KVM Virtualized I/O Performance

Achieving Unprecedented I/O Performance Using Virtio-Blk-Data-Plane Technology Preview in SUSE Linux Enterprise Server 11 Service Pack 3 (SP3)

Khoa Huynh, IBM Linux Technology Center
Andrew Theurer, IBM Linux Technology Center

June 2013

Version 1.3

Table of Contents

Executive Summary

The Kernel-based Virtual Machine (KVM) hypervisor has earned a reputation as the highest performing hypervisor in virtualization benchmarks, holding the top seven results in SPECvirt_sc2010 [1] and recently achieving leadership results on SAP's 2-tier SD benchmark [2] and TPC-C benchmark [3]. One of the key ingredients to this success is KVM's ability to handle the high I/O rates required by enterprise workloads, such as databases, ERP systems, and low-latency financial trading applications that are running in virtual machines.

A new I/O virtualization technology, called **virtio-blk-data-plane**, allows KVM to handle even higher storage I/O rates. It is available as a Technology Preview in SUSE® Linux Enterprise Server 11 Service Pack 3 (SP3). This paper describes a test setup that included an IBM® System x3850 X5 host server with QLogic® QLE 256x Host Bus Adapters (HBAs) and SUSE Linux Enterprise Server 11 SP3 running on both the host server and a single KVM guest. It also details test results that demonstrate the highest storage I/O performance ever reported for a single virtual machine (guest): 1.37 million I/O operations per second (IOPS) at 8KB I/O request size and more than 1.61 million IOPS at 4KB and smaller request sizes. This achievement is the result of a collaborative effort between IBM's Linux Technology Center's Performance organization and the major Linux distributions. These results show that KVM is ready for the most demanding enterprise workloads.

1. Introduction

A hypervisor is an object that manages virtual machines (guests) running on a physical machine (host). It can be a computer program, firmware, or hardware that provides the end user the ability to create, configure, and manage virtual machines that operate as if they were physical machines.

1.1. The KVM hypervisor

The Kernel-based Virtual Machine (KVM) project represents the next generation in open-source hypervisors. KVM is fully integrated into the Linux operating system both as a host and a guest. Unlike other hypervisors, KVM makes no distinction between running in either host or hypervisor mode. This duality in design has helped KVM to rapidly mature into a stable, high-performing hypervisor, positioned to outperform other hypervisors available on the market today.

The first design principle includes the following:

- Leverage all hardware-assisted virtualization capabilities provided by Intel® Virtualization Technology (VT) and AMD® Secure Virtual Machine (SVM)
- Feature the latest hardware virtualization extensions, including:
    - Hardware nested paging (EPT/NPT)
    - Pause loop exiting and pause filtering
    - I/O off-load features, such as secure PCI pass-through using Intel VT-D or AMD I/O Memory Management Unit (IOMMU)
- Exploit hardware capabilities while keeping the KVM virtualization overhead to the absolute minimum

The second design principle includes the following:
- Leverage the Linux operating system
- Fulfill many components required by a hypervisor, such as memory management, scheduler, I/O stack, and device drivers by reusing optimized, off-the-shelf Linux implementations

The Linux kernel, with its 20 years of development, is the industry leader in terms of performance and availability. The Linux process scheduler, for example, provides completely fair scheduling (CFS) that is optimized to manage complex workloads and NUMA systems, while offering low latency, high performance determinism, and fine-grained Service Level Agreement (SLA) for applications. By placing the KVM hypervisor directly into the Linux kernel, all of these services and advantages have a direct, positive impact on the hypervisor performance.

1.2. SUSE® Linux Enterprise Server 11 Virtualization

SUSE Linux Enterprise Server is a modular, enterprise-class Linux server operating system, runs on five processor architectures and is suitable for a wide range of mission-critical workloads. It is optimized to run on leading hypervisors and supports an unlimited number of virtual machines (guests) per physical system with a single subscription, making it the perfect guest for virtual computing.

Included with SUSE Linux Enterprise Server 11 Service Pack 3 (SP3) are the latest open source virtualization technologies, Xen and KVM. With these hypervisors, SUSE Linux Enterprise Server can be used to provision, de-provision, install, monitor and manage multiple virtual machines (VMs) on a single physical system. SUSE Linux Enterprise Server 11 SP3 with Xen or KVM acts as a virtualization host server (VHS) that supports VMs with their own guest operating systems. The SUSE VM architecture consists of a hypervisor and management components that constitute the VHS, which runs many application-hosting VMs. In KVM, where the Linux kernel acts as the hypervisor, the management components run directly on the VHS.

