An Experimental Approach Towards Big Data for Analyzing Memory Utilization on a Hadoop cluster using HDFS and MapReduce.

Amrit Pal
Stdt, Dept of Computer Engineering and Application,
National Institute of Technical Teachers' Training and Research
Bhopal, India
er.amritkabir@gmail.com

Sanjay Agrawal
Dean R&D, Prof, Dept of Computer Engineering and Application,
National Institute of Technical Teachers' Training and Research
Bhopal, India
sagrawal@nitttrbpl.ac.in

***Abstract*—When the amount of data is very large and it cannot be handled by the conventional database management system, then it is called big data. Big data is creating new challenges for the data analyst. There can be three forms of data, structured form, unstructured form and semi structured form. Most of the part of bigdata is in unstructured form. Unstructured data is difficult to handle. The Apache Hadoop project provides better tools and techniques to handle this huge amount of data. A Hadoop distributed file system for storage and the MapReduce techniques for processing this data can be used. In this paper, we presented our experimental work done on big data using the Hadoop distributed file system and the MapReduce. We have analyzed the variable like amount of time spend by the maps and the reduce, different memory usages by the Mappers and the reducers. We have analyzed these variables for storage and processing of the data on a Hadoop cluster.**

Keywords*—SLOTS_MILLIS_MAPS; HDFS; MapReduce; Name Node; Secondary NameNode; Data Node; SLOTS_MILLIS_REDUCES;

I. Introduction

Data are tokens which can be interpreted or converted in some kind of information or values. These values of data can be quantified or can be qualitative. Further, these can be interpreted as qualitative and quantitative facts. Data can be converted into values or variables, then interpret them into some information [1]. The origin of data could be anything like the logs, social communication, sensors etc. all is generating data. Data is required to be stored. The storage of data can be done on some storage media like disk. The storage of the data should in such manner that it can be retrieved in an effective manner. By using the word efficient we means that it should take less time, Less CPU instruction, Less bandwidth, etc. the sources of data are different and heterogenous. Due to that a data can be in form of the structured, semi structured or unstructured. It is an urgency to manage all these types of data. As mentioned before there can be different sources of data. The source of data has a direct affect upon the storage of the data. Big data is a big challenge for the data analysts. The different aspect of the big data makes it difficult to manage. Big data require speed for its processing. This huge amount of data requires fast information retrieval techniques that can retrieve data from this huge amount. There are different tools available for handling big data. Most of them use a distributed storage for storing the data and for processing the data uses parallel computing techniques. Hadoop provides the solution for the big data. The Hadoop distributed file system is an efficient system for providing storage to the big data. Yahoos' MapReduce provides terminology for processing the data. Apache Hadoop uses both hadoop distirbuted file system and the map reduce. HDFS stores data on the different nodes. This storage is in the form blocks. The default size of the block is 64 MB. The Hadoop system consists of the Name Node, Secondry Node, Data Node, Job Tracker, Task Tracker. Name Node work as a centralized node in the big data setup. Any request for the retrieval of information pass through the Name Node. There can be two types of setup the Hadoop. One is the single Node setup, Multi Node setup. In case of the first all the component of the Hadoop will be on the same node and in case of the second component can be the different nodes. The paper is divided into five sections. The first section is the introduction section, the second is the related work done in this area. The third section is the experimental setup that we have for performing those experiments. The fourth section shows the result that we have obtained during our experiments. Fifth is the conclusion and the recommendation that can be taken care in establishing a Hadoop cluster.

II. Related Work

Cloud computing has different application of big data with it. Data in cloud computing as different issues [2] for management. Aggregation of data is very important [3]. Data is generated by different resources. Social network is very big source of big data. Google map reduce and Hadoop Distributed file system provide efficient distributed storage and parallel processing for big data [4]. Data can be provided as a service by using big data in Cloud Computing [5].

