Parallel Processing - A Case Study of PVM Application Programming Interface

Nurul Athirah Abdul Rahim [#1], Mohamed Faidz Mohammed Said [#2]

[#] *Faculty of Computer & Mathematical Sciences, Universiti Teknologi MARA*
70800 Seremban, Negeri Sembilan, MALAYSIA

[1] athirahrahim9502@gmail.com

[2] faidzms@ieee.org

***Abstract*—Nowadays computer modelling has been used regularly throughout the geosciences in an effort to produce synthetic data for comparison with real data. Parallel programming model itself is an abstraction of parallel computer architecture. Parallel computing itself is a kind of computation in which numerous calculations or the implementation of processes are carried out concurrently. Parallel Virtual Machine (PVM) system uses the message-passing model to let programmers to exploit dispersed computing across wide diversity of computer categories. PVM is an alternative which is cheap, useful and manageable. Besides that, by using PVM it can integrate current departmental services without experiencing additional hardware costs. Virtual machine or also known as VM can appraise enormous expressions and it was written in numerically fast language. Furthermore, this method can express the parallel computation of a single phase space point in an easy and understandable ways. The objective of this research is to study the application and the use of PVM in parallel computing. The result shows that PVM has many advantages and also has some disadvantages.**

***Keywords*: Parallel Virtual Machine, API**

I. Introduction

Parallel Virtual Machine (PVM) is a software platform that allows a varied group of Unix and Windows computers connected by network to use as a single large parallel computer [1]. Regardless of the diverse number of computers that users use and where the computers are placed, PVM is intended to connect computing resources and offer users with parallel stages to run their computer applications. When PVM is properly set up, it is capable of connecting the collective resources of usually varied networked computing platforms to send high levels of presentation and functionality.

The PVM system has developed in the past more than a few years into a feasible technology for selection and parallel processing in a diversity of disciplines. PVM is a sustainable straight-forward but functionally achieved message-passing model. Even PVM computing model is quite straightforward and very common. It can accommodate a wide diversity of application program structures. In addition, large computational difficulties can be answered more cost-efficiently by using the collective power and memory of several computers.

Besides that, during summer of 1989 the PVM was introduced at Oak Ridge National Laboratory. Then, PVM 1.0 version was created by Vaidy Sunderam and Al Gaist. Next, PVM 2.0 version was written at the University of Tennessee and released in March 1991. Then, version 3 was finished in February 1993 and the latest software of PVM has been spread freely and is being used in computational applications around the world.

In addition, application programming interfaces or also known as API is a set of subroutine definitions, protocols and tools for constructing application software. So in general terms, it is a set of obviously defined techniques of communication between several software components (Fig. 1). There are a few types of Web APIs such as Simple Object Access Protocol, Remote Procedure Call (RPC) and perhaps the most popular is Representational State Transfer.

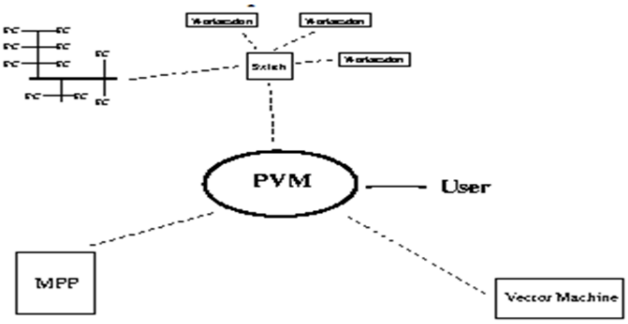


Fig. 1. PVM allows easy use of heterogeneous computer networks

II. Literature Review

According to [2], an API is used for programming applications that begin from scratch or since a remaining sequential version. Fortran and C are presently supported language. Paper [3] found that PVM is a popular domain software which can be found in FORTRAN or C. For each machine architecture, the PVM software administers the setting up of the collections, administers the communication between machines in the collections and the change of the data to accurate format. There are so many writings regarding about the function of PVM. In addition [4], there are two parts of PVM which are a daemon procedure and user library. Any user can install on a machine called as daemon procedure and routines for starting on other machines as well as for interactive between processes and synchronizing the processes called as user library.

