MRG Grid

1.0

Deployment Guide

ISBN:
Publication date:

This book contains basic overview and installation procedures for the MRG Grid component of the Red Hat Enterprise MRG distributed computing platform.

MRG Grid: Deployment Guide

Copyright © 2008 Red Hat, Inc

The GPG fingerprint of the security@redhat.com key is:

CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

1801 Varsity Drive
Raleigh, NC 27606-2072
USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588
Research Triangle Park, NC 27709
USA

Preface

Red Hat Enterprise MRG.

This book contains basic overview and installation information for the MRG Grid 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 Grid.

Grid computing allows organizations to fully utilize their computing resources to complete high-performance tasks. By monitoring all resources - rack-mounted clusters and general workstations - for availability, any spare computing power can be redirected towards other, more intensive tasks until it is explicitly required again. This allows a standard networked system to operate in a way that is similar to a supercomputer.

MRG Grid provides High Throughput and High Performance computing and enables enterprises to achieve higher peak computing capacity as well as improved infrastructure utilization by leveraging their existing technology to build high performance grids. MRG Grid provides a job queueing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management. Users submit their jobs to MRG Grid, where they are placed into a queue. MRG Grid then chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion.

MRG Grid is based on the *Condor Project* [http://www.cs.wisc.edu/condor/] developed within the *University of Wisconsin-Madison* [http://www.wisc.edu/]. Condor also offers a comprehensive library of freely available documentation in its *Manual* [http://www.cs.wisc.edu/condor/manual/].

1. Document Conventions

Certain words in this manual are represented in different fonts, styles, and weights. This highlighting indicates that the word is part of a specific category. The categories include the following:

`Courier font`

Courier font represents `commands`, `file names and paths`, and `prompts` .

When shown as below, it indicates computer output:

```
Desktop       about.html       logs      paulwesterberg.png
Mail          backupfiles      mail      reports
```

`bold Courier font`

Bold Courier font represents text that you are to type, such as: **`service jonas start`**

If you have to run a command as root, the root prompt (#) precedes the command:

```
# gconftool-2
```

`italic Courier font`

Italic Courier font represents a variable, such as an installation directory:
`install_dir``/bin/`

bold font

Bold font represents **application programs** and **text found on a graphical interface**.

When shown like this: **OK** , it indicates a button on a graphical application interface.

Additionally, the manual uses different strategies to draw your attention to pieces of information. In order of how critical the information is to you, these items are marked as follows:

Note

A note is typically information that you need to understand the behavior of the system.

Tip

A tip is typically an alternative way of performing a task.

Important

Important information is necessary, but possibly unexpected, such as a

configuration change that will not persist after a reboot.

Caution

A caution indicates an act that would violate your support agreement, such as recompiling the kernel.

Warning

A warning indicates potential data loss, as may happen when tuning hardware for maximum performance.

2. We Need Feedback!

If you find a typographical error in this manual, 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/bugzilla/* against the product **Red Hat Enterprise MRG.**

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

Overview

MRG Grid provides High Throughput and High Performance computing and enables enterprises to achieve higher peak computing capacity as well as higher IT utilization by leveraging their existing infrastructure and building high performance grids. High Throughput Computing (HTC) delivers large amounts of computing power over a sustained period of time (months or years), whereas High Performance Computing (HPC) delivers significant computing power over a short period of time (hours or days).

In the future, MRG Grid will leverage MRG Messaging and MRG Realtime to provide new capabilities:

- Optimized MRG Realtime scheduling with Quality of Service (QoS)
- Message-based job submission

MRG Grid enables enterprises and research organizations to bring the power of distributed computing across their entire infrastructure to tackle the largest computational problems in a highly efficient and effective manner.

Installing MRG Grid using Yum

The best strategy for installing MRG Grid components is to use the `mrg-beta` yum repository.

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* [http://www.redhat.com/mrg/hardware/].

1. Become the root user, then download the `mrg-beta` repository to `/etc/yum.repos.d`.

```
# cd /etc/yum.repos.d
# wget ftp://ftp.redhat.com/pub/redhat/linux/beta/MRG/RHEL-5/mrg-beta.repo
```

2. Once `mrg-beta.repo` exists in your local yum repository, you can view the list of available packages by using the `list available` command. This command will list *only* those packages not already installed on your system.

```
# yum --disablerepo='*' --enablerepo=mrg-beta list available
```

3. Install MRG Grid:

```
# yum install mrg-grid
```

4. You can check the installation location and that the components have been installed successfully by using the `rpm -ql` command. You will need to specify `condor` as the package name.

```
# rpm -ql condor
/etc/condor
/etc/condor/condor_config
/usr/bin/condor
/usr/bin/condor_check_userlogs
/usr/bin/condor_checkpoint
/usr/bin/condor_cod
/usr/bin/condor_compile
/usr/bin/condor_config_val
...
[output truncated]
```

5. You will now able to configure the software, and start the service. For detailed configuration instructions, see the *Condor Manual* [http://www.cs.wisc.edu/condor/manual/v7.0/3_3Configuration.html].

6. It is advised that you review the configuration file stored at `/etc/condor/condor_config` before attempting to start MRG Grid.

7. MRG Grid is usually run as a service. To start MRG Grid run the following command:

```
# service condor start
```

Tip

See *Further Reading*, for places to turn for help if you have trouble with installing MRG Grid.

