



# **NightTune User's Guide**

**Version 3.2**

**(RedHawk™ Linux®)**

Copyright 2008 by Concurrent Computer Corporation. All rights reserved. This publication or any part thereof is intended for use with Concurrent products by Concurrent personnel, customers, and end-users. It may not be reproduced in any form without the written permission of the publisher.

The information contained in this document is believed to be correct at the time of publication. It is subject to change without notice. Concurrent Computer Corporation makes no warranties, expressed or implied, concerning the information contained in this document.

To report an error or comment on a specific portion of the manual, photocopy the page in question and mark the correction or comment on the copy. Mail the copy (and any additional comments) to Concurrent Computer Corporation, 2881 Gateway Drive, Pompano Beach, FL 33069-4324. Mark the envelope “**Attention: Publications Department.**” This publication may not be reproduced for any other reason in any form without written permission of the publisher.

Concurrent Computer Corporation and its logo are registered trademarks of Concurrent Computer Corporation. All other Concurrent product names are trademarks of Concurrent while all other product names are trademarks or registered trademarks of their respective owners.

Linux<sup>®</sup> is used pursuant to a sublicense from the Linux Mark Institute.

NightStar’s integrated help system is based on Qt’s Assistant from Trolltech.

## Scope of Manual

This guide is designed to assist you in getting started with use of NightTune™, a process and system analysis and tuning tool.

## Structure of Manual

This manual consists of four chapters, two appendices and an index. A brief description of the contents of each of the parts of the manual follows.

- Chapter 1 introduces NightTune, its command line options, and system requirements.
- Chapter 2 describes NightTune display windows.
- Chapter 3 describes each of NightTune's functional display panels.
- Chapter 4 describes how to operate NightTune to execute specific monitoring and tuning tasks.
- Appendix A describes how to install and configure licenses.
- Appendix B describes dependencies of certain features on the RedHawk kernel.

## Syntax Notation

The following notation is used throughout this guide:

### *italic*

Books, reference cards, and items that the user must specify appear in *italic* type. Special terms and comments in code may also appear in *italic*.

### **list bold**

User input appears in **list bold** type and must be entered exactly as shown. Names of directories, files, commands, options and man page references also appear in **list bold** type.

### list

Operating system and program output such as prompts and messages and listings of files and programs appears in `list` type. Keywords also appear in `list` type.

## window

Keyboard sequences and window features such as push buttons, radio buttons, menu items, labels, and titles appear in *window* type.

[ ]

Brackets enclose command options and arguments that are optional. You do not type the brackets if you choose to specify such option or arguments.

{ }

Braces enclose mutually exclusive choices separated by the pipe (|) character, where one choice must be selected. You do not type the braces or the pipe character with the choice.

...

An ellipsis follows an item that can be repeated.

## Referenced Publications

The following publications are referenced in this document:

0898004	<i>RedHawk Linux User's Guide</i>
0898008	<i>NightStar RT Installation Guide</i>
0898009	<i>NightStar RT Tutorial</i>
0898395	<i>NightView™ RT User's Guide</i>
0898398	<i>NightTrace™ RT User's Guide</i>
0898458	<i>NightSim™ RT User's Guide</i>
0898465	<i>NightProbe™ RT User's Guide</i>

# Contents

## Chapter 1 Introducing NightTune

Point and Click Operation . . . . .	1-2
Local or Remote Operation . . . . .	1-2
Capabilities . . . . .	1-2
Command Line Options. . . . .	1-4
IRQ Balancing. . . . .	1-5

## Chapter 2 NightTune Windows

Menu Bar . . . . .	2-1
File . . . . .	2-2
View . . . . .	2-4
Monitor. . . . .	2-6
Tools . . . . .	2-9
Help . . . . .	2-10
Toolbar . . . . .	2-12
Pages . . . . .	2-14
Preferences. . . . .	2-16

## Chapter 3 NightTune Panels

System Status Panel. . . . .	3-1
System Status Context Menu . . . . .	3-2
Context Switches Panel . . . . .	3-4
Context Switches Text Pane . . . . .	3-4
Context Switches Bar Graph Pane . . . . .	3-4
Context Switches Line Graph Pane . . . . .	3-5
CPU Shielding and Binding Panel. . . . .	3-7
CPU Shielding Operations . . . . .	3-8
CPU Shielding and Binding Drag and Drop Operations . . . . .	3-10
CPU Shielding and Binding Context Menu . . . . .	3-11
CPU Usage Panel . . . . .	3-13
CPU Usage Text Pane . . . . .	3-13
CPU Usage Bar Graph Pane. . . . .	3-14
CPU Usage Line Graph Pane . . . . .	3-15
Disk Activity Panel . . . . .	3-16
Disk Activity Text Pane . . . . .	3-16
Disk Activity Bar Graph Pane . . . . .	3-17
Disk Activity Line Graph Pane . . . . .	3-18
Interrupt Activity Panel . . . . .	3-19
Interrupt Activity Text Pane . . . . .	3-19
Interrupt Control Drag and Drop Operations . . . . .	3-20
Interrupt Activity Bar Graph Pane . . . . .	3-21
Interrupt Activity Line Graph Pane . . . . .	3-22
Interrupt Activity Context Menu . . . . .	3-23

Interrupt Affinity Dialog . . . . .	3-24
Memory Activity Panel . . . . .	3-25
Memory Activity Text Pane . . . . .	3-26
Memory Activity Bar Graph Pane . . . . .	3-27
Memory Activity Line Graph Pane . . . . .	3-27
Kernel Virtual Memory Panel . . . . .	3-28
Kernel Virtual Memory Text Pane . . . . .	3-28
Kernel Virtual Memory Bar Graph Pane . . . . .	3-29
Kernel Virtual Memory Line Graph Pane. . . . .	3-29
Physical Memory Panel . . . . .	3-30
Physical Memory Text Pane . . . . .	3-30
Subdivision by Usage . . . . .	3-31
Subdivision by Recency . . . . .	3-31
Subdivision by Low vs. High Memory . . . . .	3-32
Physical Memory Bar Graph Pane . . . . .	3-32
Physical Memory Line Graph Pane . . . . .	3-33
Swap Panel. . . . .	3-34
Swap Text Pane. . . . .	3-34
Swap Bar Graph Pane . . . . .	3-35
Swap Line Graph Pane . . . . .	3-35
Network Activity Panel . . . . .	3-36
Network Activity Text Pane . . . . .	3-36
Network Activity Bar Graph Pane . . . . .	3-37
Network Activity Line Graphs . . . . .	3-38
NUMA Panel . . . . .	3-39
NUMA Text Pane . . . . .	3-39
Subdivision by Usage . . . . .	3-39
Subdivision by Recency . . . . .	3-40
Subdivision by Low vs. High Memory . . . . .	3-40
NUMA Bar Graph Pane . . . . .	3-41
NUMA Line Graph Pane . . . . .	3-42
NUMA Activity Panel . . . . .	3-42
NUMA Activity Text Pane . . . . .	3-43
NUMA Activity Bar Graph Pane . . . . .	3-44
NUMA Activity Line Graph Pane . . . . .	3-44
NUMA Configuration Panel . . . . .	3-45
NUMA Configuration CPUs Pane . . . . .	3-45
NUMA Configuration Distance Pane . . . . .	3-45
NUMA Configuration Context Menu. . . . .	3-46
Process List Panel . . . . .	3-46
Process List Drag and Drop Operations . . . . .	3-48
Process List Context Menu . . . . .	3-49
Filter Processes Dialog . . . . .	3-52
Trace System Calls Window . . . . .	3-54
Trace Library Calls Window . . . . .	3-54
Debug Process . . . . .	3-55
Process Scheduler Dialog . . . . .	3-55
Process Scheduling Operations . . . . .	3-58
Process Details Window . . . . .	3-59
Memory Usage Tab . . . . .	3-59
Subdivision by Usage . . . . .	3-60
Subdivision by Shared / Nonshared . . . . .	3-61
Subdivision by Residency . . . . .	3-61
Subdivision by NUMA node. . . . .	3-62

Memory Tab . . . . .	3-62
File Descriptors Tab . . . . .	3-66
Signals Tab . . . . .	3-68
Capabilities Tab . . . . .	3-70
Environment Tab . . . . .	3-72
Process Fields Menu . . . . .	3-73
Systems Menu . . . . .	3-77

**Chapter 4 Guide to Operations**

Monitoring User Processes . . . . .	4-2
Selecting the User Process . . . . .	4-2
Customizing the Process Information . . . . .	4-4
Changing User Process Scheduling Attributes . . . . .	4-6
Using Drag and Drop to Change Process CPU Affinity . . . . .	4-7
Shielding a CPU . . . . .	4-10
Changing the CPU Affinity of an Interrupt . . . . .	4-14
Using Drag and Drop to Change Interrupt CPU Affinity . . . . .	4-17

**Appendix A NightStar Licensing**

License Keys . . . . .	A-1
License Requests . . . . .	A-2
License Server . . . . .	A-2
License Reports . . . . .	A-3
Firewall Configuration for Floating Licenses . . . . .	A-3
License Support . . . . .	A-4

**Appendix B Kernel Dependencies**

Advantages for NightView . . . . .	B-1
Advantages for NightTrace . . . . .	B-1
Advantages for NightProbe . . . . .	B-2
Advantages for NightTune . . . . .	B-2
Frequency Based Scheduler . . . . .	B-3
PCI Bar File System . . . . .	B-3

**Index**

**Illustrations**

Figure 2-1. File Menu . . . . .	2-2
Figure 2-2. View Menu . . . . .	2-4
Figure 2-3. Monitor Menu . . . . .	2-6
Figure 2-4. Tools Menu . . . . .	2-9
Figure 2-5. Help Menu . . . . .	2-10
Figure 2-6. NightTune Main Tool Bar . . . . .	2-12
Figure 2-7. Tab Context Menu . . . . .	2-14
Figure 2-8. Rename Page Dialog . . . . .	2-14
Figure 2-9. Move Page Dialog . . . . .	2-15
Figure 2-10. Preferences Dialog . . . . .	2-16

Figure 3-1. Typical Context Menu . . . . .	3-2
Figure 3-2. Context Switches Text Pane . . . . .	3-4
Figure 3-3. Context Switches Bar Graph Pane . . . . .	3-5
Figure 3-4. Context Switches Line Graph Pane . . . . .	3-6
Figure 3-5. CPU Shielding and Binding Panel . . . . .	3-7
Figure 3-6. CPU Shielding Dialog . . . . .	3-8
Figure 3-7. CPU Shielding and Binding Context Menu . . . . .	3-11
Figure 3-8. CPU Usage Text Pane . . . . .	3-13
Figure 3-9. CPU Usage Bar Graph Pane . . . . .	3-14
Figure 3-10. CPU Usage Line Graph Pane . . . . .	3-15
Figure 3-11. Disk Activity Text Pane . . . . .	3-16
Figure 3-12. Disk Activity Bar Graph Pane . . . . .	3-17
Figure 3-13. Disk Activity Line Graph Pane . . . . .	3-18
Figure 3-14. Interrupt Activity Text Pane . . . . .	3-19
Figure 3-15. Interrupt Activity Bar Graph Pane . . . . .	3-21
Figure 3-16. Interrupt Activity Line Graph Pane . . . . .	3-22
Figure 3-17. Interrupt Activity Context Menu . . . . .	3-23
Figure 3-18. Interrupt Affinity Dialog . . . . .	3-24
Figure 3-19. Memory Activity Text Pane . . . . .	3-26
Figure 3-20. Memory Activity Bar Graph Pane . . . . .	3-27
Figure 3-21. Memory Activity Line Graph Pane . . . . .	3-27
Figure 3-22. Kernel Virtual Memory Text Pane . . . . .	3-28
Figure 3-23. Kernel Virtual Memory Bar Graph Pane . . . . .	3-29
Figure 3-24. Kernel Virtual Memory Line Graph Pane . . . . .	3-29
Figure 3-25. Physical Memory Text Pane . . . . .	3-30
Figure 3-26. Physical Memory Bar Graph Pane . . . . .	3-32
Figure 3-27. Physical Memory Line Graph Pane . . . . .	3-33
Figure 3-28. Swap Memory Text Pane . . . . .	3-34
Figure 3-29. Swap Memory Bar Graph Pane . . . . .	3-35
Figure 3-30. Swap Memory Line Graph Pane . . . . .	3-35
Figure 3-31. Network Activity Text Pane . . . . .	3-36
Figure 3-32. Network Activity Bar Graph Pane . . . . .	3-37
Figure 3-33. Network Activity Line Graph Pane . . . . .	3-38
Figure 3-34. NUMA Text Pane . . . . .	3-39
Figure 3-35. NUMA Bar Graph Pane . . . . .	3-41
Figure 3-36. NUMA Line Graph Pane . . . . .	3-42
Figure 3-37. NUMA Activity Text Pane . . . . .	3-43
Figure 3-38. NUMA Activity Bar Graph Pane . . . . .	3-44
Figure 3-39. NUMA Activity Line Graph Pane . . . . .	3-44
Figure 3-40. NUMA Configuration Text Pane . . . . .	3-45
Figure 3-41. NUMA Configuration Distance Pane . . . . .	3-45
Figure 3-42. NUMA Configuration Context Menu . . . . .	3-46
Figure 3-43. Process List Panel . . . . .	3-47
Figure 3-44. Process List Filter Bar . . . . .	3-48
Figure 3-45. Process List Find Bar . . . . .	3-48
Figure 3-46. Process List Context Menu . . . . .	3-49
Figure 3-47. Filter Processes Dialog . . . . .	3-53
Figure 3-48. Process List Trace System Calls Window . . . . .	3-54
Figure 3-49. Trace Library Calls Window . . . . .	3-55
Figure 3-50. Process Scheduler Dialog . . . . .	3-56
Figure 3-51. Process Details Memory Usage Tab . . . . .	3-60
Figure 3-52. Process Details Memory Tab . . . . .	3-62
Figure 3-53. Process Details File Descriptors Tab . . . . .	3-67
Figure 3-54. Process Details Signals Tab . . . . .	3-69



Figure 3-55. Process Details Capabilities Tab . . . . .	3-71
Figure 3-56. Process Details Environment Tab . . . . .	3-72
Figure 3-57. Process Fields Menu . . . . .	3-73
Figure 3-58. Systems Menu . . . . .	3-77
Figure 4-1. Monitoring User Processes . . . . .	4-2
Figure 4-2. Selecting the User Process . . . . .	4-3
Figure 4-3. Monitoring Multi-threaded Processes . . . . .	4-4
Figure 4-4. Customizing the Process Information . . . . .	4-5
Figure 4-5. Process Scheduler Dialog . . . . .	4-6
Figure 4-6. Changing User Process Scheduling Attributes . . . . .	4-7
Figure 4-7. Viewing CPU Status . . . . .	4-8
Figure 4-8. Bound Processes . . . . .	4-9
Figure 4-9. Using Drag and Drop to Change Process CPU Affinity . . . . .	4-10
Figure 4-10. CPU Shielding and Binding Panel . . . . .	4-11
Figure 4-11. CPU Shielding Dialog . . . . .	4-12
Figure 4-12. Shielding a CPU . . . . .	4-13
Figure 4-13. Error Shielding CPU . . . . .	4-13
Figure 4-14. Monitoring Interrupt Activity . . . . .	4-15
Figure 4-15. Interrupt Affinity Dialog . . . . .	4-16
Figure 4-16. Changing the CPU Affinity of an Interrupt . . . . .	4-17
Figure 4-17. Using Drag and Drop to Change Interrupt CPU Affinity . . . . .	4-18



# Introducing NightTune

---

NightTune's graphical user interface (GUI) provides a powerful and intuitive point-and-click style of operation that allows you to analyze and adjust system activities with ease.

NightTune has the following features:

- Process monitoring and tuning

Through the **Process List** panel you can monitor their CPU time, memory size, scheduling parameters, CPU affinity and other attributes of individual processes. You can monitor the same attributes of individual threads within a process. You can modify the scheduling parameters and CPU affinity with the **Process Scheduler** dialog or with drag-and-drop operations to the **CPU Shielding and Binding** panel. Finally, you can view their memory usage, file descriptors, signals, capabilities, and environment variables.

- CPU control

You can monitor the status of a **CPU**, including information about shielding and process and interrupt bindings. If you have the appropriate privileges, you can set shielding and hyper-threading attributes for each CPU.

- System monitoring and tuning

With the system activity panels, you can monitor:

- Context Switches
- CPU Usage
- CPU Shielding and Binding
- Disk Activity
- Interrupt Activity
- Memory Activity
- Kernel Memory
- Physical Memory
- Swap Space
- Network Activity
- NUMA Memory
- NUMA Memory Activity
- NUMA Memory Configuration

Information is available both numerically and graphically. The Interrupt Activity panel allows you to change the CPU affinity for individual interrupts.

- Tuning Activity log

This facility provides optional logging of tuning activity to a user-specified log file.

- Comprehensive online help

The help system includes context-sensitive help, accessible by clicking on any item in NightTune's window.

NightTune uses the NightStar License Manager (NSLM) to control access to the NightStar RT tools. See "NightStar Licensing" on page A-1 for more information.

## Point and Click Operation

NightTune provides a point-and-click interface. Most operations utilize the mouse.

NightTune makes extensive use of right-click context menus and drag-and-drop actions.

## Local or Remote Operation

NightTune can operate on local or remote systems. When operating remotely, the graphical user interface runs on the host system where NightTune is invoked and communicates to a NightTune server process which is launched automatically on the remote target system.

The Pluggable Authentication Module (PAM) is used to authenticate connection requests to remote systems (see **pam(8)**). The file `/etc/pam.d/ntune` is installed as part of the NightTune product; it defines the specific authentication mechanisms that are used for each target system. For security reasons, NightTune encrypts usernames and passwords during all authentication requests and never stores authentication information to disk.

See "Command Line Options" on page 1-4 for more information on remote operation of NightTune.

## Capabilities

Most operations with NightTune do not require any special privileges. However, if you wish to modify the scheduling attributes, view detailed memory usage of other users' processes, or adjust CPU shielding attributes, you must have the `CAP_SYS_NICE` capability or invoke NightTune as the `root` user.

Linux provides a means to grant otherwise unprivileged users the authority to perform certain privileged operations. The Pluggable Authentication Module (see `pam_capability(8)`) is used to manage sets of capabilities, called *roles*, required for various activities.

Linux systems should be configured with an `ntuneuser` role which provides the `CAP_SYS_NICE` capability.

Edit `/etc/security/capability.conf` and define the `ntuneuser` role (if it is not already defined) in the “ROLES” section:

```
role ntuneuser CAP_SYS_NICE
```

Additionally, for each NightTune user on the target system, add the following line at the end of the file:

```
user username ntuneuser
```

where *username* is the login name of the user.

If the user requires capabilities not defined in the `ntuneuser` role, add a new role which contains `ntuneuser` and the additional capabilities needed, and substitute the new role name for `ntuneuser` in the text above.

In addition to registering your login name in `/etc/security/capability.conf`, certain files under the `/etc/pam.d` directory must also be configured to allow capabilities to be activated.

To activate capabilities, add the following line to the end of selected files in `/etc/pam.d` if it is not already present:

```
session required pam_capability.so
```

The list of files to modify is dependent on the list of methods that will be used to access the system. The following table presents a recommended configuration that will grant capabilities to users of the services most commonly employed in accessing a system.

**Table 1-1. Recommended /etc/pam.d Configuration**

<code>/etc/pam.d</code> File	Affected Services	Comment
<code>remote</code>	telnet rlogin rsh (when used <u>w/o</u> a command)	Depending on your system, the <code>remote</code> file may not exist. Do not create the <code>remote</code> file, but edit it if it is present.
<code>login</code>	local login (e.g. console) telnet* rlogin* rsh* (when used <u>w/o</u> a command)	*On some versions of Linux, the presence of the <code>remote</code> file limits the scope of the <code>login</code> file to local logins. In such cases, the other services listed here with <code>login</code> are then affected solely by the <code>remote</code> configuration file.
<code>rsh</code>	rsh (when used <u>with</u> a command)	e.g. <code>rsh system_name a.out</code>

**Table 1-1. Recommended /etc/pam.d Configuration**

/etc/pam.d File	Affected Services	Comment
<b>sshd</b>	ssh	You must also edit <code>/etc/ssh/sshd_config</code> and ensure that the following line is present: <b>UsePrivilegeSeparation no</b>
<b>gdm</b>	gnome sessions	
<b>kde</b>	kde sessions	

If you modify `/etc/pam.d/sshd` or `/etc/ssh/sshd_config`, you must restart the **sshd** service for the changes to take effect:

```
service sshd restart
```

In order for the above changes to take effect, the user must log off and log back onto the target system.

#### **NOTE**

To verify that you have been granted capabilities, issue the following command:

```
/usr/sbin/getpcaps $$
```

The output from that command will list the roles currently assigned to you.

## **Command Line Options**

Use the following command to start NightTune:

```
ntune [--help]  

  [--config=config-file | -c config-file]  

  [--logfile=logfile | -l logfile]  

  [--target=system-name | -t system-name]
```

Options are described as follows:

**--help**

Causes NightTune to display its command line syntax (followed by a brief description of each option), then exit.

**--config=config-file**  
**-c config-file**

Specifies the configuration file that contains the start-up information for NightTune. In the absence of the `--config` option, NightTune searches for a configuration file using the following ordered criteria:

- a `.NightTunerc` file in the current working directory
- a `.NightTunerc` file in the user's home directory
- `/usr/lib/NightTune/lib/config`

`--logfile=logfile`  
`-l logfile`

Specifies the file to be used for logging tuning activity. If logging is enabled via the `Logging` option from the `NightTune` menu, tuning events are logged to this file. The name of the file can be entered on the command line or specified in the configuration file via the GUI; a log file named on the command line overrides one specified in the configuration file.

`--target=system-name`  
`-t system-name`

Causes NightTune to operate on the specified remote target system. The NightTune GUI will run on the system where `ntune` was invoked and will communicate with a NightTune server process, which is automatically launched on the target system.

NightTune connects to the target system using the `nstar.d` daemon, supplied as part of NightStar RT tools, running on that target system.

## IRQ Balancing

By default, most Linux distributions enable a system service called `irqbalance`, which automatically sets the affinity of interrupts to balance them across CPUs. This will interfere with user-selected interrupt affinities. To disable this service both immediately and on subsequent reboots, use these commands:

```
service irqbalance stop
chkconfig irqbalance off
```





## NightTune Windows

---

NightTune features are available through a number of functional panels that are placed in tabbed pages in NightTune windows.

Much of the time you will operate NightTune with a single window displaying several panels of information within one or more tabbed pages. However, NightTune allows you to have multiple windows active at once, customized to display any panels of interest.

Each window has:

- a menu bar (see “Menu Bar” on page 2-1)
- a toolbar that provides shortcuts and affects how data is refreshed (see “Toolbar” on page 2-12)
- a display and control area, which is the main body of the window where tabbed pages display selected panels (see “NightTune Panels” on page 3-1)

### Menu Bar

The menu bar provides access to controlling which panels are displayed in the window as well as the appearance of items inside panels, preference settings, launching additional tools, and obtaining help.

The menu bar provides the following menus:

- File
- View
- Monitor
- Tools
- Help

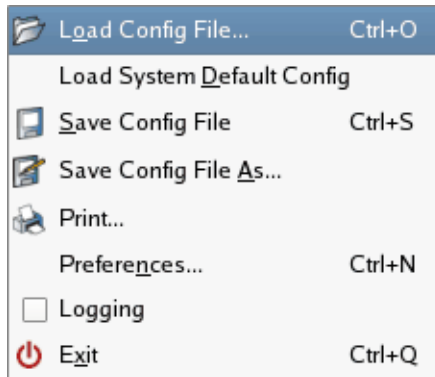
Each menu is described in the sections that follow.

An icon appearing at the left of a menu item indicates that the corresponding icon on the toolbar can also be used to perform that function.

## File

Mnemonic: Alt+F

The **File** menu allows you to set preferences, load or save configuration data, activate or deactivate logging, and exit NightTune.



**Figure 2-1. File Menu**

The following paragraphs describe the options on the **File** menu in more detail.

### Load Config File...

Mnemonic: O  
Accelerator: Ctrl+O

This menu item displays a file selection dialog. By selecting a previously-saved configuration file, NightTune closes all current windows, tabbed pages, and panels and opens windows, tabbed pages, and panels as defined in the configuration file.

### Load System Default Config

Mnemonic: D

This menu item loads the system default configuration, reverting the configuration of windows, tabs, panels and panes back to their original configuration before any user modifications were made.

### Save Config File

Mnemonic: S  
Accelerator: Ctrl+S

This menu item allows you to save the configuration from the current session to a configuration file. Configuration data includes the layout of panels in tabbed pages for windows in the current session as well as settings in the **Preferences** dialog (see “Preferences” on page 2-16).

When invoked without the `--config` or `-c` options, NightTune automatically searches for a configuration file. It first looks for `.NightTunerc` in the current working directory and then in your home directory, and finally in `/usr/lib/NightTune/lib/config`. The first file found is automatically loaded when NightTune launches.

The configuration will be saved to whatever configuration file was saved or loaded last. If no configuration file was ever saved or loaded during the current NightTune session, then the configuration file name will be `.NightTunerc` and will be placed in your home directory.

For configurations involving remote systems, the names of those systems can be saved in the configuration, if so desired. See “Bound to System” on page 2-18 for more information.

### Save Config File As...

Mnemonic: A

This menu item allows you to save the configuration from the current session to a configuration file with a name of your choice.

When you select this menu item, NightTune displays a file selection dialog. After making a selection, the configuration data from the current session is saved in the selected file. Configuration data includes the layout of panels in tabbed pages for windows in the current session as well as settings in the **Preferences** dialog (see “Preferences” on page 2-16).

### Print

This menu item displays the Printer dialog which allows you to print the NightTune window.

### Preferences...

Mnemonic: N

Accelerator: **Ctrl+N**

This menu item displays the **Preferences** dialog which allows you to customize logging attributes and refresh intervals for various panels.

See “Preferences” on page 2-16 for more information.

### Logging

This menu item toggles the state of logging.

When logging is enabled, all modifications to processes scheduling attributes, CPU shielding settings, and interrupt CPU affinity are logged in ASCII text to a log file.

By default, the name of the log file is `ntune.log`, placed in the current working directory from which NightTune was launched. The name of the log file can be changed with the **Preferences** dialog (see “Preferences” on page 2-16).

## Exit

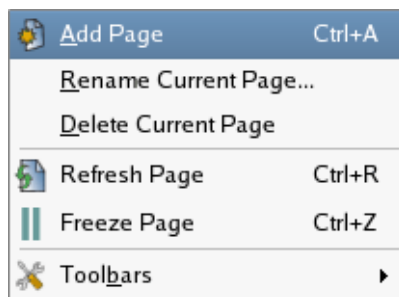
Mnemonic: X  
Accelerator: Ctrl+Q

This menu item exits NightTune. NightTune will not prompt you to save any unsaved configuration data before exiting – select **Save Config File** prior to exiting to retain configuration data for subsequent invocations of NightTune.

## View

Mnemonic: Alt+V

The **View** menu allows you to control tabbed pages and toolbars in the current NightTune session.



**Figure 2-2. View Menu**

The following paragraphs describe the options on the View menu in more detail.

### Add Page

Mnemonic: A  
Accelerator: Ctrl+A

This menu item creates a new tabbed page in the current window.

A dialog can be displayed by right-clicking on the tab that allows you to rename, delete or move the new page.

### Rename Current Page...

Mnemonic: R

This menu item opens a dialog which allows you to change the name associated with the tabbed page currently displayed in the window.

### Delete Current Page

Mnemonic: D

This menu item deletes the tabbed page currently displayed in the window.

### Refresh Page

Accelerator: Ctrl+R

The **Refresh** menu item immediately refreshes the displayed data in all panels in the current tabbed page.

### Freeze Page

Mnemonic: Z

Accelerator: Ctrl+Z

This menu item toggles the **Freeze** setting for all panels in the current tabbed page. When frozen, data values are not refreshed automatically; however, you can manually refresh the data on an otherwise frozen page by using the **Refresh Page** menu item.

### Toolbars

Mnemonic: B

This menu item opens a sub-menu to control which toolbars are visible. If an item in this menu is checked, its corresponding toolbar will be visible; otherwise it will not. The toolbars are:

- File Toolbar
- View Toolbar
- Monitor Toolbar
- Connect Toolbar
- Drop Site

The first four contain icons for functions from their corresponding menus. The **Drop Site** toolbar contains two icons used as targets for drag and drop operations (see “CPU Shielding and Binding Drag and Drop Operations” on page 3-10 and “Process List Drag and Drop Operations” on page 3-48).

In addition, the sub-menu contains the **Show All Toolbars** and **Hide All Toolbars** menu items, which check all toolbars and unchecked them, respectively.

## Monitor

Mnemonic: Alt+M

The Monitor menu allows you to select which panels are to be displayed in the tabbed page currently visible in the NightTune display area.

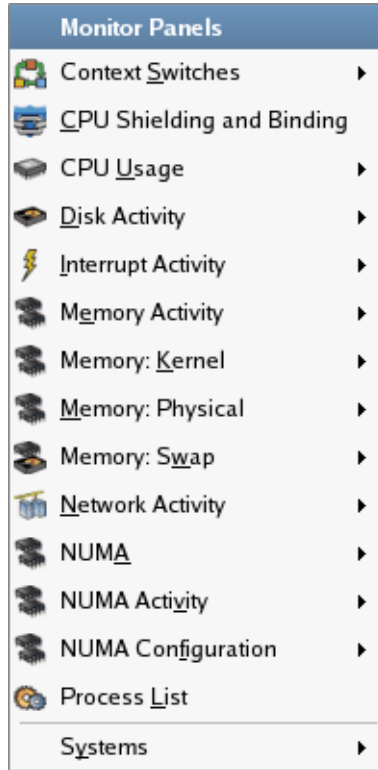


Figure 2-3. Monitor Menu

### NOTE

The Monitor menu, like the other menus, can be “torn off” from the menu bar to reside in its own window separate from NightTune by clicking on the dashed line at the top of the menu. This is especially useful when making multiple changes to the panels to avoid having to re-select the menu after every choice.

Menu items with an arrow at the right display a sub-menu for controlling which panes for that panel are displayed in the NightTune display area. For example, many panels have a Text pane, a Bar Graph pane and a Line Graph pane. Also on such sub-menus, an additional choice will exist to display all of those panes.

The following paragraphs summarize individual panel activities. Detailed descriptions for each panel are provided in “NightTune Panels” on page 3-1.

## **Context Switches**

Mnemonic: S

This panel displays context switches per CPU per second using textual and graphical displays.

## **CPU Shielding and Binding**

Mnemonic: C

This panel describes each CPU in the system and includes information on the state of shielding, process binding, and interrupt binding. It allows you to change the shielding attributes of CPUs and to redefine the CPU binding of specific processes or interrupts.

## **CPU Usage**

Mnemonic: U

This panel displays CPU utilization using textual and graphical displays. It provides User, System, Wait, and Idle time information per CPU.

## **Disk Activity**

Mnemonic: D

This panel provides detailed disk activity for all disks in the system. It includes information about read and write operations and service times using textual and graphical displays.

## **Interrupt Activity**

Mnemonic: I

This panel displays interrupt information for the system using textual and graphical displays. It allows you to change the CPU affinity of individual interrupts.

## **Memory Activity**

Mnemonic: E

This panel displays physical memory page transfer rates on the system using textual and graphical displays.

## **Memory: Kernel**

Mnemonic: V

This panel displays allocation of kernel virtual memory (vmalloc) on the system using textual and graphical displays.

### **Memory: Physical**

Mnemonic: M

This panel displays allocation of physical memory on the system using textual and graphical displays.

### **Memory: Swap**

Mnemonic: W

This panel displays allocation of swap space on the system using textual and graphical displays.