978-1-4799-3486-7/14/$31.00©2014 IEEE

Hadoop can be used for big data management [6]. Different types of analysis on the big data hadoop are done. Tera sort and the teragen programs can provide the performance benchmarks of a hadoop cluster. Teragen can provide the performance benchmarks for storage of the data and the terasort can provide the processing benchmark of the bigdata storage on Hadoop.

III. Experimental Setup

We have set up a testbed for running our experiment. This testbed contains five numbers of nodes. The configuration of the node is given in the table1 given below.

Table 1. System configurations

System	Dell
RAM	4GB
Disk	30 GB
Processor	Intel(R) Xeon(R) CPU W3565 @3.20GHz 3.20GHz
CPU	64 bit
Operating System	Ubuntu 12.04
Installation Process	Wubi installer
Hadoop	Hadoop-1.2.1-bin.tar [8]
Java	Java OpenJdk 6
IP addresses	Class B address

Jps command outputs shows that all components of the nodes are running the sample output of the command is shown in the figure 2. The interface can be accessed from http//:localhost:50070. This page shows that the name node is running. Through this we can also navigate through the file system. On this page the information about the name node and the storage available on the different datanode can be accessed. Domain names must be configured or set for accessing this page. Class B addresses are used for the nodes. The IP address arrangement for the system is shown in the figure 1. The subnet mask used is 255.255.255.0.

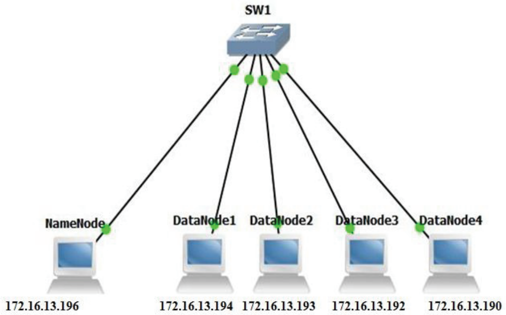
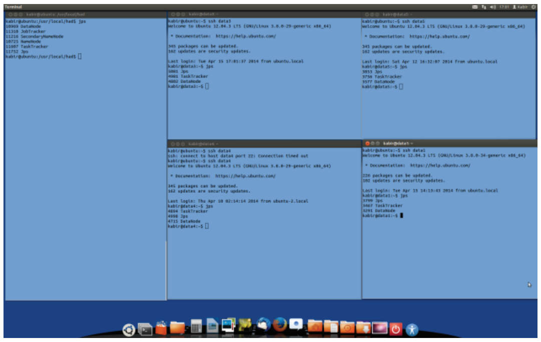


Figure 1 Network Architecture

We started our experiment with having the name node and the secondary name node on the same system and the datanode also on the same node. As we proceed further we have added a datanode each time in our experiments.



```
kabir@ubuntu:/usr/local/had$ jps
10969 DataNode
11310 JobTracker
11216 SecondaryNameNode
10725 NameNode
11607 TaskTracker
11752 Jps
kabir@ubuntu:/usr/local/had$
```

```
kabir@ubuntu:~$ ssh data3
Welcome to Ubuntu 12.04.3 LTS (GNU/Linux 3.8.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

345 packages can be updated.
162 updates are security updates.

Last login: Tue Apr 15 17:01:37 2014 from ubuntu.local
kabir@data3:~$ jps
5081 Jps
4981 TaskTracker
4802 DataNode
kabir@data3:~$
```

```
kabir@ubuntu:~$ ssh data5
Welcome to Ubuntu 12.04.3 LTS (GNU/Linux 3.8.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

345 packages can be updated.
162 updates are security updates.

Last login: Sat Apr 12 16:32:07 2014 from ubuntu.local
kabir@data5:~$ jps
3853 Jps
3750 TaskTracker
3577 DataNode
kabir@data5:~$
```

```
kabir@ubuntu:~$ ssh data4
ssh: connect to host data4 port 22: Connection timed out
kabir@ubuntu:~$ ssh data4
Welcome to Ubuntu 12.04.3 LTS (GNU/Linux 3.8.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

345 packages can be updated.
162 updates are security updates.

Last login: Thu Apr 10 02:14:14 2014 from ubuntu-2.local
kabir@data4:~$ jps
4894 TaskTracker
4998 Jps
4715 DataNode
kabir@data4:~$
```

```
kabir@ubuntu:~$ ssh data1
Welcome to Ubuntu 12.04.3 LTS (GNU/Linux 3.8.0-34-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

226 packages can be updated.
102 updates are security updates.

Last login: Tue Apr 15 14:13:43 2014 from ubuntu.local
kabir@data1:~$ jps
3799 Jps
3467 TaskTracker
3291 DataNode
kabir@data1:~$
```

