High-Performance PVM Based on Fast Message Passing*

XIA Hua-xia, ZHENG Wei-min

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

E-mail: huaxia@est4.cs.tsinghua.edu.cn; huaxiax@yahoo.com

http://www.tsinghua.edu.cn

Received June 17, 1999; accepted November 29, 1999

Abstract: PVM (Parallel Virtual Machine) software is one of the most popular software environments used on parallel workstation cluster system. However, with the rapid increase in the speed of local area networks as well as in the performance of workstation, PVM, which is based on the low-efficiency internal and the low-efficiency communication protocol, has become the bottleneck of the cluster systems. In this paper, it is explained how PVM restrains the performance of clusters, by pointing out and analyzing the limitation of PVM's mechanism. Then the details of the design and implementation of HPVM (High-performance PVM) are given, which is based on a high-speed reduced communication layer named FMP (Fast Message Passing).

Key words: parallel virtual machine; parallel workstation cluster; Myrinet; fast message passing; HPVM

With the fast development of the technique of microprocessor and storage, the performance of workstation and personal computer has been increased at a rate of 50% per year while the price is keeping decreasing. This leads to the generation of a kind of parallel computer system named Workstation Cluster or NOW (Network of Workstations). Because of its low price, high scalability, and high usability, the cluster is becoming a popular kind of high-performance computer system both in research and in business area[1~3].

Parallel software environment refers to the environment that enables users to program and run parallel applications in the cluster system. PVM (Parallel Virtual Machine) is a typical one. It is developed by Oak Ridge National Laboratory, University of Tennessee, and Emory University. It supplies a parallel programming environment, making the network communication transparent, and supporting a flexible parallel computation over heterogeneous workstation cluster. The computers with different architectures, connecting with each other and under the control of PVM, can act as an integration. In the sight of users, the cluster is just like a single computer. These characters make PVM become one of the most popular programming environments of the cluster systems.

However, as the design of PVM pays so much attention to the software's applicability, the communication performance is decreased to a great degree[4,5]. In order to achieve some convenient functions and to adapt to the different architectures of the computers in the cluster system (the cluster is called heterogeneous cluster), PVM is based on a low-efficiency communication and some complex mechanism, which increase the communication overhead greatly. Especially because of the rapid improvement of network techniques, PVM is far away from taking

* This project is supported by the National Natural Science Fundation of China under Grant No. 69873023 (国家自然科学基金). **XIA Hua-xia** was born in January, 1976. He is a graduate student in the Department of Computer Science and Technology of Tsinghua University. His research interests include parallel and distributed systems, high performance computing, fault tolerance systems and advanced networks. **ZHENG Wei-min** was born in 1945. He is a professor in the Department of Computer Science and Technology of Tsinghua University. His current research interests include computer architectures and parallel/distributed computing.

full advantages of the high-speed and the stability of the advanced network. As a result, the communication performance of the cluster is affected, which limits the efficiency of parallel computing, the adaptability to the applications, and the scalability of the cluster.

Thus, it is important to study PVM and to design a special programming environment with high-performance for homogeneous cluster. The environment which we have designed is called HPVM (High-performance PVM).

In this paper we first analyze the PVM system and find out the mechanism unfit for the homogeneous cluster. After that we make a brief introduction of FMP protocol, which HPVM is based on. Then the design of HPVM is given, followed by the performance result of one implementation of HPVM.

1 Problems of PVM

Now we will analyze the internal mechanism of PVM and find out the reason of its low efficiency. The details of PVM can be found in Ref. [3].

As shown in Fig. 1, in PVM, there is a daemon process named pvmd on each node, which serves as a maintainer and manager of the whole PVM system, spawning and distributing the tasks, adding and deleting the hosts, and so on. Three kinds of communication can be found in PVM system from the figure:

(1) Communication between pvmds: This kind of communication works using the UDP/IP network protocol. Since the UDP/IP protocol is an unreliable protocol, the acknowledge-and-retransmit mechanism is used in PVM so as to get a reliable communication.

