

VMware vs KVM

A hypervisor is a combination of software, hardware or firmware that creates, executes and manages virtual machines (VMs). The computer that runs a hypervisor is known as its host, while each VM on the host is known as a guest. A hypervisor provides the guests with a virtual operating platform that manages their operating systems (OSs), allowing multiple OSs to share virtualized resources on the same host.

The ability to share resources is one of the most significant reasons for a business to implement hypervisors. However, the variety of hypervisors currently available can make this decision process challenging. The selection of a hypervisor often comes down to a choice between VMware and Kernel-based Virtual Machine (KVM). VMware is the name of a company that develops a range of hypervisors, including the enterprise-class ESXi. KVM is an infrastructure for the Linux kernel that provides it with the capabilities of a hypervisor.

The following points of comparison can help organizations in need of a hypervisor choose between VMware and KVM.

- Performance
- Integration
- Cost
- Complexity
- Maturity
- Scalability
- Functionality support

1 Overview

A hypervisor virtualizes a computing environment, meaning the guests in that environment share physical resources such as processing capability, memory and storage encompassing a private cloud. Each guest runs its own operating system, which makes it appear as if it has its own resources, even though it doesn't. The efficient sharing of resources requires the physical processor to support virtualization, which is called AMD-V for AMD processors and VT-x for Intel processors.

A hypervisor needs to effectively isolate each guest, such that a guest's operation can't affect the other guests running on the host. This requirement means that a hypervisor must accurately emulate the physical hardware to prevent guests from accessing it except under carefully controlled circumstances. The method that a hypervisor uses to do this is a key factor in its performance.

Hypervisors often use “paravirtualized” (PV) drivers to emulate physical hardware, which act as hardware such as storage disks and network cards. These drivers are OS-specific and often specific to a particular hypervisor. PV drivers can improve a hypervisor’s performance by an order of magnitude.

2 Performance

Hypervisors may be classified into two types, which can impact their performance. Type 1 hypervisors, also known as “bare metal” hypervisors, run directly on the physical hardware, and the OS of each guest runs on top of the hypervisor. These hypervisors typically allow some guests to control the hypervisor. Most businesses use Type 1 hypervisors.

A Type 2 hypervisor, also known as a hosted hypervisor, runs within an OS that runs on the physical hardware. The OS of each guest then runs on top of the hypervisor. Desktop hypervisors are usually Type 2 hypervisors.

Xen is probably the best example of a pure Type 1 hypervisor, although ESXi is clearly a Type 1 hypervisor as well because it isn’t an application that’s installed onto an OS. ESXi includes a kernel and other OS components that it integrates into the native OS.

The classification of KVM is more challenging because it shares characteristics of both types of hypervisor. It’s distributed as a Linux component, meaning that a Linux user can start KVM from a command line or graphical user interface (GUI). These methods of starting KVM make it appear as if the hypervisor is running on the host OS, even though KVM is actually running on the bare metal.

The host OS provides KVM with a launch mechanism and establishes a co-processing relationship with it, allowing KVM to share control over physical hardware with the Linux kernel. KVM uses the processor’s virtualization instructions when it runs on x86 hardware, allowing the hypervisor and all of its guests to run directly on the bare metal. The physical hardware performs most of the resource translations, so KVM meets the traditional criteria for a Type 1 hypervisor.

A Type 1 hypervisor should outperform a Type 2 hypervisor, all other factors being equal. Type 1 hypervisors avoid the overhead that a Type 2 hypervisor incurs when it requests access to physical resources from the host OS. However, other factors also play an important role in a hypervisor’s performance. For example, ESXi generally requires more time to create and start a server than KVM. ESXi also has slower performance when running servers, although this difference may be insignificant for typical loads.

3 Integration

Hypervisors use different methods to communicate with the host's physical hardware. KVM uses an agent installed on the host to communicate with hardware, while ESXi uses VMware's management plane to communicate with hardware. The process does provide the advantage of allowing ESXi to access other VMware products that use this management plane. However, it also requires ESXi to use VMware's control stack, which can increase hardware requirements.

Close integration with the host OS is the primary reason that Linux developers typically prefer KVM, which was incorporated into the Linux kernel shortly after its release in 2007. In comparison, Xen didn't become officially part of the Linux kernel until 2011, eight years after its initial release. Linux developers are also more likely to use KVM because Red Hat and other Linux distributors have adopted it in preference to other hypervisors. Illumos is an open-source OS based on OpenSolaris that also chose KVM over other hypervisors when it added support for hardware virtualization.

