Red Hat Enterprise MRG 2
Realtime Installation Guide

Installation information for the Realtime component of Red Hat Enterprise MRG

Lana Brindley Alison Young Cheryn Tan

Red Hat Enterprise MRG 2 Realtime Installation Guide

Installation information for the Realtime component of Red Hat Enterprise MRG

Lana Brindley
Red Hat Engineering Content Services

Alison Young
Red Hat Engineering Content Services

Cheryn Tan
Red Hat Engineering Content Services
cheryntan@redhat.com

Legal Notice

Copyright 2013 Red Hat, Inc. The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at . In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version. Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law. Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries. Linux is the registered trademark of Linus Torvalds in the United States and other countries. Java is a registered trademark of Oracle and/or its affiliates. XFS is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries. MySQL is a registered trademark of MySQL AB in the United States, the European Union and other countries. All other trademarks are the property of their respective owners. 1801 Varsity Drive Raleigh, NC 27606-2072 USA Phone: +1 919 754 3700 Phone: 888 733 4281 Fax: +1 919 754 3701

Keywords

Abstract

This book will show you how to download and install the MRG Realtime component of the Red Hat Enterprise MRG distributed computing platform. For detailed information on tuning MRG Realtime, see the MRG Realtime Tuning Guide.

Table of Contents

Preface

Red Hat Enterprise MRG

This book contains basic installation and tuning information for the MRG Realtime component of Red Hat Enterprise MRG. Red Hat Enterprise MRG is a high performance distributed computing platform consisting of three components:

1. *M*essaging — Cross platform, high performance, reliable messaging using the Advanced Message Queuing Protocol (AMQP) standard.
2. *R*ealtime — Consistent low-latency and predictable response times for applications that require microsecond latency.
3. *G*rid — Distributed High Throughput (HTC) and High Performance Computing (HPC).

All three components of Red Hat Enterprise MRG are designed to be used as part of the platform, but can also be used separately.

MRG Realtime

Many industries and organizations need extremely high performance computing and may require low and predictable latency, especially in the financial and telecommunications industries. Latency, or response time, is defined as the time between an event and system response and is generally measured in microseconds (μs). For most general applications running under a Linux environment, basic performance tuning can improve latency sufficiently. For those industries where latency not only needs to be low, but also accountable and predictable, Red Hat have now developed a 'drop-in' kernel replacement that provides this. MRG Realtime is distributed as part of Red Hat Enterprise MRG and provides seamless integration with Red Hat Enterprise Linux 6. MRG Realtime offers clients the opportunity to measure, configure and record latency times within their organization.

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the Liberation Fonts set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keys and key combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a key, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from an individual key by the plus sign that connects each part of a key combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F2`** to switch to a virtual terminal.

The first example highlights a particular key to press. The second example highlights a key combination: a set of three keys pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`mono-spaced bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

> Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).
>
> To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find…** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

> To connect to a remote machine using ssh, type **`ssh`** ***username@domain.name*** at a shell prompt. If the remote machine is **`example.com`** and your username on that machine is john, type **`ssh john@example.com`**.
>
> The **`mount -o remount`** ***file-system*** command remounts the named file system. For example, to remount the **`/home`** file system, the command is **`mount -o remount /home`**.
>
> To see the version of a currently installed package, use the **`rpm -q`** ***package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

> Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **`mono-spaced roman`** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **`mono-spaced roman`** but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }
}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. We Need Feedback

If you find a typographical error in the *MRG Realtime Installation Guide*, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: http://bugzilla.redhat.com/ against the product **Red Hat Enterprise MRG** version *2.3*.

When submitting a bug report, be sure to mention the manual's identifier: *Realtime_Installation_Guide*.

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

Chapter 1. Why Use MRG Realtime to Optimize Latency?

For anyone considering evaluating the performance benefits of the MRG Realtime kernel, it is crucial to understand both the importance of tuning and how to establish the right performance expectations.

MRG Realtime is designed to be used on well-tuned systems for applications with extremely high determinism requirements. Kernel system tuning offers the vast majority of the improvement in determinism. For example, in many workloads thorough system tuning improves consistency of results by around 90%. This is why we typically recommend that customers first perform the suggested system tuning of standard Red Hat Enterprise Linux to see if it meets their objectives, before using MRG Realtime.

