

WHITE PAPER

Version 2.0
February 2010

Performance Report Hyper-V

Pages 29

Extract

Server virtualization is at present one of the major topics of the IT environment. Today, modern hardware is often so powerful that it is not run to full capacity by a single software application. Virtualization is ideal for old software solutions, particularly where old hardware is replaced.

This document looks at the use of Hyper-V in Windows Server 2008 R2 as a virtualization solution with the aim of making recommendations for practical use and forecasting performance data.



Content

Introduction.....	2	WebBench	9
What is a virtual server?	2	Measurement environment.....	10
Typical virtualization architectures.....	3	Performance Analyses	13
General overview of Hyper-V.....	5	CPU and Memory	13
Virtualization forms	5	Disk I/O.....	14
I/O Structure	6	Data throughputs	15
Hyper-V on the basis of Windows Server 2008 R2		Database server application scenario.....	17
Enterprise (full installation)	7	Network	18
What's new in Windows Server 2008 R2.....	7	Data throughput.....	18
Measurement methods.....	8	Web server application scenario.....	20
Benchmarks.....	8	Scaling of web server VMs	22
Iometer.....	8	Application mix	23
Netperf	8	Energy-saving features	25
vServCon	8	Summary	27
SPECjbb2005	8	Literature	28
SysBench.....	9	Contact.....	29

Introduction

The subject of server consolidation is one of the principal topics when it comes to saving costs in the IT environment. A better and more effective server workload as well as a reduction in the number of servers are called for. Nowadays individual applications are allocated to a dedicated server so as to prevent any reciprocal interference. Therefore, with a large number of applications the number of servers in a data center also increases. As a result of this allocation the available computing performance frequently lies idle because the individual applications do not utilize the server to the full.

Using virtualization enables several virtual servers to be consolidated on a single physical server. Thus the use of virtual machines even enables several different operating systems, e.g. various Linux derivatives or Windows versions or even 32-bit and 64-bit versions of an operating system, to be run in parallel on the same physical server.

A further advantage of virtual servers is the possibility of operating legacy systems or existing applications together with their environments in virtual machines.

Currently available on the market are various popular virtualization products for x86/x64-based servers, this document concentrates primarily on Hyper-V with Windows Server 2008 R2 Enterprise.

It begins with a general introduction to virtualization technology and its current architectures and, based on this, is followed by an architecture overview of Hyper-V. The main part consists of performance analyses of various application scenarios for individual virtual machines as well as scaling measurements for several virtual machines and the consideration of an application mix.

What is a virtual server?

Virtualization is a technology that allows several operating systems to be run on a single physical server at the same time. Virtualization can be realized with the help of hardware or software. In the case of software-based virtualization a virtualization program is used to insert an additional layer, the so-called virtualization layer, on the physical server between the actual system resources and the virtual servers, also known as virtual machines. The hardware of the physical server is made available to the virtual servers in a suitable form via the interface of the virtualization layer. In this way, the virtual machines can be fully separated and isolated from each other.

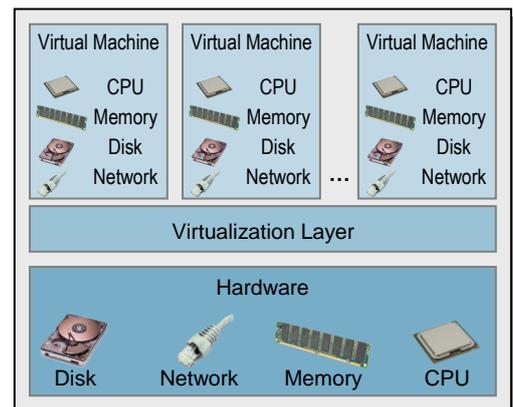
Hardware resources - emphasis here is always placed on the four core components CPU, memory, network and disk resources - are mapped in every virtual machine. Each access of a virtual machine to and from the physical hardware of the host server passes through the virtualization layer.

As a means of distinguishing the servers, the physical server is also denoted as the "host" and the virtualization layer as the hypervisor (and also as "Virtual Machine Monitor"/VMM), on which the virtual servers, also known as virtual machines (VM), run with their guest operating systems. The virtualization layer completely separates the virtual machines from the host hardware and its hardware/driver dependencies. In such a configuration it is possible, for example, to run a virtual server which has been created on a PRIMERGY RX300 S4, on a PRIMERGY RX600 S4.

The volume of hardware resources of the virtual machine can frequently be changed manually using the virtualization program. In this way, it is possible to change CPU resources during ongoing operations. Depending on the requirements of the virtual machines they can be allocated more or less CPU time.

Each virtual machine must be seen as a separate server, which can in turn be run on the host fully independently of the other virtual machines. The virtual machines are isolated from each other to the effect that data security is ensured even with business-critical and confidential data.

Virtual machines (VMs) can in the simplest case consist only of a configuration file, a disk file and a log file, which makes it relatively easy for the administrator to back them up.

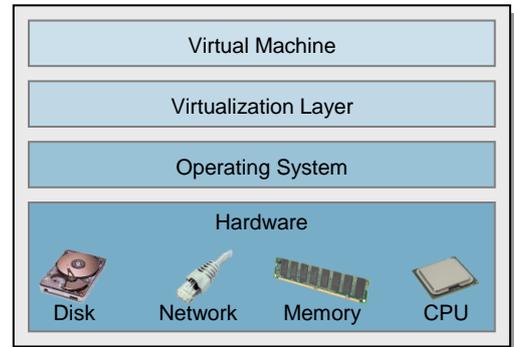


Typical virtualization architectures

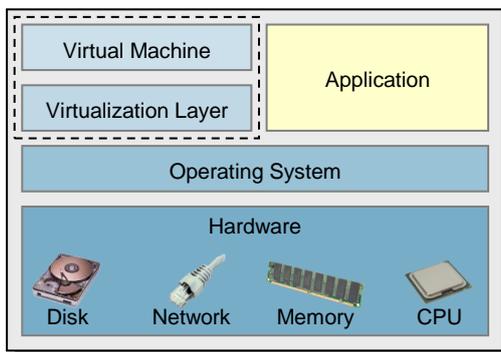
In principle, there are four types of virtualization techniques, which enable one or more virtual servers to be run on a shared, physical hardware platform.

Type 1:

In the first variant an operating system is installed on a physical server, upon which a virtualization program is positioned. Consequently, a virtualization layer is added to the operating system, on top of which the virtual machine is positioned. Every CPU, disk, memory or network access must pass through this layer. The diagram on the right shows the basic structure.



The advantage of this type of virtualization is that in addition to the virtual server other applications can be run on the host operating system, as shown in the figure on the left. In this way, applications can be



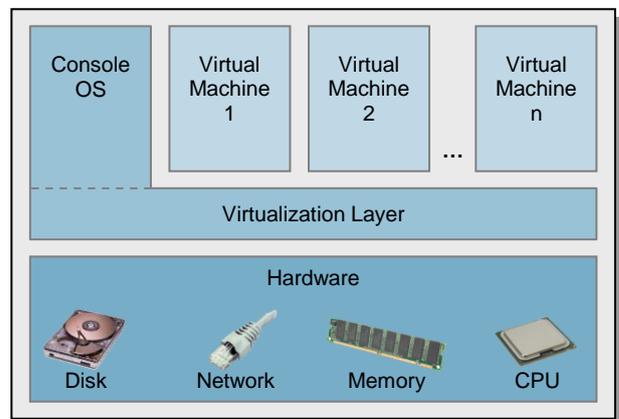
run on a physical server in parallel with the virtual server. The disadvantage of such a virtualization solution is the actual overhead of the host operating system. The host server runs system services, which need resources that are only required to operate the applications, and not to operate the virtual machines. The performance of the virtual machines is consequently reduced.

All hardware accesses to and from the virtual machine must pass through the virtualization layer and the host operating system.

Type 2:

The second variant actually does not use a host operating system, as is the case with virtualization type 1, but already implements all the functions directly required for virtualization in the virtualization layer. Since this also includes control of the I/O devices, the virtualization layer must also implement the necessary drivers or at least provide a general interface, via which third-party drivers can be integrated.

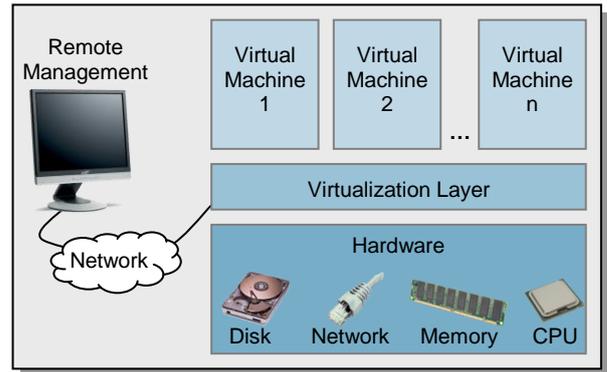
A special auxiliary operating system ("Console OS"), the scope of which has been reduced to the functionality genuinely required, is used for the administration of the system. The auxiliary operating system itself can already be seen as a VM with a special status, thus it can have e.g. very direct access to the I/O devices.



The advantage of such a virtualization solution is that it is not burdened with the overhead of a host operating system, and hardware access to and from the virtual machine only has to pass through one layer, the virtualization layer. For example, VMware vSphere 4 is based on this concept and uses a specially adapted Linux version both as the basis for the virtualization layer and for the auxiliary operating system.

