Volume 2, Issue 3, March 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Practical Performance Analysis of Modern Software used in High Performance Computing

Rafiqul Zaman Khan[1]
Department of Computer Science,
Aligarh Muslim University, Aligarh.
rzk32@yahoo.co.in

Javed Ali[2]
Department of Computer Science,
Aligarh Muslim University,Aligarh.

Abstract-**The parallel computing without the use of efficient, flexible, portable and interoperable tool is not effective in high performance distributed computing systems. The inter-process communication amongst the various processes during the computation is the major factor for the high speed distributed computing. In this paper, the performance of the most common several parallel processing tools are discussed and compared. PVM is a software infrastructure that emulates a generalized distributed memory in heterogeneous networked environments. Now parallel MATLAB is most widely used computing environment in the distributed parallel computing. MPI is a standardized interface for inter-process communications. This paper gives the valuable comparison of the different parallel computing tools to select the best performance tool amongst the various tools available.**

Keywords: **Distributed Computing, Embarrassingly Parallel, Asynchronous Transfer Mode (ATM), interactive environment.**

1. INTRODUCTION

MPI has better performance in high performance massively parallel processing computer systems to provide highly optimized and efficient implementations than parallel virtual machine (PVM).The communication tools implement one of the two communication techniques: message passing and shared memory [8].Message passing interface (MPI) and PVM are the most popular examples of the message passing tools. The software tools available for a high performance computing environment need the knowledge of internal architecture and components of the distributed system .In case of distributed systems different products are combined with different strengths, to achieve a combination which is better than any of the individual component. The communication and synchronization among the various users is the most important issue for the parallel computing environment .Therefore, over the last few years, a number of application development and representing tools are designed to solve the message passing problems in distributed environment.

A virtual parallel machine may be constructed using PVM due to its simple but complete programming interface. PVM supports heterogeneous machines, application and networks. The PVM system was gained to realize a more general and encompassing interpretation of heterogeneous computing. The message passing model appears to be gaining predominance from the prospective of number and variety of languages, and software systems for its support. Task partitioning is the major factor for any parallel computing tool amongst the distributed environment [22].PVM system has gained widespread acceptance in the high performance scientific computing community. It supports dynamic process management, while in other systems the processes are statistically defined. Dynamic process group are layered above the core PVM routines. Routines are provided for tasks to join and leave a new group. Group member are uniquely numbered from zero to the number of group member minus one.The problem solving environment may be analyzed through the parallel computing tools [20].

The PVM is a unified framework within which large parallel systems can be developed in a straightforward and efficient manner. Portability issue is more important than performance due to the two reasons: communication across the internet slow; and, the research focused on problem with scaling, fault tolerance and heterogeneity of the virtual machine [10].

This paper summarizes the important features and the limitations of the various parallel computing tools amongst the distributed computing system. The following tools performance discussed in this paper:

- **Parallel MATLAB**
- **PVM**
- **MPI**

According to the performance of the several parallel programming tools for the distributed systems PVM and MPI become the most popular due to wide range of their usage. PVM and MPI programming environments effectively harness the capability of a distributed network of UNIX workstations as well as heterogeneous network of UNIX and Window NT workstations. Portability is a measure of the ease with which software can be moved between heterogeneous computing platforms. Software which can run on diverse platforms increased longevity because it can migrate from older to newer platforms. How are these programming environments compared? How easy is it to program one versus the other? How the portability and interoperability play the important role in the parallel computing distributed systems? What are the advantages and disadvantages of each? In this paper we shall discuss each of these programming environments in some detail. Parallel Computing Toolbox and MATLAB Distributed Computing Server software solve computationally and data-intensive problems using MATLAB on multi-core and multiprocessor computers.

