Performance Measurement and Analysis Tools Installation Guide S-2474-63

Contents

About Cray Performance Measurement and Analysis Tools

This publication includes software installation procedures for Cray Performance Measurement and Analysis Tools on Cray XK and XE systems.

Publication Date

This is release 6.3.0 of this publication. This version was published on September 24, 2015.

Typographic Conventions

`Monospace`	Indicates program code, reserved words, library functions, command-line prompts, screen output, file/path names, key strokes (e.g., `Enter` and `Alt-Ctrl-F`), and other software constructs.
`Monospaced Bold`	Indicates commands that must be entered on a command line or in response to an interactive prompt.
`Oblique` or *`Italics`*	Indicates user-supplied values in commands or syntax definitions.
Proportional Bold	Indicates a graphical user interface window or element.
\ (backslash)	At the end of a command line, indicates the Linux® shell line continuation character (lines joined by a backslash are parsed as a single line). Do not type anything after the backslash or the continuation feature will not work correctly.

Scope and Audience

This publication is written for users wishing to install Cray Performance Measurement and Analysis tools on Cray XK or XE systems.

Feedback

Visit the Cray Publications Portal at *http://pubs.cray.com* and make comments online using the *Contact Us* button in the upper-right corner or Email *pubs@cray.com*. Your comments are important to us and we will respond within 24 hours.

Install Performance Measurement and Analysis Tools on Cray Systems

This publication includes instructions for installation on Cray XE and Cray XK systems. The Cray Performance Measurement and Analysis Tools package is distributed on DVD-ROM media. It is also available as a downloadable file. Instructions to install the performance tools package on the user's system are included in this publication. These instructions assume that the user is working with a DVD-ROM, and that has `root` permission on the Cray system.

For Cray XC series systems, the Cray Performance Measurement and Analysis Tools package is distributed and installed as part of the Cray Developers Toolkit (CDT). For installation instructions and more information, see Cray Programming Environments Installation Guide (S-2372).

For Cray CS300 systems, the Cray Performance Measurement and Analysis Tools package is distributed and installed separately. For installation instructions and more information, see Installing the Cray Programming Environment for CS300 (S-2800).

The process for installing the Cray Performance Measurement and Analysis Tools package varies depending on the Cray system on which it is being installed. For information about the new features and changes included in this release, or limitations on specific hardware platforms, see the Perftools 6.3 Release Notes.

Install the Performance Tools `rpm` files

1. Log onto the SMW as `root`.

   ```
   % ssh root@smw
   ```

2. Load and mount the distribution media, if necessary.

   ```
   smw:~# mount /dev/cdrom /media/cdrom
   ```

3. Create a temporary directory on the boot node for the installation files, if one does not already exist.

   ```
   smw:~# ssh boot mkdir /tmp/install.perftools
   ```

4. Copy the installation files from the distribution media to the boot node.

   ```
   smw:~# scp -pr /media/cdrom/cray-dwarf-version-release.x86_64.rpm \
   boot:/tmp/install.perftools
   smw:~# scp -pr /media/cdrom/craypkg-perftools-utils-version-release.x86_64.rpm \
   boot:/tmp/install.perftools
   smw:~# scp -pr /media/cdrom/perftools-version-release.x86_64.rpm \
   boot:/tmp/install.perftools
   ```

```
smw:~# scp -pr /media/cdrom/perftools-clients-version-release.x86_64.rpm \
boot:/tmp/install.perftools
smw:~# scp -pr /media/cdrom/cray-papi-version-release.x86_64.rpm \
boot:/tmp/install.perftools
```

> **NOTE:** The RPM file `cray-papi-acc-version-release.x86_64.rpm` is no longer required for Cray XK systems. This capability has been merged into `cray-papi-version-release.x86_64.rpm`.