4 Cost

KVM clearly wins over VMware on the basis of cost. KVM is open source, so it doesn't incur any additional cost to the user. It's also distributed in a variety of ways, often as part of an open-source OS.

VMware charges a license fee to use its products, including ESXi. It's able to do this because VMware was the first company to release enterprise-class virtualization software and is still the market leader in this segment. Its brand is therefore still relevant to an business's end users, regardless of what developers may think about it. An ESXi user must also purchase a license to use vSphere, VMware's suite of tools for cloud computing that uses ESXi. Additional software licenses may be needed, which will further increase the cost of implementing ESXi.

IBM performed some calculations regarding the total cost of ownership (TCO) for KVM and VMware in 2012. These calculations showed the KVM's TCO was typically 39 percent less than VMware, although the actual TCO will depend on site-specific factors such as the operational setting and workload. This difference in TCO indicates that cloud service providers will probably want to implement KVM on at least one cluster, regardless of the other factors to consider.

5 Complexity

A comparison of KVM and VMware also show a clear difference in the size of the code base, which affects a hypervisor's maintenance costs. KVM was initially released to take advantage of processor extensions that allowed them to virtualize guests without translating binary code. This origin meant that the first stable release of KVM was essentially a lightweight virtualization driver, with little more than 10,000 lines of code (LOC).

VMware is believed to have over 6 million LOC, although this fact can't be verified since its source code isn't publicly available. This total doesn't directly affect performance since VMware uses hardware extensions to virtualize guest. Nevertheless, its original code has never been completely rewritten, resulting in a more complex code base than KVM.

6 Maturity

KVM and ESXi are both highly mature and stable. KVM has been part of the Linux kernel for over a decade, and ESXi has been publicly available since 2006. However, KVM is more widely deployed since it's open source and is included in many packages such as Redhat Enterprise Virtualization (RHEV). KVM also supports more features than any other hypervisor.

7 Scalability

KVM is generally more scalable than VMware, primarily because vSphere has some limitations on the servers it can manage. Furthermore, VMware has added a large number of Storage Area Networks (SANs) to support various vendors. This feature means that VMware has more storage options than KVM, but it also complicates VMware's storage support when scaling up.

8 Functionality Support

Hypervisors vary greatly in their support of functionality. Network and storage support are especially important and are probably more important than any other factor besides OS integration. It shouldn't come as a surprise to learn that ESXi's support for other VMware products is unmatched by any other hypervisor. On the other hand, KVM offers more options for network support than VMware.

