TELKOMNIKA, Vol.14, No.3, September 2016, pp. 1059~1066
ISSN: 1693-6930, accredited A by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v14i3.1886

Hadoop Performance Analysis on Raspberry Pi for DNA Sequence Alignment

Jaya Sena Turana*, Heru Sukoco, Wisnu Ananta Kusuma
Bogor Agricultural University,
Jl. Raya Darmaga Kampus IPB Darmaga Bogor 16680. Phone. +62 251 8622642
*Corresponding author, e-mail: sena.turana@gmail.com

Abstract

The rapid development of electronic data has brought two major challenges, namely, how to store big data and how to process it. Two main problems in processing big data are the high cost and the computational power. Hadoop, one of the open source frameworks for processing big data, uses distributed computational model designed to be able to run on commodity hardware. The aim of this research is to analyze Hadoop cluster on Raspberry Pi as a commodity hardware for DNA sequence alignment. Six B Model Raspberry Pi and a Biodoop library were used in this research for DNA sequence alignment. The length of the DNA used in this research is between 5,639 bp and 13,271 bp. The results showed that the Hadoop cluster was running on the Raspberry Pi with the average usage of processor 73.08%, 334.69 MB of memory and 19.89 minutes of job time completion. The distribution of Hadoop data file blocks was found to reduce processor usage as much as 24.14% and memory usage as much as 8.49%. However, this increased job processing time as much as 31.53%.

Keywords*: big data, hadoop, raspberry Pi, DNA sequence alignment*

1. Introduction

IBM defines big data as having three characteristics: volume, variety and velocity. Volume refers to the size of the data, variety refers to the type of the data (text, sensory data, audio, video, etc.) and velocity refers to the frequency of the data that is produced by an application or the analyzing speed of the data produced [1]. Two major challenges with big data are how to store it and how to process it, and the most important thing is how to understand the data and transform it into meaningful information. The main problems in processing big data are the high cost, both for the hardware and the software, and computational power [2]. One other problem is it requires a lot of electricity to power the hardware that in turn has an adverse effect on the environment [3]. Hadoop [4] is one of the open source software frameworks developed to manage big data. Hadoop is also designed to be able to run on commodity hardware, so it can cut the cost to manage big data. There are two main components of Hadoop: Hadoop File System and MapReduce. These components are inspired by Google GFS and MapReduce projects [5]. HDFS is a distributed file system and MapReduce is a framework for analysing and transforming large data sets. HDFS stores metadata and application data separately. Metadata is stored in the NameNode while application data is stored in the DataNode [6].

Raspberry Pi is a commodity hardware the size of a credit card produced by the Raspberry Pi Foundation. Raspberry Pi as a mini computer has a capability to doing everything a desktop computer to do, such as browsing, playing a video, making a spreadsheet, and playing games [7]. Raspberry Pi has two major advantages than the others mini computer. First is a simple installation. Raspberry Pi operating system by default installed on SD Card, this feature will speed up the creation of the cluster because only make SD card duplication and do a few configuration changes. Second, Raspberry Pi are much more cheapest than the others. For example, in June 2015, the pricing was $125 to $149 for BeagleBoard Models, $49 to $89 for BeagleBone Models, $174 to $182 for PandaBoard Models, and $25 to $35 for Raspberry Pi Models. The cost of Raspberry Pi is low and it requires little electricity [8]. Raspberry Pi enables the construction of low-cost and energy efficient cluster. However, it has several limitations. One of them is the slow performance of the SD card. The lifetime of the SD card is also significantly shorten with applications that frequently performs writing operation on the SD card

Received April 16, 2016; Revised July 15, 2016; Accepted July 29, 2016

[9]. There is an increasing interest in using Hadoop to manage Big Data in various research fields. One of them is bioinformatics [10]. A lot of massive data sets are used in this field to apply mathematical, statistical and informatical methods to solve biological problems, mainly related to DNA sequence and amino acids. Next-generation DNA sequencing are generating billions of sequence data; it's made sequence alignment cannot perform on standalone machines. One solution to this problem is running the algorithm on a cloud or cluster. Applying the algorithm to run parallel with Hadoop [16] has many advantages; there are scalability, redundancy, automatic monitoring and high performance. The aim of this research is to implement and analyze Hadoop Cluster on Raspberry Pi for DNA sequence alignment [11] using Biodoop library [12]. Through this research, it is expected that an easy and cost effective manner to manage big data is found. It is also expected that this research can be used to assist in developing an environmentally friendly technology which has energy conservation as one of the indicators [13].

