ISSN: 2597-8993

Multithread to Accelerate Process Data Sync Using MapReduce Model Programming

Murti Retnowo

Department of Informatics Management, Faculty of Information Technology and Electrical Engineering,
University of Technology Yogyakarta, Indonesia
Murti.retnowo@Staff.uty.ac.id,

ARTICLE INFO

Article history:
Received December 28, 2017
Revised on January 27, 2018
Accepted February 28, 2018

Keywords:
MapReduce, the data synchronization, multithread, Message passing Interface, Static Tasks assignment

ABSTRACT

Research in the processing of the data is the data shows that the larger Increasingly requires a longer time. Processing of huge amounts of data on a single computer has limitations that can be Overcome by parallel processing. This study utilized the MapReduce programming model synchronization by duplicating data, data from database client to database server. MapReduce is a programming model that was developed to speed up the processing of large data. MapReduce application models on the training process performed on the data sharing that is adapted to the number of sub-process (thread) and data entry into the database server and display data from the synchronization process. The experiments were performed using 1,000, 10,000, 100,000 and 1,000,000 data, and use the thread as much as 1, 5, 10, 15, 20 and 25 threads. The results of this process showed that the use of the MapReduce programming model can result in faster but require a longer time to create many threads. The results of use MapReduce programming model can provide more efficiency time in processing synchronizing data, both on a single database or a distributed database

I. Introduction

The need for massive computing capabilities at this time to perform data processing on a large scale and a short time became very fundamental things in order to maximize and achieve the objectives to be achieved within the company. Some fields that require high-end computing, simulation, calculation and data processing. Such problems often require repetitive calculations on large amounts of data to obtain valid results. Besides computing system should be able to resolve the problem within a reasonable time.

Traditional computing that has a single processor to perform the duties of a program is one way to improve performance in the data processing. However, with a single processor still requires a long time, and this is because the computer will retrieve the data one by one from each server data residing in different locations. In fact, companies often demand that simulation, calculation and data processing can be completed in minutes and even seconds. Simulations were completed in a matter of days usually unacceptable.

The time required to perform the simulation, calculation and data processing should be as short as possible so that information can be received in a short time and accurately. There are some issues that have a window of time in processing computation. An example is the processing of large amounts of data in the server where the data is located on several different computers so that the program requires a long time to carry out the process of synchronizing or merging data from each computer [11]. Based on the background of the problems described above, the formulation of the problem in this research is: synchronizing data to provide information quickly and accurately by utilizing an existing computer on each server data using the MapReduce programming model. Issues to be discussed in this study had a fairly wide scope so that the problem is limited as follows:

a. MapReduce process is done on a local network with the specifications of a different computer (grid).
b. The computers are used to form a grid or as a node or a worker is a few computers connected in a network and stay in one switch (local network).
c. The method used Message Passing Interface and Static Task assignment coupled with the MapReduce programming model.

[5] doing research on how to analyze a Unique identity of every resident on the ID card to avoid duplication of ID cards at the time of manufacture, the database stored sized Very Large Database will require a Framework

DOI:

MapReduce and a data warehouse in order to analyze Identity population that is stored on a database. Data is exported into Text format then do the mapping process [6]. [7] conducted a study to find the Linux mailing list members in Indonesia with MapReduce. MapReduce function for searching for data members, additions, deletions, see the topic mailing list and run the data nodes on a Linux-based operating system.

II. RESEARCH METHODS

The need for fast and accurate information becomes important at this time. Information needed by the management to take decisions in the production process and raw material purchase the next period. To get information fast and accurate is needed computers with hardware specifications that support for data processing fast and high level of performance and programs based on client-server with distributed databases and can be accessed from all the computers in a local network or the public network is one way of increasing efficiency in data access time [10].

Substitution computer hardware course is very helpful in increasing data processing speed but by the turn of the computer hardware cost, a lot and the program that has been running well is not necessarily in support of new computer hardware. It also calls for changes to the settings of the new computer so that the program can run well [9]. Changing the existing program at this point would require a long time because the source code of a program is not known where the most final and how the flow of the running programs, this is because the programmers who create the programs already moved new workplace. Creating a new program that would take much longer for a new programmer should be able to understand the workflow of programs running at this time and implement the new program, requiring a long time to learning a new program for users and error-testing program to determine errors that exist and will disrupt the work process that is already well underway.

