Development of small scale cluster computer for numerical analysis

N H N Zulkifli¹, A Sapit¹, A N Mohammed¹
¹ Flow Analysis, Simulation and Turbulence Research Group, Faculty of Mechanical and Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia
E-mail: azwans@uthm.edu.my

Abstract. In this study, two units of personal computer were successfully networked together to form a small scale cluster. Each of the processor involved are multicore processor which has four cores in it, thus made this cluster to have eight processors. Here, the cluster incorporate Ubuntu 14.04 LINUX environment with MPI implementation (MPICH2). Two main tests were conducted in order to test the cluster, which is communication test and performance test. The communication test was done to make sure that the computers are able to pass the required information without any problem and were done by using simple MPI Hello Program where the program written in C language. Additional, performance test was also done to prove that this cluster calculation performance is much better than single CPU computer. In this performance test, four tests were done by running the same code by using single node, 2 processors, 4 processors, and 8 processors. The result shows that with additional processors, the time required to solve the problem decrease. Time required for the calculation shorten to half when we double the processors. To conclude, we successfully develop a small scale cluster computer using common hardware which capable of higher computing power when compare to single CPU processor, and this can be beneficial for research that require high computing power especially numerical analysis such as finite element analysis, computational fluid dynamics, and computational physics analysis.

1. Introduction
Cluster computer can be understand as a set or a number of computers that connected to work together and can be form as a single system. As stated in previous research, cluster computer has better performance than single unit computer from time consuming, data storage, data processing, and many more [1]. Therefore, when these computers act as one system, it can be used to solve a problem much faster compared to single unit computer. A study was conducted and the aim of this study is to develop a cluster computer in small scale which involved two units of processors that multicore [2]. It contains four cores each, thus make this whole cluster of two computers have eight processor. These together will form a single cluster unit that will have greater processing power. The cluster computer will provide larger storage capacity, high speed of processing and wider availability of resources. Therefore, this cluster computer can calculate and generate result in a shorter time when compared to single unit computer.

Since numerical analysis require much time especially on difficult and large area problems, time is an important resource. All of this work will surely take longer time to be complete without having access to high computing power. Dedicated custom high performance computer is expensive. Such the
idea of using currently accessible computing power (available desktop) and by networking and clustering it together, we can develop a system of clustered computer that can combine their computing power, without any expensive cost [1]. Therefore, two main objectives of this study are to develop the cluster computer system and to evaluate the performance of parallel computing system.

The development of this system will use the ready existing technology with the open source software LINUX installer and the parallelization by MPI program [3]. The writing of the coding will be done by using FORTRAN software. The performance of the system will be test by solving a Laplace Equation in a Square Domain by Jacobi Iteration problem as the benchmarking.

2. Methodology

In order to build a parallel computer, a few steps are needed. Here, the procedure or the steps are divided into five parts which start from Ubuntu installation until the testing part at the end. In this project, both computers are in the same brand, same system and processor type. One of these computers has to become the master and the other one is slave.

All the components were set up and combine to form a complete set of computers with two monitor, 2 multi-core processors and LAN connection using network switch. This setup is currently located at the CEIES Building (Block C19), University Tun Hussein Onn Malaysia (UTHM). In this research, the software’s used are Ubuntu, MPICH2, and FORTRAN. Ubuntu is the operation system (environment), while MPICH2 is a freely available, portable implementation of MPI, a standard for message-passing for distributed-memory applications used in parallel computing. Also, FORTRAN is a well-known programming language.

The MPICH2 will help the Ubuntu or the system to distribute the works or loads to the processors involve. All the distributions are by itself. In order to build a parallel computer, a few steps are needed. Here, the procedure or the steps are divided into five parts which start from Ubuntu installation and will have testing part at the end. All the procedures were followed when the hardware set-up was ready at the working station. In this project, both computers are in the same brand, same system and processor type. Since these computers were from the vendor, both computers are named as UTHM-HP-ProDesk-400-G2-MT from the beginning. One of these computers has become the master and the other one is slave.
The key to parallel processing especially in this case which is the Message-Passing Interface or commonly called as MPI is a message-passing library interface specification. It is stated here that the MPI is specification; which mean it is not an implementation and also not a language. In message-passing, the data is moved from the address space of one process to another process through cooperative operations on each process. Since MPI is specification, this specification is for a library interface and all of its functions are expressed as functions, subroutines, or methods. C, C++, and FORTRAN are the appropriate language binding and are part of MPI standard [4].

MPI is used across the entire spectrum of parallel machines that from small Linux Cluster to the largest parallel machines in the world such as IBM Blue Gene/L and Cray XT3. It has existed for a long time and also has many main advantages such as portability and ease of use. It can provides the vendors with a clearly defined base set of routines that can implement efficiently, or in some cases for which they can provide hardware support, thereby enhancing scalability. Previously, it is stated that MPI is not implementation; so there are numbers of MPI implementations includes MPICH, MPICH2, MVA-PICH, MVAPICH2, LAM, and Open MPI [5].

3. Initial setup and testing
Before the SSH, keygen, and communication setup, the computers were already networked and can communicate with each other validated through “ping” command. The SSH, Keygen, and communication allowed the master computer to log into the slave without any password. The setup start from the update and then downloading the OpenSSH-server.