2. Related Works

Iridis-Pi cluster was build with 64 nodes Raspberry Pi Model B. This cluster was design for educational applications, where it enables students to understand and apply high-performance computing and data handling for complex engineering, and scientific challenges [17]. Glasgow Cloud Data Center was build with 54 nodes Raspberry Pi Model B. This cluster was emulated every layer of cloud stack, ranging from resource virtualization to network behaviour, providing a full-featured of cloud computing research and educational environment [8]. Bolzano Cloud cluster was build with 300 nodes Raspberry Pi Model B. This cluster was designed to create affordable and energy-efficient cluster [9]. High Performance Computing (HPC) with 14 Raspberry Pi model B was tested for running 1000x1000 matrix and test result show HPC can process the data to complete [18].

3. Research Method

Six Model B Raspberry Pi with Raspbian operating system were used in this research. One Raspberry Pi was used as Hadoop NameNode and five others as Hadoop DataNode. Wordcount application was used for the initial testing of the Hadoop cluster architecture. This application was available in Hadoop installation. Ganglia software was used to monitor Hadoop cluster resource [14]. The Ganglia was divided into two data sources: hadoop-masters and hadoop-slaves. The hadoop-masters consisted of Hadoop NameNode and the hadoop-slaves consisted of the entire Hadoop DataNodes. Ganglia meta daemon and Ganglia web frontend were installed on a virtual machine on a Notebook. The virtualization software used for this research was VirtualBox. The DNA sequence alignment process was performed using Biodoop software. The steps carried out by Biodoop for DNA sequence alignment are as follow:

1. DNA data with fasta format upload to HDFS by web monitoring
2. Biodoop is running fasta2tab application. This application converts fasta format sequence to TAB-delimited format.
3. Biodoop is running biodoop_blast application. This application is a wrapper based MapReduce implementation of BLAST for Hadoop.

For sequence alignment performance result comparison, this research is using six PC IBM Lenovo MT-M 8800-5CJ with dual core processor 1.86GHz, one 160 GB hard disk and 1 GB RAM. The DNA data for this research was obtained from the National Center for Biotechnology Information (NCBI). The DNA data is shown in Table 1.

Table 1. DNA data

DNA Name	Length (bp)	Size (KB)
Ancylostoma duodenale mitochondrion (NC_003415.1)	13,271	14.00
Necator americanus mitochondrion (NC_003416.2)	13,605	13.88
Chaetoceros tenuissimus DNA virus (NC_014748.1)	5,639	5.81
Chaetoceros lorenzianus DNA Virus (NC_015211.1)	5,813	5.98
Human papillomavirus type 132 (NC_014955.1)	7,125	7.31
Human papillomavirus type 134 (NC_014956.1)	7,309	7.49

For the purpose of this research, a web-based monitoring application was built to analyze Hadoop Job Tracker on the Hadoop cluster. The data for the analysis was obtained from the Ganglia Monitoring using a web service protocol provided by the Ganglia API [15]. A daemon application was built using Java programming language to monitor resources that cannot be monitored by the Ganglia, such as temperature and disk input/output. This application communicated with the web monitoring through a socket protocol.

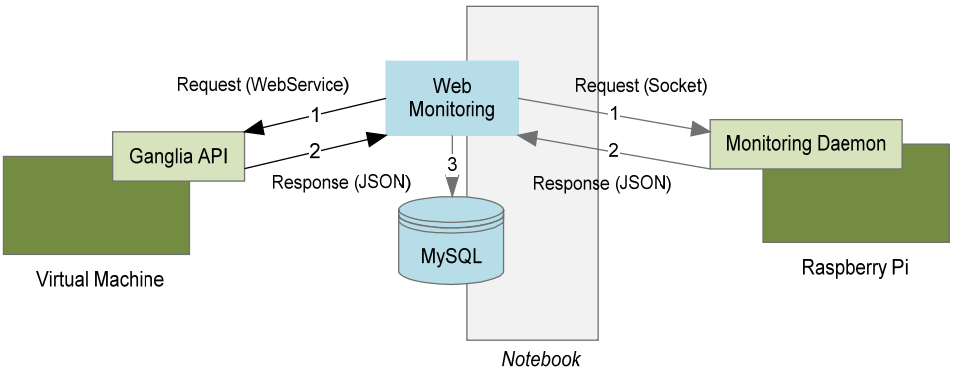

Figure 1. Web monitoring communication process

The process of data collection from Ganglia API and monitoring daemon can be seen in Figure 1. The steps carried out by the web monitoring are as follow:

1. The web monitoring makes a request to Ganglia API and monitoring daemon.
2. The Ganglia API and the monitoring daemon send a data text response with a Javascript Object Notification (JSON) format to the web monitoring.
3. The web monitoring parses the response data and stores it in the database.

