Windows 98 系統下之分散式運算

Hsin-Chien Huang
Department of Physics, Chinese Culture University
Taipei, Taiwan, ROC

黃信健
中國文化大學物理系
Tel: (02)2861-0511 ext.372
email: hchuang@faculty.pccu.edu.tw

Abstract

Keywords：parallel virtual machine, network, pc farm, SPMD, MPMD, benchmark study, distributed computing

The development of PVM, the Parallel Virtual Machine, has received more and more emphases from scientific and engineering sectors. The power of a PVM is that, as a single virtual machine, it can consist of computers with different architectures including workstations and pc¡¦s with Intel or alpha chips, and running under Unix or Linux operating systems, even under windows NT and 98. Currently, most institutions participating in the PVM projects favor the Unix-like systems, as PVM is most readily implemented in these systems. On the other hand, running PVM on a local windows network maybe more attractive since the windows system is more popular. In this paper, we would like to discuss the process of building a local windows NT/98 pc farm using currently available facility, and to observe the performance when SPMD and MPMD programs are executed in this system. At this stage, 4 pc are enrolled in our PVM, and considerable saving on CPU time has been achieved. We will add more pc¡¦s in the next step to increase the reliability of the benchmark analysis and enhance the ability of our PVM to do high performance scientific computing.

關鍵詞：平行虛擬主機，網路，個人電腦陣列，SPMD，MPMD，電腦時間標記研究，分散式計算

平行虛擬主機(PVM)的發展，越來越受到科學及工程部門的重視。作為一個單一虛擬主機，PVM 的威力在於它可由採用英代爾或 alpha 晶片等不同架構的工作站及個人電腦組成；而且可在 Unix 或 Linux 甚或 windows NT 或 98 系統下執行。目前，大部分投入 PVM 計畫的機構都採用 Unix 家族系統，因其最易執行 PVM。然而在 windows 區域網路上執行 PVM 的計畫可能更具吸引力，因為 windows 系統更流行。本文討論如何在目前現有的電腦設施上建立一座區域的 windows 個

人電腦陣列，繼而觀察此系統在執行 SPMD 及 MPMD 架構之效能。目前，連線之 PC 數為四，針對我們所採用的程式，已可節省可觀之 CPU 時間。未來我們將逐步擴充 PC 數，不僅提升效能分析之可信度，亦可增強我們的 PVM 從事高效能科學運算之能力。

Introduction

PVM is an integrated set of software tools and libraries for distributed computing that combines a collection of heterogeneous computers, which are connected over a network into a parallel virtual machine[1, 2, 3]. The overall objective of the PVM system is to enable such a collection of computers to be used cooperatively for concurrent or parallel computation. This machine can consist of computers with different architectures such as Intel or alpha chips, run under different operating systems such as Unix, Linux and even windows NT/98/95 operating systems, and still be able to act as if it were a single parallel machine. As the software is in public domain, many organizations that already have a cluster of workstations and pc¡¦s can get a parallel machine for free and solve larger problems using existing hardware resources. At present, most institutions participating in the PVM projects favor the Unix-like systems, as PVM is most readily implemented in these systems. On the other hand, running PVM on a local windows network maybe more attractive since the windows system is more popular.

The major complexity in building a Windows 98 PVM lies on the fact that, unlike the UNIX system, a remote shell daemon / service is not intended to be supported by Windows. This service enables access of the other remote computers in such a way that processes can be placed on remote hosts. Thus, as a first step to build a Windows PVM, a suitable rsh client, preferably BSD compliant needs to be installed.

The procedures

Our PVM consists of 4 pc¡¦s connected over a TCP/IP network. It includes one PIII 500, one PIII 233 and 2 PII 200. The setup is shown in Figure 1

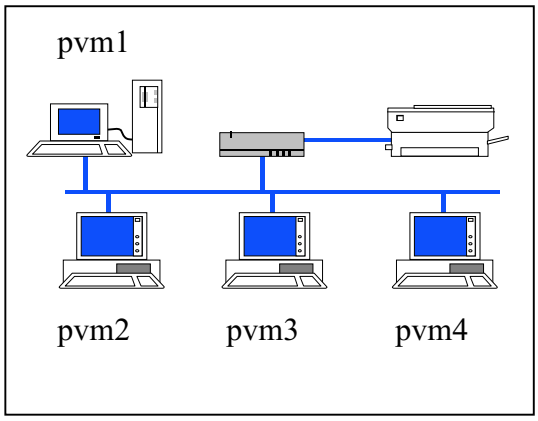

Fig.1 The PVM setup

The major procedures to build and test our system are as follow,

1. Installing a remote shell daemon
   Commercial rsh client programs are available. These include the Arshd package from Ataman[4] and Wrshd from PC consult[5]. The latter proves to be working smoothly in our system. However, an equivalence list in the install directory must be properly edited to grant access to the other machines that are to be enrolled.
2. Installing the PVM software
   The PVM software comes with an InstallShield. Care must be taken, however,

to set up the environment variables correctly. These variables indicate the root directory, the machine architecture, and locations of C and Fortran compilers as well as rsh client program.