System tuning is just as important when using the MRG Realtime kernel as it is for standard Red Hat Enterprise Linux. In fact, if you simply took an untuned system running standard Red Hat Enterprise Linux and substitute the MRG Realtime kernel for the stock kernel supplied as part of the Red Hat Enterprise Linux release, you are unlikely to notice any benefit. Standard tuning will yield 90% of the determinism gains. The MRG Realtime kernel provides the last 10% of determinism required by the most demanding workloads.

Establishing the right performance expectations refers to the fact that the MRG Realtime kernel is not a panacea. Its objective is consistent, low-latency determinism offering predictable response times. There is some additional kernel overhead associated with the MRG Realtime kernel. This is due primarily to handling hardware interrupts in separately scheduled threads. The increased overhead in some workloads results in some degradation in overall throughput. The exact amount is very workload dependent, ranging from 0% to 30%. However, it is the cost of determinism.

For typical workloads with kernel latency requirements in the millisecond (ms) range, then the standard Red Hat Enterprise Linux 6 kernel is sufficient. If your workload has stringent low-latency determinism requirements for core kernel features such as interrupt handling and process scheduling in the microsecond (μs) range, then the MRG Realtime kernel is for you.

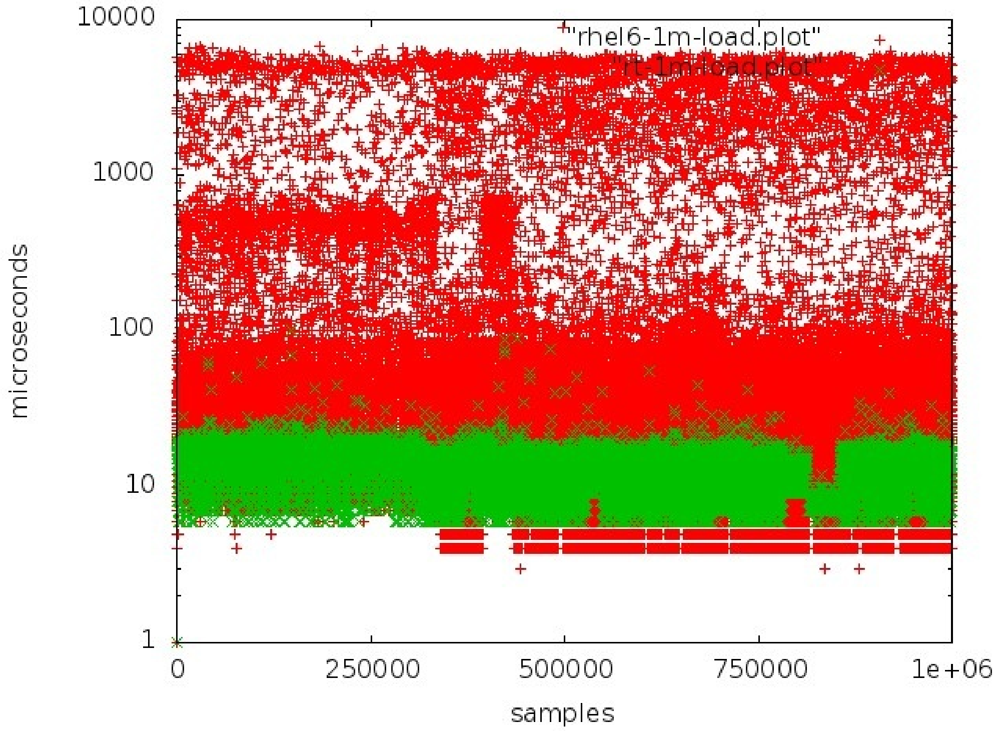

Figure 1.1. Benefit of using MRG Realtime over standard kernel system tuning

This graph compares a million samples of machines using the Red Hat Enterprise Linux 6 and MRG Realtime kernels respectively. The red points in this graph represent the system response time (in microseconds) of machines running a tuned Red Hat Enterprise Linux 6 kernel. The green points in the graph represent the system response time of machines running a tuned MRG Realtime kernel. It is clear from this graph that the response time of the MRG Realtime kernel is very consistent, in contrast to the Red Hat Enterprise Linux 6 case which has greater variability with points scattered across the graph.

