MOSIX: IMPLEMENTATION, TREND AND CHALLENGES

Ts. Mohd Hakim Abdul Hamid
Technical University of Malaysia Malacca

MOSIX: IMPLEMENTATION, TREND AND CHALLENGES

Hakim Hamid, J. Jais, M. Azman
Department of Informatics
College of Information Technology
Universiti Tenaga Nasional
E-mail: mohdhakim@uniten.edu.my, jamilin@uniten.edu.my, yunusy@uniten.edu.my,
shahrul@uniten.edu.my, hidzadrawks@gmail.com

ABSTRACT
Computing has progressed through four distinct eras, classified according to four paradigms: batched (circa 1960s), time-shared era (circa 1970s), ensured desktop (circa 1980s), and shared network (circa 1990s) [1]. Each progression through the eras has brought us closer to a fully distributed computing infrastructure. This infrastructure is called distributed computing; it includes networked and parallel systems. Clustering is in between parallel processing and distributed computing systems. This paper is part of a larger project on MOSIX cluster computing. This paper mainly focuses on distributed computing implementation. In this paper, the information is presented in a broad to narrow approach, which is from an overview of shared network and all the way to the MOSIX Cluster Management System. A questionnaire was created and distributed to assess the general knowledge of students about parallel processing and clustering in general. An analysis of the survey conducted is presented in this paper. The survey was made on several universities representing IPTA (public university) and IPTS (private university) in Klang valley. The survey also indicated a general overview of MOSIX operating system among students in Klang valley area.

KEYWORDS
MOSIX, Distributed Computing, Parallel Computing, Clustering

1. Introduction

Before a product or service can be created, comprehensive research on related topics and issues should be conducted beforehand. Excellent quality research is needed to provide a solid foundation before any work related to designing and implementing can be done. If the research done prior to implementing the project is shoddy, it is practically guaranteed that the amount of difficulties that occur will be extensive. In regards to this project, research is crucial to first comprehend parallel processing, then to understand how MOSIX works and its features, subsequently to design a cluster, to implement it, and finally test the system. Conducting a thorough research was especially crucial for this author because of the lack of any significant knowledge, but only interest in abundance, of the matters stated above.

To gain an overview and information of the project as a whole, proceedings, journals and books were the most helpful. Parallel processing has been around for quite awhile and there were many books by respected authors that helped in getting a good grounding of the history, the evolution and operation. While looking for relevant material on MOSIX, all the information gathered was from the World Wide Web. The MOSIX homepage at http://www.mosix.org was especially helpful for general information and for technical details.

1.1 Distributed Computing

Usually, a distributed computing architecture consists of lightweight software agents installed on client systems (for example: PCs), and dedicated distributed computing management servers. Distributed computing works when the agent running on the client system detects that the client is idle, notifies the management server that the client is free for processing, and requests an application package. The client then receives the application package from the server, runs the software when it has spare CPU cycles, and sends the results back to the server. The application runs in the background of the client system and does not have an effect on normal use of the computer. If the user of the client system needs to run any of his own applications, control is immediately returned, and processing of the distributed application package ends [2].

An important fact to know is that not all resources of a distributed system are devoted to solving a single problem. A significant fact that distinguishes a distributed system from parallel processing is that it is wide ranging and can work on many problems simultaneously. All the computers connected within this system are able to function independently, but can share their data. The most popular application of distributed computing is SETI@Home; which analyzes data from a telescope to search for potential signs of extraterrestrial life. In the commercial world, distributed computing is called client/ server computing. The overview is now slightly narrowed down to parallel computing in the section below.

1.2 Clustering

Clustering is a sub-paradigm of parallel processing and distributed computing. Networked computers, on a local level, work together just like a parallel machine. Although the work is split across multiple computers, they behave in a parallel fashion.

To put it more clearly, a definition of a cluster is a type of parallel or distributed system that:

a) Consists of a collection of interconnected computers,
b) Is used as a single, unified computing resource [3].

For this project; a cluster will be developed that will utilize parallel processing. As also stated previously, parallel computing is not widely used because due to the difficulty of programming for parallel machines. This is where the MOSIX Cluster Management System comes in.

1.3 MOSIX Cluster Management System

MOSIX is a cluster management system that can make an x86 Linux cluster run like a single high performance parallel computer. It is particularly useful to run intensive computing and massive I/O applications. In a MOSIX cluster there is no need to modify or to link applications with any library, or even to assign processes to different nodes, MOSIX does load balancing automatically - just 'fork and forget', like in a symmetric multi processor (SMP). MOSIX supports different cluster types, including clusters with different CPU speeds. It is best for running a large number of resource hungry processes and parallel applications with unpredictable resource requirements and/or execution times [4].

