Comparison on Efficiency of Computational Efforts between Cluster Computation (MapReduce) and Single Host Computation

Mulkan Fadhli[#+], Taufiq Abdul Gani[#+], Melinda[#+], Yuwaldi Away[#]
[#]*Center for Computational Engineering, Department of Electrical Engineering, Faculty of Engineering,*
[+]*Div. R/D on Computer Cluster, UNSYIAH-ADOC TAIWAN e-Learning Center,*
University of Syiah Kuala Darussalam - Banda Aceh 23111 Indonesia
{mulkan.fadhli; topgan; melinda; yuwaldi}@cce.unsyiah.ac.id,

Abstract – **The complexities of research in science have been increasing extremely. Numerous mathematical models have been developed. Matrix has been used popularly to model numerous and complex science and engineering problems. It is found that as the dimension of the matrix grows in size, the complexities of matrix computation increase. This problem may be solved by using large computer system (i.e. mainframe). However, its operational is very costly. Another solution is to utilize parallel computing, which are able to cut off the operational cost. A recent advance in parallel programming is the introduction of MapReduce, as a new approach in parallel programming. MapReduce can perform calculations with distributed method by utilizing an idle processor. In this research, the performance of MapReduce in matrix operation is compared to other conventional methods, which are Single Processor and Threads. The performances are assessed by comparing the execution time, CPU usage, and RAM usage of each approach. The results show that MapReduce performed better than the other approaches.**

Keyword: **MapReduce, Matrix, Parallel computing.**

1. INTRODUCTION

Development of parallel computing has increased extremely. Google, a giant IT Company introduced a very powerful framework for parallel and distributed computing on the lower-middle computer, which is called MapReduce framework. MapReduce is very easy to use and produces great scalability with a very high level of tolerance for error. Hence, if there is disorder it can be still executed. Google and Yahoo have used MapReduce system for their cluster system, especially to analyze search logs of the likelihood of user characteristics [7].

In this paper, the performance of MapReduce is assessed for common science and engineering problems, which is matrix computation [7]. The problem of matrix computation is the greater size of the matrix dimension, the greater of computational effort is required. The efforts are in terms of both memory and CPU time [6].

There are two approaches that can be done, namely: (i) designing robust algorithms for matrix operations (ii) utilizing idle processor in a network system. The use of idle processor in a network system is known as clustering [11]. A computer cluster has been recognized as a solution to improve the computational speed.

Furthermore, in this study the performance of parallel computing with MapReduce above method will be compared to the method of computing a single machine and the Single Processor Threads in the matrix calculation method. With variable ratio execution time, CPU usage and memory usage.

In the following discussion, there will be described parallel computing, hadoop and MapReduce as the latest developments in high performance computing. Then, it will be explained how MapReduce used for one of the basic operations of matrix multiplication. MapReduce source codes for the method of matrix multiplication can be downloaded from http://trac.nchc.org.tw/ to make some changes for experimental purposes.

2. FUNDAMENTAL THEORY

A. Parallel Computing

Parallel computing is a computational calculation using two or more CPU/Processor in the same computer or different computers in which case each instruction is divided into a few instructions and then it is sent to a processor, which involved computing and performed simultaneously [2]. The distribution of the computing process was carried out by a framework that served to set the example of MapReduce computing, MPI, and others.

The difference between a single computing with parallel computing, namely: a problem (job) done in a queue, where the job is the header that will be executed first, while in a parallel computing problems (job) is done in a distributed, which sorted out job (mapping) into the next few sections that will be distributed to the computer or when the CPU is idle.

B. Technology of Cluster Server

Clustering is the use of multiple computers, typically PCs or UNIX workstations, multiple storage devices, or complex interconnections, it also makes a format what

appears to users as a single system. Cluster computing can be used for load balancing as well as the needs of high availability server or the availability of high-level server [8].

In addition, Cluster computing can also be used as a relatively inexpensive form of systems, parallel processing for scientific applications in a reliable parallel operation of system interoperability. The most recent implementations mentioned previously are the implementation of Hadoop that can be applied in various needs of the server system.

C. Hadoop

Hadoop is a top-level project from the Apache software foundation. Hadoop only provides basic services to the developers to build cloud-computing environment using the medium and the API of Hadoop. MapReduce framework and Hadoop Distributed File System (HDFS)that are major part of the Hadoop [9].

