WHITE PAPER

IDEAS INTERNATIONAL

custom consulting services

RED HAT ENTERPRISE LINUX PROGRESSES TO NEXT LEVEL OF RAS

November 2010

PREPARED FOR

Red Hat

TABLE OF CONTENTS

Executive Summary

The arrival of Intel's Nehalem-EX processor – its first to contain eight computing cores – will enable x86 systems to meet the performance requirements for more of the enterprise server market than ever before. As the scalability limitations of x86 systems diminish, the focus of enterprise users is shifting to the reliability, availability, and serviceability (RAS) that can be achieved on industry-standard platforms. The growing performance of x86 servers is attracting users who are interested in rehosting workloads from their aging high-end servers in order to benefit from the volume economics of x86 systems. Until now, though, efforts to consolidate enterprise workloads on Linux have been limited by perceptions that Linux distributions could not match the environment users were accustomed to on their existing high-end systems.

With a central role in developing the Linux kernel itself, Red Hat is helping to prepare Linux for the most critical workloads in enterprise and datacenter environments. Representing a key complement to the scalability leap delivered by Intel Xeon hardware, Red Hat® Enterprise Linux® 6 (RHEL 6) introduces a variety of new operating system functions targeting the RAS requirements of enterprise workloads. The Nehalem-EX processor introduced many new features designed to respond to faults that occur in low-level hardware, and RHEL 6 includes the necessary operating system functionality to support most of these features. When deployed on Xeon 7500/6500 systems, RHEL 6 can improve uptime by exploiting new hardware-based resiliency functions in the processor; delivering new KVM-based virtualization functions that can be used to maintain service levels; and providing higher-level protection against workload failures through clustering with Red Hat's HA Add-on. These capabilities will allow high-end users to put many of their familiar RAS disciplines into practice on a broad choice of industry-standard servers.

Introduction: Red Hat Takes on the Datacenter

Linux broke into the mainstream in the mid-1990s as an optimal environment for hosting edge-of-network workloads such as web, file, and print services. More recently, Linux has become widely adopted as a general-purpose operating system for a broad range of departmental and workgroup applications. It has also often been deployed in appliance servers, and in clustered environments for HPTC and certain types of database applications. Many have even predicted that Linux will ultimately become the dominant operating system for high-end servers in datacenter environments, though the exact timeframe for when this might occur has remained vague. Now, that timeframe is becoming much clearer.

With the convergence of several major technology introductions, the ability to host critical workloads on Red Hat Enterprise Linux and industry-standard x86 servers will significantly increase. This year's release of the Intel Xeon 6500 and 7500 processors (code-named "Nehalem-EX") dramatically increases the range of performance that can be achieved on industry-standard systems. Since 2006, Intel has sharply accelerated its performance improvements, largely due to its adoption of multicore techniques. In addition to the normal

IDEAS RECOMMENDATIONS FOR USERS

Ideas International (IDEAS) offers the following recommendations for users who are considering the deployment of Red Hat Enterprise Linux on Intel Xeon 7500 systems:

- Understand the performance characteristics of the application workloads being considered for deployment on a scalable Linux system. Evaluate the tradeoffs in terms of operational and acquisition costs between scaling out on clusters of multiple smaller servers versus. scaling up on larger SMP systems.
- Develop processes for logging detailed runtime data and error conditions on critical servers, so that administrators can perform predictive analysis using their preferred system management software or self-developed tools.
- Understand the potential causes of downtime, and study how single-system versus multisystem approaches for availability can help to overcome these conditions.
- Develop a strategy for maintaining availability when deploying virtualization; determine the need to maintain uptime on virtual machine hosts, and evaluate tradeoffs with multisystem approaches for protecting virtual machines (i.e., live migration and clustering).

performance improvements gained by Moore's law (which states that processors double their performance roughly every 18 months), Xeon 7500's design for "glueless" 8-socket servers gives another boost. With a powerful chip that can easily be deployed in 2-, 4-, and 8-socket configurations, servers based on the Intel 7500 can support up to 128 cores and 144 I/O slots in a single Symmetric Multiprocessing (SMP) form factor. As a result, these systems will now be able to meet the performance requirements for more of the enterprise server market than ever before.

Representing a key complement to the scalability leap delivered by Intel Xeon hardware is the introduction of Red Hat Enterprise Linux 6 (RHEL 6), which features a variety of new software-based functions targeting the requirements of enterprise workloads. The SMP scalability capabilities of Linux started to improve dramatically in late 2003, with the release of version 2.6 of the Linux kernel. The 2.6 kernel introduced fundamental improvements that specifically targeted the ability to run on SMP servers with many cores and very large amounts of memory. It also introduced fine-grained locking for kernel resources and improved many of the algorithms used to enable simultaneous access to resources by multiple processes. The operating system scheduler in the 2.6 kernel was also replaced with the so-called "O(1)" algorithm, which can schedule processes in a fixed amount of time, regardless of the number of processes involved.

Since then, developers from the Linux kernel community, distribution suppliers, and systems vendors have continued to refine the scalability of Linux operating systems on large SMP servers. RHEL 6 supports up to 4,096 cores at the kernel level, and its current tested limit is 128 cores, which is the range of most enterprise x86 servers at present (the kernel is enabled for greater SMP scalability in order to be "future proof" for larger x86 servers as they emerge in the market). With the growing maturity of the core RHEL technology (the Linux 2.6 kernel has been shipping for nearly seven years), users are now more comfortable than ever with deploying critical workloads on Linux platforms such as RHEL.