Consider a heavily-used University server that has many simultaneous login sessions and an extremely high load. MOSIX is ideal for these types of situations, since the high load and large number of processes means that MOSIX will have many opportunities to distribute the load among nodes in the cluster.

The example above also manages to show the benefits of clustered parallel processing:

a) Performance: A task that has many processes can be completed in a much shorter time.
b) Availability: It allows the computers in the cluster to back each other up in the case of failure. When a computer fails, the other computer picks up the workload of the failed computer.
c) Scalability: Usually when the performance of a computer is unsatisfactory, it is replaced by a newer and faster version. But by using this system, another computer can be added to the cluster to improve performance.
d) Productivity: Tasks and operations are completed more efficiently with the added performance and availability of the system.

e) Price/ Performance: Eugene Brooks of the Lawrence Livermore National Laboratory (famed laboratory in the United States that concentrates on energy issues) was quoted saying, "One computer center after the other is lining up to put a cluster of microprocessors on the floor. Their budgets are shrinking and these systems are much more cost-effective than traditional supercomputers" [5]. This proposed system will consist of personal computers, connected by cables and a network switch, and implemented with a parallel processing facility that is distributed freely. The price/ performance ratio of our developed system should be lower than using more expensive computers, such as mainframes and servers, with propriety technology.

Besides that, MOSIX programming facility specifically improves performance by:

a) Load balancing;
b) Moving processes from slower to faster nodes;
c) Migrating processes from nodes that run out of memory;
d) Parallel I/O – by migrating intensive I/O processes to file servers [6].

2. Experimental Setup

This part of the documentation will put in plain words the work procedure involved in constructing the proposed system. This will basically be the basis of the documentation for the System Design activity of the waterfall model which is used as a framework to develop this project. This section of the project explains in general terms the steps involved in constructing the parallel processing system.

2.1 Activity 1: Verify and Install Hardware

All physical hardware needed to implement this project must be made available. The list of needed hardware can be found in Chapter 1 under Hardware Requirements. Besides only acquiring the required hardware, all of these physical components need to be verified that they are in working order and able to function properly. This is done by installing and integrating the hardware. The PCs are connected to form a local area network (LAN) via cables and a network switch. Thus, the PCs are properly physically configured. Other than that, a few simple diagnostics tests are run on the individual PCs are run to ensure they are fully operational.

2.2 Activity 2: Install Linux Operating System

The individual units will be currently running their own versions of the operating system. To follow the requirements of this project, the PCs will subsequently be installed with Linux. The first PC will be installed with Linux RedHat 7.2, while the second one is installed with Linux Suse 7.3. Once installed, diagnostics tests will be run again to ensure they are functioning properly. Figure 1 below shows technical specifications on the proposed cluster.

2.3 Activity 3: Kernel Source Installation and Modification

Before installing MOSIX, a virgin kernel source must be installed into each PC beforehand. In other words, the source is not modified form the original Linux distribution of the kernel. Each version of MOSIX is associated with a specific type of kernel. Another way of saying this is that each version of MOSIX was designed to operate with a specific Linux kernel version. The version of MOSIX for this project will be using the 2.4 kernel. The kernel is obtained from http://www.kernel.org/pub/linux/kernel. This new kernel source needs to be unpacked and installed on each PC. Once completed, the installation of MOSIX can be started.

