

WHITE PAPER

KVM for Server Virtualization: An Open Source Solution Comes of Age

Sponsored by: IBM

Gary Chen

Al Gillen

October 2011

IN THIS WHITE PAPER

Server virtualization has quickly become a trusted, mainstream solution aboard x86 servers. KVM, which stands for kernel-based virtual machine, is the newest hypervisor to enter the virtualization market, and it joins a market that already has no fewer than four other major solutions in place. Like several other Type 1 hypervisors, it is embedded into an operating system (OS), namely Linux. However, the implementation of KVM into the OS is unique in several ways. This IDC White Paper examines the architecture, progression, and future of KVM as well as the related management software and surrounding ecosystem.

KVM OVERVIEW

KVM started out as an open source project created by the Israeli company Qumranet, which was acquired by Red Hat in 2008. KVM is implemented as a module within the Linux kernel. Curiously, the founders of Qumranet were focused on using KVM as the foundation for a centralized virtual desktop solution (commonly referred to as VDI) for Windows clients.

Red Hat, however, had different objectives and believed KVM to be a better server virtualization solution than the existing open source technology, Xen. Following its classic approach to software development, Red Hat embraced the incorporation of KVM into the upstream Linux code base.

IBM also started investing in KVM back in 2007 and has contributed to areas such as memory management, performance, and virtual I/O as part of the open source community. IBM has over 60 developers working on KVM and has also stated its commitment to support KVM as a virtualization environment across its server software portfolio.

Other Linux distributors also started investing in KVM and including it in their distributions. SUSE became a KVM contributor and released a technology preview of KVM in 2009, and today SUSE includes and supports SLES with KVM, both as a virtualization host and as a guest.

KVM officially became part of the mainline Linux kernel as of version 2.6.20 (released in February 2007), which had several implications:

- ☒ KVM is now an integral part of Linux and will eventually become part of all Linux distributions, once those distributions pick up the later version of the kernel. In reality, today, most Linux distributions are on kernel versions newer than 2.6.20 and therefore have KVM capabilities — regardless of whether KVM is formally supported in the commercialized versions or not.
- ☒ KVM is able to leverage the entire Linux development community as any improvements to Linux in general also benefit KVM. It also means that Linux developers can now presume that KVM is available to be utilized.
- ☒ KVM revs with the Linux kernel and does not have any easily determinable version of its own. As of this writing (August 2011), the Linux kernel is at 2.6.38, implying that KVM has undergone 18 revisions since its initial acceptance into the Linux kernel. Commercial Linux vendors, though, may attach their own version numbers to their KVM releases.
- ☒ KVM inherits the drivers and broad hardware compatibility of Linux, allowing it to run nearly anywhere Linux for x86 runs, and no unique effort is required to produce new drivers specifically for KVM. While KVM was originally developed for x86, ports are in progress for other architectures.

KVM ARCHITECTURE

KVM is a loadable kernel module within the Linux kernel that allows the Linux operating system to function as a Type 1 bare metal hypervisor. A hypervisor can be thought of in many ways as a very specialized operating system designed to run VMs instead of arbitrary applications. Other than the virtualization aspect, the rest of a hypervisor deals with standard OS tasks such as managing memory, scheduling processes, handling drivers, doing I/O, etc. The KVM module implements within Linux the essential capabilities to do the magic that allows virtual machines to function, but it adopts the philosophy of not reinventing the wheel and uses the established and proven Linux OS functions for the rest. By not having to rewrite basic functions, developers can focus their efforts on optimizing Linux for VM processes — not replicating those functions within the hypervisor code stack. It also means that all the advances in Linux as an OS apply to virtualization as well:

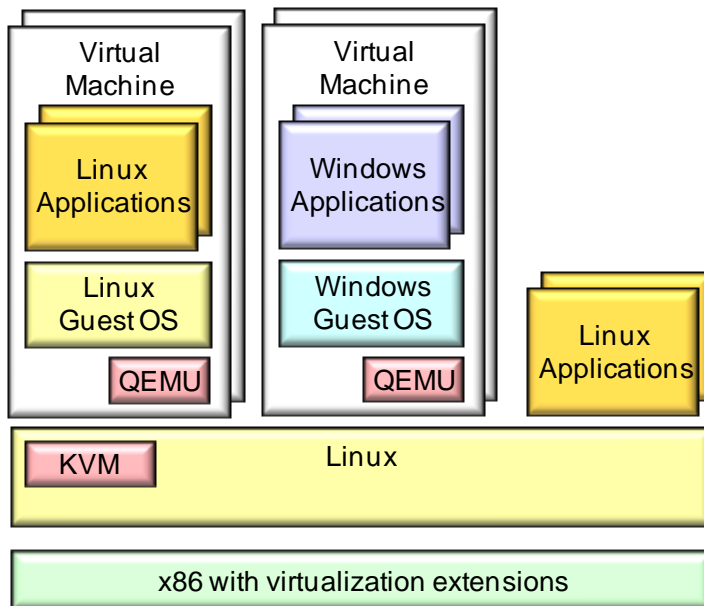
- ☒ **Scheduling, resource control, and memory management.** Virtual machines under KVM in Linux are simply treated as any other running process during execution. All the advances in Linux that deal with core OS functions of handling processes, such as scheduling, resource control, and memory management, have been — and will continue to be — applied to VMs. For example, improvements to the scheduler such as process priority, CFS, and control groups allow fine-grained controls to implement QoS for VMs by managing the resources such as CPU, memory, and I/O just as users can do with processes today. Another example would be adding NUMA support to Linux, which allows Linux processes and, by inheritance, VMs to address large amounts of memory.

- ☒ **Storage.** VM images are treated like any other Linux file on a disk device. Thus storage support is any type of storage that is supported by Linux today, which includes local disk, a variety of file systems, NAS, iSCSI, and SAN. Improvements to the storage I/O stack and support for the storage vendor's infrastructure all carry over to KVM and allow VMs to leverage Linux's robust and proven storage stack.
- ☒ **Hardware support.** KVM inherits the entire Linux device ecosystem and is able to access any device that Linux supports. QEMU — a related open source project — is used to provide I/O device virtualization inside the virtual machines created by KVM. Linux is a popular, general-purpose OS and thus is compatible with a very wide variety of devices and systems, allowing KVM to run on nearly every hardware platform today. Scalability enhancements to Linux that allow it to run on systems with a large number of CPUs/cores and very large RAM sizes also carry over to KVM, allowing it to scale as Linux scales.
- ☒ **Security.** KVM also is able to leverage the Linux security model, SELinux, which essentially "sandboxes" processes so that if a process becomes compromised, the problem is limited to that process and does not compromise the entire system. Because VMs are run as normal Linux processes under KVM, this protection extends to VMs as well. SELinux sandboxes VMs not only from each other but also, more importantly, from the hypervisor, protecting against any potential hypervisor vulnerabilities that could emerge.

While strongly tied to Linux, KVM also supports Windows as a first-class guest (see Figure 1).

FIGURE 1

KVM Architecture



Source: IBM, 2011

The roots of Windows support started from the very inception of KVM. Qumranet initially developed KVM for hosting Windows desktops. Further, Red Hat entered into a virtualization interoperability agreement with Microsoft in February 2009, which ensures that each company will test and support its operating systems aboard the other company's hypervisor. This raises the potential that performance won't be compromised when running on the other vendor's hypervisor and will ensure that support will be provided without finger-pointing. SUSE and Microsoft have a similar virtualization interoperability agreement as part of their broad collaboration on Windows and Linux interoperability and support.