MRG Grid Benefits and Features

Benefits.

MRG Grid provides significant benefits and value for enterprises, including:

Power
: MRG Grid can process the largest computational workloads, from massively parallel High Performance Computing jobs to long-running High Throughput Computing jobs

Peak Workload Handling
: MRG Grid can add on-demand computational power for handling peak loads through capabilities from cycle-stealing on Linux or Windows desktop computers and scheduling to remote grids

Flexibility
: MRG Grid provides complete flexibility, from running high-burst to lengthy computations, in a centralized or distributed grid, and running jobs on various platforms including Linux and Windows. Furthermore, MRG Grid can schedule virtualized environments and workloads for the upmost flexibility in utilizing infrastructure.

Powerful Management Tools
: Managing MRG Grid is simplified by leveraging the Red Hat Enterprise MRG unified, browser-based management console. The Red Hat Enterprise MRG integrated management tools enable administrators to manage, configure, provision, deploy, and monitor their grid deployments using the same tools they use for MRG Messaging and MRG Realtime.

Features.

MRG Grid provides a broad set of features across both High Throughput Computing and High Performance Computing, including:

Virtualization
: Allows for submission of a virtual machine (VM) as a user job, supporting migration of the VM

Dedicated and Undedicated Node Management (Cycle-Stealing)
: Allows for dedicated resources (clusters) to be augmented with otherwise undedicated (desktops) using flexible policies

Multiple Standards-Based APIs
: Web Service interface provides job submission and management functionality; CLI provides

a highly scriptable, with consistent output, interface to all functionality

Security
: Authentication using multiple mechanisms

  Privacy provided by network encryption

  Integrity of network traffic

  Authorization through flexible configuration policies

Federated Grids/Clusters
: A mechanism known as flocking allows independent pools to use each others' resources, controllable by customizable policies

Management Tools
: Powerful browser-based management tools for managing daemons and machines, security, compute jobs, scalability settings, priorities, and more. Also provides sophisticated monitoring capabilities.

Workflow Management
: The ability to specify job dependencies, via DAGMan, allows for construction and execution of complex workflows

  The ability to schedule data placement, via Stork, assists in creation of workflows that intelligently handling data

Accounting
: User and group resource utilization is tracked and accessible to adminstrators

ClassAds
: A flexible language for policy and meta-data description

Policies
: Flexible, customizable policies specified by jobs and resources via ClassAds

High Availability
: The Negotiator and Collector, via HAD, and the Schedd, via Schedd Fail-over, can have their state replicated to allow for graceful fail-over upon service disruption

Disk Space Management
: Through a multi-protocol storage management system, called NeST, the ability to manage (allocate, free, reserve, etc) disk space is exposed to a user's jobs

Database Support
: All data about jobs and resources can be stored in a database via Quill

Compute On-Demand (COD)
: The ability for a node or set of nodes to be claimed by a user in such a way that others may use the claimed nodes until the user needs them

Dynamic Pool Creation
: Through a technology known as Glide-ins, nodes can be dynamically added to a pool to service user jobs

Priority Based Scheduling
: Priority scheduling is performed at the granularity of a user

: Fair-share scheduling can be performed on groups of users

: Priority management is controllable by adminstrators

Account Remapping
: Allows for execution across administrative domains

: Enhance security by using a restructed pool of users to run jobs on execute machines

Privilege Separation
: Only a single, specialized, audited component requires root/administrator permissions on execute nodes

Parallel Universe
: Provides an extensible framework for running parallel (including MPI) jobs

: Co-allocation of compute nodes is done automatically

: Framework implementation for MPICH1, MPICH2, and LAM provided

Java Universe
: Explicit support of jobs written in Java

Time Scheduling for Job Execution (Cron)
: Allows a job or multiple jobs to be started at specific times, with customizable policy for failures such as missed deadlines

Backfill
: Allows otherwise unused nodes to run jobs provided by BOINC

File Staging
: Support for automatic file staging, e.g. job input, and online file io (i.e. file streaming from submit to execute nodes) via Chirp and remote syscalls, in the absense of a shared filesystem

Master-Worker (MW)
: A C++ framework allowing a single master process to allocate and manage multiple worker processes, which process data based on master specified policies

Condor-C
: Allows for jobs in one queue to be moved to another queue

Hawkeye
: Allows for automated monitoring of one or more pools

More Information

Reporting Bugs.

Follow these instructions to enter a bug report:

1. You will need a *Bugzilla* [https://bugzilla.redhat.com/index.cgi] account. You can create one at *Create Bugzilla Account* [https://bugzilla.redhat.com/createaccount.cgi].
2. Once you have a Bugzilla account, log in and click on *Enter A New Bug Report* [https://bugzilla.redhat.com/enter_bug.cgi].
3. You will need to identify the product (Red Hat Enterprise MRG), the version (beta), and whether the bug occurs in the software (component=grid) or in the documentation (component=Grid_Deployment_Guide).

Further Reading.

- Red Hat Enterprise MRG and MRG Grid Product Information
  - *http://www.redhat.com/mrg*
- Condor Manual
  - *Condor Manual* [http://www.cs.wisc.edu/condor/manual/]

Appendix A. Revision History

Revision History

Revision 1.1	4 February, 2008	Lana Brindley

Updated Installation Instructions

Revision 1.0	23 November 2007	Lana Brindley

Initial Draft