Figure 2 Jps output

While doing the setup of the cluster we carefully set the parameters of the different files of the Hadoop. These files are Hadoop-env.sh, mapred-site.xml, core-site.xml. These files are having the information about the namenode and the datnode. In case when the number of nodes are greater than one, then for setting the namenode for each node we set the values in the slave file on the name node. With the help of the slve file the name node can identify the datanode which are configured with that name node. This is the setup that we have used for our experiments.

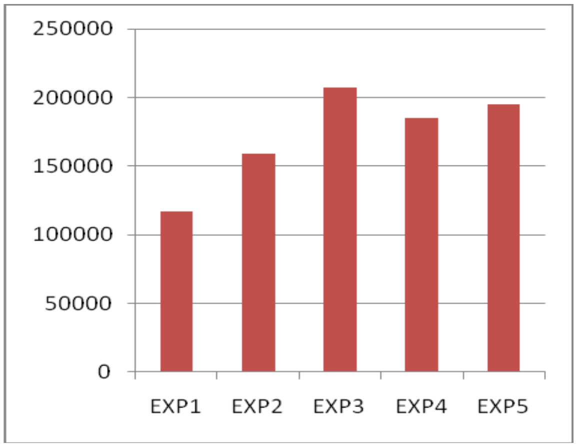
IV. Results

We worked in such a manner that, in every iteration, we will increase the amount of processing power and the amount storage area available for storing the data. Two programs are used. One program is used for storing the data and one for processing the data. There are different types of memory parameters used by the Hadoop. These parameters are the virtual memory, heap usage, physical memory. The information about these memories are stored in the forms of the snapshots. We have collected and combine the information for the snapshots of the different types of memory. We have analyzed different parameters of the Hadoop distributed file system and the map reduce tasks. The first parameter is the amount of time spend by the maps in an occupied slot. It is represented by SLOTS_MILLIS_MAPS [7]. It is measured in milliseconds. The table 2 below shows the amount of time spent by the maps in slots with different numbers of nodes.

Table 2. Time spent in a slot

SLOTS_MILLIS_MAPS	
Experiment No	Value
EXP1	117015
EXP2	159016
EXP3	206981
EXP4	185059
EXP5	195332

The time as shown in the graphical output increases as increase in the number of nodes in the.



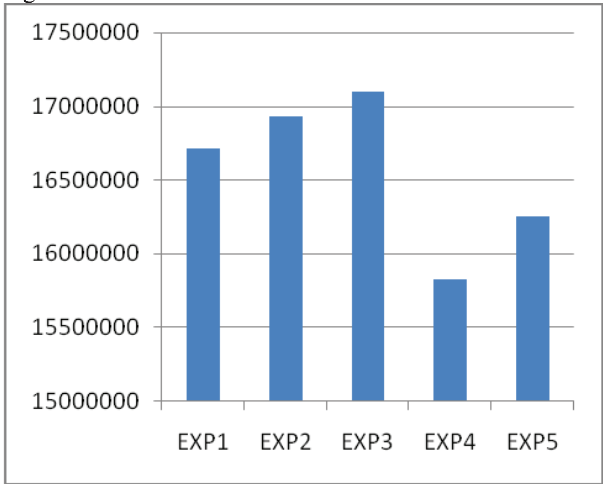
Graph 1 Time spent in a slot

The second parameter that we have analyzed is Physical memory snapshot.These snapshots are taken in bytes. These snapshots are automatically taken by the Hadoop. These snapshots outputs come in concluded form, and a value for each job id submitted by the user. Table 3 shows the snapshot for generating the data on the cluster.

