RED HAT ENTERPRISE MRG: MESSAGING, REALTIME, AND GRID

BRYAN CHE

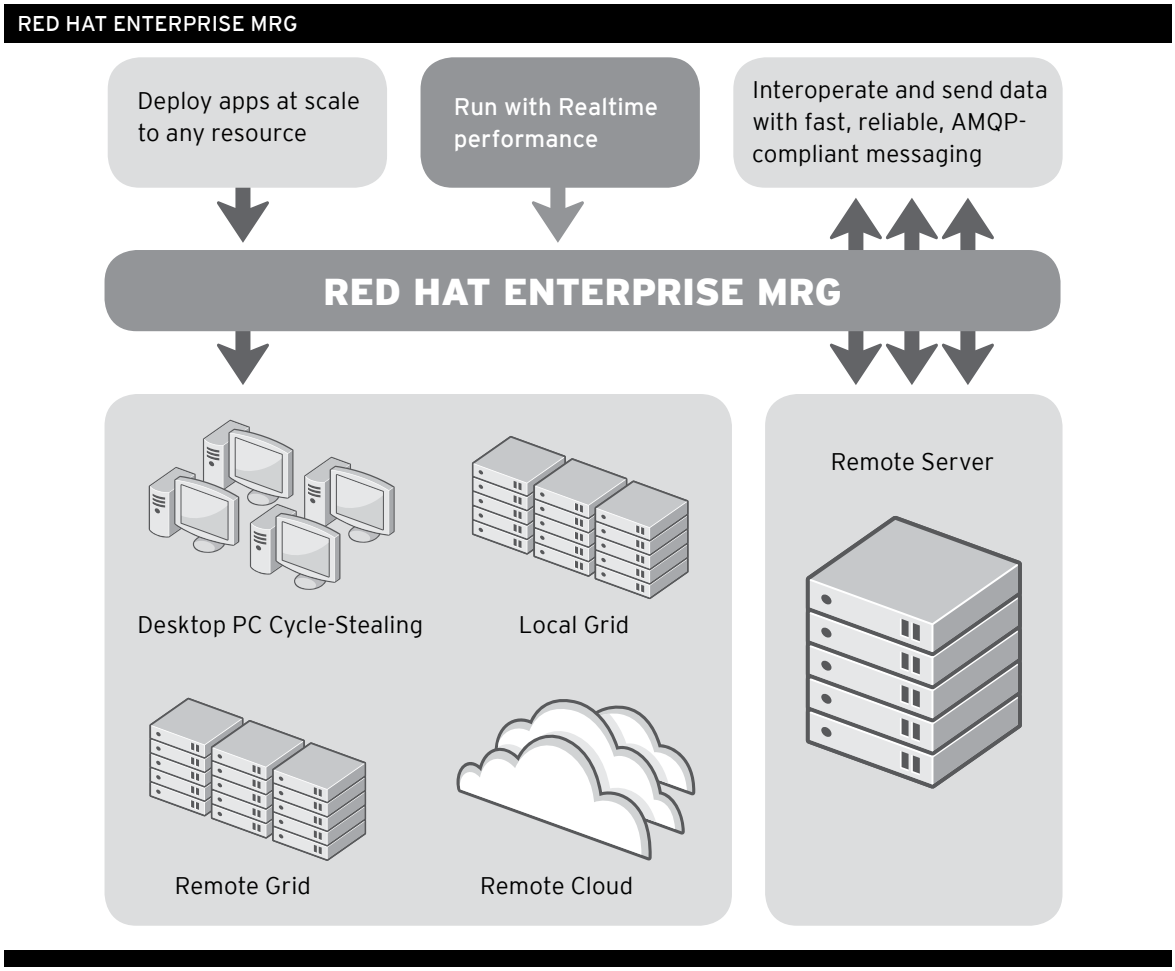
TABLE OF CONTENTS

HIGH PERFORMANCE DISTRIBUTED COMPUTING

Red Hat Enterprise MRG is a new platform that enables enterprises to increase dramatically the value of their Information Technology (IT) infrastructure by reducing complexity and decreasing costs while increasing performance, reliability, interoperability, and flexibility. By integrating messaging, realtime, and grid technologies, Red Hat Enterprise MRG provides enterprises with a revolutionary foundation for high performance distributed computing for everything from SOA to virtualization to cloud computing to bare-metal mission-critical applications.