5. Unmount and remove the distribution media.

```
smw:~# umount /media/cdrom
```

6. Log into the boot node as `root`.

```
smw:~# ssh root@boot
```

7. Change to your temporary directory.

```
boot001:~# cd /tmp/install.perftools
```

8. Create a target directory on the shared root and copy the installation files from your temporary directory to the shared root.

```
boot001/:/tmp/install.perftools # mkdir -p /rr/current/software/
install.perftools
boot001/:/tmp/install.perftools # cp -p cray-dwarf-version-release.x86_64.rpm \
/rr/current/software/install.perftools
boot001/:/tmp/install.perftools # cp -p craypkg-perftools-utils-version-
release.x86_64.rpm \
boot:/tmp/install.perftools
boot001/:/tmp/install.perftools # cp -p perftools-version-release.x86_64.rpm \
/rr/current/software/install.perftools
boot001/:/tmp/install.perftools # cp -p perftools-clients-version-
release.x86_64.rpm \
/rr/current/software/install.perftools
boot001/:/tmp/install.perftools # cp -p cray-papi-version-release.x86_64.rpm \
/rr/current/software/install.perftools
```

9. Open an `xtopview` session.

```
boot001/:/tmp/install.perftools # xtopview
```

10. Change to the temporary directory you created on the shared root.

```
default/:/# cd /software/install.perftools
```

11. Optional: If you want the versions you are about to install to become the new default versions, set the environment variable.

```
default/:/software/install.cpat # export CRAY_INSTALL_DEFAULT=1
```

If you do not set this environment variable, any previously installed default version remains the default version, and your users will need to load a specific module in order to select the newly installed version.

12. Optional: Use the `rpm` command to install the files.

NOTE: When running `rpm` from within `xtopview`, the `rpm` utility issues a warning that it cannot find `/rr/current`. This warning may safely be ignored.

To install the performance analysis tools for use on a Cray system, use these commands:

```
default/:/software/install.perftools # rpm -ivh cray-dwarf-version-
release.x86_64.rpm
default/:/software/install.perftools # rpm -ivh craypkg-perftools-utils-
version-release.x86_64.rpm \
boot:/tmp/install.perftools
default/:/software/install.perftools # rpm -ivh --oldpackage perftools-
version-release.x86_64.rpm
default/:/software/install.perftools # rpm -ivh --oldpackage \
perftools-clients-version-release.x86_64.rpm
default/:/software/install.perftools # rpm -ivh --oldpackage cray-papi-
version-release.x86_64.rpm
```

13. Optional: After RPM file installation is complete, if you set the `CRAY_INSTALL_DEFAULT` environment variable earlier, unset it now:

```
default/:/software/install.perftools # unset CRAY_INSTALL_DEFAULT
```

14. Optional: Exit from the `xtopview` session:

```
default/:/software/install.perftools # exit
```

15. Optional: Log out of the boot node:

```
boot001/:/tmp/install.perftools # exit
logout
Connection to boot closed.
smw:~#
```

16. Optional: Log out of the SMW.

```
smw:~# exit
logout
%
```

FlexNet License Key Activation

Cray Performance Measurement and Analysis Tools release 6.2.0 or later requires a new FlexNet software license key for all installations. Even if you are upgrading from an earlier version of the Cray Performance Measurement and Analysis Tools, you must obtain and install a new license key.

To activate your software license, insert the FlexNet software license key information provided by Cray into a FlexNet license file on your system. The FlexNet license file contains data that determines whether a licensed software product is allowed to run.

The license file contains the following information:

- The FlexNet software license key for your Cray Inc. product
- Initial installation instructions
- Update instructions
- License manager utilities
- Technical Support information

Cray Inc. recommends that you name your license file `/opt/cray/perftools/perftools.lic`. These instructions assume that the FlexNet license manager is already running, that your license file is located in the directory `/opt/cray/perftools`, and that the file is named `perftools.lic`.

