PVM on Windows and NT Clusters

Stephen L. Scott[1,+], Markus Fischer[2], and Al Geist[1]

[1] Oak Ridge National Laboratory, Computer Science and Mathematics Division, P.O. Box 2008, Bldg. 6012, MS-6367, Oak Ridge, TN 37831.
scottsl1@ornl.gov, geist@msr.epm.ornl.gov

[2] Paderborn Center for Parallel Computing, University of Paderborn, 33100 Paderborn, Germany. getin@uni-paderborn.de

Abstract. This paper is a set of working notes[1] based on recent experience using PVM on NT clusters and Windows machines. Included in this document are some techniques and tips on setting up your own cluster as well as some of the anomalies encountered during this work.

1 Introduction

Cluster computing over a network of UNIX workstations has been the subject of research efforts for a number of years. However, this familiar environment of expensive workstations running UNIX has begun to change. Interest has started to focus on off-the-shelf Intel based Pentium class computers running Microsoft's NT Workstation and NT Server operating systems. The NT operating system is a departure in both function and philosophy from UNIX. Regardless of the differences, this interest is being driven by a combination of factors including: the inverse relationship between price and performance of the Intel based machines; the proliferation of NT in industry and academia; and the new network technologies such as Myrinet, Easynet, and SCI. However, lacking in this equation is much of the effective cluster computing software developed in the past for the UNIX environment - that is with the notable exception of PVM. PVM[1] has been available from ORNL for the Windows and NT world for approximately 2-years. However, the transition from UNIX to the W/NT[2] world has not been without problems. The nature of the Windows operating system makes PVM installation and operation simple yet not secure. It is this lack of security

[+] This research was supported in part by an appointment to the Oak Ridge National Laboratory Postdoctoral Research Associates Program administered jointly by the Oak Ridge National Laboratory and the Oak Ridge Institute for Science and Education.

[1] As "working notes" implies - updated information may be found at http://www.epm.ornl.gov/~sscott

[2] W/NT is used to represent both Windows and NT operating systems. All comments are for both Windows and NT unless otherwise specified.

that provides both of these *benefits* by making all Windows users the equivalent of UNIX root. The NT operating system, on the other hand, comes with a plethora of configuration and security options. When set properly, these options make NT far more secure than Windows. However, when used improperly they can render a system insecure and unusable.

First is a look into the ORNL computing environment used to generate the information of this report. This provides an example of tested hardware and software for constructing an NT cluster. This is followed by a variety of cluster configuration options that have been tried along with comments regarding each.

2 The Oak Ridge Cluster Environment

The Oak Ridge NT cluster is a part of the Tennessee / Oak Ridge Cluster (TORC) research project. This effort is designed to look into Intel architecture based common off the shelf hardware and software. Each cluster varies in both specific hardware configuration and in the number of machines running NT and Linux operating systems. UTK/ICL administers the University of Tennessee portion of the cluster and the Distributed Computing group at ORNL administers their cluster.

2.1 Hardware and Operating System Environment

The Oak Ridge portion of this effort consists of dual-Pentium 266MHZ machines using Myrinet, gigabet Ethernet, and fast Ethernet network hardware. For testing purposes, three machines are always running the NT 4.0 operating system. The other machines in the cluster are generally running Red Hat Linux. However, if additional NT machines are desired, Linux machines may be rebooted to serve as NT cluster nodes.

Of the three machines always running NT, Jake is configured as NT 4.0 Server and performs as NT Domain server for both the cluster and any remote Domain logins. The other two machines, Buzz and Woody, are configured with NT 4.0 Workstation. Any machine on the network or internet for that matter may access the cluster. However, for this work it was generally accessed via the first author's desktop NT machine (U6FWS) running 4.0 Workstation. Also used in this work was a notebook Pentium providing the Windows 95 component.

Further information and related links regarding the TORC cluster may be found at the ORNL PVM web page.