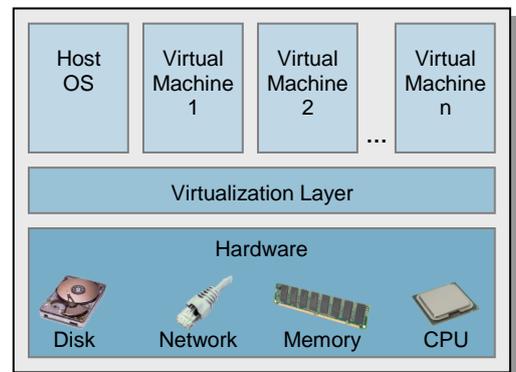
Type 3:

The third variant is the most straightforward implementation. In principle it is very similar to the second variant, in that it also provides all the functions directly required for virtualization, including I/O control, itself. In contrast to the second variant, however, merely administration interfaces and no administration functions/programs are provided directly on the system. As a result, the console operating system is no longer applicable, as is the resource consumption it causes. The administration functionalities also required with this variant must be provided by an external system. Virtualization solutions that are based on this principle can be so compact that they can in fact already be provided by the firmware of a system. One example worth mentioning for such a solution would be VMware ESXi.



Type 4:

The fourth variant refers to both variant 1 and variant 2. As with variant 1, a fully fledged host operating system is used to perform all hardware accesses. However, this host operating system already runs under the supervision of the hypervisor (refer to variant 2), but compared with conventional VMs it has the privilege of being able to access the hardware directly. The I/O operations of the conventional VMs are not performed by the hypervisor, but by this host operating system. The advantage of this approach can be seen in the fact that the hypervisor can not only be implemented in a compact way here, but has, as in the second variant, full control over the host system and can in this way - at least with regard to CPU and memory resources - avoid the overhead caused by a regular host operating system. However, with regard to I/O activities the overhead not only continues to exist, but can - depending on the architecture of the host operating system - be even higher than in variant 1. In order to avoid this overhead hypervisors, such as XEN, provide the possibility for individual HW controllers, such as a PCI card, to be allocated to an individual VM on a dedicated basis. Consequently, very fast access from the VM is possible, but this is paid for with the disadvantage that only one VM can access this HW at the same time. Moreover, HW independence is no longer given, therefore such a VM can no longer easily "move" to another HW platform.

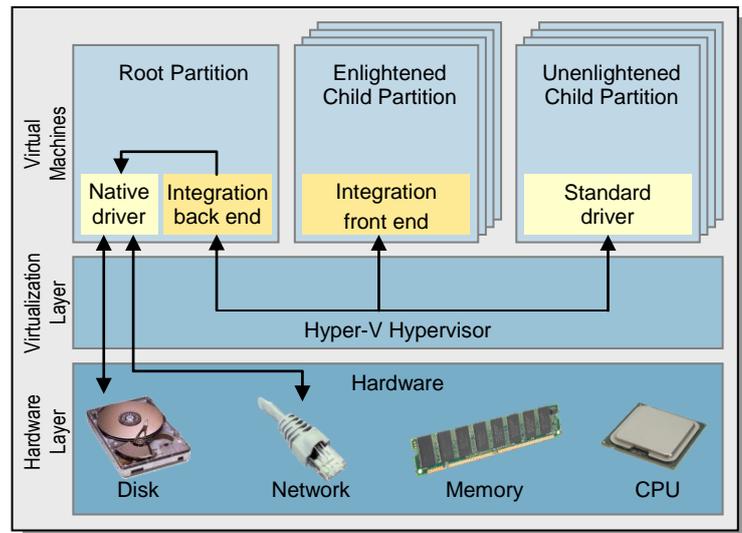


Since the host operating system analog to variant 1 can be a fully fledged operating system, it is in principle here also possible to provide further server services in addition to the actual virtualization functionality. XEN and Hyper-V are examples of this type.

General overview of Hyper-V

Hyper-V is virtualization software that has become a component part of most Windows Server 2008 x64 editions and is provided in one form as an additional "role". However, Microsoft also offers a special server version, which is solely designed to be used as a Hyper-V server. Here we are dealing with a reduced Server Core version, that is to say only the services that are absolutely necessary to run a Hyper-V server are available. Further-reaching services, such as a GUI, are not included. Such a server is at least very similar to a virtualization solution of type 1, like for example the VMware Server. However, Hyper-V is a virtualization solution of type 4 and therefore more likely to be compared with XEN. The decisive difference to a virtualization solution of type 1 can be seen in the fact that Hyper-V already virtualizes the host operating system.

However, compared with a regular VM, the VM of the host operating system has a privileged status, which allows it e.g. to access the hardware directly. This is necessary because the privileged VM is used by the conventional VMs among other things as a kind of "proxy" for their I/O operations. Depending on the VM type, the device drivers in the VMs are denoted in the very simplified diagram as "Integration front end" and as "Standard driver". The I/O requests are sent by the device drivers to the components of the "Integration back end" in the privileged root partition, which pass the requests to its own native drivers.



All VMs are denoted as "partitions" in Hyper-V. The privileged status of the VM of the host operating system is also expressed in the naming convention for this VM. The latter defines "root partition" as the name for the VM of the host operating system, whereas all other VMs are denoted with the generic term "child partition". Since practically each virtualization solution defines its own terms, the far more conventional abbreviation "VM" is used in this document instead of "child partition". On the other hand, the name "root partition" is also used below, because none of the other virtualization solutions stated here has an appropriate equivalent.

In comparison with other virtualization solutions, the functional scope of the actual Hyper-V hypervisor is restricted to controlling CPU and memory resources as well as handling asynchronous events (e.g. interrupts). Moreover, it controls the scheduling of the VMs, without influencing the scheduling within the VMs. More detailed information about Hyper-V, such as a more accurate architecture diagram, is available in the ["Hyper-V Guide"](#) [L14].

Virtualization forms

Due to the history of their development in the pre-virtualization era the classic x86/x64 processors have various characteristics (particularly in the privileges for CPU instructions and in memory management), which make virtualization considerably more difficult. Today's virtualization solutions have to avoid these deficits in a complex way with software. The two virtualization concepts below are generally used within the framework of Hyper-V.

Para-virtualization is a form of virtualization, in which the operating system of the VM "knows" that it is virtualized. It supports virtualization by only using the CPU in such a way that it can be virtualized without any problems. This calls for modifications in the operating system kernel and in the device drivers. Virtualization support through the processor ([Intel-VT](#) [L4] / [AMD-V](#) [L5]) is not necessary in this form of virtualization and does not generate any advantages. Para-virtualization is at present the form of virtualization which has by far the lowest overhead and thus also the smallest losses in performance compared with a native operating system. The "root partition" in Hyper-V is accordingly based on this form of virtualization, because it has to perform all the I/O operations in an acting capacity for the "child partitions". Hyper-V offers so-called "integration services" for Windows Server 2008, Windows Vista and SUSE Enterprise Linux 10, which enable these operating systems to run in a VM in a para-virtualized way.

With so-called **full virtualization** the OS of the VM can remain unchanged. The problem mentioned at the outset is resolved by newer x86/x64 processors with extensions ([Intel-VT](#) [L4] / [AMD-V](#) [L5]), which can be used by the hypervisor. Since no OS modifications are necessary with this form of virtualization, non-

adapted operating systems can be virtualized with it. The primary disadvantage of full virtualization can be seen in the fact that it entails on the one hand greater losses in performance than para-virtualization and that the implementation is very complex despite the innovative processor support. This is among other things based on the fact that with classic full virtualization the most important hardware components of a typical computer have to be elaborately emulated in software. Generally, losses in performance cannot be avoided, but can mostly be minimized through additional "integration services". As a result of this add-on, special in principle para-virtualized disk and LAN drivers are also made available to a fully virtualized VM in order to at least achieve a similar performance for low-level activities, as in a para-virtualized VM. Such "integration services" are made available as part of Hyper-V for example for Windows Server 2003.

In the Hyper-V documentation Microsoft does not distinguish between the two named forms of virtualization. Instead, a difference is merely made between "**Enlightened Child Partitions**" and "**Unenlightened Child Partitions**". "Unenlightened child partitions" are VMs, for which neither "integration services" are provided by Microsoft nor in which "integration services" have been installed. Due to the missing "integration services" such VMs are always fully virtualized. Even if the "integration services" have been installed, a VM need not be para-virtualized in the classic sense. In the event of a Windows Server 2003 VM the "integration services" merely include para-virtualized device drivers and no extensions for the original Windows OS kernel. Although all I/O operations are accordingly performed at optimal speed in such a VM, functions such as memory handling still have to be emulated by the hypervisor. In this connection, the term "Driver enlightened" is also used in the Internet. For VMs that are based on Windows Server 2008, Windows Vista or SUSE Enterprise Linux the "integration services" also contain - in addition to device drivers - kernel extensions, which put the OS kernel in a position to communicate with the hypervisor directly so as to avoid the complex emulation of individual functions. In principle, this corresponds to classic para-virtualization, but with one important difference - namely under Hyper-V a para-virtualized VM can still have individual functions emulated by the hypervisor. Thus, for example a virtual network card configured as a "Legacy Network Adapter" is still based on a hardware emulation.

I/O Structure

A detailed description of the way in which I/O operations run under Hyper-V would fall outside the scope of this document. Therefore, the basic process is only roughly explained here.

The fundamental I/O concept of Hyper-V is based on the "root partition" running the I/O operations in an acting capacity for the conventional VMs that initiated the operations. Current literature denotes the SW components that control this procedure as "Virtual Service Client VSC" (initiating VMs) and "Virtual Service Provider VSP" (root partition). Regardless of how quickly actual communication is handled, with this concept there will always be the problem that two interdependent instances here can really only work in parallel in optimal situations - even on multiprocessor systems. Therefore, in addition to the pure communication costs, latency times arise which result from the scheduling of the two instances. A virtualization solution without this additional interim instance for the path of the I/O requests (see type 1 or 2) could work in an undoubtedly more straightforward way in this regard, because it would always be possible here to trigger the execution of an I/O request immediately after its receipt. Prerequisite to this is that the used host operating system provides an asynchronously working I/O API.