9 Summary

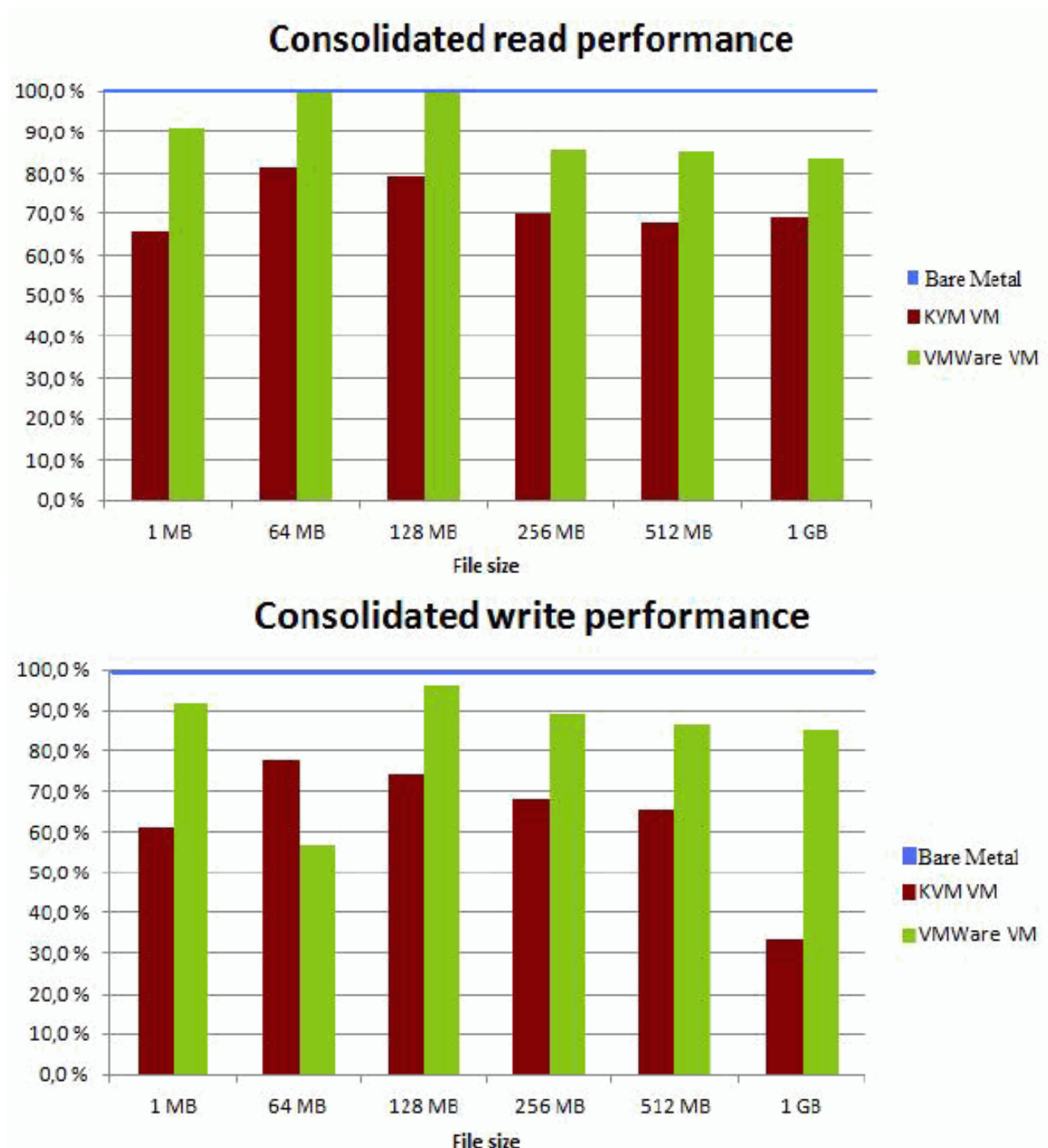
KVM is typically the most popular choice for users who are concerned about the cost of operating each VM and less interested in enterprise-level features. This rule primarily applies to providers of cloud and host services, who are particularly sensitive to the cost and density of their servers. These users are highly likely to choose open-source hypervisors, especially KVM.

The tight integration with the host OS is one of the most common reasons for developers to choose KVM, especially those who use Linux. The inclusion of KVM in many Linux distributions also makes it a convenient choice for developers. KVM is also more popular among users who are unconcerned about brand names.

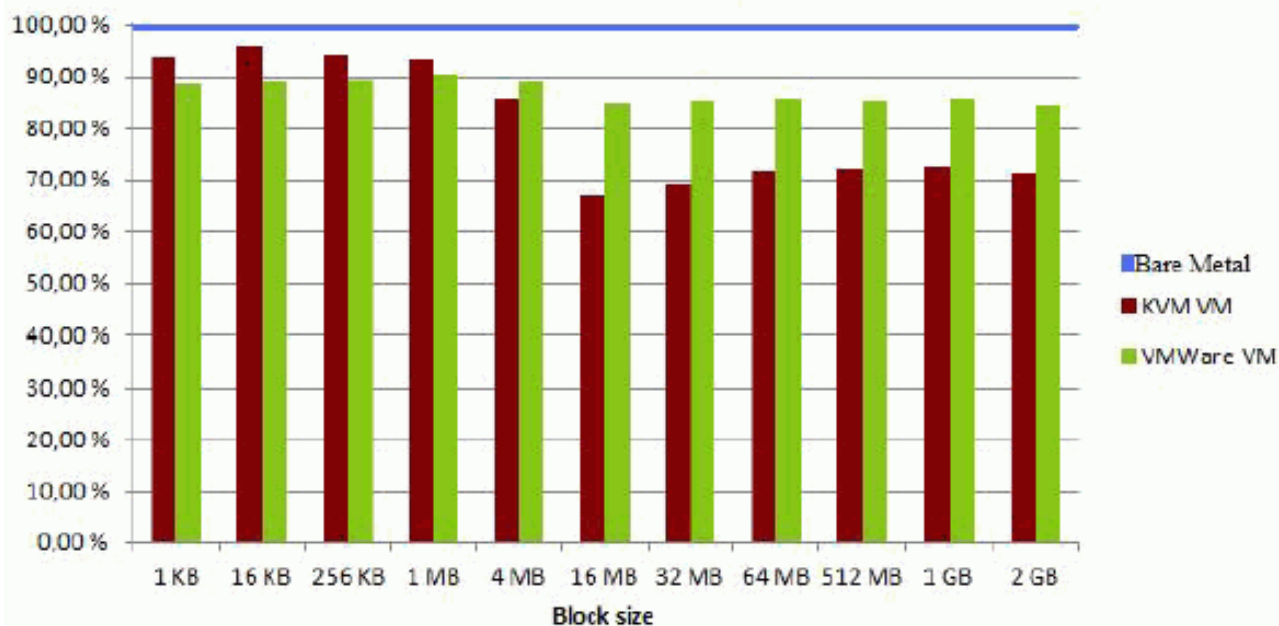
The VMWare I/O disk performance is mostly from 20 to 30% better than KVM, with a few exceptions. And generally the VMWare I/O performance is 10-15% less than the Bare Metal performance. The memory performance overhead is relatively smaller.