With the scalability limitations of x86 systems nearly eliminated, the focus of enterprise users is shifting to the reliability, availability, and serviceability (RAS) that can be achieved on industry-standard platforms. The growing performance of x86 servers is attracting users who are interested in rehosting workloads from their aging high-end servers in order to benefit from the volume economics of x86 systems. These users are concerned with matching the reliability attributes of their existing servers, which have been proven to host the most critical workloads. In other cases, users of existing x86 systems may be interested in consolidating multiple workloads on larger servers in order to manage them with better economies of scale. However, these users may be concerned about increasing the risk to uptime by putting "too many eggs in one basket." Industry developments such as cloud computing will also result in significant growth of server workloads, as an increasing range of data and computing activities are accessed remotely over enterprise and global networks.

How does the combination of RHEL 6 and Xeon 7500/6500 hardware help to address concerns about reliability? The ability for RHEL 6 to maintain uptime when deployed on new Xeon 7500/6500 systems spans three broad areas:

- Exploiting Xeon Hardware Functions – The Xeon 7500/6500 processors contain many new features implemented in silicon to improve RAS capabilities at the level of memory and I/O interconnects, and RHEL 6 includes much of the basic operating system functionality required to exploit these features. When combined with the Xeon 7500/6500 processors, RHEL 6 will significantly improve the uptime of single systems.
- Leveraging Virtualization – Virtualization can be used to fundamentally improve the overall reliability of a computing infrastructure. Virtualization enables fewer physical servers to be deployed after consolidation, which reduces the footprint for potential hardware failures that result in unplanned downtime. Moreover, the servers that are deployed can be configured with high availability (HA) features such as redundancy and hot-plug components to reduce

With the convergence of several major technology introductions, the ability to host critical workloads on Red Hat Enterprise Linux and industry-standard x86 servers will significantly increase.

downtime. The ability to migrate virtual machines from one host to another with little or no interruption to processing provides yet another means to reduce planned downtime. Such migration allows workloads to be temporarily moved so that hardware maintenance can be performed on the hosts with minimal disruption. RHEL 6 has significantly improved support for virtualization with new Kernel-based Virtual Machines (KVM) capabilities.

» Implementing HA Clusters – Red Hat's HA cluster solution, called the HA Add-On, can maintain uptime more efficiently when deployed on RHEL 6. Red Hat HA Add-On is one of the more mature HA clustering solutions for Linux. Further, HA clustering can be made much simpler and more practical when combined with virtualization. HA implementations normally require applications and their dependencies to be adapted so that they can be restarted on backup systems – which is a notoriously complex and error-prone process. With virtualization, an entire workload can easily be relaunched simply by restarting the virtual machine on which it is hosted.

On top of these technical capabilities, Red Hat has also developed all of the other competencies necessary for credibility in datacenter and enterprise environments, including support capabilities, certifications, partnerships, and reputation. As a leading operating environment for industry-standard hardware, Red Hat has to support a vast array of diverse server platforms from many vendors, which makes consistently achieving production-grade availability a notoriously challenging endeavor. In order to understand how Red Hat meets this challenge, it is worth examining the development process underlying new releases such as RHEL 6.

The RHEL Development Process: Attacking RAS Upstream

Red Hat is an integral member of the open source software development community. The company has steadily maintained its commitment to open source ideals since it was founded in 1995, and it has contributed to or founded a wide variety of community-based software projects. Red Hat is deeply involved in working on the Linux kernel itself. For example, according to the most recent development statistics for the latest 2.6.36 kernel, Red Hat is unsurpassed by any other company in terms of contributing "changesets" (i.e., specific functional enhancements).[1] Moreover, Red Hat's strong relationships with other parties involved with developing Linux are essential for driving its efforts to develop a reliable platform. The leading hardware vendors have a particularly strong interest in the evolution of Linux, and in fact, much of the Linux kernel development today is done by engineers who are employed by server manufacturers.

Red Hat's kernel development team works closely with all of the major hardware vendors to plan, develop, and test distributions of RHEL. Red Hat's engagement begins with x86 processor vendors such as Intel and AMD, as the majority of Red Hat's revenue – and system vendors' growth – is derived from x86-based servers. The deep development relationship between Red Hat and Intel has progressed for at least the last 10 years. Red Hat generally knows 18 to 24 months in advance of Intel shipping a new processor what features will be included in the silicon, and then makes an effort to deliver the necessary software in time for production release of the processor. For example, while developing RHEL 6 in preparation for the release of Xeon 7500/6500, Red Hat held weekly meetings with Intel to discuss the various new system features that would be enabled by the new processor. During these discussions, Red Hat and Intel negotiated which parts of the code targeting the new RAS features would be developed by their respective Linux development teams.

As part of the development process, Red Hat receives some of the first pre-ship units of Intel's new processor early on, so that it can start its development using reference boards in "white box" servers. Red Hat then holds frequent meetings with the main server vendors that it works with, in which the various server product roadmaps are reviewed under NDA. In these

[1] See "Statistics for the 2.6.36 development cycle," *LWN.net*, October 13, 2010.