Table 3. Data generating snapshot of physical memory

Physical memory (bytes) snapshot	
Experiment No	Value
EXP1	167092224
EXP2	169299968
EXP3	170983424
EXP4	158232576
EXP5	162500608

Graph 2 shows the behavior of the system's physical memory by taking its snapshots on a regular basis one for each job. There is variation in the amount of physical memory. Although these snapshots can also be accessed manually by using the URL of the node name.



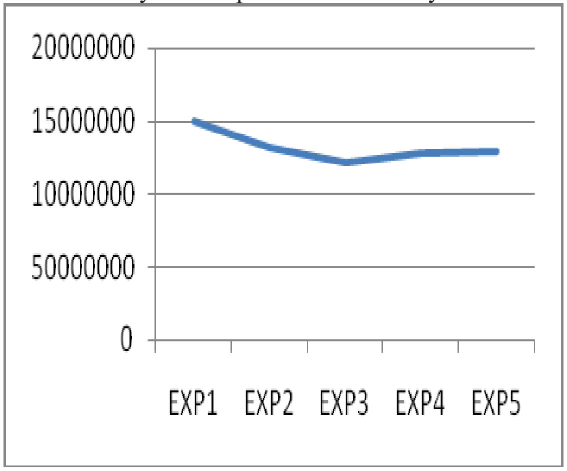
Graph 2 System physical memory

The HADOOP_HEAPSIZE is a parameter whose value can be set in the conf/hadoop.env.sh file. The size is in MB. The default value of this variable is the 100 MB. While performing the experiment we have also analyzed the values of this parameter. The values are in bytes. The amount of heap size used by our different experiment is shown in the table 4.

Table 4. Heap Usage

Total committed heap usage (bytes)	
Experiment No	Value
EXP1	149422080
EXP2	131530752
EXP3	121438208
EXP4	127467520
EXP5	128712704

The graphical output of the total committed heap usage is shown in the graph 3. The amount is in bytes. The Hadoop mapreduce task dy default produce result in bytes.



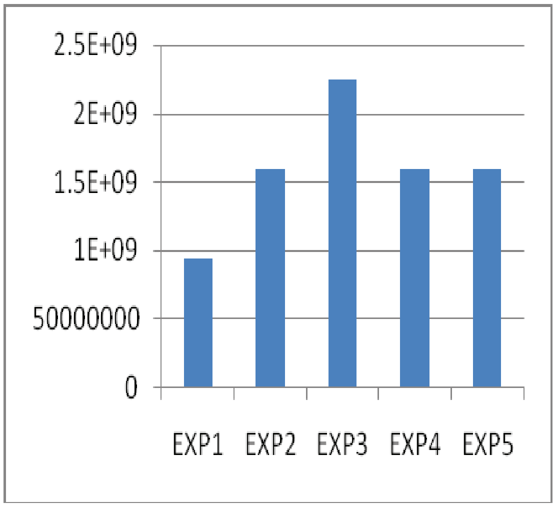
Graph 3. Heap Usage

For monitoring the memory usage Task tracker can be configured. If the tasks behave abruptly then they can consume large amount of memory. So, the monitoring of the different types of memory must be done carefully. With monitoring all these types of memory the task can be provided with the amount of memory they can be used. Our next parameter that we have analyzed is the amount of virtual memory that is used by our experiments.

