Performance Evaluation of Mini Single Board Computer in Hadoop Big Data Cluster

IOP Conf. Series: Materials Science and Engineering **875** (2020) 012037 doi:10.1088/1757-899X/875/1/012037

Performance Evaluation of Mini Single Board Computer in Hadoop Big Data Cluster

Adnan[1]*, Z Tahir[1], C Yohannes[1], and Ariel[1]

[1]Department of Informatics, Faculty of Engineering, Hasanuddin University, Makassar, Indonesia

*Email: adnan@unhas.ac.id

Abstract. Hadoop is a Java-based and open source software framework that serves to process big size data in a distributed manner. Hadoop uses a framework for applications and programming called MapReduce. Along with the development of Hadoop, big data research using Hadoop is running using a single board computer (SBC). SBC is a mini computer which has a specification similar to a conventional computer but has lower specifications and power usage. However, the use of storage used on a single board computer has constraints, namely low bandwidth and limited local storage, therefore NAS is used because it has large storage and bandwidth speeds is upto1 Gbps. It uses as storage that is carefully used on Hadoop. Based on testing by running Hadoop in a single node and multi nodes using Micro SD storage and NAS does have a significant difference. When running Hadoop on a single node the difference in data access speed generated by Micro SD and NAS has a difference of 2 seconds. On multi nodes (clusters) the ratio is 4 seconds. The difference in NAS bandwidth that is 2 times faster than Micro SD show a significant difference if it is run on Hadoop.

1. Introduction

Evolution of Information and communication technology nowadays is mainly in unstructured data. Researchers faced to the rapid growth of data. The rapid growth of data lead to the trend called Big data. Big data is a term that describe those both structured and unstructured data sized extremely large. Big data is difficult to store, collected, managed as well as analyzsed using legacy database. It is because those data growth rapidly and continously. International Data Corporation estimated that the size of data in digital world is around 0.18 zettabytes. IDC also predicted big data will be 10 times fold every five years [1].

Parallel computing is a kind of computing process in which a large scale of of computation is performed simultaneously by multiple computing resources. Nowadays the popular parallel computing platform for which to process a large amount of data distributedly is Hadoop. Hadoop constitute a framework software developed based on Java. Dough cutting, the same person who developed Apache Lucene, was the first person to introduce Hadoop at the first time. The Hadoop project started with Apache Nutch project as a part of Apache Lucene [2].

Among the many ways used to analyze big data, Hadoop is the one framework used because of its Hadoop Distributed File Syatems as well as its method called MapReduce. Mapreduce make Hadoop is possible to process big data in parallel and distributed manner on hundreds or even thousands of computer.

In this research, we make use of Banana Pi M3. Banana Pi M3 is a single board computer that has component similar to the desktop computer. Banana Pi M3 has better specifications. They supported by eight core processors but lower price. However, Banana Pi is simlar to other type mini SBC. They use Micro SD as main storage. Unfortunately, Micro SD storage is known to slower and smaller capacity than HDD. Consequently we propose to leverage Network Attached Storage (NAS) as a alternative. We propose to relocate HDFS from Micro SD to Network Attached Storage. We suggest that MapReduce application may have better performance when we make use of NAS compared to default storage Micro SD. NAS is kown to have large capacity and low overhead [3].

This paper presents our experimental result on evaluating the performance of MapReduce aplication on SBC equipped with Network Attached Storage and Micro SD. We compared the result on both using Network Attached Storage and Micro SD.