Chapter 2. Download

System Requirements

A prerequisite of installing the MRG Realtime kernel is that the user or administrator must first perform a fresh installation of Red Hat Enterprise Linux 6. It is important to note, however:

- A full system installation of Red Hat Enterprise Linux 6 is required.
- Red Hat Enterprise Linux 5 is not supported for use with MRG Realtime.
- When doing the initial operating system installation, select the set of packages that meet the needs of your application environment. Nothing MRG Realtime specific is required on the initial install.

Once you have installed Red Hat Enterprise Linux 6 follow the instructions given in Section 2.2, “Installing MRG Realtime Using Yum” to add the MRG Realtime specific packages from a separate yum repository.

Differences Between MRG Realtime and the Standard Kernel

MRG Realtime differs substantially from the standard Red Hat Enterprise Linux 6 kernel.

- *Third-party kernel modules are incompatible with standard Red Hat Enterprise Linux 6*
  - Kernel modules are inherently specific to the kernel they are built for. Since the MRG Realtime kernel is substantially different from the standard Red Hat Enterprise Linux 6 kernel, kernel modules are incompatible. In other words, you can't take third-party driver modules from Red Hat Enterprise Linux 6 and use them as-is on MRG Realtime.
  - The following are some example third-party drivers which ship for standard Red Hat Enterprise Linux 6 which do not currently have a MRG Realtime build:
    - EMC Powerpath
    - NVidia graphics
    - Advanced storage adapter configuration utilities from Qlogic

Important

The user space **`syscall`** interface is compatible with standard Red Hat Enterprise Linux 6. These compatibility restrictions pertain *only* to kernel modules not supplied by Red Hat.

2.1. MRG Realtime Kernel Variants

There are numerous kernel variants provided. Each variant is simply a version of the MRG Realtime kernel compiled with support for different configuration options. In this case, the variants offer differing diagnostic capabilities. The set of kernels is provided for x86_64 (64-bit) systems. As of MRG 2.0, i686 (32-bit) kernels are no longer supported.

The main deployment MRG Realtime kernel is identified below as *Production*. Additionally there are several debug kernels which have progressively more diagnostic code compiled in. The reason for doing this is that as the amount of debug code is increased, so does the overhead. **`kernel-rt-trace`** has less overhead than **`kernel-rt-debug`**.

Finally, the **`Vanilla`** kernel does not include the MRG Realtime features. This is used to help distinguish whether bugs were introduced in the MRG Realtime features, or are inherent bugs in the baseline kernel. See Section 4.1, “Reporting Bugs” for more information.

Table 2.1. MRG Realtime Kernel Variants for AMD64 and Intel 64 systems

Variant	Intended Usage	Notes
x86_64 (kernel-rt)	*Production* - Standard 64-bit production kernel	
x86_64 tracing (kernel-rt-tracing)	*Debugging* - 64-bit trace kernel	Latency tracer enabled - Used to locate latency hotspots
x86_64 debug (kernel-rt-debug)	*Debugging* - 64-bit debugging kernel	Includes debugging options, with latency tracer disabled - Used to debug the MRG Realtime kernel
x86_64 vanilla (kernel-rt-vanilla)	*Debugging* - 64-bit base kernel	No MRG Realtime features, used for comparison

2.2. Installing MRG Realtime Using Yum

In order to install MRG Realtime you will need to have registered your system with Red Hat Network. This table lists the Red Hat Enterprise MRG channels available on Red Hat Network for MRG Realtime.

Table 2.2. Red Hat Enterprise MRG Realtime channels available on Red Hat Network

Channel Name	Operating System	Architecture
Red Hat Enterprise MRG Realtime	RHEL-6 Server	32-bit, 64-bit

Important

Before you install Red Hat Enterprise MRG check that your hardware and platform is supported. A complete list is available on the Red Hat Enterprise MRG Supported Hardware Page.