Representing a key complement to the scalability leap delivered by Intel Xeon hardware is the introduction of Red Hat Enterprise Linux 6 (RHEL 6), which features a variety of new software-based functions targeting the requirements of enterprise workloads.

meetings, Red Hat determines what it needs to do in order to support specific server products in RHEL. Since these requirements may vary from vendor to vendor, Red Hat developers identify a definitive subset of functionality with the most relevance across the greatest number of systems. In some cases, Red Hat may take on special development efforts in order to support a unique component, such as particular storage controllers or network controller, and this work may be done in conjunction with individual vendors. At the time of release, RHEL should then be supported on systems from all vendors, at least at the level of their motherboards.

Vendors will sometimes ask for specific changes to the kernel to meet the needs of certain customers. For example, in certain geographies such as Japan, system vendors are the most eager to demonstrate RAS capabilities, and they are the most eager for hooks into the kernel to trigger error events so they can be tested. These vendors are also asking for user interface controls that can be used to set thresholds to force certain RAS features to kick in. Red Hat assigns Partner Managers and Technical Account Managers (TAMs) to each of the major vendors to manage requests for features in RHEL, and the Red Hat development team evaluates these requests to determine which to accept.

The new RAS capabilities in Xeon 7500/6500 raised the bar somewhat for executing this process with RHEL 6. For this release, Red Hat allocated far more development resources to RAS than in previous releases. It also worked more closely with Intel than ever before to make sure that the necessary code development was getting done upstream (i.e., in the Linux kernel development community), and that this code was properly integrated into RHEL. Red Hat also invested in the development of testing tools to validate the RAS features, such as programs to inject errors for verifying the Machine Check Architecture (MCA) function (see below). Developers also saved memory modules that had previously shown signs of failure, so that the OS functions for detecting future memory failures could be tested. All of these were new steps that developers had never taken in previous releases, and they are now built into the Quality Assurance (QA) regression tests so that they can be applied to every single build of the OS hereafter.

Exploiting New RAS Functions in Xeon 7500/6500

In general, server hardware has become more reliable over time. Server designs increasingly build on highly integrated components, reducing complexity and hence the number of points of failure. The parts of servers that are particularly vulnerable to mechanical failure, such as storage, can be protected through techniques such as RAID and multipathing. Systems now build in redundancy for components such as fans to further improve reliability. However, critical failures can still occur in electronic components such as memory, CPUs, and I/O interfaces. For industry-standard systems to handle the most critical workloads, they require the ability to adapt to certain failures at this level, in some cases drawing on techniques that have traditionally been implemented in mainframe and other high-end environments.

The Xeon 7500/6500 processors introduced more than 20 new features to help improve the uptime of systems. These features are designed to react to faults that occur in low-level hardware, either in response to material failure or unexpected operational conditions (particularly "particle hits," in which random cosmic particles unavoidably interfere with the on-chip signals carrying data and instructions). Many of these new functions isolate their implementation at the level of silicon, so that their effects will apply independently of any operating system running on the server (see Table 1). For example, On-Die Error Protection protects the processor's registers (which store the results of every operation) against particle hits.

Xeon 7500/6500 also introduced a variety of measures to maintain the integrity of its Scalable Memory Interconnect (SMI), which enables data to move back and forth between processors and memory, and its Quick Path Interconnect (QPI), which enables multiple processors to work together in SMP configurations. These are the most critical connections in a server, and with

With the scalability limitations of x86 systems nearly eliminated, the focus of enterprise users is shifting to the reliability, availability, and serviceability (RAS) that can be achieved on industry-standard platforms.

Xeon 7500/6500, both SMI and QPI are protected with a variety of resiliency functions, including redundant clocks to synchronize the flow of data (so that if one clock fails, another can take its place); retry mechanisms, so that data will be retransmitted if a transaction fails for some reason; and the ability to perform failover or self-healing for the interconnect in the event a permanent failure in the path is detected. These protections are transparent to operating systems, which do not need any special enhancements to benefit from their resiliency measures.

Table 1. Support for Xeon 7500 RAS Features in RHEL 6

	RAS Feature	Requires OS Support	Supported in RHEL 6
New RAS Features in Xeon 7500/6500	Recovery from Uncorrected Data Errors (MCA)	Yes	Yes
	OS CPU On-Lining	Yes	Yes
	OS IOH On-Lining	Yes	Yes
	OS Memory On-Lining (Capacity Change)	Yes	Yes
	DIMM Isolation	Yes	Yes
	Physical CPU Hot Add and Remove	Yes	Yes
	Physical IOH Hot Add	Yes	Yes
	OS-Assisted Memory Migration	Yes	Yes
	QPI Poisoning/Viral Mode	Yes	Yes
	CPU Sparing/Migration	Yes	No
	Direct Connect Flash	Yes	No
	Scalable Memory Interconnect (SMI) Clock Failover	No	
	Scalable Memory Interconnect (SMI) Lane Failover	No	
	Scalable Memory Interconnect (SMI) Packet Retry	No	
	QPI Clock Failover	No	
	QPI CRC	No	
	QPI Self-Healing	No	
	QPI Packet Retry	No	
	Single-Core Disable for Fault Resilient Boot	No	
	On-Die Error Protection	No	
	Out-of-Band Access to Uncore MCA Registers	No	
Existing RAS Features in Xeon Architecture	Memory Board Hot Add	Yes	Yes
	Mirrored Memory Board Hot Add/Remove	Yes	Yes
	Intra- and Inter-Socket Memory Mirroring	Yes	Yes
	Static Hard Partitioning	Yes	Yes
	PCI Express Hot Plug	Yes	Yes
	Memory Demand and Patrol Scrubbing	Yes	Yes
	DIMM and Rank Sparing	Yes	Yes
	DRAM SDDC	No	

Red Hat's kernel development team works closely with all of the major hardware vendors to plan, develop, and test distributions of RHEL.