In developing of Hadoop, there are three categories of machinery such as master node, slave node, and client. Master node takes a part in keeping two main functions, namely running Hadoop data set (HDFS) and parallel computing on all data (MapReduce). Name node manages and oversees the functions of data storage (HDFS), while Job tracker regulates and supervises the processing of data in parallel with the method of MapReduce. Slave node sets some machines (computers) and organizes data and run the computation. Each slave has Data node and Task tracker daemons to communicate and receive instructions from the master node. Task tracker daemon is part of Job tracker while Data node is part of Name node.

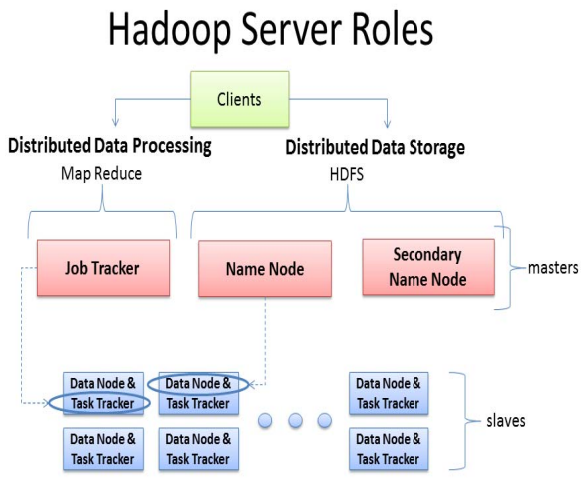

Fig 1 Hadoop Server Role. [8]

D. MapReduce

MapReduce is a framework introduced by Google in 2004 both a programming model and connects the implementation of processing and generating large data sets. This framework is inspired from the concept of Map (sort) and Reduce (collect) which is used in functional programming [9].

Figure 2 illustrates how the MapReduce perform matrix multiplication. Matrix is broken down into several logical chunks, each chunk is processed by computer separately with the use maps-task called a mapping. The results of each process are performed by computers partitioned into several distinct sets, which are compiled. Any chunks that have are arranged to distribute to the Reduced the task is called a reducer [10].

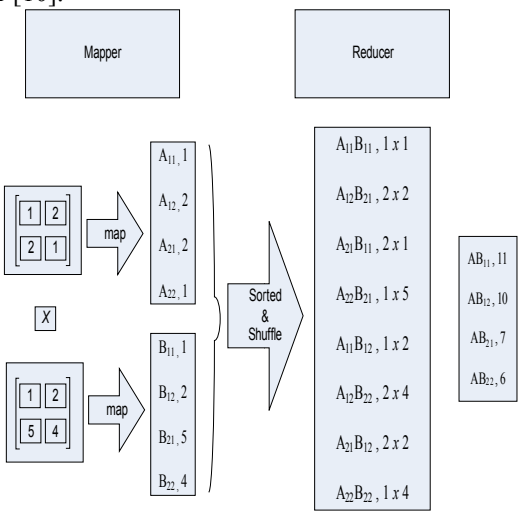

Fig 2 MapReduce

3. SOFWARE DEVELOPMENT

This research will use three models of the matrix multiplication as follows: Single Procesor, Threads, and MapReduce. Software development will be done in three stages, namely: design input, data matrix multiplication, and result.

A. Create Input

Matrix A and B have the means or rectangular dimensions n x n. In this study the authors use a matrix with dimensions: 500 x 500, 1000 x 1000, 1500 x 1500, 2000 x 2000, 2500 x 2500,and 3000 x 3000, respectively. Matrix formed from the real numbers, from 1 through 10 are shaped into text file. Each dimension will be in two different files that describes the form of A and B, here are signed with the 500A with 500B, and so on.

B. Processing of Data Matrix

Matrix multiplication program with the method of MapReduce can be downloaded from the following sites: http://trac.nchc.org.tw/ downloaded on 22 March 2011. In the program, authors insert class read sequence matrix and write sequence matrix so that the matrix text files can be converted into binary form. This is done to facilitate insertion operasioanal experiments.

a. Single Processor

In the process of multiplication by the matrix multiplication method, Single Processor runs according to a

given queue until the multiplication is completed and it provides output. Queue forms are described in Figure 7.

b. *Threads*

Multiplication using Threads is one example of form of parallel computing by using the number of available processors. Thread is the smallest unit of execution of a program. Thread executes a series of instructions one by one. When the system is running the program, the computer will create a new thread. (Thread in this context referred to the process). Instructions in the program are executed by this thread in sequence, one after another from beginning to end [9]. The thread is "dead" when the program finishes the execution.