In case of a fully virtualized VM without the "integration services" I/O handling is on account of the necessary device emulation considerably more elaborate. Since appropriate "integration services" are available for nearly all guest operating systems that are officially supported by Hyper-V, we shall not go into the detail of such configurations here. More information about this is available in [Hyper-V: Guest OS Support](#) [L15].

Hyper-V on the basis of Windows Server 2008 R2 Enterprise (full installation)

As a matter of principle, a Hyper-V server can be installed in three ways:

- either based on a regular Microsoft Windows Server 2008 R2 edition as a "full installation"
- or as a "server core installation"
- or based on the special "Microsoft Hyper-V Server 2008 R2" version.

From a functional viewpoint, the latter is largely equivalent to a "server core installation". In all forms of installation the actual Hyper-V functionality is installed as a so-called "role". The essential advantage of a "server core installation" can be seen in it also being less prone to security problems due to the fact that its scope is reduced to a minimum. This is offset by the need for an additional system for the administration of the Hyper-V server. As regards performance, a "server core installation" at least offers no notable advantages compared with a "full installation" if no further "roles" and "features" are installed other than the Hyper-V role. All the measurements listed in this document are based on a "full installation" that has been reduced to the minimum. As the relevant difference between the installation types is only the existence of a graphic user interface, no measureable performance differences are to be expected between them.

What's new in Windows Server 2008 R2

Windows Server 2008 R2 includes the following relevant innovations as regards Hyper-V:

- Accelerated memory handling.
This takes place as a result of Hyper-V now utilizing the "Extended Page Tables" feature (EPT for Intel and NPT for AMD) of the current x86 processors. This feature provides support when mapping between the physical memory and the virtual memory that the hypervisor makes available to the VMs.
- Improvements in network performance for VMs through:
 - Support of the "Virtual Machine Queues" (VMQ).
Here the hypervisor makes use of the new LAN controller feature VMQ, which pre-filters the packets from an external network in a VM-specific queue in order to send them directly to the VMs. This relieves the hypervisor from the task of routing the packets to the VMs.
 - Jumbo frames
- Up to 64 logical processors on the host
- Improved energy efficiency through redesign of the "Balanced" power plan
- Easier installation of SUSE Linux Enterprise Server (SLES) in VMs (standard kernel suffices compared with the XEN-enabled kernel that was previously required)
- Red Hat Enterprise Linux (RHEL) in VMs possible

Measurement methods

This section is to present the measurement methods for performance analysis, the measuring tools used and the measurement environments. The respective configuration of the server and storage hardware used, as well as the configuration of the native and virtual operating systems are described specific to the measurement method.

Benchmarks

Since there is no universal tool for the analysis of a complex entity such as a virtual server, various benchmark tools are used depending on the purpose.

Iometer

[Iometer](#) [L7] is an Open-Source measuring tool that is excellently suited for the generation of disk and network load on a rather lower system level. Version 2006.07.27 is used. In the Windows environment the original download compilation is used, in the Linux environment on the other hand a modified version is used. This is due to an error in the query to determine if errors have occurred in the program run and which frequently prevents a correct end to a measurement run, especially with multiprocessor configurations. The modifications affect the file *IOCompletionQ.cpp*; where line 308 must be changed as follows:

```
if ((cqid->element_list[j].error == 32) || (cqid->element_list[j].error == 104) || (DWORD) * bytes_transferred < (DWORD) 0) {
```

This is only a "quick and dirty" solution which has proved to be adequate for regular measurement operations.

Netperf

[Netperf](#) [L6] is an Open-Source measuring tool that is excellently suited for the generation of network load on a rather lower system level. Version 2.4.2 is used.

vServCon

For measuring server consolidation in virtual environments Fujitsu Technology Solutions has defined the benchmark "vServCon" [L9], which is based on "[vConsolidate](#)" [L8] from Intel. "vServCon" comprises several standard benchmarks. Each of the standard benchmarks is allocated to a dedicated virtual machine (VM). These VMs, supplemented by an idle VM, then form a "tile". Depending on the performance capability of the underlying server hardware, you may as part of a measurement also have to start several identical tiles in parallel in order to achieve a maximum load. A detailed description of this environment can be found in the document "[vServCon - Benchmark Overview](#)" [L9]. A separate vServCon load profile was used for Hyper-V, which is why the vServCon scores can only be compared within the series of measurements of this document.

SPECjbb2005

The [SPECjbb2005](#) benchmark [L10] is a JAVA-based benchmark and measures the performance of server-side Java through the emulation of a 3-tier client/server system with the focus placed on the middle tier. See the document "[SPECjbb2005 - Benchmark Overview](#)" [L11] for a detailed description of this benchmark. For use as part of the benchmark framework vServCon the benchmark was modified according to the vServCon specifications. This includes cyclical sleep pauses, as otherwise the VM would on account of the absolute lack of I/O accesses with this benchmark use its full CPU time as allocated by the hypervisor and would thus show a rather untypical load profile for server applications in a VM. Therefore, no compatible result is generated, but merely a single indicator that provides information about the number of transactions made.

The benchmark runs directly on the system and does not need any external load generators.

SysBench

[Sysbench](#) [L12] is an "Open Source" benchmark for databases that is available for a large number of different target platforms.

SysBench is used as part of the vServCon framework. A version modified by Intel is also used here, and is based on Sysbench version 0.3.3.

The benchmark runs directly on the system and does not need any load generators.

WebBench

The benchmark [WebBench 5.0](#) [L13] is a benchmark used to determine the performance of a web server in a client/server environment. Client PCs are used here to simulate web browsers, which send requests to the web server and log performance-relevant access information after receipt of the data.

Measurement environment

Depending on the benchmark used it is necessary to define a suitable VM. When a "core" is mentioned below, this always means in the context of VMs a "virtual CPU" (vCPU).

When defining the VMs for the application scenarios an attempt was made to remain as close as possible to the vServCon standard (FTS Profile V1.0 in "[vServCon - Benchmark Overview](#)" [L9]). As far as the operating systems and number of cores are concerned, the prerequisites in Hyper-V were taken into account. The new "Integration Services" from Windows Server 2008 R2 were installed in the VMs.

Iometer

Number of CPUs	1 core
Available RAM	1536 MB
Disk subsystem	FibreCAT CX500 RAID 0 made up of five 36 GB hard disks with 15,000 rpm
Operating systems	Microsoft Windows Server 2003 R2 Enterprise x64 Edition (SP2) and Microsoft Windows Server 2008 Enterprise x64 Edition

Netperf

Number of CPUs	1 core
Available RAM	1536 MB
Disk subsystem	FibreCAT CX500 RAID 0 made up of five 36 GB hard disks with 15,000 rpm
Operating systems	Microsoft Windows Server 2003 R2 Enterprise x64 Edition (SP2) and Microsoft Windows Server 2008 Enterprise x64 Edition

SPECjbb

Number of CPUs	2 cores
Available RAM	2 GB
Disk subsystem	FibreCAT CX500 RAID 0 made up of five 36 GB hard disks with 15,000 rpm
Operating system	Microsoft Windows Server 2008 Enterprise x64 Edition
Application	BEA JRockit R27.2.0

SysBench

Number of CPUs	2 cores
Available RAM	1536 MB
Disk subsystem	FibreCAT CX500 RAID 0 made up of five 36 GB hard disks with 15,000 rpm
Operating system	Microsoft Windows Server 2003 R2 Enterprise x64 Edition (SP2)
Application	Microsoft SQL Server 2005

WebBench

Number of CPUs	1 core
Available RAM	1536 MB
Disk subsystem	FibreCAT CX500 RAID 0 made up of five 36 GB hard disks with 15,000 rpm
Operating system	SLES10 SP2 64-bit
Application	Apache 2 (SLES10)

Idle

Number of CPUs	1 core
Available RAM	400 MB
Disk subsystem	FibreCAT CX500 RAID 0 made up of five 36 GB hard disks with 15,000 rpm
Operating system	Microsoft Windows Server 2003 R2 EE SP2 32-bit

The following systems of the current and previous server generation are the basis for the measurements:

SUT Hardware: Server with Xeon 5500 series	
Model	PRIMERGY RX300 S5
Processor	2 x Xeon E5540 (2.53 GHz, 8 MB L3 cache)
Memory	48 GB (a PC3-8500R each, 8 GB, in DIMM-1A until DIMM-1F)
Network interface	2 x Dual Port 1 Gb LAN controllers onboard (Intel 82575EB); one for load, one for control. SW components to be installed subsequently in order to run VMQ: 1) LAN driver for Intel 82575EB (Zoar) onboard, version 11.0.103 (W2Kx x64) 2) Intel LAN Mgmt-Appl PROset from SW-Kit ≥ 14.3 (W2K8 R2 x64) In the settings there are the following differences from the default under "Network Connection Properties > Advanced Adapter Settings": Virtual Machine Queues = Enabled Receive Side Scaling Queues = 4 Queues
Disk subsystem	FibreCAT CX500. Boot LUN, in addition for each VM or each tile a dedicated 50 GB LUN configured as a RAID 0 array consisting of 6 Seagate ST373454 disks each (15 krpm, 3.5")
Storage connection	FC-Controller QLogic QLE2460 in slot 1
SUT Software: Server with Xeon 5500 series	
Operating system	Windows Server 2008 R2 Enterprise x64 (WS08 R2) and Windows Server 2008 Enterprise x64 (WS08). With WS08 R2 following deviations from default: Unless otherwise described, the following setting is made: "Control Panel > Power Options > Preferred plans" to "High performance"
OS updates	With WS08 R2: - With WS08: SP2
BIOS	6.00 R1.09.2619 Default settings

SUT Hardware: Server with Xeon 5400 series	
Model	PRIMERGY RX300 S4
Processor	2 × Xeon E5420 (2.50 GHz / 2 × 6M L2 / 1333)
Memory	32 GB (4 × 8 GB PC2-5300F)
Network interface	2 × 1-Gbit LAN Broadcom (onboard); one for load, one for control
Disk subsystem	FibreCAT CX500. Boot LUN, in addition for each VM or each tile a dedicated 50 GB LUN configured as a RAID 0 array consisting of 6 Seagate ST373454 disks each (15 krpm, 3.5")
Storage connection	FC-Controller QLogic QLE2460 in slot 2
SUT Software: Server with Xeon 5400 series	
Operating system	Windows Server 2008 R2 Enterprise x64
OS updates	-
BIOS	PhoenixBIOS Version 4.06 Rev. 1.04.2519 Following deviations from default: Speed-Step: disabled

Load Generator Hardware	
Model	2 server blades BX620 S4 per tile in PRIMERGY BX600 S3 Chassis
Processor	2 × Xeon 5130, 2000 MHz
Memory	1 – 2 GB
Network interface	2 × 1 Gbit LAN each
Operating system	W2K3 EE

Performance Analyses

Every access to and from the virtual machine must pass through the virtualization layer. The task of this layer is to interpret and convert the I/O operations. All processor, memory, disk and network accesses are from the host's viewpoint converted for the virtual machine. Conversely, all accesses from the virtual machine to the host must also be converted.