### **Network Activity**

Mnemonic: N

This panel describes network activity in terms of packet I/O rates, errors, and collisions for each network device on the system using textual and graphical displays.

### **NUMA**

Mnemonic: A

This panel describes physical memory allocation for each node on NUMA systems using textual and graphical displays.

### **NUMA Activity**

Mnemonic: V

This panel describes physical memory activity for each node on NUMA systems using textual and graphical displays.

### **NUMA Configuration**

Mnemonic: F

This panel describes the configuration of the NUMA systems, including the CPUs connected directly to each node and the relative distances from CPUs to memory on each node.

### **Process List**

Mnemonic: L

This panel provides information about processes. Additional detailed information is available for individual processes via a dialog window. The ability to change scheduling attributes for processes is available via another dialog window.



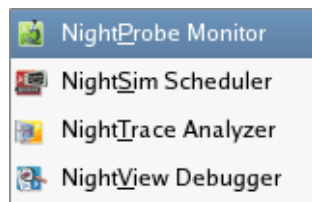
## Systems

Mnemonic: Y

This menu item allows the user to display NightTune panels for other systems and allows connection and disconnection to/from those systems.

## Tools

Mnemonic: Alt+T



**Figure 2-4. Tools Menu**

The following describe the options on the TOOLS menu:

### NightProbe Monitor

Mnemonic: P

Opens the NightProbe Data Monitoring tool. NightProbe is a real-time graphical tool for monitoring, recording, and altering program data within one or more executing programs without significant intrusion. NightProbe can be used in a development environment as a tool for debugging or in a production environment for data capture or to create a “control panel” for program input and output.

See also:

- *NightProbe RT User’s Guide*

### NightSim Scheduler

Mnemonic: S

Opens the NightSim Application Scheduler. NightSim is a tool for scheduling and monitoring real-time applications which require predictable, repetitive process execution. With NightSim, application builders can control and dynamically adjust the periodic execution of multiple coordinated processes, their priorities, and their CPU assignments.

See also:

- *NightSim RT User’s Guide*

## NightTrace Analyzer

Mnemonic: T

Opens the NightTrace Analyzer. The NightTrace Analyzer is a graphical tool for analyzing the dynamic behavior of multi-process and/or multi-processor user applications and operating system activity. NightTrace allows you to control user and kernel trace collection daemons and can graphically display the interplay between many real-time programs and processes across multiple processors and systems.

See also:

- *NightTrace RT User's Guide*

## NightView Debugger

Mnemonic: V

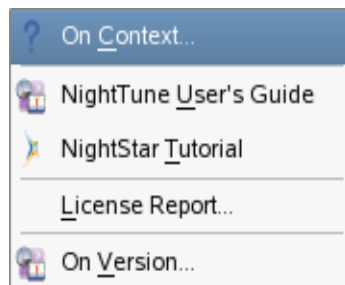
Opens the NightView Source-Level Debugger. NightView is a graphical source-level debugging and monitoring tool specifically designed for real-time applications. NightView can monitor, debug, and patch multiple real-time processes running on multiple processors with minimal intrusion.

See also:

- *NightView RT User's Guide*

## Help

Mnemonic: Alt+H



**Figure 2-5. Help Menu**

The following describe the options on the Help menu:

### On Context...

Mnemonic: C

Gives context-sensitive help on the various menu items, dialogs, or other parts of the user interface.

Help for a particular item is obtained by first choosing this menu item, then clicking the mouse pointer on the object for which help is desired. The mouse pointer becomes a floating question mark when the **On Context** menu item is selected.

In addition, context-sensitive help may be obtained for the widget with the current focus by pressing the **F1** key. NightTune's online help system will open with the appropriate topic displayed.

### **NightTune User's Guide**

Mnemonic: **G**

Opens the online version of the *NightTune RT User's Guide* in the online help viewer.

### **NightStar RT Tutorial**

Mnemonic: **T**

Opens the online version of the *NightStar RT Tutorial* in the online help viewer.

### **License Report...**

Mnemonic: **L**

Provides licensing information.

### **On Version**

Mnemonic: **V**

Displays a short description of the current version of NightTune.

### **Check for Updates...**

Mnemonic: **U**


Opens NUU, the Concurrent Network Update Utility. This tool checks online repositories for updates to NightStar and other tools.











## Toolbar

The NightTune tool bar provides icons for commonly used actions.



**Figure 2-6. NightTune Main Tool Bar**

	<b>Load Config File</b> Loads the config file
	<b>Save Config File</b> Saves the config file
	<b>Print</b> Displays the Print dialog that allows you to print the NightTune Window
	<b>Add Page</b> Adds a tabbed page to the display
	<b>Freeze Page</b> Halts the automatic refresh of displayed data within the window. It affects all panels within the window; however, individual panels can subsequently override the window frozen state using context menus.
	<b>Refresh Page</b> Causes all data displayed in all panels associated with the window to be refreshed once, regardless of the state of freeze on the window or in individual panels.
	<b>Process List</b> Displays the Process List panel
	<b>CPU Shielding and Binding</b> Displays the CPU Shielding and Binding panel
	<b>CPU Usage</b> Displays the CPU Usage panel
	<b>Memory: Physical</b> Displays the Physical Memory panel

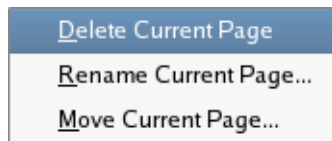
	<p><b>Memory: Swap</b> Displays the Swap Memory panel</p>
	<p><b>Interrupt Activity</b> Displays the Interrupt Activity panel</p>
	<p><b>Network Activity</b> Displays the Network Activity panel</p>
	<p><b>Disk Activity</b> Displays the Disk Activity panel</p>
	<p><b>Context Switches</b> Displays the Context Switches panel</p>
	<p><b>Kill dragged processes</b> If a process or thread is dragged to this icon, that process or thread is killed with a SIGKILL.</p>
	<p><b>Unbind dragged processes/interrupts</b> If a process, thread, or interrupt is dragged to this icon, that process, thread, or interrupt is unbound from any CPUs. That is, its CPU affinity is set to include all CPUs.</p> <p>Note that interrupts can be dragged to this icon only if NightTune is running as root or if the user running NightTune has been granted the proper capabilities (see “Capabilities” on page 1-2).</p>
	<p><b>Create panels for</b> When a new NightTune panel is created, the system shown here is the one which will be monitored by that new panel. The dropdown lists all NightTune remote systems to which you are connected. To create a panel for a system other than the one shown here, select it in the dropdown first.</p>
	<p><b>Connect to a system</b> Displays a dialog allowing connection to another system with NightTune installed</p>
	<p><b>Disconnect from a system</b> Displays a dialog allowing disconnection from a system</p>

## Pages

The remaining area of the main window is reserved for various tabbed pages which reflect and control the current configuration, provide browsing of programs, and viewing of live and recorded data.

Each page has a tab which contains the page title. When clicked or right-clicked, the page is raised to the top and becomes the current page.

Each tab has a context menu which allows you to manipulate the page position and title.



**Figure 2-7. Tab Context Menu**

### Delete Current Page

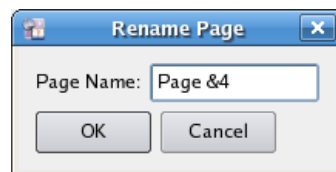
Mnemonic: D

This option deletes the current page.

### Rename Current Page

Mnemonic: R

This option displays a dialog which allows you to rename the current page.



**Figure 2-8. Rename Page Dialog**

If the page title contains an ampersand character (&), it causes the next character to be underlined in the title, provides a keyboard shortcut for that page, and the ampersand becomes invisible in the title that is shown for the page. In the example above, the keyboard shortcut for this page will be Alt+4 and the displayed title will become **Page 4**. Activating the shortcut for a page causes it to be raised to the top and it becomes the current page. Care should be taken when choosing shortcuts for pages so they do not conflict with other shortcuts. If you wish to have an ampersand displayed in the actual page title (as opposed to defining a shortcut), use two consecutive ampersand characters in the **Rename Page** dialog.

## Move Current Page

Mnemonic: M

This option displays a dialog which allows you to reposition the current page among other pages. This option will be disabled unless at least two viewing pages exist.

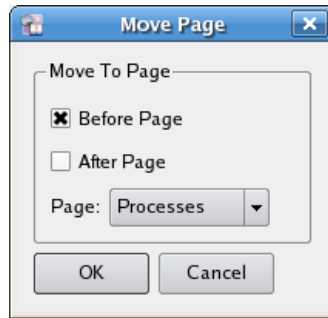


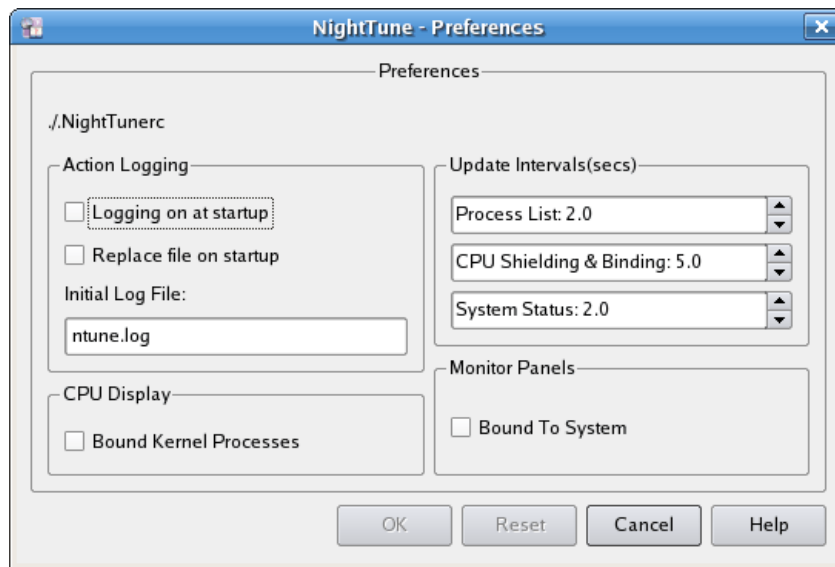
Figure 2-9. Move Page Dialog

## Preferences

The Preferences dialog, available from the File menu, allows you to tailor aspects of how NightTune operates.

These preferences are part of NightTune configuration data which can be saved to a configuration file so that subsequent invocations of NightTune can use the customized settings.

The following illustrates the Preferences dialog:



**Figure 2-10. Preferences Dialog**

The following paragraphs describe the labels and control features of this dialog:

### Configuration File

This area shows the name of the current configuration file. The file name can be changed using the **Save Config File As...** menu item from the **File** menu, or a new configuration file can be loaded using the **Load Config File...** menu item, also from the **File** menu.

### Action Logging

This area allows you to select logging attributes.

#### Logging on at startup

This checkbox controls whether logging is automatically initiated when NightTune is launched. The logging status is controlled using the **Logging** option of the **File** menu.



### Replace file on startup

This checkbox controls whether the log file should be truncated when NightTune is launched, so that no logged items from a previous session are retained. It has no effect unless **Logging on at startup** is activated.

### Initial log file:

This text area allows you to specify the name of the log file. This setting can be overridden by the `--logfile` or `-l` command line options.

## CPU Display

This area allows you to configure the CPU Shielding and Binding panel.

### Bound Kernel Processes

This checkbox controls whether or not the CPU Shielding and Binding panel shows kernel processes bound to particular CPUs. The Linux kernel includes some processes which are visible to the user (e.g. `ksoftirqd/0`). Generally, users do not wish to see these processes, and so the default behavior is to exclude them from the CPU Shielding and Binding panel. But if this checkbox is checked, they will be displayed as appropriate for their CPU binding.

## Update Intervals (secs)

This area allows you to control the update intervals used by NightTune to refresh displayed data. The units of all intervals are in seconds and can include fractional values.

An update interval of zero indicates that automatic updates will not occur. In such an instance, displays will be updated only when you click the **Refresh** tool icon or select a **Refresh** option from panel context menus.

### Process List

The interval between which the **Process List** panel scans for new processes and updates the **Process List** panel display.

### CPU Shielding & Binding

The interval between which the shielding attributes, bound processes and interrupts are updated in the CPU Shielding and Binding panel.

### System Status

The interval between which data within all system status panels are updated (see “System Status Panel” on page 3-1).

### **Bound to System**

If this item is checked then, when saving a configuration, information about the particular systems being monitored by each panel is saved as part of this configuration. If the configuration is used in the future, the panels will monitor those same systems. This would be useful if you wanted to create a configuration that always monitored the same system or set of systems.

If this item is not checked then, when saving a configuration, information about the particular systems being monitored by each panel is not saved as part of the configuration. If the configuration is used in the future, every panel would monitor the target system if specified, or the local system otherwise. This primarily is used for configurations designed to monitor a single system.

### **Control Buttons**

The operations are controlled using the buttons at the bottom of the dialog:

#### **OK**

Clicking OK applies the preferences selected in the dialog and closes the dialog.

#### **Reset**

Clicking the **Reset** button discards any changes not yet applied and refreshes the displayed values to the current values.

#### **Cancel**

Clicking the **Cancel** button closes the dialog; any changes made to the dialog that have not been applied are discarded.

#### **Help**

Clicking the **Help** button displays this section in the online help viewer.

This chapter describes NightTune's panels, which provide functional units for displaying and modifying process and system activities. They are:

- Context Switches
- CPU Shielding and Binding
- CPU Usage
- Disk Activity
- Interrupt Activity
- Memory Activity
- Kernel Memory
- Physical Memory
- Swap Space
- Network Activity
- NUMA Memory
- NUMA Memory Activity
- NUMA Memory Configuration
- Process List

## System Status Panel

Many panels are of a class called System Status Panels. These panels are divided into three panes: **Text**, **Bar graph**, and **Line graph**. NightTune allows the panes to be resized within the panel, but they cannot be moved or reordered in the same way that full panels can. However, it is possible to create panels where one or two of the panes are hidden. When creating new system status panels via the **Monitor** menu, there is a sub-menu. If one of the **Text** pane, **Bar graph** pane, or **Line graph** pane items is selected, then a panel will be created with only that single pane visible. Alternately, after the creation of the panel by any means, the context menu can be used to control which panes are visible. The three menu options, **Show text pane**, **Show bar graph pane**, and **Show line graph pane** control the visibility of the **Text**, **Bar graph**, and **Line graph** panes, respectively.

In the **Text** pane, the current values for the last update are displayed numerically so that exact numbers can be seen.

In the **Bar graph** pane, the current values for the last update are displayed visually with horizontal bars. This makes it possible to compare relative values at a glance.

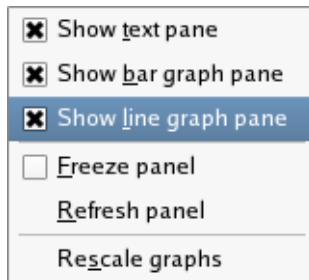
In the **Line graph** pane, the current values for the last update and a number of historical values are displayed. Each value is displayed along the vertical axis while time is represented by the horizontal axis. The rightmost value in the display represents the most recent value.

The displayed information in system status panels is automatically refreshed periodically as controlled by the refresh interval, which can be adjusted using the **Preferences** dialog displayed from the **Preferences...** option of the **File** menu (see “Preferences” on page 2-16). The values in the **Line graph** pane automatically scroll after each refresh once the number of values collected is sufficient to fill the graph.

Bar and line graphs are scaled automatically by NightTune, usually based on peak values over the duration of the NightTune session. You can force NightTune to rescale the graphs based on current values using the **Rescale graphs** option from the System Status context menu.

## System Status Context Menu

While positioned in a system status panel, right-clicking displays a context menu, which typically contains the following content:



**Figure 3-1. Typical Context Menu**

The following paragraphs describe the menu items in detail:

### Show legend

Mnemonic: G

This menu item toggles the visibility of the legend defining the colors used in the panel. Not every system status panel context menu has this item, but many do.

### Show/Hide text pane

Mnemonic: T

This menu item toggles the visibility of the Text pane within the panel.

### **Show/Hide bar graph pane**

Mnemonic: B

This menu item toggles the visibility of the **Bar graph** pane within the panel.

### **Show/Hide line graph pane**

Mnemonic: L

This menu item toggles the visibility of the **Line graph** pane within the panel.

### **Freeze/Unfreeze panel**

Mnemonic: F

This menu item toggles the **Freeze** setting for the panel. When frozen, data values are not refreshed automatically. This menu item overrides the **Freeze** setting for the window, but only applies to the particular panel.

### **Refresh panel**

Mnemonic: R

This menu item causes all data within the panel to be refreshed once, regardless of the **Freeze** setting.

### **Rescale graphs**

Mnemonic: S

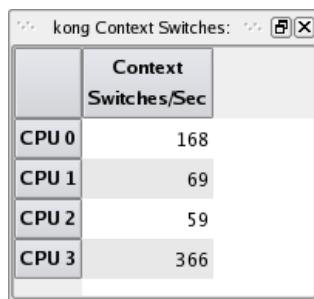
This menu item causes NightTune to rescale all graphs within the panel based on the current data for the panel.

## Context Switches Panel

The Context Switches panel is a system status panel (see “System Status Panel” on page 3-1). Each pane displays the number of context switches per second per CPU that occurred over the period of time defined by the refresh interval.

### Context Switches Text Pane

The following illustrates the Context Switches Text pane:



	Context Switches/Sec
CPU 0	168
CPU 1	69
CPU 2	59
CPU 3	366

**Figure 3-2. Context Switches Text Pane**

The information displayed in this area includes:

#### **CPU Number**

The logical CPU number is displayed in this column.

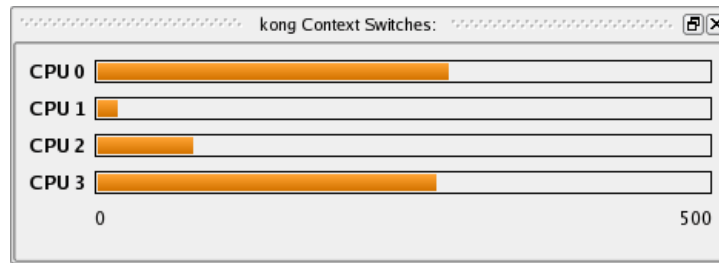
#### **Context Switch Rate**

The number of context switches per second is displayed in this column.

### Context Switches Bar Graph Pane

The Context Switches Bar Graph pane provides individual bar graphs for each CPU.

The following illustrates the Context Switches Bar Graph pane:



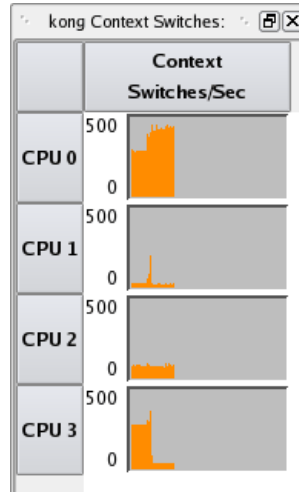
**Figure 3-3. Context Switches Bar Graph Pane**

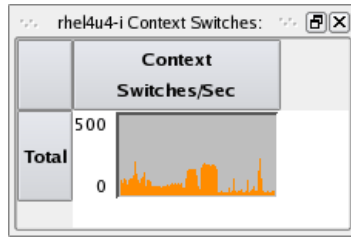
The number of context switches per second is displayed as a horizontal bar for each CPU.

## Context Switches Line Graph Pane

The Context Switches Line Graph pane provides individual line graphs detailing the context switch rate for each CPU.

The following illustrates the Context Switches Line Graph pane:





**Figure 3-4. Context Switches Line Graph Pane**

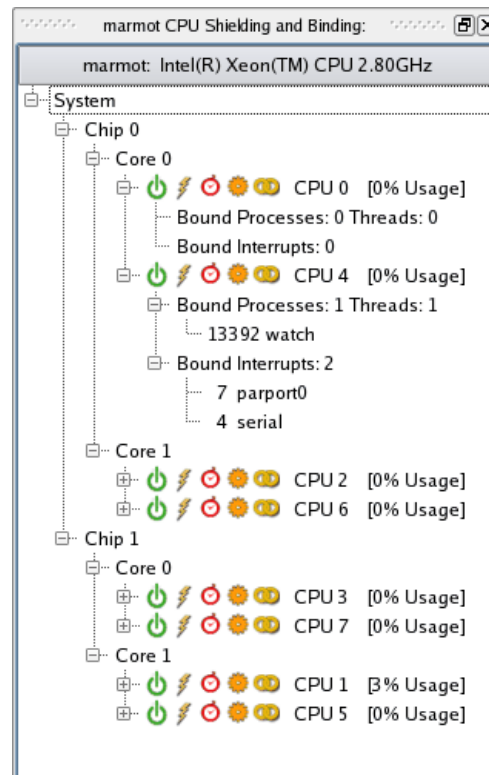
The number of context switches per second for a specific CPU is displayed vertically in each graph.



## CPU Shielding and Binding Panel

The CPU Shielding and Binding panel displays a summary of the state of shielding, process binding, and interrupt binding on each CPU. It allows you to change the shielding attributes of CPUs and to redefine the CPU binding of specific processes or interrupts.

The following illustrates the CPU Shielding and Binding panel:



**Figure 3-5. CPU Shielding and Binding Panel**

All CPUs on the system are displayed in tree format. A chip typically represents a physical component within the system. On multi-core systems, each chip contains two or more cores which essentially act as independent processors. On hyper-threaded architectures, each core can be subdivided into two logical processors which behave similarly to independent processors in some ways, although they do share some logic and so execution on one processor can cause delays in execution in its sibling processor.

For each CPU, several icons represent attributes of that CPU. The symbols are described at “CPU Shielding Operations” on page 3-8. Changes to the shielding attributes of a CPU are made from the CPU Shielding dialog.

When the display is expanded, processes and interrupts that are bound to each CPU are shown. The display can be expanded by clicking on the plus sign (+) in the tree to expand the corresponding entry, or by selecting an entry, right-clicking to display the CPU Shielding and Binding panel context menu and selecting **Expand All** (see “CPU

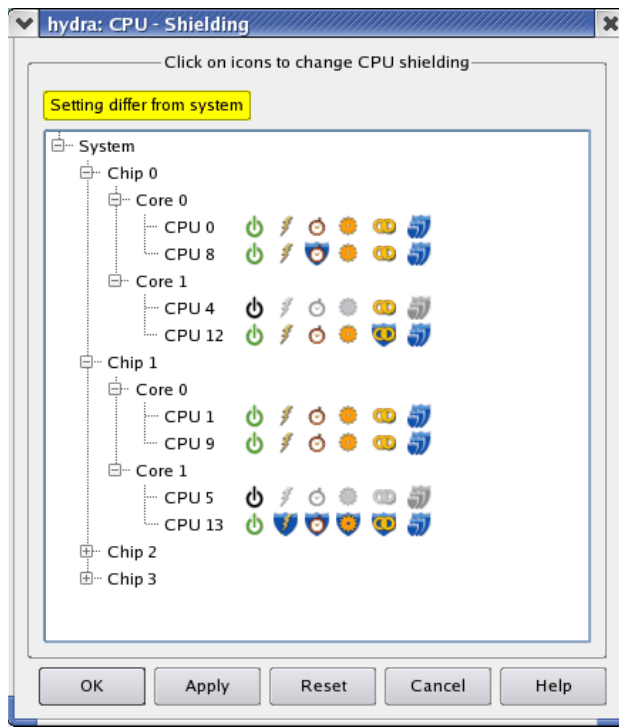
Shielding and Binding Context Menu” on page 3-11). Bound kernel processes also are displayed when **Bound Kernel Processes** is activated from the **Preferences** dialog.

## CPU Shielding Operations

The shielding attributes of a CPU can be changed with the CPU Shielding dialog.

To display the CPU Shielding dialog, right-click in the CPU Shielding and Binding panel and select the **Change Shielding...** menu item from the context menu (see “CPU Shielding and Binding Context Menu” on page 3-11).

The following illustrates the CPU Shielding dialog:



**Figure 3-6. CPU Shielding Dialog**

All CPUs on the system are displayed in tree format. CPUs which are hyper-threaded siblings (two hyper-threaded CPUs on the same core) are grouped together.

Each line shows the shielding attributes for one CPU.

By clicking on an icon for a CPU, the corresponding attribute is toggled and the icon display changes. If shielding is activated, a blue shield will appear for that attribute; if not activated, no shield appears.

Other changes that occur based on toggling an icon will be displayed, too. For example, if shielding a hyper-threaded CPU from hyper-threading, its sibling CPU is marked down.

To apply the changes, click on the **Apply** button at the bottom of the dialog. To apply the changes and dismiss the dialog, click on the **OK** button at the bottom of the dialog.

The following paragraphs describe the controls in detail:

### Active Toggle

The **Active** icon indicates whether the CPU is marked **Down** (black), meaning unavailable for processing; or marked **Active** (green), meaning available for processing. A common reason to mark a CPU **Down** is so that it does not interfere with processing on its sibling hyperthreaded CPU.

### Interrupt Shielding Toggle

The **Interrupt Shielding** toggle indicates whether the CPU will be shielded from interrupts. Note that the **Local Timer** interrupt is not affected by this setting. When shielded, the only interrupts that will be handled by this CPU are those whose CPU affinities include only this CPU.

### Local Timer Shielding Toggle

The **Local Timer Shielding** toggle indicates whether the CPU will be shielded from the **Local Timer** interrupt.

### Process Shielding Toggle

The **Process Shielding** toggle indicates whether the CPU will be shielded from processes. When shielded, the only processes, including kernel daemons, allowed to execute on the CPU are those whose CPU affinities include only this CPU.

### Hyper-Threading Shielding Toggle

The **Hyper-Threading Shielding** toggle indicates whether the hyper-threaded sibling CPU is marked **Active** or **Down**. When shielded, the **Active** toggle for the sibling is set to the **Down** state.

### Shield All Button

The **Shield All** button enables each of the **Interrupt Shielding**, **Local Timer Shielding**, and **Process Shielding** toggles. This button is not a toggle.

The buttons at the bottom of the CPU Shielding dialog perform the following functions:

#### OK

Clicking **OK** attempts to apply the shielding changes to the system and closes the dialog when complete. This will fail if the user does not have sufficient capabilities; see “Capabilities” on page 1-2 for more information. Additionally, the shielding changes may conflict with current CPU usage. Attempting to set a CPU’s status to **Down** will be rejected if processes or interrupts are currently bound to the CPU. A diagnostic will be displayed if the shielding operation fails.

### Apply

Clicking **Apply** attempts to apply the shielding changes to the system. This will fail if the user does not have sufficient capabilities; see “Capabilities” on page 1-2 for more information. Additionally, the shielding changes may conflict with current CPU usage. Attempting to set a CPU's status to **Down** will be rejected if processes or interrupts are currently bound to the CPU. A diagnostic will be displayed if the shielding operation fails.

### Reset

Clicking **Reset** restores the CPU Shielding dialog display with current information, discarding any changes that have not yet been applied.

### Cancel

Clicking **Cancel** closes the dialog. Any changes that have not been applied are discarded.

### Help

Clicking **Help** presents this section of the manual in the online help viewer.

## CPU Shielding and Binding Drag and Drop Operations

To drag a process or interrupt from one CPU to another, select the items of interest under **Bound Processes** or **Bound Interrupts** using the mouse. Click and hold the mouse button, then drag the pointer to the destination area and release. Alternatively, if no processes or interrupts are selected in the area, click the mouse button anywhere on the row which describes the process or interrupt and drag to the destination area and release.

The CPU Shielding and Binding panel supports the following drag and drop operations:

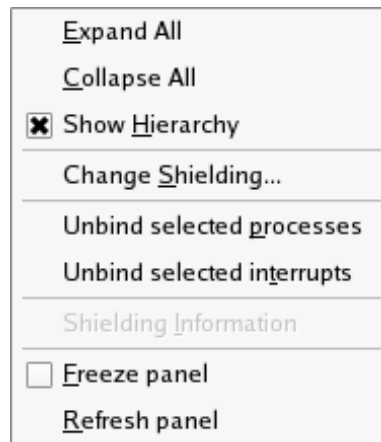
- Dragging bound processes or interrupts from one CPU line to another CPU line in the CPU Shielding and Binding panel binds them to the CPU corresponding to the target line. When dragging a multi-threaded process (identified by a numeric pair enclosed in parentheses following the process name) from a CPU line to another CPU line, only the threads bound to the source CPU are rebound.
- Dragging bound processes or interrupts to the Unbind tool icon causes the processes or interrupts to be unbound from any CPUs.
- Dragging processes to the Kill tool icon causes the processes to be killed with a **SIGKILL** signal.
- Dragging a CPU to another CPU will rebind all processes and interrupts bound to the first CPU to the second.
- Dragging a CPU to the Unbind tool icon causes all processes and interrupts bound to the CPU to become unbound from any CPUs.

- Dragging a CPU to the Kill tool icon causes all processes bound to the CPU to be killed with a **SIGKILL** signal.

Note that interrupts can be dragged only if NightTune is running as `root` or if the user running NightTune has been granted the proper capabilities (see “Capabilities” on page 1-2).

## CPU Shielding and Binding Context Menu

While positioned over a CPU line in the CPU Shielding and Binding panel, right-clicking displays a context menu for the associated CPU, as shown below. While positioned over a bound process or interrupt, right-clicking creates a context menu with the choices for unbinding the processes or interrupts enabled too.



**Figure 3-7. CPU Shielding and Binding Context Menu**

All activities in the CPU Shielding and Binding context menu apply solely to the associated CPU, except where noted in the detailed descriptions below:

### Expand All

Mnemonic: E

This menu item expands the selected item and all interior items that can be expanded.

### Collapse All

Mnemonic: C

This menu item collapses the selected item and all interior items.

## Show Hierarchy

Mnemonic: H

This menu item toggles the display mode of the panel. When this item is selected, CPUs are displayed in a hierarchy which indicates the relationship between CPUs and their parent chip. When not selected, the CPUs are shown in a flat list.

Clearing Show Hierarchy can be useful when you have a number of CPUs and need to save display space in the panel in order to see all CPUs.

## Change Shielding

Mnemonic: S

This menu item displays the CPU Shielding dialog which allows you to change the shielding attributes of CPUs on the system. See “CPU Shielding Operations” on page 3-8 for details.

## Unbind selected processes

Mnemonic: P

This menu item immediately changes the CPU affinity of the selected processes in the Bounded Processes list to include all CPUs; effectively unbinding them from the CPU. This item is not available on a system with only a single CPU.

## Unbind selected interrupts

Mnemonic: T

This menu item immediately changes the CPU affinity of the selected interrupts in the Bounded Interrupts list to include all CPUs; effectively unbinding them from the CPU. This item is not available on a system with only a single CPU.

Note that interrupts can be unbound only if NightTune is running as `root` or if the user running NightTune has been granted the proper capabilities (see “Capabilities” on page 1-2).

## Shielding Information

Mnemonic: I

This menu item pops up a dialog containing a textual description of the state of shielding for the selected CPU.

See the *RedHawk Linux User's Guide* for more information on CPU shielding and process and interrupt CPU affinity.

## Freeze Panel

Mnemonic: F

This menu item toggles the **Freeze** setting for this panel. When frozen, this panel is not refreshed automatically.

### Refresh Panel

Mnemonic: R

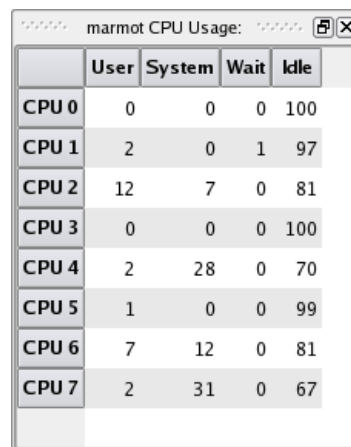
This menu item causes all data in all displays within the **CPU Shielding and Binding** panel to be refreshed once, regardless of the **Freeze** setting.

