The Parallel Applications Development Environment (PADE) Installation Guide

Judith E. Devaney
Robert Lipman
Minwen Lo
William F. Mitchell
Computing and Applied Mathematics Laboratory

Mark Edwards
Charles W. Clark
Physics Laboratory

National Institute of Standards and Technology

PADE – Major Release 1.4
November 21, 1995

Abstract

This manual describes the installation of the Parallel Applications Development Environment (PADE) developed at the National Institute of Standards and Technology (NIST). PADE is a flexible, customizable environment for developing parallel applications that use the Parallel Virtual Machine (PVM) message-passing library. It provides an integrated framework for all phases of development of a message-passing parallel application: editing, compilation, execution, and performance monitoring and enhancement. The PADE package consists of an intuitive graphical user interface, a suite of PVM utilities, and extensive documentation in PostScript, ASCII, and HTML formats. PADE is most useful with a heterogeneous virtual machine consisting of multiple architectures and/or multiple file systems where it aids the parallel developer by helping to keep track of which source files must be sent and compiled on which hosts, sending the files, issuing the compilation commands, and displaying the results of the compilation on the developer's terminal.

Some eminent scientists sprinkle their speech with foreign or foreign-sounding phrases, and in emulation of this practice we have adopted the pronunciation of PADE as "pah-DAY", which incidentally honors the French mathematician Henri Padé (1863-1953). Those who telephone NIST to ask how they can *get paid* may engender a misunderstanding.

Contents

Chapter 1

System Requirements

PADE is designed to execute on workstations running the Unix operating system with the X Window System. PADE operates on only one node of the virtual machine, called the *development host*, therefore the requirements of this workstation are greater than those of the other nodes of the virtual machine.

The development host uses the following software:

- **PADE (required)**. This is included in the PADE package.
- **CAML Utilities (required)**. These are included in the PADE package.
- **PVM** (required, version 3.0 or later). The Parallel Virtual Machine message-passing library is a software system that enables a collection of heterogeneous computers to be used as a coherent and flexible concurrent computational resource. This software can be obtained by anonymous ftp to
  ftp://www.netlib.org/pvm3
- **Tcl/Tk** (required, version 8.0 of Tcl/Tk. *PADE will not run with earlier versions.*)
  Tool Command Language (Tcl) is an embeddable scripting language written by John Ousterhout of the University of California at Berkeley. *ToolKit (Tk)* is an X11 toolkit that provides the Motif look and feel and is implemented in *Tcl*. Both of these programs are available by anonymous ftp at the following site.
  http://sunscript.sun.com/TclTkCore/
- **XPVM** (Optional). This is a graphical console and monitor for PVM. It provides a graphical interface to the PVM console commands and information, along with several animated views to monitor the execution of PVM programs. These views provide information about the interactions among tasks in a parallel PVM program, to assist in debugging and performance tuning. *XPVM* is available by anonymous ftp from the address.
  http://www.netlib.org/pvm3/xpvm
- **Web browser** (Optional). PADE's on-line help and local manual browsing capabilities require a program that can interpret hypertext markup language (HTML) and access documents on the Web.

The machines in the virtual machine that are not used as a development host only require the following software packages:

- **CAML Utilities**
- **PVM**

Chapter 2

Contents of the PADE Package

The PADE package is supplied as a tar file, **nist_pade.1.4.x.tar.gz**, that has been compressed with the GNU *gzip* utility. The package can be obtained by following the instructions on the PADE Web page
http://www.itl.nist.gov/div895/pade/pade.html
For those without access to a Web browser, the PADE package can be obtained by anonymous ftp at
ftp://math.nist.gov/pub/pade
in the file *nist_pade.1.4.x.tar.gz*. This file can be unpacked in any directory in which the user has write privileges with the command

```
gunzip < nist_pade.1.4.x.tar.gz | tar -xvf -
```

Note that this command requires the presence of the GNU *gzip* utility which can be obtained by anonymous ftp to
ftp://prep.ai.mit.edu/pub/gnu
Once unpacked, the user can check that all files are present by comparing a listing of the files with the *MANIFEST* file in the top PADE directory.

PADE unpacks into a directory structure consisting of the top directory (entitled `nist_pade`) and four subdirectories: `pade`, `caml_utils`, `pade_doc`, and `examples`.

The top `nist_pade` directory contains the following four (ASCII) files.

- *RELEASE-NOTES* – this file describes any changes or additions to the PADE package that did not make it into this manual, changes since the previous version, known bugs, etc.
- *INSTALL* – a brief guide to installation of the PADE package.
- *MANIFEST* – a complete listing of all files that should be found in the package.
- *CHECKLIST* – check lists for the actions performed during installation and user set up.

The rest of the PADE package has four principal components:

1. **PADE Interface** – A set of Tcl/Tk scripts that together implement the graphical user environment of PADE. These scripts are contained in the `pade` subdirectory.
2. **CAML Utilities** – There are four utilities in the `caml_utils` subdirectory:
   - *PVMmake* provides both the file transfer and command execution capabilities of PADE.
   - *pvmstat* returns information about the status of the user's virtual machine, whether PVM has been started, and the current configuration of the virtual machine.