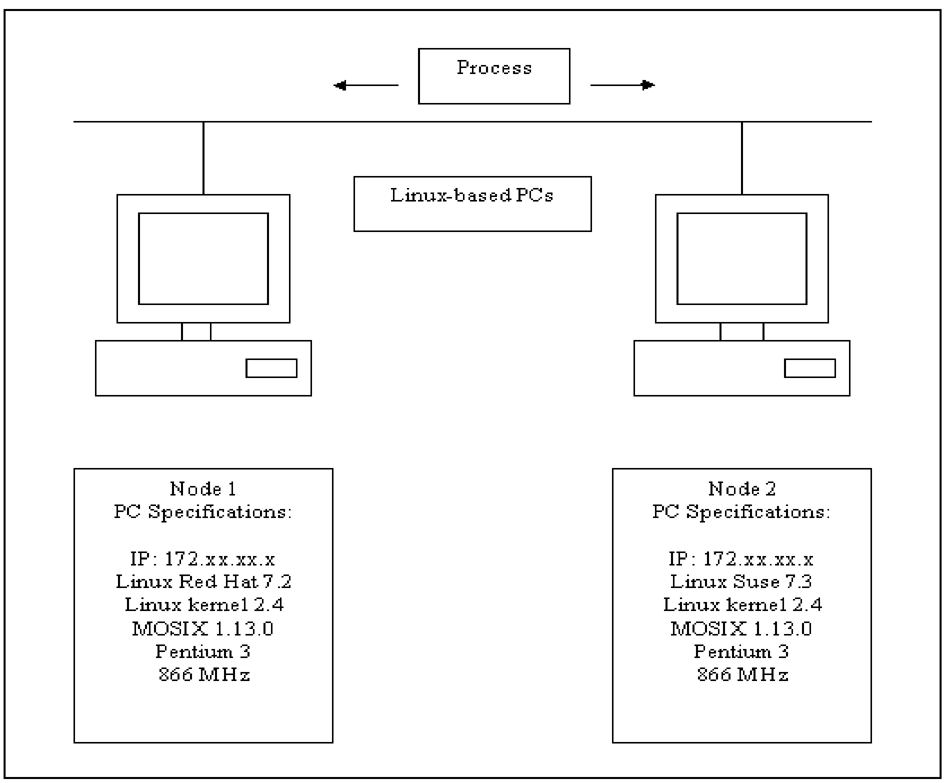

Figure 1 Experimental Cluster Setup

2.4 Activity 4: Installation of MOSIX

The files needed to install MOSIX are downloaded from the MOSIX website in the Download section, at this address: http://www.mosix.org/txt_distribution.html. The installation activity consists of two phases which are the:

a) MOSIX kernel patch, and
b) MOSIX user level tools.

Once downloaded, the first step is to crate the MOSIX kernel by applying the MOSIX kernel patch to the Linux kernel source installed earlier. The next step is to configure the kernel. This must be done on each PC. The configurations applied must be identical throughout the entire cluster. In other words, the MOSIX-enabled kernel is installed on all the Linux-based PCs that form the cluster. Once those steps are completed, the user level tools need to be installed. The user level tools need to be installed in each node. The PCs then need to be rebooted.

After that task is completed, the MOSIX configuration files need to be set up by making the necessary changes to each nodes system configuration files. This step is important to map where each node is by using the IP address or hostname of the node. Besides that, this step will also create the MOSIX file system (MFS). MFS enables a user to access file systems of each node throughout the cluster. Figure 2 below shows a summary of how each of the described activities above executed.

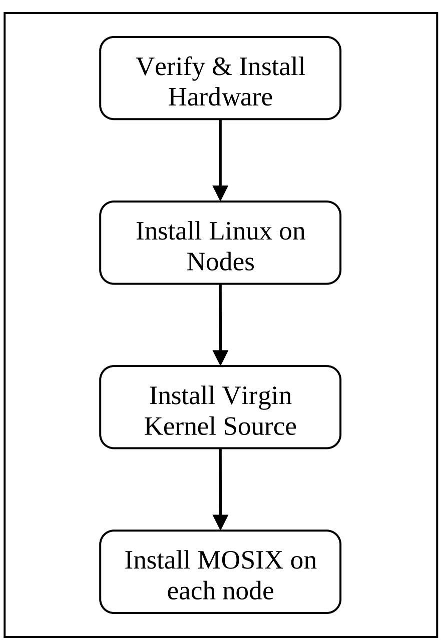

Figure 2 Simplified Work Procedures

MOSIX manages tasks and processes without the users' knowledge. It allows for the execution of sequential and parallel applications without regard for where the processes are running or what the other users are doing. Below, an explanation is given on how MOSIX improves overall performance and its scheduling policy of processes.

Once a process is created from a task given, MOSIX attempts to assign it to the best node available at the time. MOSIX continues to monitor this process and any other processes that might be running, and will move this or these processes among the nodes to maximize overall performance. MOSIX's algorithms, which are used to migrate processes, are decentralized. Each node is both a master for processes created locally and a server for processes that migrated from other nodes. This implies that nodes can be added or removed from the cluster at any time, with minimal disturbances to the running processes.

3. MOSIX SCHEDULING AND TESTING

MOSIX schedules newly started programs in the node with the lowest current load. However, if the machine with the lowest load level announces itself to all the nodes in the cluster, then all the nodes will migrate newly started jobs to the node with the lowest load and soon enough this node will be overloaded. However, MOSIX does not operate in this manner. Instead, every node sends its current load status to a random list of nodes. This prevents a single node from being seen by all other nodes as the least busy and prevents any node from being overloaded.