## CPU Usage Panel

The **CPU Usage** panel is a system status panel (see “System Status Panel” on page 3-1). Each pane displays percentages that reflect CPU activity over the period of time defined by the refresh interval.

### CPU Usage Text Pane

The following illustrates the **CPU Usage Text** pane:



	User	System	Wait	Idle
<b>CPU 0</b>	0	0	0	100
<b>CPU 1</b>	2	0	1	97
<b>CPU 2</b>	12	7	0	81
<b>CPU 3</b>	0	0	0	100
<b>CPU 4</b>	2	28	0	70
<b>CPU 5</b>	1	0	0	99
<b>CPU 6</b>	7	12	0	81
<b>CPU 7</b>	2	31	0	67

**Figure 3-8. CPU Usage Text Pane**

The **CPU Usage Text** pane provides information on CPU usage for each CPU in the system.

The information displayed in this area includes:

#### CPU

The logical CPU number is displayed in this column.

**User**

The percentage of time the CPU was executing in user mode is displayed in this column. This excludes kernel execution but does include execution of kernel daemons which handle post-interrupt processing.

**System**

The percentage of time the CPU was executing in the operating system kernel is displayed in this column. This includes time spent executing system service calls on behalf of user processes as well as interrupt and machine exception processing.

**Wait**

The percentage of time the CPU was idle with at least one process waiting for an I/O operation (on any CPU) to complete is displayed in this column.

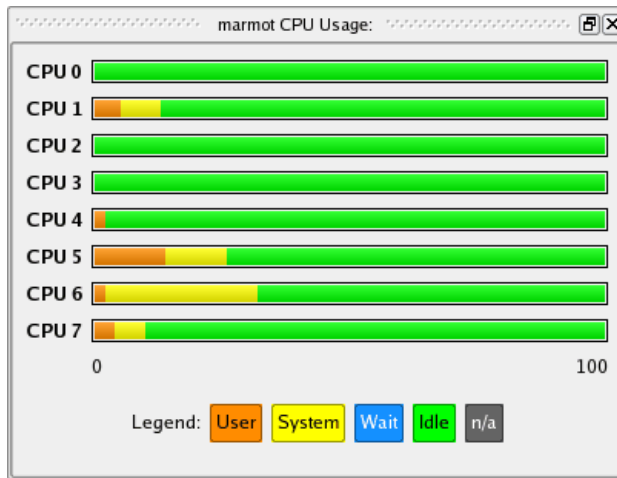
**Idle**

The percentage of time the CPU was executing the idle loop is displayed in this column. This excludes Wait time.

**CPU Usage Bar Graph Pane**

The CPU Usage Bar Graph pane shows the percentage of CPU time used for each CPU.

The following illustrates the CPU Usage Bar Graph pane:



**Figure 3-9. CPU Usage Bar Graph Pane**

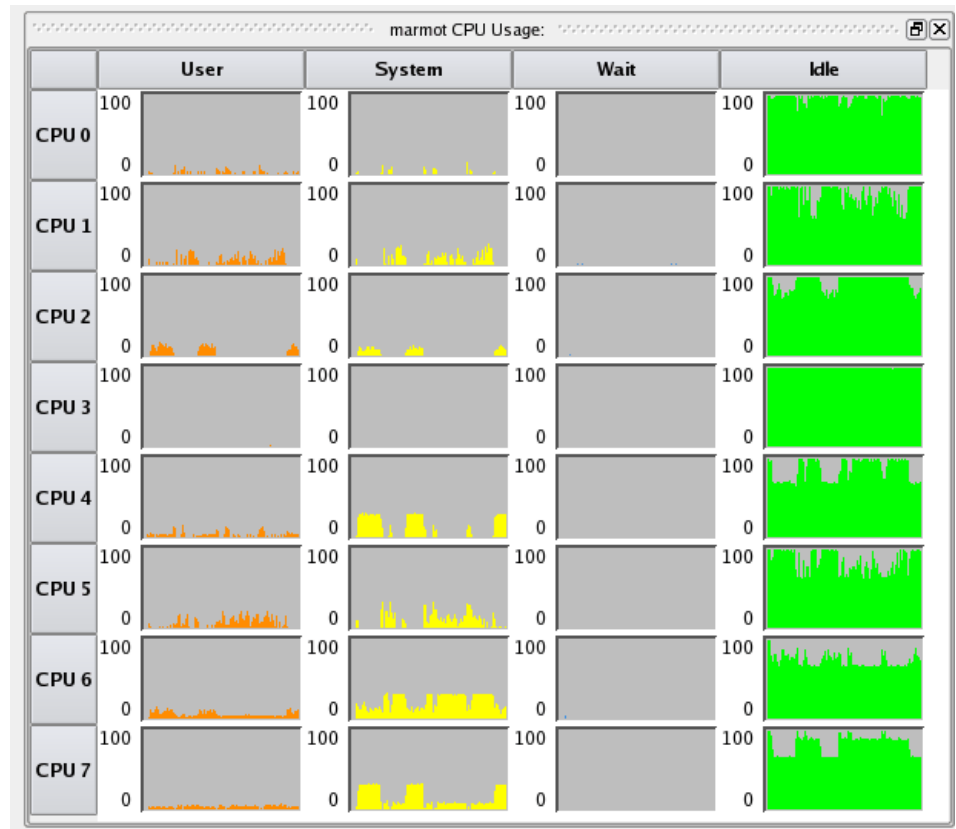
For each CPU, the User, System, Wait, and Idle times are shown as percentages of CPU execution using color-coded, horizontal bars. See “CPU Usage Text Pane” on page 3-13 for the definitions of User, System, Wait, and Idle times.



## CPU Usage Line Graph Pane

The CPU Usage Line Graph pane provides individual line graphs for User, System, Wait, and Idle times, expressed as a percentage of CPU execution for each CPU.

The following illustrates the CPU Usage Line Graph pane:



**Figure 3-10. CPU Usage Line Graph Pane**

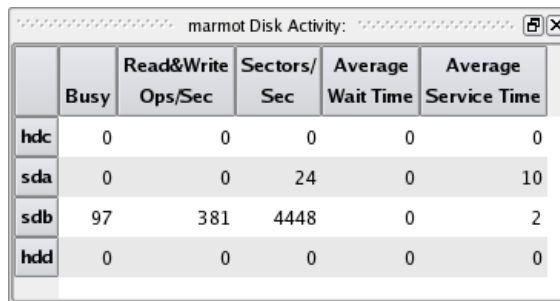
See "CPU Usage Text Pane" on page 3-13 for the definitions of User, System, Wait, and Idle times.

## Disk Activity Panel

The Disk Activity panel is a system status panel (see “System Status Panel” on page 3-1). Each pane displays statistics related to disk operations that occurred over the period of time defined by the refresh interval. Statistics for individual disks are displayed.

### Disk Activity Text Pane

The following illustrates the Disk Activity Text pane:



	Busy	Read&Write Ops/Sec	Sectors/ Sec	Average Wait Time	Average Service Time
<b>hdc</b>	0	0	0	0	0
<b>sda</b>	0	0	24	0	10
<b>sdb</b>	97	381	4448	0	2
<b>hdd</b>	0	0	0	0	0

**Figure 3-11. Disk Activity Text Pane**

The information displayed in this area includes:

#### Disk ID

The disk device name is displayed in this column.

#### Busy Percentage

The percentage of time the disk was busy servicing a transfer request is displayed in this column.

#### Read & Write Operations per Second

The number of read and write transfer requests serviced by the disk per second is displayed in this column.

#### Sectors per Second

The number of sectors transferred to or from the disk per second is displayed in this column.

### Average Wait Time

The average wait time, in milliseconds, that transfer requests waited on the queue before being serviced is displayed in this column.

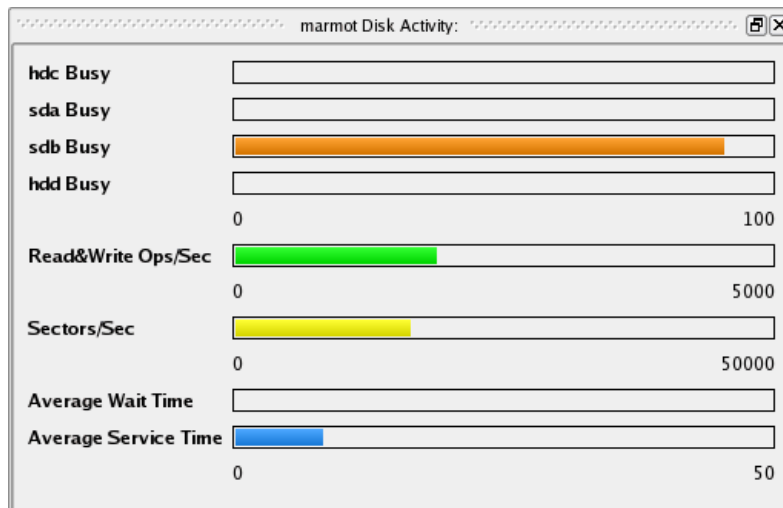
### Average Service Time

The average service transfer time, in milliseconds, is displayed in this column. The time includes seek time, rotational latency, and data transfer time.

## Disk Activity Bar Graph Pane

The Disk Activity Bar Graph pane provides individual bar graphs detailing metrics related to disk operations.

The following illustrates the Disk Activity Bar Graph pane:



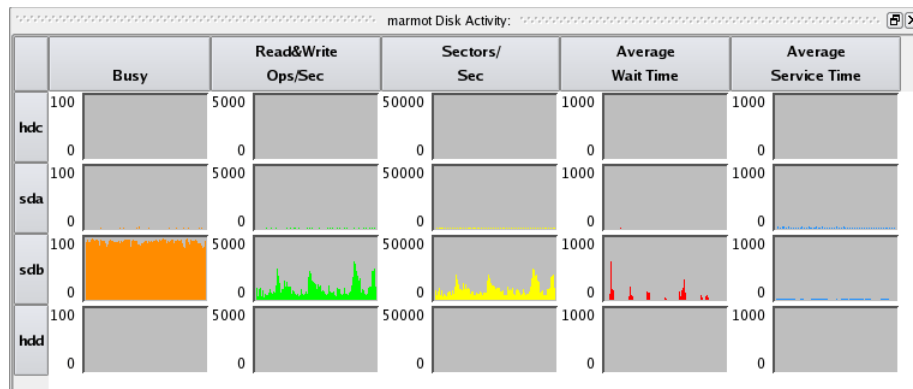
**Figure 3-12. Disk Activity Bar Graph Pane**

Individual bar graphs are shown for each disk for the Busy metric. The remaining metrics pertain to the system as a whole. See “Disk Activity Text Pane” on page 3-16 for definitions of these metrics. The values of the metrics are displayed horizontally in each graph.

## Disk Activity Line Graph Pane

The Disk Activity Line Graph pane provides individual line graphs detailing metrics related to disk operations.

The following illustrates the Disk Activity Line Graph pane:



**Figure 3-13. Disk Activity Line Graph Pane**

Individual line graphs are shown for each disk for each metric. The value of each metric is displayed vertically in each graph. See “Disk Activity Text Pane” on page 3-16 for definitions of the metrics.

## Interrupt Activity Panel

The Interrupt Activity panel is a system status panel (see “System Status Panel” on page 3-1). Each pane displays interrupt activity on the system.

## Interrupt Activity Text Pane

The following illustrates the Interrupt Activity Text pane:

	0	1	2	3	Description
0	1000	0	0	0	timer
3	0	0	0	0	KGDB-stub
4	0	0	0	0	serial
9	0	0	0	0	acpi
14	0	0	0	0	ide0
177	0	0	0	0	uhci_hcd
185	0	0	0	0	uhci_hcd
193	0	0	0	0	uhci_hcd
201	0	0	0	0	ehci_hcd
209	0	0	0	0	Intel 82801DB-ICH4
217	0	0	0	0	rcim
225	126	0	0	0	eth0
NMI	0	0	0	0	Non-maskable interrupts
LOC	1000	1000	1000	0	Local interrupts
RES	122	0	128	35	Rescheduling interrupts
CAL	0	0	0	0	function call interrupts
TLB	0	0	0	0	TLB shootdowns
TRM	0	0	0	0	Thermal event interrupts
SPU	0	0	0	0	Spurious interrupts
ERR	0	0	0	0	Error interrupts

Legend: Unshielded Shielded Inactive Bound

**Figure 3-14. Interrupt Activity Text Pane**


The Interrupt Activity Text pane provides information on individual interrupts and allows the user to change an interrupt’s CPU affinity.

The information displayed in this area includes:

### Interrupt Value

The IRQ value or mnemonic for the interrupt is displayed.

## Interrupts Per Second

For each CPU, the number of interrupts per second is displayed. For interrupts which have a CPU affinity which does not include all CPUs, a shield icon  is displayed for that interrupt in columns associated with CPUs in its CPU affinity mask. Thus, at a glance you can determine which interrupts have an affinity with which CPUs.

For CPUs which are shielded from interrupts, the column header for that CPU will have a blue shield.

## Interrupt Description

For interrupts associated with devices, the device name is displayed. For other interrupts, a short functional description of their purpose is displayed.

## Interrupt Control Drag and Drop Operations

Individual interrupts can be dragged onto various destination panels and drop targets.

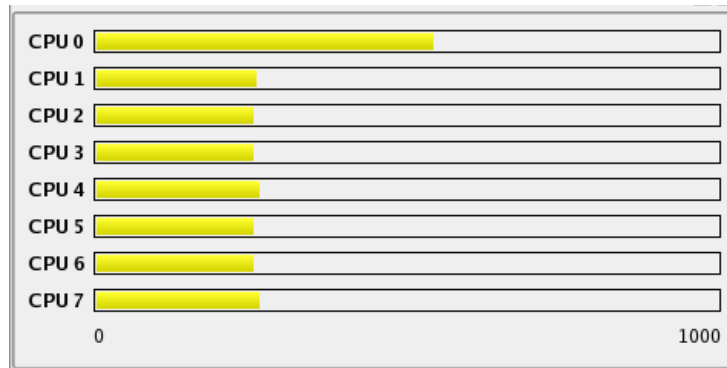
To drag an interrupt, click anywhere on the interrupt row and drag the pointer to the destination area and release.

- Dragging an interrupt onto the **Unbind** tool icon causes the interrupt to be unbound from any CPUs.
- Dragging an interrupt onto a CPU in the **CPU Shielding and Binding** panel (see “CPU Shielding and Binding Panel” on page 3-7) will bind it to that CPU.

## Interrupt Activity Bar Graph Pane

The Interrupt Activity Bar Graph pane provides individual bar graphs showing the number of interrupts per second per CPU.

The following illustrates the Interrupt Activity Bar Graph pane:



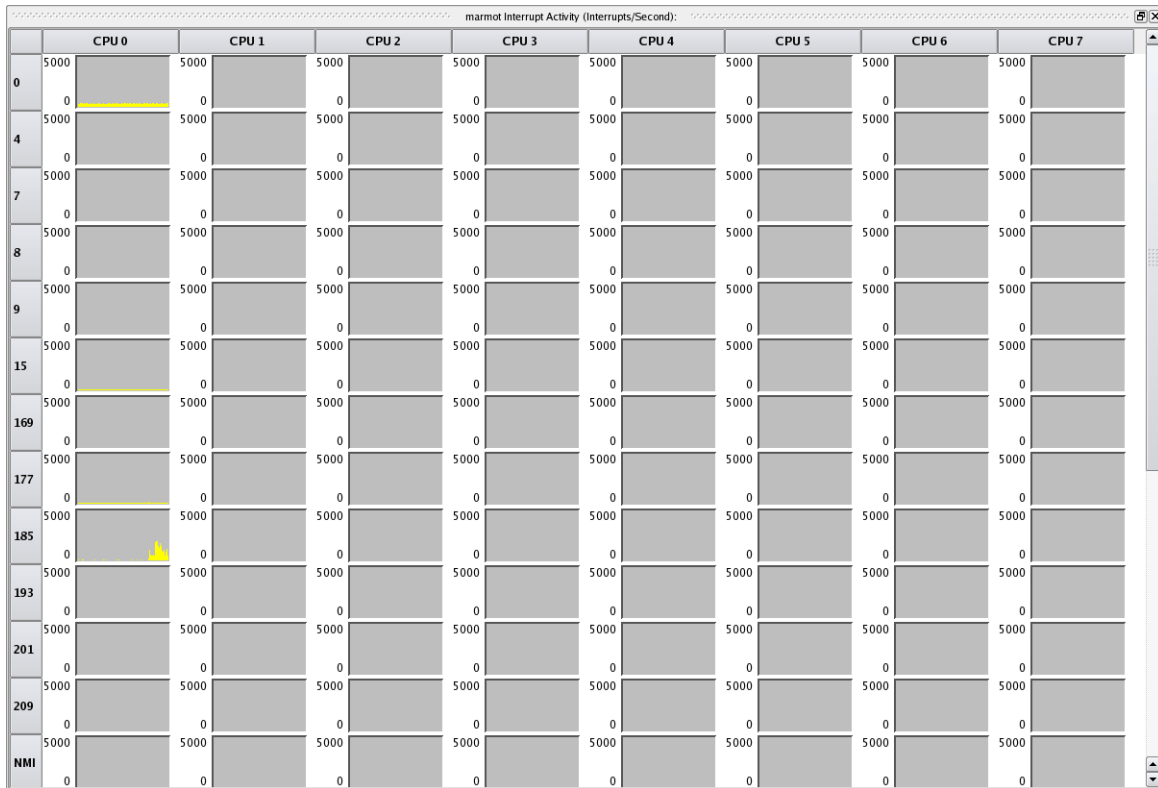
**Figure 3-15. Interrupt Activity Bar Graph Pane**

Regardless of which interrupts are displayed in the Interrupt Activity Text and Interrupt Activity Line Graph panes, these bar graphs represent **all** interrupt activity on the system.

## Interrupt Activity Line Graph Pane

The Interrupt Activity Line Graph pane provides individual line graphs for each interrupt for each CPU.

The following illustrates the Interrupt Activity Line Graph pane:



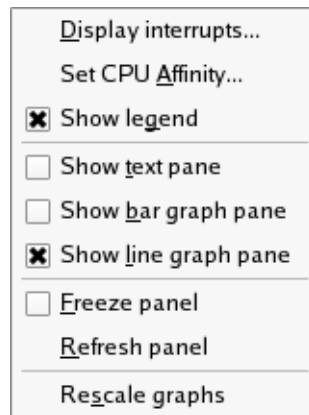
**Figure 3-16. Interrupt Activity Line Graph Pane**

The number of interrupts per second is displayed in each graph.



## Interrupt Activity Context Menu

While positioned in the Interrupt Activity panel, right-clicking displays the Interrupt Activity context menu:



**Figure 3-17. Interrupt Activity Context Menu**

Many items are common to all system status context menus (see “System Status Context Menu” on page 3-2), but some new menu items are present in this menu:

### Display Interrupts...

Mnemonic: D

This menu item displays the Interrupt Selection dialog which allows you to select which interrupts will be displayed in the Interrupt Activity Text and Interrupt Activity Line Graph panes in the panel. This menu item does not affect the interrupt activity as shown in the Interrupt Activity Bar Graph pane.

### Set CPU Affinity...

Mnemonic: A

This menu item displays the Interrupt Affinity dialog which allows you to view and change the affinity of the selected interrupt. See “Interrupt Affinity Dialog” on page 3-24.

## Interrupt Affinity Dialog

The Interrupt Affinity dialog displays the current CPU affinity of a specific interrupt and allows you to change it. It is accessed by right-clicking on the Interrupt Activity Text pane and selecting Set CPU Affinity... from the context menu.

The following illustrates the Interrupt Affinity dialog:

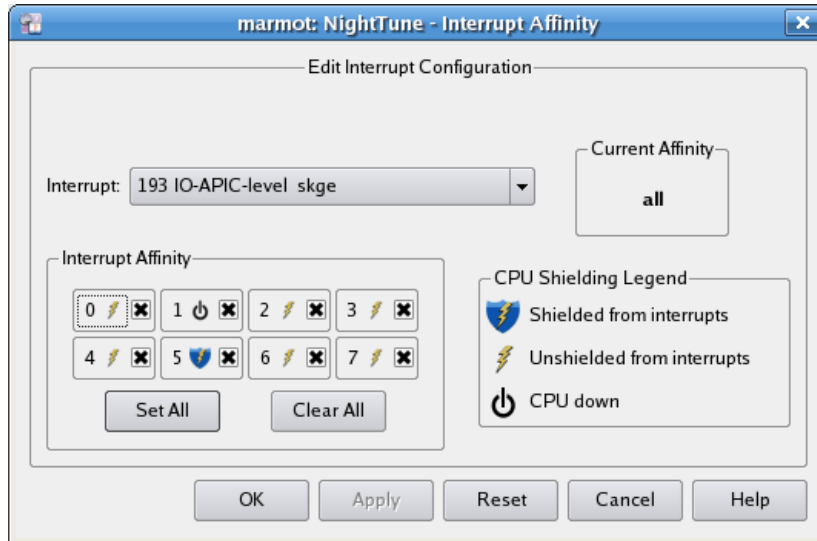


Figure 3-18. Interrupt Affinity Dialog


The dialog consists of the following functional labels and controls:

### Interrupt

The IRQ value, name, and description currently associated with the dialog is displayed. A drop-down list allows you to select a different interrupt.

### Interrupt Affinity

The Interrupt Affinity area allows you to specify individual CPUs where the interrupt can be processed. If a single CPU checkbox is checked, the interrupt is bound to that CPU and the interrupt is displayed in the Bound Interrupts list for the associated CPU in any CPU Shielding and Binding panel.

For each CPU that has been shielded from interrupts, the 'shielded from interrupts' icon  is displayed. If a CPU has been shielded from interrupts, no interrupt will be processed on that CPU unless its affinity includes that CPU and includes no non-shielded CPUs.

The Set All and Clear All buttons make it possible to set or clear all CPU checkboxes quickly without having to click on each one individually.

### **Current Affinity**

This area displays the affinity currently in effect for the interrupt as a hexadecimal mask or as the word **all**.

### **CPU Shielding Legend**

This area displays the symbols used to indicate the status of the CPUs on the system: shielded from interrupts, unshielded from interrupts, and CPU down.

### **OK**

Clicking **OK** applies the CPU affinity selected in the dialog to the interrupt, and then closes the dialog. If the user lacks the capabilities to change the CPU affinity, the **OK** button will be desensitized. See “Capabilities” on page 1-2 for more information.

### **Apply**

Clicking the **Apply** button applies the CPU affinity selected in the dialog to the interrupt. If the user lacks the capabilities to change the CPU affinity, the **Apply** button will be desensitized. See “Capabilities” on page 1-2 for more information.

### **Reset**

Clicking the **Reset** button discards any changes not yet applied and refreshes the displayed values to the current affinity of the interrupt.

### **Cancel**

Clicking the **Cancel** button closes the dialog; any changes made to the dialog that have not been applied are discarded.

### **Help**

Clicking the **Help** button displays this section in the online help viewer.

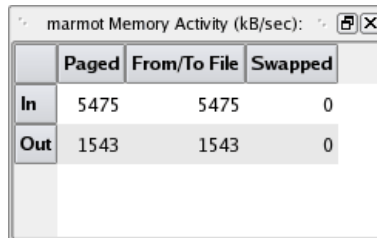
## **Memory Activity Panel**

The Memory Activity panel is a system status panel (see “System Status Panel” on page 3-1). Each pane displays interrupt activity on the system. Each pane displays the number

of physical memory page transfers per second that occurred over the period of time defined by the refresh interval.

## Memory Activity Text Pane

The following illustrates the Memory Activity Text pane:



	Paged	From/To File	Swapped
In	5475	5475	0
Out	1543	1543	0

**Figure 3-19. Memory Activity Text Pane**

The information displayed in this area includes:

### Paged

The number of KB read in or written out per second due to paging is displayed in this column.

### From/To File

The number of KB read in from disk files to memory and read out from memory to disk files is displayed in this column.

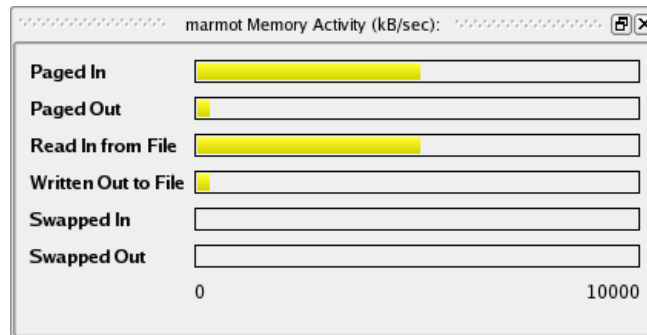
### Swapped

The number of KB read in from swap space to memory and read out from memory to swap space is displayed in this column.

## Memory Activity Bar Graph Pane

The Memory Activity Bar Graph pane provides individual bar graphs detailing the paging rates.

The following illustrates the Memory Activity Bar Graph pane:



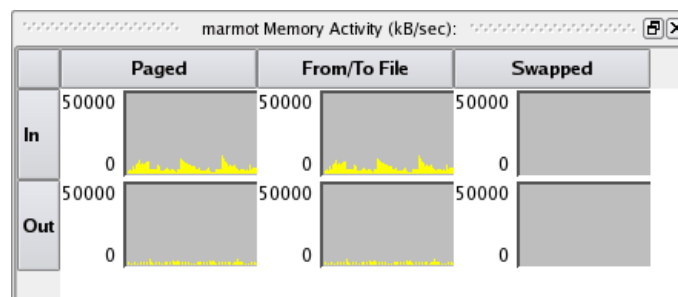
**Figure 3-20. Memory Activity Bar Graph Pane**

The number of KB in physical memory page transfers per second for each page operation type is displayed as a horizontal line in each graph. See “Memory Activity Text Pane” on page 3-26 for definitions of the metrics.

## Memory Activity Line Graph Pane

The Memory Activity Line Graph pane provides individual line graphs detailing the paging rates.

The following illustrates the Memory Activity Line Graph pane:



**Figure 3-21. Memory Activity Line Graph Pane**

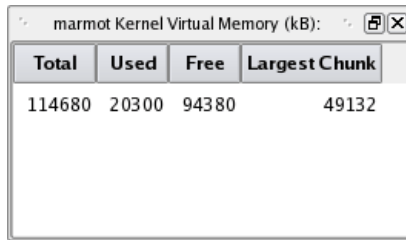
The number of KB in physical memory page transfers per second for each operation type is displayed as a vertical line in each graph. See “Memory Activity Text Pane” on page 3-26 for definitions of the metrics.

## Kernel Virtual Memory Panel

The Memory Activity panel is a system status panel (see “System Status Panel” on page 3-1). Each pane displays the allocation of kernel virtual memory that exists at the time of the last refresh.

### Kernel Virtual Memory Text Pane

The following illustrates the Kernel Virtual Memory Text pane:



Total	Used	Free	Largest Chunk
114680	20300	94380	49132

**Figure 3-22. Kernel Virtual Memory Text Pane**

The information displayed in this area includes:

#### **Total**

The total number of KB reserved for kernel virtual memory (vmalloc) on the system is displayed in this column.

#### **Used**

The number of KB of kernel virtual memory (vmalloc) allocated by the kernel or user processes is displayed in this column.

#### **Free**

The number of KB of kernel virtual memory (vmalloc) not being used is displayed in this column.

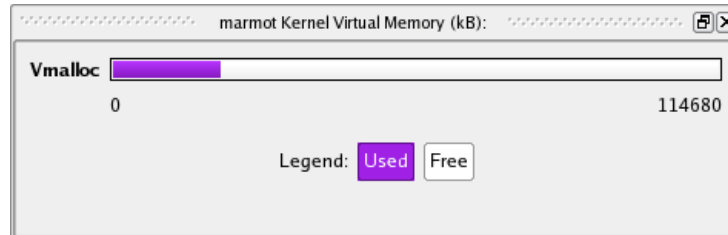
#### **Largest Chunk**

The number of KB in the largest contiguous block of free kernel virtual memory (vmalloc) is displayed in this column.

## Kernel Virtual Memory Bar Graph Pane

The Kernel Virtual Memory Bar Graph pane provides individual bar graphs detailing the allocation of kernel virtual memory on the system.

The following illustrates the Kernel Virtual Memory Bar Graph pane:



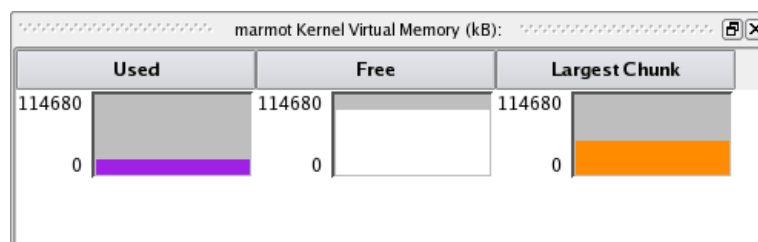
**Figure 3-23. Kernel Virtual Memory Bar Graph Pane**

The Total kernel virtual memory for the system is represented by the full width of the graph, and is expressed in KB. The amount that currently is Used is displayed in purple, and the amount that is Free is displayed in white. See “Kernel Virtual Memory Text Pane” on page 3-28 for definitions of the metrics.

## Kernel Virtual Memory Line Graph Pane

The Kernel Virtual Memory Line Graph pane provides individual line graphs detailing the allocation of kernel virtual memory (vmalloc).

The following illustrates the Kernel Virtual Memory Line Graph pane:



**Figure 3-24. Kernel Virtual Memory Line Graph Pane**

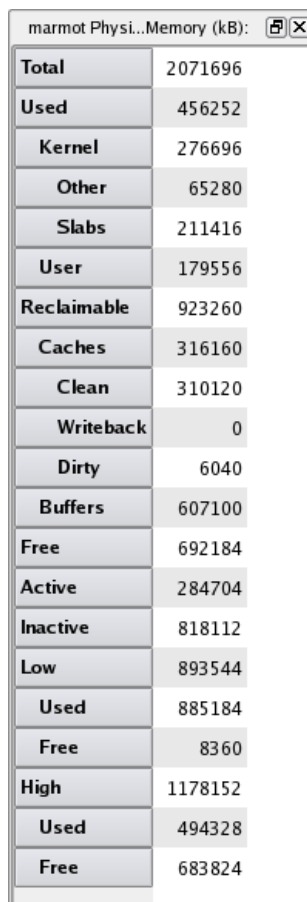
The allocation of kernel virtual memory is displayed as a vertical line in each graph. See “Kernel Virtual Memory Text Pane” on page 3-28 for definitions of the metrics

## Physical Memory Panel

The Physical Memory panel is a system status panel (see “System Status Panel” on page 3-1). Each pane displays the allocation of physical memory that exists at the time of the last refresh.

## Physical Memory Text Pane

The following illustrates the Physical Memory Text pane:



The screenshot shows a window titled "marmot Physi...Memory (kB):" with a close button. It contains a table of memory usage statistics. The table has two columns: the category name and the value in kB. The categories are listed in descending order of total usage.

<b>Total</b>	2071696
<b>Used</b>	456252
<b>Kernel</b>	276696
<b>Other</b>	65280
<b>Slabs</b>	211416
<b>User</b>	179556
<b>Reclaimable</b>	923260
<b>Caches</b>	316160
<b>Clean</b>	310120
<b>Writeback</b>	0
<b>Dirty</b>	6040
<b>Buffers</b>	607100
<b>Free</b>	692184
<b>Active</b>	284704
<b>Inactive</b>	818112
<b>Low</b>	893544
<b>Used</b>	885184
<b>Free</b>	8360
<b>High</b>	1178152
<b>Used</b>	494328
<b>Free</b>	683824

**Figure 3-25. Physical Memory Text Pane**

The information displayed in this area is as follows. All values are in KB.

