Paradyn Parallel Performance Tools

Installation Guide

Release 5.1
May 2007

Paradyn Project
Computer Sciences Department
University of Wisconsin
Madison, WI 53706-1685
`paradyn@cs.wisc.edu`

Table of Contents

1 OVERVIEW

The Paradyn Parallel Performance Tools are available for research use free of charge. This Guide describes how to obtain, configure, and install the binary version of Paradyn. A source code distribution is also available.

The Paradyn installation process is similar for all supported UNIX and UNIX-like operating systems, and is described in Section 2. The installation process for Windows is described in Section . Currently, the installation process for Windows is similar to that for UNIX, so you may wish to read Section 2 before reading Section .

Paradyn supports the IBM PMAPI performance library. This allows us to access the hardware cycle timers on IBM machines. See Section 5 for information on obtaining and installing the necessary support libraries.

1.1 Supported platforms

The Paradyn user interface can run on any of the systems listed in Figure 1, which are also the same workstations and parallel computers on which Paradyn can monitor programs. Paradyn can also monitor application programs running on heterogeneous combinations of these platforms.

System Identifier	Description
sparc-sun-solaris2.9	Sun Solaris operating system version 9, running on SPARC processors.
i386-unknown-linux2.4	Linux operating system with kernel version 2.4 or 2.6, running on x86 processors.
i386-unknown-nt4.0	Microsoft Windows NT operating system, version 4.0, running on x86 processors. This should work, but we no longer test on this platform.
i386-unknown-w2k	Microsoft Windows 2000 operating system on x86 processors.
rs6000-ibm-winxp	Microsoft Windows XP operating system on x86 processors.
rs6000-ibm-aix5.1[a]	IBM AIX operating system version 5.1, running on rs6000 processors.
ia64-unknown-linux2.4	Linux 2.4 or 2.6 running on Itanium 2 (or higher) processors.
x86-64-unknown-linux2.4	Linux 2.4 or 2.6 running on x86-64 based systems.

Figure 1: Platforms on which Paradyn's user interface can run

a. Power3-based AIX systems will only function properly if APAR IY03550 has been applied. This is detailed in the AIX installation notes, below.

Paradyn supports monitoring MPI applications on IBM AIX, Solaris, and Linux platforms.

2 INSTALLING PARADYN FOR UNIX

This section describes the steps that should be followed to install the binary distribution of Paradyn on a UNIX system. The steps are: prepare a Paradyn directory and obtain the desired files (Section 2.1), process the installation archive files (Section 2.2), install Tcl/Tk if necessary (Section 2.3), configure the paths in make.config.local (Section 2.4), and set environment variables and executables search path (Section 2.5).

2.1 Prepare a Paradyn directory and obtain the distribution archive

It is recommended that you install Paradyn and its associated files into its own directory. If you have any previous versions of Paradyn installed, you may wish to create a new directory for this release to avoid confusion between the installations. After creating an installation directory, the next step is to download the version(s) of Paradyn that match your system.

You can obtain the Paradyn binary distribution from the Paradyn web site at `www.paradyn.org`, or by anonymous ftp from `ftp.cs.wisc.edu`, in the `paradyn` directory. This directory contains `README` and `LICENSE` files, as well as the archive files containing the Paradyn binaries named according to their platform. The archive files are `tar` archives, compressed using the `gzip` utility.[1] (E.g., `solaris_sparc_5.0.tar.gz` is the Paradyn 5.0 binary distribution for SPARC Solaris, archived by `tar` and then compressed with `gzip`.) The archives also contain several Paradyn test applications and the headers needed to build or interface your own tools to various Paradyn libraries.

The ftp directory contains several subdirectories. The `doc` directory contains gzip'd tar archives of the Paradyn manuals (including the ***Paradyn User's Guide*** and this document) in both PostScript and PDF formats[2]. The `etc` directory contains additional software Paradyn may require which may not be installed on your system. The `src` directory contains a gzip'd tar archive of the source code and Makefiles for building Paradyn. (For further information on the source code organization and contents, please refer to the ***Paradyn Developer's Guide***).