Procedure 2.1. To install MRG Realtime components

1. Use the **yum** command to install the MRG Realtime group:

   ```
   # yum groupinstall "MRG Realtime"
   ```

   The **MRG Realtime** group installs five packages:

   - **rt-setup** sets up the basic environment required by MRG Realtime.
   - **kernel-rt** is the standard MRG Realtime kernel package.
   - **rtctl** is a startup script that sets the priorities of the various kernel threads.
   - **rtcheck** is a program that tests the running system for MRG Realtime capabilities.
   - **tuna** is a graphical tool used to manage your MRG Realtime application. For information on running Tuna, see the *MRG Realtime Tuning Guide*.

2. You can check the installation location and that the components have been installed successfully by using the **rpm -ql** command.

```
# rpm -ql rt-setup
/etc/security/limits.d/realtime.conf

# rpm -ql kernel-rt
```

Note

See Section 4.2, “Further Reading”, for places to turn for help if you have trouble with installing the MRG Realtime kernel.

2.3. Installing MRG Realtime Using Diskless Boot

Diskless booting allows a system administrator to deploy a large number of compute nodes without having to configure an equal number of local storage devices. This section provides instructions on how to set up a remote diskless system using an NFS filesystem mounted by a PXE booting client. Perform the following procedure on an existing installation of MRG Realtime.

Procedure 2.2. To set up diskless boot for MRG Realtime

1. Install the following packages required to set up a basic remote diskless system booted over PXE.

   ```
   # yum install tftp-server xinetd dhcp syslinux dracut-network -y
   ```

   Important

   - Ensure that the NFS and DHCP services are properly configured and started.
   - Firewall rules which have been predefined by **system-config-firewall** can block **tftp** traffic and prevent the clients from booting. It is necessary to adjust the firewall rules on the server.

2. Enable the **tftp** service to allow PXE booting via the network. To do so, edit the **/etc/xinetd.d/tftp** file and set the ***Disabled*** parameter to ***No***.
3. Copy **/usr/share/syslinux/pxelinux.0** to the **tftp** root directory in **/var/lib/tftpboot**.

   ```
   # cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/
   ```

4. Create a **pxelinux.cfg** directory inside the **tftp** root directory.

   ```
   # mkdir -p /var/lib/tftpboot/pxelinux.cfg/
   ```

5. Configure firewall rules to allow **tftp** traffic. Using a text editor, add the following line to **/etc/hosts.allow**:

   ```
   tftpd: .hostname.com
   ```

   Replace ***.hostname.com*** with the domain name of the target clients on which the installations will be performed. For more information on firewall rules refer to the man page for **hosts_access**.
6. Restore the SELinux context on the **tftp** server. To provide the necessary files via the tftp protocol for PXE boot, run the following command:

```
# restorecon -R /var/lib/tftpboot
```

7. To enable PXE booting on the DHCP server, add the following configuration to **`/etc/dhcp/dhcpd.conf`**.

   ```
   allow booting;
   allow bootp;
   class "pxeclients" {
      match if substring(option vendor-class-identifier, 0, 9) = "PXEClient";
      next-server server-ip;
      filename "pxelinux.0";
   }
   ```

   Replace ***`server-ip`*** with the IP address of the host machine on which the **`tftp`** and DHCP services reside.

8. The root directory of the exported file system (used by diskless clients in the network) is shared via NFS. Configure the NFS service to export the root directory by adding it to the **`/etc/exports`** file in the following format:

   ```
   /exported/root/directory hostname.com(rw,sync,no_root_squash)
   ```

   The example configuration exports the directory ***`/export/root/directory`*** to all hosts on the ***`hostname.com`*** network with ***`read-write`*** permissions. The ***`sync`*** option ensures that the NFS server will not reply to requests before changes made by previous requests are written to disk. The ***`no_root_squash`*** option enables the NFS server to connect to the client with root privileges.

9. To accommodate completely diskless clients, the root directory should contain a complete MRG Realtime installation. You can synchronize this with a running system via **rsync**. Run:

   ```
   # rsync -a -e ssh --exclude='/proc/*' --exclude='/sys/*' hostname.com:/
   /exported/root/directory
   ```

   Replace ***`hostname.com`*** with the hostname of the running system with which to synchronize via rsync. The ***`/exported/root/directory`*** is the path to the exported file system.