1.3. Technology Preview – Virtio-Blk-Data-Plane

SUSE Linux Enterprise Server 11 Service Pack 3 (SP3) introduces, as a Technology Preview, a new I/O virtualization technology for KVM, called *virtio-blk-data-plane*. It accelerates I/O operations going through the para-virtualized I/O block driver (*virtio-blk*) with dedicated per-device threads. This approach, shown in **Figure 1**, allows the block I/O processing to run concurrently with other device emulation, and therefore, achieves some of the best I/O performance results to date. Device emulation in KVM is provided by the Quick EMUlator (QEMU) running in user space.

The key concept of the virtio-blk-data-plane approach is that by isolating the virtio-blk processing into dedicated threads, synchronization with other components can be avoided. Consequently, these dedicated threads, called *virtio-blk-data-plane threads*, do not need to acquire the global mutex that protects the hardware emulation code in QEMU when they submit guest I/O operations to the host, resulting in higher performance. There is a dedicated virtio-blk-data-plane thread for each block device in the guest. For even better performance, virtio-blk-data-plane also exploits the *ioeventfd / irqfd* mechanism, which decouples the I/O processing from the guest's execution. The Asynchronous I/O (AIO) support in the host Linux kernel is used by virtio-blk-data-plane to process the actual I/O requests on behalf of the guest. Due to the architectural changes required to move virtio-blk processing to dedicated threads, this technology preview in SUSE Linux Enterprise Server SP3 currently limits some storage features:

- Only raw image files are currently supported.

- Storage migration, hot unplug, I/O throttling, image streaming, and drive mirroring are currently not supported.

These limitations apply only to virtio-blk-data-plane and not to the existing virtio-blk device emulation.
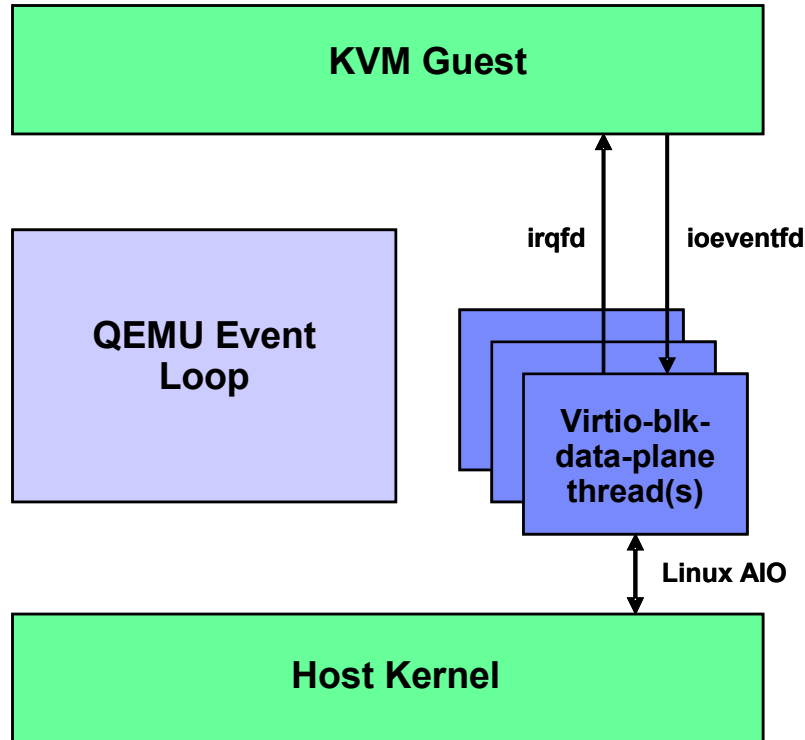
Figure 1. Virtio-blk-data-plane

1.4. IBM® System x® Servers

IBM System x servers support Microsoft® Windows®, Linux, and hypervisors. System x servers are intelligent systems, designed to reduce costs and complexity for enterprise workloads. With the introduction of eX5 – IBM's fifth-generation industry-leading enterprise X-Architecture® servers – IBM engineers have redefined x86 servers by expanding their capabilities. A member of the eX5 server family, the x3850 X5 is a scalable 4-socket, 4U, rack-optimized enterprise server that delivers the following benefits to enterprise customers:

- High memory capacity (up to 3TB, 3 times the memory capacity of other 4-socket x86 servers, using the industry-unique IBM MAX5 memory expansion unit)
- Processor scalability up to 8 sockets (up to 80 processor cores) by connecting two 4-socket x3850 X5 systems together and doubling all system resources (including up to 6TB of memory, using two MAX5 memory expansion units)
- The broadest range of network and storage support in the industry for ultimate flexibility and choice
- Support for IBM eXFlash solid-state storage technology for extreme storage I/O performance
- Integrated Emulex 10 GbE Virtual Fabric Adapter with capability for upgrades to Fiber Channel over Ethernet (FCoE)
- Fifth-generation eX5 chipset design enhancements, built on the latest X-Architecture blueprint

- Balanced systems for virtualization, database and enterprise workloads
- Workload-optimized systems with customizable configurations for target workloads
- Greater performance and utilization at a lower total cost
- Mainframe-inspired reliability
- Simplified power and systems management with an energy-smart design and remote access

Figure 2 shows the front exterior of the IBM System x3850 X5 server.

Figure 2. IBM System x3850 X5 Server

1.5. Motivation

IBM customers drive enterprise workloads, such as databases, ERP systems, and low-latency financial trading applications. In the past, these workloads were seldom virtualized in production due to scaling and time-sensitive barriers, so they were unable to exploit the many benefits of virtualization, such as hardware abstraction, live migration, dynamic resource allocation, and more. Proving that KVM is able to sustain high I/O rates is very critical in enabling the migration of these workloads into the virtualized environment.

2. Test setup

2.1. Test Hardware

To demonstrate how KVM can handle extremely high I/O rates, it was necessary to set up a storage back-end that was capable of delivering at least one million I/O operations per second (IOPS). For a diagram of the test environment, refer to **Figure 3.**

The KVM host server was an IBM System x3850 X5 with four Intel Xeon® E7-4870 processors and 256 GB of memory. Each E7-4870 processor had 10 cores running at 2.40 GHz. The x3850 X5 server had seven available PCI slots, each of which were fitted with QLogic® QLE 256x Host Bus Adapters (HBAs) . Each of these adapters had two ports, each supporting full-duplex, 8-Gigabit-per-second data links.

Each QLogic HBA in the KVM host was directly connected to a unique Fiber Channel SCSI target server. For more information about the Fiber Channel SCSI target server, see _http://linux-iscsi.org_. Each SCSI target server had four 15-GB RAM disks configured so that these RAM disks would appear as Logical Unit Numbers (LUNs) at the SCSI host (KVM host server). As a result, from the KVM host's perspective, the storage back-end had a total of 14 (7 PCI slots x 2 ports) PCI devices and 56 (14 PCI devices x 4 RAM disks) storage LUNs.

2.2. Workload

The Flexible I/O (FIO) benchmark (_http://linux.die.net/man/1/fio_) was used to generate disk I/O workloads and measure the resulting I/O rates, throughput, and latency. This workload had the following FIO parameter settings:

• Direct I/O operations

• Asynchronous I/O operations (engine = libaio)

• Random read and write operations (50% reads, 50% writes)

• I/O request sizes = 512 bytes, 1KB, 2KB, 4KB, 8KB

• One job per storage LUN

• Queue depth = 32

Both random read and write operations were included in the workload for two reasons. First, a workload with both random reads and writes would be more realistic and similar to actual enterprise workloads, such as database applications. Secondly, having both reads and writes in the workload allowed the ability to fully exploit the full-duplex data links supported by the QLogic HBAs. A range of I/O request sizes, from 512 bytes to 8KB, was also considered as these are the typical I/O sizes used in many real-world applications.