Table 5. Virtual Memory snapshot

Virtual memory (bytes) snapshot	
Experiment No	Value
EXP1	942039040
EXP2	1595789312
EXP3	2245279744
EXP4	1591631872
EXP5	1593749504

Graph 4 shows the virtual memory snapshots. The amount of memory varies a little and remains around a constant value. As the number of nodes increases in the cluster, the amount of virtual memory increases for first three experiments then it remains slightly different.



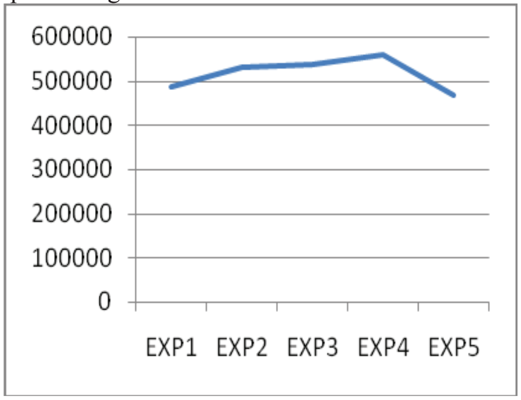
Graph 4. Virtual Memory snapshot

There are two types of output that are generated by our experiments. The output that is already discussed are for storing the data on the cluster. The second type of the output that we have is for processing the data. The first parameter that we have discussed is the SLOTS_MILLIS_MAPS which is the amount of time spent by the maps in slots with different numbers of nodes for processing the data.

Table 6. Time spent in slot for processing data

SLOTS_MILLIS_MAPS	
Experiment No	Value
EXP1	486850
EXP2	528451
EXP3	535697
EXP4	557874
EXP5	467145

Graph 5 shows the behaviour of the SLOTS_MILLIS_MAPS with increasing number of nodes in the cluster. This output is for the processing of the data.



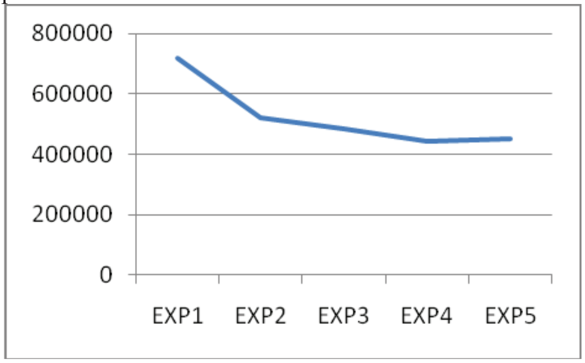
Graph 5. Time spent in slot for processing data

Now in case of the generation of the data or the storage of the data on the Hadoop cluster the amount of reducers required by the system was equal the zero. This is because the storage of data on the Hadoop cluster requires only the mappers. So there were no reducers. But in case of the processing of the data we need the reducers. We have even seprately specified the number of reducer. They tell about the amount of time spent by the reducer in a given slot. This time is also in milliseconds.

Table 7. Reducer time in a slot

SLOTS_MILLIS_REDUCES	
Experiment No	Value
EXP1	720142
EXP2	521457
EXP3	482919
EXP4	442648
EXP5	452687

Table 6 provides the amount of time spent by the reducer in a slot for a experiment. The behavior is graphically shown in the graph 6.



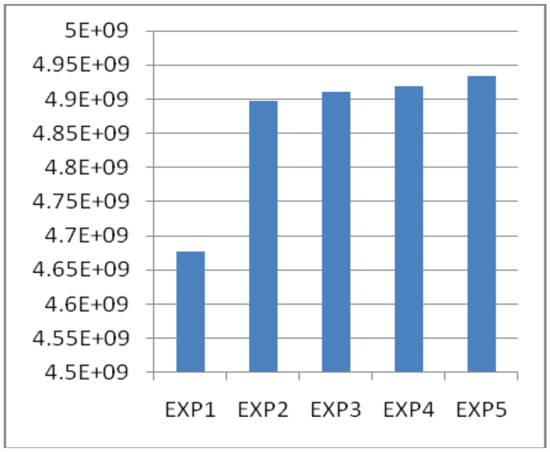
Graph 6. Reducer time in a slot

Next we have analyzed the physical memory snapshot parameter. This parameter stores the value of the physical memory snapshot. We have analyzed this parameter for our each experiment. The amount of data or the memory snapshot taken are shown in the table 8.