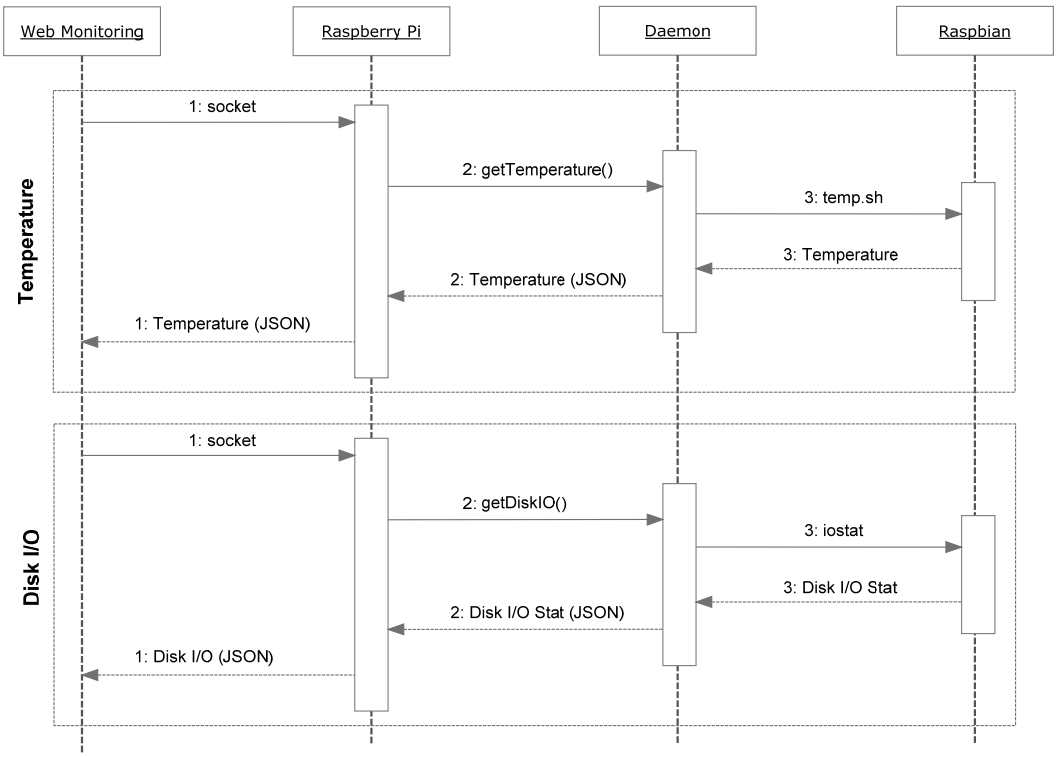

Figure 2. Sequence diagram of daemon monitoring application

The summoning process of the temperature and the disk I/O daemon application by the web monitoring is shown in Figure 2. The steps to obtain temperature and disk I/O values are as follow:

1. The web monitoring makes a request through a socket protocol to the monitoring daemon.
2. The daemon application summons temp.sh script for temperature request or iostat for disk I/O request.
3. The daemon application sends a data text response in JSON format to the web monitoring.

Three DNA sequence alignments were performed in this research and each alignment was tested twice. In the first trial, the default block size was used while in the second one the values of the block size were modified. The metrics to be measured were processing time, processor usage, memory usage, disk input/output, network input/output and temperature. The results of the two trials were then compared to see the effect of the modifications. The value of the change in percent was calculated using the following formula:

$$Performance = 100 - ((\frac{Research\ Result\ Resource\ II}{Research\ Result\ Resource\ I}) x100)$$

A positive value means the block modification improves performance, and a negative one means it lowers the performance.

4. Results and Analysis

Some of the Hadoop variable values had to be adjusted so they could run on Raspberry Pi. The specification of Raspberry Pi was below the minimum requirements for Hadoop. The adjusted values were as follow:

1. **HADOOP_HEAPSIZE**. Hadoop is an application written in the Java language. Maximum memory allocation for Java Virtual Machine (JVM) or commonly called heap size is important for running Java application. This research was conducted a few experiments to get heap size value and determined that the value of heap size is 384 MB. Heap size experiment is shown in Table 2.