The total memory of the system is subdivided in three different, orthogonal, ways:

- Usage
- Recency
- Low vs. High Memory



## Subdivision by Usage

### Total

The total memory on the system in KB.

### Used

The total memory in KB allocated by the kernel or user processes but not including reclaimable memory. This category is further subdivided as follows:

**Kernel** shows the total amount of memory allocated by the kernel. **Other** indicates memory allocated by the kernel for page tables and other internal data stored in whole pages. **Slabs** indicates memory allocated by the kernel for internal data structures.

**User** shows the total amount of memory allocated by user processes, but not including reclaimable memory.

### Reclaimable

The total memory in KB allocated by the kernel but easily discardable and therefore available to the kernel or user processes if necessary:

**Caches** indicates total memory used for cached copies of pages from disk files: **Clean** shows memory used for cached pages which remain unmodified, **Write-back** shows memory used for cached pages which have been modified and are in the process of being written back to disk, and **Dirty** shows the memory used for cached pages which have been modified but have not yet been written back to disk.

**Buffers** indicates memory for pages intended to be written to disk but which have not yet been written. This differs from Dirty in that the pages were not originally cached. For example, calls to `write(2)` may create such pages.

### Free

The total memory in KB that is not being used.

## Subdivision by Recency

This subdivision includes **Total**, **Kernel** (and its further subdivisions), **Free**, and also:

### Active

User, cache or buffer memory that has been used recently and therefore is unlikely to be reclaimed when memory is needed.

### Inactive

User, cache or buffer memory that has been used less recently and therefore is likely to be reclaimed when memory is needed.

## Subdivision by Low vs. High Memory

### Low

Total low memory, which is usable for any purpose by either user processes or the kernel. **Used** indicates low memory including reclaimable memory that is allocated. **Free** indicates low memory not allocated.

### High

Total high memory, which is usable for user processes or the kernel's page cache but not for most kernel data structures. **Used** indicates high memory that is allocated for user processes, page tables or buffers, including reclaimable memory. **Free** indicates high memory not allocated for any purpose.

## Physical Memory Bar Graph Pane

The Physical Memory Bar Graph pane provides individual bar graphs detailing the allocation of physical memory on the system.

The following illustrates the Physical Memory Bar Graph pane:

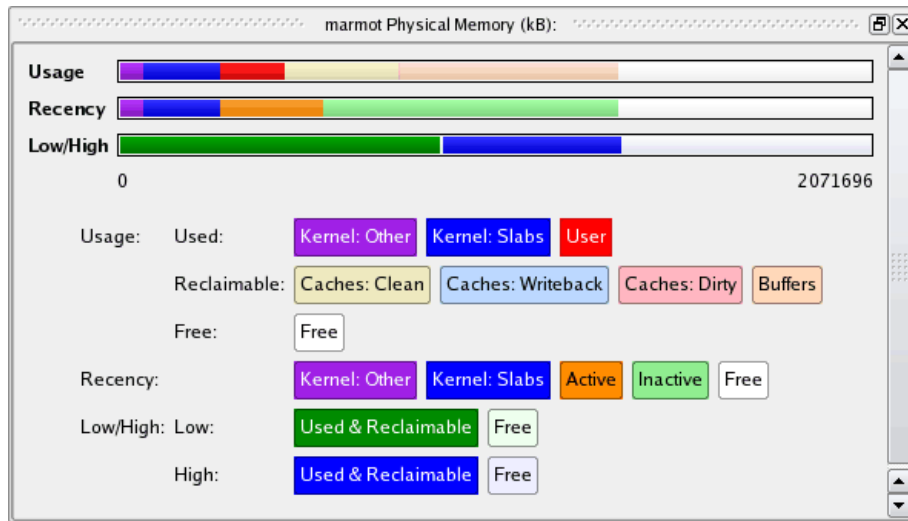


Figure 3-26. Physical Memory Bar Graph Pane

The allocation of physical memory is displayed as horizontal lines. See “Physical Memory Text Pane” on page 3-30 for definitions of the metrics. One bar is displayed for each method of subdivision. In the Usage bar, dark colors are used to represent **Used** memory, while light colors are used to represent **Reclaimable** memory, and white is used for **Free** memory.

## Physical Memory Line Graph Pane

The Physical Memory Line Graph pane provides individual line graphs detailing the allocation of physical memory.

The following illustrates the Physical Memory Line Graph pane:

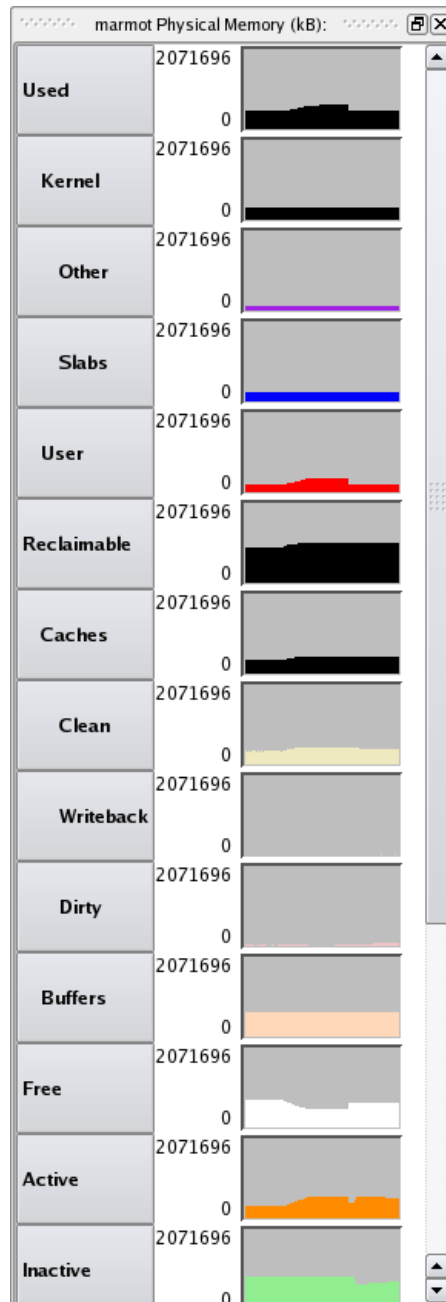


Figure 3-27. Physical Memory Line Graph Pane

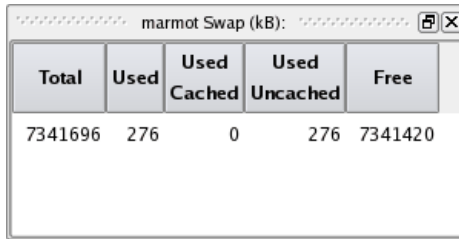
The allocation of physical memory is displayed as a vertical line in each graph. See “Physical Memory Text Pane” on page 3-30 for definitions of the metrics.

## Swap Panel

The Swap panel is a system status panel (see “System Status Panel” on page 3-1). Each pane displays the swap space allocation at the time of the last refresh.

## Swap Text Pane

The following illustrates the Swap Memory Text pane:



Total	Used	Used Cached	Used Uncached	Free
7341696	276	0	276	7341420

**Figure 3-28. Swap Memory Text Pane**

The information displayed in this area includes:

### **Total**

The total swap space on the system in KB.

### **Used**

The total swap space in use on the system. This is further subdivided as Cached and Uncached.

### **Used: Cached**

The total swap space that, after having been swapped out, has been swapped back in and therefore is in both physical memory and swap space. If additional swapping becomes necessary in the future, the memory characterized in this way can be swapped for nearly no cost, because the swap area already contains a copy of them memory. Conversely, if swap space becomes low, this swap space can be reclaimed because its content already has been swapped back into physical memory.

**Used: Uncached**

The total swap space that has not been swapped back in and therefore exists only in swap space.

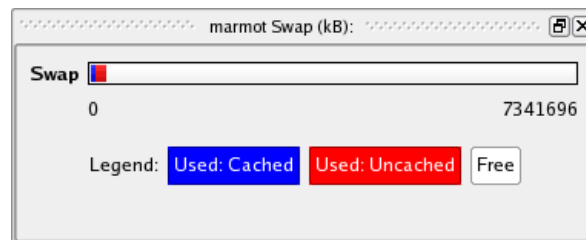
**Free**

The total swap space that is not being used.

## Swap Bar Graph Pane

The Swap Memory Bar Graph pane provides a bar graph illustrating the used and free swap space on the system. The total width of the bar represents the total amount of swap space on the system. The metrics are described in “Swap Text Pane” on page 3-34.

The following illustrates the Swap Memory Bar Graph pane:

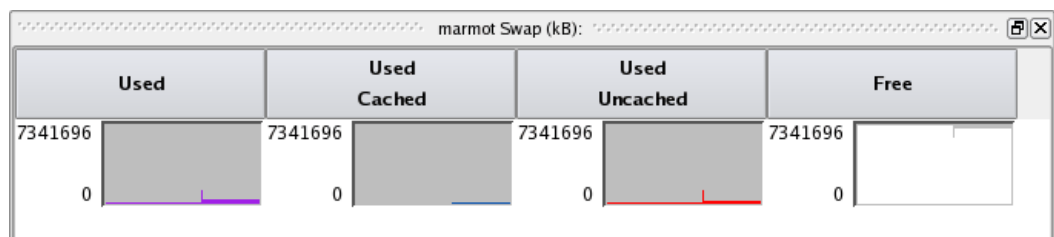


**Figure 3-29. Swap Memory Bar Graph Pane**

## Swap Line Graph Pane

The Swap Memory Line Graph pane provides individual line graphs detailing the swap space.

The following illustrates the Swap Memory Line Graph pane:



**Figure 3-30. Swap Memory Line Graph Pane**

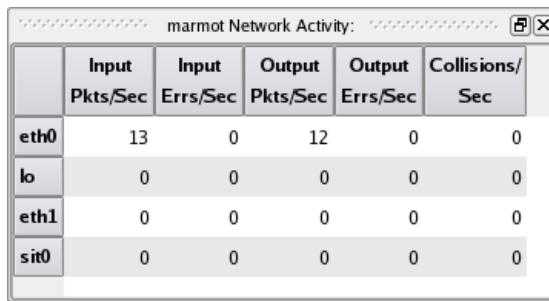
The allocation of swap memory is displayed as a vertical line in each graph. The metrics are described in “Swap Text Pane” on page 3-34.

## Network Activity Panel

The Network Activity panel is a system status panel (see “System Status Panel” on page 3-1). Each pane displays statistics related to network operations that occurred over the period of time defined by the refresh interval. Statistics for individual network devices are displayed.

## Network Activity Text Pane

The following illustrates the Network Activity Text pane:



	Input Pkts/Sec	Input Errs/Sec	Output Pkts/Sec	Output Errs/Sec	Collisions/ Sec
eth0	13	0	12	0	0
lo	0	0	0	0	0
eth1	0	0	0	0	0
sit0	0	0	0	0	0

Figure 3-31. Network Activity Text Pane

The information displayed in this area includes:

### Interface

The name of the network interface is displayed in the left-most column.

### Input Pkts/Sec

The number of packets per second transferred on behalf of input operations is displayed in this column.

### Input Errs/Sec

The number of errors per second that occurred during input operations is displayed in this column.

### Output Pkts/Sec

The number of packets per second transferred on behalf of output operations is displayed in this column.

### Output Errs/Sec

The number of errors per second that occurred during output operations is displayed in this column.

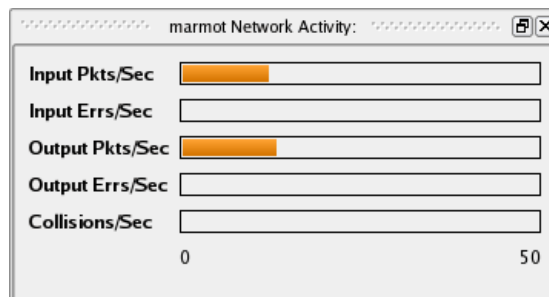
### Collisions/Sec

The number of network collisions per second that occurred is displayed in this column.

## Network Activity Bar Graph Pane

The Network Activity Bar Graph pane provides individual bar graphs detailing metrics related to network operations.

The following illustrates the Network Activity Bar Graph pane:



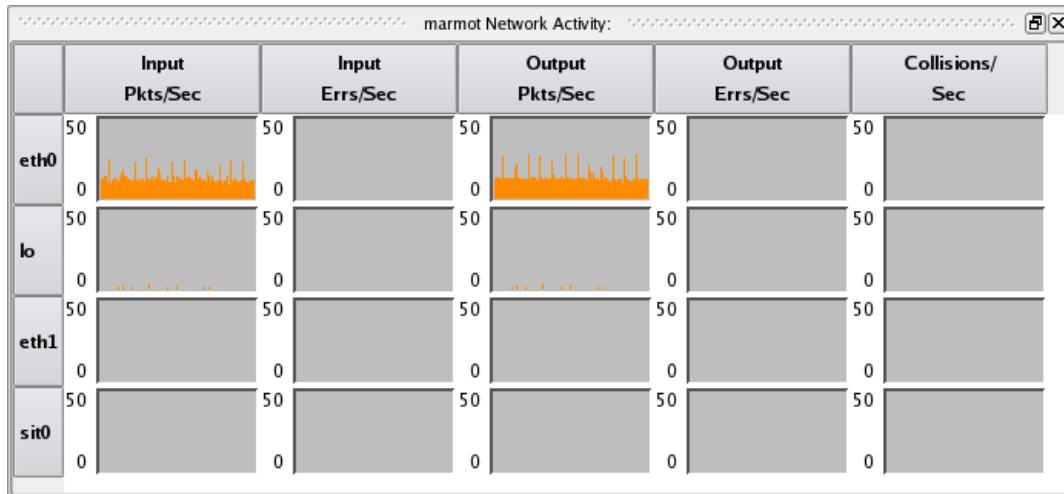
**Figure 3-32. Network Activity Bar Graph Pane**

Individual bar graphs are shown for each metric. The value of the metric is displayed horizontally in each graph. See “Network Activity Text Pane” on page 3-36 for definitions of these metrics.

## Network Activity Line Graphs

The Network Activity Line Graph pane provides individual line graphs detailing metrics related to network operations.

The following illustrates the Network Activity Line Graph pane:



**Figure 3-33. Network Activity Line Graph Pane**

Individual line graphs are shown for each network device for each metric. The value of the metric is displayed vertically in each graph. See “Network Activity Text Pane” on page 3-36 for definitions of these metrics.

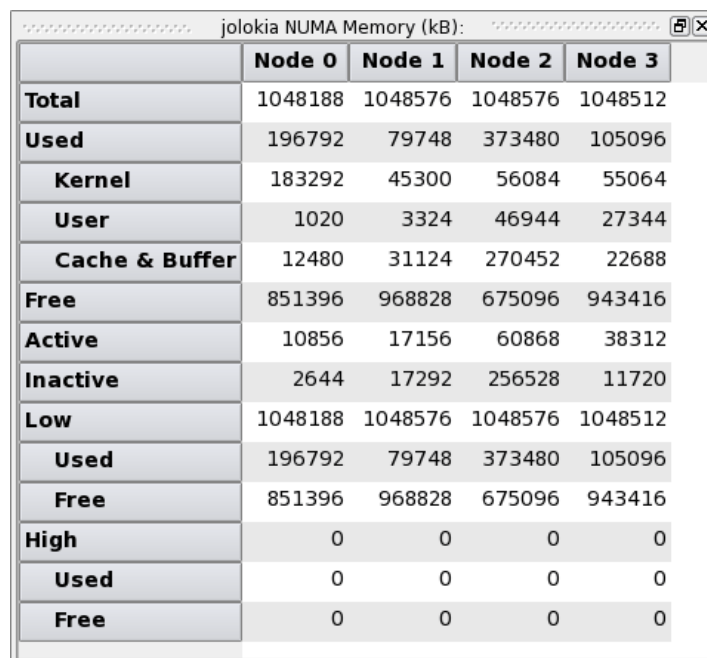


## NUMA Panel

The NUMA panel is a system status panel (see “System Status Panel” on page 3-1). Each pane displays the allocation of memory on each NUMA node at the time of the last refresh.

## NUMA Text Pane

The following illustrates the NUMA Text pane:



	Node 0	Node 1	Node 2	Node 3
<b>Total</b>	1048188	1048576	1048576	1048512
<b>Used</b>	196792	79748	373480	105096
<b>Kernel</b>	183292	45300	56084	55064
<b>User</b>	1020	3324	46944	27344
<b>Cache &amp; Buffer</b>	12480	31124	270452	22688
<b>Free</b>	851396	968828	675096	943416
<b>Active</b>	10856	17156	60868	38312
<b>Inactive</b>	2644	17292	256528	11720
<b>Low</b>	1048188	1048576	1048576	1048512
<b>Used</b>	196792	79748	373480	105096
<b>Free</b>	851396	968828	675096	943416
<b>High</b>	0	0	0	0
<b>Used</b>	0	0	0	0
<b>Free</b>	0	0	0	0

**Figure 3-34. NUMA Text Pane**

The total memory for each NUMA node is subdivided in two different, orthogonal, ways:

- Usage
- Recency
- Low vs. High Memory

## Subdivision by Usage

### Total

The total memory in the NUMA node.

### **Used**

The total memory in KB from the NUMA node allocated by the kernel or user processes, including reclaimable memory. This is further subdivided into **Kernel**, **User** and **Cache & Buffer** memory.

### **Kernel**

The total memory in KB from the NUMA node allocated by the kernel.

### **User**

The total memory in KB from the NUMA node allocated as user memory.

### **Cache & Buffer**

The total memory in KB allocated by the kernel but easily discardable and therefore available to the kernel or user processes if necessary. Cache memory is for cached copies of pages from disk files. Buffer memory is for pages intended to be written to disk but which have not yet been written. For example, calls to `write(2)` may create buffer memory.

### **Free**

Total unused memory in KB available on the NUMA node.

## **Subdivision by Recency**

This subdivision includes Total, Kernel, Free and the following:

### **Active**

User, cache or buffer memory that has been used recently and therefore is unlikely to be reclaimed when memory is needed.

### **Inactive**

User, cache or buffer memory that has been used less recently and therefore is likely to be reclaimed when memory is needed.

## **Subdivision by Low vs. High Memory**

### **Total**

The total memory in the NUMA node.

## Low

Total low memory, which is usable for any purpose by either user processes or the kernel. **Used** indicates low memory allocated, including reclaimable memory. **Free** indicates low memory that is not allocated for any purpose.

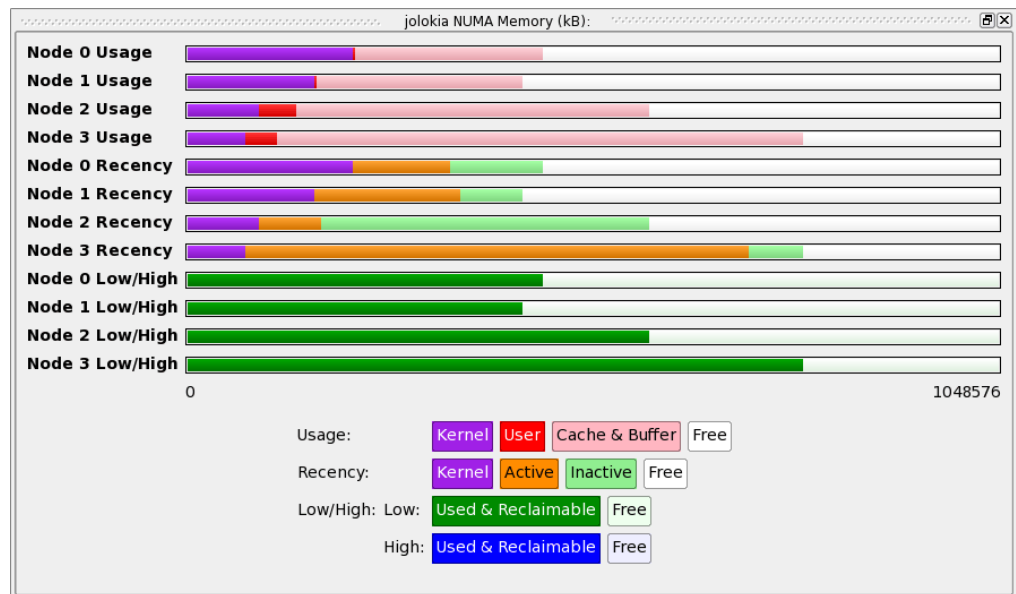
## High

Total high memory, usable for user processes or the kernel's page cache but not for most kernel data structures. **Used** indicates high memory allocated, including reclaimable memory. **Free** indicates high memory that is not allocated for any purpose.

## NUMA Bar Graph Pane

The NUMA Bar Graph pane provides individual bar graphs detailing metrics related to memory allocation on NUMA systems.

The following illustrates the NUMA Bar Graph pane:



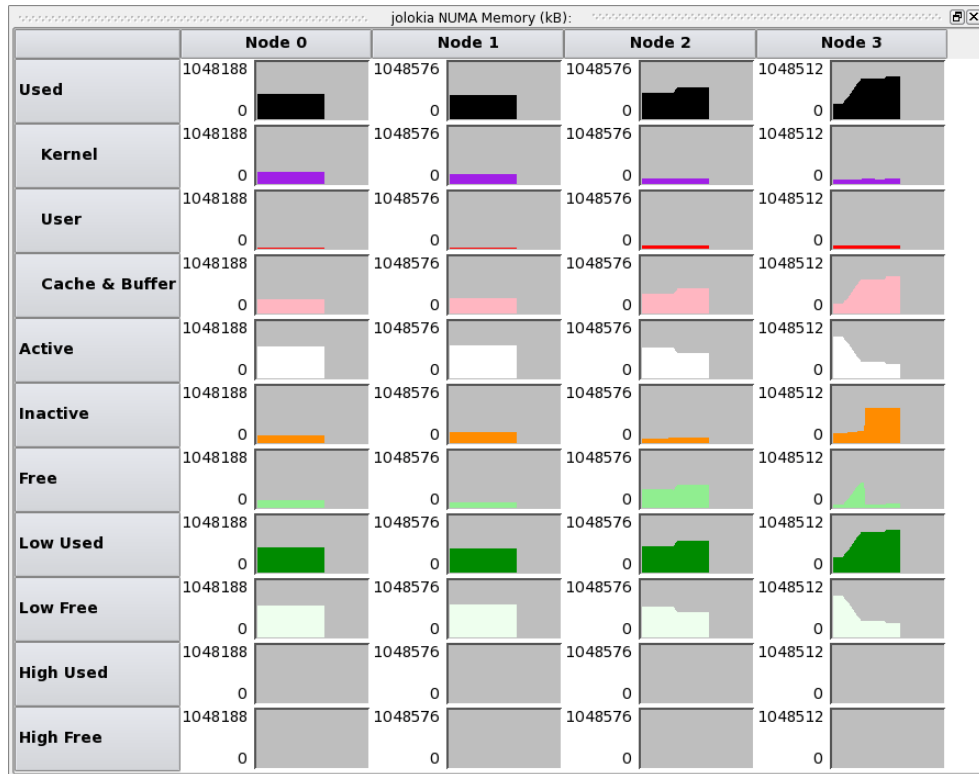
**Figure 3-35. NUMA Bar Graph Pane**

The allocation of memory for each NUMA node is displayed as horizontal lines. One bar is displayed for each method of subdivision for each NUMA node. See “NUMA Text Pane” on page 3-39 for definitions of the subdivisions and metrics.

## NUMA Line Graph Pane

The NUMA Line Graph pane provides individual line graphs detailing metrics related to memory allocation on NUMA systems.

The following illustrates the NUMA Line Graph pane:



**Figure 3-36. NUMA Line Graph Pane**

Individual line graphs are shown for each node's memory allocation and for each metric. See "NUMA Text Pane" on page 3-39 for definitions of these metrics.

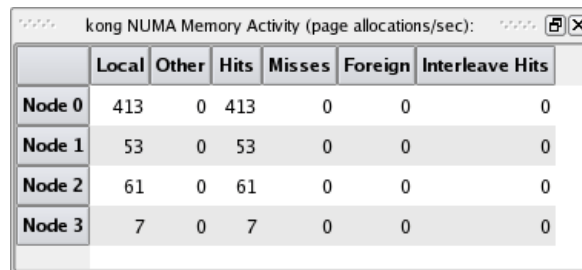
## NUMA Activity Panel

The NUMA Activity panel is a system status panel (see "System Status Panel" on page 3-1). Each pane displays statistics related to memory activity on NUMA systems that

occurred over the period of time defined by the refresh interval. Statistics for individual nodes are displayed.

## NUMA Activity Text Pane

The following illustrates the NUMA Activity Text pane:



	Local	Other	Hits	Misses	Foreign	Interleave Hits
<b>Node 0</b>	413	0	413	0	0	0
<b>Node 1</b>	53	0	53	0	0	0
<b>Node 2</b>	61	0	61	0	0	0
<b>Node 3</b>	7	0	7	0	0	0

**Figure 3-37. NUMA Activity Text Pane**

The information displayed in this area includes:

### Local

Pages allocated from this node by a CPU associated with this node.

### Other

Pages allocated from this node by a CPU not associated with this node.

### Hits

Pages allocated from this node when this was the preferred node in the allocation's memory policy.

### Misses

Pages allocated from this node when this was not the preferred node in the allocation's memory policy.

### Foreign

Pages allocated from another node when this was the preferred node in the allocation's memory policy.

### Interleaved Hits

Pages allocated from this node with interleaved memory policy when this was the appropriate interleaved node.

## NUMA Activity Bar Graph Pane

The NUMA Activity Bar Graph pane provides individual bar graphs detailing metrics related to memory activity on NUMA systems.

The following illustrates the NUMA Activity Bar Graph pane:

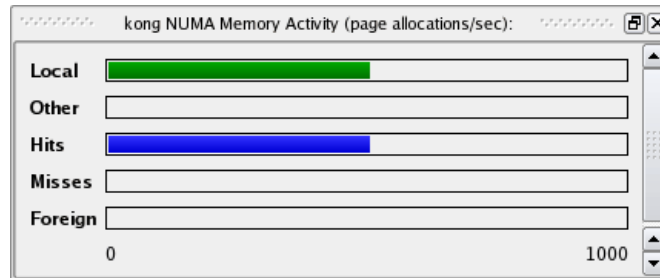


Figure 3-38. NUMA Activity Bar Graph Pane

Individual bar graphs are shown for each node's memory activity. See "NUMA Activity Text Pane" on page 3-43 for definitions of these metrics.

## NUMA Activity Line Graph Pane

The NUMA Activity Line Graph pane provides individual line graphs detailing metrics related to memory activity on NUMA systems.

The following illustrates the NUMA Activity Line Graph pane:

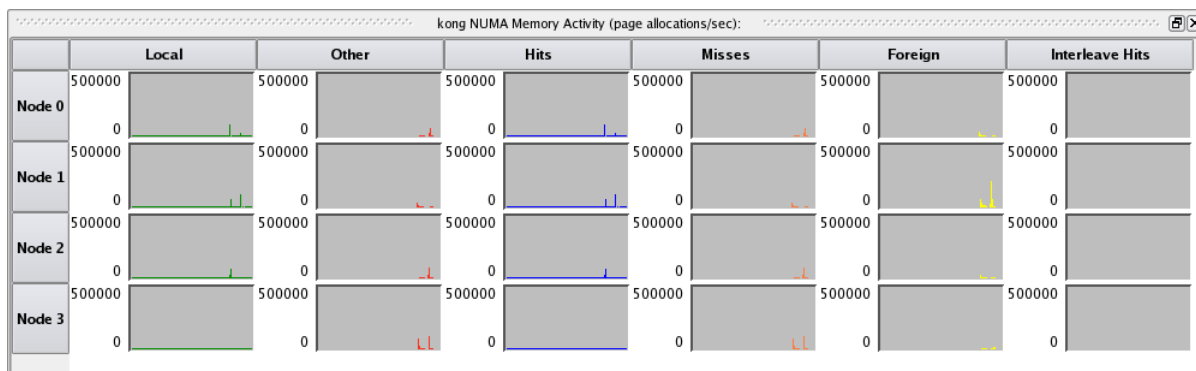


Figure 3-39. NUMA Activity Line Graph Pane

Individual line graphs are shown for each node's memory activity. See "NUMA Text Pane" on page 3-39 for definitions of these metrics.

## NUMA Configuration Panel

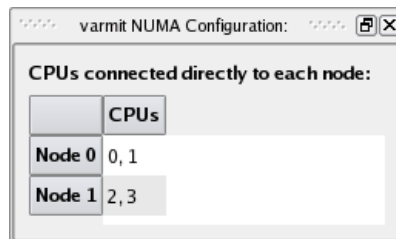
The NUMA Configuration panel provides two panes that display configuration statistics for each node on NUMA systems: the CPUs pane and the Distance pane.

Because this panel provides statistics related to the NUMA configuration of the system, the statistics do not change with refresh intervals. Changes appear only when the system is reconfigured.

### NUMA Configuration CPUs Pane

This pane shows the CPUs connected to each NUMA node.

The following illustrates the NUMA Configuration CPUs pane:



The screenshot shows a window titled "varmit NUMA Configuration:" with a sub-header "CPUs connected directly to each node:". Below this is a table with two columns: "Node" and "CPUs".

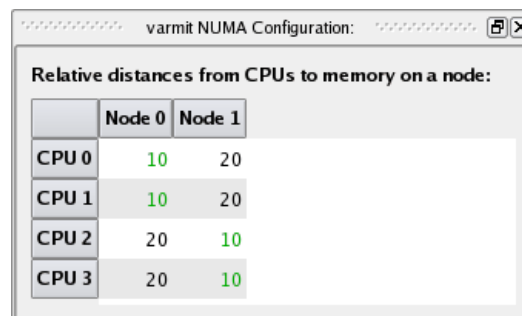
	CPUs
Node 0	0, 1
Node 1	2, 3

Figure 3-40. NUMA Configuration Text Pane

### NUMA Configuration Distance Pane

The NUMA Configuration Distance pane provides the relative distances from CPUs to memory on NUMA systems.

The following illustrates the NUMA Configuration Distance pane:



The screenshot shows a window titled "varmit NUMA Configuration:" with a sub-header "Relative distances from CPUs to memory on a node:". Below this is a table with three columns: "CPU", "Node 0", and "Node 1".

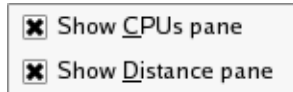
	Node 0	Node 1
CPU 0	10	20
CPU 1	10	20
CPU 2	20	10
CPU 3	20	10

Figure 3-41. NUMA Configuration Distance Pane

Values displayed in green show minimum distances for each CPU. Generally, that means that the a CPU is connected directly to the node if the cell in the table for that pair is green.

## NUMA Configuration Context Menu

While positioned in the NUMA Configuration panel, right-clicking displays the NUMA Configuration context menu.



**Figure 3-42. NUMA Configuration Context Menu**

The following paragraphs describe the menu items in detail:

### Show CPUs pane

Mnemonic: C

This menu item toggles the visibility of the NUMA Configuration CPUs pane within the panel.

### Show Distance pane

Mnemonic: D

This menu item toggles the visibility of the NUMA Configuration Distance pane within the panel.

## Process List Panel

The purpose of the Process List panel is to provide detailed descriptions of individual processes and threads.

The process list can be organized with a parent-child hierarchy or can be a flat list of processes. In addition, either the hierarchical or flat structures can be organized on a per-user-ID basis, if desired. These selections can be made from the Process List context menu (see “Process List Context Menu” on page 3-49).