In modern computer systems, multiple Threads created at a time. Programs run on computers with multiple processors and hyper-threading. Each processor performs a different thread so the thread will be used a lot of speed up program execution, because each thread runs separately. The advantages of multi threads include increased responsiveness of the user, resource sharing process, economical, and the ability to take advantage of multiprocessor architectures.

c. MapReduce

MapReduce matrix multiplication method is multiplication form, which distributes the multiplication of each segment to a computer that resides in its cluster (parallel computing). MapReduce works with the function key and value. Matrixes where each entry will be given a key those facilitate the distribution of value that will be executed.

Figure 3 is a flow chart of a method MapReduce. From figure 3 we can see the difference with the previous method of addition of the Map and Reduce. Where the key is the position matrix and the value is the value of the entry of the matrix. Provision of key and value are the marking in the distribution process.

Flow chart of the method of single processor and threads do not have the Map and Reduce tasks. Scheme is no different with the passage of MapReduce programs.

C. Getting the Output

Output gained a text file that contains the time needed to complete the calculation of the matrix and the entries of the matrix C = A x B. Apart from the matrix file also contained the output from Sar file that provides information processor usage percentage. Both of these files are separated for different applications. The Sar application records of variable information such as CPU, memory, network, I/O, transfer statistics, and others are saved into a text file that are easy to analyze.

The process of getting the output on the method of single processor and threads together, the output will be printed after all the multiplication and addition. While on the multiplication of hadoop distributed systems, job tracker completed distribution of tasks to all Tasktracker and process matrix multiplication and add them up chunks of work which are done by the task tracker, task tracker sent the results to the job tracker. Furthermore Job tracker will sort/organize based on its key and save the file on Name node AB matrix.

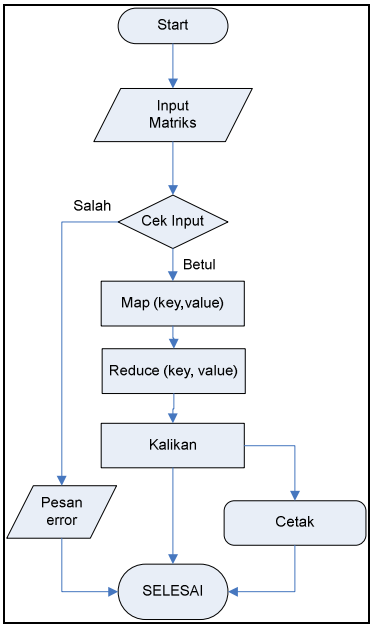

Fig 3 flowchart MapReduce

4. RESULTS AND ANALYSIS

Data retrieval (Execution time, CPU Usage, Memory Usage) was obtained by performing matrix multiplication on the dimensions of 500, 1000, 1500, 2000, 2500, and 3000, respectively. The data used in this study that come from the Matrix Market [10] can be downloaded from http://math.nist.gov/MatrixMarket/.

Data (CPU Usage and Memory Usage) obtained by running sar on the computer program is stored into text file, while the Execution Time is obtained by inserting a timer function in the program run. Applications sar (systat) can be downloaded from www.systat.com

A. Results from Pilot

A trial was made as many as 10 (ten) times the matrix multiplication for each dimension in order to aqcuire the average time required to complete the multiplication.

a) SingleProcesor

Data from Table 1 is the average value of 10 attempts with a single processor method. The data consists of the execution time, CPU usage and memory usage.

Table 1 Execution Time, CPU and Memory Usage

No	Dimension	Time (Second)	CPU (%)	RAM (%)
1	500	3.2	42.2	51.8
2	1000	30.2	61.2	54.0
3	1500	112.3	58.6	57.1
4	2000	280.7	53.1	62.5
5	2500	540.0	51.6	70.6
6	3000	973.6	51.0	92.0

b) Threads

As can be seen that data from Table 2 is the average value of 10 attempts with Threads method. The data comprise of the execution time, CPU usage and memory usage.

Table 2 Execution Time, CPU and Memory Usage

No	Dimension	Time (Second)	CPU (%)	RAM (%)
1	500	1.9	63.8	81.4
2	1000	13.3	79.4	83.6
3	1500	29.4	92.3	86.2
4	2000	72.3	94.9	89.9
5	2500	142.0	97.6	95.7
6	3000	973.6	51.0	92.0

c) MapReduce

Data from Table 3 is the average value of 10 attempts with MapReduce method. The data is divided by the execution time, CPU usage and memory usage.