With Red Hat Enterprise MRG, enterprises can:

- Fully utilize all computing resources available, from dedicated grids in local data centers, to physically remote servers in other data centers, to desktop workstations via cycle stealing, to dynamically provisioned grids in the cloud
- Schedule any workload, from sub-second to long-running, from single-threaded to massively parallel, from Windows jobs to Linux jobs, from virtualized environments to native environments, across all computing resources from desktops to the cloud in a highly efficient manner



- Distribute data via MRG's AMQP-compliant Messaging bus with performance up to two orders of magnitude better than other solutions and deterministic low latency
- Provide deterministic performance—especially for messaging and networking workloads in distributed environments—and meet stringent SLA's via MRG's Realtime kernel
- Submit, manage, and monitor various distributed workloads with a single, unified interface
- Simplify their software architecture and deployment stacks by leveraging one integrated platform rather than deploying numerous, specialized, incompatible point products to cover the spectrum of distributed computing

These capabilities offer tremendous value at the cores of what enterprises do. For example, with MRG:

- Investment banks can gain competitive advantage by building a more deterministic and lower latency reliable trading platform and also leveraging resources ranging from dedicated grids to dynamically-provisioned cloud-based resources for running Monte Carlo analysis of market conditions
- Governments and the defense industry can create more reliable simulations or control systems and process and disseminate critical information faster
- Telecommunication companies can provide increased reliability, performance, and interoperability for their networks and infrastructure
- Animation studios can render movies more efficiently and complete them sooner with improved workflows
- Oil and gas companies can save precious drilling time by performing seismic analysis with increased speed and efficiency

and so on. In short, Red Hat Enterprise MRG can provide any enterprise with distributed data or computational resources with competitive advantage, greatly simplified complexity, and the flexibility and performance to address peak computational demands by leveraging and integrating the three fundamental technologies of messaging, realtime, and grid.

RED HAT ENTERPRISE MRG COMPONENTS

Red Hat Enterprise MRG integrates three technologies: messaging, realtime, and grid. Even though the three technologies provide unique value in combination, they also offer significant individual advantages in their respective domains.

MRG MESSAGING

Messaging is the backbone of enterprise and high-performance computing, SOA deployments, and platform services. However, messaging offerings to date have offered restricted capabilities, forcing users to choose between high-speed messaging systems for deployments like market-data applications, reliable messaging systems for scenarios where guaranteed delivery is crucial, and other solutions for messaging of large, multi-gigabyte payloads. Furthermore, these different messaging systems are fundamentally incompatible with each other—meaning that enterprises that have broad messaging requirements have often had to purchase multiple, incompatible messaging products and deploy them in silos—increasing complexity and cost.

Red Hat Enterprise MRG Messaging brings tremendous value to enterprise messaging by providing the scalability, reliability, and performance to handle all messaging use cases. Red Hat Enterprise MRG provides messaging that is up to 100-fold faster than other solutions—with full reliability and durability—and support for multi-gigabyte messages. Furthermore, Red Hat Enterprise MRG implements Advanced Message Queuing Protocol (AMQP), the industry's first open messaging standard, for unprecedented interoperability that is cross-language, cross-platform, multi-vendor, spans hardware and software, and extends down to the wire level. Red Hat is a founding member of the AMQP working group, which is developing the AMQP standard.

MRG Messaging uses Linux-specific optimizations to achieve optimal performance on Red Hat Enterprise Linux and MRG Realtime but can also be deployed on non-Linux platforms like Windows and Solaris without the full performance and quality of service benefits that Red Hat Enterprise Linux provides. MRG Messaging supports most major development languages.