Table 2. Heap size trial

Heap (MB)	Error	Error Message
64	Yes	Java heap space
128	Yes	Java heap space
192	Yes	Java heap space
256	Yes	Java heap space
320	Yes	Java heap space
384	**No**	
448	Yes	Could not reserve enough space for object heap
512	Yes	Could not reserve enough space for object heap

Java heap space or commonly called java.lang.OutOfMemoryError Thrown when the Java Virtual Machine cannot allocate an object because it is out of memory, and no more memory could be made available by the garbage collector.

2. **Timeout**. This research was used a small file size. Split file into a smaller blocks file shows that MapReduce jobs will be longer than usual because of overhead the splitting process and creating Hadoop task. The value of dfs.client.file-block-storage-locations.timeout (the default value of 1 second) was modified to 1200 seconds.

3. **Block file size**. Hadoop has a minimum block file size equal to 1 MB and block file size equal to 128 MB. In this research, because of using a small file size (7-14 KB), minimum block size (dfs.namenode.fs-limits.min-block-size) was changed to 512 bytes and a block size (dfs.blocksize) modified accordingly by DNA sequence alignment trial.

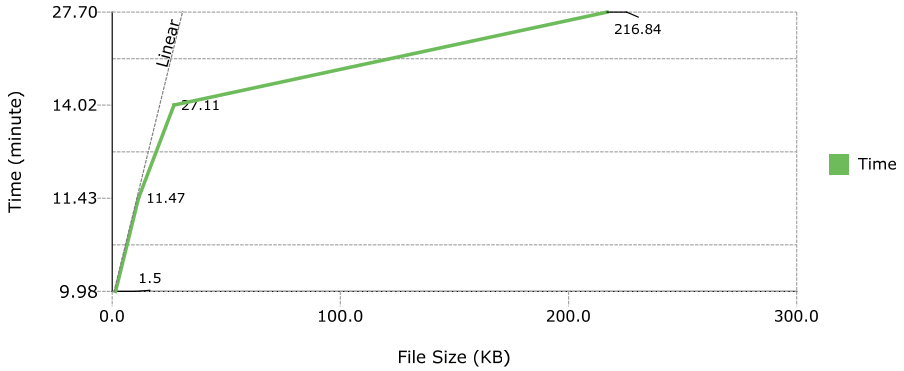

Figure 3. Hadoop cluster test using wordcount application

The result of Hadoop cluster test using Wordcount application is shown in Figure 3. Four trials were performed with different file sizes. This application was running on the cluster with average job time completion is 19.89 minutes. The result of the test shows that the relation between the file size and the completion time is not linear, and Hadoop works optimally when the file is larger.

For the temperature data collection, the daemon application summoned a shell script in the Raspbian operating system, while for the disk I/O, the daemon application obtained the data from the iostat application.

4.1. Hadoop Cluster Performance Analysis for DNA Sequence Alignment

DNA sequence alignment was performed three times and each alignment was tested twice. The first trial was performed using the default block size of the Hadoop and the second one was performed with block size modification. The modification was done to distribute file blocks to the entire Hadoop DataNode.

1. *Ancylostoma duodenale mitochondrion* (NC_003415.1) and *Necator americanus mitochondrion* (NC_003416.2)

Reference sequence : Ancylostoma duodenale mitochondrion
Query sequence : Necator americanus mitochondrion

On the first trial, a 128 MB block size was used for both sequences. On the second trial, a 3 KB block size was used for the NC_003415.1 sequence and a 10 KB block size was used for the NC_003416.2 sequence. A more than 89% DNA similarity was found on the sequence alignment with a bit score of 1225. The test result of the Raspberry Pi resource usage and processing time for DNA 1 sequence alignment is shown in Table 3.

Table 3. Test result for NC_003415.1 and NC_003416.2

			Disk (Kbps)		Network (Kbps)			
	Processor (%)	Memory (MB)	Read	Write	Input	Output	Temp. (C)	Time (m)
Trial I (Default block size)								
Raspberry	82.69	344.76	22.35	3.82	0.51	4.21	38.26	17.66
PC	24.07	502.07	79.65	3.30	0.32	0.60	50.12	1.13
Trial II (Block size : 10 KB)								
Raspberry	62.50	335.91	8.28	6.45	7.28	9.51	49.04	25.86
PC.	22.10	415.24	15.97	9.82	2.17	12.91	49.10	1.16

It can be gathered from Table 3 that the DNA sequence alignment used almost the entire capacity of the processor and memory on Raspberry Pi. On PC, block size changes does not significantly affect to processor and execution time. File block distribution had the following effects:

a. A decrease in average processor and memory use.
b. A decrease in disk read speed and an increase in disk write speed.

c. An increase in network input/output speed.
d. An increase in temperature and job completion time.