The intensity of the arrival of high raw material and the location of the demolition of the less comprehensive result in the location of unloading come and data processing of transactions carried out in some places if the data processing of transactions carried out on the spot with the current program is certainly going to take a long time to enter the data so that data processing transactions are conducted in several places in accordance with the location of unloading come[2]. The arrival of high intensity which also resulted in vast amounts of data and the large database for storing transaction data daily.

In addition to the problem of computer hardware and programs that are running are transactions or data processing that is performed in many different locations and the database that caused the program is still running a stand-alone. Transactions occurring in several different places in the program is still running stand-alone course will lead to its own problems in the process of data synchronization while already using a distributed database, in this case, using a database MySQL, but MySQL is installed on each computer with using the settings that the user can only access from one place (local computer). One trend that is currently used in IT companies a great deal in large volumes of data and distributed is by MapReduce programming model [3], where there are two processes, namely the processes folder where data will be divided into equal parts and processes Reduce where data will be merged back and eliminated if the same data from previous mapping process. MapReduce process will be run in a cluster or a grid made up of several independent computers to process simultaneous or parallel (multithreaded) used to perform the data synchronization, import data or search data on the distributed database [1].

MapReduce programming model combined with the method of the Message-Passing Interface (MPI) and Static Tasks Assignment (STA) and is used in the programming language Delphi apart expected to improve system performance in handling data in a large size and distributed also able to increase the efficiency and effectiveness of time work in the process of synchronizing data from many database client to the database server.

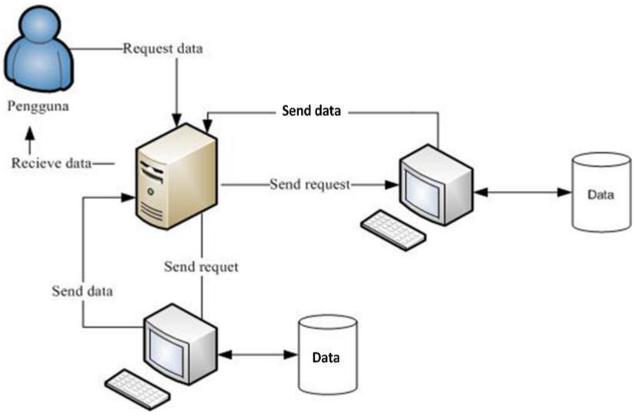

Fig. 1. Description of the system

The system will be built is a system consisting of a server computer that serves as a data server, data source, node worker, and chat servers are enabled to distribute requests data, requested can created on client computer or computer server and others put in a network with some client computer that serves as a data processing node worker, chat client, and as a source of data (resource).

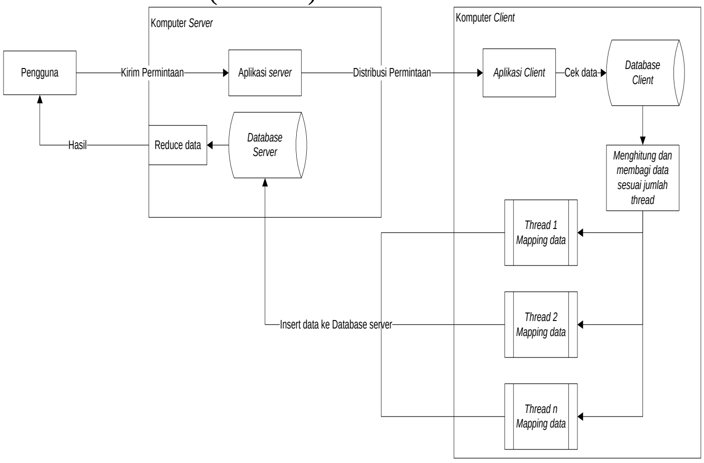

Fig. 2 The process of data analysis

Description of the system, in general, started with the process data requests from users. Process requests and data delivery as shown in Figure 1 where there is a user performs a data request, the data request will be sent to the server computer. The computer server that accepts requests for data automatically distributes data requests to each client computer to a server computer here already finished his job. The next process occurs on the client computer. Upon receiving a request further data client computer will prepare the requested data in accordance with the criteria that have been obtained from the server computer, the next process is to conduct mapping data or collect data that will be made to the database server synchronization process in parallel (multithreaded). Data synchronization process has been completed will be accessible to users who request such data and conduct the process to reduce or eliminate the same data with certain