The following illustrates the Process List panel:

PID	State	Size	Data	%CPU	CPU Time	User	System	CPU	Affinity	Nice	RPri	CL	Command
Users													
													ldap
													lp
													mail
													messageb
													nobody
													ntp
													root
													sms
													todd
4397	Waiting	20020	10516	0.0	14.96	14.44	0.52	0	all	0	0	OT	emacs
6326	Waiting	4716	668	0.0	0.24	0.20	0.04	0	all	0	0	OT	ksh
4279	Waiting	1368	108	0.0	0.18	0.13	0.05	0	all	0	0	OT	prefix
4434	Waiting	1728	248	0.0	0.00	0.00	0.00	0	all	0	0	OT	be-polite
4278	Waiting	1512	252	0.0	0.19	0.15	0.04	0	all	0	0	OT	watch
4329	Waiting	3884	556	0.0	0.05	0.02	0.03	0	all	0	0	OT	xpmxload
4330	Waiting	6180	1372	0.0	50.34	30.92	19.42	0	all	0	0	OT	xpmxterm
4707	Waiting	9364	4556	0.0	0.11	0.09	0.02	0	all	0	0	OT	xpmxterm
4720	Waiting	4880	832	0.0	0.10	0.08	0.02	0	all	0	0	OT	ksh
4726	Waiting	9360	4552	0.0	0.09	0.06	0.03	0	all	0	0	OT	xpmxterm
4734	Waiting	13648	8840	0.0	0.15	0.08	0.07	0	all	0	0	OT	xpmxterm
4773	Waiting	12156	7348	0.0	0.10	0.07	0.03	0	all	0	0	OT	xpmxterm

**Figure 3-43. Process List Panel**

The columns of fields that are displayed are controlled from the **Process Fields** menu from the **Display Fields** item of the **Process List** context menu (see “Process Fields Menu” on page 3-73). The command name of the process is displayed in short form by default, although this also can be changed from the context menu. The long form of the command displays when the mouse hovers over the short form of the command name.

Information in the process line is automatically updated at a selectable interval. The interval can be changed using the **Preferences** dialog from the **Preferences...** item from the **File** menu. Immediate refresh of information is available using the **Refresh** tool icon or through the **Process List** context menu.

Some operations in the panel operate on processes that are selected in the process list. Selection of individual processes is done using the left mouse button, which toggles whether the associated process is selected or deselected. Multiple selection is done by selecting individual processes with the shift and/or control keys. Selecting or deselecting a multi-threaded process selects or deselects all its threads.

You can use the **Process List** context menu (see “Process List Context Menu” on page 3-49) to filter the processes shown. When the panel is filtered, a status bar appears at the bottom of the panel to describe the filter, shown in Figure 3-44. The status bar also contains a **Change Filter** button to bring up the **Filter Processes** dialog (see “Filter Processes Dialog” on page 3-52) and a **Clear Filter** button to remove the filter.



Figure 3-44. Process List Filter Bar

### Find Bar

You can search for a process with the Find... item in the Process List context menu (see “Process List Context Menu” on page 3-49) or by using Ctrl+F. The find bar appears at the bottom of the panel.



Figure 3-45. Process List Find Bar

Enter a regular expression in the text entry field; for example, a program name. If a matching line is found, the line is highlighted. If no match is found, the background color of the text entry field changes.

Note that hidden fields are searched. For example, if you search for your username, processes you own will match even though the Username field is not displayed.

Press Enter or click on Next to go to the next match. Click on Previous to go to the previous match. The Match case check box indicates whether the search should be case-sensitive.

Press Esc or click on the close button to hide the find bar.

## Process List Drag and Drop Operations

Individual processes or groups of processes can be dragged onto various destination panels and drop targets.

To drag an individual process, click anywhere on the row describing the process and drag the pointer to the destination area and release the mouse button. Similarly, you can drag all the processes currently associated with a user by clicking on the user line and dragging the pointer to the destination area.

The Process List panel supports the following drag and drop operations:

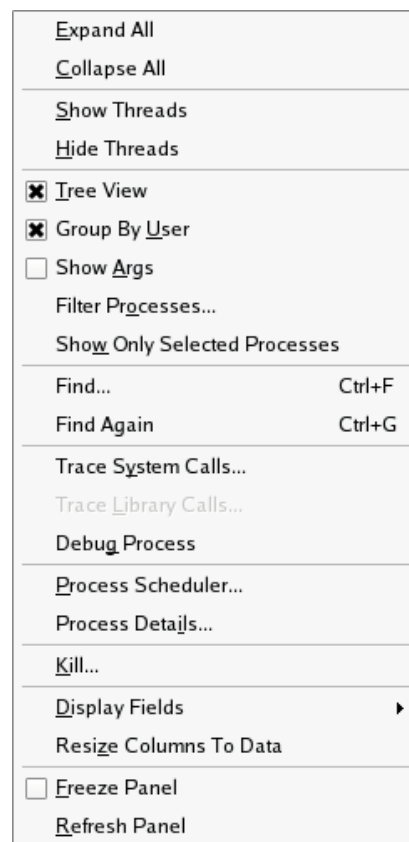
- Dragging a process or user line onto a CPU line in the CPU Shielding and Binding panel binds the processes to the corresponding CPU.
- Dragging a process onto the Process Scheduler dialog changes the dialog to refer to that process.

- Dragging a process or user line onto the Kill tool icon causes the processes to be killed with a **SIGKILL** signal.
- Dragging a process or user line onto the Unbind tool icon causes the processes to be unbound from any CPUs.

## Process List Context Menu

To display the Process List context menu, right-click while positioned in the Process List panel.

The following illustrates the Process List context menu:



**Figure 3-46. Process List Context Menu**

The following paragraphs describe the options on the menu in more detail:

### Expand All

Mnemonic: E

This menu item expands the selected item and all its interior items.

### **Collapse All**

Mnemonic: C

This menu item collapses the selected item and all its interior items.

### **Show Threads**

Mnemonic: S

This menu item causes the list of threads to be expanded for the selected processes.

### **Hide Threads**

Mnemonic: H

This menu item causes the list of threads for the selected processes to be hidden, leaving a single process line that summarizes all the threads.

### **Tree View**

Mnemonic: T

This menu item, when selected, organizes the processes into a hierarchy according to their parent-child relationships. When not selected, the processes are organized as a flat list.

### **Group by User**

Mnemonic: U

This menu item, when selected, organizes processes by user, with top-level items for each user with running processes and all processes placed beneath their user, with the organization under each user controlled by the **Tree View** setting. When not selected, processes are organized merely according to the **Tree View** setting.

### **Show Args**

Mnemonic: A

This menu item, when selected, causes each process to be described in a long form including arguments in the Command column. If not selected, the short form is displayed in the Command column. In either case, the long form is displayed in a tooltip if the mouse hovers over the Command column.

### **Filter Processes...**

Mnemonic: O

This menu item displays the **Filter Processes** dialog, which allows you to select the processes shown in the **Process List** panel. See “Filter Processes Dialog” on page 3-52.

### Show Only Selected Processes

Mnemonic: W

This menu item changes the display to show only the processes you have selected. If you selected user lines, the display includes only the processes owned by those users. If you selected process lines, the display includes only those processes. If you selected both user lines and process lines, the process lines are ignored.

### Find...

Accelerator: Ctrl+F

This menu item displays the find bar (see “Find Bar” on page 3-48) at the bottom of the panel. Enter a regular expression in the text entry area to initiate a search.

### Find Again

Accelerator: Ctrl+G

This menu item searches for the next match of the regular expression previously entered in the find bar (see “Find Bar” on page 3-48).

### Trace System Calls...

Mnemonic: Y

This menu item displays **strace** output for the selected process. See “Trace System Calls Window” on page 3-54.

### Trace Library Calls...

Mnemonic: L

This menu item displays **ltrace** output for the selected process. See “Trace Library Calls Window” on page 3-54.

### Debug Process

Mnemonic: G

This menu item launches the NightView debugger to debug the selected process.

### Process Scheduler...

Mnemonic: P

This menu item displays the **Process Scheduler** dialog where the scheduling properties of a selected process can be viewed or modified. See “Process Scheduler Dialog” on page 3-55 for details.

### **Process Details...**

Mnemonic: I

This menu item displays the **Process Details** dialog which shows memory usage, detailed information about memory pages, file descriptors, signal status, capabilities and environment variables for the selected process. See “Process Details Window” on page 3-59 for details.

### **Kill...**

Mnemonic: K

This menu item displays the **Kill Process** dialog where you may select a signal (e.g. **SIGKILL**) to be sent to the processes.

### **Display Fields**

Mnemonic: D

This menu item displays the **Process Fields** menu that allows you to select the fields (columns) that are visible in the **Process List** panel. See “Process Fields Menu” on page 3-73 for definitions of these metrics.

### **Resize Columns to Data**

Mnemonic: Z

This menu item resizes the columns to fit the data visible.

### **Freeze Panel**

Mnemonic: F

This menu item toggles the **Freeze** setting for this panel. When frozen, this panel is not refreshed automatically.

### **Refresh Panel**

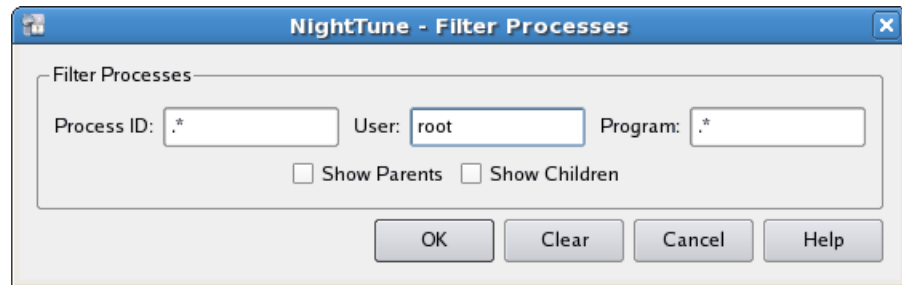
Mnemonic: R

This menu item causes the panel to refresh with the latest list of users and processes, regardless of the **Freeze** setting.

## **Filter Processes Dialog**

This dialog pops up when you click on **Filter Processes...** in the **Process List** context menu, or when you click on the **Change Filter** button in the **Process List** panel (the button is visible only when processes are filtered).

The following illustrates the Filter Processes dialog.



**Figure 3-47. Filter Processes Dialog**

The following fields describe which processes you want shown in the **Process List** panel. A process must match all the fields to be visible.

**Process ID:**

Enter a regular expression to describe the process IDs. For example, enter a single process ID.

**User:**

Enter a regular expression to describe the user names. For example, enter your user name.

**Program:**

Enter a regular expression to describe the program names. For example, enter “sh\$”.

**Show Parents  
Show Children**

These check boxes indicate whether to display the parents and children of processes that match the filter.

The following buttons control the dialog.

**OK**

Clicking **OK** applies the filter to the **Process List** panel and closes the dialog.

**Clear**

Clicking **Clear** sets the filter controls to show all processes. If you want to show all processes in the **Process List** panel, click **Clear** and then click **OK**.

**Cancel**

Clicking **Cancel** closes the dialog without making any changes.

## Help

Clicking Help presents this section of the manual in the online help viewer.

## Trace System Calls Window

The Trace System Calls window is displayed from the Trace System Calls option of the Process List context menu. This window shows the output of an `strace(1)` of the selected process. Buttons provide the ability to save trace output to a file, print the screen and search trace output. The output also may be paused and resumed.

The following illustrates the Trace System Call window:

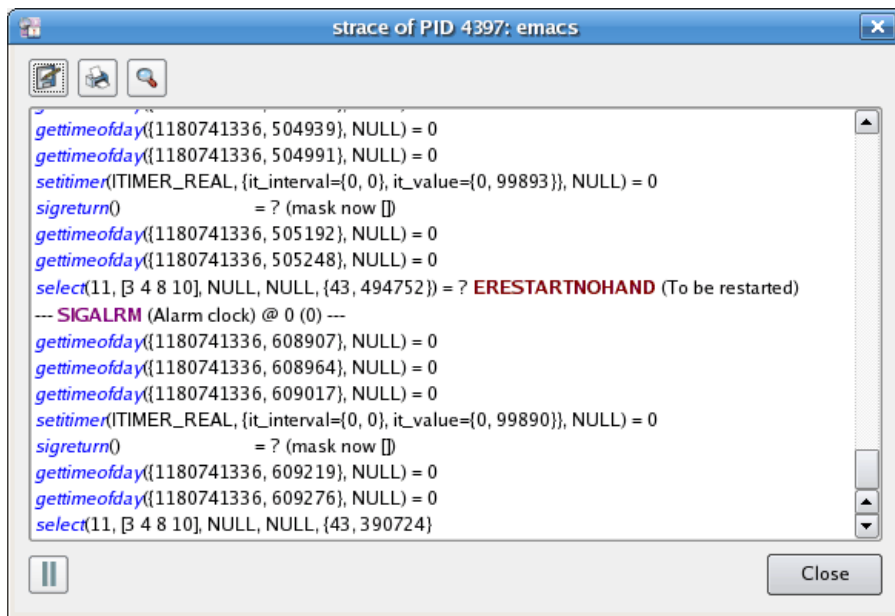


Figure 3-48. Process List Trace System Calls Window

## Trace Library Calls Window

The Trace Library Calls window is displayed from the Trace Library Calls option of the Process List context menu.

This window shows the output of an `ltrace(1)` of the selected process. Buttons provide the ability to save trace output to a file, print the screen and search trace output. The output also may be paused and resumed.

The following illustrates the Trace Library Calls window:



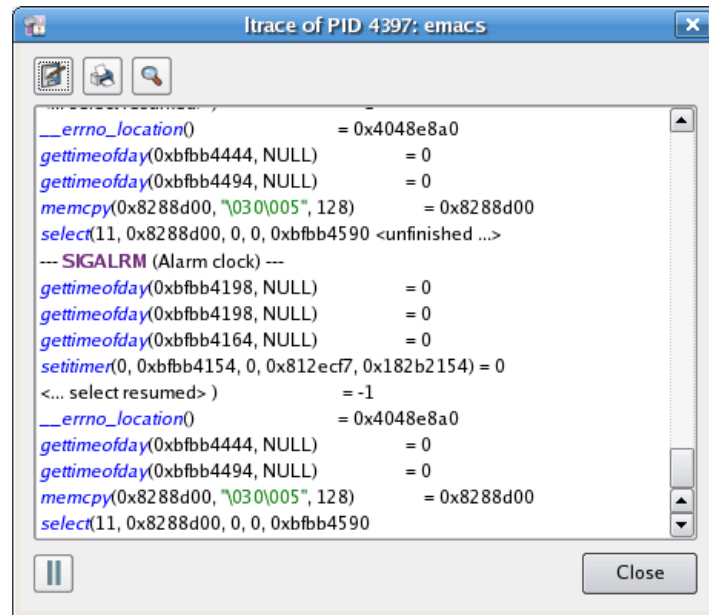


Figure 3-49. Trace Library Calls Window

## Debug Process

Opens the NightView Source-Level Debugger and arranges for it to attach to the selected process. NightView is a graphical source-level debugging and monitoring tool specifically designed for real-time applications. NightView can monitor, debug, and patch multiple real-time processes running on multiple processors with minimal intrusion.

See also:

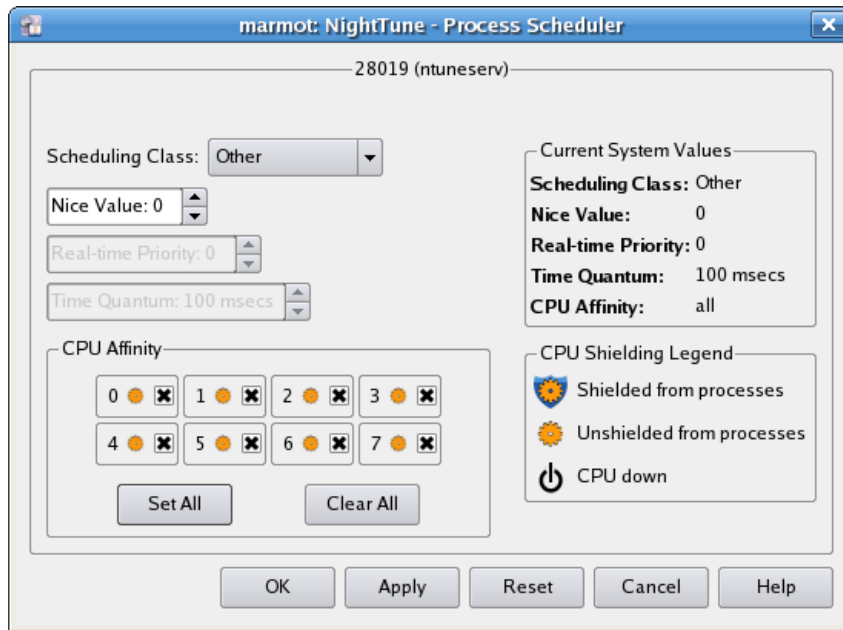
- *NightView RT User's Guide*

## Process Scheduler Dialog

You can use the Process Scheduler dialog to alter the scheduling attributes and CPU affinity for any process for which you have appropriate privileges.

The Process Scheduler dialog is displayed when the process is selected in the Process List panel and Processor Scheduler is selected from the Process List context menu.

The following illustrates the Process Scheduler dialog:



**Figure 3-50. Process Scheduler Dialog**

Descriptions of the text fields and controls contained in the Process Scheduler dialog follow:

**Process**

The process ID (PID) and simple name of the process currently referenced by the dialog is displayed at the top of the dialog. You can use this window as a drop target for processes; dropping a process onto the window changes the dialog to refer to that process. You cannot drop multiple processes or a process with more than one thread on the dialog.

**Scheduling Class**

This menu item allows you to select one of the three scheduling classes supported by the operating system:

**Other**

Selects the **SCHED\_OTHER** scheduling policy which is the default universal time-sharing policy used by most processes. The priority of the process is first determined by the **Nice Value**, but then is adjusted by the operating system based on CPU utilization. Processes using this policy will always have a less favorable priority than those in the other two classes.

**First In-First Out**

Selects the **SCHED\_FIFO** scheduling policy. The priority of processes within this policy are static — they are not adjusted by the operating system. Pro-

cesses retain use of the CPU until they block, voluntarily yield the CPU, or are preempted by higher priority processes or interrupts.

### Round Robin

Selects the **SCHED\_RR** scheduling policy. It is very similar to **SCHED\_FIFO**, but every process with this scheduling policy has an associated *time quantum*. In addition to the cases listed for **SCHED\_FIFO**, if a process with policy **SCHED\_RR** has been running for longer than its time quantum, it will be preempted and any other process with the same priority will be allowed to run.

The current scheduling policy for the process is displayed within the **Current System Values** box.

See `sched_setscheduler(2)` for more information on the scheduling classes.

### Nice Value

This text field allows you to specify a nice value for processes scheduled under the **Other** scheduling class. Nice values are inverted: lower numbered values have a more favorable priority than higher numbered values.

The current nice value for the process is displayed within the **Current System Values** box.

The **Nice Value** is only sensitized for processes using the **Other** scheduling class.

### Real-Time Priority

This text field allows you to specify the priority within an associated real-time scheduling policy, either **Round Robin** or **First In-First Out**.

Values for the priority must be in the range 1..99. Higher numbered values have a more favorable priority than lower numbered values.

The current real-time priority for the process is displayed within the **Current System Values** box.

The **Real-Time Priority** is desensitized for processes using the **Other** scheduling class.

### Time Quantum

This text field allows you to specify the duration in milliseconds of the execution time-slice for processes using the **Round Robin** scheduling class. It is not applicable to any other scheduling class.


The current time quantum for the process is displayed within the **Current System Values** box.

The **Time Quantum** is only sensitized for processes using the **Round Robin** scheduling class.

The operating system only supports certain values for the **Time Quantum**; if you enter a value by hand and it is not a supported value, it will be rounded to the nearest supported quantum value.

### CPU Affinity

The **CPU Affinity** area allows you to specify CPUs on which the process is allowed to execute. If a single CPU checkbox is checked, the process is bound to that CPU and the process is displayed in the **Processes** list in the associated CPU box in the **CPU Status** panel.

For each CPU that has been shielded from processes, the 'shielded from processors' icon  is displayed. If a CPU has been shielded from processes, no processes will execute on that CPU unless its affinity includes that CPU and no non-shielded CPUs.

The **Set All** and **Clear All** buttons eliminate having to click on each individual interrupt to set or clear all interrupts quickly.

For more information about CPU affinity and scheduling policies and priorities, refer to the `mpadvise(2)`, `sched_setaffinity(2)`, `sched_setscheduler(2)`, and `sched_setparam(2)` manual pages. Additional information is available in the *RedHawk Linux User's Guide*.

### Current System Values

This area displays the values currently in effect for the process.

### CPU Shielding Legend

This area displays the symbols used to indicate the status of the CPUs on the system: shielded from processes, unshielded from processes, and CPU down.

## Process Scheduling Operations

The process scheduling operations are controlled using the buttons at the bottom of the dialog:

### OK

Clicking **OK** applies any scheduling changes made in the dialog. You may not see a change reflected immediately since the operating system may defer certain operations until the next time the process becomes active. This is particularly likely if a process is being starved of CPU time; in this case try unbinding the process, or binding it to a different CPU, and making the change again. The dialog is closed when the actions are complete.

### Apply

Clicking **Apply** applies any scheduling changes made in the dialog. You may not see a change reflected immediately since the operating system may defer certain operations until the next time the process becomes active. This is particularly likely

if a process is being starved of CPU time; in this case try unbinding the process, or binding it to a different CPU, and making the change again.

### **Reset**

Clicking **Reset** causes the current process scheduling attributes to be reflected in the dialog, discarding any changes that have not yet been applied.

### **Cancel**

Clicking **Cancel** closes the dialog. Any changes that have not been applied are discarded.

### **Help**

Clicking **Help** presents this section of the manual in the online help viewer.

## **Process Details Window**

The **Process Details** window is displayed from the **Process Details** option of the **Process List** context menu. This window contains memory usage, detailed information about memory pages, file descriptors, signal status, capabilities and environment variables for the selected process in separate tabbed sections.

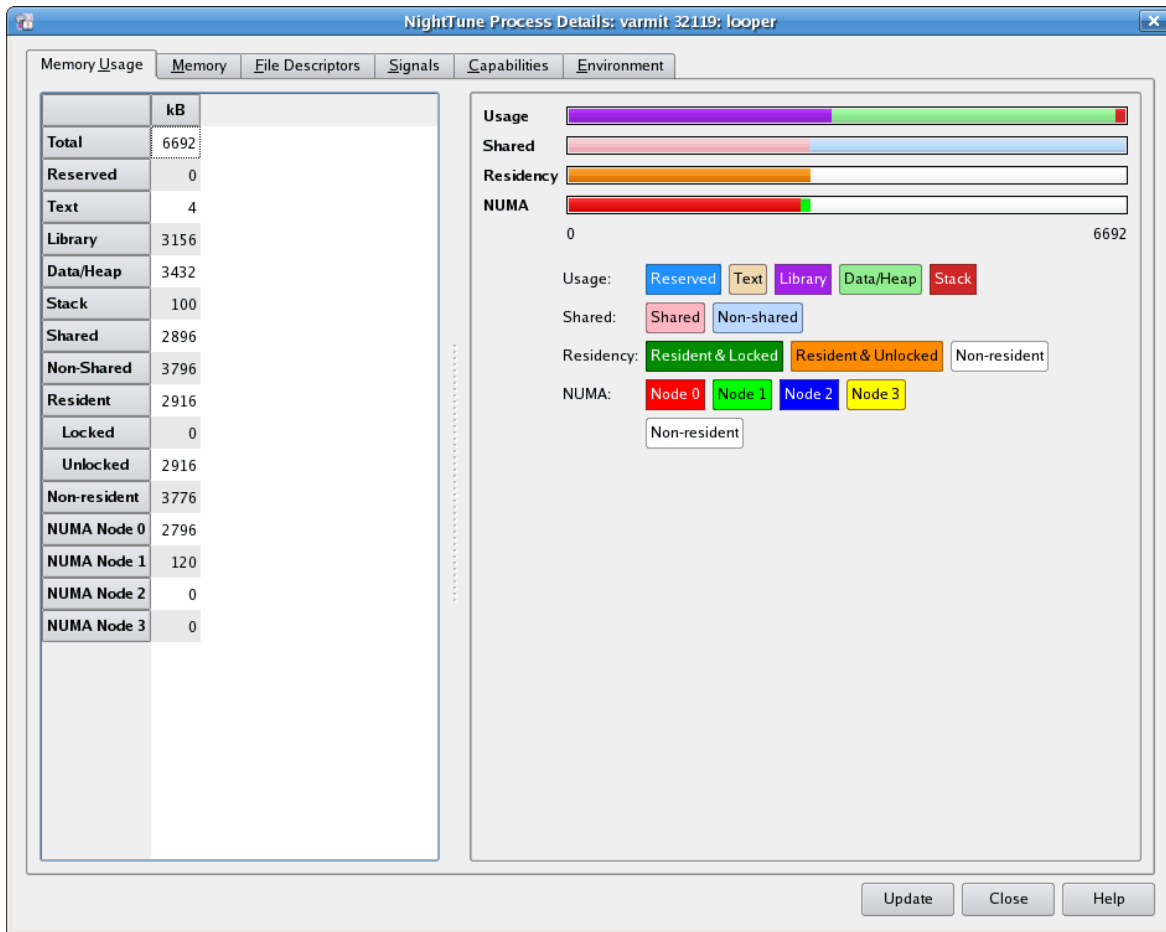
The information displayed in this dialog is a snapshot of the process at one point in time. Initially, it is the snapshot taken when the dialog was created. The user may press the **Update** button to obtain a new snapshot for the process.

The **Close** button may be used to close the dialog.

### **Memory Usage Tab**

The **Process Details Memory Usage** tab provides an overview of memory usage for the process.

The following illustrates the **Memory Usage** tab:



**Figure 3-51. Process Details Memory Usage Tab**

The total memory of the process is subdivided in four different, orthogonal, ways:

- Usage
- Shared / Nonshared
- Residency
- NUMA node

### Subdivision by Usage

#### Total

Total virtual memory allocated for this process.

#### Reserved

Virtual memory reserved for I/O or by the kernel such that swapping of that memory is prohibited.

**Text**

Virtual memory used for executable instructions in the process' main executable.

**Library**

Virtual memory used for executable instructions in any shared libraries used by the process.

**Data/Heap**

Virtual memory used for static or heap data.

**Stack**

Virtual memory used for call stacks.

Subdivision by Shared / Nonshared

**Total**

Total virtual memory allocated for this process.

**Shared**

Virtual memory associated with a file.

**Nonshared**

Virtual memory not associated with a file.

Subdivision by Residency

**Total**

Total virtual memory allocated for this process.

**Resident**

Virtual memory stored in physical memory. This total is further subdivided into memory that is **Locked** in physical memory and that which is **Unlocked**, and therefore could become nonresident in the future.

**Nonresident**

Virtual memory not stored in physical memory, but rather on disk, either in swap space or in a disk file (if mapped).

### Subdivision by NUMA node

One item exists for each NUMA node on the system, with one final item for non-resident memory. The item shows the total virtual memory allocated by the process from each NUMA node, or that is not resident in physical memory.

### Memory Tab

The Process Details Memory tab provides the ability to view information about individual pages in a process' address space. The display can be zoomed and panned to represent the entire address space or as few as four pages. Various methods allow navigation to all areas of the address space.

The following illustrates the Memory tab:

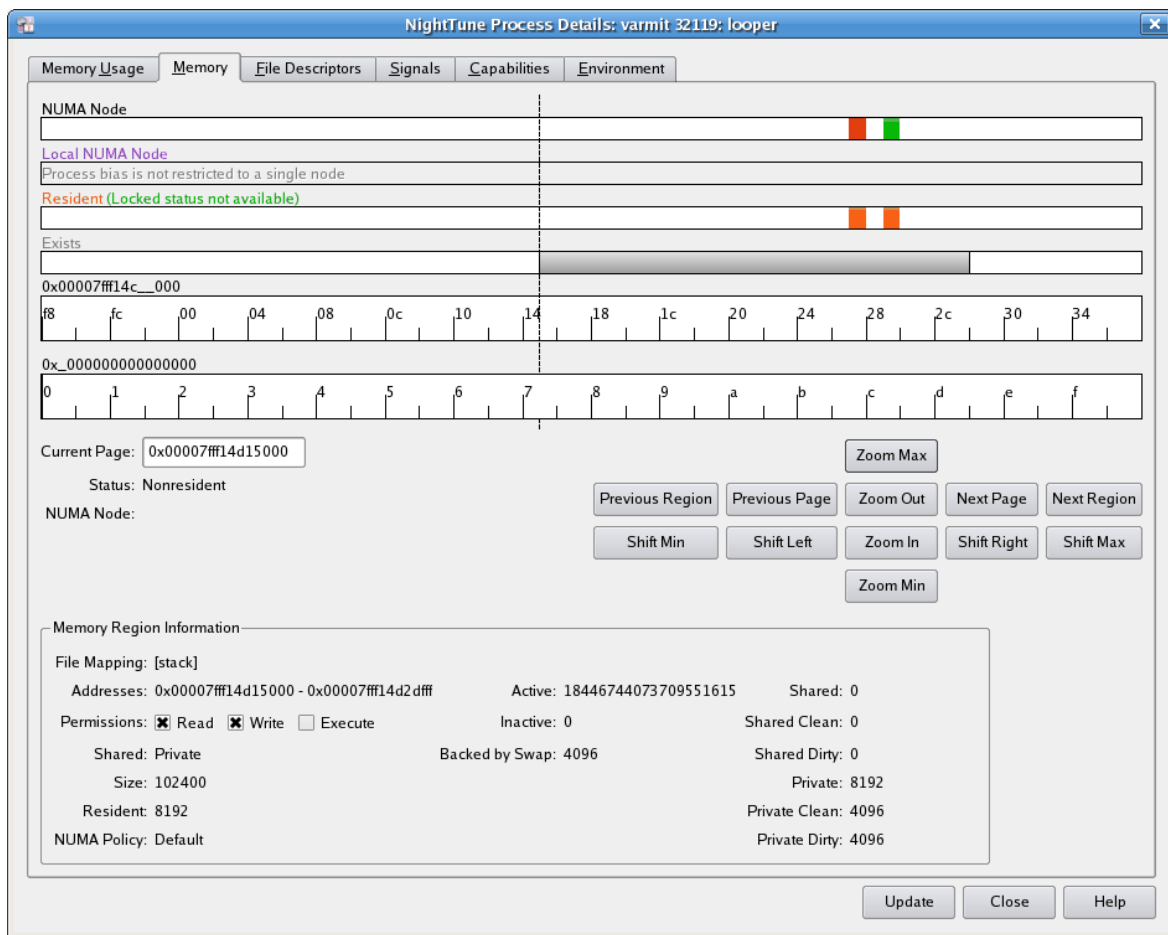


Figure 3-52. Process Details Memory Tab

The horizontal axis in the bars on this tabbed page is used to describe memory addresses.

The bottommost bar is called the Global Address Bar and always covers the entire range of the address space of the process. It exists to provide contextual information about



the position and size of the memory region being viewed by all the other bars when they are zoomed or panned.

All the other bars are synchronized to each other to show information for the zoomed or panned range of memory addresses. A vertical dashed line spans all the bars and shows the location of the **Current Page**, about which additional information is shown below the bars.

The bar second from the bottom is called the **Address Bar** and describes the memory addresses being viewed by the current zoom & pan settings. The bars above it describe characteristics of the pages associated with those memory addresses. Their meanings are:

### **NUMA Node**

The vertical bars in this graph indicate which NUMA node contains the physical memory associated with this virtual memory page. Each NUMA node is assigned a color. It displays bars only on a NUMA system, and only for pages which are resident. The vertical lines in this bar are synchronized with those in the **Address Bar**.

### **Local NUMA Node**

Vertical bars are displayed in this graph only on a NUMA system, and only when the process has a CPU affinity whose CPUs are all from a single NUMA node. In that case, the vertical bars indicate pages which are contained in the memory associated with that NUMA node. The vertical lines are synchronized with those in the **Address Bar**.

### **Resident**

Vertical bars in this bar indicate pages which are resident in memory. The vertical lines in this bar are synchronized with those in the **Address Bar**.