10. Add the following configuration to the exported file system's **`/etc/fstab`** file:

    ```
    none   /tmp   tmpfs defaults 0 0
    tmpfs  /dev/shm tmpfs defaults 0 0
    sysfs  /sys  sysfs defaults 0 0
    proc  /proc  proc  defaults 0 0
    ```

11. Select the kernel that diskless clients should use (**`vmlinuz-`*`rt-kernel-version`***) and copy it to the **`tftpboot`** directory:

    ```
    # cp /boot/vmlinuz-rt-kernel-version /var/lib/tftpboot/
    ```

12. Create the **`initramfs`** disk image (i.e. **`initramfs-`*`rt-kernel-version`*`.img`**) with network support.

    ```
    # dracut -a "network nfs" initramfs-rt-kernel-version.img rt-kernel-version
    ```

    The MRG Realtime kernel modules must be available for the kernel version **`dracut`** is run with. For instance, if you create the **`initramfs`** image for the MRG Realtime kernel on the tftp server, the *kernel-rt* and *kernel-rt-firmware* packages must be installed on the tftp server. This kernel does not need to run, but the files must be available when running the **dracut** command.

13. Copy the **initramfs** disk image to the **tftpboot** directory.

    ```
    # cp initramfs-rt-kernel-version.img /var/lib/tftpboot/
    ```

14. Edit the default boot configuration to use the **initramfs** disk image and Realtime kernel inside **/var/lib/tftpboot**. This configuration instructs the diskless client to mount the exported file system (**/exported/root/directory**) as read-write. To do this, configure **/var/lib/tftpboot/pxelinux.cfg/default** with the following:

    ```
    default realtime

    label realtime
      kernel vmlinuz-rt-kernel-version
      append initrd=initramfs-rt-kernel-version.img root=nfs:server-
    ip:/exported/root/directory rw
    ```

    Replace ***server-ip*** with the IP address of the host machine on which the exported NFS root partition resides.

The NFS share is now ready for exporting to diskless clients. These clients can boot MRG Realtime over the network via PXE.

Important

You must disable SELinux on diskless clients which use NFS as the root file system. Add ***SELINUX=disabled*** to the **/etc/selinux/config** file, or add ***selinux=0*** on the kernel command line when booting the client.

Note

For more information on setting up remote diskless systems, see the *Red Hat Enterprise Linux 6 Storage Administration Guide*.

2.4. Available Packages — RPM

This section lists the RPM packages available in the repository for MRG Realtime.

The **MRG Realtime Specific** column indicates if the RPM differs from the standard Red Hat Enterprise Linux 6 maintenance stream, or is not applicable to the standard kernel. A No in this column indicates that the RPM performs equivalently on Red Hat Enterprise Linux 6.

The **Required** column indicates whether or not the package is mandatory for correct MRG Realtime behavior. A No in this column indicates that usage is optional.

Table 2.3. Basic MRG Realtime Kernel Packages

RPM Package Name	Description	MRG Realtime Specific?	Required?
kernel-rt	Low latency and preemption functionality	Yes	Yes
rtctl	System start-up script used to configure the default MRG Realtime scheduling priorities of kernel threads	Yes	Yes
kernel-rt-firmware	Firmware files loaded by some hardware drivers to enable them	Yes	Yes
kernel-rt-doc	MRG Realtime documentation	Yes	Advised

The following packages contain test programs for use with MRG Realtime.

Table 2.4. MRG Realtime Test Packages

RPM Package Name	Description
kernel-rt-devel	Headers and libraries for kernel development
kernel-rt-trace	MRG Realtime kernel with tracing functions compiled in
kernel-rt-trace-devel	Headers and libraries for development on trace kernel
kernel-rt-debug	MRG Realtime kernel with debugging functions compiled in (slow)
kernel-rt-debug-devel	Headers and libraries for development on debug kernel
kernel-rt-vanilla	Base kernel for comparisons
kernel-rt-vanilla-devel	Headers and libraries for development on vanilla kernel
rt-tests	Utilities for measuring system latencies and for proving that priority-inheritance mutexes function properly
rtcheck	A program that tests the running system for MRG Realtime capabilities
perf	Performance monitoring for the Linux kernel

The following set of packages are provided for use with **oprofile**, **systemtap** and the crash utility for analyzing kernel crashdumps. The debugging packages consist of symbol tables and are quite large. For this reason, they are separately delivered from the other MRG Realtime packages.