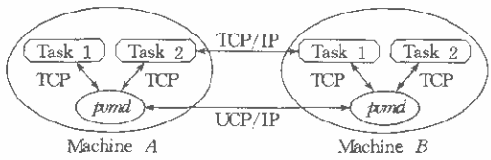


Fig. 1 Architecture of PVM system

(2) Communication between the tasks in the same host: This kind of communication works using the TCP protocol in UNIX domain, with pvmd as an agent. For instance, if task 1 wants to communicate with task 2, it needs to establish the TCP connection with pvmd first, then sends the message to pvmd. Pvmd will check the destination of the message after having received the message, and once it finds that the destination is a local task it will establish the TCP connection with the destination and sends it the message. It is obvious that the communication costs a lot although it doesn't occupy the network resource.

(3) Communication between the tasks in different hosts: This kind of communication includes two types of implementation. One is similar to that between the tasks in the same host, named normal-route. If task 2 wants to send a message to task 3 in another host, it sends the message to the local pvmd, then from the local pvmd to the foreign pvmd in the same host as task 3, and finally from the foreign pvmd to task 3. In the second implementation, task 2 will firstly get the socket address of task 3 from pvmd and send the connection request to task 3 via pvmd before it runs the direct TCP transmission with task 3. This is called direct-route. It is more efficient than the first one.

From the above analysis, we come to a conclusion that there are several factors affecting the PVM's efficiency:

(1) Neither the TCP/IP protocol nor the UDP/IP protocol is a high-efficiency communication protocol; they themselves cannot take full advantage of the high-speed network.

(2) The communication between the tasks needs the participation of pvmd, which increases the communication overhead greatly. This is obvious in the normal-route mode. Even in the direct-route mode, pvmd is needed in order to establish the TCP connection between the tasks.

(3) PVM needs its own message buffer and a complex management of the buffer. When a task wants to send a message, firstly it must allocate a buffer in umbuf format. Then it copies the user's data to the buffer and changes the data's format to form a message. In pvmd, a complex buffer management also exists corresponding to

the acknowledge-and-retransmit mechanism. Plus the buffer managemental in TCP and UDP, there are three kinds of buffer management altogether. So complex buffer management must lead to a large communication overhead.

Moreover, all the buffers are allocated dynamically, which also increases the overhead.

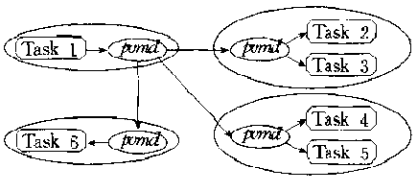


Fig. 2 The ulticast algorithm in PVM

(4) The algorithm of multicast is simple but of low-efficiency. The algorithm is based on the point-to-point communication, as shown in Fig. 2. A shortcoming of the algorithm is the heavy burden of the sender, which has been the bottleneck of the multicast. Furthermore, with the increasing of the number of destination hosts, both the load of the sender and the communication latency also have a linear increase.

2 A Brief Introduction of the Fast Message Passing Protocol

In order to build a high-performance parallel-programming environment, we use a high efficiency communication layer: the FMP (Fast Message Passing) protocol based on a high speed Myrinet network system. We will give a brief introduction of the protocol and the details can be found in Ref. [6].

2.1 Myrinet network

Myrinet is a new type of high-speed switch network. It is produced by the Myricom Company. There is one sending channel and one receiving channel on each Myrinet adapter card and the two channels can work in bandwidth of 1.28Gbps simultaneously. A 32-bit CPU is included in the Myrinet card, which enhances the communication ability greatly. Furthermore, the Myrinet channel has a high reliability with the error rate less than 10^{-15}. All these characteristics are in favor of the implementation of a high-efficiency communication protocol[7].

2.2 The principle of FMP

The Fast Message Passing includes two parts: the inter-host communication and the intro host communication.

• Inter-host communication

It is based on the Myrinet system. FMP achieves very good communication performance by using several techniques such as memory mapping, pipeline data transmission, credit flow control, cache, multithreading, DMA (Direct Memory Access) for long message and PIO (Programming I/O) for short message.

• Intro-host communication

It is based on the mechanism of shared memory in UNIX. The header and the body of a message are stored in a header list and a memory heap respectively, which makes an efficient use of the memory.

In order to speed up the processing, all the memory mentioned above is allocated statically when the system is being initialized.

3 The Design of High-performance PVM

Now let us take a look at the techniques used in HPVM to improve the performance.

3.1 Build HPVM based on FMP instead of TCP & UDP

All the communications in HPVM are over the FMP protocol, which gives HPVM a simple but effective internal environment. At the same time, HPVM supplies the same application interface as PVM so that the users can transplant all the applications in PVM to HPVM without modifying the source codes. The two kinds of architecture are shown in Fig. 3.