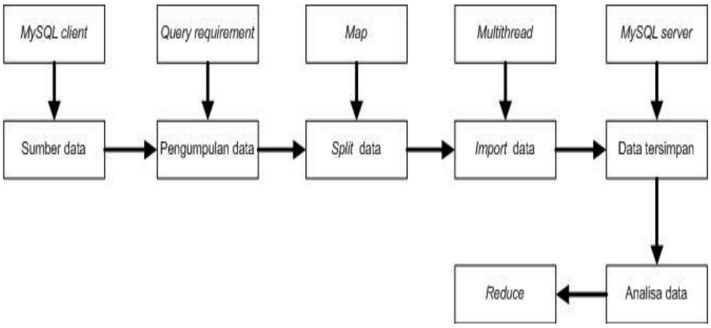

Fig. 3. Flow data transmission process

The stages of data analysis can be seen in Figure 2. The process begins with the delivery of data requests made by the user by making criteria such as date restrictions and the number of threads to perform the synchronization process, criteria for the data request will be sent to the server computer, the server computer after receiving criteria data query criteria will distribute the data request to all client computers contained within a single network. The client computer has received the data criteria that will perform data collection and sharing of data in accordance with the number of threads and synchronizing the data to the server computer.

III. RESULTS AND DISCUSSION

Training results used to determine differences in the speed of data synchronization is done with conventional single process compared with the MapReduce programming model. Which in turn indicates the speed at which obtained from each program and then make comparisons to determine the difference in time of each program. In addition to checking processing data how long a computer can finishing work also tried out the problems that occurred during the creation of a system such as checking the connection, checking ports and user restrictions (user) to a transaction.

The testing process begins with a test to determine the time used to make a thread, data processing 1,000, 10,000, 100,000 and 1,000,000 of data that will be synchronized with the number of processors, 1.5, 10, 15, 20 and 25 on each -masing amount of data.

A. Making Thread

Thread or lightweight process (LWP) is a basic unit of CPU Utilization containing a program counter, a register set, and stack space. The thread will cooperate with other threads in the use of the code section, data section, and a resource of the operating system are collectively (task)[8]. MapReduce is a programming model that is distributed and run in parallel (multithreaded) which serves to speed up the processing of large amounts of data[4]. The results of the average time spent to create 25 threads can be seen in Table 3.1. Time used to make thread is influenced by the amount of data, the number of threads that have been created and used computer hardware.

Table 3.1 thread creation time (25 threads)

Computer Name	Creation Time (Seconds)			
	1.000	10.000	100.000	1.000.000
Computer 1	1,52	1,36	2,8	15,96
Computer 2	1,82	1,08	1,72	9,8
Computer 3	1,32	1,96	1,68	20,53
Computer 4	0,68	2,24	5,32	28,24

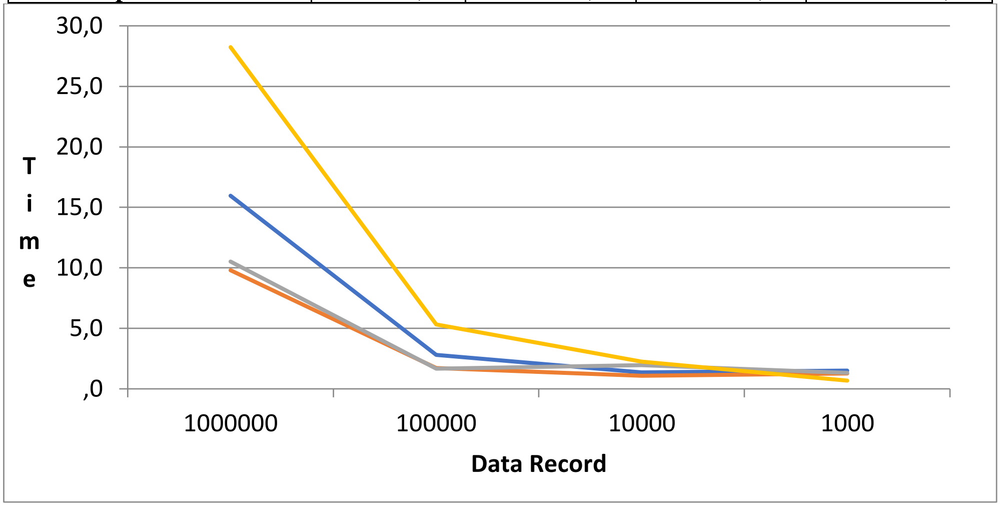

Fig. 4. Graph of created threads

B. *Experiments with Single process*

Testing the data synchronization process is done by using a single process on each computer. Training data synchronization with a single process (single) made of 4 computers with hardware specifications are different and use the same amount of data that exists on each computer. Testing conducted in order to determine the speed of data processing by using a single process.