For example, to download the SPARC Solaris version of Paradyn and the entire set of Paradyn manuals in PDF format using ftp, the session might appear as follows:

```
% ftp ftp.cs.wisc.edu
Name: anonymous
331 Guest login ok, send your complete e-mail address as password.
ftp> binary
200 Type set to I.
ftp> cd pub/paradyn/releases/current_release
ftp> get solaris_sparc_5.0.tar.gz
ftp> get PDF_manuals_5.0.tar.gz
ftp> quit
```

1. The `tar` utility is available on most UNIX systems. The `gzip` utility is available from the many web sites that distribute GNU sources. Windows users may obtain `tar` and `gzip` implementations as part of the Cygwin package, available at www.cygnus.com/cygwin.
2. PDF documents are designed for on-line reference and require the Acrobat reader (available for free for many platforms from Adobe at www.adobe.com), or a similar viewer that supports PDF such as a recent version of GhostView (*gv*).

2.2 Process distribution archive

Once you have obtained the appropriate archive for your platform (e.g., `sparc_solaris_5.0.tar.gz`) and any documents you desire (e.g., `users_guide.tar.gz`), you need to *unzip* and *untar* these files to extract Paradyn's executables, libraries, and other suport files. First, make sure you are in your installation directory. Then, perform commands similar to the following to unzip and untar the archives:

```
% gunzip -c sparc_solaris_5.0.tar.gz | tar xvf -
% gunzip -c PDF_manuals.tar.gz | tar xvf -
```

After executing these commands, your Paradyn installation directory should look similar to the listing below. (DyninstAPI components are indicated with italics in the figure.) Notice that the subdirectories *sparc-sun-solaris2.8/bin*, *sparc-sun-solaris2.8/lib*, etc., have been created automatically by the `tar` command. A description of all the Paradyn files is given in Figure 2.

Paradyn Distribution File Layout

```
solaris_sparc_5.0.tar.gz (as an example)
     paradyn.rc
     /bin
          igen
          paradyn
          paradynd
          tclVisi
          barChart
          tableVisi
          phaseTable
          rthist
          terrain
          termWin
          dyner
     /lib
          libpdutil.a
          libpdthread.so
          libparadynRT.so.1
          libdyninstAPI_RT.so.1
          libvisi.a
          libdyninstAPI.so
     /include
          /dyninstAPI
          /visi
PDF_manuals_4.0.tar.gz
     /PDF_manuals
          developerGuide.pdf
```

```
          installationGuide.pdf
          libthread.pdf
          MDLprogGuide.pdf
          Overview.pdf
          tutorial.pdf
          userGuide.pdf
          visiprogGuide.pdf
          dyninstAPI_TestSuite.pdf
          dyninstProgGuide.v40.pdf
          dynerGuide.v40.pdf
PS_manuals_4.0.tar.gz
     /PS_manuals
paradyn_apps_4.0.tar.gz
     /apps
          /README
          /bubba_seq
          /decomp_MPI
```

2.3 Install Tcl/Tk

The user interface of Paradyn is implemented using the Tcl and Tk packages, currently maintained by Scriptics Corporation. On all platforms, Paradyn needs to find several files from Tcl and Tk version 8.4.x or 8.3.x in order to run. Furthermore, the executables included in the Paradyn binary distribution are built to use the Tcl and Tk libraries as shared objects.

The installation of Tcl/Tk for use with Paradyn involves two steps. The first step, discussed in this section, is to obtain the necessary files if you don't already have Tcl/Tk installed. The second step is to set environment variables which tell Paradyn where Tcl and Tk are installed, and is discussed in Section 2.5.