2. *Human papillomavirus type 132* (NC_014955.1) and *Human papillomavirus type 134* (NC_014956.1)

Reference sequence : Human papillomavirus type 132
Query sequence : Human papillomavirus type 134

On the first trial, a 128 MB block size was used for both sequences. On the second trial, a 3 KB block size was used for the NC_014955.1 sequence and a 5 KB block size was used for the NC_014956.1 sequence. A more than 93% DNA similarity was found on the sequence alignment with a bit score of 43.5. The test result of the Raspberry Pi resource usage and processing time for DNA 2 sequence alignment is shown in Table 4.

Table 4. Test result for NC_014955.1 and NC_014956.1

			Disk (Kbps)		Network (Kbps)			
	Processor (%)	Memory (MB)	Read	Write	Input	Output	Temp. (C)	Time (m)
Trial I (Default block size)								
Raspberry	92.14	355.49	16.34	4.73	4.91	7.69	48.04	15.45
PC	20.93	719.73	46.62	22.81	7.28	32.72	44.75	1.10
Trial II (Block size : 5 KB)								
Raspberry	67.01	312.36	14.02	7.91	3.08	6.52	47.29	19.84
PC	17.48	679.31	37.88	8.62	4.03	30.59	51.25	1.19

It can be gathered from Table 4 that the DNA sequence alignment used almost the entire capacity of the processor and memory on Raspberry Pi. On PC, block size changes does not significantly affect to processor and execution time. File block distribution had the following effects:

a. A decrease in average processor and memory use.
b. A decrease in disk read speed and an increase in disk write speed on Raspberry Pi.
c. A decrease in disk read/write speed on PC.
d. A decrease in network input/output speed.
e. A decrease in temperature on Raspberry Pi and an increase in temperature on PC.
f. An increase in job completion time.

3. *Chaetoceros tenuissimus DNA virus* (NC_014748.1) and *Chaetoceros lorenzianus DNA Virus* (NC_015211.1)

Reference sequence : Chaetoceros tenuissimus DNA virus
Query sequence : Chaetoceros lorenzianus DNA Virus

On the first trial, a 128 MB block size was used for both sequences. On the second trial, a 3 KB block size was used for both the NC_014748.1 dan NC_015211.1 sequences A more than 83% DNA similarity was found on the sequence alignment with a bit score of 63. The test result of the Raspberry Pi resource usage and processing time for DNA 3 sequence alignment is shown in Table 5.

Table 5. Test result for NC_014748.1 and NC_015211.1

			Disk (Kbps)		Network (Kbps)			
	Processor (%)	Memory (MB)	Read	Write	Input	Output	Temp. (C)	Time (m)
Trial I (Default block size)								
Raspberry	71.24	348.62	13.97	5.03	4.74	7.43	47.61	15.07
PC	34.91	798.56	43.33	11.58	5.69	3.93	43.62	1.29
Trial II (Block size : 3 KB)								
Raspberry	62.88	311.02	16.46	10.32	7.24	9.89	48.36	25.44
PC	30.19	628.63	46.80	12.28	8.17	9.13	46.99	1.57

It can be gathered from Table 5 that the DNA sequence alignment used almost the entire capacity of the processor and memory on Raspberry Pi. On PC, block size changes does not significantly affect to processor and execution time. File block distribution had the following effects:

1. A decrease in average processor and memory use.
2. An increase in disk read speed and disk write speed.
3. An increase in network input/output speed.
4. An increase in temperature and job completion time.

From all of the DNA sequence alignment trials on Raspberry Pi above, it can be concluded that file block distribution on DataNode can lower the average processor use as much as 24.14% and the average memory use as much as 8.48%. However, this increases the job processing time as much as 31.53%. On PC, block size changes does not significantly affect the processor, memory and processing time. Processor use lower as much as 12.73%, memory use as much as 14.73% and increasing processing time as much as 10.85%. Split file into a smaller blocks file shows that MapReduce jobs will be longer than usual because of overhead the splitting process and creating Hadoop task. File block distribution not directly affects disk I/O, network I/O and temperature.