### **Exists**

Grey vertical bars in this bar indicate pages which are mapped as part of the address space. The boundaries of distinct memory regions are denoted by black vertical lines. The vertical lines in this are synchronized with the **Address Bar**.

### **Address Bar**

This bar represents the part of the address space being viewed. If zoomed all the way out, this covers the entire address space. But the panning and zooming operations allow it to represent a smaller part of the total address space. The label above it (e.g., 0x\_0000000) indicates with the “\_” what the numbers in the bar represent (e.g., 0x73000000).

### **Global Address Bar**

This bar represents the entire address space. The shaded area shows which part of the address space is being viewed by all the other bars. A vertical line in this bar represents the current page. The label above it (e.g., 0x\_0000000) indicates with the “\_” what the numbers in the bar represent (e.g., 0x80000000).

## Current Page

This gives the first address within the current page. The current page also is indicated by the vertical dashed line running through the horizontal bars. The current page may be changed by typing a new value into the **Current Page** field, or by clicking in the horizontal bars.

## Status

Indicates the locked and residency status of the page, which may be one of the following:

- **Locked:** forced into physical memory (also implies Resident)
- **Resident:** in physical memory
- **Non-resident:** not in physical memory

## NUMA Node

For pages resident in physical memory on a NUMA system, this shows the NUMA node which contains the page's physical memory.

## Navigation Buttons

Clicking on these buttons adjusts the display:

- **Zoom Max (or Alt-Up)** zooms out to show the entire address space
- **Zoom Min (or Alt-Down)** zooms in to show four pages (the minimum number)
- **Zoom Out (or Up)** zooms out the visible range of addresses by a factor of two to view more of the address space, but with less detail
- **Zoom In (or Down)** zooms in the visible range of addresses by a factor of two to view less of the address space, but with more detail
- **Previous Region (or Alt-Left)** searches backward from the current page to the preceding memory region and makes the last page of that region the current page
- **Next Region (or Alt-Right)** searches forward from the current page to the preceding memory region and makes the first page of that region the current page
- **Previous Page (or Left)** sets the current page to the previous page
- **Next Page (or Right)** sets the current page to the next page
- **Shift Min (or Home)** sets the current page to the lowest page in the address space
- **Shift Max (or End)** sets the current page to the highest page in the address space

- **Shift Left** (or **Ctrl-Left**) shifts the current page to the left (lower numbered address) by approximately 25% of the addresses currently visible
- **Shift Right** (or **Ctrl-Right**) shifts the current page to the right (higher numbered address) by approximately 25% of the addresses currently visible

### Memory Region Information

Information in this area applies to a contiguous region of memory which was mapped as one operation and which contains the current page. The memory may be private to the process or may be mapped to a file. If mapped as a file, the memory will be from a contiguous region of that file and will have consistent permissions. The pieces of information are:

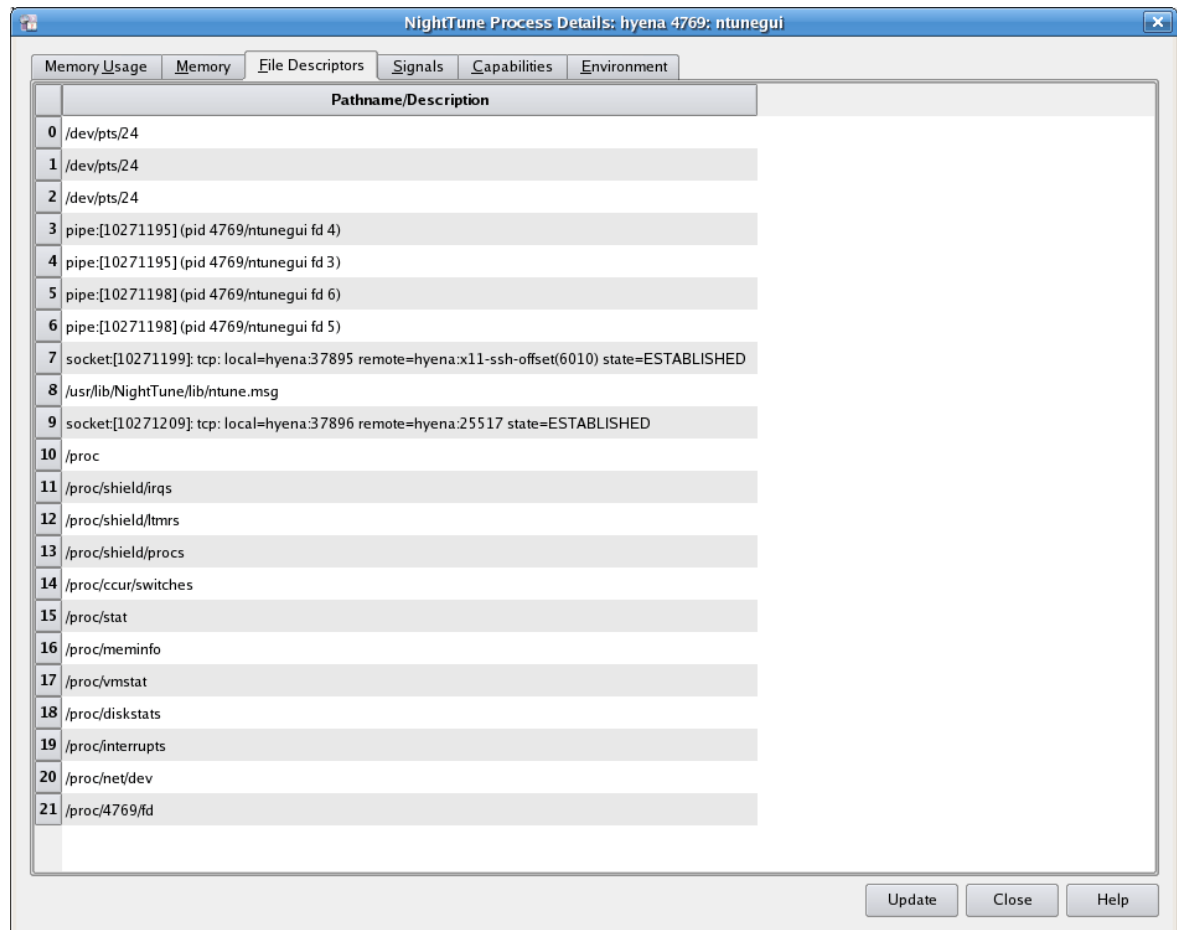
- **File Mapping** shows the name of the file with which this memory is associated, if any
- **Addresses** shows the range of memory addresses for this memory region
- **Permissions** shows the **Read**, **Write**, and **Execute** permissions for the pages in this memory region
- **Shared** shows whether the pages in the memory region are:
  - **Shared**, where modifications to the pages affect any associated file
  - **Private**, where modifications to the pages affect an in-memory copy of the content from the associated file and not the file itself
- **Size**, which shows the number of bytes in the memory region
- **Resident**, which shows the number of bytes from the memory region that are resident in physical memory
- **NUMA Policy**, which describes how memory from this region will be assigned to NUMA nodes. It may be one of:
  - **Default**, meaning that pages will be selected from the NUMA node associated with the CPU on which the process is running when the mapping occurs, if possible; or from any NUMA node otherwise
  - **Preferred**, meaning that pages will be selected from a specified NUMA node, if possible; or from any NUMA node otherwise
  - **Bind**, meaning that pages will be selected from a specified set of NUMA nodes and that those nodes are enforced strictly (i.e. if no memory is available on any of those nodes, an error will occur)
  - **Interleaved**, meaning that pages will be selected from a specified set of NUMA nodes, assigned in a round-robin order

- **Active**, shows the number of bytes in pages in this memory region which have been used recently and therefore are unlikely to be reclaimed when memory is needed
- **Inactive**, which shows the number of bytes in pages in this memory region which have been used less recently and therefore are likely to be reclaimed when memory is needed
- **Backed by Swap**, which shows the number of bytes in pages for which swap space has been allocated
- **Shared**, which shows the number of bytes in pages which are shared by at least two processes with independent address spaces. It is further subdivided into:
  - **Shared Clean**, which shows the number of bytes in Shared pages for which the physical memory is consistent with any associated file or swap space
  - **Shared Dirty**, which shows the number of bytes in Shared pages for which the physical memory is not consistent with any associated file or swap space
- **Private**, which shows the number of bytes in pages which are not shared by at least two processes with independent address spaces. It is further subdivided into:
  - **Private Clean**, which shows the number of bytes in Private pages for which the physical memory is consistent with any associated file or swap space
  - **Private Dirty**, which shows the number of bytes in Private pages for which the physical memory is not consistent with any associated file or swap space

## **File Descriptors Tab**

The **Process Details File Descriptors** tab lists the files or devices associated with the file descriptors for the process.

The following illustrates the File Descriptors tab:



**Figure 3-53. Process Details File Descriptors Tab**

The following formats are used:

*filename*

The file descriptor is associated with the name *filename*.

*filename* (deleted)

The file descriptor is associated with a file that previously had the name *filename*, but which has been deleted since this process opened it.

pipe:[*inode*] (*other-pid*) ...

The file descriptor is associated with a pipe with the specified *inode* number. If any other processes on the same system also have the pipe open, they will be listed as *other-pid*.

socket:[*inode*]: tcp/udp/raw: local=*ip:port* remote=*ip:port* state=*s* (*other-pid*) ...

The file descriptor is associated with a TCP, UDP, or RAW socket with the specified *inode* number. The local and remote IP addresses and ports are shown, if available. They will be displayed as symbolic names if possible. If the protocol is stateful (e.g. TCP), state information is shown. States are values like ESTABLISHED, LISTEN, FIN\_WAIT1, etc. If the remote end of the socket is on the same system and some other process is connected to that remote end, then the other process will be listed as *other-pid*.

socket:[*inode*]: unix/*type*: name=*associated-filename* state=*s*

The file descriptor is associated with a UNIX domain socket with the specified *inode* number. It may have a *type* of either **stream** or **dgram**. If the socket is associated with a file, its *associated-filename* will be shown. If the protocol is stateful (e.g. stream), state information is shown. States are values like LISTENING, CONNECTED, etc.

socket:[*inode*]: packet

The file descriptor is associated with a PACKET socket with the specified *inode* number.

## Signals Tab

The Process Details Signals tab shows the signal handling capabilities of the process.

The following illustrates the Signals tab:

Number	Name	Pending	Shared Pending	Blocked	Ignored	Handled	Restart	Description
1	SIGHUP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Hangup
2	SIGINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Interrupt
3	SIGQUIT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Quit
4	SIGILL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Illegal instruction
5	SIGTRAP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Trace/breakpoint trap
6	SIGABRT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Aborted
7	SIGBUS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Bus error
8	SIGFPE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Floating point exception
9	SIGKILL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Killed
10	SIGUSR1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	User defined signal 1
11	SIGSEGV	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Segmentation fault
12	SIGUSR2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	User defined signal 2
13	SIGPIPE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Broken pipe
14	SIGALRM	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Alarm clock
15	SIGTERM	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Terminated
16	SIGSTKFLT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Stack fault
17	SIGCHLD	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Child exited
18	SIGCONT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Continued
19	SIGSTOP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Stopped (signal)
20	SIGTSTP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Stopped
21	SIGTTIN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Stopped (tty input)
22	SIGTTOU	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Stopped (tty output)
23	SIGURG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Urgent I/O condition

**Figure 3-54. Process Details Signals Tab**

The columns' meanings are:

### Number

The numeric value of the signal.

### Name

The symbolic name of the signal.

### Pending

If this column is checked, then an instance of this signal is pending for this process (and this particular thread if multi-threaded).

### **Shared Pending**

If this column is checked, then an instance of this signal is pending for this process (and may be accepted by any thread in the process).

### **Blocked**

If this column is checked, then the process has blocked delivery of this signal.

### **Ignored**

If this column is checked, then the process is ignoring any instances of this signal.

### **Handled**

If this column is checked, then the process has a signal handler for this signal.

### **Restart**

If this column is checked, then if this signal interrupts a system call, it will be restarted after the signal is handled (SA\_RESTART).

### **Description**

A description of the signal.

## **Capabilities Tab**

The Process Details Capabilities tab shows the capabilities associated with the process.



The following illustrates the Capabilities tab:

Number	Name	Inheritable	Permitted	Effective
0	CAP_CHOWN	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
1	CAP_DAC_OVERRIDE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	CAP_DAC_READ_SEARCH	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	CAP_FOWNER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	CAP_FSETID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	CAP_KILL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	CAP_SETGID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	CAP_SETUID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	CAP_SETPCAP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	CAP_LINUX_IMMUTABLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	CAP_NET_BIND_SERVICE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	CAP_NET_BROADCAST	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	CAP_NET_ADMIN	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	CAP_NET_RAW	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	CAP_IPC_LOCK	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	CAP_IPC_OWNER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	CAP_SYS_MODULE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17	CAP_SYS_RAWIO	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	CAP_SYS_CHROOT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
19	CAP_SYS_PTRACE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
20	CAP_SYS_PACCT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
21	CAP_SYS_ADMIN	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
22	CAP_SYS_BOOT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**Figure 3-55. Process Details Capabilities Tab**

The columns' meanings are:

### Number

The numeric value of the capability.

### Name

The symbolic name of the capability.

### Inheritable

If this column is checked, then any new process created by this process (e.g. with `fork(2)`) will inherit the capability.

### Permitted

If this column is checked, then the process is permitted to have the capability, even if it voluntarily disabled it.

### Effective

If this column is checked, then the process has the capability currently.

### Environment Tab

The Process Details Environment tab lists the values of all environment variables associated with the process and their values.

The following illustrates the Environment tab:

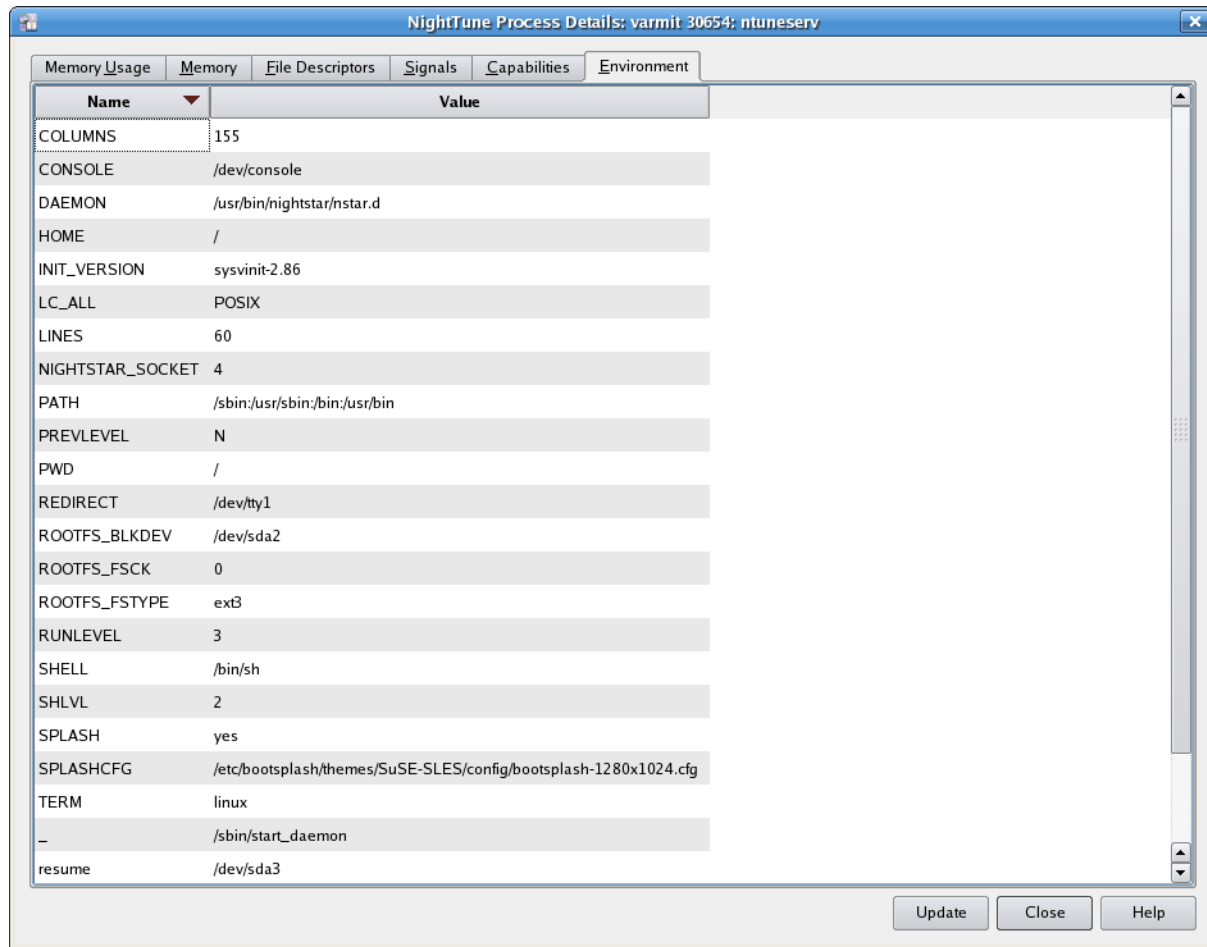
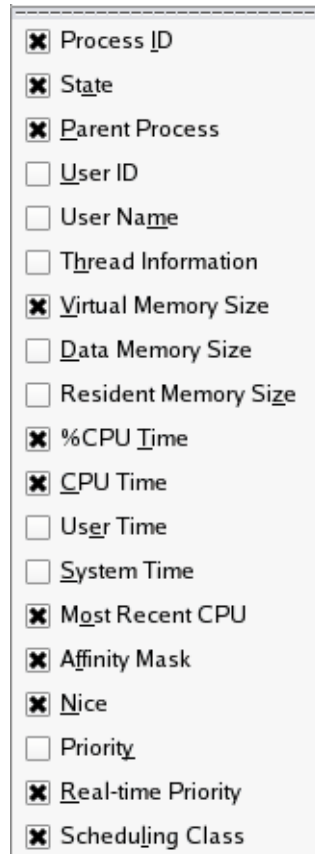


Figure 3-56. Process Details Environment Tab

## Process Fields Menu

The Process Fields menu is displayed from the Display Fields option of the Process List context menu. This menu allows you to select which fields (columns) are visible in the Process List panel.

The following illustrates the Process Fields menu:



**Figure 3-57. Process Fields Menu**

### NOTE

The Process Fields menu, like the other menus, can be “torn off” from the menu bar to reside in its own window separate from NightTune by clicking on the dashed line at the top of the menu. This is especially useful when making multiple changes to avoid having to select the menu after every choice.

The following paragraphs describe each of the selectable fields in more detail:

### Process ID

Mnemonic: I

This field displays the **PID** column which contains the process ID as returned by `getpid(2)`.

### State

Mnemonic: A

This field displays the **State** column, which contains the process state, which can be one of:

- **Running**: currently running or on a run queue and able to run
- **Waiting**: also known as sleeping
- **Disk**: performing an uninterruptible sleep (usually disk I/O)
- **Stopped**: stopped or traced
- **Paging**
- **Zombie**: a defunct process

### Parent Process

Mnemonic: P

This field displays the **Parent** column, which contains the process ID of the parent process.

### User ID

Mnemonic: U

The field displays the **UID** column, which contains the user ID of the process.

### User Name

Mnemonic: M

This field displays the **Username** column, which contains user name corresponding to the user ID.

### Thread Information

Mnemonic: H

This field displays the **Threads** column, which contains the number of threads in the process.

### Virtual Memory Size

Mnemonic: V

This field displays the **Size** column, which contains the number of KB of virtual memory associated with the process.

### Data Memory Size

Mnemonic: D

This field displays the **Data** column, which contains an approximation of the memory used by the program. It is essentially the total number of KB of virtual memory used by the process less memory used for instruction (`.text`) pages. It includes data pages used by shared libraries.

### Resident Memory Size

Mnemonic: Z

This field displays the **Resident** column, which contains the number of KB of memory associated with the process that is resident in physical memory.

### %CPU Time

Mnemonic: T

This field displays the **%CPU** column, which contains the percentage of CPU time used by the process.

### CPU Time

Mnemonic: C

This field displays the **CPU Time** column, which contains the amount of CPU time used by the process in seconds.

### User Time

Mnemonic: E

This field displays the **User** column, which contains the amount of time used on behalf of the process, excluding time spent in system calls, in seconds.

### System Time

Mnemonic: S

This field displays the **System** column, which contains the amount of time used on behalf of the process for system calls in seconds.

### Most Recent CPU

Mnemonic: O

This field displays the CPU column, which contains the number of the CPU upon which the process last executed.

### Affinity Mask

Mnemonic: F

This field displays the Affinity column, which contains the CPU affinity mask which specifies on which CPUs the process can execute. The column may display the term **all**, indicating that the process is free to run on any CPU on the system (excluding shielded CPUs). When the CPU affinity does not designate all CPUs on the system, the mask is displayed as a hexadecimal number. The least significant bit in the mask represents logical CPU 0.

### Nice

Mnemonic: N

This field displays the Nice column, which contains the *nice* value as set by the **nice (1)** command or the **nice (2)** system service. The nice value provides an initial basis used by the kernel for determining the priority of processes in the **SCHED\_OTHER**. Lower values represent more favorable priorities.

### Priority

Mnemonic: Y

This field displays the Pri column, which contains the internal kernel priority value. Lower values represent more favorable priorities. For processes in the **SCHED\_OTHER** scheduling class, the priority is adjusted by the kernel as the process runs, based on CPU utilization. For processes in other scheduling classes, the values are determined from the real-time priority, negated and adjusted by a bias.

### Real-time Priority

Mnemonic: R

This field displays the RPr column, which contains the real-time priority within the process's scheduling class, as specified by the program or by the **run (1)** command. Higher values represent more favorable priorities.

### Scheduling Class

Mnemonic: L

This field displays the CL column, which contains the scheduling class associated with the process. The possible values are:

- FF: FIFO policy

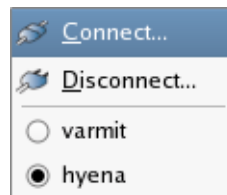
- RR: round-robin policy
- OT: other policy (used for non-real-time processes)

The scheduling classes are explained in a bit more detail in “Scheduling Class” on page 3-56, and defined in even greater detail in `sched_setscheduler(2)`.

## Systems Menu

The **Systems** selection from the **Monitor** menu allows you to control connection to other systems on which NightTune is installed.

The following illustrates the **Systems** menu:



**Figure 3-58. Systems Menu**

The following paragraphs describe each of the selections in more detail:

### Connect...

Mnemonic: C

This displays the Remote System Connect dialog allowing you to connect to another system. This dialog contains two tabs:

#### System Selection/Login

To connect to a system, select it from the dropdown list, type in the system name or IP address or double-click in the list (which derives its information from the `/etc/hosts` file). You will only be able to connect to a system on which the same version of the ntune server is installed. Provide the appropriate login name and password, then click on the **Connect** button.

#### Advanced

This tab allows you to set the CPU affinity, scheduling class and priority for the ntune server on the target system, if desired.

### Disconnect...

Mnemonic: D

This displays a dialog that allows you to disconnect from any of the systems to which you currently are connected.

### **System Names**

The names of the systems to which you are connected are listed. Selecting a system name activates it as the “current” system for NightTune activities. When creating new panels, they always correspond to the current system. The current system name is indicated by the filled radio button.



This chapter guides you in operating NightTune to execute the following specific tasks:

- Monitoring User Processes (see “Monitoring User Processes” on page 4-2)
- Changing User Process Scheduling Attributes (see “Changing User Process Scheduling Attributes” on page 4-6)
- Shielding a CPU (see “Shielding a CPU” on page 4-10)
- Changing the CPU Affinity of an Interrupt (see “Changing the CPU Affinity of an Interrupt” on page 4-14)

## Monitoring User Processes

If the Process List panel is not visible currently, select the Process List panel from the Monitor menu or click on the Process List tool icon.

The Process List panel will appear in the window as illustrated below:

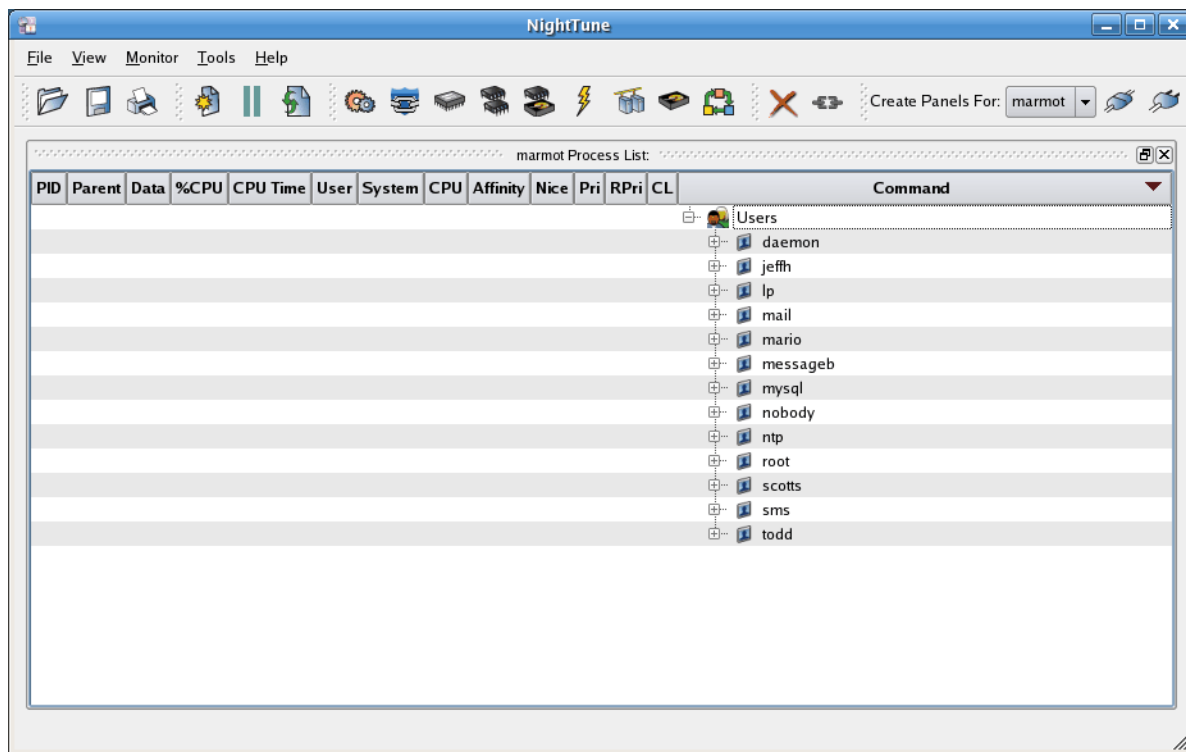
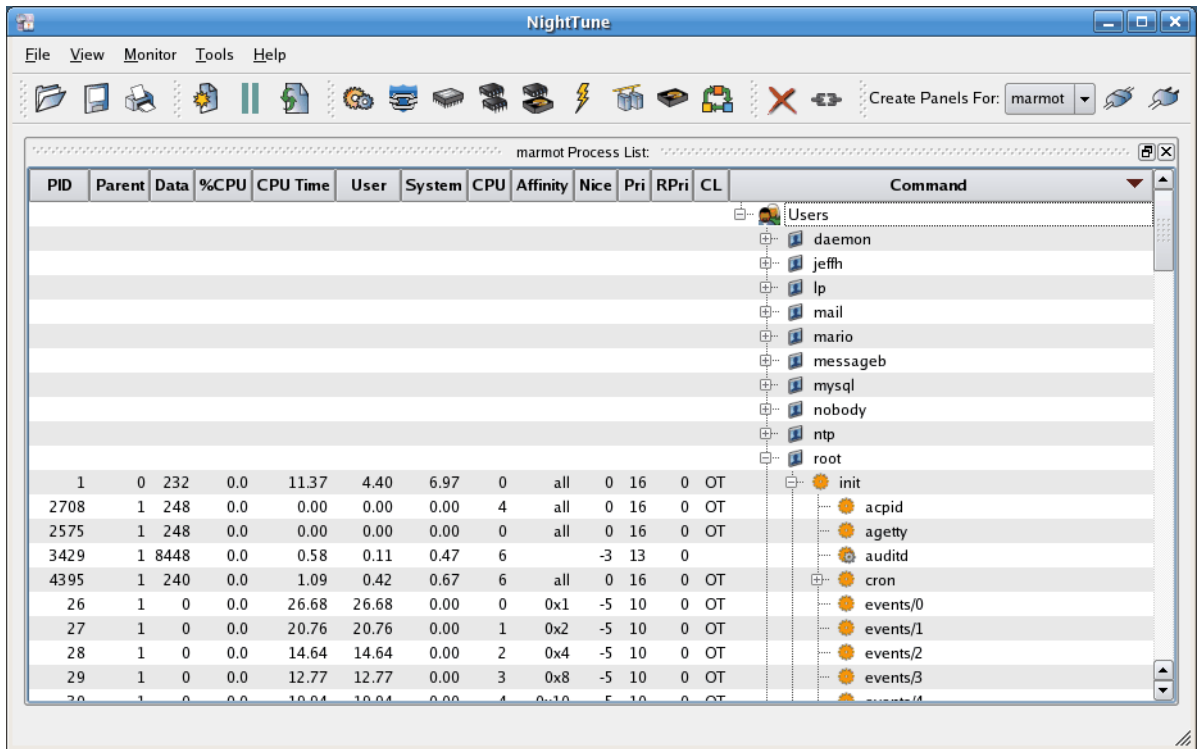


Figure 4-1. Monitoring User Processes

## Selecting the User Process

The Process List panel is populated with user lines which, when expanded, describe the processes associated with each user.

Click on the plus sign associated with the user name of the process you wish to monitor. The list of processes is expanded beneath the user name as shown below. If you wish to see children of a given process (e.g. init), then click on its plus sign.



**Figure 4-2. Selecting the User Process**

Process attributes are displayed and updated periodically.

If a process is multi-threaded, the list of individual threads will be displayed when the parent process is selected and Show Threads is checked on the Process List context menu as shown below:

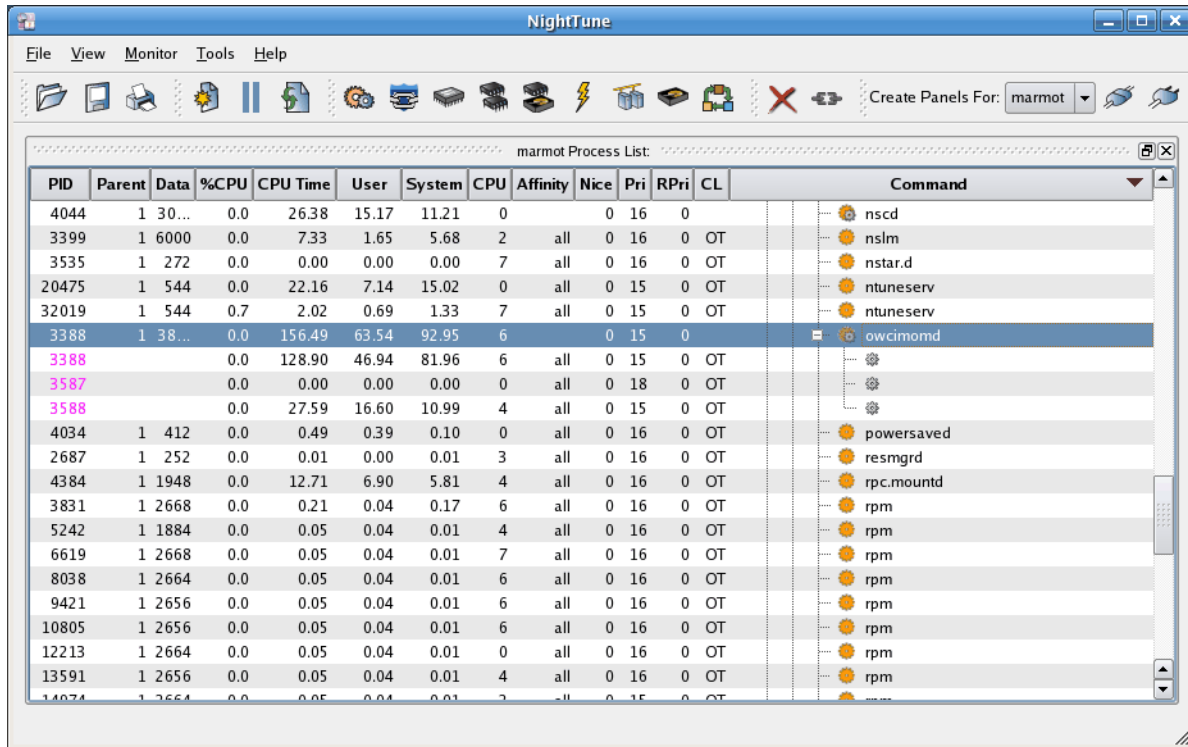


Figure 4-3. Monitoring Multi-threaded Processes

## Customizing the Process Information

Select the Display Fields item from the Process List context menu to display the Process Fields menu.