Table 3 *Execution Time, CPU and Memory Usage*

No	Dimension	Time (Second)	CPU (%)	RAM (%)
1	500	50.2	17.3	90.5
2	1000	121	17.1	73.8
3	1500	156.1	27.7	72.9
4	2000	364.8	31.3	83.1
5	2500	543.1	37.8	71.4
6	3000	770.3	41.3	78.8

B. Data Analysis

Analysis of matrix multiplication is done by three methods: single processor, thread, and mapreduce using a variable execution time, the percentage of CPU usage and memory usage as well.

a. Execution Time

Execution Time is the amount of time required by the program (matrix multiplication) to execute to completion [11]. Execution time of each method will be compared, so that it can be concluded with a method that has a faster time.

Figure 4 illustrates the comparison of the consumption of time needed to complete the matrix multiplication. From Figure 4, it can be concluded that the matrix multiplication using multiplication threads consume less time than others. This Threads method set of instructions execute one after another in sequence so that shorten the time is required. However, it would also affect the excessive consumption of resources.

In the method of single processor, execution time was very large, this is because the calculation of the matrix only performed on a single processor and the use queue method. This means that the matrix multiplication AB12 will be made after executing the matrix AB11 multiplication.

Where as the time required in MapReduce method also very large border or equal to the single processor method. From this study, the author tried to conclude the thread method execution time that is faster than other because of no traffic cost like MapReduce at the time of sorting methods (mapping) and combine (reduce) data.

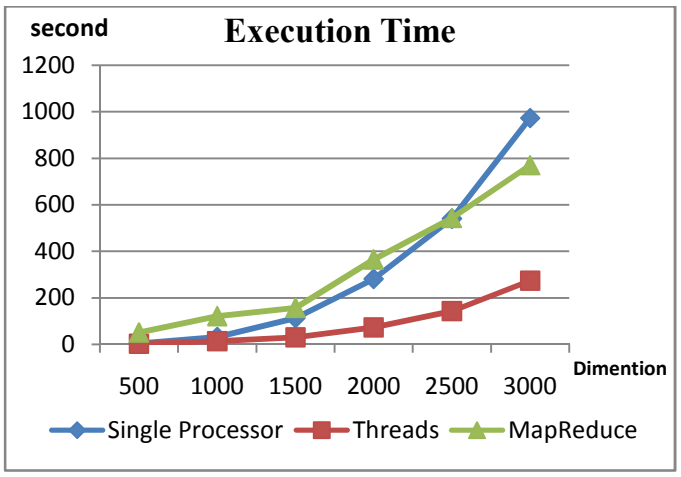

Fig 4 Comparison graph Execution time

b. CPU Usage

CPU usage is the amount of CPU consumption used to solve the matrix multiplication in all three methods. On the single processor the percentage of CPU looks so constant in number by an average of 50%, this is because an application Sar record data and provide output by adding a second processor owned by a computer. If we do otherwise to divide by 2 processor then we will obtain a huge figure on average and this method will take 90% of CPU to run programs.

Thread method, kind of parallel computing, is going to use all of available processors on the computer, so the big record data with a percentage are going to continue by increasing size of the matrix multiplication dimension.

Figure 5 describes the percentage of processor usage during the process of matrix multiplication, which is executed. As shown in the graph, the multiplication by using a method very large CPU Threads has average of almost 95%. Compared to another method, this method is very wasteful threads in resources. The results from Treads method that executes a series of instructions one by one in sequence so that it becomes very much the execution of the multiplication with the hope that can be completed in a short time, as shown in Figure 4 where the thread using faster matrix calculation is completed.

Single processor, in the opposite method, has an average power consumption is on 50%. This is because matrix multiplication is done by only one processor and is a queue. It can be seen by monitoring applications that are available to use it on the operating system.

Processor consumption is very low on the MapReduce method an average of 35%. This is because MapReduce with mapper system divides the execution of matrix multiplication to idle processors on Hadoop cluster. Then the MapReduce method is more efficient in CPU usage compared to other methods.

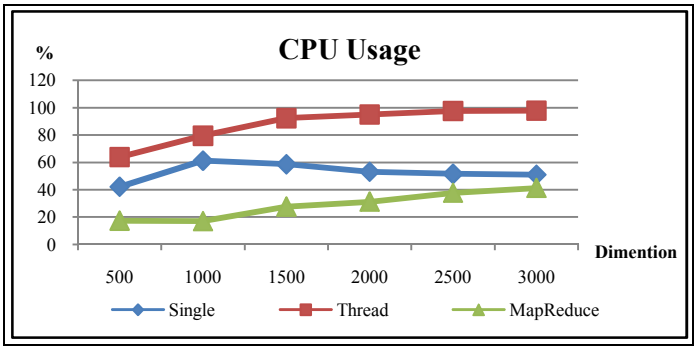

Fig 5 Comparison graph CPU Usage

c. Memory Usage

Memory Usage is the amount of memory used to store data during the process of matrix multiplication (program) running. So the multiplication result stored permanently or removed. The amount of memory consumption will be compared to the third methods.