MOSIX comes with its own monitoring algorithms that detect the speed of each node, its used and free memory, as well as IPC and I/O rates of each process. This information is used to make near optimal decisions on where to place the processes. The algorithms try to reconcile different resources (bandwidth, memory and CPU cycles) based on economic principles and competitive analysis. Using this strategy, MOSIX converts the total usage of several heterogeneous resources, such as memory and CPU, into a single homogeneous cost. Jobs are then assigned to the machine where they have the lowest cost. This strategy provides a unified algorithm framework for allocation of computation, communication, memory and I/O resources. It also allows the development of near-optimal on-line algorithms for allocation and sharing these resources.

The testing plan will be used primarily as a checklist during the Implementation phase of the project to ensure that the construction goes as smoothly as possible.

System testing will be including:
a) Testing on the implementation of MOSIX
b) Testing on system performance (utilizing MOSIX)
c) Testing on system performance (compared to other Parallel OSes)

4. SURVEY RESULTS

A survey was conducted in relation to this project. The questions asked in this survey were associated with the knowledge of respondents on parallel processing, clustered computers and MOSIX. Besides that, other issues were on the usage of Linux, tasks that respondents would most like to be completed quickly and if the respondents had spare PCs that are unused. All of the respondents were students of Computer Science students from Higher Learning Institutions (IPTA and IPTS) around Klang Valley. Below, the results of the survey are broken down based on the question asked. There are number of questions asked in the questionnaires. The first two questions asked in this survey are to gather background information on the respondents. The majority of respondents are studying Information Technology. It can be assumed then that respondents studying Information Technology will have a better understanding of this project as it is related closely to that field. Table 1 below shows the distribution of selected questions taken from the survey.

3.	Clustered Computing can be best described as: **Answers** a. A collection of interconnected computers used as a single resource b. Computers located closely to one another c. A set of computers connected via Local Area Network d. Never heard of the term before
4.	Parallel Processing can be best described: **Answers** a. Devoting the entire system to solve a single problem b. Distributing a workload across multiple interconnected computers c. Making sure that a process is completed correctly d. Never heard of the term before
5.	In your opinion, would usage of Parallel Processing and Clustered Computing complete certain tasks faster? **Answers** a. Yes b. No
6.	Have you ever used the Linux Operating System? **Answers** a. Never b. Occasionally c. Proficient user
7.	In your opinion, can a clustered parallel processing system be constructed by an individual? **Answers** a. Yes b. No
8.	Have you heard of the MOSIX Cluster Management System? **Answers** a. Yes b. No
9.	Do you have any other PCs besides the one that you use that are older or unused? **Answers** Yes No
10.	What is the task that you would most like to be completed more quickly by your computer? **Answers** a. Computationally intensive jobs b. Multimedia related tasks c. Database related tasks d. Networking related tasks

Table 1 Questionnaire on Different Trait of Cluster Computing Technology

Questions 3 until 5 are to assess the respondents' general information on parallel processing and clustered computing. Respondents seem to have a better grasp of the most accurate description of parallel processing when compared to the results of their understanding of their description of clustered computing. Figure 3 shows the percentage of respondents that accurately described the definition of parallel processing and clustered computing. This might be because the term 'parallel processing' crops up more frequently in the respondents education. Besides that, the vast majority of respondents are in the opinion that the usage of parallel processing and clustered computing will complete certain tasks faster.

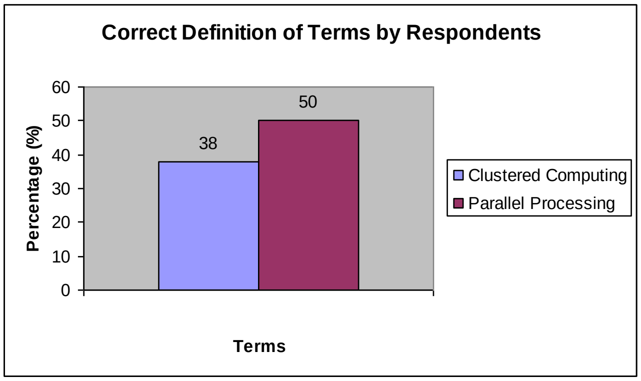

Figure 3 Comparisons of Correct Terms

Questions 6 until 10 relate directly to this project. The matter queried in Question 6 is if the respondents have ever used the Linux operating system. This question was asked as Linux is used as the platform for this project. From the survey, we found out 75% of the target user know what Linux is but never use it. Indirectly, it shows that Open Source Software (OSS) and Open Source Technology are not really implemented by most of higher Learning Institution in Malaysia.