Table 8. Physical memory snapshot for processing data

Physical memory (bytes) snapshot	
Experiment No	Value
EXP1	4676485120
EXP2	4896800768
EXP3	4911095808
EXP4	4918480896
EXP5	4934275072

The physical memory snapshots are taken statically in this manner, but the monitoring can be done with the help of the interface of the Name Node. These statistics are also available at each datanode on the datanode interface. They can be referred from there also. The port numbers used for these processes are the default port numbers. These port numbers are defined in the mapred and the core-site.xml file. Proper mapping of the Datanode to the name Node is required for proper functioning. Interface shows a graphical output of the disk space available and used in the cluster.



Graph7 Physical memory snapshot

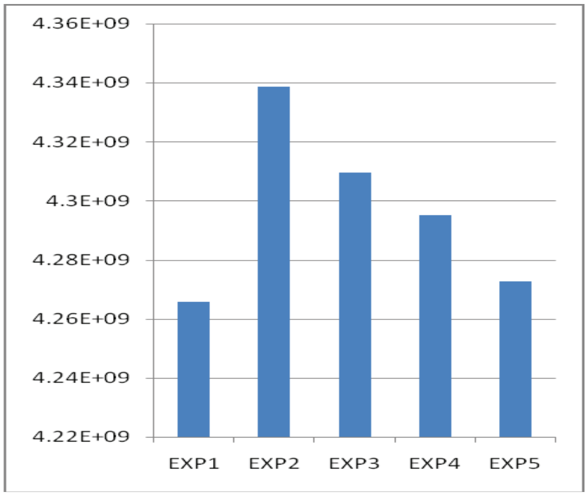
Again the HADOOP_HEAPSIZE is a parameter whose value can be set in the conf/hadoop.env.sh file. The size is in MB. The default value of this variable is the 100 MB. While performing the experiment we have already analyzed the value of this parameter for storing data on the cluster now we have also analyzed the values of this parameter in case of processing the data. The values are in bytes. The amount of heap size used by our different experiment is shown in the table 9.

Table 9 Heap size for processing data

Total committed heap usage (bytes)	
Experiment No	Value
EXP1	4265869312
EXP2	4338679808
EXP3	4309647360
EXP4	4295229440
EXP5	4272685056

The graphical output of the total committed heap usage is shown in the graph 8. The amount is in bytes. The Hadoop mapreduce task dy default produce result in bytes.

.



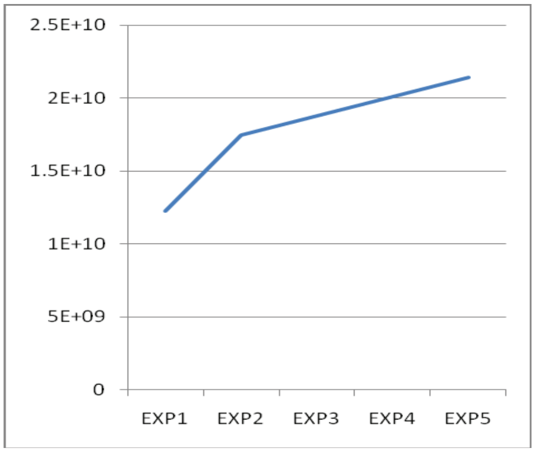
Graph 8 Heap size for processing data

There are different jobs and processes running on a single time in the Hadoop cluster. It is required to that the behavior of the jobs should be smooth. If a job behaves badly that can put a bad impact on the memories in the Cluster.