MRG Messaging includes the following features:

- **High Performance Messaging Journal** – MRG Messaging can achieve sustained durable messaging throughput rates of 500,000 messages per second per LUN on typical hardware, a rate that is up to 100-fold better than previously possible. MRG Messaging achieves these rates through the inclusion of a new AIO journal specifically optimized for persistent messaging storage on Red Hat Enterprise Linux. This AIO journal can achieve maximum LUN speeds and fiber channel write speeds. Red Hat has performed extensive work in its journal to optimize the cache on disk arrays and also to write in rings per queue, which makes it possible to map IO over different LUNs. This allows scale up in terms of IO rates and also dramatically minimizes seeks while managing separate read and write contexts. Together, these optimizations provide dramatically better performance while consuming less CPU. MRG Messaging can deliver durable messaging at rates that are otherwise found only from transient messaging.
- **Native OFED Infiniband Support** – MRG Messaging offers native OFED Infiniband support. This will enable MRG Messaging to achieve unique throughput/latency ratios not currently found in messaging systems. Red Hat is applying the same types of work it did for achieving maximum theoretical messaging rates for storage to the network.
- **MRG Realtime Optimizations** – MRG Messaging is optimized to run on MRG's Realtime kernel to provide optimal, deterministic latencies.
- **AMQP Support** – MRG Messaging implements AMQP, an open and interoperable messaging standard developed by the AMQP Working Group. Red Hat is one of the founding members of the AMQP Working Group. Red Hat Enterprise MRG 1.0 is compliant with AMQP 0-10, the most current version of AMQP.
- **Flexible Messaging Paradigms** – MRG Messaging includes comprehensive support for various messaging paradigms, including store-and-forward, transaction distribution, publish-subscribe, content-based routing, queued file transfer, point-to-point connections among peers, and market data distribution.
- **Multi-language Client Support** – MRG Messaging offers clients in a variety of languages across several platforms, including Java (JMS), C++, Python, .NET, and more. MRG Messaging's <100k scripting clients enable rapid application development with a light-weight tool, while still offering all the performance and transactional power of AMQP.
- **Transient and Durable Messaging** – Durable messaging ensures that messages and queues are restored in the unlikely event of a broker crash or an operating system crash; transient messaging

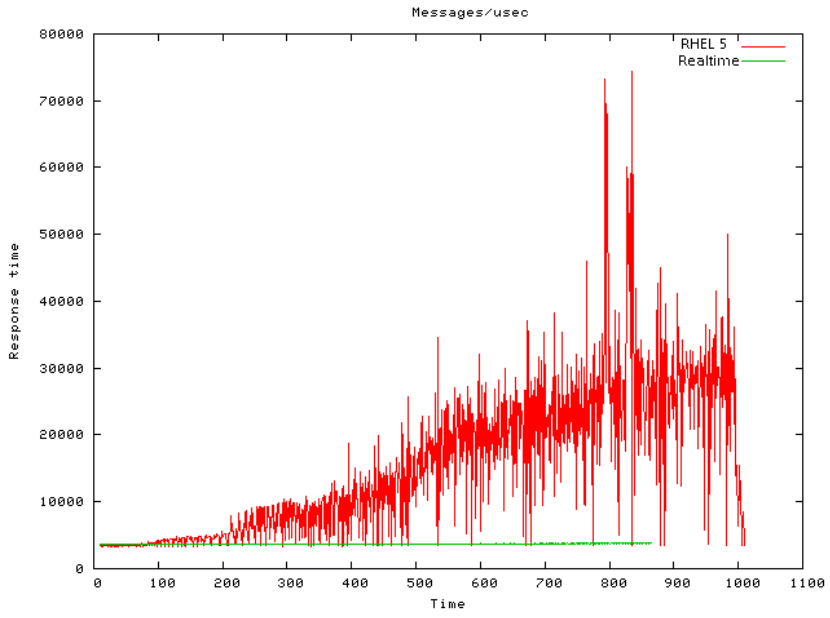
reduces the processing needed for messages and guarantees delivery as long as the broker continues to run.