If you already have appropriate versions of Tcl/Tk installed on your system, then you simply need to determine where the packages were installed. (If you can't find the files, your system administrator should be able to tell you where they are located.) You are now ready to continue

File	Description
`paradyn.rc`	Paradyn configuration file containing default definitions for performance metrics and visualizations.
`paradyn` `paradyn.exe`	Paradyn's user interface (or "*front-end*").
`paradynd` `paradynd.exe`	Paradyn's daemon: runs on every host or node where Paradyn is monitoring application processes.
`termWin`	Window that displays output from the application process(es). *(Not yet available on Windows)*
`barChart` `barchart.exe`	*BarChart* visualizer.
`phaseTable` `phaseTable.exe`	*PhaseTable* visualizer.
`rthist` `rthist.exe`	*Runtime Histogram* visualizer.
`tableVisi` `tableVisi.exe`	*Statistics Table* visualizer.
`terrain`	*3D Terrain* visualizer. (*Not yet available on Windows*)
`libparadynRT.so.1` `libparadynRT.dll`	Paradyn's dynamically loaded run-time dynamic instrumentation library.

`libvisi.a`
`libvisi.lib`
`include/visi/*.h`
`tclVisi`
`tclVisi.exe`
	Library API and include files used to build Visis which interface with Paradyn, and a Tcl interface to allow use of Paradyn performance data in Tcl applications. For more information, refer to the ***VisiLib Programmer's Guide***.

`libdyninstAPI_RT.so.1` `libdyninstAPI_RT.dll`	DyninstAPI runtime library. This is needed when using Paradyn.

`libdyninstAPI.so`
`libdyninstAPI.dll`
`dyninstAPI/*.h`
	DyninstAPI mutator library and include files distributed with Paradyn, but not needed when using Paradyn. For more information, refer to the ***DyninstAPI Programmer's Guide***.

`apps/bubba_seq`	Sample sequential application (for all current platforms).
`decomp/decomp_MPI`	Sample MPI application.

Figure 2: Files included in Paradyn binary distribution

with the Paradyn installation and can skip to the next section.

You can obtain a binary Tcl/Tk distribution from `http://tcl.activestate.com`, and source distribution from `http://www.sourceforge.net`. If you do this, ensure that you build the Tcl and Tk libraries as shared objects and note where they are installed.

2.4 Setup make.config.local

The make.config.local file tells paradyn where to find the necessary packages on your system. Edit this file and point the variables at the appropriate packages. A sample make.config.local that is used in Wisconsin is below:

```
#TCL/TK is required for all Paradyn builds, but not DyninstAPI
TCLTK_DIR       = /p/paradyn/packages/tcl8.4.5
TCLTK_LIB_DIR   = $(TCLTK_DIR)/lib
TCLTK_INC_DIR   = $(TCLTK_DIR)/include

#Libelf is required by Linux/x86 and Linux/ia64
LIBELFDIR       = /p/paradyn/packages/libelf
LIBELF_LIB      = $(LIBELFDIR)/lib
LIBELF_INC      = $(LIBELFDIR)/include

#Libdwarf is required by Linux/x86 and Linux/ia64; released 2006-03-27 or later.
LIBDWARFDIR     = /p/paradyn/packages/libdwarf
LIBDWARF_LIB    = $(LIBDWARFDIR)/lib/
LIBDWARF_INC    = $(LIBDWARFDIR)/include

#libunwind is required by Linux/ia64; version 0.98.5 or higher.
LIBUNWINDDIR    = /p/paradyn/packages/libunwind
LIBUNWIND_LIB   = $(LIBUNWINDDIR)/lib
LIBUNWIND_INC   = $(LIBUNWINDDIR)/include

#Uncomment the following line to build Paradyn with Tcl/Tk 8.3
# instead of 8.4
#USE_TCL83      = true

#Uncomment these lines to force the use static libraries
#LIBELFSTATIC   = true
#LIBDWARFSTATIC = true
#LIBTCLTKSTATIC = true

#nasm is needed for the Linux/x86 dyninst tests
NASM            = /p/paradyn/packages/nasm/bin/nasm

# Set to the directory containing libflex.a if it's in an unusual place.
# Needed to build MRNet and Paradyn.
ifeq ($(PLATFORM),ia64-unknown-linux2.4)
        FLEX_DIR                = /usr/lib
else
        FLEX_DIR                = /s/flex/lib
endif
```

Libelf and libdwarf are required only on Linux platforms. Libunwind is required on

Linux/IA64. TCL/TK is required on all platforms. To make Paradyn build with Tcl/Tk 8.3 set USE_TCL83 to true.