Furthermore, there are three major advantages from PVM which are firstly is its high accessibility where PVM can be simply and easily obtained from the web. Next is simple programming in which PVM is easy to use so that programs can be coded in a usual way. Last but not least is its low cost which is PVM only involves doing work by hand and does not acquire extra costs of hardware [5]. However, there is also the an analysed view on the effectiveness of the utilization of available workstation based on use of PVM [6]. Paper [7] found that even though PVM has many benefits, it also has disadvantages. PVM does not back important sending of the existing version which mean the message packages are passing in the similar sequence as their sending request.

III. Methodology

Paper [7] shows that TCP/IP based on PVM can be used as message interface for more accessibility, flexibility and other deliberation. Paper [8] stated that a group of collaborating tasks which are units of parallelism from a PVM program can access PVM assets through a library of standard interface routines.

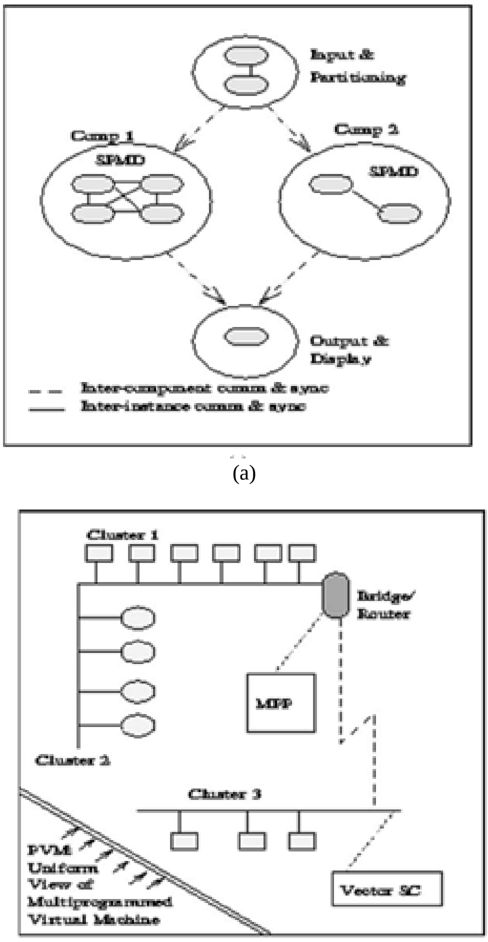
They believed that [9] virtual machines or VMs are produced on underlying physical machines or also known as PMs and performances like actual computer with an operating system. To form a virtual machine, this software uses to combine a set of processors. Moreover, in [10] VM lets the difficulty of the computation to be only set by the existing hardware and not controlled by software project or middle steps. Besides, the development process of a VM scheduling for a homogeneous cloud environment emphasizes mostly on resource running for high performance computing applications. Energy consumption was increased when VMs affect performance excess because of the reduction in the physical servers resources that host the VMs both application runtime and server [11].

Moreover, VM migration [12] is the process of transferring a VM from a host to a different one. The initial VM placement must be frequently reorganized because of modifications in the application load, programmed server maintenance, happened errors, consolidation of servers to decrease energy consumption, over or under utilization of a given host and the need to measure VM resources up or down. VM migrations have to be implemented with some frequency whether VM migrations can be fixed or enthusiastically modified. A lot of problems can happen if incorrect VM placement and migration, resulting in frequently incorrect migrations such as resource consumption by some customers and nonexistence of them to the others. Hence, the most challenging matters in the management of virtualized data centers are the VM placement and migration.

Besides that, paper [9] shows that the server virtualization is one key technology used to enhance resource utilization. The researchers state that [13] the purpose of task scheduling is to produce a schedule and allocate each task to a node or virtual machine for specific time, and also to make all tasks performed in least time span. Task scheduling is NP finish problem in the field of computer science due to the number and length of task transformed very quickly in cloud environment. Hence, other than reducing the span time, this algorithm also can decline the probability of over-loading and under-loading of a virtual machine.

However, other than VM migration it also has VM cluster migration which means services could be migrated to many clouds based on dissimilar restraints. The examples of constraints are computational resources and enhanced economical aids [14]. In a cloud system, making portability to other systems is a very stimulating issue because VMs are arranged by the cloud platform to work under the data center system [14].

In addition, paper [15] shows that PEVPM or Performance Evaluating Virtual Parallel Machine can keep a demonstration of the state of the program and the parallel machine that is finishing. The user can arouse both of the reasons and the consequences of this non-determinism by using PEVPM. Therefore, PEVPM is a suitable method for modelling of message-passing programs as PEVPM methodology is conventional, adaptable and cost-effective (Fig. 2).