- **Large Message Support** – MRG Messaging supports multi-gigabyte messages
- **Clustering and Failover** – MRG Messaging provides clustering and failover to ensure that your applications continue to function if a broker or operating system should crash.
- **Transactions** – MRG Messaging provides support for messaging transactions, including distributed transaction (XA) support. With transactions, you can ensure that all messages in a group are delivered as a whole—if delivery of one message fails, delivery of all messages in the group fails.
- **Distributed Management console** – MRG Messaging contains a console for instrumentation (including historical), configuration and the ability to perform management operations through a web interface to a network of deployed machines.

MRG REALTIME

For the most time-critical workloads, "close enough" isn't good enough. For instance, in the highly competitive financial services marketplace, milliseconds make the difference between a trade and a lost customer. In these environments, consistently fast response times are a huge competitive advantage. MRG Realtime provides the highest levels of predictability for consistent low-latency response times to meet the needs of time-sensitive workloads. For most customers, a properly tuned standard Red Hat Enterprise Linux system provides acceptable levels of determinism. For applications with extremely demanding latency requirements, MRG Realtime provides new levels of determinism by optimizing lengthy kernel codepaths to ensure that they do not become bottlenecks and allowing better prioritization of applications. This results in consistent, predictable response times for high priority applications.

RESPONSE TIMINGS COMPARISON



MRG Realtime is a set of RPM packages which are add-ons or replacements to standard Red Hat Enterprise Linux 5 components. To install MRG Realtime, you must have already installed Red Hat Enterprise Linux 5.1 or later. MRG Realtime consists of:

- A realtime kernel, which replaces the standard Red Hat Enterprise Linux 5 kernel.
- A set of configuration utilities which facilitate realtime tuning.
- A set of performance monitoring tools to allow realtime performance to be monitored and provide information to support realtime tuning.
- Documentation, including information on how to tune a system for optimal latency.

MRG's realtime capabilities are contained entirely in its realtime kernel, which is currently based on the Linux 2.6.24 kernel. This means that MRG Realtime systems' runtime environments, system utilities, and glibc runtime libraries are completely unchanged, and all their applications will continue to run. There is no need to recompile applications to run them with MRG Realtime; applications which work on Red Hat Enterprise Linux will continue to work unchanged on Red Hat Enterprise Linux with MRG Realtime.

Red Hat is leading the development of MRG Realtime's capabilities in the mainstream open source kernel community; to date, Red Hat engineers have written about 90% of Linux's realtime kernel code. Furthermore, by developing these realtime capabilities as part of the broader Linux community, Red Hat ensures that its MRG Realtime offering continues to benefit from the latest Linux kernel features like support for new hardware or bug fixes. Other solutions which are not accepted by the mainstream Linux kernel community will eventually become dead-end forks of Linux.

Red Hat Enterprise MRG Realtime includes a plethora of features developed over many years which enhance determinism and performance. Some of the key enhancements in MRG Realtime include:

- **Low latency interrupt handling** – The lengthy, and non-interruptable, codepaths inherent in device interrupt handlers are a primary source of system non-determinism. MRG Realtime addresses this by breaking up these long-running kernel codepaths into separately schedulable portions. This ensures that lower priority event handling does not block priority realtime processing.
- **Enhanced priority handling** – MRG Realtime includes more fine-grained tuning control. For example it is possible to schedule network processing at higher priority than disk and storage processing. Additionally, there are scheduler enhancements to further guarantee that higher priority processes get uninterrupted service.
- **Priority inversion avoidance** – MRG Realtime, when used with the standard Red Hat Enterprise Linux glibc runtime library, provides priority inheritance. This is a mechanism to ensure that lower priority processes do not block higher priority processes which are contending for shared resources. A scalable priority inheritance algorithm is a requirement for realtime java.
- **Timer precision** – All realtime applications inherently require highly accurate timing events. This makes it possible to have accurately timestamped messages, and event timeouts that occur precisely as scheduled. Conventional Linux has had its timing accuracy governed by a periodic system interrupt, which results in imprecise timing. In contrast, MRG Realtime uses hardware based event timers, resulting in substantially more accurate and predictable timing.
- **Latency tracer** – The latency tracer is a peak detector. It identifies the longest running, non-preemptable kernel codepaths, which is invaluable when differentiating whether non-deterministic response times are occuring in kernel vs application space.