Table 3.2 Result of data synchronization process with a single process

Computer Name	Processing Time (in Second)			
	1.000	10.000	100.000	1.000.000
Computer 1	34	348	3.600	36.001
Computer 2	36	365	3.694	36.940
Computer 3	36	365	3.693	36.928
Computer 4	36	364	3.693	36.932

Test performed on all computers with different hardware specs. The results of the testing process data synchronization are performed on all computers using a single process (single process) with the amount of data bit (1000 data) shows that the time used to synchronize the data will not be long ranged between 34 to 36 seconds, but when done with a lot of data such as 1,000,000 of data, then the data synchronization process takes a long time ranged from 36 001 to 36 940 seconds or 10 hours 1 second to 10 hours 15 minutes 40 seconds as shown in Table 6.2. Of the time required to synchronize data with the data number 1,000,000 for up to 10 hours then needed a program that is required to perform data processing with a relatively shorter time.

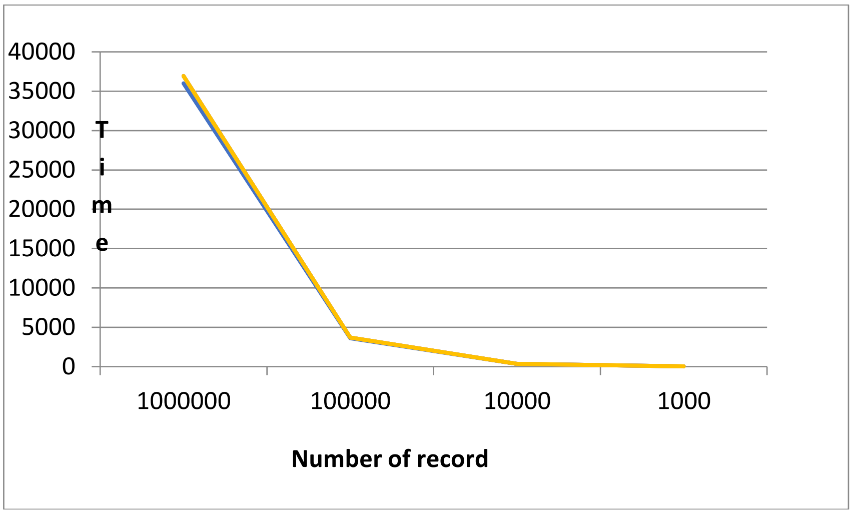

Fig 5. Chart with the single process

C. Tests with MapReduce programming model

1. Mapping Data

On the second test using the MapReduce programming model, combined with Message Passing and Static Task Assignment able to reduce the data processing time of 60% to 70%. The process begins with the distribution of the data according to the number of threads that will synchronize the data, the data mapping process is then performed in accordance with the results of data sharing. Process mapping is done simultaneously with the process of making a thread, each thread will get the same data. Examples of data sharing can be seen in Figure 6:

1. Number of Data	: 1.000.000
2. Number of Thread	: 15
3. Mod	: 1.000.000 mod 15 = 10
4. Data of 1 thread	: (1.000.000 - 10)/15 = 66.666
5. Data thread 1 – 14	: 66.666 data
6. Data thread 15	: 66.666 + 10 = 66.676 data

Fig 6 Example of mapping data

Table 3.3 Average time mapping of data (in seconds)

Name	Data	Number of Thread					
		25	20	15	10	5	1
PC 1	1.000	1,52	0,85	1	0,6	0	1
PC 2	1.000	1,32	0,7	0,73	1,2	1	1
PC 3	1.000	1,28	0,6	0,67	1,2	1	1
PC 4	1.000	0,68	1,15	1	0,9	0,4	0
PC 1	10.000	1,36	1,5	0,93	0,7	1	0
PC 2	10.000	1,08	1,25	1,07	0,9	0,6	1
PC 3	10.000	1,96	1,2	1	0,8	0,6	1
PC 4	10.000	2,24	1,6	0,53	1	1	0
PC 1	100.000	2,8	1,2	1,267	0,8	1	1