2.5 Set environment variables

To operate, Paradyn requires the following environment variables to be defined:

- `PARADYN_ROOT` - the directory containing the *paradyn.rc* configuration file
- `PLATFORM` – If paradyn cannote determine your platform, you may have to indicate the hardware/software platform (see Figure 4 for a list of current valid `PLATFORM` values)
- `PARADYN_LIB` - the full path (including the file name) to the Paradyn runtime dynamic instrumentation library, `libparadynRT.so.1`
- `DYNINSTAPI_RT_LIB` - the full path (including file name) to the Dyninst runtime dynamic instrumentation library, `libdyninstAPI_RT.so.1`
- `LD_LIBRARY_PATH` - (UNIX only) update so that the directory that contains the libpdthread.so file is specified, which is used by the front-end. On Linux, update so that the directory that contains the libdwarf.so file (which can be retrieved off of the release web page) is specified. The libdwarf.so file is used by the Paradyn daemon on Linux. See figure Figure 3 for help in setting this variable.
- `LIBPATH` - (AIX only) - set to the directory that contains the runtime dynamic instrumentation libraries, such as `libparadynRT.so.1` and `libdyninstAPI_RT.so.1`
- `PARADYN_RSH` - indicates the program to use when starting daemons on remote hosts. The pcl file variable, remoteShell is an alternative way to specify the shell used for starting remote daemons. For example, one might set it to 'ssh'.

o To allow instrumentation of multi-threaded programs, additionally define the following.

- `PARADYN_LIB_MT` - the full path (including file name) to the Dyninst runtime dynamic instrumentation library, `libparadynMT.so.1`
- `AIXTHREAD_SCOPE` - (AIX only) set to the value of 'S'. For example, in c-shell, do: setenv AIXTHREAD_SCOPE S

In addition, you must add the location of the Paradyn executables to your `PATH` environment variable. Finally, you may need to update the value of your `LD_LIBRARY_PATH` environment variable: it must contain both the directory that contains the Tcl and Tk shared object files and the directory that contains the C++ standard library, `libstdc++.so`. Note that, because Paradyn is a C++ application, you may need to ensure that you have a copy of libstdc++.so that is ABI-compatible with Paradyn in order to run the Paradyn front-end or the Paradyn daemon. The README file in the distribution archive indicates which compiler version the Paradyn binaries were compiled with.

Tcl/Tk generally can locate its initialization files from directories specified when it is built and installated: if this isn't the case, you may also need to define two additional environment variables, `TCL_LIBRARY` and `TK_LIBRARY` which specify the correct locations for the required initialization files (e.g., *init.tcl, tk.tcl*).

Figure 3 shows example commands to set the necessary environment variables as described above. You may wish to add these commands to your shell configuration file (e.g., *.cshrc* if you are using *csh*) so the environment variables are set correctly each time you log on.

Operation: Script commands:

1. Setting `PARADYN_ROOT`, `PLATFORM` & `PARADYN_LIB` environment variables:

csh, tcsh:
```
setenv PARADYN_ROOT ~/paradyn
setenv PLATFORM sparc-sun-solaris2.8
setenv PARADYN_LIB $PARADYN_ROOT/$PLATFORM/lib/libparadynRT.so.1
```

sh, bash, ksh:
```
PARADYN_ROOT=$HOME/paradyn
PLATFORM=sparc-sun-solaris2.8
PARADYN_LIB=$PARADYN_ROOT/$PLATFORM/lib/libparadynRT.so.1

export PARADYN_ROOT PLATFORM PARADYN_LIB
```

2. Adding `paradyn` executables to your search path:

csh, tcsh:
```
set path=(~/paradyn/sparc-sun-solaris2.8/bin $path)
```

sh, bash, ksh:
```
PATH=$HOME/paradyn/sparc-sun-solaris2.8/bin:$PATH
export PATH
```

3. Updating LD_LIBRARY_PATH to point to directory containing libtcl8.4.so and libtk8.4.so.

- For this example, the directory ~/tcl8.4.0/lib contains the tcl and tk libraries. This directory will be dependent on one’s system configuration.