MRG GRID

Red Hat Enterprise MRG Grid enables enterprises to scale their IT infrastructure to meet business demands and also decrease costs by fully utilizing all computing resources for all workloads in a dynamic and flexible manner. MRG Grid can dynamically and rapidly run jobs across computing resources from dedicated grids in local data centers, to physically remote servers in other data centers, to desktop workstations via cycle stealing, to platforms from Windows to Linux, to virtualized containers or bare metal, or to dynamically provisioned grids in the cloud. Furthermore, MRG Grid can schedule workloads ranging from sub-second executions to long-running, massively parallel jobs. For everything from adding on-demand capacity for Christmas-rush-like scenarios to simplifying capital infrastructure to adding additional computing power for the most demanding workloads, Red Hat Enterprise MRG Grid offers tremendous value and revolutionary flexibility.

MRG Grid is API-less from a workload standpoint. This means that users do not need to code applications and jobs to a specific API in order to take advantage of MRG Grid's scheduling—simplifying development and preventing lock-in. MRG Grid does, however, include a full Web-Services based API for the management, monitoring, administration of jobs so that enterprises can include MRG Grid as part of their broader workflow. Additionally, MRG Grid provides powerful command-line tools, a Web-based management console, and leverages MRG Messaging for an AMQP-compliant messaging interface for job submission.

MRG Grid is based on the Condor Project started and hosted by the University of Wisconsin-Madison. Red Hat and the University of Wisconsin have signed a strategic partnership to release Condor under an OSI-approved open source license (making it possible to be included in open source distributions), and co-develop together to bring innovation from the research community to the enterprise. Condor has been in active development and used by a wide community since 1988. Now, based on their joint vision of advancing open source software, Red Hat and the University of Wisconsin are collaborating to add enhanced enterprise stability and functionality to Condor, add high throughput computing capabilities to Linux, and ultimately advance and strengthen the Condor project and community.

MRG Grid includes the following features:

- **Management Tools** – MRG Grid includes powerful browser-based management tools for managing daemons and machines, security, compute jobs, scalability settings, priorities, and more. MRG also provides sophisticated monitoring capabilities.
- **Desktop Cycle-Stealing** – Desktop Cycle-Stealing allows you to leverage the unused capacity of desktops to add processing power to your grid.
- **Cloud scheduling** – MRG Grid enables leveraging computing resources at cloud-based environments like Amazon EC2. MRG Grid will automatically and dynamically:
  - Provision a grid at cloud environments like Amazon EC2
  - Execute jobs in those cloud environments
  - Save and return your results
  - Tear down your cloud infrastructure when your jobs are complete so that you only pay for what you use
- **ClassAds** – ClassAds provides a flexible language for policy and meta-data description.

- **Policies** – MRG Grid enables flexible, customizable policies specified by jobs and resources via ClassAds.
- **Virtualization** – MRG Grid allows submitting of a virtual machine (VM) as a user job and supports migration of the VM.
- **Federated Grids/Clusters** – A mechanism known as flocking allows independent pools to use each others' resources, controllable by customizable policies.
- **Multiple Standards-Based APIs** – A Web Services interface provides job submission and management functionality; a CLI provides a highly scriptable interface to all functionality, with consistent output.
- **Workflow Management** – MRG Grid provides sophisticated workflow managemeint capabilities, including:
  - The ability to specify job dependencies via DAGMan for construction and execution of complex workflows.
  - The ability to schedule data placement via Stork for the creation of workflows that intelligently handle data.
- **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)** – MRG Grid provides 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** – MRG Grid can schedule based on priority so that:
  - 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 administrators.
- **Accounting** – User and group resource utilization is tracked and accessible to administrators.
- **Security** – MRG Grid includes comprehensive security and support for:
  - Authentication, multiple mechanisms (kerberos, ssl, shared secret, claimtobe, filesystem, remote-filesystem)
  - Privacy, network encryption (blowfish, 3des)
  - Integrity, of network traffic (md5)
  - Authorization, through flexible configuration policies