This "transformation" costs computing performance and thus also computing time. The following performance analyses are intended to provide information about the extent to which this virtualization under Hyper-V affects the performance of the virtual machines.

The main focus of the analyses is on the one hand the comparison of Windows Server 2008 R2 Hyper-V and Windows Server 2008 Hyper-V on a PRIMERGY server with the Xeon 5500 processor generation, and on the other hand a comparison of the Xeon 5500 (e.g. in PRIMERGY RX300 S5) and Xeon 5400 (e.g. in PRIMERGY RX300 S4) processor generations as hardware platforms for Windows Server 2008 R2 Hyper-V.

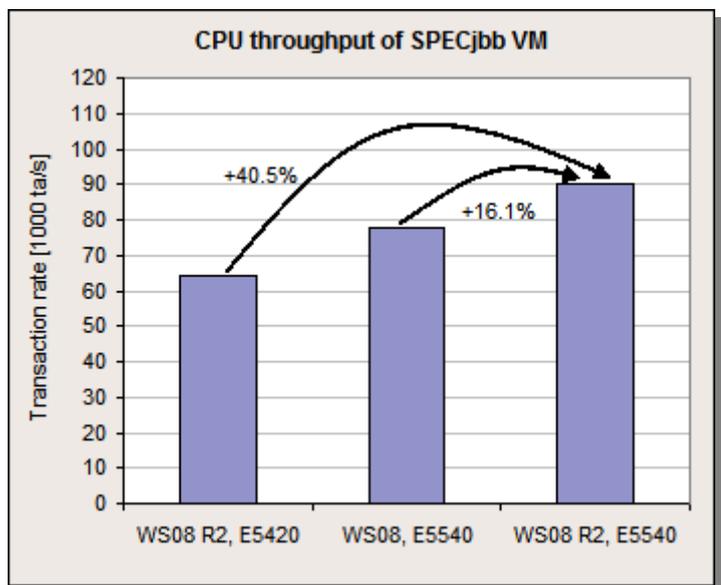
Below the four performance-relevant components



are first looked at individually, although CPU and memory are so interwoven that they are considered jointly.

CPU and Memory

The diagram opposite depicts the performance of a SPECjbb VM on various virtualization platforms. The Hyper-V versions in Windows Server 2008 R2 (WS08 R2, E5540) and in Windows Server 2008 (WS08, E5540) are compared for the server with the Xeon 5500 processor series. Added to this is the newer Hyper-V version on the server with the Xeon 5400 processor series (WS08 R2, E5420). The growth in performance through the newer hypervisor version is already evident, as can be seen in the 16.1% growth of the transaction rate when changing from "WS08, E5540" to "WS08 R2, E5540". The impact of the improved memory handling (use of the processor feature EPT by the hypervisor) can be seen here. However, the increase in performance achieved is significantly more noticeable if you compare the newer Hyper-V version on the system of the previous generation with the current system. As a result of this pure change of hardware (from "WS08 R2, E5420" to "WS08 R2, E5540") the performance increases by 40.5%. Here we can see the effectiveness of the improved processor architecture, which also clearly provides better support for virtualization regardless of the improvements in the hypervisor.



For the purpose of better comparability the cyclical sleep pauses, which are integrated in the SPECjbb VM according to the vServCon load profile, were deactivated for these measurements.

Disk I/O

Under Hyper-V a virtual hard disk (HD) can from the view of a VM be provided in two different forms: either as an "IDE-HD" or as an "SCSI-HD". On the Hyper-V side only a HD, connected via a virtual IDE controller, is possible as a boot HD for a VM. However, IDE controllers have the key disadvantage that they permit a maximum of four devices. Such a restriction would probably be acceptable for a desktop VM, but not for many server VMs. Therefore, Hyper-V also provides a purely para-virtualized SCSI controller, which can address up to 64 virtual "SCSI-HDs". Up to four SCSI controllers can be configured per VM, and accordingly a single VM can have a maximum of 256 virtual "SCSI-HDs". Attributes such as high reliability and high performance are usually associated with SCSI-HDs, whereas IDE-HDs are generally regarded as less reliable and above all as considerably slower. Such prejudices are not appropriate for a virtual "IDE-HD"; compared with a virtual "SCSI-HD" the performance is completely identical. However, the installation of the "integration services" with the para-virtualized IDE device drivers in the VM is prerequisite to this.

At hypervisor level it is possible for a virtual HD to either be provided as a file (file VHD) or by a native HD or SAN-LUN ("pass-through disk" in Hyper-V, in other sources often referred to as a "raw device" or "raw disk"). With a file the required space can either be reserved on a once-only basis or it is requested dynamically. The latter has the advantage that such VHDs can be created very quickly. However, this is offset by the disadvantage that the VHD file is later inevitably subjected to fragmenting and therefore considerable losses in performance must be expected. Thus, a VHD with a dynamic extension is not to be recommended for high-performance productive operations.

Measurement environment

All the tests concerning disk I/O are based on a VM and a native system of the following configuration:

Number of CPUs	1 core
Available RAM	1536 MB
Operating systems	Microsoft Windows Server 2008 Enterprise x64 Edition (VM) and Windows Server 2008 R2 Enterprise x64 Edition (native)

Windows is used as the operating system, because both the operating system and the measuring tool Iometer under Windows are known to support asynchronous disk I/O. Problems of the virtualization layer or the host with asynchronism would be provable in this way. The VM measurements take both the case of a virtual "IDE-HD" and a virtual "SCSI-HD" into account. Since no relevant differences in performance can be seen, this section only looks at the results of the "IDE" measurements more closely. However, all the statements also apply accordingly for virtual "SCSI-HDs". With file VHDs it is assumed for performance reasons that the VHD file in the file system is not fragmented.

To determine the data throughput we will consider the following access patterns (in Iometer: "Access specification"):

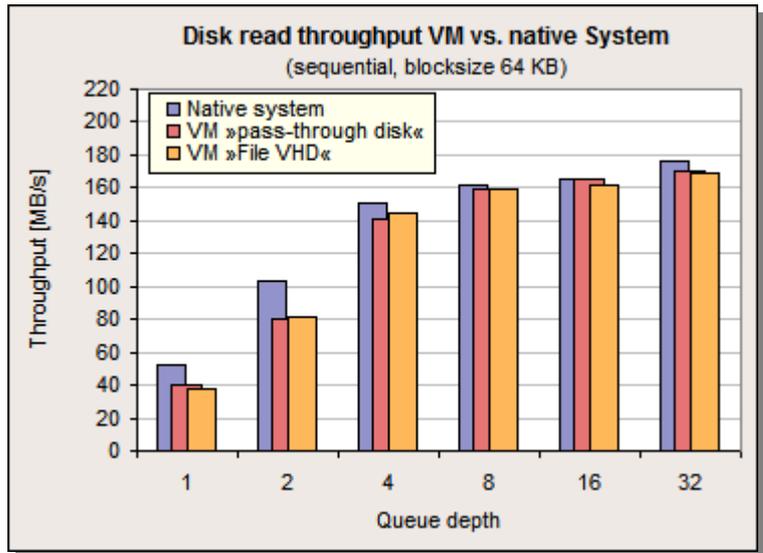
Access pattern	Block sizes	Number of Read/Write	Random share	Queue depths
Sequential Read/Write	64 kB	100% Write; then 100% Read	0%	1,2,4,8,16,32
Database	8 kB	67% Read, 33% Write	100%	1,2,4,8,16,32
File server	64 kB	67% Read, 33% Write	100%	1,2,4,8,16,32

Data throughputs

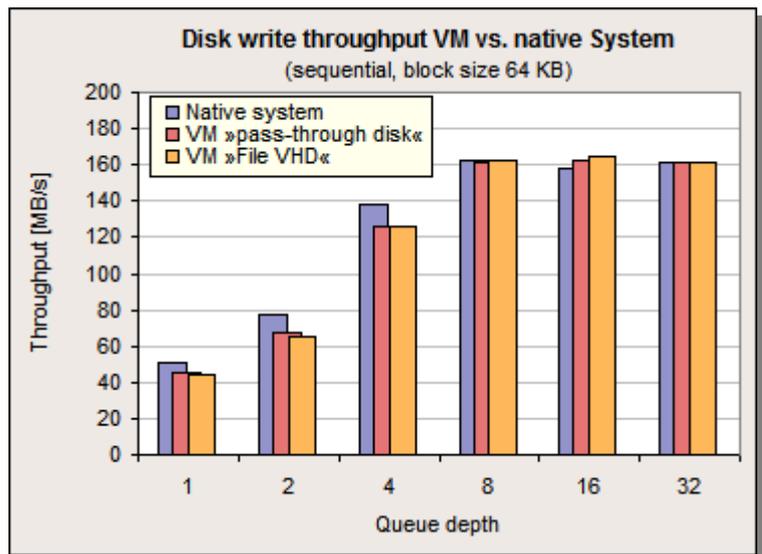
It had already become clear with Windows Server 2008 Hyper-V (WS08) that there are no relevant differences in the disk-I/O throughput rates if you compare Windows Server 2008 and Windows Server 2003, namely neither on a native nor on a virtual basis. Moreover, the version at that time also showed that there are no notable differences between the connection of a VHD as an "IDE-HD" and as an "SCSI-HD" for accesses from VMs. Thus the performance of the disk I/O is discussed below as an example based on the operating system Windows Server 2008. A virtual "IDE-HD" is taken as the basis for the measurements with VMs.