- *pvmtidy* enables the user to clean up hopelessly lost PVM daemons and user processes after PVM is halted ungracefully.
- *padeinituser* enables PADE users to create the default directory tree in which PVM binaries are located and to place the NIST CAML utilities in their proper places.

3. **The PADE User's Manual and The PADE Installation Guide** – These documents are provided in both PostScript and ASCII formats as well as an HTML version that is accessible from the PADE `Help` menu. These documents are located in the `pade_doc` subdirectory.

4. **PADE Examples** – A set of sample files to aid the user in following the *manager/worker* example described in the *PADE User's Guide*. Two forms of a PADE setup script (*pade_setenv* for csh and tcsh, and *pade_export* for sh, ksh and bash) to help the user set environment variables and the executables path are also included. These files are found in the `examples` subdirectory.

Chapter 3

PADE Installation

This section describes the installation of PADE on the user's virtual machine. After the PADE distribution has been unpacked as described in Chapter 2, there are three steps to install the PADE package:

1. **Installation of the PADE interface on the development host** – The first line in the main PADE *Tcl* script (the file *pade* in the directory `nist_pade/pade`) must be edited so that it points to the *Tk* windowing shell program *wish*. The first line of this script *must* have the form:

   ```
   #!<directory-path-to-wish>/wish -f
   ```

   where the angle-brackets expression above should be replaced by the absolute pathname of the directory that contains the *wish* program.

2. **Installation of the NIST CAML Utilities on the development host** – Installation of the CAML Utilities requires four steps.

   (a) Edit the file *padeinituser*. Set the variable CAML_UTILS_HOME in this file to the absolute pathname of the top of where the CAML Utilities will be installed (for example, `$PADE_ROOT/caml_utils`).

   (b) Edit the file *Makefile.aimk*. Set the environment variables PVM_ROOT, PVM_ARCH, ARCHLIB and CAMLDIR as directed in that file.

   (c) Type the following command at the Unix prompt.

   ```
   $PVM_ROOT/bin/$PVM_ARCH/aimk
   ```

   This command will create architecture-dependent executables in an ARCH subdirectory of the `caml_utils` directory, where ARCH is the host architecture name, for example, SUN4.

   (d) Type the following command at the Unix prompt.

   ```
   $PVM_ROOT/bin/$PVM_ARCH/aimk install
   ```

   This command will install the NIST CAML Utilities in the directory specified by CAMLDIR.

3. **Installation of the NIST CAML Utilities on other hosts** – The CAML Utilities must also be executable from all nodes of the virtual machine. This means it must be installed on each architecture/file-system pair of the virtual machine. First, transfer the `caml_utils` directory to every file system. Note that it does not *have* to be under the `nist_pade` directory on non-development hosts. In many cases, transfer can be accomplished by the command

```
rcp -r caml_utils remote-host-name:/path-to-caml_utils-directory
```

If *rcp* is not available, the `caml_utils` directory can be packed with

```
tar -cvf caml_utils.tar caml_utils
```

transfered by *ftp*, and unpacked with

```
tar -xvf caml_utils.tar
```

Once transfered, compile the CAML Utilities as described above.

Chapter 4

Systems Tested and Known Bugs

The development team has run PADE on the following operating systems and computers to date:

- NeXT OS 3.1/NeXT Motorola 68040
- NEXTSTEP 3.2/Intel Pentium
- AIX 3.2.5/IBM RS/6000
- IRIX 5.2 and 5.3/Silicon Graphics
- SunOS 4.1.3/SUN SparcStation 20
- Linux 1.0.9 and 1.2/Intel 80486
- UNICOS 8.0.3.2/Cray Y-MP
- ConvexOS 11.0.149/Convex 3820
- HP-UX 9.05/HP 735

This list is presented to indicate the range of testing that has taken place, not as a guarantee of performance on any particular system. We would appreciate receiving information on usage on other systems. Send electronic mail to the following address.

`robert.lipman@nist.gov`

It should be kept in mind that the functionality of PADE depends upon that of the other programs mentioned in Chapter 1, and the recommended first step in troubleshooting is to verify that they are properly implemented on the development host.

The following problems have been encountered during testing:

- On some hosts, *PVMmake* has difficulty finding the *wc* program. *PVMmake* uses this program (for packing purposes) to count the number of lines in a file that is to be transferred. If this error occurs, *PVMmake* will so indicate and exit. In this case, the user should create a symlink in the PVM binary directory to the actual location of the *wc* program.
- On the NIST Cray Y-MP, *PVMmake* requires the definition of the macro `O_RDONLY`. If *PVMmake* fails to compile, edit the file *jlib.h* in the `caml_utils` directory to remove the comment around line 9.

- On rare occations, PADE has produced the error message

  ```
  invalid command name "FindCmd"
  ```

  before creating the main PADE window. This indicates that either the PADE_ROOT environment variable points to the wrong directory, or there is a problem with the *tclIndex* file. If PADE_ROOT is set correctly, then regenerate the *tclIndex* file by:

  ```
  host-prompt> tclsh
  tcl-prompt> auto_mkindex . *.tcl
  tcl-prompt> exit
  host-prompt>
  ```