2.2 Supporting Software

In addition to PVM 3.4 beta-6 for Intel machines, there are two software packages that were used extensively during this work. First is Ataman RSHD software that is used to provide remote process execution. This is a reasonably priced shareware package available at http://www.ataman.com. A RSHD package is required for the use of PVM on W/NT systems. Second is the freeware VNC (Virtual Network Computing) available from ORL at http://www.orl.co.uk. Although, not required for PVM's operation, VNC provides a simple and free way to perform remote administration tasks on W/NT systems.

Ataman RSHD

There are three versions of this software - one for NT on Intel systems (version 2.4), a second for NT on Alpha systems (version 2.4 - untested here), and a third for Windows 95 systems (version 1.1). At this writing it is unknown if the Windows 95 version will work for 98 or for that matter if it is even needed for Windows 98. All indications are that the NT version will operate on NT 5.0 when released. This section will become a moot point should Microsoft decide to field RSHD software. However, all indications are that they are not interested in doing so.

Although the Ataman RSHD software is a straightforward installation, it MUST be installed and configured on each machine. This is not difficult but is time consuming. One way to simplify the configuration of multiple machines with the same user and host set is to do one installation and propagate that information to the other machines. For a setup with many users or many machines this procedure will save some time. However, not much is gained in the case of few users or few machines. Furthermore, this process can only be done on machines with the same operating system. For example - NT 4.0 on Intel to NT 4.0 on Intel.

After successfully installing on one machine (the donor machine) perform the following steps while logged in as the NT Administrator or from an account with Administrator privileges:

1. From the donor machine, copy the entire directory that contains the Ataman software suite to the same location on the target machine.
2. On the donor machine, run the register editor (regedit) and export the entire Ataman registry branch to a file. This branch is located at {HKEY_LOCAL_MACHINE\SOFTWARE\Ataman Software, Inc.}
3. Move the exported file to a temporary location on the target machine or if you have shared file access across machines it may be used directly from the donor machine.
4. On the target machine, perform installation per instructions in Ataman manual. Do not setup user information, as it will be imported in next step.
5. On the target machine, run the registry editor - go to the {HKEY_LOCAL_MACHINE\SOFTWARE} level in the registry and perform an import registry file using the donor's file.
6. On the target machine, invoke Ataman icon from the windows control panel folder and reenter all user passwords. Granted, reentering all passwords is a lengthy process, but not as lengthy as reentering user information for every user.

VNC - Virtual Network Computer

Although, the Virtual Network Computer software is not necessary for the operation of a PVM cluster, it greatly simplifies the administration of a group of remote W/NT machines. This software package provides the ability to remotely operate a W/NT machine and control it as if you were sitting in front of the local keyboard and monitor. While there are some commercial packages that provide the same services as VNC, none tested performed any better than this freeware package.

There are a number of versions of VNC available including W/NT, Unix, and Macintosh. There are also two sides to the VNC suite. One is the VNCviewer and the other is the VNCserver. VNCviewer is the client side software that runs on the local machine that wants to remotely operate a W/NT machine. VNCserver must be running on a machine before VNCviewer can attach for a session. It is recommended that all remote machines have VNCserver installed as a service so that it will be automatically restarted when the W/NT reboots. When installed as a service, there will be one VNC password that protects the machine from unauthorized access. User passwords are still required if no one is logged in at the time a remote connection is established. CAUTION: a remote connection attaches to a machine in whatever state it is presently in. This can present a large security problem if someone has the VNC machine password and connects to a machine that another person has left active. However, restricting VNC access to only administrator access users should not present a problem since it is a package essentially designed for remote administration.

One other warning regarding VNC: The VNChooks (see VNC documentation) were activated on one Windows 95 machine. Error messages were generated during the installation process. Although the software was uninstalled, there are still some lingering problems on that machine that did not exist prior to the hook installation. While it is not known for certain that the VNChooks caused problems, it is recommended that this option be avoided until more information is known.