csh, tcsh:
```
setenv LD_LIBRARY_PATH ~/tcl8.4.0/lib:$LD_LIBRARY_PATH
```

sh, bash, ksh:
```
LD_LIBRARY_PATH=~/tcl8.4.0/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

Figure 3: Examples of shell and environment variable operations for common UNIX shells

System Identifier	Set PLATFORM variable to:
sparc-sun-solaris2.8	sparc-sun-solaris2.8
sparc-sun-solaris2.9	sparc-sun-solaris2.8
i386-unknown-linux2.4	i386-unknown-linux2.4
i386-unknown-nt4.0	i386-unknown-nt4.0
i386-unknown-w2k	i386-unknown-nt4.0
i386-unknown-winxp	i386-unknown-nt4.0
rs6000-ibm-aix4.3	rs6000-ibm-aix5.1
rs6000-ibm-aix5.1	rs6000-ibm-aix5.1

Figure 4: Platform variable for each system identifier

3 INSTALLING PARADYN FOR WINDOWS

Currently, the process for installing Paradyn on Windows is similar to that used on a UNIX system. However, there are some significant differences, which are detailed in this section.

Because the installation process is modelled after the UNIX installation process, several of the steps require you to issue commands rather than interact with the Windows GUI shell. A command prompt is accessible from the Start menu by selecting the "Command Prompt" item under the "Programs" submenu. Alternatively, you may wish to use a "UNIX on Windows" package such as Cygwin (`http://www.cygnus.com/cygwin`) to obtain a more UNIX-like command shell and set of utilities.

3.1 Prepare a Paradyn folder and obtain the distribution archive

As stated above, it is desirable to install each release of Paradyn into its own directory ("folder" in Windows parlance). Create a folder for Paradyn using the graphical shell, or by issuing an appropriate `mkdir` command from a command shell.

The process for obtaining the Paradyn binary distribution is the same for UNIX and Windows. See Section 2.1 for details on obtaining the distribution archive.

3.2 Process distribution archive

Like the UNIX binary distribution, the Paradyn binary distribution for Windows is a gzip'd tar archive. To extract the files in the archive, you must have access to an implementation of tar and gzip for Windows. One source for these utilities is the Cygwin package, available at (`http://www.cygnus.com/cygwin`).

To extract the Paradyn files from the archive, issue the following commands at a command prompt (we assume that the archive file is located in the Paradyn installation folder, and that that folder is named “`d:\Paradyn`”):

```
d:
cd \Paradyn
gunzip nt_x86_4.1.tar.gz
tar -xvf nt_x86_4.1.tar
```

3.3 Install Tcl/Tk

As on UNIX platforms, Paradyn on Windows uses Tcl/Tk to implement its user interface. Because there is an installer program for Tcl/Tk on Windows, we suggest that you install Tcl/Tk by downloading the “official” Tcl/Tk binary distribution from `http://tcl.activestate.com`. (Note where you installed Tcl/Tk in case you need to know the location of its initialization files to set the `TCL_LIBRARY` and `TK_LIBRARY` environment variables).

3.4 Set environment variables

In general, the same environment variables needed to run Paradyn on a UNIX system are needed to run Paradyn under Windows. Since Windows uses the `PATH` environment variable to find dynamic link libraries, the `LD_LIBRARY_PATH` variable need not be defined.

Unlike the UNIX method of setting environment variables in a command shell initialization file (like *.cshrc*), you may set environment variables for Windows using the System control panel, accessible via the Settings item in the Start menu. In this control panel, select the Environment tabbed page. To update an existing variable (e.g., for updating the `PATH` variable), select the variable in the “User Variables” section of the dialog, update the value in the Value field, and click the Set button to apply your change. To add a new variable (e.g., for the `PARADYN_ROOT` variable), select *any* variable in the “User Variables” section of the dialog, change both the name and value of the variable in the Variable and Value fields, and click the Set button.

Note that changes made in to the environment are not visible to programs that are already running, so if you have a Command Prompt running when you add the Paradyn variables, they will not be set in the existing Command Prompt shell. They will be available, however, for Command Prompt shells created after the environment variable modifications are complete.

If you are using a CYGWIN shell on Windows, here are some examples of how the PARADYN_ROOT and PARADYN_LIB environment variables can be set. You’ll of course need to adjust the paths specified below according to where you have Paradyn installed on your system.