The FIO workload was first run directly on the KVM host server to determine the maximum I/O rates that this test setup could support. These "bare-metal" results indicated that our storage

setup could deliver up to 200,000 IOPS per SCSI target server, or a total of 1.4 million IOPS with all 7 SCSI target servers, using 8KB requests. This was determined to be sufficient for the KVM storage performance testing.
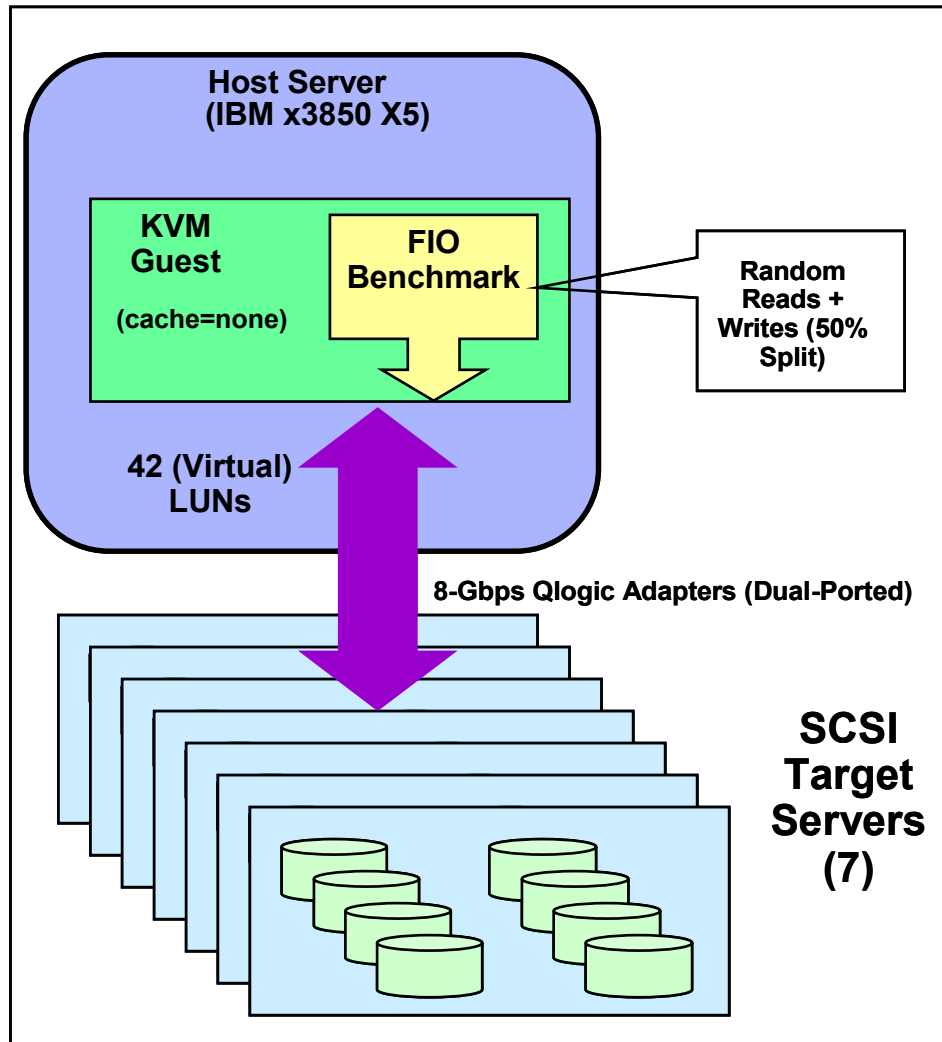

Figure 3. The test setup

2.3. KVM Configuration

To achieve the best possible I/O rates for the KVM guest, the virtio-blk-data-plane feature was enabled for each LUN (a disk or partition) that was passed from the host to the guest. To enable virtio-blk-data-plane for a LUN being passed to the guest, the *x-data-plane=on* option was added for that LUN in the *qemu-kvm* command line used to set up the guest. For example:

*qemu-kvm -drive if=none,id=drive0,cache=none,aio=native,format=raw,file=<disk or partition> -device virtio-blk-pci,drive=drive0,scsi=off,**x-data-plane=on***

After the guest finished booting up, all I/O operations going through the para-virtualized block driver (virtio-blk) would use the fast I/O path enabled by the virtio-blk-data-plane feature for this specific LUN.

SUSE Linux Enterprise Server 11 Service Pack 3 (SP3) was used on both host and guest.