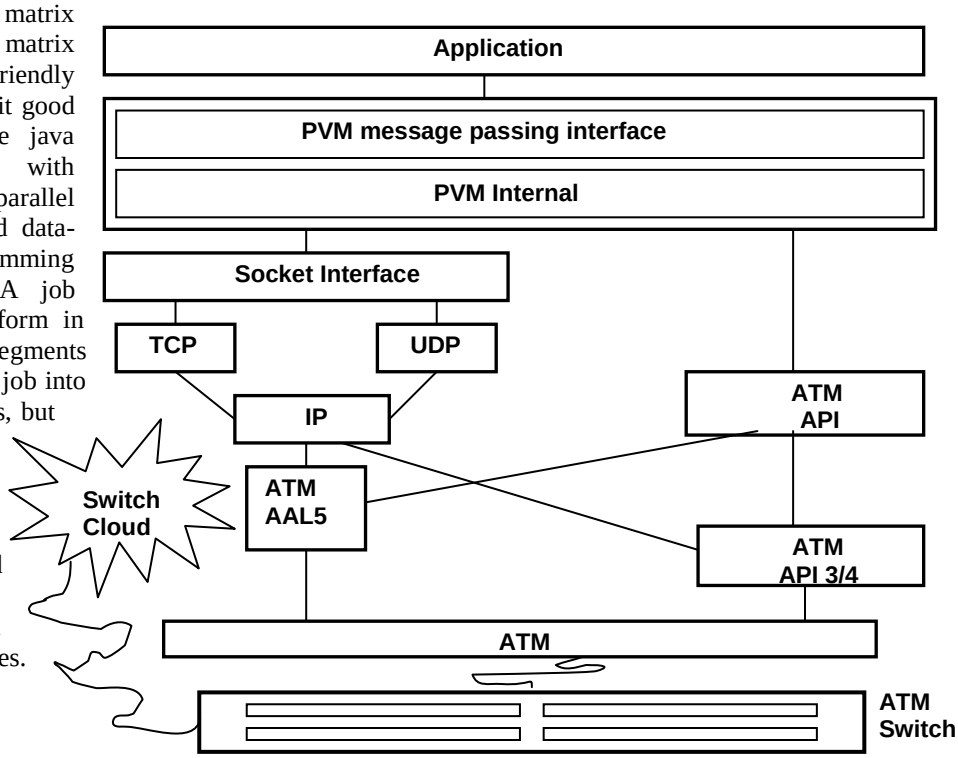
II PARALLEL MATLAB

It started in 1970s as an inactive interface to EISPACK[23],a set of eigenvalue and a linear system solution routines. In 1995 the use of parallel MATLAB was negligible in high performance parallel computing. Its interactive environment that provide high performance parallel computational routines making the drastic significant interest in the field of high performance computing .Interpreted built in language of MATLAB which is similar to C and the flexible matrix indexing are the innovative feature for matrix programming problem [9].MATLAB user friendly feature and strong graphical capabilities makes it good data analysis tool. It Provide hooks to the java programming language, making integration with compiled program easy. With the help of parallel MATLAB we can implement task-parallel and data-parallel algorithms at a high level without programming for specific hardware and network architectures. A job is some large operation that we need to perform in MATLAB session. A job is broken down into segments called tasks. We decide how best to divide your job into tasks. We can divide our job into identical tasks, but tasks do not have to be identical. The MATLAB Distributed Computing Server product performs the execution of jobs by evaluating each of its tasks and returning the result to client session [12]. In file system based communication feature, MatlabMPI [3] implement basic MPI functions in parallel MATLAB using cross-mounted directories. Parallelism may achieve through polymorphism property of parallel MATLAB [1].

III PVM

The PVM is specifically designed for heterogeneous networks of computers. The PVM has more users than any other portable parallel programming environment and is de facto standard for message passing environment [11]. The PVM provide the facility of portability and encapsulation of the functions, amongst the heterogeneous distributed computing system. It offers good facilities for parallel distributed computing but a poor scheduling facility for the distribution of the workload. A process can live to multiple groups, and groups can change dynamically at any time during a computation [18].

1) *THE EVOLUTION of PVM***:** The development of PVM started in the summer of 1989 when Vaidy Sunderam, a professor at Emory University, visited Oak Ridge National Laboratory to do research with AI Giest on heterogeneous distributed computing. The another research associate Bob Manchek joined the team and implemented a portable version PVM 2.0 of PVM in 1991.The PVM 2.0 become publicly available by the valuable efforts of Jack Dongarra. The virtual machine concept had come into the consideration due to the high speed computing on the distributed networks. On the basis of the requirement of the users, some additional interfaces are designed by the experts that allow third party debuggers and resource managers to be seamlessly



incorporated into the virtual machine [4]. The PVM facilitate the ability to transparently utilized shared memory and high speed networks like ATM to move data between clusters of shared memory multiprocessors.