2. Related Works and Tools

2.1. Hadoop

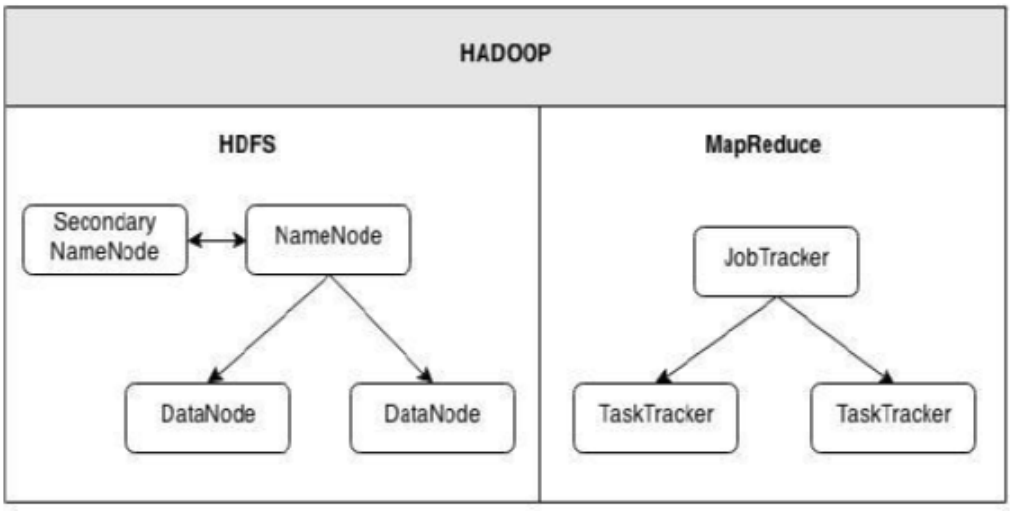
Hadoop constitutes an open source software framework developed on top the Java programming language. Hadoop functions to process extreeme size of data on cluster. Cluster is a kind of distributed memory parallel computer build from multiple computer connected using high speed network interconnection [1]. According to [4], Hadoop MapReduce is able to process data of very large size. The size of data could be in petabyte (10^{12}) order on top large scale cluster (upto thousand nodes of computer). Originally, Dough Cutting developed Hadoop as part of Nutch project belong to Apache Foundation.

Hadoop consists of common Hadoop which is usefull in providing access to Hadoop Distributed File System. This common hadoop constains packages needed by JAR files, scripts needed for strarting Hadoop, and documentatins for the jobs which are completed by Hadoop.

According [1], the core of the Hadoop is consist of :

1. Hadoop Distributed Files System (HDFS)
   This is a file system used by Hadoop to distribute data being processed by multiple nodes
2. Mapreduce framework
   Thiss a programming framework work for distributed processing using Hadoop.

Figure 1 shows the core a Hadoop which is consist of HDFS and MapReduce. Fig. 1(a) shows HDFS components which consist of NameNode and DataNode. Further, Secondary node is a backup for the NameNode. Fig. 1(b) shows that in MapReduce there is jobTracker as well as TaskTracker.



(a) (b)

Figure 1. Two main component of Hadoop. (a) HDFS (b) MapReduce

Hadoop on a small cluster could consist of one node of master and some nodes of slaves. The master node is consistof NameNode and JobTracker, whereas the Slave nodes are consist of some DataNodes and TaskTracker. Hadoop required JRE version 1.6 or newer version. When starting and also stopping the Hadoop sistem, SSH connections are needed among the nodes of the Hadoop cluster [5].

2.2. Hadoop Distributed File System

Hadoop Distributed File System (HDFS) is Java-based file system that is distributed for Apache Hadoop [1]. As a distributed file system, HDFS useful for handling large amounts of data which is stored and distributed in many connected computers which is commonly called a cluster. Distributed file systems on Hadoop can be assumed as a file system that stores data not in one local Hard Disk Drive (HDD) or other media storages , but the data is fragmented (files are distributed in block form with a size of 64MB – this size of block if configurable however), and distributedly stored in multiple nodes of cluster.
HDFS stores data in a way such that HDFS split the data into chunk of data of size 64MB (default). The chunk of the data are stored scattered in each nodes of cluster. These pieces of data in HDFS are called blocks. The block size in each file is not necessarly to be fixed to 64 MB, where the block size can be adjusted to the user's preference. Although the data is stored scattered to several nodes, but from the user's perspective, the data still looks like we access files on one computer. Files that are physically spread across multiple computers can be treated like treating files on one computer [2].
As a distributed file system, HDFS has main components such as NameNode, DataNode, and Secondary NameNode [1]. The architecture of the three components can be seen in Figure 2.

2.3. MapReduce

MapReduce is a framework for applications and programming that was introduced by Google and used to do a job of distributed computing that is run on a cluster [6]. MapReduce consists of the concept of map and reduce functions that are commonly used on functional programming [5].

One program that uses the MapReduce concept that has been provided by Hadoop is WordCount. WordCount is a benchmark program that aims to count words in a plaintext file. The MapReduce process in WordCount is divided into 2 stages: the process of mapping and reducing.