These packages can be found and downloaded from the **Red Hat Enterprise MRG Realtime Debuginfo** channel on the Red Hat Network.

Table 2.5. MRG Realtime Debugging Packages

RPM Package Name	Description
`kernel-rt-debuginfo`	Symbols for profiling and debugging use, such as `oprofile` or `systemtap`
`kernel-rt-trace-debuginfo`	Symbols for profiling and tracing
`kernel-rt-debug-debuginfo`	Symbols for profiling and tracing
`kernel-rt-vanilla-debuginfo`	Symbols for profiling and tracing
`kernel-rt-debuginfo-common`	Common debuginfo files shared between the other debuginfo packages

Important

The packages in Table 2.4, “MRG Realtime Test Packages” and Table 2.5, “MRG Realtime Debugging Packages” are not essential in order to run MRG Realtime. They are provided as diagnostic tools only and should not be run as a matter of course. To do so will negatively impact performance and could render any benefit from the use of the MRG Realtime kernel negligible.

2.5. Post-Installation Instructions

The MRG Realtime kernel is not automatically specified as the default boot kernel during the installation process. The recommended approach after installing the kernel replacement is to reboot, then manually select the MRG Realtime kernel in the grub menu.

Procedure 2.3. To run MRG Realtime as the default kernel

1. Once you know that the MRG Realtime kernel is fully operational on your system you can modify **`grub.conf`** to make it the default boot kernel. The **`grub.conf`** file is located in **`/boot/grub/grub.conf`**. View the file using your preferred text editor. It should look similar to the following:

```
default=1
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux (realtime) (kernel-rtversion)
 root (hd0,0)
 kernel /vmlinuz-kernel-rtversion ro root=/dev/Root rhgb quiet
 initrd /initrd-kernel-rtversion.img
title Red Hat Enterprise Linux Server (kernel-version)
 root (hd0,0)
 kernel /vmlinuz-kernel-version ro root=/dev/Root rhgb quiet
 initrd /initrd-kernel-version.img
```

2. In the example, **`default=1`** indicates that the default kernel in use is the standard Red Hat Enterprise Linux kernel. To use the MRG Realtime kernel, change the value of **`default`** to ***`0`*** as follows and save your changes.

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux (realtime) (kernel-rtversion)
 root (hd0,0)
 kernel /vmlinuz-kernel-rtversion ro root=/dev/Root rhgb quiet
 initrd /initrd-kernel-rtversion.img
title Red Hat Enterprise Linux Server (kernel-version)
 root (hd0,0)
 kernel /vmlinuz-kernel-version ro root=/dev/Root rhgb quiet
 initrd /initrd-kernel-version.img
```

3. You will also need to edit **`/etc/sysconfig/kernel`**. This will ensure that the changes you made to the grub file will remain as the default when you perform a system upgrade. To edit this file, open it in any text editor. It should look similar to this:

   ```
   # UPDATEDEFAULT specifies if new-kernel-pkg should make
   # new kernels the default
   UPDATEDEFAULT=yes

   # DEFAULTKERNEL specifies the default kernel package type
   DEFAULTKERNEL=kernel
   ```

4. Simply change the **`DEFAULTKERNEL=`** parameter to read ***`kernel-rt`***.

   ```
   # UPDATEDEFAULT specifies if new-kernel-pkg should make
   # new kernels the default
   UPDATEDEFAULT=yes