3. Results

In this test, the goal was to determine the maximum I/O rates that could be sustained with a single KVM guest. A very large guest was configured with 40 virtual CPUs to test this goal. Because our FIO workload was not very memory-intensive, 8GB of memory was configured for the guest. More specifically, the single-guest configuration included the following:
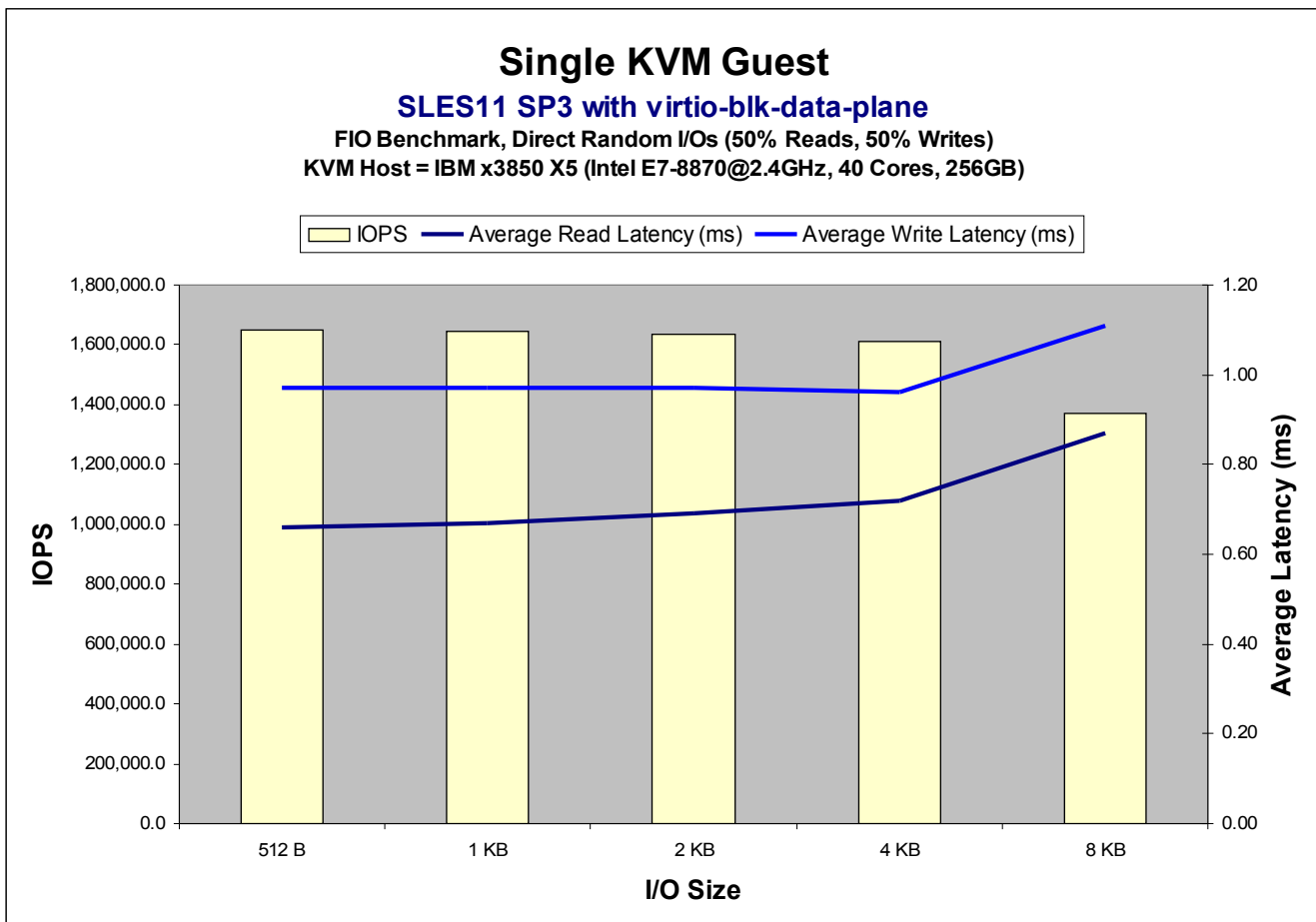
- Host Server: IBM System x3850 X5
  - 4 Intel Xeon E7-4870 processors (40 cores at 2.40 GHz), 256 GB memory (total)
  - SUSE Enterprise Linux Server 11 SP3
- Storage:
  - 7 QLogic QLE 256x (8 Gbps, dual-ported) connected to 7 SCSI target servers
  - 56 LUNs
- KVM Guest (Virtual Machine):
  - 40 virtual CPUs, 8 GB memory
  - 42 virtual LUNs
  - SUSE Enterprise Linux Server 11 SP3
- FIO Workload
  - Random reads and writes (50% reads, 50% writes)
  - 1 job per LUN
  - Direct I/O operations
  - Engine = libaio
  - Queue depth = 32