3 W/NT Cluster Configuration

There are a number of factors to consider when implementing a cluster of computers. Some of these factors are thrust upon the cluster builder by virtue of the way W/NT machines tend to be deployed. Unfortunately it is not always the case that there is a dedicated W/NT cluster sitting in the machine room. Unlike in the UNIX environment, PVM's installation and use is directly affected by W/NT administration policy. Users in the UNIX world are easily insulated from one another. W/NT unfortunately does not provide this insulation. Thus, when setting up a W/NT computing cluster one must consider a number of factors that a UNIX user may take for granted.

The three basic configuration models for PVM W/NT clusters are the *local*, *server*, and *hybrid* models. Adding to the complexity of these three models are the three cluster computing models that one must consider. These are the *cooperative cluster*, the *centralized cluster*, and the *hybrid cluster*. At first glance, it appears that there is a one-to-one mapping of PVM model to cluster model. However, the decision is not that simple.

3.1 Cluster Models

The first cluster model is that of the *cooperative* or adhoc cluster. The cooperative environment is where a number of users, generally the machine owner, agree to share their desktop resources as part of a virtual cluster. Generally, in this environment, each owner will dictate the administrative policy for those resources they are willing to share. The second cluster model is that of the *centralized* cluster. Generally a centralized cluster is used so that the physical proximity of one machine to another can be leveraged for administrative and networking purposes. The centralized cluster is usually a shared general computing resource and frequently individual machines do not have monitors or other external peripherals. The third cluster model is the *hybrid* cluster. The hybrid cluster is generally what most researchers will use. This cluster environment is a combination of a centralized cluster with the addition of some external machines as the cooperative cluster component. Many times the cooperating machines are called into the cluster as they have special features that are required or advantageous for a specific application. Examples would include special display hardware, large disk farms, or perhaps a machine with the only license for a visualization application.

The ORNL cluster consists of a centralized cluster and the addition of remote machines makes the tested configuration a hybrid cluster.

3.2 PVM Models

First is the *local* model where each machine has a copy of PVM on a local disk. This method has the benefit of being conceptually the most direct and producing the quickest PVM load time. The downside is that each machine's code must be individually administered. While not difficult or time consuming for a few workstations the administration quickly becomes costly, cumbersome, and error prone as the number of machines increases. Second is the *server* model where each local cluster of machines contains a single instance of PVM for the entire cluster. This method exhibits the client-server benefit of a centralized software repository providing a single point of software contact. On the negative side, that central repository represents a single point of failure as well as a potential upload bottleneck. Even with these potential negatives, the centralized server approach is generally the most beneficial administration technique for the cluster environment. Third is the *hybrid* model that is a mixture of the local and server models. An elaborate hybrid configuration will be very time consuming to administer. PVM and user application codes will have to be copied and maintained throughout the configuration. The only significantly advantageous hybrid configuration is to maintain a local desktop copy of PVM and application codes so that work may continue when the cluster becomes unavailable.

3.3 Configuration Management

This is where the W/NT operating system causes the operation of PVM to diverge from that of the UNIX environment. These difficulties come from the multi-user and remote access limitations of the W/NT operating system and not PVM.

One such difference is that the W/NT operating system expects all software to be administrator installed and controlled. Since there is only one registry in the W/NT system, it is maintained by the administrator and accessed by all. Thus, registry values for PVM are the registry values for all users of PVM. Essentially, this means that there is no such thing as an individual application. While it is possible to have separate individual executables, and to restrict access to an application through file access privileges, it is not possible to install all of these variants without a great deal of confusion and overhead. Thus, for all practical purposes, W/NT permits only one administrator installed version of PVM to be available. This is a direct departure from the original PVM philosophy that individual users may build, install, and access PVM without any special privileges. Furthermore, each PVM user under UNIX had the guarantee of complete autonomy from all other users including the system itself. This meant that they could maintain their own version of PVM within their own file space without conflicting with others or having system restrictions being forced on them. It is important to note that PVM on W/NT, as on UNIX, does not require privileged access for operation. However, it is very important to remember that a remote user of a Windows machine has complete access to all machine resources as if they were sitting directly in front of that machine.