Other new Xeon 7500/6500 resiliency functions require enhancements in the operating system to take full advantage of their protection. These include:

- Machine Check Architecture (MCA) – MCA enables the operating system to take action in response to certain kinds of processor-level errors that cannot be recovered from automatically in hardware. Perhaps the most important application of MCA is for the OS to dynamically isolate memory areas that have suffered single-bit (i.e., "correctable") errors, so that software no longer risks using memory pages residing in these unreliable areas. Most systems can detect and automatically correct single-bit failures in hardware with error-correcting code (ECC) memory, but a second "double-bit" error can result in a system crash. With dynamic-memory resilience based on MCA, the operating system registers repeated single-bit failures in software so that it can isolate affected areas before fatal double-bit errors occur. To accomplish this, the operating system and diagnostic tools continuously check for memory errors reported by the MCA to determine whether a single-bit hard (or repeating soft) error has occurred, and which pages are affected. If a single-bit error is identified, the affected memory page is marked as bad (i.e., "poisoned") so that it is no longer used, even after reboots. In-flight processes that are associated with those pages are killed, and the error is logged for later analysis.
- DIMM Isolation – DIMM Isolation is a processor-level function for "indicting" entire DIMM memory modules that have been determined to be corrupt. The operating system marks any pages associated with these DIMMS as bad, so that they are no longer used by applications. This information can also be made available to an administrator for any statistics they may be logging or for potential action during a planned hardware maintenance window.
- Hot Addition and Removal of Physical CPUs and I/O Devices – These dynamic reconfiguration capabilities can reduce the amount of planned downtime needed for hardware maintenance and upgrades. Hardware reconfigurations, such as adding or removing processors and I/O devices, have traditionally required that servers be powered down for the duration of the maintenance procedure. Xeon 7500/6500 is designed to allow CPUs and I/O devices to be physically added and removed at the level of the QPI interface without interrupting the processor's operation. Dynamic reconfiguration requires specific support in the operating system, since the kernel needs to be aware at all times of what physical devices are attached.[2]
- Logical Component On-Lining – Separate from dynamically reconfiguring physical processors, memory, and I/O, in some cases it may be necessary for the operating system to dynamically add logical processors, memory, and I/O to the pool of resources it manages in a way that is independent of their physical counterparts (for example, virtual machines may dynamically be assigned additional resources by a hypervisor). The Xeon 7500/6500 provides the necessary abstractions – which the operating system should know how to interpret – for logically bringing CPUs, memory, and IO Hubs (IOH) online without interrupting processing.
- OS-Assisted Memory Migration – An existing feature for the Xeon architecture is the ability to configure spare "mirror" DIMM memory modules that can take over should a primary DIMM fail (see DIMM Rank and Sparing below). To take advantage of this capability, the operating system needs to be able to switch over to the spare DIMMs when appropriate.
- QPI Poisoning/Viral Mode – Similar to the way that the operating system can respond to MCA for fencing off corrupted memory, it can determine when single-bit errors are occurring in the QPI interconnect, and quarantine lines so that they are no longer used, preventing fatal double-bit errors from happening during communication between processors. The operating system should try to capture these errors, correct them on the fly by initiating automatic retries on I/O requests, or alternatively, prevent the corrupted connections from being used in the future.

[2] However, note that physically removing a CPU module may also require physical removal of memory, which is packaged together with processors in many server designs. Online removal of memory, a notoriously challenging operating system engineering problem, is not yet supported in RHEL, which may make it impractical to support physically uncoupling processors online.

The new RAS capabilities in Xeon 7500 raised the bar somewhat for executing [the development] process with RHEL 6. For this release, Red Hat allocated far more development resources to RAS than in previous releases.

RHEL 6 contains the necessary code to perform most of these RAS functions in conjunction with the Xeon 7500/6500 processors, with just a few exceptions. For example, RHEL 6 does not yet support Direct Connect Flash, which provides a QPI interface for Flash memory devices. Another function that Red Hat targeted for inclusion in RHEL 6, but was not able to implement in time for the release, is CPU Sparing and Migration. This function allows a processor to be held in reserve in case another processor fails. The operating system continuously monitors the health of processors, and when some threshold in error conditions is reached (predetermined by Red Hat, or tunable by customers), or when a hard failure occurs as flagged by the MCA, the environment that is running on the existing CPU can be migrated to the standby processor. However, Intel itself has not yet declared CPU Sparing ready for production in Xeon 7500/6500, so it is not yet possible for RHEL 6 to fully exploit it. These features are planned in future updates to RHEL; customers interested in them should reach out to Red Hat for an accurate current list of capabilities that takes into account the ongoing updates that Red Hat provides as part of its subscription model.