The PVM system, a software framework for heterogeneous concurrent computing in network environment is available in the variety of discipline [18].The PVM supports dynamic process management while in other systems the processes are statistically defined. It is capable of harnessing the combined resources of typically heterogeneous networked computing platforms to deliver high level of performance and functionality. Performance issue dealing primarily with communication overhead at the time of message passing in the distributed high computing[3].In the dynamic PVM the task migration is possible within the system without affecting the operation of the user or application programs.

2) PVM, Process Communication, Portability and Interoperability: The advancement of network technology, improve the computing speed of the workstations rapidly day by day. A software with the high degree of portability is highly valuable because it can be use by the various users efficiently.The great diversity of computer hardware and software poses a fundamental threat of software portability. The computing enhancement through the network may be achieved by the PVM due to its portable capability [19].Amongst the various parallel distributed environments, the efficient portability facility provided by the PVM. Heterogeneity is becoming increasingly important for high performance computing. At the different architectures PVM programs can be copied, compiled and executed without the modification. A user defined collection of serial, parallel, and vector computers emulates a large distributed memory computer under the PVM. The PVM facilitates to automatically startup of the tasks on the virtual machine with message passing and synchronization. Multiple users can configure overlapping virtual machines, and each user can execute several PVM applications simultaneously. The performance of the PVM is highly remarkable in ATM [21]. Figure 2 shows protocol hierarchy of PVM in ATM network.

A set of routines provided by the different standard of PVM that enable a user to add and delete hosts from the virtual machine. These routines used to initiate and terminate the PVM tasks and the cooperation amongst them. Synchronization may be achieved by sending a Unix signal to another task, or by using barriers. To manage and create the multiple send and receive buffers, some special routines provided by the PVM for the high speed computing over the network [21]. Message buffers are allocated dynamically to save the data temporarily by the native machine parameters [14].PVM also provides the set of functions to enable applications to communicate directly with each other via TCP sockets. Dynamic PVM support the portability with effective and efficient parallel distributed computing for the PVM applications.

3) PVM, Fault Tolerance: The successful execution of the large simulation without fault detection and recovery is not possible. The failure of the any workstation in the large simulation will cause the unexpected results. Hence there are several types of applications that explicitly require a fault tolerance execution environment, due to safety or level of requirements. When the status of a virtual machine changes or a task fails, under the control of users, the PVM gives the special event message which contain the information about the particular event [17]. The notify message allows the tasks an opportunity to respond to the fault without hanging or failing. The message due to the fault tolerance is beneficial in controlling the computing resources. The fault tolerance facility of PVM boosts the performance of the message passing environment.

IV MPI

The MPI specifications can be used for writing portable parallel programs [13]. The impetus for developing MPI, according to the computing experts, is that each Massively Parallel Processors (MPP) vendor creating there own proprietary message passing-API.MPI is intended to be a standard message passing specification that each MPP vendor would implement on there system. MPI would be a library for writing application programs, not a distributed operating system. Only some specifications of the MPI provide the thread safety, which is the most important features for the heterogeneous network environment. Here a fixed set of processes is created at program initialization; one process is created at each processor. The model depicts the implementation issues on a socket for the MPI communication.

Each process knows its personal number that is called the rank of the process and each process knows rank of the other processes also. MPI would be modular, to accelerate the development of portable parallel libraries. MPI has more than one freely available, quality implementation and full asynchronous communication. The groups of MPI are solid, efficient and deterministic.

1) MPI, Evolution: The processes use a library to exchange message with another process in parallel programming. This message passing approach allows processes to run on the multiple processors to cooperate to each other to solve the large problems efficiently. A full MPI implementation

MPICH [16] is portable to all parallel machines and workstations running Chameleon , p4 or PVM [2].The MPI interface is meant to provide essential virtual topology, synchronization, and communication functionality between a set of processes in a language-independent way, with language specific syntax (binding). The applications of MPI-1 were not portable across a network of workstations because there was no standard method to start MPI tasks on separate hosts. In 1995 the MPI committee starts the effort to design the MPI-2 specifications to correct the problem of portability and to add additional communication function. After the evolution of MPI-2, the one sided communication operation and dynamic process management came into the existence across a number of tasks. The key aspect of dynamic process management feature is ability of an MPI process to participate in the creation of new MPI processor to establish communication with MPI process that has been started separately. Portability play an highly important role to reuse the software tool for the communications in the parallel computing [15].