3.2 Reduce the internal communication mechanism of PVM

We have analyzed that the co nunication in PVM has three different kinds and pvmd often serves as an

agent. As shown in Fig. 4, the mechanism is so complex that the communication is of low-efficiency and the source codes are organized badly. While in HPVM the address table is stored in the shared memory, which can be queried by all the tasks so that the tasks can communicate directly without the medium of pvmd. Moreover, since a reliable communication service is supplied by FMP, the mechanism of acknowledge-and-retransmit which is used in PVM is not necessary in HPVM. A reduced PVM internal communication mechanism is shown in Fig. 5.

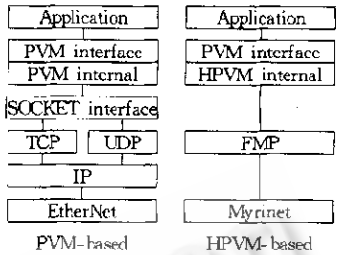


Fig. 3 The architectures of the two kinds of parallel systems

3.3 The translation between TID and communication address

To each task in the parallel programming environment, an integer ID is allocated as the task's identity (TID), through which the tasks can be identified and can communicate between each other. However, the low-level communication needs the communication address. Thus the translation must be done between the TID and the communication address.

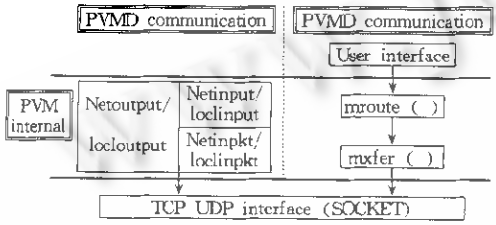


Fig. 4 Communication levels in PVM

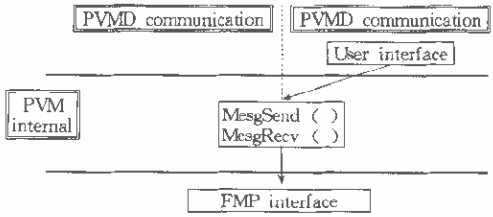


Fig. 5 Communication levels in HPVM

In PVM, a mapping table of TID and SOCKET address is maintained by pvmd. When a task sends or receives a message, it must talk to pvmd to get the translated address.

In HPVM, an address-mapping table named "PortTable" is stored in the shared memory of each host. The table can be modified only by pvmd in order to keep the concurrency in the hosts. And it can be read by all the tasks in the same host, which enables the tasks to query the table directly without the help from pvmd. In order to quicken the query, the table consists of two Hash tables used for the translation of "TID⇒FMP address" and "FMP address⇒TID" respectively.

3.4 Simplify the message structure

Since the communication mechanism has been reduced in HPVM, the complex message structure is redundant, especially the part involving the acknowledge-and-retransmit mechanism.

So we simplify the message structure to a two-level structure: message and fragment. A message consists of zero, one or more fragments. The time of the data replication in a send and receive process goes down to four (in intro-host communication) or five (in inter-host communication) which is at least 7 times more in PVM.

3.5 Optimize the buffer management

In the implementation of FMP, there are one buffer for local message and one buffer for foreign message, both of which are allocated statically to quicken the communication. However, a disadvantage of static buffer is that the buffer may be used up and deadlock is caused. This often occurs when some types of messages sent out are not received by the destination immediately, as shown in Fig. 6. The "mesg c" in the figure may be a message for A in unexpected message type or for B or for other process.

So the dynamic memory is used to avoid the shortage of buffer. Each task in HPVM has a dynamic buffer. Once a receive or a send operation fails, which is usually because of the full common buffer, the receiver of the sender will receive all its messages and put them in the private buffer before retrying the operation.

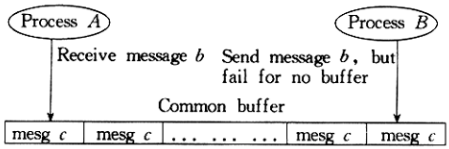


Fig. 6 Deadlock for no enough static buffer

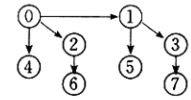


Fig. 7 Binomial-tree multicast algorithm

The use of the static common buffer and the dynamic private buffer combines the advantages in communication speed and buffer size.