A number of the RAS functions in Xeon 7500/6500 were already introduced in earlier x86 processors. Again, some of these are transparent to operating system, and do not require any special support in RHEL 6. For example, DRAM Single Device Disable Code (SDDC) protects individual memory modules from two sequential errors by applying a Hamming code, providing resiliency that is transparent to the operating system. Other RAS functions do require some support by the operating system to work properly, and these capabilities are implemented in RHEL 6:

- Memory Board Hot Add is the ability to add a physical memory module to a system without interrupting its processing.
- Inter- and Intra-Socket Memory Mirroring makes it possible to mirror memory so that if a DIMM fails, its access can be redirected to another dedicated DIMM that is kept on standby.
- Mirrored Memory Board Hot Add/Remove enables standby memory modules to be added or removed when mirrored memory is implemented, without interrupting the processing of a system.
- DIMM and Rank Sparing is another approach to protecting against memory failure that does not require each DIMM to have its own dedicated standby DIMM. Instead, the operating system monitors a "rank" of memory in the primary DIMMs for single-bit errors, and when an error threshold is exceeded, it copies the contents of that memory rank to the spare rank of memory. The operating system takes the rank of failed memory offline, puts the spare rank online, and subsequently uses it instead of the failed rank. Much of this functionality is implemented by server providers in hardware, but RHEL 6 has the necessary code to support this capability in the operating system.
- Memory Demand and Patrol Scrubbing gives the operating system visibility over functions in memory controller hardware that continuously troll memory modules for errors (previously, this function had to be performed on the memory modules themselves).
- Static Hard Partitions allow multiple instances of an operating system to run simultaneously within a single server, using processor-based mechanisms to enforce isolation between the instances in hardware. Compared with software-based partitioning mechanisms such as virtual machines, hard partitions can potentially provide more robust barriers between instances, preventing a catastrophic failure in one partition from affecting others. Hard partitions were already supported in RHEL 5, but with Xeon 7500/6500, Intel is introducing a new variant of its partitioning, in which there is a stronger dependency on the basic input/output system (BIOS) of x86 servers. The technology is still under development at Intel, but RHEL 6 already has the necessary code to support it.
- PCI Express Hot Plug allows I/O devices based on PCI Express (PCI-E), such as disk adapters and network cards, to be added or removed without rebooting the operating system.

RHEL 6 provides several channels for delivering diagnostic information originating from hardware.

Increasing Visibility over System Health with Improved Error Reporting

A key requirement for maintaining the highest levels of availability in a system is collecting and analyzing post-mortem data about failures when they do occur, and interpreting run-time data for warning signs of potential failures before they occur. The new RAS functions in Xeon 7500/6500 can yield a wealth of data about the health and performance of key system components related to processors, memory, and I/O. On traditional high-end systems, this information was typically captured, logged, and then fed to system vendors through a phone-home technique, or perhaps locked up in non-volatile memory that a service technician could reference for diagnostics after a system failure. On industry-standard systems, though, some users (or their consultants) may prefer to perform their own analysis of diagnostic data. Thus, it becomes important for operating systems such as RHEL to provide user-space packages that can present the collected runtime data to administrators, so that they can perform their own predictive analysis using their preferred system management software.

RHEL 6 provides several channels for delivering diagnostic information originating from hardware. First, the MCA log captures all types of errors detected in CPU and memory, and presents them to the end user. The log can be accessed by administrators for purposes of predictive failure analysis either through home-grown tools, or through third-party management tools. Much of the log focuses on handling errors in ECC memory, but it also detects and reports PCI bus parity errors.

RHEL 6 also implements ACPI Platform Error Interface (APEI), which handles the exchange of error information between a server's BIOS and the operating system. Some system vendors prefer to implement the front line of hardware error handling at the BIOS level. In these cases, the Advanced Configuration and Power Interface (ACPI) traps hardware errors in BIOS, and then lets the BIOS make an intelligent decision about how to process the error. The BIOS will take action as needed, and then pass that information back to the operating system. With APEI, the operating system can accept that feedback from the BIOS and pass it on to administrators.

Finally, both RHEL 5 and RHEL 6 implement Advanced Error Reporting (AER) for collecting data on failures related to PCI Express peripherals. Just as with CPU errors and memory errors, administrators want to have visibility over errors in I/O devices so they can perform their own predictive failure analysis. In the past, if errors started appearing on a peripheral, and the board manufacturer provided enough intelligence in the firmware, administrators were notified that there was a problem, but that was all. No additional information was given; all diagnostic data was internal to the peripheral, and administrators had no visibility over details about the failure condition. Now, with PCI AER, administrators gain access to far more information about failure conditions when sporadic errors arise in I/O devices, giving them the ability to perform predictive failure analysis on I/O.

Red Hat had already started work on PCI AER in RHEL 5, and integrated the capability with RHEL 5.5. That code had to be pushed upstream, into the 2.6.30 kernel, in order to integrate the necessary error-reporting infrastructure. A lot of work had to go into PCI stack, so that it could capture events that had not been there before. Further, for PCI advanced error reporting to be fully effective, peripherals have to be adapted so that they pass the errors to the redesigned PCI stacks in operating systems such as RHEL 6. Red Hat worked with a several vendors to update their drivers with the necessary extensions, including Intel, Emulex, Qlogic, and a number of Infiniband suppliers. Red Hat educated these vendors on the requirements for their devices, and worked with them on updating drivers with the ability to generate the appropriate error data. These efforts have paid off. While RHEL 5 had only eight peripherals supporting AER, dozens are available for RHEL 6.

Since 2006, Red Hat has continuously been optimizing RHEL to control virtual machines, changing the memory management and I/O scheduler to become more efficient for handling virtual machine traffic and resource prioritization.