```
export PARADYN_ROOT='P:/paradyn/config'

export PARADYN_LIB='P:/paradyn/i386-unknown-nt4.0/lib/libparadynRT.dll'
```

4 LINUX INSTALLATION NOTES

Note for users of Paradyn 3.x: Previous releases of the Paradyn tools supported high-resolution hardware timers on Linux via the `hrtime` kernel patch and `libhrtime` user-space library. `hrtime` and `libhrtime` are deprecated and no longer maintained; a future version of the Paradyn tools will provide access to high-resolution hardware timers on Linux via an alternative mechanism.

5 AIX INSTALLATION NOTES

Paradyn uses a section of the target program's data space for its instrumentation. On the Power3 processor, and any other processor which has split data and instruction caches, changes made using ptrace() to a data space will not necessarily be executed correctly. This behavior causes Paradyn to appear to hang with the message "installing default inst..." before dying. To correct this problem, apply APAR IY03550 -- DBX CANNOT SET BREAK POINTS IN DATA SECTION.

Paradyn supports the IBM PMAPI performance library, available from *http://www.alphaworks.ibm.com/tech/pmapi*. To build Paradyn with PMAPI support, first download and install PMAPI from IBM. Then edit the top level make.config file in the Paradyn build directory and uncomment the USES_PERF_LIBRARY definition. The default Paradyn distribution does not include support for the PMAPI library.

Running Paradyn on AIX requires setting some additional environment variables. Make sure that your LIBPATH environment variable includes the directory into which you installed `libDyninstText.a`. Also, set the environment variable AIXTHREAD_SCOPE to `S`.

```
export LIBPATH=/path-to-paradyn-install/rs6000-ibm-aix4.2/lib:${LIBPATH}
export AIXTHREAD_SCOPE=S
```

6 COMMON INSTALLATION ERRORS

Figure 5 is a list of common installation errors, with a description of the error and its possible cause which may help you in case you have problems during the installation process. If you have any problem that does not appear on this list or if you have any comments, please send us a detailed message to `paradyn@cs.wisc.edu`.

Error	Description	Possible solution
paradyn: Command not found.	Cannot find or execute the `paradyn` executable.	Check the value of the `PATH` variable. The `paradyn` executable is not in your command search path.
tcl_init() failed.	Cannot start paradyn.	Tcl checks several locations to locate its initialization files, such as init.tcl - depending on your Tcl/Tk installation, you may need to define environment variables `TCL_LIBRARY` and `TK_LIBRARY`, pointing to the correct locations. Also, make sure that you have the required versions of Tcl and Tk installed. For more information, see Section 2.3. On UNIX systems, make sure you have the `DISPLAY` variable set, and that you have permission to use the X display (see the `xhost` man page for details).
paradyn: Exec format error. Wrong Architecture.	Cannot start paradyn.	You are perhaps trying to start Paradyn using the executables for a different architecture. Check that you have the binaries to match your platform.
paradyn: cannot create daemon process.	Cannot start paradynd.	`paradynd` may not be on your command search path or you may be using a `paradynd` executable for a different platform.
`PARADYN_LIB` has not been defined for process	Cannot access runtime instrumentation library `libparadynRT.`	The `PARADYN_LIB` environment variable should contain the full path name of the library. The file must also be readable.
The dynamic library `oncrpc.dll` could not be found on the path.	Cannot start paradynd.	When running on Windows, the RPC library `oncrpc.dll` should be installed and locatable on the environment `PATH`.
visi processes not available.	There are no visi processes defined in Paradyn.	Check the value of `PARADYN_ROOT` and make sure that it is the directory containing the file `paradyn.rc`.

Figure 5: Common installation errors