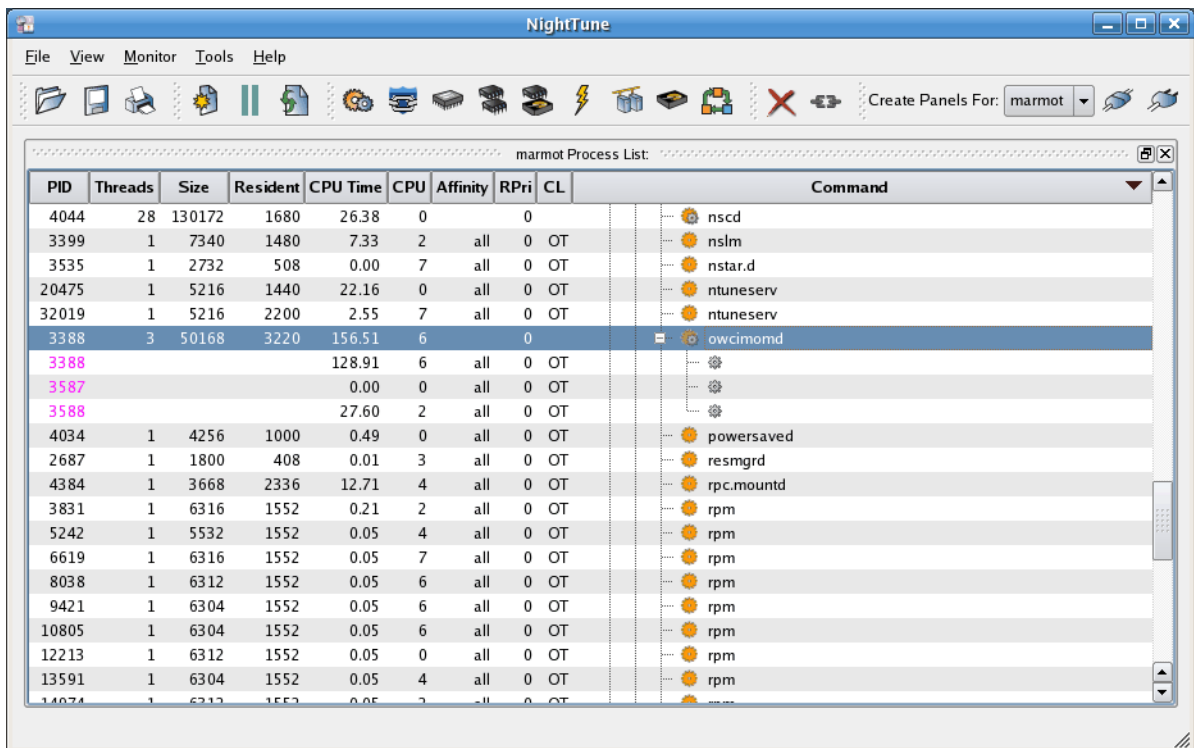
### NOTE

The Process Fields menu, like the other menus, can be “torn off” from the menu bar to reside in its own window separate from NightTune by clicking on the dashed line at the top of the menu. This is especially useful when making multiple changes to avoid having to select the menu after every choice.

Select or deselect process fields of interest to customize the display.

In the illustration below, the following process fields have been selected for display:

- Process ID
- Thread Information
- Virtual Memory Size
- Resident Memory Size
- CPU Time
- Most Recent CPU
- Affinity Mask
- Real-time Priority
- Scheduling Class

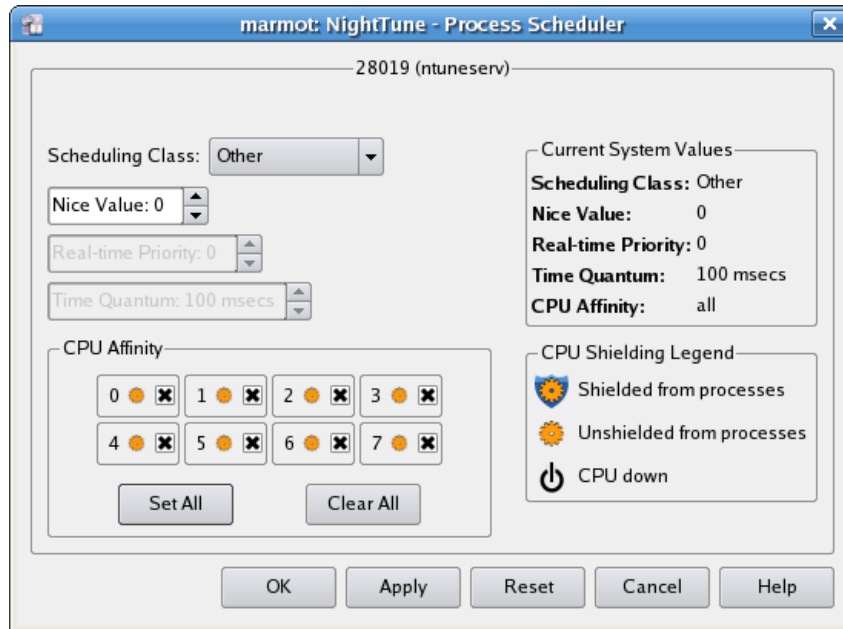


**Figure 4-4. Customizing the Process Information**

See “Process List Panel” on page 3-46 for more information on process monitoring.

## Changing User Process Scheduling Attributes

Select the thread or process by clicking on the line displayed for that thread or process, then right-click to display the Process List context menu. Select Process Scheduler from this menu to display the Process Scheduler dialog, as shown below:



**Figure 4-5. Process Scheduler Dialog**

The current scheduling attributes are shown in the dialog.

Change the Scheduling Class to Round Robin, the Real-time Priority to 20, and clear all CPU Affinity boxes except for CPU 0, which should be selected.

Click on Apply to apply the changes to the process; the Process Scheduler dialog displays the changes and the Process List panel display changes on the next refresh indicating the new scheduling attributes, as shown below:

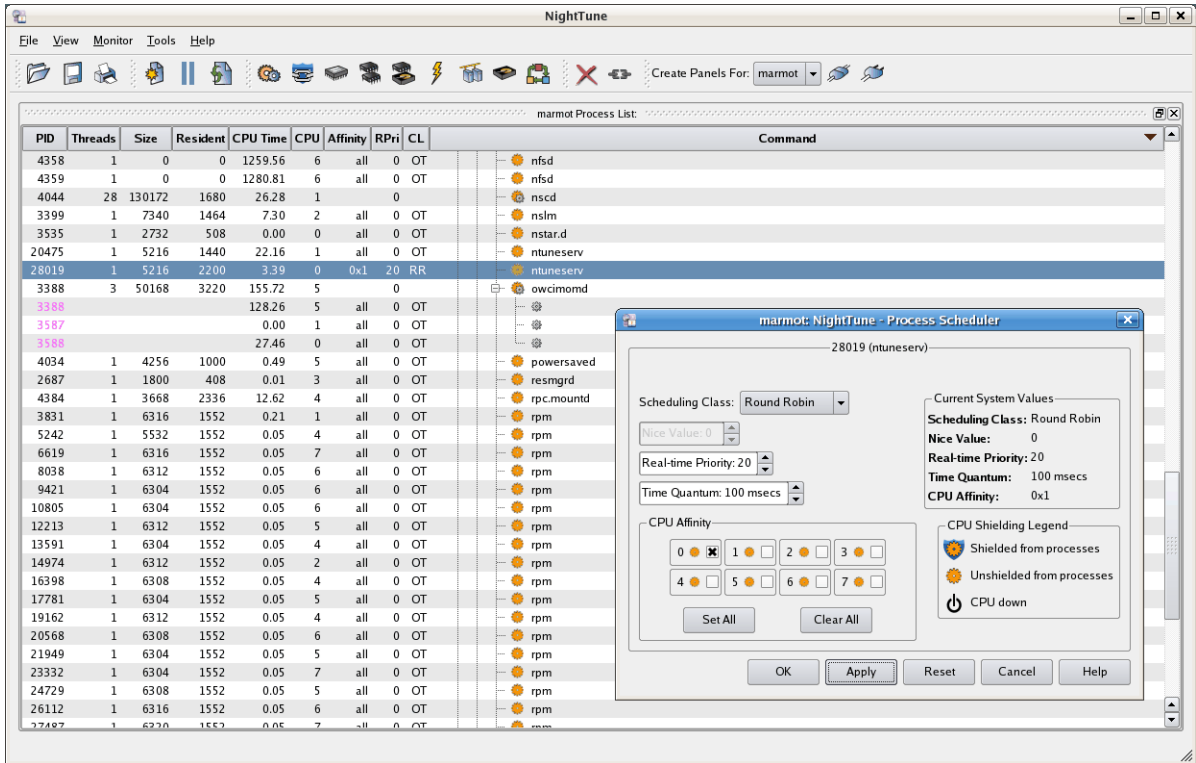


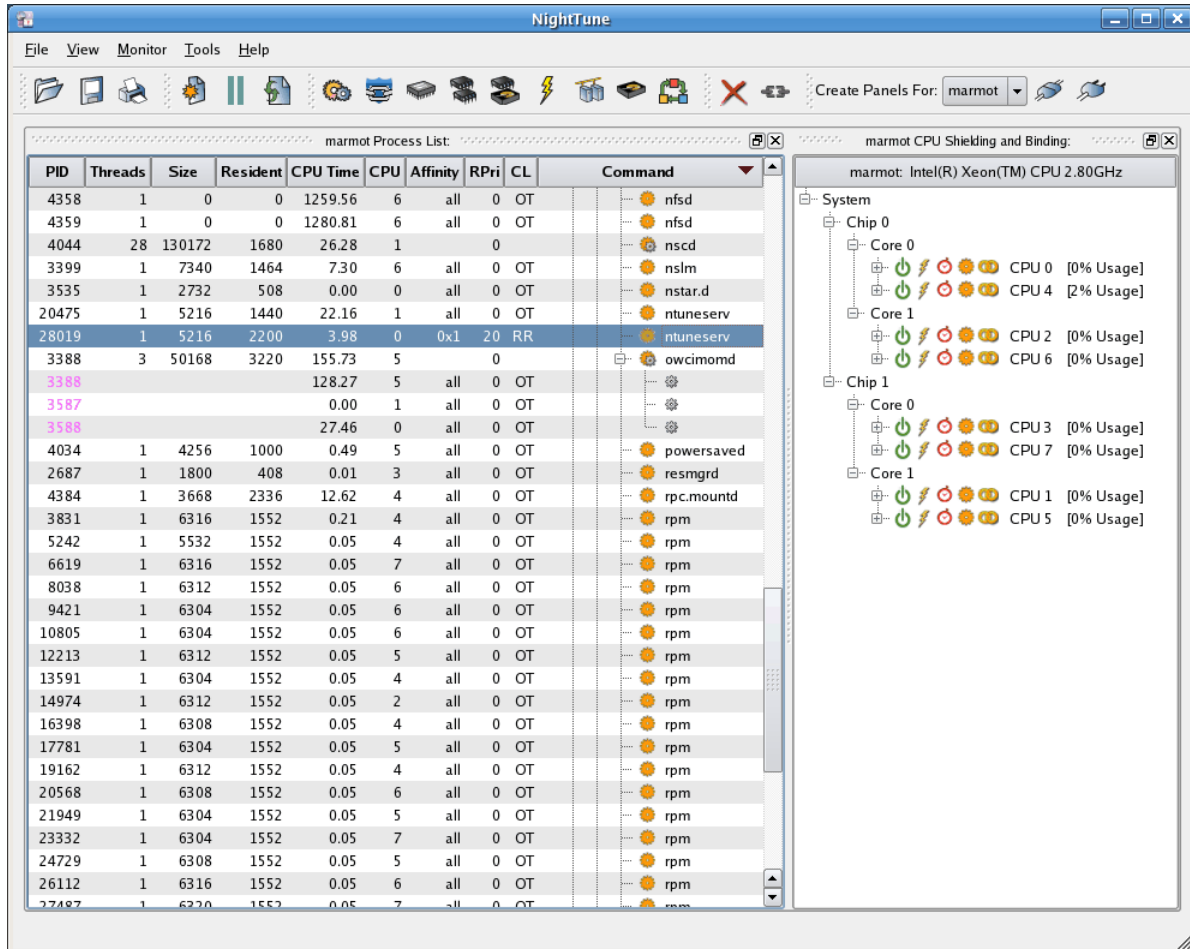
Figure 4-6. Changing User Process Scheduling Attributes

## Using Drag and Drop to Change Process CPU Affinity

NightTune allows you to use drag and drop actions to change the CPU affinity of a process, group of processes, or individual threads.

If the CPU Shielding and Binding panel is not visible currently, select the CPU Shielding and Binding panel from the Monitor menu or click on the CPU Shielding and Binding tool icon.

The CPU Shielding and Binding panel will appear in the window next to the Process List panel as shown below:



**Figure 4-7. Viewing CPU Status**

For each CPU displayed, a list of processes and interrupts that are bound to the CPU are available in the CPU Shielding and Binding panel. To see them for a CPU, select that CPU, right click and select Expand All.

A process or interrupt is considered bound to a CPU if the CPU affinity of the process or interrupt specifies only that single CPU.



When you changed the CPU affinity of your process in the section above to CPU 0, it became bound to CPU 0. The process now appears under Bound Processes for CPU 0, as shown below:

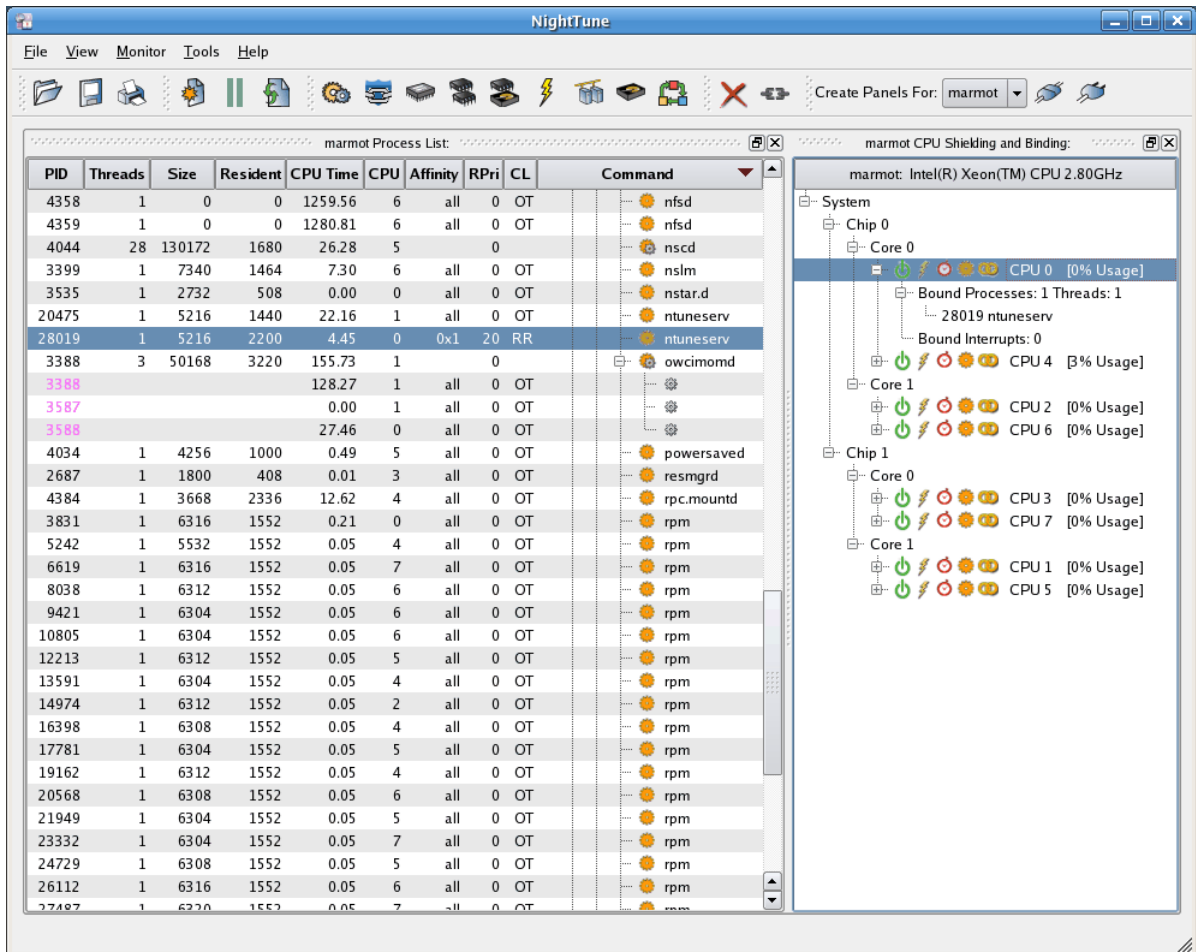


Figure 4-8. Bound Processes

To change the CPU of that process again, press and hold the middle mouse button over the process entry under CPU 0 Bound Processes. Drag the pointer to CPU 1 and release the mouse button. Then right click on CPU 1 and select Expand All to see the bound processes for that CPU.

The new CPU affinity is reflected in the Process List panel and under Bound Processes for CPU 1:

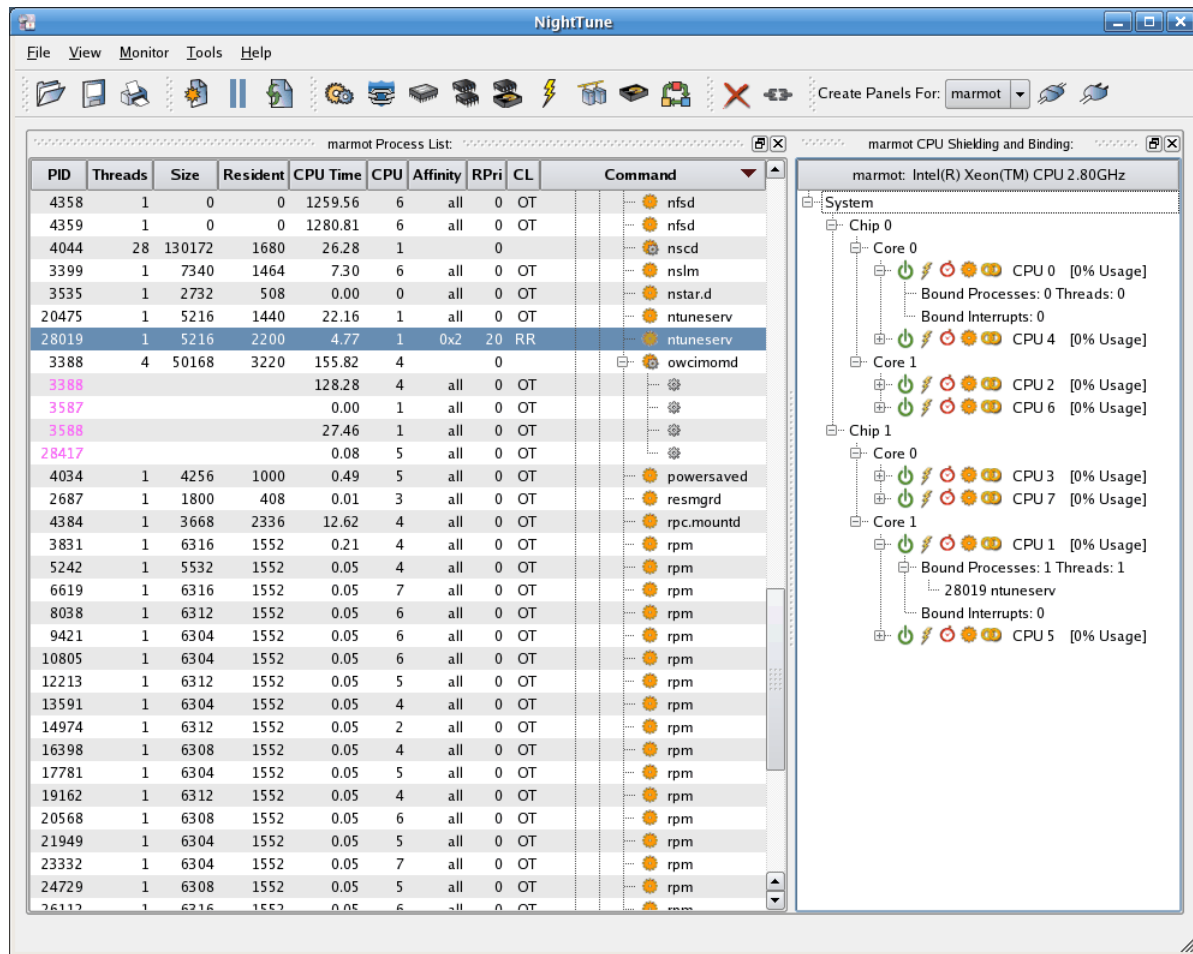


Figure 4-9. Using Drag and Drop to Change Process CPU Affinity

Similarly, you can drag a process or group of processes from the Process List panel and drop them onto a CPU entry in the CPU Shielding and Binding panel to change their CPU affinity.

See “Process List Panel” on page 3-46 and “CPU Shielding and Binding Panel” on page 3-7 for more information on process CPU affinity.

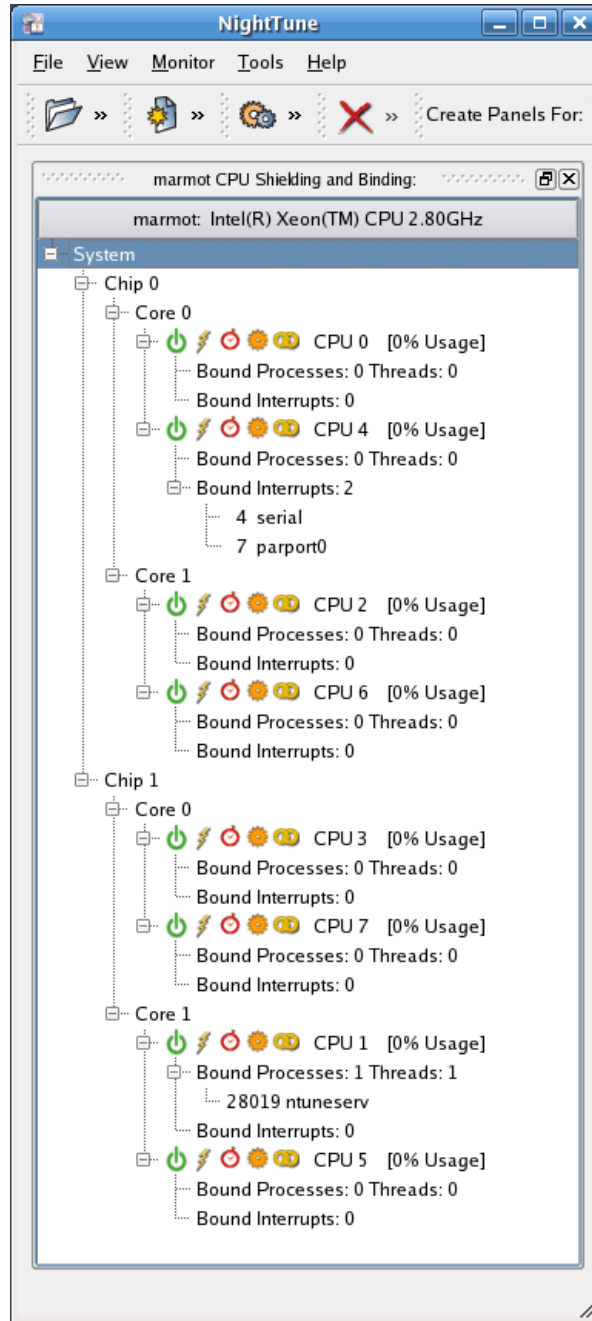
## Shielding a CPU

This section describes activities associated with shielding a CPU.

Close the Process List panel from the window by clicking on the X in the upper right corner of the panel.

Ensure that the CPU Shielding and Binding panel is in the window.

Next, ensure that bound processes and interrupts are visible for all CPUs by right clicking on the System item and selecting Expand All.



**Figure 4-10. CPU Shielding and Binding Panel**

The CPU Shielding and Binding panel above describes a system with two physical CPU chips, each of which has two cores, each of which has two hyper-threaded logical CPUs. CPU 0 and CPU 4 are the two logical CPUs which comprise physical chip 0's core

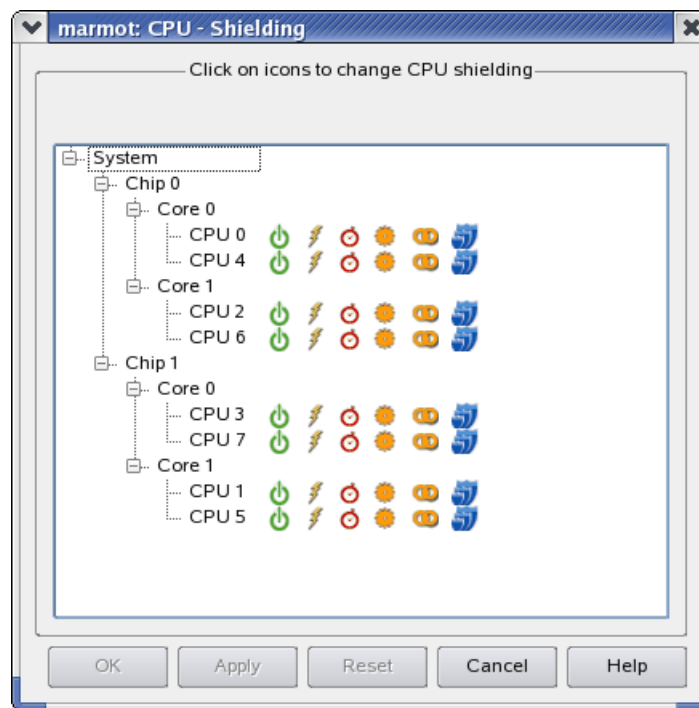
0; CPU 2 and CPU 6 are the two logical CPUs which comprise physical chip 0's core 1; etc.

**NOTE**


Not all systems support hyper-threading. Systems with hyper-threading have two CPUs per core in the CPU Shielding and Binding panel. However, the sibling CPU is not always a logical CPU with a consecutive number. In this example, CPU 4 is CPU 0's hyper-threaded sibling.

The panel above indicates that no shielding is currently active and that a single process has been bound to CPU 1.

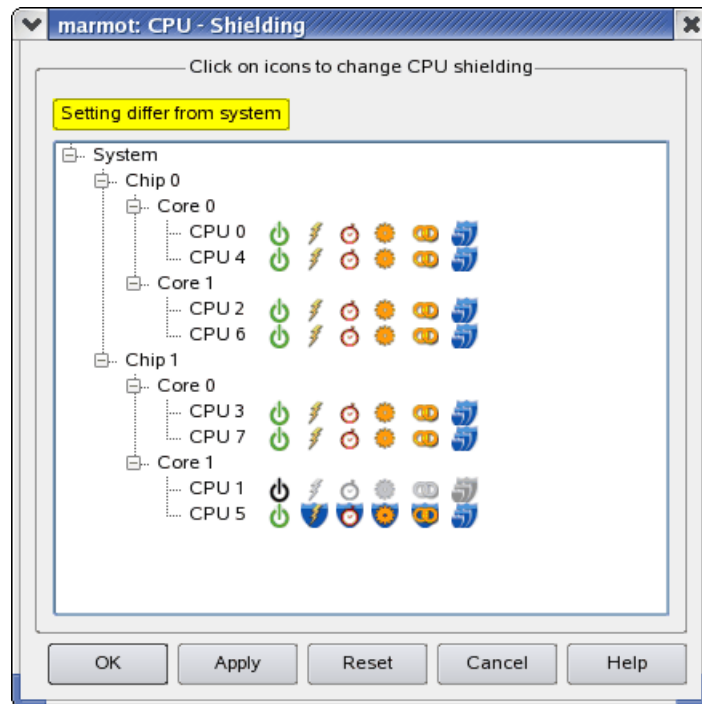
To specify shielding on a CPU, right-click on the CPU Shielding and Binding panel and select the Change Shielding menu item. The CPU Shielding dialog appears, as shown below:



**Figure 4-11. CPU Shielding Dialog**

To specify maximum shielding on CPU 5, click the maximum shield icon.  in the box for CPU 5.

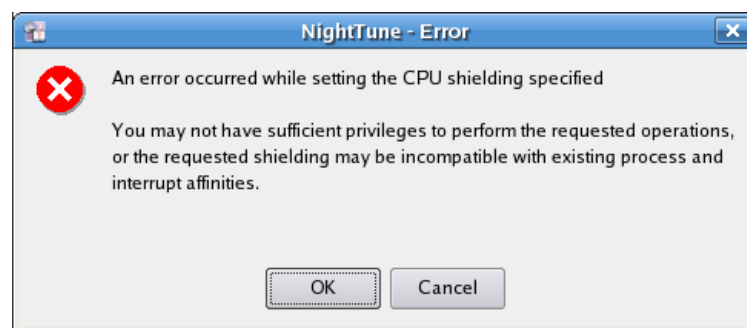
The CPU Shielding dialog display changes to indicate that CPU 5 is to be shielded from interrupts, processes, the local timer, and hyper-threading. To achieve the shielding from hyper-threading for CPU 5, its hyper-threading sibling, CPU 1, is marked **DOWN** automatically. If your system does not support hyper-threading, CPU 1 will remain unchanged.



**Figure 4-12. Shielding a CPU**

Click on Apply to apply the shielding changes.

In the illustration above, there was a process bound to CPU 1. This prevents CPU 1 from being marked down. A diagnostic dialog similar to the following will appear if this situation exists:



**Figure 4-13. Error Shielding CPU**

The process that is bound to CPU 1 must first be moved to another CPU before it can be marked Down.

Press Cancel to close the CPU Shielding dialog. Drag the process in the Bound Processes list under CPU 1 using the mouse button from CPU 1 to CPU 3. Re-open the

CPU Shielding dialog by right-clicking on the CPU Shielding and Binding panel and selecting the **Change Shielding** menu item. In the CPU Shielding dialog, click the **Max Shield** icon on CPU 1. Now click the **Apply** button again and the CPU Shielding dialog disappears and the CPU Shielding and Binding panel will reflect the changes.

Interrupts may also be bound to specific CPUs. The **Bound Interrupts** entry in the CPU Shielding and Binding panel displays the list of interrupts that are bound. Bound interrupts prevent downing a CPU just as bound processes do. Interrupts may be dragged from the list to other CPUs in the same manner as with processes.