   # DEFAULTKERNEL specifies the default kernel package type
   DEFAULTKERNEL=kernel-rt
   ```

5. You will now be able to confirm that your system is running the MRG Realtime kernel, by running the **`uname`** command as the root user at the shell prompt. Check the output for the ***RT*** designation. If it appears, the MRG Realtime kernel is running.

   ```
   [root@mrg-rt-server ~]# uname -a
   Linux mrg-rt-server.redhat.com kernel.number-rtversion.number.el6rt.x86_64
   #1 SMP PREEMPT RT Tue May 10 00:49:01 EDT 2011 x86_64 x86_64 x86_64
   GNU/Linux
   ```

Note

MRG Realtime can be configured to provide crash dump information by enabling **`kexec/kdump`**. Further information and instructions on how to configure your system to obtain kernel crash information can be found in the *MRG Realtime Tuning Guide*.

Chapter 3. MRG Realtime Kernel Tuning

The MRG Realtime kernel offers many performance tuning parameters not otherwise available in Red Hat Enterprise Linux 6. In order to achieve optimal low-latency determinism it is necessary to perform MRG Realtime specific system tuning.

> **Note**
>
> For comprehensive tuning information, see the *MRG Realtime Tuning Guide*.

Tuna

The primary diagnostic facility provided with the MRG Realtime kernel is Tuna. Tuna provides both a command line tool and a graphical interface that can be used to change attributes of threads (scheduling policy, scheduler priority and processor affinity) and interrupts (processor affinity). The tool is designed to be used on a running system, and changes take place immediately. This allows any application-specific measurement tools to see and analyze system performance immediately after the changes have been made.

Diagnostic Tools

The MRG Realtime kernel provides several diagnostic tools to test and report on latencies.

The **`latency tracer`** is a peak detector which is used to identify the longest running non-preemptable kernel codepaths. This is particularly useful for identifying whether non-deterministic performance results are attributable to the kernel or to user-space components. In customer deployments, the tool is most useful to differentiate whether delays are in the kernel or the application.

The **`ftrace`** utility is used to analyze and debug latency and performance issues that occur outside of user-space. It has a variety of options that allow you to use the utility in a number of different ways. It can be used to trace context switches, measure the time it takes for a high-priority task to wake up, measure the length of time for which interrupts are disabled, or list all the kernel functions executed during a given period.

Direct Memory Access with `rmem.ko`

MRG Realtime includes a kernel module called **`rmem.ko`**. This module is not loaded by default, and is provided only to meet the realtime Java conformance tests - specifically the Technology Compatibility Kit (TCK) test in the Real Time Specification for Java's (RTSJ) conformance suite. It is an RTSJ requirement of this conformance test that Java programs have direct access to physical memory. This **`/dev/rmem`** capability allows user applications to map any arbitrary memory region.

The capability can only be enabled as a result of direct root system administrator action. To further prevent accidental usage, the capability is disabled unless ***`unprotected_address_space=1`*** is specified as an option for the module in the **`grub.conf`** file. If the **`rmem.ko`** kernel module is explicitly loaded, the kernel will have a ***`tainted`*** flag. In this case, Red Hat Global Support may require you to reproduce any problems without the presence of **`rmem.ko`**.

Warning

Enabling **`rmem.ko`** gives any application direct access to physical memory. Many normal security mechanisms are bypassed and the system becomes much more vulnerable to attacks by malicious users. For this reason, it is strongly suggested that this feature be used *exclusively* for RTSJ certification purposes only. *DO NOT* use the **`rmem.ko`** module for production deployment.

There is another related boot option for this feature. This boot option reserves contiguous physical kernel memory at boot time for later usage by Java runtime. This memory is allocated at boot prior to the system memory becoming fragmented. The purpose of this is to avoid allocation failures for large contiguous memory. The line that would need to be added to the **`grub.conf`** file is the following, where the ***`memsize`*** parameter is expressed in bytes.