The FlexNet license manager should be installed on your system already. If it is not, follow the installation instructions in Appendix A, "Installing FlexNet," in *Cray Compiling Environment Release Overview and Installation Guide* (S-5212).

Add a new key to a license file

1. Optional: Login to the license server as administrator or superuser.
2. Optional: Locate the existing license file, if any.

   ```
   # ls /opt/cray/perftools
   ```

   If the directory does not exist, create it.

   ```
   # mkdir -p /opt/cray/perftools
   ```

3. Optional: In `/opt/cray/perftools`, create the plain text file `perftools.lic`. Copy the FlexNet license key you received from Cray to `perftools.lic`.
4. Optional: Set the file access permissions to 644.

```
# chmod 644 /opt/cray/perftools/perftools.lic
```

5. Optional: Update the FlexNet license server to use the new key. Verify that the license server is running.

   ```
   # lmstat
   ```

   If the server is not running, follow the installation instructions in Appendix A, Installing FlexNet, in *Cray Compiling Environment Release Overview and Installation Guide* (S-5212).

   Assuming the server is running, re-read the license file.

   ```
   # lmreread
   ```

   The license is now ready to use.

Install Cray Apprentice2 on Microsoft Windows

This release includes a version of Cray Apprentice2 that can be installed and used on Microsoft Windows systems.

> **NOTE:** The Windows version works on Windows 7 only. It is not supported on earlier versions of the Microsoft Windows operating system and is untested on Microsoft Windows 8 at this time.
>
> To install this version of Cray Apprentice2 on Windows, follow these steps.

1. Optional: Locate the installer file. The Windows installer file is named `Apprentice2Installer-`*`version`*`.exe` and is included on the distribution media. For added convenience, it is also installed on the Cray system in the `$CRAYPAT_ROOT/share/desktop_installers` directory.
2. Optional: Copy the installer to the Windows system.
3. Optional: Double-click on the installer file to begin the installation.
4. Optional: Follow the on-screen prompts to complete the installation process.

   After Cray Apprentice2 is installed on the Windows system, launch it either by double-clicking on the Cray Apprentice2 desktop icon, or by double-clicking on an `.ap2` file.

Install Cray Apprentice2 on Apple Macintosh

This release includes a version of Cray Apprentice2 that can be installed and used on Apple Macintosh systems. The Mac version works on Mac OS X only. It is not supported on earlier versions of the Mac operating system. It does not work on the iPad at this time.

To install this version of Cray Apprentice2, follow these steps.

1. Optional: Locate the installer file. The Mac installer file is named `Apprentice2Installer-`*`version`*`.dmg` and is included on the distribution media. For added convenience, it is also installed on the Cray system in the `$CRAYPAT_ROOT/share/desktop_installers` directory.
2. Optional: Copy the installer to the Macintosh system being used.
3. Optional: Double-click on the installer file to begin the installation.
4. Optional: Follow the on-screen prompts to complete the installation process.

   After Cray Apprentice2 is installed on the Mac system, launch it either by double-clicking on the Cray Apprentice2 desktop icon or by double-clicking on an `.ap2` file.

Use the Cray Performance Measurement and Analysis Tools

Assuming your site has the correct licenses, use the `module` command to load the tools.

Man pages are included in the associated Module files and become available only after the Module file is loaded.

Use Modules

The `module` command can accept a number of arguments. The arguments most commonly used are listed in *Commonly Used Module Arguments*.

Table 1. Commonly Used Module Arguments

Argument	Description
`list`	View the list of modules that are currently loaded
`avail`	View the list of modules currently available to be loaded
`load`	Load a module file
`swap`	Swap a currently loaded module for another module
`unload`	Unload a currently loaded module file without swapping it for another module
`use`	Use a different set of module files
`help`	Release notes and module command usage information

Use CrayPat, CrayPat-lite, Apprentice2, or Reveal

To use CrayPat, CrayPat-lite, Cray Apprentice2, or Reveal, load the `perftools-base` module. This module provides access to man pages, utilities such as Reveal, Cray Apprentice2, and `grid_order`, and instrumentation modules. It does not add compiler flags to enable performance data collection (such as symbol table information), as the earlier `perftools` or `perftools-lite` did or the newly available instrumentation modules do. It is a low-impact module that does not alter program behavior and can be left loaded even when building and running programs without CrayPat instrumentation.