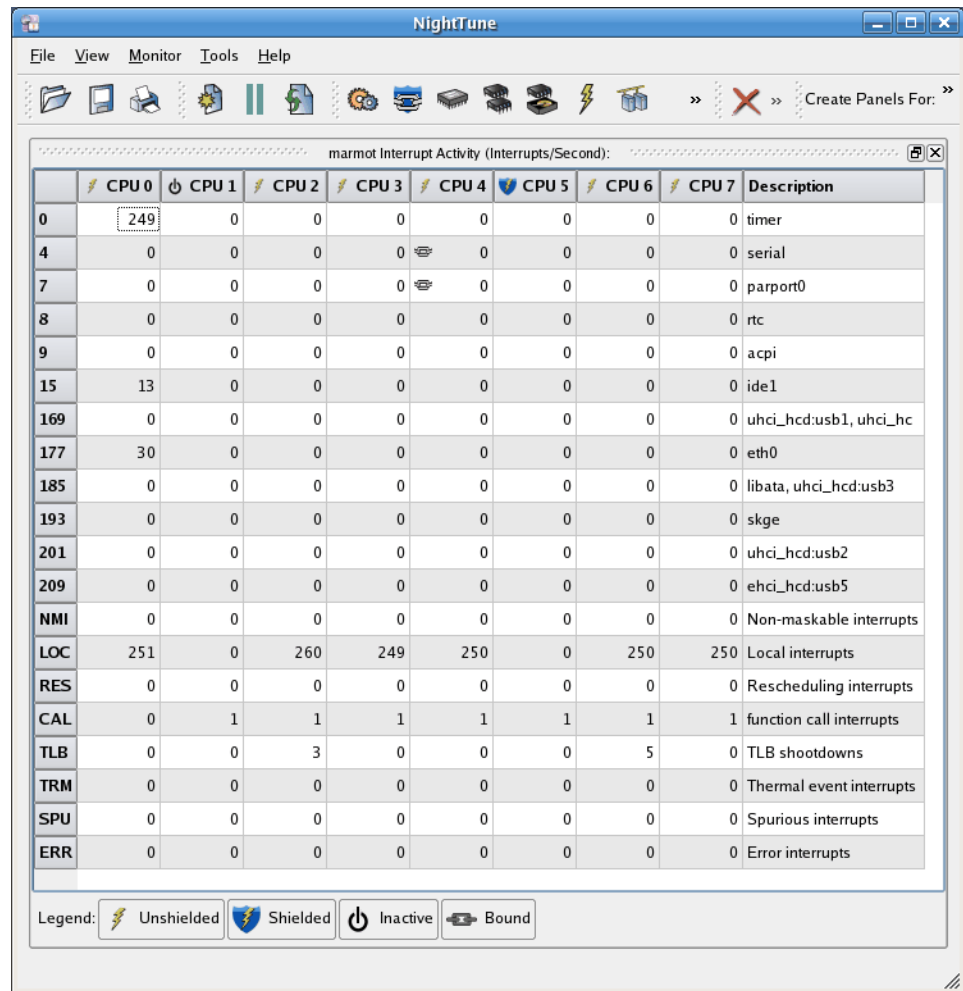
See “CPU Shielding and Binding Panel” on page 3-7 for more information on CPU shielding.

## Changing the CPU Affinity of an Interrupt

Add the **Interrupt Activity** panel to the window using the **Monitor** menu or the **Interrupt Activity** tool icon. Close all other panels if they exist by clicking on the X in the upper right corner of each panel.

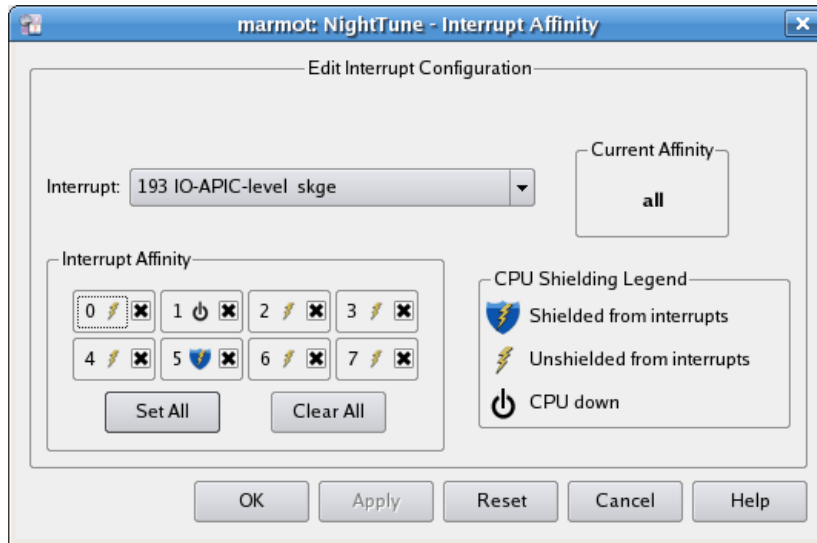
If all the **Interrupt Activity** panes are displayed, hide the bar graph and line graph panes by unchecking the **Show bar graph pane** and **Show line graph pane** items from the **Interrupt Activity Context Menu**. The context menu is displayed by right-clicking while positioned in the **Interrupt Activity** panel.

The NightTune window will display the **Interrupt Activity Text** pane as shown below:



**Figure 4-14. Monitoring Interrupt Activity**

Click on the line associated with an interrupt, right-click to display the Interrupt Affinity context menu, then select the Set CPU Affinity menu item to display the Interrupt Affinity dialog, as shown below:



**Figure 4-15. Interrupt Affinity Dialog**

Clear all the CPU affinity settings by clicking on **Clear All**. Then select CPU 0 by clicking on its checkbox. Apply the change by clicking on the **Apply** button.

The CPU affinity change will be reflected in the **Interrupt Activity** panel. A **Bound** icon will now appear in the cell for the serial interrupt on CPU 0, indicating that its affinity mask has selected CPU 0 but not all other CPUs.



The illustration below reflects the affinity change for the serial interrupt:

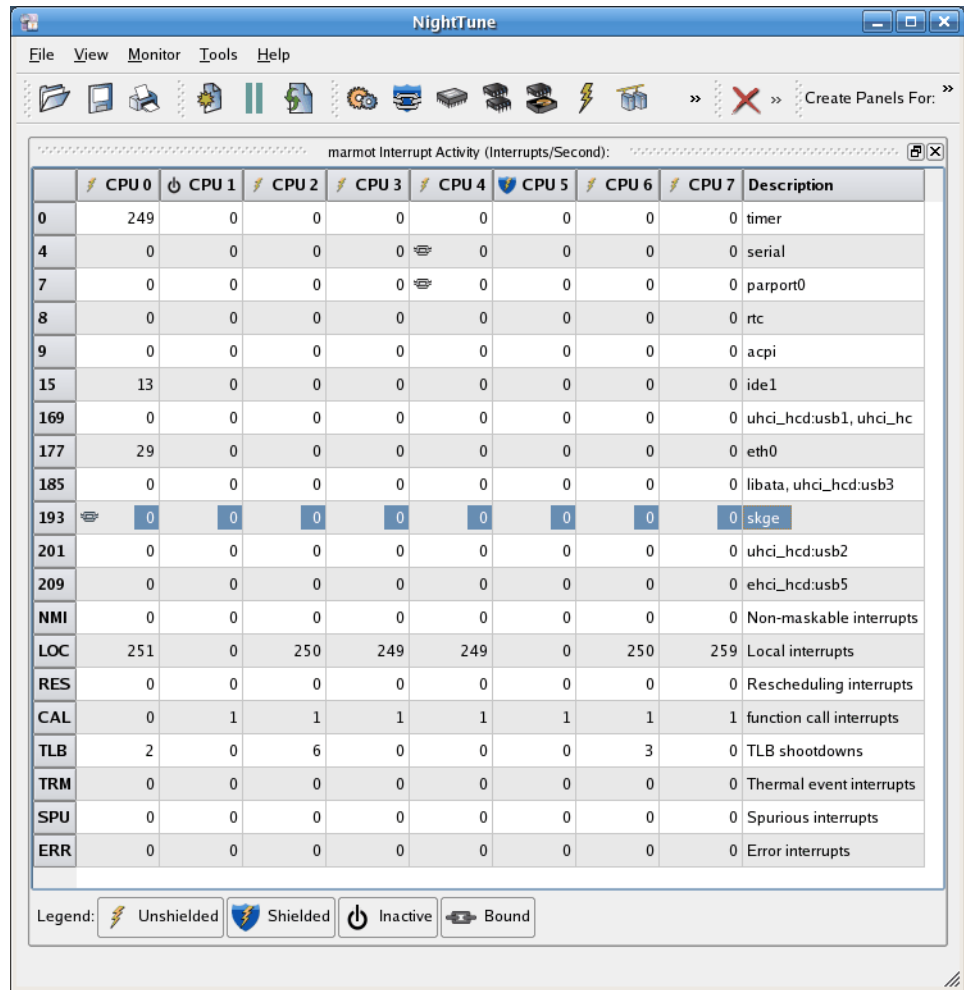


Figure 4-16. Changing the CPU Affinity of an Interrupt

## Using Drag and Drop to Change Interrupt CPU Affinity

NightTune allows you to use drag and drop actions to change the affinity of an interrupt.

In the CPU Shielding and Binding panel, under CPU 0, is an item called Bound Interrupts. Under it is a list of interrupts. Click the line describing an interrupt, then drag the pointer to another CPU and release the mouse button.

The Bound Processes and the Bound Interrupts lists under the destination CPU reflect the change, as shown below:

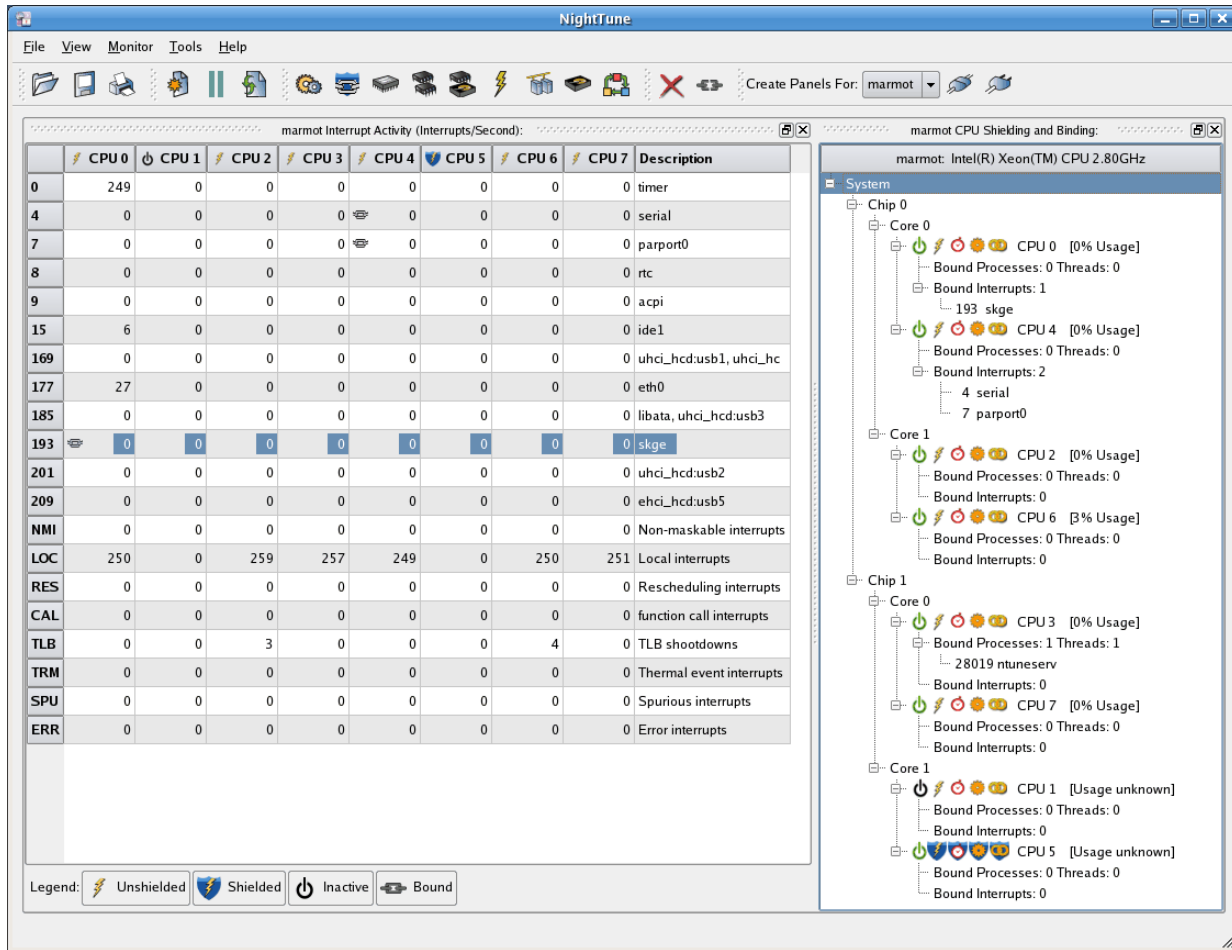


Figure 4-17. Using Drag and Drop to Change Interrupt CPU Affinity

See “Interrupt Activity Panel” on page 3-19 for more information on interrupt CPU affinity.

# NightStar Licensing

NightStar RT uses the NightStar License Manager (NSLM) to control access to the NightStar RT tools.

License installation requires a licence key provided by Concurrent (see “License Keys” on page A-1). The NightStar RT tools request a licence (see “License Requests” on page A-2) from a license server (see “License Server” on page A-2).

Two license modes are available, fixed and floating, depending on which product option you purchased. Fixed licenses can only be served to NightStar RT users from the local system. Floating licenses may be served to any NightStar RT user on any system on a network.

Tools are licensed per system, per concurrent user. A single license is shared among any or all of the NightStar RT tools for a particular user on a particular system. The intent is to allow  $n$  developers to fully utilize all the tools at the same time while only requiring  $n$  licenses. When operating the tools in remote mode, where a tool is launched on a local system but is interacting with a remote system, licenses are required only from the host system.

You can obtain a license report which lists all licenses installed on the local system, current usage, and expiration date for demo licenses (see “License Reports” on page A-3).

The default configuration includes a strict firewall which interferes with floating licenses. See “Firewall Configuration for Floating Licenses” on page A-3 for information on handling such configurations.

See “License Support” on page A-4 for information on contacting Concurrent for additional assistance with licensing issues.

## License Keys

Licenses are granted to specific systems to be served to either local or remote clients, depending on the license model, fixed or floating.

License installation requires a license key provided by Concurrent. To obtain a license key, you must provide your system identification code. The system identification code is generated by the `nslm_admin` utility:

```
nslm_admin --code
```

System identification codes are dependent on system configurations. Reinstalling Linux on a system or replacing network devices may require you to obtain new license keys.

To obtain a license key, use the following URL:

<http://www.ccur.com/NightStarRTKeys>

Provide the requested information, including the system identification code. Your license key will be immediately emailed to you.

Install the license key using the following command:

```
nslm_admin --install=xxx-xxx-xxx-xxx-xxx
```

where `xxx-xxx-xxx-xxx-xxx` is the key included in the license acknowledgment email.

## License Requests

By default, the NightStar RT tools request a license from the local system. If no licenses are available, they broadcast a license request on the local subnet associated with the system's hostname.

You can control the license requests for an entire system using the `/etc/nslm.config` configuration file.

By default, the `/etc/nslm.config` file contains a line similar to the following:

```
:server @default
```

The argument `@default` may be changed to a colon-separated list of system names, system IP addresses, or broadcast IP addresses. Licenses will be requested from each of the entities found in the list, until a license is granted or all entries in the list are exhausted.

For example, the following setting prevents broadcast requests for licenses, by only specifying the local system:

```
:server localhost
```

The following setting requests a license from `server1`, then `server2`, and then a broadcast request if those fail to serve a license:

```
:server server1:server2:192.168.1.0
```

Similarly, you can control the license requests for individual invocations of the tools using the `NSLM_SERVER` environment variable. If set, it must contain a colon-separated list of system names, system IP addresses, or broadcast IP addresses as described above. Use of the `NSLM_SERVER` environment variable takes precedence over settings defined in `/etc/nslm.config`.

## License Server

The NSLM license server is automatically installed and configured to run when you install NightStar RT.

The `nslm` service is automatically activated for run levels 2, 3, 4, and 5. You can check on these settings by issuing the following command:

```
/sbin/chkconfig --list nslm
```

In rare instances, you may need to restart the license server via the following command:

```
/sbin/service nslm restart
```

See `nslm(1)` for more information.

## License Reports

A license report can be obtained using the `nslm_admin` utility.

```
nslm_admin --list
```

lists all licenses installed on the local system, current usage, and expiration date (for demo licenses). Use of the `--verbose` option also lists individual clients to which licenses are currently granted.

Adding the `--broadcast` option will list this information for all servers that respond to a broadcast request on the local subnet associated with the system's hostname.

See `nslm_admin(1)` for more options and information.

## Firewall Configuration for Floating Licenses

RedHawk does not support a firewall configuration by default, because iptables support is disabled. However, it is possible to build a custom kernel with iptables support enabled. If that is done, and floating licenses are used, the iptables firewall rules must be configured to allow the license requests and responses to pass.

If the system with iptables support and firewall rules is serving licenses, then the firewall rules must be arranged to allow license requests on UDP port 25517 and TCP port 25517 from any systems that will make license requests. For example, in a simple firewall, rules like the following, inserted before any DROP or REJECT rules, might work:

```
iptables -A INPUT -p udp -m udp -s subnet/mask --dport 25517 -j ACCEPT
iptables -A INPUT -p tcp -m tcp -s subnet/mask --dport 25517 -j ACCEPT
```

If the system with iptables support and firewall rules is running NightStar RT tools and receiving floating licenses, then the firewall rules must be arranged to allow license responses on UDP port 25517 from any system serving licenses. For example, in a simple firewall, rules like the following, inserted before any DROP or REJECT rules, might work:

```
iptables -A INPUT -p udp -m udp -s subnet/mask --sport 25517 -j ACCEPT
```

## License Support

For additional aid with licensing issues, contact the Concurrent Software Support Center at our toll free number 1-800-245-6453. For calls outside the continental United States, the number is 1-954-283-1822. The Software Support Center operates Monday through Friday from 8 a.m. to 5 p.m., Eastern Standard Time.

You may also submit a request for assistance at any time by using the Concurrent Computer Corporation web site at [http://www.ccur.com/isd\\_support\\_contact.asp](http://www.ccur.com/isd_support_contact.asp) or by sending an email to [support@ccur.com](mailto:support@ccur.com).

## Kernel Dependencies

---

Concurrent's RedHawk kernel provides features and performance gains that are critical for the optimal operation of the NightStar RT tools.

The NightStar RT tools can operate in a host-only mode on Red Hat systems without Concurrent's RedHawk kernel, cross-targeting to RedHawk systems.

Additionally, the NightStar RT tools can function on Red Hat systems without the RedHawk kernel, but will lack the numerous advantages afforded by running with it.

The following sections describe the additional functionality and capabilities of the NightStar RT tools when running Concurrent's RedHawk kernel

### Advantages for NightView

The following advantages are afforded NightView when Concurrent's RedHawk kernel is running:

- Application speed conditions

Provides "execution-speed" patches, conditions, and ignore counts.

- Signal handling

Allows NightView to pass signals directly to a particular process, avoiding context switching.

### Advantages for NightTrace

The following advantage is afforded NightTrace when Concurrent's RedHawk tracing kernel is running:

- Kernel tracing

Users of NightTrace gain the ability to obtain kernel trace data and combine that with user trace data. Kernel tracing is an incredibly powerful feature that not only provides insight into the operating system kernel but also provides useful information relating to the execution of user applications.

The RedHawk kernel is provided in three flavors:

- Tracing

- Debug
- Plain

The Tracing and Debug flavors provide the features required for NightTrace kernel tracing. These kernels can be selected at boot-time from the boot-loader menu.

## Advantages for NightProbe

The following advantages are afforded NightProbe when Concurrent's RedHawk a RedHawk or SLERT kernel is running:

- Minimal intrusion

Allows NightProbe to read and write variables without stopping the process for each sample or write operation.

- Sampling performance

Allows NightProbe to use direct memory fetches for data sampling (as opposed to programmed I/O) which is important for high-rate data acquisition.

- Concurrent debugging/probing

Allows NightProbe to probe programs already under the control of a debugger or another NightProbe session.

- PCI Device probing

Allows NightProbe to probe PCI device memory via the Base Address Register (BAR) file system.

## Advantages for NightTune

The following advantage is afforded NightTune when Concurrent's RedHawk a RedHawk or SLERT kernel is running:

- Context switch rate

Allows NightTune user to display the context switch counts per CPU instead of for the overall system.

- CPU shielding

Individual CPUs can be shielded from interrupts and processes allowing CPUs to be dedicated solely to specific interrupts and processes that are bound to the CPU.



- CPU sibling interference

Individual CPUs can be marked down to avoid interfering with hyperthreaded sibling CPUs and dual-core sibling CPUs. Hyperthreaded CPUs share all the resources of their sibling CPU. Dual-core CPUs share the CPU cache and a path to memory with their sibling CPU.

- Detailed memory information

Detailed process memory descriptions include the residency and lock state of any page in a process, and their association with physical memory pools for NUMA systems.

## **Frequency Based Scheduler**

The Frequency Based Scheduler is only available on RedHawk systems from Concurrent Computer Corporation. It is required for all NightSim usage.

NightSim is only included in NightStar distributions intended for use on RedHawk systems.

## **PCI Bar File System**

The PCI Bar File System is only available with the RedHawk kernel from Concurrent Computer Corporation and SLERT versions 1.0-1.6 kernel from Novell.

On other systems, PCI Device probing will be disabled within NightProbe.



## A

Add Page 2-4, 2-12

## B

Bound Interrupts 3-7, 3-24  
Bound Processes 3-7  
Bound to System 2-18

## C

Capabilities 1-2, 3-70  
Changing

- CPU Shielding 3-8
- Interrupt CPU Affinity 3-24, 4-14
- Scheduling Attributes 3-55, 4-6

Command Line Options 1-4  
Config File

- Current Name 2-16
- Load 2-2, 2-12
- Save 2-2, 2-12
- Save As 2-3

Connect to a Remote System 2-13, 3-77  
Context Switches Panel

- Bar Graph Pane 3-4
- Line Graph Pane 3-5
- Menu Selection 2-7
- Text Pane 3-4
- Toolbar Icon 2-13

CPU Affinity 3-24, 3-58, 3-76, 4-6, 4-7, 4-14  
CPU Monitoring 1-1  
CPU Shielding and Binding Panel 3-7

- CPU Shielding Dialog 3-8
- Drag and Drop Operations 3-10
- Menu Selection 2-7
- Pop-up Menu 3-11
- Refresh Intervals 2-17
- Toolbar Icon 2-12

CPU Shielding Dialog 3-8

CPU Shielding Operations 3-8  
CPU Time 3-75  
CPU Usage Panel

- Bar Graph Pane 3-14
- Line Graph Pane 3-15
- Menu Selection 2-7
- Text Pane 3-13
- Toolbar Icon 2-12

Create panels for 2-13

## D

Data Memory Size 3-75  
Debug 3-55  
Delete Current Page 2-5  
Disconnect from a Remote System 2-13, 3-77  
Disk Activities

- Average Service Time 3-17
- Average Wait Time 3-17
- Read & Write Operations 3-16
- Sector Transfers 3-16

Disk Activity Panel

- Bar Graph Pane 3-17
- Line Graph Pane 3-18
- Menu Selection 2-7
- Text Pane 3-16
- Toolbar Icon 2-13

Drag and Drop 3-48, 3-56, 4-7, 4-17

## E

Environment variable

- NSLM\_SERVER A-2

Environment Variables 3-72  
Exit NightTune 2-4

## F

File Descriptors 3-66

File Menu 2-1, 2-2  
Filter Processes Dialog 3-52  
Fixed licenses A-1  
Floating licenses A-1  
Freeze Page 2-5, 2-12

## G

Guide to Operations 4-1

## H

Help Menu 2-1, 2-10  
Hyper-threading 3-9, 4-11

## I

Idle Time 3-14  
Interrupt Activity Panel  
    Affinity Dialog 3-24, 4-15  
    Bar Graph Pane 3-21  
    Drag and Drop Operations 3-20  
    Interrupt Affinity Dialog 4-16  
    Line Graph Pane 3-22  
    Menu Selection 2-7  
    Pop-up Menu 3-23  
    Text Pane 3-19  
    Toolbar Icon 2-13  
Interrupt Affinity 3-24, 4-14  
Interrupt Affinity Dialog 3-24, 4-16  
Interrupts, Bound 3-7  
IRQ Balancing 1-5

## K

Kernel Virtual Memory 3-28  
Kernel Virtual Memory Panel  
    Bar Graph Pane 3-29  
    Line Graph Pane 3-29  
    Menu Selection 2-7  
    Text Pane 3-28  
Kill Process 2-13

## L

Library Calls, trace 3-54  
License A-1  
    fixed A-1  
    installation A-1  
    keys A-1  
    modes A-1  
    nslm\_admin A-1, A-3  
    report A-3  
    requests A-2  
    server A-2  
    support A-4  
License manager 1-2  
licenses 1-2  
Load Config File 2-2, 2-12  
Load System Default Config 2-2  
Local Operation 1-2  
Logging 1-2, 1-5, 2-3, 2-16  
ltrace 3-54

## M

Memory  
    Kernel Virtual 3-28  
    NUMA 3-42  
    Physical 3-30  
    Swap 3-34  
Memory Activity Panel  
    Bar Graph Pane 3-27  
    Line Graph Pane 3-27  
    Menu Selection 2-7  
    Text Pane 3-26  
Menu Bar 2-1  
    File 2-1, 2-2  
    Help 2-1, 2-10  
    Monitor 2-1, 2-6  
    Tools 2-1, 2-9  
    View 2-1, 2-4  
Monitor Menu 2-1, 2-6  
Monitoring  
    CPU Status 1-1, 3-13  
    Processes 1-1, 3-46, 4-2  
    System Activities 1-1  
Most Recent CPU 3-76

## N

Network Activity

- Collision Rate 3-37
  - Input Packet Rate 3-36
  - Output Packet Rate 3-37
  - Packet Error Rate 3-37
  - Network Activity Panel
    - Bar Graph Pane 3-37
    - Line Graph Pane 3-38
    - Menu Selection 2-8
    - Text Pane 3-36
    - Toolbar Icon 2-13
  - New Page 2-4, 2-12
  - Nice Value 3-57, 3-76, 4-6
  - NightProbe Monitor 2-9
  - NightSim Scheduler 2-9
  - NightStar Licence Manager 1-2
  - NightTrace Analyzer 2-10
  - NightView 3-55
  - NightView Debugger 2-10
  - NLSM 1-2
  - nslm\_admin A-1, A-3
  - NSLM\_SERVER A-2
  - NUMA Activity Panel
    - Bar Graph Pane 3-44
    - Line Graph Pane 3-44
    - Menu Selection 2-8
    - Text Pane 3-43
  - NUMA Configuration Panel
    - Context Menu 3-46
    - CPUs Pane 3-45
    - Distance Pane 3-45
    - Menu Selection 2-8
  - NUMA Panel
    - Bar Graph Pane 3-41
    - Line Graph Pane 3-42
    - Menu Selection 2-8
    - Text Pane 3-39
  - NUMA Systems 3-39, 3-42, 3-45
- O**
- Options 1-4
- P**
- Page
    - Add 2-4, 2-12
    - Freeze 2-5, 2-12
    - Refresh 2-5, 2-12
  - Page Transfer Rates 3-25
  - Panels 3-1
  - Context Switches 3-4
  - CPU Shielding and Binding 3-7
  - CPU Usage 3-13
  - Disk Activity 3-16
  - Interrupt Activity 3-19, 4-14
  - Kernel Virtual Memory 3-28
  - Memory Activity 3-25
  - Network Activity Table 3-36
  - NUMA 3-39
  - NUMA Activity 3-42
  - NUMA Configuration 3-45
  - Physical Memory 3-30
  - Process List 3-46, 4-2
  - Swap 3-34
  - Parent Process ID 3-74
  - Physical Memory Allocation 3-30
  - Physical Memory Panel
    - Bar Graph Pane 3-32
    - Line Graph Pane 3-33
    - Menu Selection 2-8
    - Text Pane 3-30
    - Toolbar Icon 2-12
  - Preferences 2-2, 2-3, 2-16
  - Print NightTune Window 2-3, 2-12
  - Priority 3-76
  - Privileges 1-2
  - Process
    - Bound 3-7
    - Kill 2-13
    - Monitor 2-12
    - Unbind 2-13
  - Process Details
    - Capability Tab 3-70
    - Environment Tab 3-72
    - File Descriptors Tab 3-66
    - Memory Tab 3-62
    - Memory Usage Tab 3-59
    - Signals Tab 3-68
  - Process Fields
    - CPU Affinity 3-76
    - CPU Time 3-75
    - Customize 4-4
    - Data Memory Size 3-75
    - Most Recent CPU 3-76
    - Nice Value 3-76
    - Parent Process ID 3-74
    - Priority 3-76
    - Process ID 3-74
    - Real-Time Priority 3-76
    - Resident Memory Size 3-75
    - Scheduling Class 3-76
    - State 3-74
    - System Time 3-75
    - Threads 3-74

- User ID 3-74
- User Name 3-74
- User Time 3-75
- Virtual Memory Size 3-75
- Process ID 3-74
- Process List Panel 4-4
  - Context Menu 3-49
  - Drag and Drop 3-48
  - Filter Processes Dialog 3-52
  - Menu Selection 2-8
  - Process Details Window 3-59
  - Process Fields Menu 3-73
  - Process Scheduler Dialog 3-55, 3-58, 3-77
  - Refresh Intervals 2-17
  - Toolbar Icon 2-12
- Process Monitoring 1-1, 4-2
- Process Scheduler Dialog 3-55, 3-77
- Process Scheduling Operations 3-58, 4-6

## R

- Real-Time Priority 3-57, 3-76, 4-6
- Refresh Page 2-5, 2-12
- Remote Operation 1-2, 1-5, 2-13, 3-77
- Rename Current Page 2-4
- Resident Memory Size 3-75

## S

- Save Config File 2-2, 2-12
- Save Config File As 2-3
- Scheduling Class 3-56, 3-76
- Shield-All 3-9
- Shielding a CPU 4-10
- Signal Handling 3-68
- State 3-74
- strace 3-54
- Swap Memory Panel
  - Menu Selection 2-8
  - Toolbar Icon 2-13
- Swap Panel
  - Bar Graph Pane 3-35
  - Line Graph Pane 3-35
  - Text Pane 3-34
- System
  - Bind to Current 2-18
  - Connect 2-13, 3-77
  - Current 2-13, 3-78
  - Disconnect 2-13, 3-77
  - Target 1-5

- System Activity Monitoring 1-1
- System Calls, trace 3-54
- System Status Panel
  - Refresh Intervals 2-17
- System Time 3-14, 3-75
- Systems Menu 3-77

## T

- Tabbed Pages 2-1
  - Delete Current Page 2-5
  - New Page 2-4, 2-12
  - Rename Current Page 2-4
- Threads 3-74
- Time Quantum 3-57, 4-6
- Toolbar 2-12
- Tools Menu 2-1, 2-9
- Trace Library Calls 3-54
- Trace System Calls 3-54

## U

- Unbind Process 2-13
- User ID 3-74
- User Name 3-74
- User Time 3-14, 3-75
- Using NightTune 4-1

## V

- View Menu 2-1, 2-4
- Virtual Memory Size 3-75
- vmalloc 3-28

## W

- Wait Time 3-14
- Windows
  - Menu Bar 2-1
  - Multiple 2-1