Table 10 Virtual Memory snapshot for processing the data

Virtual memory (bytes) snapshot	
Experiment No	Value
EXP1	12229713920
EXP2	17475268608
EXP3	18773729280
EXP4	20090236928
EXP5	21404524544

There can be different kinds of memory which are working in the Hadoop cluster. Table 10 shows the behavior of the virtual memory in the cluster with increasing number of nodes for processing the data in the cluster. The graph 9 below shows the graphical view of the output. This view shows that the use of the virtual memory by our setup is normal and all jobs are performing well. Any uncertain change in this graph will tells that something went wrong in the Hadoop cluster running jobs, which affecting the use of the virtual memory.



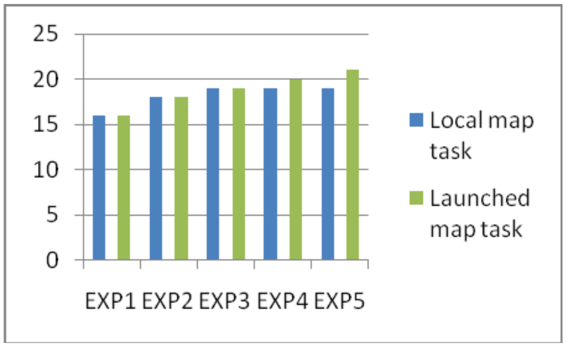
Graph 9 Virtual Memory snapshot

The map tasks are performed in the cluster for finding the location of the exact block on which the data is actually stored. In our next analysis, we have analyzed the number of map tasks and the data local map task for processing the data.

Table 11 Map tasks

Data-local map tasks		
Experiment No	Local map tasks	Launched map task
EXP1	16	16
EXP2	18	18
EXP3	19	19
EXP4	19	20
EXP5	19	21

Table 11 shows the map task which are run on local and are launched on the cluster. The graphical output is shown in the graph 10.



Graph 10 Map tasks

V. Conclusion and Suggestions

This work shows the behavior of the hadoop cluster with increasing number nodes. The parameter for which the behavior is analyzed is the memory parameters. This work will be useful for the developing a hadoop cluster.The amount of communication increases as the size of the cluster size increases. We will not recommend using the Wubi installation of the Ubuntu for developing the cluster because after some time it will start giving problem. Ubuntu can be very efficient for developing the hadoop cluster. There is no need to develop the repository for the installation of the Hadoop cluster. If the size of the data increases and there may be a chance of out of disk then the standard copy script should be used for increasing the size of the virtual disks.

References

[1] http://en.wikipedia.org/wiki/Data

[2] Changqing Ji, Yu Li, Wenming Qiu, Uchechukwu Awada, Keqiu Li, "Big Data Processing in Cloud Computing Environments" 2012 International Symposium on Pervasive Systems, Algorithms and Networks.

[3] Wei Tan, M. Brian Blake and Iman Saleh, Schahram Dustdar "Social-Network-Sourced Big Data Analytics" IEEE Computer Society 1089-7801/13 2013 IEEE

[4] Aditya B. Patel, Manashvi Birla, Ushma Nair, "Addressing Big Data Problem Using Hadoop and Map Reduce", NUiCONE-2012, 06-08DECEMBER, 2012.

[5] Zibin Zheng, Jieming Zhu, and Michael R. Lyu , "Service-generated Big Data and Big Data-as-a-Service: An Overview" , 978-0-7695-5006-0/13 © 2013 IEEE.

[6] Yang Song, Gabriel Alatorre, Nagapramod Mandagere, and Aameek Singh,"Storage Mining: Where IT Management Meets BigData Analytics", IEEE International Congress on Big Data 2013.

[7] Parameters http://grepcode.com/file/repo1.maven.org/maven2/org.apache.hadoop/hadoop-mapreduce-client-core/0.23.1/org/apache/hadoop/mapreduce/JobCounter.properties

[8] http://hadoop.apache.org/releases.html.