3.6 About the translation of data format

Since PVM is designed for heterogeneous cluster, the format of data is translated before the data are packed into the message. However, HPVM is designed for homogeneous cluster, so the translation is omitted.

3.7 Improve the multicast algorithm

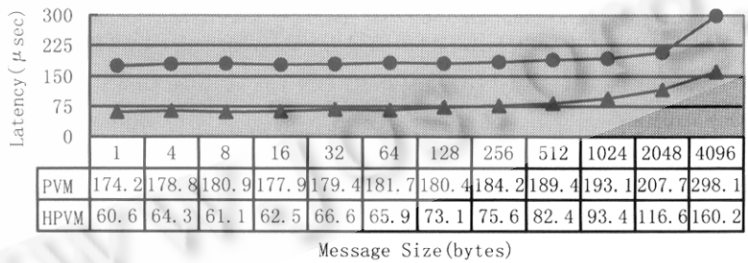
Besides the algorithm based on host-to-host communication, HPVM has an alternative algorithm based on a binomial tree, as shown in Fig. 7. The algorithm achieves better performance if there are many hosts in the cluster or if the message size is very large.

4 Performance Results

This section shows the performance results of HPVM over FMP and PVM over TCP/IP. Both the parallel systems are based on Sun UltraSparc 2 running SunOS 5.5.1. The machines are connected to each other through a 1.2Gbps Myrinet switch.

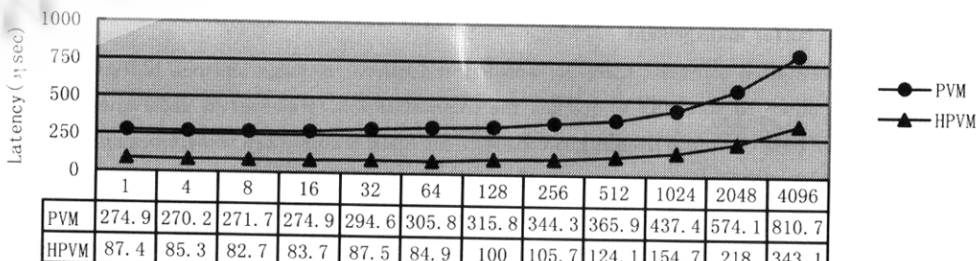
Figures 8 and 9 plot latency values in intro-host communication and inter-host communication respectively. For the representative small size of the message, the HPVM latency is only about one-third of the PVM latency.

Figures 10 and 11 compare the bandwidth values between HPVM and PVM. We analyze the representative messages with size more than 8K. As shown in Fig. 10, the local bandwidth of HPVM is about 60% greater than that of PVM. And as shown in Fig. 11, the remote bandwidth of HPVM is nearly three times of that of PVM.



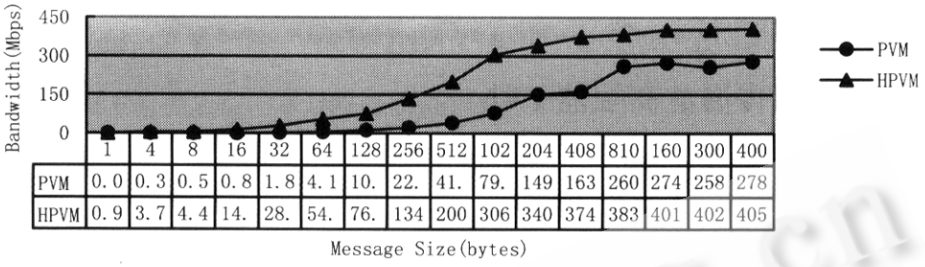
	1	4	8	16	32	64	128	256	512	1024	2048	4096
PVM	174.2	178.8	180.9	177.9	179.4	181.7	180.4	184.2	189.4	193.1	207.7	298.1
HPVM	60.6	64.3	61.1	62.5	66.6	65.9	73.1	75.6	82.4	93.4	116.6	160.2