After the SSH setting, the procedure proceed with NFS Share Folder Setup and MPICH2 setup. Both of this setup had done successfully without any problem. The setting of NFS can be check by using command “ls –a”; which then all files from master that had been mirror and shared to slave are display. In addition, the FORTRAN’s compiler also being download and installed after MPICH2 setup. When the MPICH2 setup is done, the cluster computer system is ready completely. The system communication is test by using simple Hello World program which utilize MPI capability.

![Figure 2. Result on terminal after running MPI Hello World program](image)

The system is then tested to compare the performance of the cluster computer compare to single computer. The benchmarking was done by solving Laplace Iteration of Temperature (Steady Heat) Case that applied MPI subroutine in order to make use of the cluster capability. The code was the run by varying the numbers of processor available for calculation, (1, 2, 4 and 8). The first test used single processor to run the coding. Then, the number of processor used were doubled each time, until all eight processors are call and used by the MPI. From the time taken to complete the calculation, the performance of the system then analyse..

4. Results and discussion
Table 1 shows the results of time needed to solve the code. When the number of processor used increased, the total local nodes were decrease. The number of total nodes decreased exactly half when the number of processor is doubled. This result shows that the increasing number of processor (doubled) in the system when running the code make the work done by one processor decrease significantly.
Table 1. Result for the performance

| NO OF PROCESS OR npes | TOTAL TIME TAKEN (seconds) | TOTAL LOCAL NODES | TOTAL NX_LOCAL L (x axis) | TOTAL NY_LOCAL L (y axis) |
|-----------------------|---------------------------|-------------------|--------------------------|--------------------------|
| 1                     | 378.2996                  | 262144            | 512                      | 512                      |
| 2                     | 189.4584                  | 131072            | 512                      | 256                      |
| 4                     | 102.0725                  | 65536             | 512                      | 128                      |
| 8                     | 93.9772                   | 32768             | 512                      | 64                       |

Also, the total time taken simultaneously decreased when the number of processor increased. This proved that the cluster computer shorten the time needed for calculation when the processor used were increased, and in this study, doubled. It can be state that with each doubling of computer processor used in the calculation process, time needed reduced about 45%-50%. This is true until the reduction in time is becoming less significant, shown in the case of 8 processors.

Figure 3. Contour graph made using Techplot from the output which indicate the domain and result of the calculation for each processors
From previous research and study, researcher believes that the time taken for cluster computer when doubled the processor numbers, do not perfectly reduced to half is due to overhead from the network traffic.[3] In addition, the 0-5% of the time loss maybe used for the processors, cores, and nodes communicate for distribution of information by MPICH2. Even with 0-5% of overhead from the network traffic or communication, cluster computer still proved to have higher performance and higher calculation capability when compared to single computer. This result is strengthened by the result from four and eight processors that showed the decreasing in time taken.

The contour of single processor showed that the processor run all of the calculation itself hence the result is a contour graph. However, when 2 processors are used, the contours are then divided into 2 sections, 1 section for every each processor. This is achieved due to the MPICH2 which utilize the processor according to the number of processor that user stated in command.

5. Conclusion

Based on this study, cluster computing proved to be beneficial in by providing extra computing power by harnessing collective computing power of individual computer. This provide the means of low cost high performance computing without the need of specialized custom made expensive super computer. Result also shown that works are distributed among the processors that being utilized according to the command. This is achievable by the method of MPI implementation that will make the system utilize the processors and distribute the work.

From the results, it is also shown that doubling the number of processors will reduce the calculation time to about half. Even with 0-5% of overhead from the network traffic and other factors, cluster computer still proved to have higher performance and higher calculation capability when compared to single computer. As the conclusion, the objectives of this study were successfully achieved and accomplished.

References

[1] Pfalzgraf A M and Driscoll J A 2014 A low-cost computer cluster for high-performance computing education IEEE International Conf. on Electro/Information Technology (Milwaukee, WI, USA, 5-7 Jun 2014) (Milwaukee: IEEE) pp 362-366 doi: 10.1109/EIT.2014.6871791
[2] Chai L, Gao Q and Panda D K 2007 Understanding the impact of multi-core architecture in cluster computing: a case study with intel dual-core system Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid '07) (Rio De Janeiro: IEEE) pp 471-478. doi: 10.1109/CCGRID.2007.119
[3] Bierbaum N 2002 MPI and embedded TCP/IP Gigabit Ethernet cluster computing Proc. LCN 2002, 27th Annual IEEE Conference on Local Computer Networks (Tampa, FL, USA: IEEE) pp 733-734. doi: 10.1109/LCN.2002.1181853
[4] Pjesivac-Grbovic J, Angskun T, Bosilca G, Fagg G E, Gabriel E and Dongarra J J 2005 Performance analysis of MPI collective operations 19th IEEE International Parallel and Distributed Processing Symposium (Denver, CO, USA: IEEE) p 8 doi: 10.1109/IPDPS.2005.335
[5] Gabriel E, Fagg G E, Bosilca G, Angskun T, Dongarra J J, Squyres J M, Sahay V, Kambadur P, Barrett B, Lumsdaine A and Castain R H 2004 Open MPI: goals, concept, and design of a next generation MPI implementation In Recent advances in parallel virtual machine and message passing interface vol 3241 ed D Kranzlmüller, P Kacsuk and J Dongarra (Berlin, Heidelberg: Springer) doi: 10.1007/978-3-540-30218-6_19