Increasing Uptime with Virtualization

Virtualization continues to have a major impact across the IT industry, and it is becoming a standard operational capability in enterprise and datacenter environments. Virtualization has been used on mainframes and other high-end platforms for decades, and on x86 systems, users are becoming comfortable deploying virtual machines to host increasingly critical workloads. The virtualization functions of a server platform are related to its RAS qualifications in a number of ways. First, virtualization can be an essential tool for consolidating a number of smaller systems onto larger servers, and then sharing the computing resources of the larger servers with different groups of users. The ability to deliver satisfactory service levels for these groups, while also maintaining strong isolation between the different workloads, becomes a key requirement for the effectiveness of a platform's virtualization functions.

Virtualization can fundamentally improve the overall reliability of an infrastructure. When used for consolidation, virtualization enables fewer physical servers to be deployed, which reduces the footprint for potential hardware failures that result in unplanned downtime. The servers that are deployed can be configured with HA features (such as redundancy) and hot-plug components to reduce downtime. Virtualization can greatly simplify high-availability processes. Traditional HA implementations require applications and their dependencies to be adapted so that they can be restarted on backup systems – which is a notoriously complex and error-prone process. Virtual machines essentially turn servers into data by capturing the entire state of a running system in easy-to-manage packages. As a result, an entire workload can easily be relaunched simply by restarting the virtual machine on which it is hosted.

Virtual machines also allow workloads to be migrated from one host to another relatively easily and with minimal downtime, simply by copying a workload's quiesced state to another host, and then restarting it. More sophisticated hypervisors allow virtual machines to be migrated "live" (i.e., without interrupting their processing), so that applications can continue to deliver service while they are being rehosted on a different server. The ability to migrate virtual machines from one host to another with minimal interruption to their processing provides yet another means to reduce planned downtime. Such migration allows workloads to be temporarily moved so that hardware maintenance can be performed on the hosts with minimal disruption. When coupled with HA clustering functions, virtualization can be used to restart workloads on a backup host in the wake of a primary host failure – dramatically simplifying the implementation of disaster recovery (DR) procedures. The ability to migrate virtual machines across a network simply by copying files also helps to improve service levels by dramatically simplifying the implementation of server load balancing, whereby workloads are migrated from heavily loaded servers to less-busy machines.

While some might argue that the use of live migration reduces the importance of keeping individual servers running at all times, live migration does not necessarily protect workloads against unplanned downtime, in which a host ceases to operate due to component failure or operator error (live migration requires both the source and destination hosts to be fully operable at the time of migration). Moreover, even in cases of planned downtime – when large numbers of virtual machines are hosted on a single server – the process of migrating all the virtual machines to another host may become fairly time-consuming. In these cases, it may be preferable to avoid shutting down the virtual machine host, even while maintenance is performed.

Since a single server outage can potentially affect many workloads at once, the use of virtualization increases the importance of the single-system reliability functions of the host. The scalability of the host platform also becomes important, because the more virtual machines hosted on a single platform, the greater the economies of scale achieved by consolidating with virtualization.

With RHEL 6, the network stack has been moved into the kernel to improve throughput, increasing performance by 10%, and more importantly, reducing the latency incurred by virtualized network operations.

Red Hat's Strategic Virtualization Platform: KVM

All of these uptime factors were considered in Red Hat's adoption of Kernel-based Virtual Machine (KVM) as its strategic virtualization platform. Virtual machine platforms such as Microsoft Hyper-V and Xen variants (such as Citrix XenServer and Oracle VM for x86) are all based on light-weight, dedicated software layers called hypervisors, which allocate basic computational resources such as CPU cycles and memory to virtual machines. Hypervisors are typically deployed in tandem with variants of standard operating systems to control hypervisor functions, and to perform I/O. By contrast, KVM integrates the management of virtual machines directly into the standard RHEL operating system, taking advantage of the native virtualization functions built into x86 processors (i.e., Intel VT-x and AMD-V) as much as possible. With KVM, virtual machines are represented as Linux processes, which can be managed the same way as any other application or service.

Red Hat started adding KVM virtualization functions into its operating system one year ago, with RHEL 5.4. Earlier releases of RHEL also included support for the Xen hypervisor, but starting with RHEL 6, KVM will be the only virtualization function supported (Red Hat will continue to support existing Xen implementations for the full lifecycle of earlier RHEL releases). RHEL 6 improves KVM in a number of ways. The move to a new kernel in RHEL 6 brings new features that cannot be back-ported into RHEL 5, but are quite important for virtualization, including a new I/O scheduler, tickless kernels, MCE logging, hardware poisoning, huge page support for hardware poisoning, and the integration of hardware poisoning with KVM. All of these features will have a significant impact on the reliability and performance of virtualized environments. Because KVM inherits all of these features directly from the RHEL 6 kernel, guests operating inside of virtual machines will benefit transparently from their effects.

With KVM, Red Hat developers followed two design principles:

- They avoided reinventing the wheel. If a feature existed in the Linux operating system, it was reused for virtualization purposes whenever possible (by contrast, with the hypervisor approach, functions often had to be replicated in the hypervisor and in the controlling OS).
- They tried to ensure that as much functionality as possible is performed in hardware, which usually is more efficient.

Since 2006, Red Hat has continuously been optimizing RHEL to control virtual machines, changing the memory management and I/O scheduler to become more efficient for handling virtual machine traffic and resource prioritization. As a result, any enhancements that the Linux developer community accomplishes in the Linux operating system, KVM will get for free. Users benefit from this approach, because they will gain new features more quickly, and with a more mature implementation.