Access pattern: Sequential Read/Write

With sequential read the VM is - independent of whether your VHD was provided as a "pass-through disk" or a file VHD - in a position to achieve the maximum data throughput given the storage system and which the native system achieves. However, prerequisite to this is the fact that work is done with at least a queue depth of about four, that is four asynchronous read requests. With a smaller queue depth the VM with a "pass-through disk", can still achieve a very good throughput of at least 77% of the native system and at least 72% with a file VHD. Considerably larger differences can be seen between the "pass-through disk" and the file VHD with Windows Server 2008 Hyper-V.

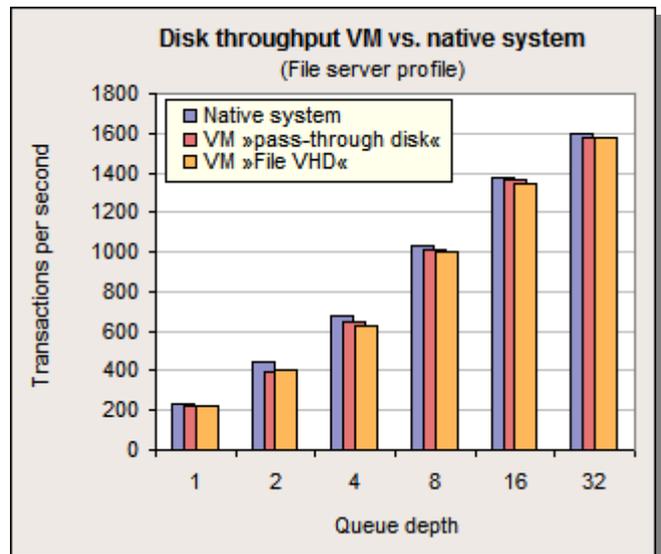
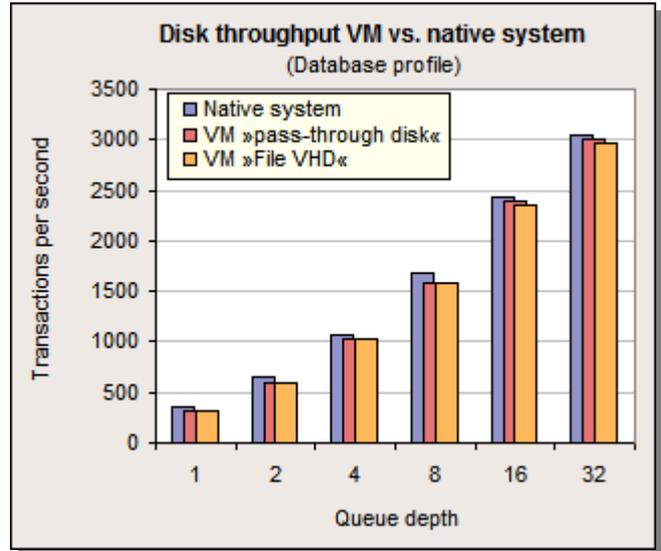


As a matter of principle, the same image as with read can also be seen with sequential write. The VM and the native system both achieve the maximum possible data throughput. The VM needs a queue depth of at least eight for this. With smaller queue depths the VM with a "pass-through disk" achieves at least 86% of the throughput of a native system, and the VM with a file VHD still achieves 84%.



Access pattern: File Server / Database Server

The VM shows optimal performance with the access patterns "File Server" and "Database Server". It profits from the fact that now the response times of the disk subsystem naturally rise as a result of the random access so that their own overhead is no longer of any consequence. In the performance of the VMs here there are no longer any significant differences compared with a native system. With these load profiles the differences between the "pass-through disk" and the file VHD are still somewhat smaller than with the previously described sequential profiles.



Database server application scenario

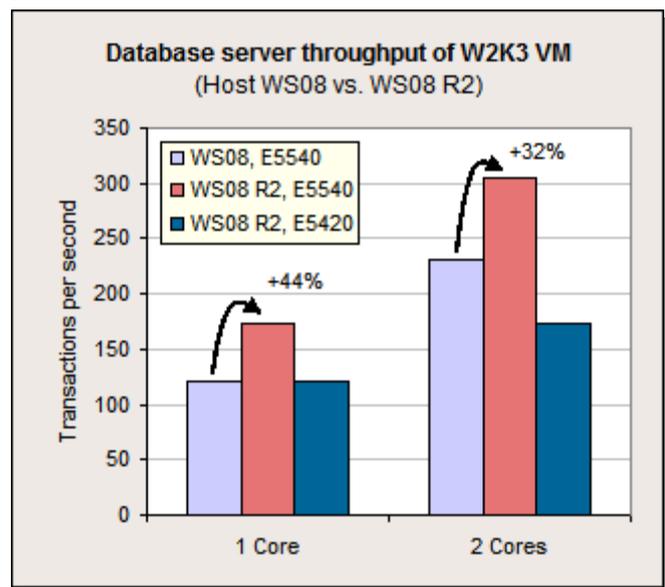
Database applications are an application scenario, in which a high disk performance is usually indispensable. In the Iometer measurements the VM shows a very good performance, and an optimal one especially with the database profile. However, the special characteristics of Iometer are that it only works in an extremely optimized way as a measuring tool and thus only creates a minimum CPU overhead. A database on the other hand is software with a complex functionality. This is why with the same intensity of disk accesses it would typically need more CPU performance than Iometer. And it is also a reason why database servers are usually designed as multiprocessor systems.

Therefore, we will consider a typical database application, as represented by the benchmark SysBench. For this purpose, the VM is configured as follows:

Number of CPUs	1 to 2 cores
Available RAM	1536 MB
VHD mode	"pass-through disk"
Operating system	Microsoft Windows Server 2003 R2 Enterprise x64 Edition (SP2)
Database	Microsoft SQL Server 2005
Benchmark	SysBench 0.3.3 ([L12])

It should be pointed out that in the definition of this VM in the vServCon standard profile the focus is placed on the field of application "Server consolidation". This is why older products are consciously represented here for the operating system and database software.

The diagram opposite compares the transaction rates achieved by this VM between Windows Server 2008 R2 Hyper-V (WS08 R2) and Windows Server 2008 Hyper-V (WS08). In both cases, the server with the Xeon 5500 series was used. In addition, the performance of the newer Hyper-V version on the server with the Xeon 5400 series is also depicted. Strong growth can be seen when a change is made from the older Hyper-V version to the newer one. This growth (44% for one core and 32% for two cores) shows very clearly that the improved virtualization support of the Xeon 5500 processor generation now in WS08 R2 can be used for the VMs. Thus, VM operating systems, such as Windows Server 2003, which have no "kernel enlightenments", can now also achieve a very high performance level. This means for the virtualization user that with WS08 R2 and a server with processors from the Xeon 5500 generation as a virtualization platform it is no longer necessary to worry about things like the kernel-mode share of an application and the number of logical cores in a VM. In the past, such considerations were still advisable.



As can be seen in the diagram, the good scaling of the application performance with the number of cores in the VM is obviously a characteristic of the processor and not of the hypervisor. This shows that this scaling on the server with the Xeon 5500 processor series is good for both Hyper-V versions: the gain in performance for Hyper-V in WS08 amounts to a substantial 92%, and it is still a good 75% for Hyper-V in WS08 R2. In contrast, the growth on the server with the Xeon 5400 processor series is only 44%.

For the case of the database VM with two cores you can also see that the improvements through the new processor are much larger than the improvements due to the new hypervisor:

- Performance increases by 76% when you use a system with Xeon E5540 processors instead of a system with Xeon E5420 processors (and an unmodified hypervisor).
- Performance increases by 32% when you exchange Windows Server 2008 Hyper-V for Windows Server 2008 R2 Hyper-V and an unmodified system with Xeon E5540 processors.

Network

After disk I/O we will now deal with the other important I/O component - the network. Maximum possible data throughput is also to be considered first here and in a further analysis the consequences are to be examined in a real application scenario.