To optimize the virtualized I/O performance, the following performance tuning steps were done prior to the test runs:

- Leveraged caching efficiency in the host processors by binding dedicated virtio-blk-data-plane threads to specific CPUs in the host
- Used deadline I/O scheduler in the host
- Exploited interrupt-coalescing capability of the QLogic HBAs
- Disabled entropy random contribution from block devices
- Disabled all cgroup and CPU delay accounting in both host and guest
- Switched the clock source in the guest to TSC

Figure 4 shows the number of I/O operations per second (IOPS) and the average latencies for random read and write operations across several I/O request sizes. For a single guest, KVM with virtio-blk-data-plane was able to achieve **1.37 million IOPS** using 8KB I/O requests and **1.61 million IOPS** using I/O requests that were 4KB or less. These I/O rates were actually limited by the physical bandwidth of our test storage setup.

The average latencies for random reads and writes were very consistent up to 4KB I/O size – at about 0.7 milliseconds (ms) for reads and 1.0 ms for writes. The average latencies for 8 KB requests were just slightly higher – 0.9 ms for reads and 1.1 ms for writes – due to larger data transfer times as we approached the physical bandwidth of the test storage setup. This shows that KVM can sustain very high storage performance across all typical I/O sizes.

Single KVM Guest
SLES11 SP3 with virtio-blk-data-plane
FIO Benchmark, Direct Random I/Os (50% Reads, 50% Writes)
KVM Host = IBM x3850 X5 (Intel E7-8870@2.4GHz, 40 Cores, 256GB)

Figure 4. I/O rates and average latencies across multiple I/O request sizes

Figure 5 shows how these storage I/O rates compare to the most recent claims for other competing hypervisors. In a recent TechNet Webcast [4], Microsoft announced new benchmark performance results for iSCSI using its Hyper-V® hypervisor, Windows Server 2008 R2, and Intel Internet Server Adapters. These results indicated that Hyper-V could achieve an aggregate rate of 700,000 IOPS using 10 virtual machines on a single host at 512-byte I/O request size. As the I/O request size increased, the I/O rates dropped – to around