The KVM implementation in RHEL 6 introduces a number of enhancements, including:

- Scalability – Virtual machines can now be configured with up to 64 virtual CPUs (vCPUs) in RHEL 6, compared to 16 vCPUs in RHEL 5.
- Redesigned Network I/O Stack – In previous releases of RHEL, the network stack for KVM ran in user space. With RHEL 6, the network stack has been moved into the kernel to improve throughput, increasing performance by 10%, and more importantly, reducing the latency incurred by virtualized network operations.
- Asynchronous I/O – Asynchronous I/O allows the operating system to perform I/O operations for virtual machines in the background, so that VMs do not have to wait for an I/O operation to finish before they can proceed with other work. When virtualizing I/O-intensive workloads with a lot of IOPS, this function can improve performance by 10 to 20%.
- Virtualized Interrupt Controller – KVM in RHEL 6 uses the X2APIC protocol to expose a new kind of device to guests, removing the need to perform emulation of interrupts. Use of this device can increase performance in virtualized workloads by 5%.

The HA Add-On designed for use with RHEL 6 gains a number of improvements, including a vastly revamped GUI-based management interface; a robust infra-structure for determining cluster membership; and newer capabilities in the "fencing" mechanism that is used to maintain consistency of data when accessed by multiple cluster nodes.

- Resource Management – The new CGroups kernel function in RHEL 6 allows administrators to apply rules for precisely allocating resources to processes, or groups of processes. Since KVM implements virtual machines as Linux processes, this capability can be used to specify service levels for VMs. For example, if three virtual machines were configured, with one hosting a virtual desktop, one hosting a web server, and one hosting a database server, each VM would be a Linux process that can be regulated with CGroups. If the virtual desktop VM initiates an anti-virus operation, it will put unusually heavy stress on I/O for storage, which could choke off bandwidth available to the web server and database. With CGroups, it is possible to apply rules limiting the I/O resources that are assigned to the process for the virtual desktop machine, so that the other VMs can continue to function normally. The CGroups function has different classes of rules, including those applied to the CPU scheduler (in terms of priorities that can define amounts of CPU dedicated to one VM or another); memory (making it possible to isolate memory to a particular VM, or give preference for real memory vs. paging for a particular VM); and I/O and network bandwidth (making it possible to specify the amount of bandwidth given to individual VMs, or groups of VMs).
- Memory Management – RHEL 6 integrates KVM with Transparent Huge Pages (THP), which make it more efficient to move large amounts of memory in and out of virtual machines. Normally, Linux memory pages are 4 KB in size, so for example, moving 8 KB would require two I/O operations (IOPs). It is not unusual for virtual machines to consume multiple gigabytes, which would require a huge number of IOPs to transfer. While Linux has always supported huge memory pages, in previous releases administrators had to explicitly specify when to use large-size versus standard-size pages. RHEL 6 automates the management of page sizes, dynamically creating large pages as needed by merging together smaller pages to create huge pages, or dynamically breaking down large pages into small pages. For workloads such as databases, this automation can result in memory capacity savings of 20% or more, leading to better performance. RHEL 6 also implements Kernel Same Page Merging (KSM), in which the kernel continuously scans memory to look for pages that are identical. This function is coordinated with the THP function, so that page size reallocations can be initiated to create opportunities for sharing (i.e., it may be easier to find 4 KB pages that are identical than 2 MB pages, so it is worth reallocating to smaller pages, if that enables more sharing).
- Nehalem Extended Page Tables (EPTs) – EPTs guide the kernel on which pages to swap to disk – a critical factor in virtual machine implementations. KVM has supported memory overcommitment since RHEL 5.5, which means that all of the virtual machines on a host can be configured with more memory than is physically available on the host. Support for overcommitment inherently risks swapping to disk, though, which can dramatically slow the performance of a VM if it later needs to access memory that happens to be swapped out. Therefore, it becomes critical to swap only when necessary, and then swap only as much memory as necessary. The Xeon 7500/6500 processors have "age bits" that provide hints on which pages can be swapped out (i.e., those that are not in regular use by a virtual machine), helping KVM make more intelligent choices about when and what to swap.
- Hot-Plug Virtual Devices – In RHEL 6, KVM can dynamically add virtual processors, disks, and network adapters to virtual machine guests.
- Single Root I/O Virtualization Specification (SR-IOV) Integration – SR-IOV is a standard issued by the PCI working group for sharing access to physical I/O devices by multiple virtual machines. SR-IOV grants a direct "pass-through" channel to devices in order to minimize performance overhead. Red Hat introduced SR-IOV support in RHEL 5, and extended its implementation in RHEL 6 with a mode for combining it with vHost and *virtio* paravirtualized devices. By exposing the virtual PCI device as a paravirtualized device to the guest, administrators gain flexibility for relocating virtual machines without trading off performance. For example, it becomes possible to start the guest on a host that does not have a SR-IOV card, and migrate it to a host that does. In this way, VMs gain the performance benefits of PCI pass-through without being tied to a particular host (or losing the ability to perform live migration).

On Nehalem-EX, Red Hat can now offer systems that come close enough to traditional UNIX systems to knock down many of the last barriers to Linux adoption in datacenters.