```
alloc_rtsj_mem.size=memsize
```

Chapter 4. More Information

4.1. Reporting Bugs

Diagnosing a Bug

Before you a file a bug report, follow these steps to diagnose where the problem has been introduced. This will greatly assist in rectifying the problem.

1. Check that you have the latest version of the Red Hat Enterprise Linux 6 kernel, then boot into it from the grub menu. Try reproducing the problem with the standard kernel. If the problem still occurs, report a bug against Red Hat Enterprise Linux 6 *not* MRG Realtime.
2. If the problem does not occur when using the standard kernel, then the bug is probably the result of changes introduced in either:
   a. The upstream kernel on which MRG Realtime is based. For example, Red Hat Enterprise Linux 6 is based on 2.6.32 and MRG Realtime is based on 3.6
   b. MRG Realtime specific enhancements Red Hat has applied on top of the baseline (3.6) kernel

   To determine the problem, try to reproduce the problem on an unmodified upstream 3.6 kernel. For this reason, in addition to providing the MRG Realtime kernel, we also provide a **`vanilla`** kernel variant. The **`vanilla`** kernel is the upstream kernel build without the MRG Realtime additions.

Reporting a Bug

If you have determined that the bug is specific to MRG Realtime follow these instructions to enter a bug report:

1. Create a Bugzilla account.
2. Log in and click on Enter A New Bug Report.
3. You will need to identify the product the bug occurs in. MRG Realtime appears under **Red Hat Enterprise MRG** in the Red Hat products list. It is important that you choose the correct product that the bug occurs in.
4. Continue to enter the bug information by designating the appropriate component and giving a detailed problem description. When entering the problem description be sure to include details of whether you were able to reproduce the problem on the standard Red Hat Enterprise Linux 6 or the supplied **`vanilla`** kernel.

4.2. Further Reading

Red Hat Enterprise MRG and MRG Realtime Product Information

http://www.redhat.com/mrg

MRG Realtime and other Red Hat Enterprise MRG manuals

http://docs.redhat.com/docs/en-US/index.html

Red Hat Knowledgebase

https://access.redhat.com/knowledge/search

Revision History

Revision 4-0 Wed Feb 27 2013 Cheryn Tan

Prepared for publishing (MRG 2.3)

Revision 3-10 Tue Feb 12 2013 Cheryn Tan

Removal of deferred features.

Revision 3-9 Mon Jan 21 2013 Cheryn Tan

SME review for diskless boot procedure.

Revision 3-7 Wed Dec 19 2012 Cheryn Tan

BZ#866858 - Kernel rebase to version 3.6.

Revision 3-5 Thu Dec 13 2012 Cheryn Tan

BZ#833634 - Edited diskless boot procedure.

Revision 3-0 Thu Jun 14 2012 Cheryn Tan

Prepared for publishing (MRG 2.2).

Revision 2-5 Fri Jun 1 2012 Cheryn Tan

BZ#805755 - Incorporated changes from tech review of diskless boot section.

Revision 2-4 Fri May 18 2012 Cheryn Tan

BZ#800226 - Updated required RHN channels for downloading MRG Realtime.

Revision 2-3 Fri Apr 27 2012 Cheryn Tan

BZ#800724 - Updated kernel version number.
BZ#748391 - Added RT vs RHEL6 system response time comparison graph.
BZ#805755 - Documented setting up diskless boot

Revision 2-2 Thu Mar 15 2012 Cheryn Tan

BZ#748391 - Removed RHEL 5.6 from system requirements.
Docs QE reviews

Revision 2-1 Tue Feb 28 2012 Tim Hildred

Updated configuration file for new publication tool.

Revision 2-0 Wed Dec 7 2011 Alison Young

Prepared for publishing

Revision 1-6 Wed Nov 16 2011 Alison Young

BZ#752406 - change RHEL versions

Revision 1-2 Mon Oct 03 2011 Alison Young

BZ#694596 - x86 to i686 update

Revision 1-1 Thu Sep 22 2011 Alison Young

Version numbering change

Revision 1-0 **Thu Jun 23 2011** **Alison Young**

Prepared for publishing

Revision 0.1-7 **Fri Jun 03 2011** **Alison Young**

Removed RHEL4 refs

Revision 0.1-6 **Thu May 19 2011** **Alison Young**

Minor updates

Revision 0.1-5 **Mon May 16 2011** **Alison Young**

Minor update

Revision 0.1-4 **Mon May 16 2011** **Alison Young**

Technical Review updates

Revision 0.1-3 **Wed May 11 2011** **Alison Young**

BZ#694596 - Removed information about 32-bit kernels

Revision 0.1-2 **Wed Apr 20 2011** **Alison Young**

BZ#694596 - Remove information about PAE enabled kernels

Revision 0.1-1 **Thu Mar 03 2011** **Alison Young**

BZ#666960 - Update MRG Realtime Installation Guide for RHEL6

Revision 0.1-0 **Tue Feb 22 2011** **Alison Young**

Fork from 1.3