Performance Co-Pilot™ Tutorials and Case Studies

Performance Co-Pilot™ Tutorials and Case Studies

Maintained by:
The Performance Co-Pilot Development Team

pcp@mail.performancecopilot.org

http://www.performancecopilot.org

About This Book

This book is a collection of tutorials, case studies, and short stories about various aspects of the Performance Co-Pilot (PCP) performance analysis toolkit. Topics range from basic installation through to detailed analysis techniques, with everything in-between, that even seasoned PCP users can expect to learn from.

Much like PCP is an open source, cross-platform software package - where customizations are actively encouraged - so too is this book. If you have interesting experiences to share, please consider contibuting - the XML source to this book is available, along with the PCP source code.

"About This Book" includes short descriptions of the chapters in this book, directs you to additional sources of information, and explains typographical conventions.

1. What This Book Contains

This book contains the following chapters:

- *Chapter 1, Supported Platforms Installation*, contains introductory topics around installation of PCP.

- *Chapter 2, Tutorials*, contains a series of hands-on tutorials covering use of specific PCP monitor tools, secure collector setup.

- *Chapter 3, Case Studies*, provides a set of case studies of use of PCP to address problems, or understand certain system behaviours using PCP.

It provides a series of real-world examples of using various PCP tools, and lessons learned from deploying the toolkit in production environments. It serves to provide reinforcement of the general concepts discussed in the other books described in the *Section 3, "Related Resources"*, with additional case studies and more detailed discussion of individual tools.

2. Audience for This Book

This book has something for everyone - early topics will aid those getting started with PCP, while later sections cover in-depth material, sometimes requiring detailed understanding of operating system internals.

Some familiarity with the basic concepts behind PCP is assumed.

3. Related Resources

The *Performance Co-Pilot User's and Administrator's Guide* is intended for system administrators and performance analysts who are directly using and administering PCP installations.

The *Performance Co-Pilot Programmer's Guide* is intended for developers who want to use the PCP framework and services for exporting additional collections of performance metrics, or for delivering new or customized applications to enhance performance management.

Additional resources include man pages and the project web site.

4. Man Pages

The operating system man pages provide concise reference information on the use of commands, subroutines, and system resources. There is usually a man page for each PCP command or subroutine. To see a list of all the PCP man pages, start from the following command:

```
man PCPIntro
```

Each man page usually has a "SEE ALSO" section, linking to other, related entries.

To see a particular man page, supply its name to the **man** command, for example:

```
man pcp
```

The man pages are arranged in different sections separating commands, programming interfaces, and so on. For a complete list of manual sections on a platform enter the command:

```
man man
```

When referring to man pages, this guide follows a standard convention: the section number in parentheses follows the item. For example, **pminfo(1)** refers to the man page in section 1 for the **pminfo** command.

5. Web Site

The following web site is accessible to everyone:

URL
> **Description**

http://www.performancecopilot.org
> PCP is open source software released under the GNU General Public License (GPL) and GNU Lesser General Public License (LGPL)

6. Conventions

The following conventions are used throughout this document:
Convention
> **Meaning**

${PCP_VARIABLE}
> A brace-enclosed all-capital-letters syntax indicates a variable that has been sourced from the global **/etc/pcp.conf** file. These special variables indicate parameters that affect all PCP commands, and are likely to be different between platforms.

command
> This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.

variable
> Italic typeface denotes variable entries and words or concepts being defined.

user input
> This bold, fixed-space font denotes literal items that the user enters in interactive sessions.
> (Output is shown in nonbold, fixed-space font.)

[ ]
> Brackets enclose optional portions of a command or directive line.

...
> Ellipses indicate that a preceding element can be repeated.

ALL CAPS
> All capital letters denote environment variables, operator names, directives, defined constants,
> and macros in C programs.

()
> Parentheses that follow function names surround function arguments or are empty if the function
> has no arguments; parentheses that follow commands surround man page section numbers.

7. Reader Comments

If you have comments about the technical accuracy, content, or organization of this document, contact the PCP maintainers using either the email address or the web site listed earlier.

We value your comments and will respond to them promptly.

Supported Platforms Installation

Binary packages for Linux (deb and rpm formats), Mac OS X, Solaris and Windows are made available by the PCP development team.

These are all freely available from the download section of the PCP project web site.

1.1. Linux Installation

Pre-packaged binaries are available for the Linux platform ... XXX

1.2. Mac OS X Installation

After downloading the **dmg** file for your platform, double-click on the PCP **dmg** icon in the Finder application, and follow the installation instructions presented by the Installer.

Mac OS X versions 10.5 and onward are generally available, although older versions can be built from source.

1.3. Solaris Installation

Download and unzip the latest binary package from the PCP project web site.

To install the package for the first time, issue the following commands:

```
pkgadd -d pcp-${PCP_VERSION}
  Say 'y' to all of the questions.
svcadm enable pmcd
```

To upgrade an existing installation, issue the following commands:

```
pkgadd -d pcp-${PCP_VERSION}
pkgadd -d pcp-${PCP_VERSION}
svcadm enable pmcd
```

1.4. Windows Installation

After downloading the PCP Glider **msi** file, right-click on the command prompt icon, select **Run As Administrator**, and enter:

```
msiexec /i pcp-glider-${PCP_VERSION}.msi
cd C:\Glider\scripts
postinst.bat
```

The command line utilities can now be accessed from a Windows shell or the provided (POSIX) shell. The graphical tools can be accessed via the Windows **Start** menu.

Tutorials

From a high-level PCP can be considered to contain two classes of software utility:

- PCP Collectors. These are the parts of PCP that collect and extract performance data from various domains, such as the kernel or a database.

- PCP Monitors. These are the parts of PCP that display data collected from hosts (or archives) that have the PCP Collector installed.

The tutorials of this section focus on various monitoring tools initially, and then finish up with some extensions to collector systems not covered in the *Performance Co-Pilot User's and Administrator's Guide*.

2.1. Using PCP Charts

...

2.2. Logging Basics

...

2.3. Automated Reasoning

...

Case Studies

3.1. Understanding System-Level Processor Performance

...

3.2. Understanding Measures of Disk Performance

...

3.3. An Operating System Upgrade Evaluation

...

3.4. Comparing Storage System Performance

...

3.5. Central Logging: Instrumenting Distributed Event Logs

...

Index