MPI-1 and MPI-2 implementation show good results in overlapping communication and computation. MPI also specifies thread safe interfaces, which have cohesion and coupling strategies. Multithreaded point-to-point MPI code is easier than other code which supported by different implmenations.MPI-2 defines three one sided communication operations, put,get and accumulate to remote memory read remote memory, and a reduction operation on the same memory across a number of tasks.MPI-2 also defined three different methods for synchronizing the communication-global, pair wise, and remote locks. The successive implementation and documentation of the Unify explore the knowledge of PVM to generalized MPI framework.

2) MPI, Fault Tolerance: MPI provide the facility of portability and interoperability in the high performance communication systems. Message passing across the task is very convenient because each task already knows about the every other task, and all communication can be made without the explicit need for special daemon. In the MPI-1 model the host and task are considered as static in terms of fault tolerance. If a task or computing resource should fail, the entire MPI-1 application must fail.MPI-2 includes a specification for spawning new process to expend the capabilities of the original static MPI-1.

In the fault tolerance mechanism the management of faults is an open issue as discussed below:

(1) During the restart of the entire application, the periodically intermediate results should be saved by the programmer on the reliable media.

(2) The function of the MPI implementation may be augmented to return information about faults and accept communicator reconfiguration.

(3) A fully automatic fault detection and recovery facility of the MPI implementation hides the fault from the programmer and user in the distributed computing system.

V Comparison

The analysis of the different features of the tools is also given by the Table-1 through which the programmers can understand the best tool for the parallel programming.

S. N.	Parameters	MATLAB	PVM	MPI-1	MPI-2
1	Server Connection Manager	Yes	Not	Not	Not
2	Parallelism through Polymorphism	Yes	Not	Not	Not
3	Interoperability Support	Yes	Yes	Not	Not
4	Virtual Machine Support	Not	Yes	Not	Not
5	Topology Support	Yes	Not	Yes	Yes
6	Dynamic Process Group Support	Yes	Yes	Good	Good
7	Lazy Evaluation	Yes	Not	Not	Not
8	Resource Management Support	Yes	Yes	Not	Not
9	Name Spaces	Not	Yes	Not	Yes
10	Matrix Manger	Yes	Not	Not	Not
11	Macro Language Support	Yes	Not	Not	Not
12	Tracing Environment	Not	Not	Yes	Not
13	Client Connection Manager	Yes	Not	Not	Not
14	Package Manager	Yes	Not	Not	Not
15	Heterogeneity Support	Yes	Yes	Not	Not
16	Scripting Capability	Yes	Not	Not	Not
17	ID-less Communication Support	Not	Not	Not	Not
18	Backend Support	Yes	Not	Not	Not
19	Portability Support	Yes	Yes	Yes	Yes
20	Process Control Support	Yes	Fully	Not	Yes

Table-1: Comparison among different parallel computing tools

Table-1: Comparison among different parallel computing tools

VI Conclusion

The selection of the parallel programming tool through the comparison is the easiest way to select the best performing tool. In this paper the main features of the PVM, MPI and MATLAB are point out which are the most popular parallel programming tools in the distributed computing systems.

MPI is very rich by the communication functions therefore it becomes very useful for the applications that exploits the special communication modes which are absent in PVM. The absence of interoperability and fault tolerance in MPI enhances the communication performance. If an application is executing in heterogeneous platform, PVM is the most suitable option. The ability to write long running PVM applications that can continue even hosts or tasks fail or loads change dynamically due to the outside influence. It is quite important to heterogeneous distributed computing. The evaluation of the performance of the any parallel high computing program may be achieved through the parallel MATLAB in the efficient manner. The performance analysis of any program by the researchers may be performed through the parallel MATLAB due to its user-friendly capability in the field of parallel computing.

VII Future work

The brief experimental comparison of the results of parallel computing tools is not given in the literature. The researcher's intension towards the result description of each parallel programming tool is a major point of interest in the area of parallel computing. The performance comparison on the basis of the experimental results of each parallel computing tool may be used to boost the knowledge of the researcher .The parallel MATLAB user friendly feature may be enhanced

References:

1. *R. Choy,A. Edelman"Parallel MATLAB:Doing it Right",Massachusetts Institute of Technology ,Cambridge,pp-10,November 2003.*
2. *A Beguelin., V.S. Sundarm, G.A.Giest, W. Jiang, J. Dongarra , R. Manchek, "PVM 3 User's Guide's and Reference Manual". Technical Report TM-12187.Oak Ridge National Leboratory,1993.*
3. *C. J. Kepner,"Parallel Programming with MATLABMPI. Accepted by High Performance Embedded Computing", HPEC , Workshop, 2001.*
4. *J. Choi, J. J.Dongarra, R. Pozo, D. C. Sorensen and D. W. Walker, "CRPC Research into Linear Algebra Software for High Performance Computers", International Journal of Super Computing Applications, Vo1.8, No.l2. pp 99-118, Summer 1994.*
5. *A. Lucio,.Parmatlab. ftp://ftp.mathworks.com/pub/contrib/v5/tools/parmatlab/, 2001.*
6. *J. J. Dongarra, J. D. Cros, S. Hammarling, and I. Duff, "A Set of Level 3 Basic Linear Algebra Subprograms, ACM Transactions on Mathematical Software, Vol. 18, No. 1, pp. 1-17, 1990.*
7. *A. George, C. Calin, and A. P. Dvaid,"Matmarks: A Shared Memory Environment for MATLAB Programming", pp 2-4,1999.*
8. *J. Dongarra at el,"An Introduction to MPI Standard",Technical Report CS-95-274,University of Tennessee, January 1995.*
9. *C. Stephane, B. M. Francois "An Environment for High Performance MATLAB".In Languages, Compilers,and Run-Time Systems for Scalable Computers, pp 27–40, 1998.*
10. *A. Geist, A. Beguelin, J. Dongarra, W.Jiang, R. Manchek and V.Sunderam, "PVM: A Users' Guide and Tutorial for Networked Parallel Computing", MIT Press", 1994.*
11. *Geist, J. A. Kohl, and P. M.Papadopoulos,"PVM and MPI: A Comparison of Features", Calculateurs Paralleles, 8(2), 1996.*
12. *R. Choy," Parallel MATLAB survey". http://theory.lcs.mit.edu/cly/survey.html, 2001.*
13. *W.Gropp, E. Lusk,"PVM and MPI are Completely Different", Mathematics and Computer Science Division Argonne National Laboratory,24 September ,1998.*
14. *C. L. Lowson, at. el., "Basic Linear Subprograms for FORTRAN Usage", ACM Transactions on Mathematical Software, pp. 308-323, 1979.*
15. *J. Mooney,"Developing Portable Software.Information Technology: Selected Tutorials", Springer Boston, ISBN 978-1-4020-8158-3, pp.55-85, 2004.*
16. *E. D. Nathan, W.Gropp, E. Lusk and A.Skjellum,"An Initial Implementation of MPI",Technical Report MCS-P393-1193, Mathematics and Computer Science Division Argonne National Laboratory,Argonne,IL 60439,1993.*
17. *J.Pruyne, M. Liveny,"Providing Resource Management Services to Parallel Applications".In Proceedings of the Second Workshop on Environments and Tools for Parallel Scientific Computing",1994.*
18. *V.S.Sundarm, G.A. Giest, J. Dongarra, R. Manchek,"The PVM Concurrent Computing System:Evolution,Experience and Trends",Compcon Spring,1993.*
19. *L. C. Sheue, H. C. Du David, H. Jenwei, P. T. Rose and L. Mengjou,"Enhanced PVM Communications over a High-Speed LAN". IEEE Parallel and Distributed Technology 3(3),20-32,1995.*
20. *H. Salim, T. Haluk, F. Wojtek, K. Dongmin, K. Yoonhee, R. Ilkyeun, B.Xue, Y. Bouqing, V. Jon,"A Problem Solving Environment for Network Computing",IEEE Computer Society ,1997.*
21. *C. Xuebin, L. Yucheng, S. Jiachang, Z. Yunquan, and Z. Peng,"Developing High performance Blas, Lapack and Scalapack on Hitachi sr 8000 in High Performance Computing in the Asia-Pacific Region, 2000". Proceedings. The Fourth International Conference/Exhibition on, vol. 2, 14-17, pp. 993–997, May 2000.*
22. *A. Javed , Z. K. Rafiqul,"A Survey Paper on Task Partitioning Strategies in Parallel and Distributed System",Globus-An International Journal of Management & IT,Vol. 01,No. 4,November 2010.*
23. *B.T. Smith, J.M. Boyle, J.J. Dongarra, B.S. Garbow, Y. Ilebe, V.C. Kelma, and C.B. Moler.,"Matrix Eigensystem Routines-EISPACK Guide". Springer-Verlag, 2nd edition, 1976.*