As depicted in Figure 6, there are the comparison of memory usage for the matrix multiplication process ran. On a single processor, the matrix multiplication based on queue so that the memory consumption is gradually increased by the amount of dimension to the limit of memory capacity.

Threads method would need very large memory because the method Threads execute sets of instructions one by one in sequence and store them in memory for a while, this will be resulted in the exhausted memory, when we raise the dimensions of 3500 turned out to be the multiplication of resources (RAM) of that computer limited so the multiplication is canceled due to insufficient memory.

In the MapReduce method, which is shown in Figure 6, the average memory consumption is 80% of memory capacity, because MapReduce works with HDFS file system to store these files. Those are not distributed on the RAM. As a result, the consumption is only used to complete the multiplication only.

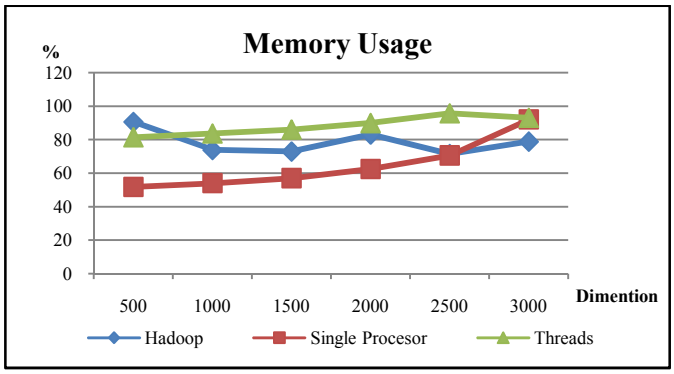

Fig 6 Comparison graph Memory Usage

5. CONCLUSIONS

The research has found some characteristics on the performance of three computation approaches, single processorm threads and map-reduc as follows:

1. The use of thread approach in matrix multiplication has reduced the execution time. The required time is the shortest compared to the other methods. However, it required large computer resources.
2. Matrix multiplication using a single processor needs longer time and a great resources.
3. While using MapReduce, it takes a long time to complete due to traffic cost among the computer, but the use of CPU is lower than the others.

As a result, MapReduce methods should be considered as a method for large and complex compution.

REFERENCES

[1] Anton. Howard, 1987, *Elementary Linear Algebra. Fifth edition*, Anton Textbooks, Inc.

[2] Ciegis. R., and V. Starikovicius, 2003,*Realistic performance prediction tool for the parallel block LU factorization algorithm,* INFORMATICA, Vol. 14, No. 2, 167–180

[3] Dean. Jeffrey and Ghemawat. Sanjay, 2004 *MapReduce: Simplified Data on Large Clusters,* Google, Inc.

[4] Lammel. Ralf, 2007, *Google's MapReduce Programming Model –Revisited,* USA, Microsoft Corp.

[5] Hedlund. Brad, accessible 7 October 2011, *Understanding hadoop Cluster and Network, bradhedlund.com*.

[6] Sascha. Hunold, Thomas. Rauber, GudulaRÃnger, 2004, *Multilevel hierarchical matrix multiplication on clusters,* Proceedings of the 18th annual international conference on Supercomputing.

[7] Seo. Sang won, and friends, 2010, *HAMA: An Efficient Matrix Computation with the MapReduce Framework,* Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference. CS/TR-2010-330.

[8] Tinetti, F., Quijano, A., Giusti, A., Luque, E., 2001, *Heterogeneous networks of workstations and the parallel matrix multiplication,* Proceedings of the Euro PVM/MPI 2001, Springer-Verlag, Berlin, pp. 296-303.

[9] Venner. Janson, 2009,*Pro Hadoop*, USA, Appress.

[10] White. Tom, 2009, *Hadoop the Definitive Guide*, USA, O'Reilly | Yahoo! Press.

[11] William.Gropp, Ewing. Lusk, Nathan. Doss, and Anthony. Skjellum, 1996, *A high-performance, portable implementation of the MPI message passing interface standard,* Parallel Computing. 22, 6, 789-828. DOI=10.1016/0167-8191(96)00024