- **Account Remapping** – MRG Grid allows for execution across administrative domains. It also Enhances security by using a restricted 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** – MRG Grid provides an extensible framework for running parallel (including MPI) jobs. In this environment, MRG Grid automatically co-allocates compute nodes. Furthermore, MRG Grid provides framework implementation for MPICH1, MPICH2, and LAM.
- **Java Universe** – MRG Grid provides explicit support of jobs written in Java.
- **Time Scheduling for Job Execution (Cron)** – MRG Grid allows a job or multiple jobs to be started at specific times, with customizable policy for failures such as missed deadlines.
- **Backfill** – MRG Grid allows otherwise unused nodes to run jobs provided by BOINC.
- **File Staging** – In the absence of a shared filesystem, MRG Grid supports automatic file staging (e.g. job input), and online file IO (e.g. file streaming from submit to execute nodes) via Chirp and remote syscalls
- **Dedicated and Undedicated Node Management** – MRG Grid enables dedicated resources (clusters) to be augmented with otherwise undedicated (desktops) using flexible policies.
- **Master-Worker (MW)** – MRG Grid provides A C++ framework that enables a single master process to allocate and manage multiple worker processes, which process data based on master specified policies.
- **Condor-C** – Condor-C enablesjobs in one queue to be moved to another queue.
- **Hawkeye** – Hawkeye enables automated monitoring of one or more pools.

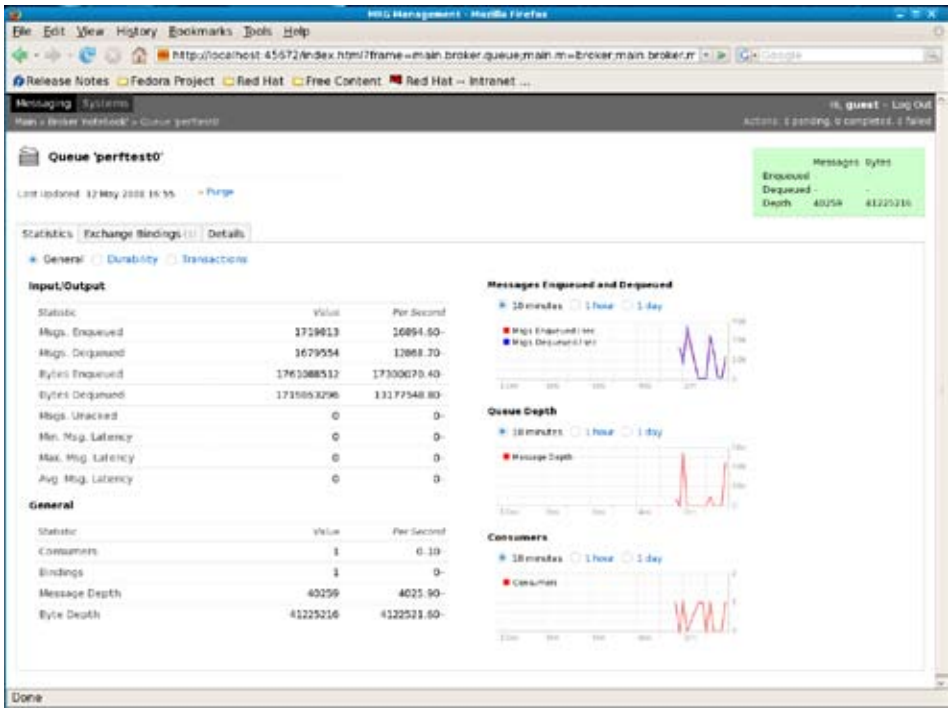
MRG Grid will leverage the other parts of Red Hat Enterprise MRG, MRG Messaging and MRG Realtime, to provide new capabilities like:

- **Optimized realtime scheduling of sub-second jobs with qualities of service (QoS)** – MRG Grid can reliably schedule jobs in less than a second to make practical running sub-second executions on a grid
- **Message-based job submission** – MRG Grid leverages MRG Messaging to provide an AMQP-compliant messaging interface for submitting jobs to MRG's Condor scheduler from a variety of platforms and languages