PC 2	100.000	1,72	2,15	0,87	0,7	1	1
PC 3	100.000	1,.68	1	1,2	1	0,8	1
PC 4	100.000	5,32	3,2	4,33	3	2	3
PC 1	1.000.000	15,96	15	11,27	8	5	2
PC 2	1.000.000	9,8	8,5	8,13	6,1	4	2
PC 3	1.000.000	10,52	8,7	8,2	6,2	4,6	2
PC 4	1.000.000	28,24	25,6	20,33	17,6	11,4	10

The average yield of the test time required to perform the mapping data and making a thread with a programming model MapReduce can be seen in Table 3.3, wherein the table shows the results of testing the manufacture of thread or mapping data at each computer with a number of different data and the number of different threads. The greater the amount of data to be processed and the number of threads created the more the time needed to perform the data mapping the longer likewise the fewer threads created will require a little time anyway.

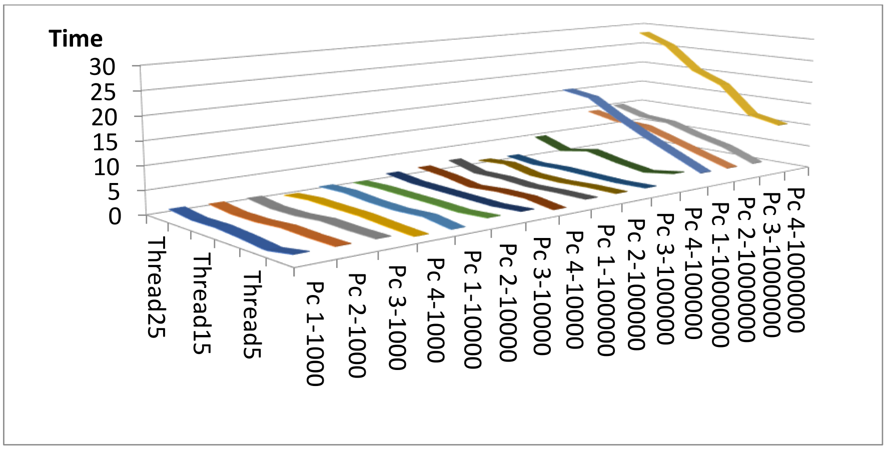

Fig 7. Average Mapping Data

2. *Reduce Data*

Reduce the process data is used to combine the data and eliminate the data in case the similarity of data or duplication of data results in the mapping. Reduce process performed by each computer after the process of mapping and the making of the thread has been created. All the threads that have been created will make the process reduce the data by duplicating data from a client database to the database server.

In Table 3.3 shows the results of data synchronization is done with the programming model MapReduce shows the change in time significantly when using a single process to the data of 1000 using the time for 68 seconds, when using a programming model MapReduce using 5 processor average time required to decline about 6.8%, while using 25 processors fell by almost 90%, while using 20 processors processing time down to 95%, difference processing time between the 20 threads and 25 threads least because many are made to do the processing, the more threads are created takes longer time than the thread a little more.

Table 3.3 Average time reduce the data (in seconds)

Name	Data	Thread					
		25	20	15	10	5	1
PC 1	1.000	6,00	4,00	4,60	6,00	11,00	68,00
PC 2	1.000	4,96	4,00	5,00	5,00	11,00	74,00
PC 3	1.000	3,48	4,00	5,00	5,00	11,00	74,00
PC 4	1.000	5,32	4,00	4,07	6,00	11,00	75,00
PC 1	10.000	30,96	37,05	46,67	60,70	101,40	479,00
PC 2	10.000	31,00	38,00	48,00	65,20	108,60	520,00
PC 3	10.000	30,68	37,90	48,33	64,80	108,20	520,00
PC 4	10.000	30,48	38,00	48,40	63,10	108,40	522,00
PC 1	100.000	351,20	297,20	342,93	407,80	1.989,60	4.361,00
PC 2	100.000	361,24	311,65	357,93	433,50	2.055,60	4.461,00
PC 3	100.000	361,44	309,70	357,13	431,50	2.052,60	4.458,00
PC 4	100.000	356,88	307,70	353,60	433,80	2.051,60	4.460,00
PC 1	1.000.000	2.754,56	3.244,65	3.449,87	4.731,30	9.464,80	47.360,00

PC 2	1.000.000	2.809,80	3.338,00	3.611,07	5.011,50	10.027,60	50.200,00
PC 3	1.000.000	2.796,48	3.316,70	3.592,40	4.995,60	9.994,80	49.970,00
PC 4	1.000.000	2.765,84	3.694,85	3.552,80	5.613,40	11.237.60	56.170,00

The graph in Figure 8 shows the average time it's best seen on computers with hardware specifications of the highest computer (Computer 2) with a time difference that is not too much. The more threads are made the longer the process of mapping data, but the faster the process reduces the data.