VIRTUALIZATION MANAGEMENT

A key tenet of successful virtualization deployments is management. One aspect that hampered Xen was a fractured management interface, which eventually led to incompatible versions of Xen in the market. Further, there is widespread adoption of open source Xen, particularly among hosting providers and cloud providers. IDC notes that KVM proponents must be careful not to tread down that same path.

Linux and KVM have standardized on libvirt and libguestfs as the base APIs for managing virtualization and images (which support other hypervisors besides KVM). Higher-level tools, such as virsh and virt-manager, are then built on top of libvirt.

Commercially, Red Hat has focused on Red Hat Enterprise Virtualization Manager (RHEV-M) as its main platform for managing virtualization. However, because RHEV-M was a .NET application inherited from the Qumranet acquisition, it is a closed source platform that also must be hosted on Windows Server. With the upcoming 3.0 release, in beta at the time of this writing, Red Hat is seeking to make significant changes to the code and engage the larger community, including the following efforts:

- ☒ The 3.0 release will be a JBoss Java application, allowing it to run on a fully open source stack (Linux and Java). This will remove the requirement for Windows Server, which many diehard Linux shops interested in KVM had difficulty implementing.
- ☒ The 3.0 release will be widely available in a Web-downloadable version. Previous releases were available only to partners because of the more complex implementation process of the old architecture.
- ☒ The 3.0 code will be made into an open source project. This will allow the KVM management system to take advantage of the open source development model and community that also drive the KVM hypervisor, allowing both aspects of the virtualization platform to be under the same model. This brings the final piece of Red Hat's KVM ecosystem to the open source community and is consistent with Red Hat's fundamental approach to software development.

In parallel, IBM has focused on adding support in its Tivoli and Systems Director systems management portfolio for KVM. This includes Tivoli Provisioning Manager, Tivoli Service Automation Manager, and a technical preview of VMControl for KVM.

SUSE has also increased its support for KVM and virtualization management in its product portfolio, including KVM provisioning and virtual machine management support in SUSE Manager, KVM virtual server support in SUSE Linux Enterprise High Availability Extension, and KVM image deployment in SUSE Studio.

IBM and Red Hat have also stated their intention to drive adoption of the open source virtualization technology by jointly developing key virtualization and cloud management interfaces and using the APIs in their respective management products, including Red Hat Enterprise Virtualization Manager, IBM Systems Director, and Tivoli software. The APIs will address cloud, datacenter automation, virtual storage and networking, virtualization security, and virtual appliance management.

COMMERCIAL KVM IMPLEMENTATIONS

Not surprisingly, Red Hat is a leading contributor to the KVM code base, and Red Hat is playing a vocal role in bringing KVM to commercially supported, enterprise Linux distributions. KVM was first featured in Red Hat Enterprise Linux (RHEL) 5.4 and is a major component in RHEL 6. Red Hat packages KVM in two different ways:

- ☒ **As part of RHEL.** This full Linux OS is Red Hat's flagship product. KVM is integrated into Linux as previously discussed, and RHEL features the complete Linux kernel, tools, and applications typical of a full Linux distribution.
- ☒ **Standalone, as Red Hat Enterprise Virtualization Hypervisor (RHEV-H).** RHEV-H is a subset of RHEL, including only the necessary components required to run KVM. It strips out unnecessary items such as some kernel drivers, tools, and applications that aren't relevant to KVM to reduce the overall distribution size and also the attack surface for greater security. It runs off of a read-only live CD, ensuring that the system always boots secure, unmodified code.

With either model, the KVM code base is identical, as RHEV-H is simply a subset of RHEL. RHEV-H will be the preferred deployment model for most customers because of its appliance-like nature. Deploying RHEV also requires little knowledge of Linux. Customers that are comfortable deploying the full Linux OS or that require the full OS for add-ons, such as certain third-party agents, can choose RHEL. RHEL can also operate in an interesting hybrid mode, running both VMs and "bare metal" applications simultaneously, supporting some unique use cases. Red Hat revises and releases KVM on a six-month schedule, with each point release of RHEL.