Another problem is that local and remote users of W/NT share the same drive map. This means that all users will immediately see and may be affected by the mapping of a drive by another user. This also limits the number of disks, shared disks, and links to less than 26 since drives on W/NT machines are represented as a single uppercase character. This is a major departure from the UNIX world where drives may be privately mounted and links created without affecting or even notifying other users. It also goes directly against the original PVM philosophy of not having the potential to affect other users.

4 Anomalies

PVM in the Windows and NT environment is somewhat temperamental. At times it appears that the solution that worked yesterday no longer works today. Here are some of the documented deviations of PVM behavior on Windows and NT systems versus its UNIX counterpart.

4.1 Single Machine Virtual Machine

Because PVM embodies the virtual machine concept, many people develop codes on a single machine and then move the application to a network of machines for increased performance. When doing so, beware of the following failure when invoking PVM from the start menu on a stand-alone machine. The PVM MS-DOS window will freeze blank and the following information is written to the PVM log file in the temporary directory.

```
[pvmd pid-184093] readhostfile() iflist failed
[pvmd pid-184093] master_config() scotch.epm.ornl.gov: can't gethostbyname
[pvmd pid-184093] pvmbailout(0)
```

This error occurs when the network card is not present in the machine. The first encounter of this error was on a Toshiba Tecra notebook computer running Windows-95 with the pcmcia ethernet card removed. The error was fixed by simply replacing the pcmcia ethernet card and rebooting. The card need only be inserted into the pcmcia slot and does not require connection to a network. So, when developing codes on the road, remember the network card.

4.2 NT User Domains

The Domain user is a feature new to the NT operating system that does not exist in the Windows world. Windows only has the associated concept of work groups. Using NT Domains intermixed with machines using work groups has great potential for creating conflicts and confusion.

While it is possible to have the same user name within multiple domains as well as various work groups on NT systems it is not recommended that you do so. This is guaranteed to cause grief when using the current version of PVM. The multiple domain problem is in both the Ataman software as well as PVM. However, the only symptoms observed throughout testing presented themselves as PVM startup errors. Ataman user documentation warns against using user accounts with the same name even if they are in different domains.

The symptoms are exhibited from the machine where PVM refuses to start. Generally, there will be a pvml.userX file in the temporary directory from a prior PVM session. Under normal circumstances this file is overwritten by the next PVM session for userX. However, if (userX / domainY) created the file, then only (userX / domainY) can delete or overwrite the file as it is the file owner. Thus all other userX are prevented from starting PVM on that machine since they are unable to overwrite the log file.

This problem was encountered most frequently when alternating where PVM is initially started. For example when experimenting with NT Domain Users on Jake, Woody, and Buzz while U6FWS was running as a local user. Experience to date has shown that there are fewer problems when workgroups are used instead of domains.

Unfortunately, this means that a PVM user will have to have a user account on every machine to be included in the virtual machine. Perhaps with more NT experience we can resolve this issue. Administrator access is required to solve this lockout problem, as the pvml.userX file must be deleted.

Related to this is the use of NT Domain based machines mixed with Windows machines. This presents a problem since Windows 95 does not support user domains. The difficulty occurs when a Windows machine attempts to add an NT machine with user domains. PVM is unable to add the NT to the virtual machine. However, an NT with or without user domains is able to successfully add and use a Windows machine. This access is permitted, as Windows does not validate the user within a user domain.

5 Conclusion and Future Work

This paper has provided some insights regarding the construction, installation, and administration of a cluster of W/NT machines running PVM. Obviously there is much more information that could be included here. However, due to time and space constraints it is impossible to do so.

First we need more time to explore all the intricacies of the W/NT operating systems. Of course this is a moving target as Windows 98 has already been released and NT 5.0 has been promised for some time. Furthermore, we are unsure that all problems can be resolved so that PVM behaves exactly on W/NT as it does on Unix.

The space problem is easily resolved today via the WWW. Look to the web links provided throughout this paper for more current and up to date information.

References

1. Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R, Sunderam, V.: PVM: Parallel Virtual Machine - A Users' Guide and Tutorial for Networked Parallel Computing, MIT Press, Boston, 1994.