Fig. 8 Local communication latency



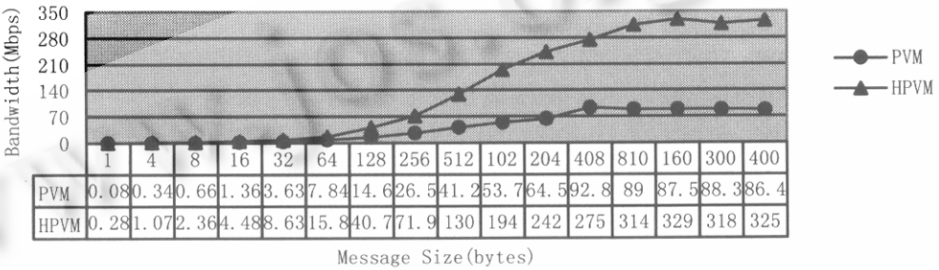
	1	4	8	16	32	64	128	256	512	1024	2048	4096
PVM	274.9	270.2	271.7	274.9	294.6	305.8	315.8	344.3	365.9	437.4	574.1	810.7
HPVM	87.4	85.3	82.7	83.7	87.5	84.9	100	105.7	124.1	154.7	218	343.1

Fig. 9 Remote communication latency



	1	4	8	16	32	64	128	256	512	102	204	408	810	160	300	400
PVM	0.0	0.3	0.5	0.8	1.8	4.1	10.	22.	41.	79.	149	163	260	274	258	278
HPVM	0.9	3.7	4.4	14.	28.	54.	76.	134	200	306	340	374	383	401	402	405

Fig. 10 Local communication bandwidth



	1	4	8	16	32	64	128	256	512	102	204	408	810	160	300	400
PVM	0.08	0.34	0.66	1.36	3.63	7.84	14.6	26.5	41.2	53.7	64.5	92.8	89	87.5	88.3	86.4
HPVM	0.28	1.07	2.36	4.48	8.63	15.8	40.7	71.9	130	194	242	275	314	329	318	325

Fig. 11 Remote communication bandwidth

5 Future Work

• We may redefine the format of TID so that the FMP address can be found out directly from the TID instead of looking up in a table.

• With support from an improved FMP, the data to be sent can be packed directly to the common buffer rather than being packed in the temporary buffer before being copied to the common buffer.

References:

[1] Gordon, Bell. 1995 observations on supercomputing alternatives: did the MPP bandwagon lead to Cul-de-Sac. Communications of the ACM, 1996,39(3):11~15.

[2] Shen Jun, Zheng Wei-min. Network performance assessment and improvement for workstation clusters. In: Xu De ed. Second Sino-German Workshop on Advanced Parallel Processings Technologies(APPT'97). Koblenz, Germany: Verlag Dietmar Folbach und den Autoren, 1997. 33~40.

[3] Geist, A., Beguelin, A., Dongarra, J. *et al.* PVM: Parallel Virtual Machine-A Users' Guide and Tutorial for Network Parallel Computing. The MIT Press, 1994.

[4] Shen Jun, Zheng Wei-min. Research on improving the communication performance of workstation cluster system. Mini-Micro Systems, 1999,18(6):8~13.

[5] Shen Jun, Zheng Wei-min. Modeling parallel computing performance for Heterogeneous workstation clusters. Computer Research and Development, 1998,35(3):193~198 (in Chinese).

[6] Shen Jun, Zheng Wei-min, Wang Ding-xing, *et al.* An implementation of fast message passing communication mechanism based on Myrinet. Journal of Software, 1998,9:33~38 (in Chinese).

[7] Boden, N. J., Danny, C., Felderman, R. E., *et al.* Myrinet: a gigabit-per-second local area network. IEEE Micro, 1995, 15(1):29~36.

附中文参考文献:
[5] 申俊,郑纬民.异构并行工作站机群系统的性能评价指标.计算机研究与发展,1998;35(3):193～198.
[6] 申俊,郑纬民,王鼎兴,等.一种基于 Myrinet 的快速消息传递机制实现技术.软件学报,1998,9:33～38.

基于快速消息传递的高性能 PVM

夏华夏, 郑纬民

(清华大学 计算机科学与技术系,北京 100080)

摘要: 并行虚拟机(parallel virtual machine,简称 PVM)是并行工作站机群系统中流行的并行软件环境之一.分析了 PVM 的实现机制,指出 PVM 低效的原因,并给了基于高速精简通信层调整消息传递(fast message passing,简称 FMP)的高性能 PVM(high-performance PVM,简称 HPVM)的详细设计和实现.

关键词: 并行虚拟机;并行机群;Myrinet;快速消息传递;HPVM(high-performance PVM)
中图法分类号: TP393 **文献标识码**: A