KVM performs better than VMWare for block sizes of 4MB and less, while the results show the opposite for block sizes larger than 4MB. When testing pure ALU usage, there is almost no virtualization overhead. There was some overhead for the other UnixBench CPU tests and in all these cases VMWare was performing better than KVM. Our general conclusion is that the virtualization overhead is less for VMWare than for KVM.

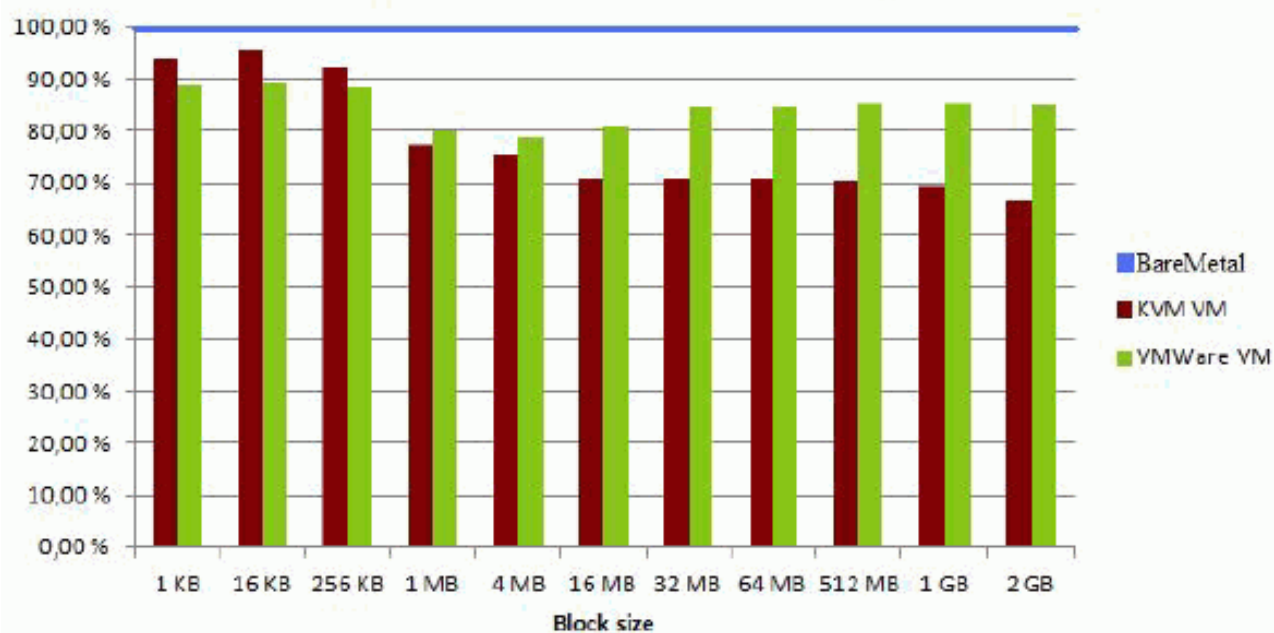
A few of the graphs in the thesis are provided below for illustrating the performance differences of VMware ESXi 5.0 compared to KVM on the RHEL 6.1 kernel and Bare Metal (no virtualization). The tests were run on a Dell Poweredge R710 server with the guest being benchmarked also running RHEL 6.1.



Integer and Float read performance



Integer and Float write performance



Ishak AYAD