All the tests are carried out with a VM and a native system of the following configuration:

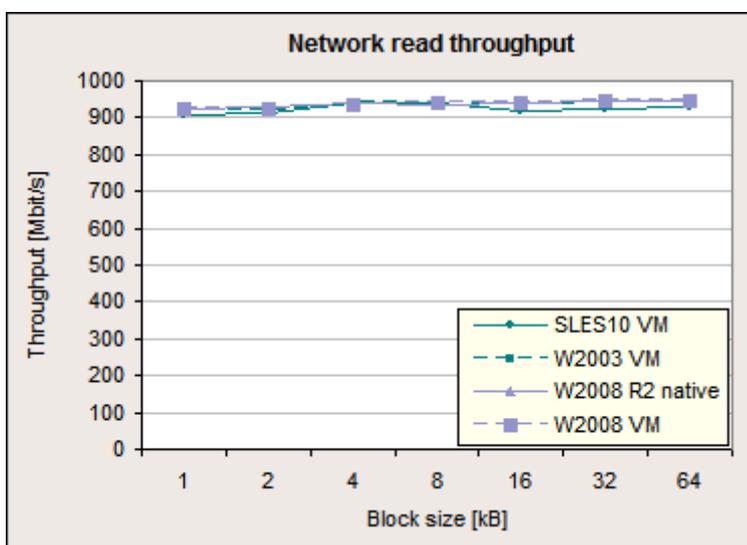
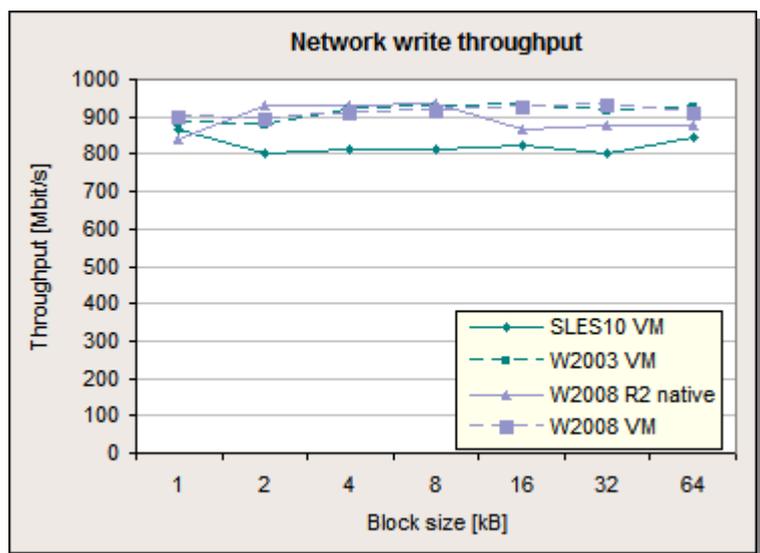
Number of CPUs	1 core
Available RAM	1536 MB
Operating systems	Microsoft Windows Server 2003 R2 Enterprise x64 Edition (SP2), Microsoft Windows Server 2008 Enterprise x64 Edition (or R2) and SLES10 SP2 64-bit

A PRIMERGY BX620 S4 server blade with two Xeon 5130 processors and Microsoft Windows Server 2003 Enterprise Edition as the operating system is used as a counterpart for the measurements.

Data throughput

Netperf is used to determine data throughput. Here a Netperf instance on the system to be measured ("System under Test", SUT) exchanges data with different block sizes with a second Netperf instance on a remote system. The first series of measurements examines the throughput a VM can achieve with a network load generation to an external counterpart. Accordingly, data are transported for this purpose via a physical network.

As with the disk, network throughput also greatly depends on the type of operation, that is to say whether data is sent or received. If you consider the write case (SUT sends), all the systems achieve good throughputs of between 800 and 940 Mbit/s close to the physical limit of 1 GbE for all the block sizes considered. The comparatively somewhat smaller throughput during send for the case of the SLES10 VM can be explained by the different TCP/IP stacks of Linux and Windows (the latter is the operating system of the counterpart). In the transfer direction under consideration the data throughput has in comparison with the previous version of Hyper-V remained on average on the same good level.



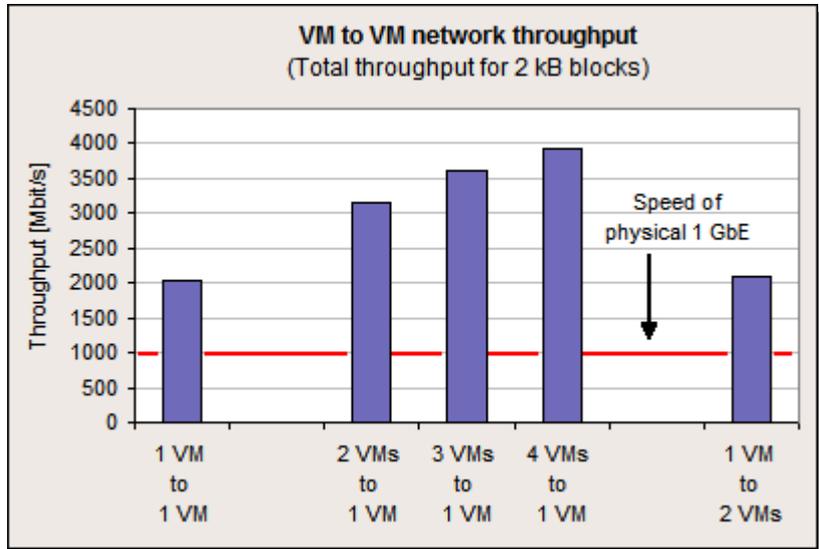
If you consider the read case (SUT receives), all the systems show very similar characteristics from a block size of 1 kB. The throughputs of the virtualized systems and the native system are very close to each other and are all above 900 Mbit/s. There has even been a slight improvement here compared with the previous version of Hyper-V.

On the whole, both the minimal and the average values of the occurring throughputs have improved visibly for all block sizes and in both directions compared with the previous Hyper-V version.

A second series of measurements examines whether a higher throughput can be achieved between two VMs on the same host than between a VM and a native system. There are two options for the configuration of the network connection in the VMs:

With a purely internal LAN, that is in a shared network of the VMs, which has no connection to a physical network, the throughput cannot be limited by the speed of the physical network. Only things such as CPU performance, speed of memory access and hypervisor architecture have an impact here.

However, even if a connection exists to an external network in the second case, the throughput should very much be determined by the parameters just mentioned because only network packets, which are sent by the VMs as a broadcast, have to be transferred to the external LAN. Network packets between the VMs are sent directly. As can be seen in the diagram opposite, the VMs actually achieve a significantly higher throughput in inter-VM traffic than during transfer via a physical network.



On the whole, the maximum possible throughput between two VMs on the same host has in comparison with the previous version of Hyper-V been further increased. As the measurements show, throughputs of up to 2037 Mbit/s can be achieved with Windows Server 2008 R2 Hyper-V (with 2 kB blocks). These only vary a little with the different block sizes.

It is also interesting that the total throughput received by a VM can as a result be raised considerably by sending from several different VMs to the same VM. The total throughput received by the latter VM can reach 3937 Mbit/s if four VMs send to the same VM. If one VM sends to various other VMs, this increase does not occur.

From an application viewpoint, it has become even more interesting with the examined Hyper-V version to virtualize VMs with strong mutual network relations on the same host.

Web server application scenario

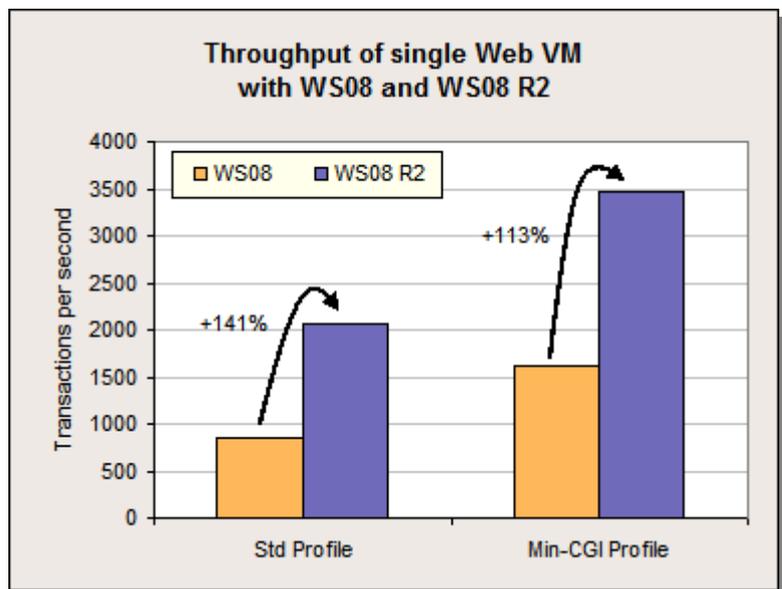
The data throughput analyses for network I/O have shown that the VMs approximately achieve the throughput of a native system. To evaluate the impact on a real application a typical web-server environment is considered below with the help of the [WebBench Benchmark](#) ([L13]). A further aspect here is the examination of the effect of the new VMQ support.

The VM as defined in the vServCon standard with SLES10 (32-bit) as the operating system was used as the VM.

The two load profiles below are considered in the following tests:

Load profiles for web servers	
STD-CGI	This defines that 16% of all HTTP requests and 2% of all HTTP-SSL requests on the web server start a CGI program. Makes great demands of a virtualization solution.
MIN-CGI	STD-CGI profile, but without the 16% CGI-HTTP requests. The load on a web server is decreased by this reduction in the number of CGI processes; but this reduces the costs within the virtualization solution a great deal more. Both effects together make so much additional CPU performance available that the web transaction rate for VMs is significantly increased.

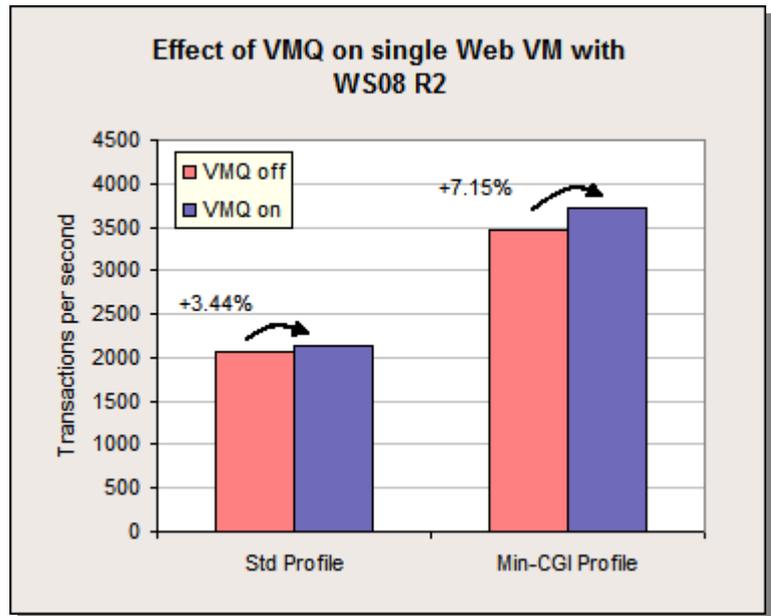
The following diagram compares Hyper-V in Windows Server 2008 R2 (WS08 R2) with the previous version (WS08) as regards the web throughput of the VMs for both load profiles. The result on the system with the Xeon 5500 series when you exchange the previous Hyper-V version for the current version is clearly more than twice the web throughput for the two load profiles. Several things are likely to be responsible for this large gain in performance: the new interface structure within the SLES10 VM (new "Integration Services", standard-SMP kernel), improved memory management (support of EPT) and network improvements (VMQ).