HA Clustering with Red Hat HA Add-On

Administrators can use HA clusters to maintain the availability of operating system services and applications in the event of a failure that affects an entire system. HA clusters allow operations to continue by failing over to a backup system. HA clusters emphasize the use of standard building blocks (i.e., traditional servers) to construct "meta-systems" with some level of a single-system image that allows the cluster to be managed as a single logical environment. A cluster ensures service restoration within a reasonable time limit by enabling one or more servers to take over for a server that has crashed or stopped processing due to any failure (in hardware, software, or otherwise). By isolating faults on the failed node, the remaining nodes can continue providing service, keeping the overall clustered system in operation, albeit at reduced capacity. In some cases, clustering can also help with some management tasks by absorbing planned downtime in addition to cases of system failure. For example, a cluster could allow new software or hardware to be tested in a working system while still protecting the rest of the system from any resulting failures.

Red Hat offers a high availability clustering solution as an add-on, which is specifically designed for Red Hat Enterprise Linux. Officially referred to as the Red Hat Enterprise Linux High Availability (HA) Add-On, it enables failover for applications and services ensuring no single point of failure and data integrity. The HA Add-On, which has been historically known as the Red Hat Cluster Suite, first began shipping in 2003, and it is now a mature offering, supporting HA clusters with up to 16 nodes. The HA Add-On has three primary use cases: protecting typical open source software services such as NFS, Samba, Apache, Tomcat, etc. (which are pre-integrated with the HA Add-On); protecting third-party ISV applications; and protecting applications and services developed by users. The application that is made highly available with the HA Add-On does not need to be made aware that it is under the control of an HA clustering framework. As a result, an application can be seamlessly integrated with the HA cluster without alteration.

The HA Add-On designed for use with RHEL 6 gains a number of improvements, including a vastly revamped GUI-based management interface; a robust infrastructure for determining cluster membership; and newer capabilities in the "fencing" mechanism that is used to maintain consistency of data when accessed by multiple cluster nodes. The new release is also designed to be used with the updated KVM functions in RHEL 6. The HA Add-On in Red Hat Enterprise Linux can be used both to protect applications and services running within KVM-based virtual machines (i.e., restarting failed applications either in the same virtual machine, or another virtual machine) and to protect the virtual machines themselves (i.e., restarting an entire virtual machine when it fails, either on the same host where it originally resided, or on another host).

The IDEAS Bottom Line

The natural choice of platforms for users who required the highest levels of scalability was traditionally limited to either mainframes or high-end servers running the UNIX operating system. Now, the dramatic growth in performance of industry-standard hardware is giving users an unprecedented opportunity to maintain a choice of system suppliers without requiring them to compromise on high-end systems features. At the same time, many users are interested in deploying Linux to host their most critical workloads, hoping to minimize dependency on specific suppliers due to the inherently transparent nature of Linux technology. The use of Red Hat Enterprise Linux affords more hardware platform choices, while also allowing users to develop application portfolios and operating procedures that are consistent across multiple platforms. Some users in large organizations, in which multiple versions of UNIX are deployed, have set a goal of consolidating their UNIX systems over time on Linux distributions.

However, serious efforts to consolidate enterprise workloads on Linux have traditionally been limited by perceptions that Linux distributions could not match the environment users were accustomed to on their existing high-end systems. In many traditional datacenter environments, users associate high-end servers with massive scalability (through large numbers of cores in SMP configurations), robust partitioning technologies, and the ability to monitor for failure conditions at extraordinary detail. The combination of RHEL 6 and Xeon 7500/6500 yields a new class of platform that directly addresses these concerns, allowing administrators to put familiar RAS disciplines into practice while also benefiting from the volume economics of industry-standard systems.

The new resiliency functions in Xeon 7500/6500, coupled with the strengthened reporting mechanisms in RHEL 6, will give administrators the ability to predict when many failure conditions might occur, while allowing the operating system to automatically take its own measures to continue running in the event of low-level hardware failures. The new RAS capabilities in Xeon 7500/6500 and RHEL 6 help to protect systems not only against failure conditions in processors and memory, but also in I/O channels, which will become an increasingly critical concern on x86 servers that are capable of supporting a large number of hot-plug I/O slots. KVM will enable the deployment of virtualization in a way that directly exploits the reliability improvements in the processor and base operating system. Red Had HA Add-On complements the increased robustness of single systems and the reliability benefits of KVM with another line of defense against downtime.

All of these capabilities will give users the confidence to host more critical workloads on x86 servers, and larger numbers of workloads in consolidation scenarios, than ever before. On Nehalem-EX, Red Hat can now offer systems that come close enough to traditional UNIX systems to knock down many of the last barriers to Red Hat Enterprise Linux adoption in datacenters.

Americas
Ideas International, Inc.
800 Westchester Avenue
Suite N337
Rye Brook, NY 10573-1354
USA
Tel + 1 914 937 4302
Fax +1 914 937 2485

Asia/Pacific and Worldwide Headquarters
Ideas International Limited
Level 3
20 George Street
Hornsby, NSW, 2077
Australia
Tel +61 2 9472 7777
Fax +61 2 9472 7788

Europe, Middle East, Africa
Ideas International Europe
Milton Park Innovation Centre
99 Milton Park
Abingdon, Oxon OX14 4RY
United Kingdom
Tel + 44 (0) 1235 841 510
Fax + 44 (0) 1235 841 511

actionable intelligence
www.ideasinternational.com