The survey also shows that the majority of the respondents have never used Linux. Only a very small number are proficient using this particular operating system. 23.33% from the respondents do know what Linux is and have some experience in using the operating system. The percentage is quite small; it shows that certain student did use Linux as their secondary operating system but they prefer to use other operating system as their primary operating system. Only a small portion (1.67%) of the respondents can be categorized as an expert user. Perhaps most of them use Linux as their primary operating system. Some perhaps use Linux for research purposes.

Question 7 relates directly to this project. This is because the question asks the respondents if it is possible for an individual to construct a clustered or parallel system. Slightly more than half of the respondents answered that it was not possible. Only 48.8% of the respondents think that MOSIX or other parallel/distributed operating system can be constructed by an individual.

For Question 8, 98.2% of the respondent never heard about MOSIX. The results are not really shocking as MOSIX is not as popular as PVM or MPI. Most probably, as DIPC, MOSIX also is a forgotten useful operating system [8]. It is hoped that this project might somewhat become a starting point for university students and public to know more about the existence of MOSIX. MOSIX is only effective if the resources of two or more PCs are combined. Thus, more than one PC is needed to construct this system. From the survey, 75% of the respondent have additional PC but did not utilize it.

Finally, Question 10 asks the respondents on the type of tasks that they prefer most to be completed faster by their computer. Most of the respondents (61.7%) would like multimedia related tasks, such as playing computer games, to be processed more quickly. The options given in this question are related to this project as once the system is constructed, several testing on certain categories of tasks will be executed in order to assess the system's performance. Figure 4 shows the breakdown in selections by the respondents.

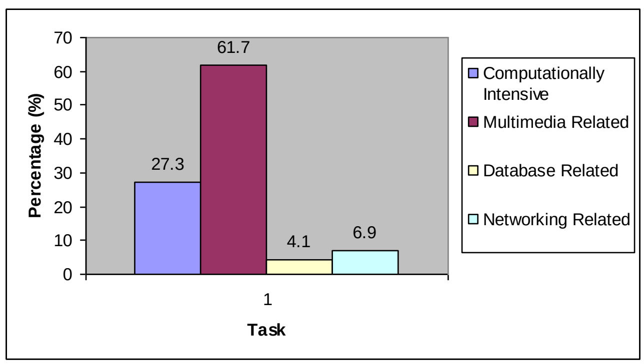

Figure 5 Task Most Preferred To Be Completed Faster

5. Conclusion

MOSIX is not a new operating system. It is a powerful distributed operating system that can work on parallel machine. Depends on the usage and function, MOSIX can be used in distributed environment, parallel machine on stand-alone mode and parallel machine in distributed mode. In this research, we were able to complete certain tasks which is gathering knowledge on matters related to the project such as parallel processing, MOSIX, clustering, Linux and networking, among others. We also have finalized the specification of the hardware and software requirements and are able to determine working procedure for future implementation. There was deep interest in this project and the matters related to it. However a lack of general knowledge on the issues related to this project was a difficulty that needed to overcome. Information on matters such as parallel processing, networking and distributed computing needed to be gained before any work could be done to understand MOSIX. At times, the information and terminology gathered on these topics clashed with each other and had to be analyzed and reasoned. Besides that, the MOSIX technology and how it operates need to be familiarized. Despite all problems, proper planning and execution during the construction of the proposed system will ensure the successful completion of this project.

References

[1] H. El-Rewini, and T. Lewis. "Distributed and Parallel Computing". Manning Publications Co. 1998.

[2] G. Couloris, J. Dollimore and T. Kindberg. "Distributed Systems: Concept and Design (3rd Edition)". Pearson Education Ltd. 2001.

[3] F. Pfister. "In Search of Clusters: The Ongoing Battle in Lowly Parallel Computing. 2nd Ed.". Prentice Hall Inc. 1998. 72.

[4] L. Amar, A. Barak and A. Shiloh. "The MOSIX Direct File System Access Method for Supporting Scalable Cluster File Systems". Cluster Computing. Springer Science + Business Media B.V. Volume 7. 2004. 141 – 150.

[5] J. Markoff. "Bigger, faster hardly matters". New York Times, August 1994.

[6] A. Barak, O. La'adan and A. Smith. "Scalable Cluster Computing With MOSIX for Linux". Proceedings of Linux Expo '99. Raleigh, N.C. 1999. 95 – 100.

[7] A. Barak, S. Guday and R. Wheeler. "The MOSIX Distributed Operating System, Load Balancing for UNIX". Lecture Notes in Computer Science. Volume 672. Springer-Verlag. 1993.

[8] M. Hakim and J. Jais. "DIPC, A Forgotten Useful Elements". The 3rd International Conference on Advances in Strategic Technologies. Kuala Lumpur, Malaysia. 2003.