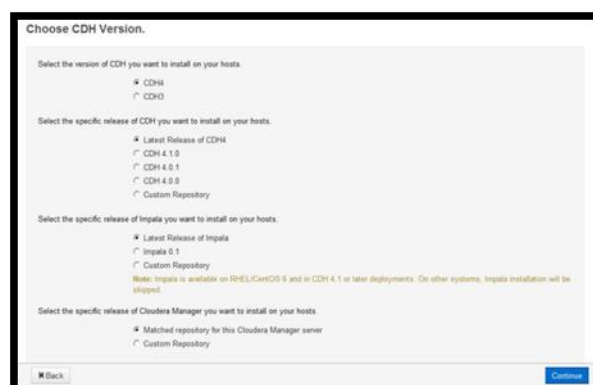
**Figure 10. Cluster Installation**



Expanded Query	Hostname (FQDN)	IP Address	Currently Managed	Result
master01.banglalink.com	master01.banglalink.com	172.16.1.84	Yes	✓ Host ready: 0 ms response time.
slave01.banglalink.com	slave01.banglalink.com	172.16.1.85	Yes	✓ Host ready: 1 ms response time.
slave02.banglalink.com	slave02.banglalink.com	172.16.1.86	Yes	✓ Host ready: 2 ms response time.

**Figure 11. Specify Host to Install Cluster**

Figure 11 showing the hosts availability, now select all the available hosts by checking the text box against it. Figure 12 shows the CDH version selection where we need to click on CDH4 cloudera manager latest version of Hadoop.



**Figure 12. CDH Version Selection**

**Figure 13. CDH Installation Credential Option**

Figure 13 shows the installation credential option where we need to click on start installation button to start the installation process on selected three nodes. Figure 14 shows this process is successfully completed click on continue button and that will run host inspector to check the host for correctness. After the inspector is finished click on continue

**Figure 14. Inspect Hosts for Correctness**

**Figure 15. Services Selection Option**

Figure 15 showing service we want to install on each host for CDH4

Figure 16 showing the cluster service installing steps where Cloudera manager is installing services on the nodes.

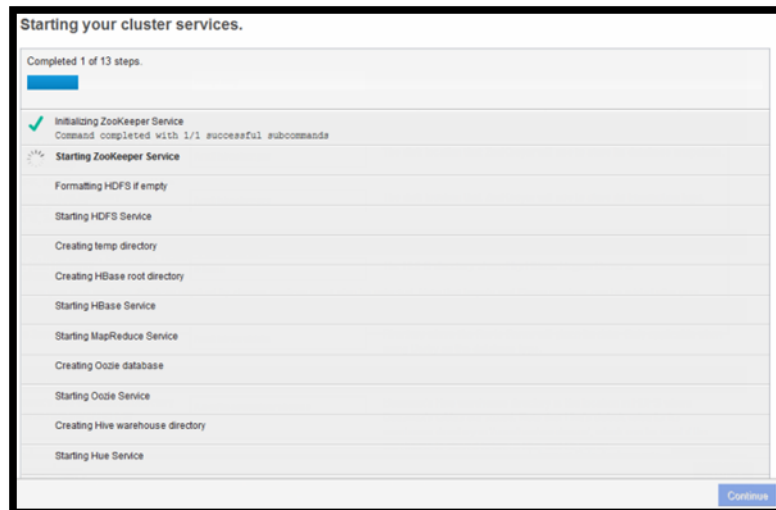


Figure 16. Cluster Service Installation

Figure 17 shows the Cloudera Manager Dashboard login in to the console through default id and password *i.e.* admin, admin respectively. Below screen shot shows various services of Hadoop running on the three nodes and managed by cloudera manager.

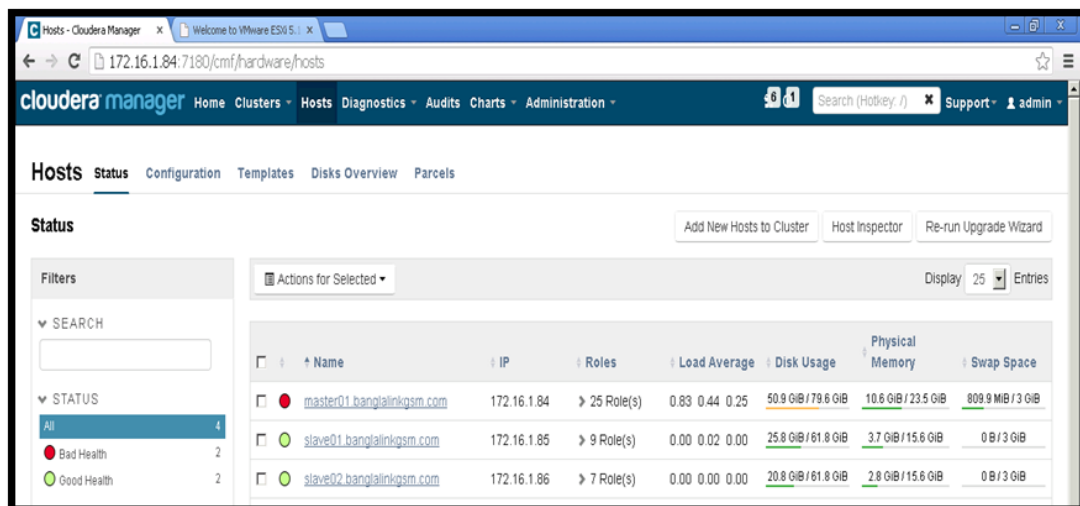


Figure 17. Cloudera Manager Dashboard

In this stage, Hadoop distribution as a Cloudera Manager CDH completely install and Hadoop cluster ready in the lab with 40 workstations for data analytics.