3. Running SPMD and MPMD programs that call the PVM message passing library routines

   Both Single Program Multiple Data (SPMD) and Multiple Program Multiple Data (MPMD) modes can be implemented in our system. The former requires that each node run an identical (single) program, though the data being processed on each node may differ. The latter allows the user to run different programs on each of the nodes[6, 7]. For the MPMD mode, we adopt the familiar manager-worker, or master and slave model, and take a specific program that solves the bifurcation map[8]. In the master and slave model, the master process is the only one started by the user on one of the machines. It then starts and controls processes on the other machines that perform the actual work. In our system, the master program can reside in any of the 4 hosts, but only one slave program should reside in each host. This implies that the master and a slave are running concurrently in one of the host, and is seen to have little impact on the performance of the system since the master is lazy in this model. On the other hand, the configuration has the advantage of increasing the number of slave by one, which is important when the total number of machines enrolled in the PVM is limited. For the SPMD mode, we take a program written by David W. Walker[9, 10] which solves the classic 2D Laplace equation in a rectangular box using Jacobi method. More detailed discussions on the subjects can be found on the NIST, NACPHY and ORNL websites.

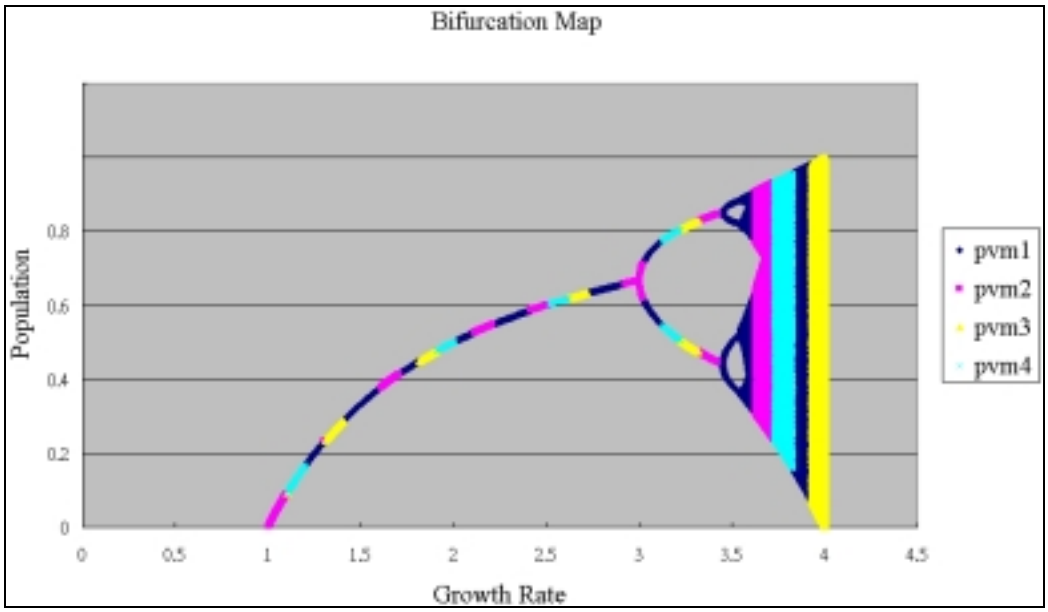

Fig.2 The bifurcation map

Results

1. MPMD - The bifurcation map

The following table shows the size of the data files generated in our hosts

PVM1 (PIII500)	PVM2 (PIII233)	PVM3 (PII 200)	PVM4 (PII 200)
4104K	2210K	1579K	1579K

Table 1. The size of data files

It is clear that the file size is larger whenever the machine is faster. This is a direct consequence of our load balancing programming strategy.

The data can be collected to make the well-known bifurcation map, as shown in figure 2. It is similar to the result given in the NACSE webpage[8]. However, the cited result was obtained using a Unix PVM.

2. SPMD - 2D Laplace equation

The following table shows the saving of CPU time in percentage as the number of nodes increases

Number of nodes	2 nodes	3 nodes	4 nodes
CPU time (%)	100	99.23	59.91

Table 2 The CPU time against number of nodes

In SPMD model, a single program takes care of all the computation and communication tasks, thus, it is necessary to have at least 2 nodes to get the program running. It can be seen from the table that considerable saving of CPU time is achieved when all 4 nodes are added to the system.

Conclusion

The above results from the test run indicate encouraging sign of success of our primitive 4 nodes system. Our next steps will be to build a larger PVM consisting of at least 15 machines and to develop 2D and 3D particle and MHD simulation codes that can be run efficiently in our PVM.

Acknowledgement

The author wishes to deeply appreciate the staffs and students in the physics department for the time and effort they contribute to the project.

References

1. A. Beguelin, J. Dongarra, G. A. Geist, W. Jiang, R. Manchek, V. Sunderam, PVM: Parallel Virtual Machine, A User's Guide and Tutorial for Networked Parallel Computing, MIT Press, Cambridge, MA, 1994.
2. G. A. Geist II, J. A. Kohl, R. Manchek, P. M. Papadopoulos, ``New Features of PVM 3.4", 1995 EuroPVM User's Group Meeting, Lyon, France, September 1995.
3. http://www.epm.ornl.gov/pvm/
4. http://www.ataman.com/
5. http://www.winrshd.com/
6. Evaluating the Performance Limitations of MPMD Communication , Chi-Chao Chang, Grzegorz Czajkowski, Thorsten von Eicken, SC97 : High Performance Networking and Computing Conference, Nov. 1997
7. http://www.nist.gov/
8. http://nacphy.physics.orst.edu/
9. http://www.csc.fi/
10. http://www.epm.ornl.gov/~walker/OLD_OR

NL_WEB_PAGE/mpi/examples/jacobi-pvm
.html