SUSE has chosen a different approach and supports both KVM and Xen as part of SUSE Linux Enterprise Server from SLES 11 SP1 onward. SUSE also continues to leverage its relationships with ISVs; in April 2011, SAP announced support for SAP applications running on SUSE Linux Enterprise Server with KVM.

KVM, as part of the mainline Linux kernel, is also essentially present in every Linux distribution today. Many other major Linux distributions, such as Ubuntu, also support and promote KVM as a hypervisor. KVM is now an inherent feature of Linux and will be widely available from all vendors in the Linux community, providing easy access to the technology for any Linux customer and a wide choice of vendors.

THE OPEN VIRTUALIZATION ALLIANCE

In May 2011, BMC Software, Eucalyptus, HP, IBM, Intel, Red Hat, and SUSE formed the Open Virtualization Alliance (OVA). A month later, the alliance announced the addition of 65 new members. As of this writing, there are over 200 members in the alliance, with hundreds more applications for membership in the queue.

The goal of the alliance is to promote KVM in the marketplace by growing the ecosystem, educating the market about KVM, and documenting best practices and case studies. OVA, in its current incarnation, is not a type of certification, does not create standards, and does not do interoperability or compatibility testing and enforcement.

Each of the alliance's four governing members — HP, IBM, Intel, and Red Hat — has a seat on the board of directors and pays higher dues than general members. The rest of the alliance comprises general members that must apply for membership, be approved by a majority of the board, and pay dues. Dues have been waived for the first year by the governing members in order to onboard members as quickly as possible and build the foundation for the alliance. In the future, dues paid by members will fund marketing and educational activities to promote KVM and the ecosystem surrounding it.

The current membership is extremely diverse, representing service providers as well as server, storage, networking, management, OS, business application, and cloud software vendors. The explosive growth of OVA in just its first few months and the large queue of potential additional members are very good signs of the interest the market has in KVM and in promoting a diverse virtualization market landscape. It will be critical for OVA to quickly organize itself and begin producing content to further the stated goals of promoting KVM.

KVM CLOUD DEPLOYMENTS

Hypervisors and virtualization technology have grown in many ways, with use cases constantly expanding. One logical evolution has been to serve as the foundation for clouds of many types: internal, external, private, and public. Key clouds that have been built on KVM include the following:

- ☒ The IBM Research Compute Cloud (RC²) is IBM's largest internal cloud deployment and functions as a global lab for IBM engineers to conduct cloud computing research. The current deployment is on over 200 iDataPlex nodes using KVM, with about 2,000 concurrent VMs at any given time, and consumes 100TB of storage. Since its inception, the RC² cloud has delivered over 15 million VM hours to researchers in 39 countries, with internal chargeback per VM hour to IBM business units.
- ☒ IBM has also used KVM in its customer-facing public cloud, called SmartCloud Enterprise. The cloud spans eight datacenters; each datacenter has one or more pods, and each pod is composed of about 150 KVM nodes. In total, the cloud has approximately 1,000 KVM hosts running 6,000 VMs. The cloud is managed with IBM's Tivoli management software, which supports KVM as a first-class hypervisor. IBM chose to deploy KVM with RHEL (rather than use RHEV-H) because of the requirements of its Tivoli management agents and supports a choice of RHEL, SLES, or Windows guests.

- ☒ CloudForms is Red Hat's infrastructure-as-a-service (IaaS) platform that allows the creation of private and hybrid clouds. It features robust application life-cycle management and also focuses on workload portability to address lock-in concerns. The platform fully supports KVM as well as other industry hypervisors.
- ☒ OpenShift is Red Hat's platform-as-a-service (PaaS) offering built on KVM and focused on enabling developers to quickly build and deploy cloud applications. It supports a wide variety of languages and development frameworks and also emphasizes portability like CloudForms to allow developers a choice of cloud provider to avoid lock-in.
- ☒ The cloud ecosystem for KVM is rapidly growing, and many members of the OVA that provide cloud infrastructure software or cloud services support KVM as the underlying virtualization technology.