2.4. Hibench

HiBench is a big data suite benchmark that favor evaluating a variety of large data frameworks in terms of speed, throughput and utilization of system resources. It contains a set of Hadoop, Spark and streaming workloads, including Sort, WordCount, TeraSort, Sleep, SQL, PageRank, indexing of Nutch, Bayes, Kmeans, NWeight and DFSIO, etc. It also contains several streaming workloads for Spark Streaming, Flink, Storm and Gearpump [7].

Basically, the purpose of TeraSort is to sort the data of 1TB (or another amount of data that is preferred) enabled as possible. This is a benchmark that complements the HDFS and MapReduce layer testing of the Hadoop cluster. There are 3 steps that are used to carry out benchmarking benchmarking, namely:

1. Generate random data using Teragen
2. Sort the generated data using Terasort program
3. Validate sorted data using Teravalidate

2.5. Performance Analysis Tools

Performance analysis tool is a flexible performance analysis framework designed for Linux operating systems. PAT collects system-level performance metrics including CPU, DISK and NETWORK usage. And generate files in the form of pdf or Excel and provide a visual display of the results of the presentation of the data that has been collected.

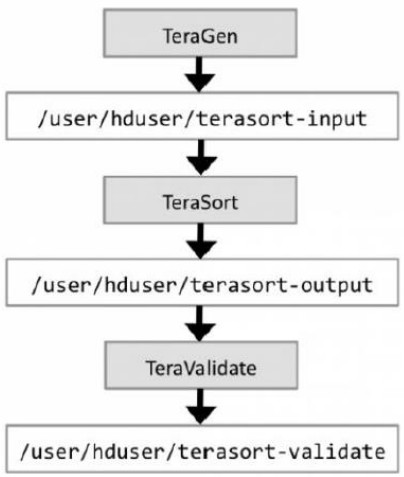


Figure 2. Basic work flow of Terasort benchmark using Hadoop MapReduce

3. Experimental Configuration

3.1. Hardware and Software Platform

In our works, we make use both hardware and software as follows : We make use six Banana Pi M3 equipped with two types of storages. In the first scenario, Banana Pi M3 used 32 GB Micro SD class 10 80 MB/sec. In the 2nd scenario, Banana PI M3 connected to a Network attached storage via gbe network using gigabit ethernet switch. The Banana PI M3's specification described as in Table 1, while the storage used in both scenario as present as in Table 2.

Table 1. Banana Pi M3 specification

Component	Specification
CPU	Octa-core A83T ARM Cortex A7 2 GHz
RAM	2 GB
Network interface	Gigabit ethernet and WiFi

Table 2. Storage specification

Storage type	Capacity and link speed
Micro SD	32 GB / 80 MB/sec
NAS	2TB /1000 Mbps

3.2. Scenario of Experiments

3.2.1. First Scenario

The first scenario intends to analyze performance of I/O (storage access) using single node. In this scenario, Hadoop MapReduce ran using single node of Banana Pi M3. We measured the performance of MapReduce using Micro SD and NAS as HDFS storage. We compared the execution time for both experiments while using two type of the storage. For both experiments in this scenario, PAT is installed on NAS, whereas the data being sorted are located in hdfs (Micro SD or NAS respectively). Performance analayzer tools (PAT) is to obtain performance data such as user CPU time, iowait, system time etc. We make the size of data being sorted varied from 320 MB up to 2 GB.

3.2.2. Second Scenario

The second scenario we intend to analyze the performance of I/O (storage access) using multiple nodes. In this scenario, Hadoop MapReduce bencmark, that is terasort, ran using multiple node of Banana Pi M3. We measured the performance of MapReduce using Micro SD and NAS as HDFS storage. We

compared the execution time for both experiments while using two type of the storage as the base of HDFS (it means that data being sorted are located on files of HDFS). For both experiments in this scenario, PAT is installed on NAS. Performance analyzer tools (PAT) is to obtain performance data such as CPU time, iowait, system time etc. We make the size of data being sorted varied from 1 GB up to 4 GB.