As already mentioned, the enabling of VMQ results in the routing of the network packets being executed by the hardware (LAN controller) instead of by the software (hypervisor). This causes an acceleration of the network transactions, which is depicted in the diagram opposite for both web server load profiles. Furthermore, the result of this relocation of software activities to the hardware is a reduction in the CPU load.

In order to activate the "Virtual machine queues" (VMQ) feature of the LAN controller Intel 82575EB it is necessary and sufficient to subsequently install the current LAN driver and the Intel LAN management application PROset.

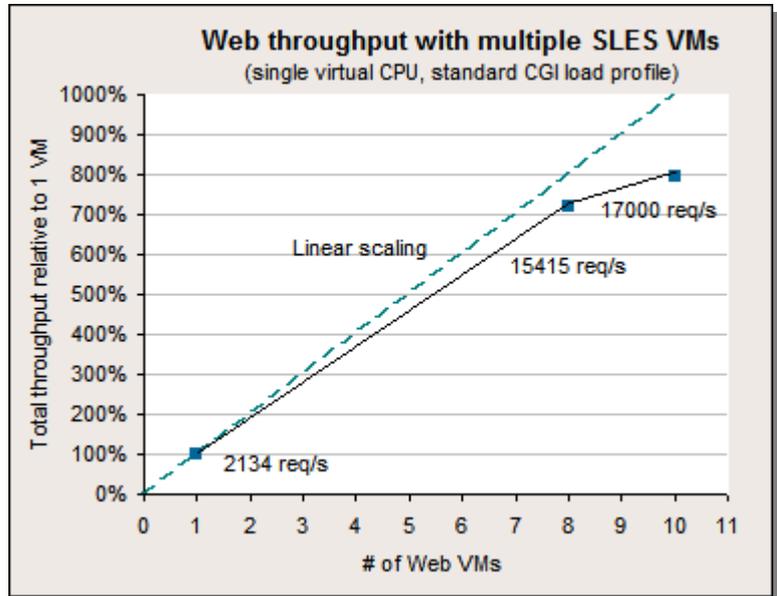
Detailed information about tuning and performance monitoring of network accesses from Hyper-V VMs is available in [Windows Server 2008 R2: High-Speed Networking Features](#) ([L16]).



Scaling of web server VMs

In the web environment load distribution to several web servers is as a typical scale-out scenario frequently possible at a relatively small cost. The following diagram shows for the STD-CGI profile that with several parallel VMs with only one CPU each and one web server per VM the web throughput can be increased very effectively in total. The diagram shows this in an exemplary way for one, eight and ten web server VMs. Due to the hyper-threading activated by default, a physical processor core is, as is widely known, divided into two logical cores so that 16 logical cores are available for the hypervisor. This high number of logical cores permits more VMs and consequently generally increases the virtualization performance of a system. This becomes evident in the diagram through the high overall throughput of ten simultaneously operated web server VMs. In the range up to eight web server VMs even every VM has a physical core exclusively at its disposal, which results in almost linear scaling of the total throughput of up to eight VMs.

With these throughputs the performance of the server is by no means at an end, because as yet unused logical cores are available. The overall performance that can be achieved in this way would never be reached by a non-virtualized web server on the same hardware. A similar trend must also be assumed with the MIN-CGI load profile. On the whole, Hyper-V exhibits very good scaling behavior with the web server VMs.

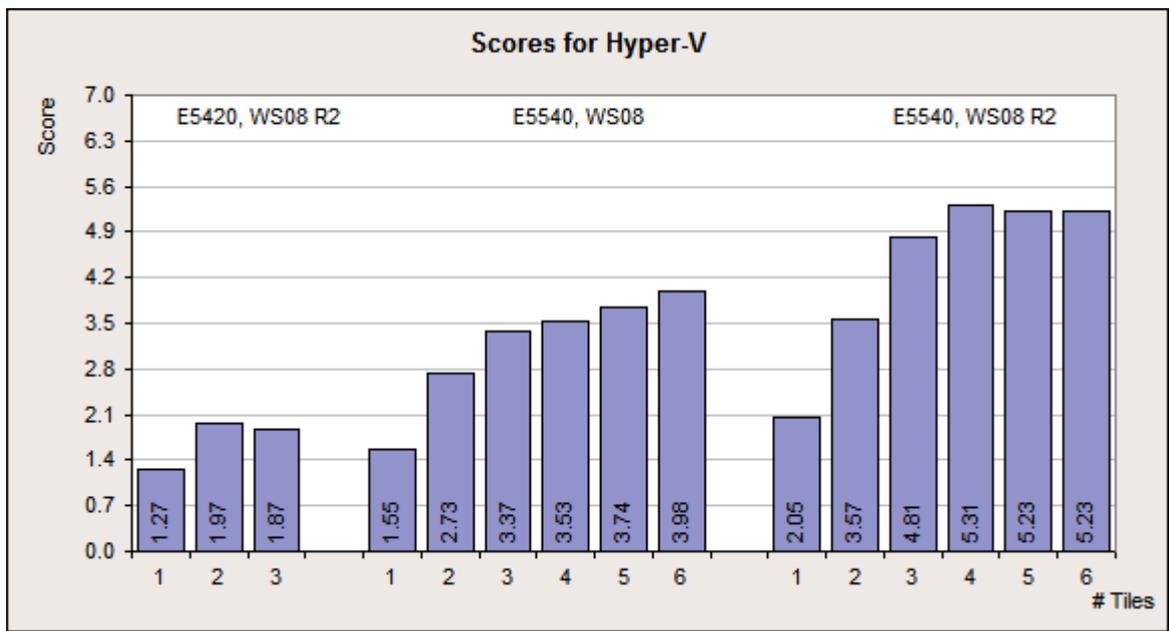


With these throughputs the performance of the server is by no means at an end, because as yet unused logical cores are available. The overall performance that can be achieved in this way would never be reached by a non-virtualized web server on the same hardware. A similar trend must also be assumed with the MIN-CGI load profile. On the whole, Hyper-V exhibits very good scaling behavior with the web server VMs.

Application mix

After we have so far considered various application scenarios in an isolated state, the performance and the scaling of a defined application mix is now to be examined. For this purpose, we assume a defined basic collection (known as a »tile«) of four different VMs: Java VM, database VM, web server VM and idle VM. It is possible to determine a performance score in a measuring run with one tile or with several instances of such a tile, operated simultaneously. More information is available about this in [»vServCon – Benchmark Overview«](#) [L9]. The configurations of the VMs included in the mix (Java VM, database VM and web server VM) are identical with those for the separate examinations of the respective application scenarios. They are described in section [Measurement environment](#). Since a separate vServCon load profile was defined for Hyper-V, the scores can only be compared within the series of measurements of this document.

The following diagram clearly illustrates the degrees to which the virtualization performance of Hyper-V is achieved on the one hand through improvements in the hardware and on the other hand in the hypervisor. For the purpose of having meaningful values two systems that are as identical as possible are compared, i. e. same number of physical cores and almost the same CPU clock speed.



In a comparison for a small number of VMs (four VMs is equivalent to a tile number of one) it make sense to first work out the effects that are not determined by the higher number of logical cores:

- The performance increases from 1.27 to 2.05 by 61% when we change from the system with the Xeon 5400 series to the system with the Xeon 5500 series, (the hypervisor in both cases is Windows Server 2008 R2 Hyper-V)
- Performance increases from 1.55 to 2.05 by 32% when we exchange the hypervisor Windows Server 2008 Hyper-V for Windows Server 2008 R2 Hyper-V (in both cases on the system with the Xeon 5500 series)

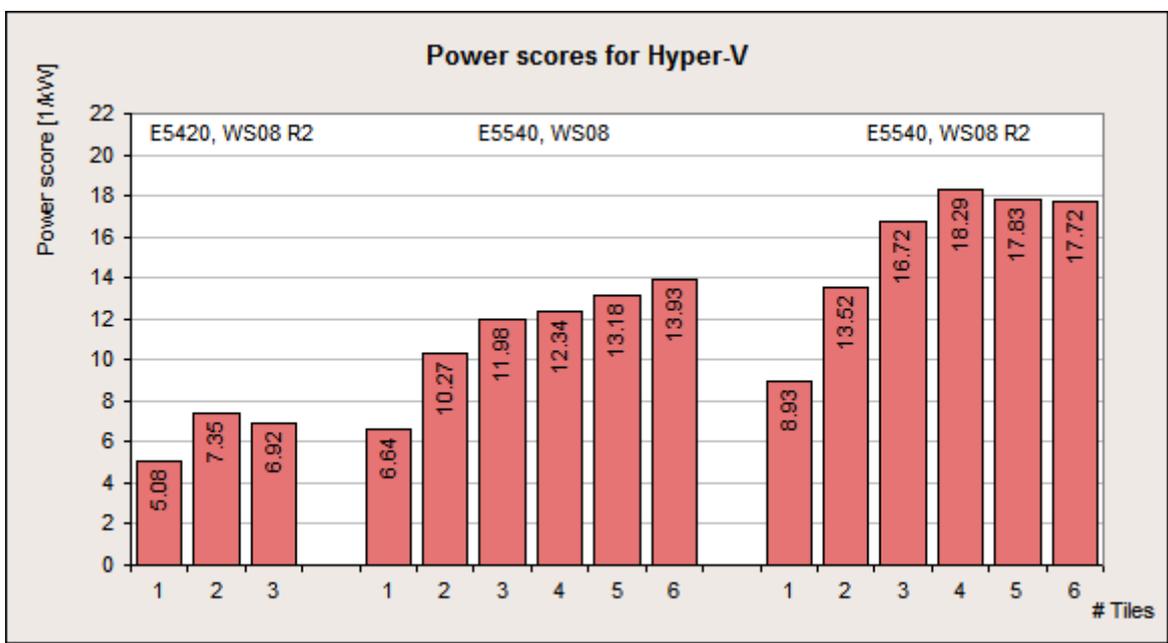
The 61% performance increase for a few VMs in the first comparison is a measure of the progress made with the processor (improved virtualization support, turbo mode), even without utilizing hyper-threading. On the other hand, the 32% growth in performance in the second comparison is a measure of the utilization of existing hardware features and the new software architecture. The impact of the new software is therefore very clear. In comparison, the effect of the new hardware is even stronger.