MRG MANAGEMENT

Red Hat Enterprise MRG includes a variety of powerful tools for management and monitoring. Foremost amongst these is a Web-based management console that provides a unified management interface across all the components of MRG: Messaging, Realtime, and Grid. With this fully AJAX-enabled management console, enterprises can do everything from manage their messaging queue rates to monitor the utilization of their grids (and set up rules to grow their grids dynamically when they reach capacity) to tune and administer the underlying operating systems in their deployment environments. Red Hat Enterprise MRG's management capabilities are built on MRG's messaging capabilities so that MRG harnesses AMQP-compliant messaging for its management and monitoring events.

MRG MANAGEMENT



In addition to its web-based console, Red Hat Enterprise MRG also includes scriptable command-line utilities for administration. Furthermore, because MRG leverages AMQP for its management, any messaging client across a variety of languages and platforms—including scripting languages like Python—can fully participate in the management of a Red Hat Enterprise MRG deployment.

Finally, Red Hat Enterprise MRG includes a Web Services API for the full administration of MRG Grid jobs.

RED HAT ENTERPRISE MRG'S INTEGRATED VALUE

Each of Red Hat Enterprise MRG's components—messaging, realtime, and grid—provide distinctive value in and of themselves. However, the integration of these technologies into a single platform provides tremendous value and benefits, including:

- **Deterministic low-latency messaging** – The combination of realtime and messaging enables messaging with highly deterministic response times and reliably low latency.
- **Messaging-based reliable, low-latency job scheduling** – By leveraging MRG's messaging and realtime components, MRG Grid can reliably schedule and execute jobs in less than a second. This vastly increases the range of jobs that MRG can meaningfully handle—if a job scheduler takes several seconds or even minutes to schedule a job, then there is no value in having that job scheduler run

rapid jobs. By providing a message-based interface for sub-second job scheduling, MRG Grid can handle all workloads, from sub-second executions to long-running, massively parallel jobs. Furthermore, by leveraging AMQP, Red Hat Enterprise MRG provides a cross-platform and cross-language protocol for submitting jobs from a variety of environments.

- **Unified management** – Red Hat Enterprise MRG builds its management capabilities on top of its AMQP-compliant messaging system. This provides the ability to manage MRG via a messaging interface across a variety of languages and platforms. It also enables Red Hat Enterprise MRG's powerful Web-based management console.
- **Reduced complexity and breakthrough value** – Red Hat Enterprise MRG provides an integrated platform for high performance distributed computing. Without MRG, enterprises would typically need to build a complicated architecture spanning a number of different products for capabilities like fast messaging, reliable messaging, large-file messaging, deterministic latency, workload scheduling, and scalable virtulization. Furthermore, enterprises have to tie these products together to meet their use cases—usually through custom development. Compounding this difficulty is that many of these products are fundamentally incompatible—requiring deployment in various silos for specialized functionality. Finally, each different product brings with it its own management tools and requirements. As an integrated platform, Red Hat Enterprise MRG vastly simplifies the deployment, management, and architecture of enterprise IT architecture while significantly increasing its functionality, performance, and value.

LINUX AUTOMATION

Linux Automation is Red Hat's architecture to create more flexible and efficient IT infrastructure, reduce IT operating and capital costs, and improve IT service levels by enabling any application certified for Red Hat Enterprise Linux to run optimally anywhere from bare-metal infrastructure to virtualized environments, to cloud infrastructure.

Linux Automation's significant benefits at the application scale compound greatly when deployed at the grid scale with Red Hat Enterprise MRG. Through MRG, you can bring the benefits of Linux Automation to all your workloads across all your computing resources.

SUMMARY

Red Hat Enterprise MRG is a revolutionary platform for high performance distributed computing. By integrating messaging, realtime, and grid technologies in MRG, Red Hat provides its customers tremendous value by greatly reducing the costs and complexity in IT infrastructure while simultaneously increasing their performance, utilization, reliability, flexibility, and interoperability. For additional information on Red Hat Enterprise MRG, visit **http://redhat.com/mrg**.

www.redhat.com
#634546_0508