300,000 IOPS at 8KB request size. In a blog just before VMworld® 2012 [5], VMware® demonstrated that it could sustain almost 1.1 million IOPS using 4KB I/O requests in a single virtual machine running on a vSphere™ 5.1 host. With virtio-blk-data-plane technology, KVM could achieve more than 1.6 million IOPS for a single guest at the same 4KB I/O request size. This means that KVM with virtio-blk-data-plane could achieve I/O rates that are **52% higher** than VMware vSphere 5.1, although these tests used different guest operating systems and hardware setups. It is clear that the virtio-blk-data-plane technology allows KVM to deliver much higher I/O rates than both of its major competing hypervisors.
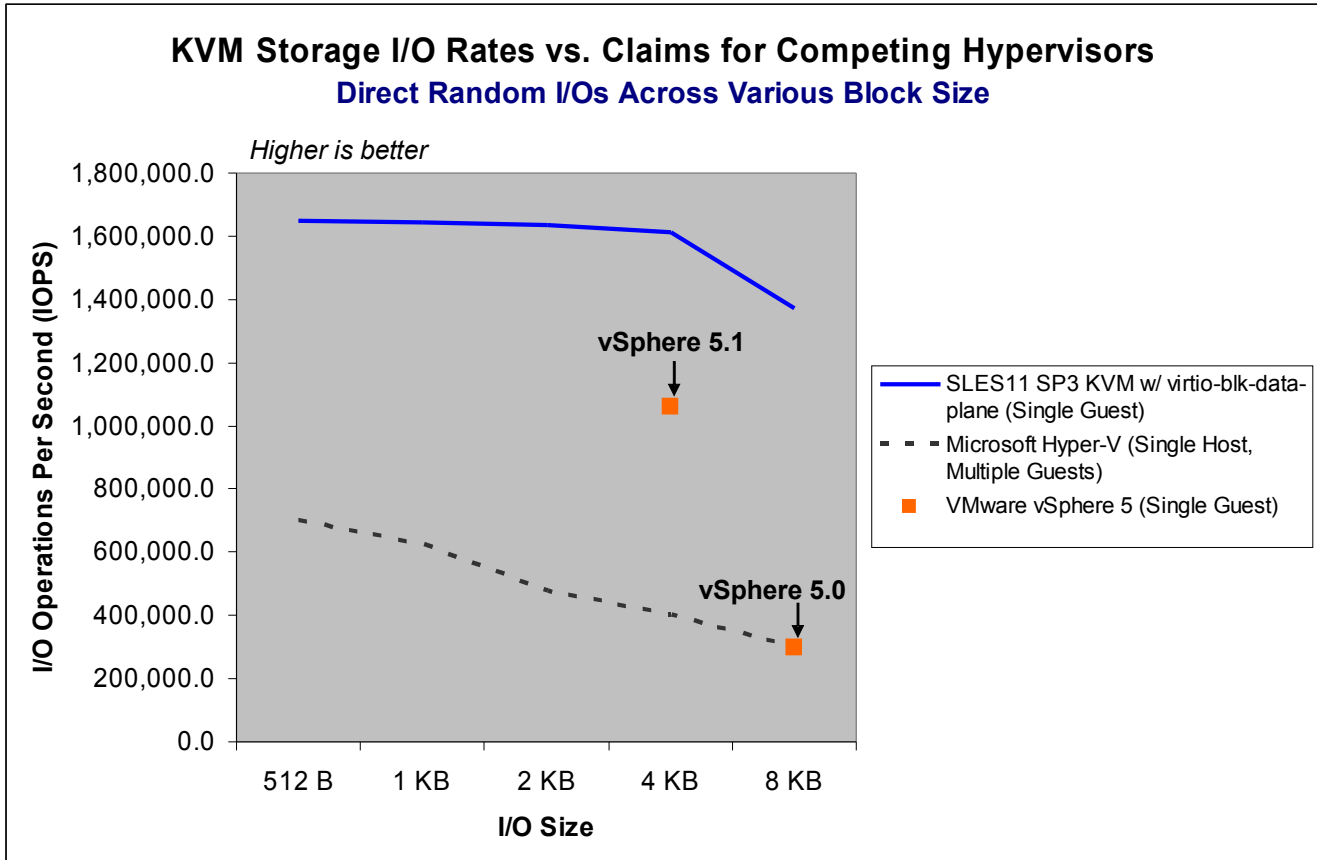
Figure 5. I/O rates at different I/O request sizes

Summary

Leveraging new silicon capabilities, with tight integration into the Linux kernel, KVM outperforms other competing hypervisor solutions in many aspects. With the virtio-blk-data-plane technology preview for KVM in SUSE Linux Enterprise Server 11 Service Pack 3 (SP3), a single KVM guest can handle up to **1.37 million I/Os per second (IOPS)** for 8KB random I/O requests and more than **1.61 million IOPS** for smaller requests, approaching the "bare-metal" limit of the test storage setup. These I/O rates are more than **50% higher** than the highest claims for VMware vSphere 5.1. The results in this paper have proven that KVM is ready for enterprise workloads that demand extremely high I/O rates, very low latencies, and consistent storage I/O performance.

References

[1] SPECvirt_sc2010 Results Published by SPEC,
http://www.spec.org/virt_sc2010/results/specvirt_sc2010_perf.html

[2] SAP SD Standard Benchmark Application Results, Two-Tier Internet Configuration,
http://www.sap.com/solutions/benchmark/sd2tier.epx

[3] First-ever Virtualized x86 TPC-C Result, Two-Socket Configuration,
http://www.tpc.org/tpcc/results/tpcc_result_detail.asp?id=113022201

[4] Microsoft TechNet webcast. "Maximizing Hyper-V iSCSI Performance With Microsoft and
Intel." https://msevents.microsoft.com/CUI/EventDetail.aspx?culture=en-
US&EventID=1032432957&CountryCode=US

[5] VMware, Inc. "1 million IOPS on 1 VM", August 27th, 2012.
http://blogs.vmware.com/performance/2012/08/1millioniops-on-1vm.html