#### 4. Performance Evaluation

With the help of TPC guideline we design Database on virtual hadoop cluster system. Generate three sets of data and table on this system. We uses TPC-DS provided Data Generator tools to generate benchmarking standard dataset.

Here Figure 3.1 shows the same dataset on a table “**store\_sales**” on of the database Hadoop. 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 18. Hadoop Dataset**

#### 4.1. 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 of the systems.

```
| 1999 | 5004001 | edu packsolar #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 | importtoedu pack #2 | 95701.84
+-----+
Returned 100 row(s) in 6.05s
[master01.banglalinkgsm.com:21000] >
```

**Figure 19. Q3 executed on Hadoop Cluster**

#### 4.2. 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 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		
AAAAAAAAAATAAAAA	41	18.06
4.33		
AAAAAAAAAATBAAAA	85.5	65.065
20.11		
AAAAAAAAAATCAAAA	82.5	70.72
21.795		
+-----+		
-----+		
Returned 100 row(s) in 3.61s		
[master01.banglalinkgsm.com:21000] >		

Figure 20. Q7 Executed on Hadoop Cluster

#### 4.3. 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 21. Query Executed on Hadoop Cluster

Here there set of experiments to investigate the performance of virtual datacenter with standard datasets and observe that data analytics easily perform in the hadoop clusters with three datasets in three different execution times. Here network bandwidth is the vital point for smooth operation of virtual cluster as well as bottlenecks of virtual cluster performance.

## 5. Conclusion

Big data processing is currently becoming increasingly important research field in computer technology professionals due to the continuous growth of the amount of data generated by various fields. However, the processing of large-scale research data requires cluster technology infrastructure which causes huge investments for educational institutions. Hadoop is an open-source framework that allows for distributed storage and processing of very large data sets on computer clusters built from commodity hardware. This technology is being widely used for the analysis of large datasets. This paper focuses on proposes a low cost scalable hadoop virtual cluster platform and the performance of hadoop virtual cluster. We first describe the design and implementation of a virtual datacenter using hadoop framework. Then we perform a set of experiments to investigate the performance of virtual datacenter with standard datasets. For experiment and evaluation, the Cloudera's distribution of Apache Hadoop (CDH) is installed and configured on the DIU Cloud and Big Data Lab as a prototype implementation of a virtual data center and tested with 40 workstations.

We design Database on virtual hadoop cluster system and generate three sets of data and table on this system. Here we tried to demonstrate a performance by executing these sets of queries over the distributed parallel computing, virtual hadoop cluster system. For the simulation of the infrastructure Hadoop cluster has been used for distributed parallel processing. In according to benchmarking standard, we use TPC BENCHMARK™ DS" by-Transaction Processing Performance Council (TPC). The contributions of this paper, is to design model and implement a cost effective elastic virtual data center with hadoop framework and resource utilizations for educational institutions to provide high performance for distributed and parallel processing; as well as, Identify the bottlenecks of this systems *e.g.* bandwidth of network connectivity with all nodes in the lab.

## References

- [1] M. James, "Big data: The next frontier for innovation, competition, and productivity", (2011).
- [2] M. Andrew, "Big data", the management revolution. Harvard Bus Rev, vol. 90, no. 10, (2012), pp. 61-67.
- [3] Z. Paul and C. Eaton, "Understanding big data: Analytics for enterprise class hadoop and streaming data", McGraw-Hill Osborne Media, (2011).
- [4] <https://hadoop.apache.org/>
- [5] [https://en.wikipedia.org/wiki/Apache\\_Hadoop](https://en.wikipedia.org/wiki/Apache_Hadoop)
- [6] W. Tom, "Hadoop: The definitive guide", O'Reilly Media, Inc., (2012).
- [7] S. Konstantin, "The hadoop distributed file system", Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on. IEEE, (2010).
- [8] B. Dhruva, "The hadoop distributed file system: Architecture and design", Hadoop Project Website, 11.2007, (2007), pp. 21.
- [9] D. Jeffrey and S. Ghemawat, "MapReduce: simplified data processing on large clusters", Communications of the ACM, vol. 51, no. 1, (2008), pp. 107-113.
- [10] D. Jeffrey and S. Ghemawat, "MapReduce: a flexible data processing tool", Communications of the ACM, vol. 53, no. 1, (2010), pp. 72-77.
- [11] <http://www.cloudera.com/content/cloudera/en/home.html>
- [12] <http://www.cloudera.com/content/cloudera/en/products-and-services/cdh.html>
- [13] M. Hossen, A. K. M. Moniruzzaman, A. B. M. and Hossain S. A., "Performance Evaluation of Hadoop platform and Oracle for Distributed parallel Processing in Big Data Environments arXiv preprint arXiv", (2015).

## Authors



**Seraj Al Mahmud Mostafa**, is an M. Sc. graduate in Computer Science and Engineering from Lulea University of Technology, Sweden. He is currently working as a lecturer at Daffodil International University, Dhaka, Bangladesh. Besides this, he is involved in a Data Mining and Big data analysis research project at this university under Computer Science research team. His research papers and technical reports can be found in IEEE explore and the Lulea University research database. He had also been in the organizing committee for two conferences in Lulea University of Technology, Sweden. His research areas include Big Data Analytics, Hadoop, Data mining, Open-source Clouds, Cloud Management Platforms, Data Engineering and Management, and privacy issues for online users.



**A B M Moniruzzaman**, is a PhD researcher at Jahangirnagar University; he 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 at 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 nineteen) different international Conferences. He has 13 (thirteen) international publications including journals and conference proceedings. He is a member of IEEE since 2012. He has been serving as a counselor DIU IEEE STUDENT BRANCH. His supervision project 'Solar Car' has received international award and funded by ICT innovation fund by Govt. of People's Republic of Bangladesh. 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.