Once the `perftools-base` module is loaded, the instrumentation modules are available for use. Use the `module avail perftools` command to view the list of available instrumentation modules.

```
$ module avail perftools
```

The following instrumentation modules are currently supported.

Table 2. Perftools Instrumentation Modules

Module	Function
`perftools`	Full access to CrayPat functionality. Load this instrumentation module then use `pat_build` to instrument a program for performance data collection or `pat_report` to process the collected data and generate reports. This is equivalent to loading the `perftools` module in earlier releases.
`perftools-lite`	CrayPat-lite program profiling. Load this instrumentation module and then build and run the program to get a basic performance profile sent to `stdout`. This is equivalent to loading the `perftools-lite` module in earlier releases.
`perftools-lite-events`	CrayPat-lite event profile. Load this instrumentation module and then build and run the program to get more in-depth performance data, which results in an event profile being sent to `stdout`. This is equivalent to loading the `perftools-lite` module and setting the `CRAYPAT_LITE` environment variable to `event_profile` in earlier releases.
`perftools-lite-gpu`	CrayPat-lite GPU kernel and data movement information. Load this instrumentation module and then build and run the program to get more in-depth GPU-specific performance data sent to `stdout`. This is equivalent to loading the `perftools-lite` module and setting the `CRAYPAT_LITE` environment variable to `gpu` in earlier releases.
`perftools-lite-loop`	CrayPat-lite loop estimates for Reveal. Load this instrumentation module and then build and run the program to get loop work estimates sent to `stdout`. This experiment is new in release 6.3.0 and automates the profile generation process used in earlier releases. The compile and link steps are automatically altered to include the `CCE -h profile_generate` option, and the program is automatically instrumented for tracing. After program execution completes, a performance data file (`.ap2`) is produced for use with Reveal. After the experiment is complete, unload the `perftools-lite-loops` module to prevent further program instrumentation.

NOTE: The `perftools-base` and `papi` modules are mutually exclusive. You can load one or the other, but not both at the same time.

When the `perftools-base` module and an instrumentation module are loaded, CrayPat, CrayPat-lite, Cray Apprentice2, and Reveal function as described in the associated man pages and in *Using Cray Performance Measurement and Analysis Tools* (S-2376). If the `perftools` instrumentation module is loaded, there are four essential commands:

pat_build Instrument your program for data collection.

pat_report After your program has completed execution, post-process the resulting data files for text reports and further analysis.

app2 Launch Cray Apprentice2, to conduct in-depth graphical analysis of the processed data files.

reveal Launch Reveal, to combine performance data captured during program execution with the annotated source code listing, in order to produce an integrated view of your code, how it is behaving during run time, and where the most productive opportunities for optimization can be found.

These commands and their options are discussed in the following man pages.

Table 3. Perftools Man Pages

Man Page	Description
`intro_craypat(1)`	A quick introduction to CrayPat usage and detailed information about runtime environment variables that affect the kind, quality, and quantity of information captured during program execution.
`craypat_lite(1)`	An easy-to-use set of CrayPat instrumentation modules.
`pat_build(1)`	Detailed information about preparing your programs for performance analysis experiments.
`perftools_base(4)`	CrayPat fundamental module, which describes the new module structure.
`pat_report(1)`	Detailed information about the reports that can be generated from performance analysis data after it has been captured.
`uncore(5)`	Intel Uncore performance counters.
`pat_help(1)`	A quick introduction to `pat_help`, an extensive help system and tutorial that includes many practical examples of Perftools usage as well as the answers to many frequently asked questions.
`papi_counters(5)`	Introduction to the PAPI event counters.
`app2(1)`	A quick introduction to the Cray Apprentice2 graphical data analysis tool.
`reveal(1)`	A quick introduction to the Reveal integrated performance analysis and code optimization tool.
`grid_order(1)`	A utility to generate MPI rank order placement lists.
`hwpc(5)`	Hardware (CPU) performance counter reference information.
`nwpc(5)`	Network performance counter reference information.
`accpc(5)`	Accelerator (GPU) performance counter reference information.