(a)

(b)

Fig. 2. PVM system overview (a) PVM computation model and (b) PVM architectural overview

IV. Conclusion

In conclusion, there are many benefits of PVM such as its high accessibility, simple programming and low cost requirement. However, there is an disadvantage of PVM which is PVM does not back important sending of the existing version. Hence, one of the experimental outcomes displays that the implementation of the algorithm running on PVM is quite imposing in comparison with the sequential one. It can be concluded that it is cost-effective to attain high speed computing by consuming parallel algorithm under a PVM environment. So it is better to use PVM in deciding combinational optimization problems.

REFERENCES

[1] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam, *PVM: Parallel Virtual Machine a users' guide and tutorial for Networked Parallel Computing*. 1994, pp. 13-20.

[2] G. Bosilca, G. Fedak, and F. Cappello, "OVM: Out-of-order execution parallel virtual machine," *Future Generation Computer Systems* vol. Vol 18, pp. 525-537, 2002.

[3] C. Niccanna and C. J. Bean, "The use of a Parallel Virtual Machine (PVM) for Finite-difference wave simulations," *Computers& Geosciences,* vol. Vol 23, pp. 771-783, 1997.

[4] C. D. Napoli, M. Giordano, and M. M. Fumari, "A PVM-based distributed parallel symbolic system," *Advances in Engineering Software* vol. Vol 28, pp. 303-312, 1997.

[5] S.-J. Shyu and B.M.T.Lin, "An Application of Parallel Virtual Machine Framework to Film Production Problem," *Computers and Mathematics with Applications* vol. Vol 39, pp. 53-62, 2000.

[6] A. K. Tiwari and K. K. S. b, "Implementation of generalized cross validation based image denoising in parallel virtual machine environment," *Digital Signal Processing,* vol. Vol 14, pp. 138–157, 2004, doi: 10.1016/j.dsp.2003.05.001.

[7] W. Li, X. Huang, and N. Zheng, "Parallel implementing OpenGL on PVM," *Parallel Computing* vol. Vol 23, pp. 1839- 1850, 1997.

[8] N. Yalamanchilli and W. Cohen, "Communication Performance of Java-based Parallel Virtual Machines," pp. 1-6, 1998.

[9] X. Li, Z. Qian, S. Lu, and J. W. b, "Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center," *Mathematical and Computer Modelling,* vol. Vol 58, pp. 1222–1235, 2013, doi: 10.1016/j.mcm.2013.02.003.

[10] B. C. Nejad, Thorsten Ohl b, and J. Reuter, "Simple, parallel virtual machines for extreme computations," *Computer Physics Communications,* vol. Vol 196, pp. 58-69, 2015, doi: 10.1016/j.cpc.2015.05.015.

[11] F. Fernandes, D. Beserra, E. D. Moreno, B. Schulze, and R. C. G. Pinto, "A virtual machine scheduler based on CPU and I/O-bound features for energy-aware in high performance computing clouds," *Computers and Electrical Engineering,* vol. Vol 56, pp. 854–870, 2016, doi: 10.1016/j.compeleceng.2016.09.003.

[12] M. C, S. Filho, C. C. Monteiro, P. R. M. Inácio, and M. M. Freire, "Approaches for optimizing virtual machine placement and migration in cloud environments: A survey," *J. Parallel Distrib. Comput,* vol. Vol 111, pp. 222–250, 2017, doi: 10.1016/j.jpdc.2017.08.010.

[13] M. Kumara and S.C.Sharmab, "Dynamic load balancing algorithm for balancing the workload among virtual machine in cloud computing," *Procedia Computer Science,* vol. Vol 115, pp. 322–329, 2017.

[14] S. Sotiriadis, N. Bessis, E. G. M. Petrakis, C. Amza, Catalin Negru d, and M. Mocanu, "Virtual machine cluster mobility in inter-cloud platforms," *Future Generation Computer Systems* vol. Vol 74, pp. 179–189, 2017, doi: 10.1016/j.future.2016.02.007.

[15] D. A. Grove and P. D. Coddington, "Modeling message-passing programs with a Performance Evaluating Virtual Parallel Machine," *Performance Evaluation,* vol. Vol 60, pp. 165-187, 2005, doi: 10.1016/j.peva.2004.10.019.