CHALLENGES/OPPORTUNITIES

- ☒ **Challenge: KVM maturity.** KVM is the newest hypervisor on the market and thus has had the least time to mature. KVM has certainly progressed at an impressive rate and by being part of the main Linux kernel. It can leverage the considerable power of the Linux development community, which has proven over time that it can evolve projects very quickly.
- ☒ **Opportunity: Leveraging the reach of Linux.** Because KVM is an inherent part of Linux, KVM will reach everywhere that Linux exists today, making it available to anyone using Linux. The Linux ecosystem is far larger than Red Hat alone, and the nonpaid Linux server operating system market accounts for 42% of the overall Linux market (for more detail, see *Worldwide Linux Operating Environment 2011–2015 Forecast: Accelerating Toward the Era of Cloud*, IDC #227778, April 2011).
- ☒ **Challenge: Expanding beyond Linux.** Being so closely tied to Linux comes with benefits and challenges. Many customers will see KVM as a solution primarily for virtualizing Linux on Linux, even though Windows is well supported. And even though vendors may package KVM for consumption by non-Linux users, many still may perceive KVM as a Linux solution that requires Linux skills. IDC notes that the majority of the x86 server market is not Linux.
- ☒ **Challenge: Creating a robust management ecosystem.** While many of the base-level APIs and libraries have been established, higher-level tools continue to progress. Red Hat's open sourcing of RHEV-M in hopes of unifying the community around a central KVM management interface will be key for KVM and the open source ecosystem. However, enterprise systems management is a multilayered onion, and KVM support must also work its way into the larger management software market, including the proprietary tools available from companies such as IBM, CA Technologies, and BMC.
- ☒ **Opportunity: The public cloud.** Public cloud providers have shown an early preference for using open source software to build their offerings because of the cost and the open, freely modifiable nature of the code. As cloud adoption grows, there is the opportunity for Linux and KVM to grow their share of the cloud provider market.

- ☒ **Challenge: Monetizing the cloud.** Service providers may be showing interest in and adopting open source, but monetizing it is a more difficult problem. Service providers traditionally have used their leverage to dictate pricing, and many consume the open source code and support it themselves, without using a commercial version.
- ☒ **Challenge: Ecosystem support.** Virtualization is no longer just a server consolidation tool; it is the foundation for the dynamic datacenter and cloud. Virtualization interconnects with every part of the datacenter (servers, storage, networking, security, management, etc.), and the ecosystem support for any hypervisor must be far reaching and diverse in order to succeed today. With considerable competitors in the hypervisor market and given KVM's late entrance, KVM will have to fight for priority among the virtualization ecosystem vendors.

CONCLUSION

KVM has made impressive progress since its inclusion in the Linux kernel a little more than four years ago. It has been able to tap into the power of the Linux open source development model to achieve widespread distribution and leverage the developer base to mature the code rapidly. The newly formed Open Virtualization Alliance has seen a surge of interest and will be key in developing KVM's ecosystem.

The road map for KVM and its management tools is also ambitious, with many enhancements planned over an aggressive release cycle. KVM and other complementary open source technologies have also done well in the cloud, drawing interest from that segment of the market and gaining key wins in clouds such as IBM's Research Compute Cloud and SmartCloud Enterprise. Existing users of Linux need to seriously evaluate KVM because it is an integral part of the Linux road map, but KVM is also advancing to become an interesting general virtualization platform outside of just Linux.

Copyright Notice

External Publication of IDC Information and Data — Any IDC information that is to be used in advertising, press releases, or promotional materials requires prior written approval from the appropriate IDC Vice President or Country Manager. A draft of the proposed document should accompany any such request. IDC reserves the right to deny approval of external usage for any reason.

Copyright 2011 IDC. Reproduction without written permission is completely forbidden.