Man Page	Description
nbpc(5)	AMD Interlagos Northbridge (including L3 cache) event counter reference information.
cray_pm(5)	Intel Running Average Power Limit and Cray Power Management performance counter reference information.

The most common cause of confusion when getting started with the Cray Performance Measurement and Analysis Tools is losing track of where you are in the Cray system. For example, some commands can be run on either service or compute nodes, while others return valid results only when run on compute nodes. To further complicate matters, some commands can be run only on specific types of nodes, and then only if launched from a mount-point on a Lustre file system and launched using the correct utility.

If a Perftools command does not seem to return the expected results, always verify that you have the correct modules loaded for the system you are using and that you are running the command from the correct location in the file system.

Use the PAPI Cray NPU Component

To use the PAPI Cray NPU Component, unload the `perftools-base` module and load the `papi` module.

```
> module unload perftools-base
> module load papi
```

> **NOTE:** The `perftools-base` and `papi` modules are mutually exclusive. You can load one or the other, but not both at the same time.

The PAPI Cray NPU Component is intended for advanced users who want to write their own tools and access the PAPI CPU, GPU, and Intel Gemini™ or Intel Aries™ network performance counters directly. For more information about using the PAPI Cray NPU Component, see the technical note, *Using the PAPI Cray NPU Component* (S-0046), which can be downloaded from the Cray website.

When the `perftools-base` module is loaded, a complete list of the Gemini network performance counters available through PAPI can be found in the `counters->gemini` topics in `pat_help`.

Likewise, when the `perftools-base` module is loaded, the Aries network performance counters can be found in the `counters->aries` topics in `pat_help`. However, given that you must unload the `perftools-base` module before loading the `papi` module, when you are using the PAPI Cray NPU Component, the Cray-originated man pages, the `pat_help` system, and the `$CRAYPAT_ROOT` path are likely to be unavailable.

Therefore, it may be helpful to dump the contents of selected `pat_help` topics to a text file, so that the information is available after the `perftools-base` module is unloaded. For example, to dump the `pat_help counters->aries->native` topic to a text file, enter this command:

```
> pat_help counters aries native all . > aries_native_help
```

The contents of the resulting text file look like this:

```
> pat_help counters aries native AR_NIC_AMO_PRF_EN
   * Set to zero to configure default values.

> pat_help counters aries native AR_NIC_AMO_PRF_EN:PRF_FLUSH_EN
    * Enable AR_NIC_AMO_PRF_STALL_DURATION_FLUSH.
```

```
> pat_help counters aries native AR_NIC_AMO_PRF_EN:PRF_FULL_EN
    * Enable AR_NIC_AMO_PRF_STALL_DURATION_FULL.
```

To make the file more readable, you can filter out the prompts, like this:

```
> pat_help counters aries native all . \
  | sed 's/> pat_help counters aries native //' > aries_native_help
```

In which case the resulting text file would look like this:

```
AR_NIC_AMO_PRF_EN
 * Set to zero to configure default values.

AR_NIC_AMO_PRF_EN:PRF_FLUSH_EN
 * Enable AR_NIC_AMO_PRF_STALL_DURATION_FLUSH.

AR_NIC_AMO_PRF_EN:PRF_FULL_EN
 * Enable AR_NIC_AMO_PRF_STALL_DURATION_FULL.
```