4. Experimental Results

4.1. The first Scenario

The first scenario aims to determine the performance of Micro SD and NAS data access speeds on Hadoop. The first scenario is to run Hadoop in a single node using Micro SD and NAS storage. The file sizes used in the first scenario are 320MB, 500MB, 1GB, and 2GB. Experiments in the first scenario carried out 10 attempts.

The results of the measurement of data access speed in the first scenario by running Hadoop in a single node (cluster) using Micro SD and NAS storage can be seen in the Table 3.

Table 3. Execution time of Terasort with different type of storage

Storage type	File size/execution time			
	320 MB	500 MB	1 GB	2 GB
Micro SD	287.5 secs	423.76 secs	852.88 secs	1584.75 secs
Network Attached Storage	284.96 secs	423.57 secs	816.15 secs	1583.75 sec

4.2. Second Scenario

The first scenario aims to obtain the performance measure of Micro SD and NAS data access speeds on Terasort of Hadoop MapReduce. The first scenario is to run Terasort on multi-node (cluster) using Micro SD and NAS storage. The file size used in the second scenario is 1 GB, 2 GB, 3 GB, and 4 GB. Experiments in the second scenario carried out 10 attempts. The results of the measurement of data access speed in the second scenario by running Hadoop Terasort in a multi node (cluster) using Micro SD and NAS storage can be seen in Table 4.

Table 4. Execution time of Terasort with different type of storage

Storage type	File size/execution time			
	1 GB	2 GB	3 GB	4 GB
Micro SD	49.07 secs	56.18 secs	59.63 secs	67.79 secs
Network Attached Storage	46.50 secs	50.52 secs	52.83 secs	60.13 sec

Based on the results of experiments conducted in the second scenario, it can be seen that the difference in data access speed on Hadoop which is run in a multi node using NAS has a significant difference from Micro SD. The tables show that the speed produced by Micro SD and NAS have a significant difference, where the difference in the speed of time produced by the two storage is only 3 seconds different in each file tested, the insignificant difference.

4.3. CPU Utilization

This subsection reports experimental result on using PAT to obtain CPU utilization while running MapReduce benchmark. Data presented is obtained from experiments while using NAS and Micro SD as HDFS media.

4.3.1. Single Node

Table 5. Single node

Componen of CPU Utilization	Micro SD as HDFS media	Network Attached Storage as HDFS media
	Portion (%)	
User time	11.20	12.88
System	4.03	5.16
Nice	0.00	0.00
I/O wait	8.25	0.79
Steal	0.00	0.00
Idle	76.50	81.15

4.3.2. Multiple Nodes

Table 6. Multiple nodes

Component of CPU Utilization	Micro SD as HDFS media	Network Attached Storage as HDFS media
	Portion (%)	
User time	17.04	22.88
System	8.28	10.89
Nice	0.00	0.00
I/O wait	41.34	44.47
Steal	0.00	0.00
Idle	33.32	21.74

From the tables, using NAS is better than Micro SD as by using NAS could improve user time and reduces idle. Iowait increase slightly because of network contention by multiple nodes. This is must be a contention as in single nodes Iowait in using NAS is lower than in Micro SD.

5. Conclusion

In the test running Hadoop in a single node did not show a significant difference of 2 seconds. While in multi-node using Micro SD and NAS storage there is a significant difference of 7 seconds (10%). This difference is caused by all processors spending more time on user code and reducing idle even though iowait is slightly increased. This increase in iowait is very small compared to the large number of processors added, from 4 to 24.

Acknowledgment

This research is supported by LBE Grant 2019, Faculty of Engineering, Hasanuddin University.

References

[1] Lam Chuck, “Hadoop in Action” Stamford : Manning Publication. 2011.
[2] P. Kusumanegara, Analisis Performan Kecepatan MapReduc pada Hadoop Menggunakan TCP packet Flow Analysis. 2014.
[3] J. Webster, “Understanding Hadoop Technology and Storage,” [online].
[4] White Colin, MapReduce and the Data Scientist. BI Research, 2012.
[5] Magang Industri, “Definisi Cloud Computing Meruvia.org Cloud Computing,” 2013.
[6] Apache, “Apache TM Hadoop” [online] : https://hadoop.apache.org.
[7] Intel Hadoop, “PAT, Performance Analysis Tools”. [online]. Available https://github.com/Intel-hadoop/.