5. Conclusion

As low cost commodity hardware, Raspberry Pi can be used as an alternative hardware to implement Hadoop cluster. Hadoop cluster can work well on Raspberry Pi. The only disadvantage is the increase of job completion time even though it is only for a simple job. Big data implementation such as DNA sequence alignment was running on the Raspberry Pi with an average usage of processor 73.08%, 334.69 MB of memory and 19.89 minutes of job time completion. The distribution of Hadoop data file blocks was found to reduce processor usage as much as 24.14% and memory usage as much as 8.49%. However, this increased job processing time as much as 31.53%.

Apache Hadoop has already released some tools to compile Hadoop source into a native library, including an ARM processor. A native library is a library that is recompiled to native code according to the platform that is run. Future research can use Hadoop native library to see the performance of Hadoop cluster on Raspberry Pi or other low-cost commodity hardware.

Acknowledgements

The authors would like to thank Simone Leo (simone.leo@crs4.it) for his valuable comments and feedback regarding this research study.

References

[1] Paul C, Chris E, Dirk D, Thomas D, George L. Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. New York: McGraw-Hill Companies. 2012: 5-9.
[2] Aisling O, Jurate D, Roy DS. 'Big data', Hadoop and Cloud Computing in Genomics. *Science Direct Journal of Biomedical Informatics.* 46(5): 774-781.
[3] Jacob L, Christos K. On the energy (in) efficiency of Hadoop clusters. *ACM SIGOPS Operating Systems Review*. 2010; 44(1): 61-65.
[4] Apache Hadoop. http://hadoop.apache.org/.
[5] Dhruba B, et al. *Apache Hadoop Goes Realtime at Facebook*. International Conference on Management of Data. New York. 2011: 1071-1080.
[6] Konstantin S, Hairong K, Sanjay R, Robert C. *The Hadoop Distributed File System*. Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on. Incline Village, NV. 2010: 1-10.
[7] Raspberry Pi. http://www.raspberrypi.org/.
[8] Fung PT, David RW, Simon J, Jeremy S, Dimitrios PP. *The Glasgow Raspberry Pi Cloud: A Scale Model for Cloud Computing Infrastructures*. Distributed Computing Systems Workshops (ICDCSW), 2013 IEEE 33rd International Conference on. Philadelphia. 2013: 108-112.
[9] Pekka A, et al. *Affordable and Energy-Efficient Cloud Computing Clusters: The Bolzano Raspberry Pi Cloud Cluster Experiment*. Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on. Bristol. 2013: 170-175.

[10] Ronald CT. *An Overview of The Hadoop/MapReduce/HBase Framework and Its Current Applications in Bioinformatics*. Proceedings of the 11th Annual Bioinformatics Open Source Conference (BOSC) 2010. Boston. 2010.
[11] Stephen FA, Warren G, Webb M, Eugene WM, David JL. Basic Local Alignment Search Tool. *Science Direct Journal of Biomedical Informatics*. 1990; 215(3): 403-410.
[12] Simone L, Federico S, Gianluigi Z. *Biodoop: Bioinformatics on Hadoop*. Parallel Processing Workshops, 2009. ICPPW '09, International Conference on. Vienna. 2009: 415-422.
[13] San M, Gangadharan. Harnessing Green IT. New York, West Sussex: A John Wiley & Sons, Ltd.. 2012: 7-10.
[14] Matthew L, Brent N, David E. The Ganglia Distributed Monitoring System: Design, Implementation, and Experience. *Science Direct Journal of Biomedical Informatics*. 2004; 30(7): 817-840.
[15] Matt M, Bernard L, Brad N, Vladimir V. Monitoring With Ganglia. California: O'Reilly Media, Inc. 2013: 66-68.
[16] Yan X, Wang Z, Zeng D, Hu C, Yao H. Design and Analysis of Parallel MapReduce based KNN-join Algorithm for Big Data Classification. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2014; 12(11): 7927-7934.
[17] Cox SJ, Cox JT, Boardman RP, Johnston SJ, Scott M, O'Brien NS. Iridis-pi: a low-cost, compact demonstration cluster. 2014; 17(2): 349-358.
[18] Ashari A, Riasetiawan M. High Performance Computing on Cluster and Multicore Architecture. *TELKOMNIKA Telecommunication Computing Electronics and Control*. 2015; 13(4): 1408-1413.