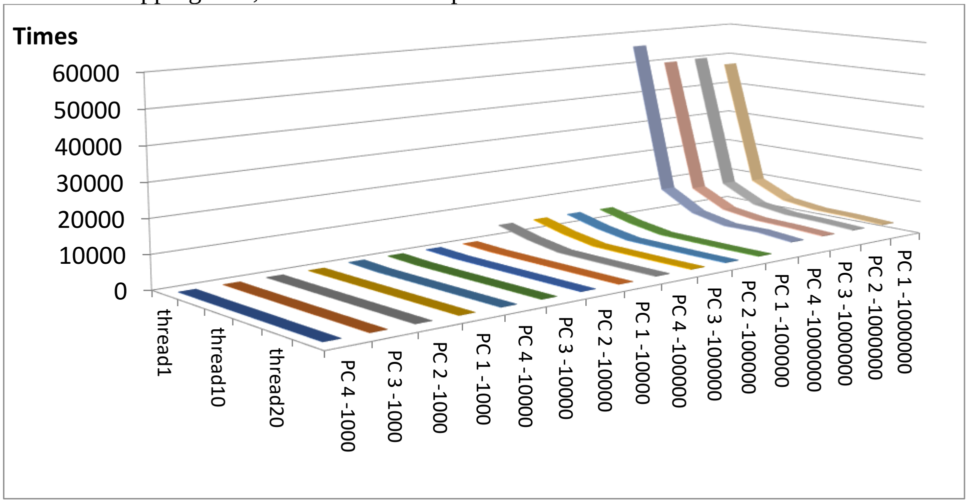

Fig 8 Average results reduce the data (synchronization)

IV. CONCLUSION

The conclusions after conducting research and experiments are as follows:

1. The use of the MapReduce programming model beyond the programming language Java programming and PHP programming can also be done to accelerate the process of data synchronization by performing parallelization of processes (multithread) like using Delphi programming language in 2010.
2. MapReduce programming model can provide efficiency increase time when synchronizing data by duplicating data from a client database to a database server.
3. Based on testing the difference between the amount of processing time by using thread as much as 15 and as many as 25 threads of processing time difference is not significant.

References

[1] Dean, J. and Ghemawat, S., 2004, MapReduce: Simplified Data Processing on Large Clusters, OSDI '04: 6th Symposium on Operating Systems Design and Implementation.

[2] Gaspersz, V, 1998, Statistical process control, Gramedia Pustaka Utama, Jakarta.

[3] Gusti, D. Z. 2012 Distributed Programming with MapReduce Hadoop framework, Final, Ten November Institute of Technology, Surabaya

[4] Handly, M. 2003 3C03 Concurrency: Message Passing, © Wolfgang Emmerich, Mark Handley 1998-2003

[5] Iqbal M. S., 2012, Implementation MapReduce On Very Large Database System, http://library.gunadarma.ac.id

[6] Khairul, F. A., and Arabia, W., 2014, The Net Indonesia Menggunakan Weighting Computing MapReduce. Faculty of Information Technology Institute of Technology

[7] Kurniawan, E. K., 2010, Application Programming Model MapReduce On Linux Mailing List Search System In Indonesia, Department of Informatics, Faculty of Engineering and Computer Science, University Computer Indonesia.

[8] Kusumadewi, S., 2002, the Operating System Issue 2, Publisher Graha Science Yogyakarta

[9] Silberschatz, Galvin, P., and Gagne. G. 2005. Operating Systems Concepts. Seventh Edition.John Wiley & Sons.

[10] Utdirartatmo, F., 2003, Programming Paralel on Linx-based DSM (Distributed Shared Memory), Publisher Andi, Yogyakarta.

[11] Wilkinson, B., and Allen, M., 2005, (Parallel Programming Techniques and Applications Using Networked Workstations and Computers ~ Second Edition, (translated by Hidayat, s., Santosa, YB Hery, AP and Himamunanto, R 2010) Publisher Andi Yogyakarta.