To also record the impacts of the higher number of logical cores through hyper-threading it is necessary to compare the appropriate performance values at full load for the optimal number of VMs:

- The performance increases from 1.97 with two tiles to 5.31 with four tiles by 170% when we change from the system with the Xeon 5400 series to the system with the Xeon 5500 series, (the hypervisor in both cases is Windows Server 2008 R2 Hyper-V)
- Performance increases from 3.98 with six tiles to 5.31 with four tiles by 33% when we exchange the hypervisor Windows Server 2008 Hyper-V for Windows Server 2008 R2 Hyper-V (in both cases on the system with the Xeon 5500 series)

When the examined server is fully loaded with the optimal number of VMs, the advance in performance due to the hypervisor exchange is just as good as in the case with a low load. It is remarkable that in this comparison with the old hypervisor the maximum score is only achieved with six tiles, but with four tiles with the new hypervisor. For mid-range numbers of tiles (three to five) that means an additional improvement in performance through the new hypervisor. Particularly impressive, however, is the effect of the new hardware, which in the case with full load is 170% due to the increased number of logical cores.

If we look at the energy efficiency of the above described constellations (score per power input of the server = power score), the outcome is also a clear superiority on the part of the new Hyper-V version, as the following diagram illustrates.



The setting options in the host operating system, which are practical depending on whether more importance is attached to high performance or to high energy savings, are dealt with in particular in the following section.

Energy-saving features

The host operating system Windows Server 2008 R2 offers (in comparison with Windows Server 2008) new features, which can reduce the power input of the server when maximum performance is not required. These features work regardless of whether the server is being used as a virtualization host or not.

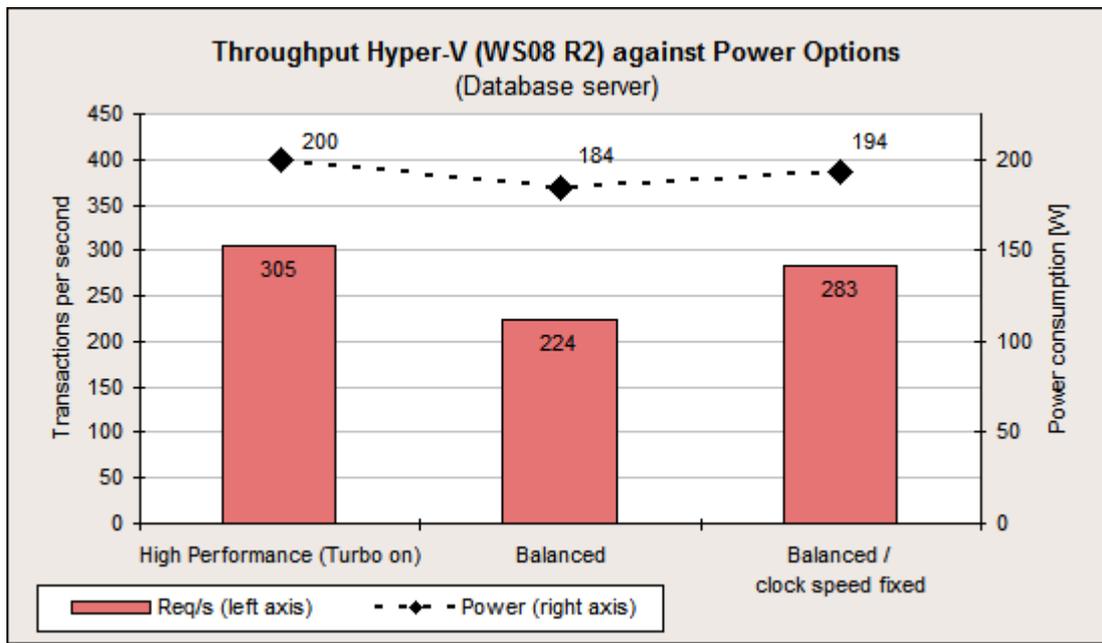
The relevant settings in the host operating system are usually made in "Control Panel > Power Options > Preferred plans". The standard setting here is "Balanced". Its aim is a balance between energy efficiency and performance, therefore it is clear that it cannot be geared to maximum performance. If the latter is the goal, it is necessary to deviate from the standard and select the alternative power plan "High Performance". This setting is taken as the basis in all other sections of this document, because its focus is the maximum performance that can be achieved with Hyper-V.

Energy savings with the power plan "Balanced" in the host operating system Windows Server 2008 R2 make use of two measures that have an impact on the server:

- The turbo mode of the Xeon 5500 processor generation, which is set as a BIOS default, is deactivated. All performance states ("P-states") of the processor, which at most have the rated speed, are available in times of lower performance requirements to reduce the clock speed. More information about turbo mode is available in [Series 5500: Turbo Mode](#) ([L17]).
- The new operating system feature "Core parking" is activated (for more information see [Windows Server 2008 R2 Power Management](#) [L18]). In the event of "Core parking" an attempt is made to consolidate the imminent CPU load on the smallest possible number of cores in order to temporarily deactivate the cores that are not needed for energy-saving purposes.

Both measures naturally work the best with a low server load.

Using the example of the database server VM, the following is intended to demonstrate in an exemplary way what impacts the two possible power plans "High Performance" and "Balanced" have on performance and power input, and what share of the overall effects the influence on the processor frequency has in particular.



The above diagram shows that the database performance increases by 36% when you switch from the "Balanced" default to "High Performance", but on the other hand the power input is also 16 W higher. If you modify the settings for "Balanced" (reducing the clock speed is permitted) in such a way that reducing the clock speed is forbidden, the database performance is only 26% higher than with "Balanced", and power input also increases by only 10 W. Thus the "High Performance" power plan proves to be an effective measure for increasing performance, but which only achieves its effect at the price of visibly increased power input.

Hence the following recommendations: If the transaction rate for the standard setting "Balanced" is sufficient for the application to be virtualized, this standard value should for the purposes of saving energy be retained.

In case of higher performance requirements for the application scenario it is firstly possible to prevent the reducing of the clock speed or to ultimately switch completely to "High Performance".

Comment:

Alternatively, it is possible to stop the reducing of the clock speed in the "Balanced" power plan as follows:

- Via the GUI of the host operating system:
By means of "Control Panel > Power Options", then "Change plan settings", and then in the „Power Options" dialog box that appears select the following: "Processor power management > Minimum processor state". Change the value set as a default under "Balanced" from 5% to 100%. The numerical value specifies the lower limit for reducing the clock speed of the processor in the form of a percentage of the processor rated speed.

- Via the command line of the host operating system ("^" hides the end of line):

```
Powercfg -setacvalueindex scheme_current sub_processor ^  
0cc5b647-c1df-4637-891a-dec35c318583 100  
Powercfg -setactive scheme_current
```

The meaning of the above cryptic argument (Power Setting GUID) of Powercfg is "Minimum processor state". The subsequent numerical value corresponds to the percentage that can also be seen via the GUI.

Summary

Microsoft has achieved a very high level in almost all areas with its revised hypervisor Windows Server 2008 R2 Hyper-V. The best impact is to be had on a current PRIMERGY server with hardware support for memory virtualization (processor feature EPT or NPT) and hyper-threading. The main improvements are:

- The pure I/O performance is now almost on the level of a native system for virtual machines (VMs) both as far as the disk and the network are concerned. In case of network accesses between different VMs of the same host the achieved transfer rate can very clearly surpass that of a physical 1-Gbit/s connection.
- VMs now scale well with the number of virtual CPUs.
- In particular the performance for the virtualization of older operating systems, such as Windows Server 2003, which were only adapted to Hyper-V through the "Integration Services", has been considerably improved.
- Compared with the previous version the high-performance consolidation of existing applications has become more uncomplicated. For example, the kernel-mode share and the number of cores required for an application are as decision-making criteria no longer as relevant as before.
- The high-performance support of SLES in application VMs is very impressive. In comparison with the previous hypervisor version its performance has increased by up to 141%. Handling has been considerably simplified by changing to the regular SMP kernel.
- Scaling for several VMs is perfect. The result of which is the opportunity to optimize the overall performance of an application by using several VMs in each case with a few virtual CPUs instead of one VM with a great many virtual CPUs. The prerequisite for this is that the application is suited for a scale-out scenario, i.e. it enables high-performance operation in a server farm.
- Compared with the predecessor hypervisor clear advances in performance of at least 30% can generally be seen in all application scenarios and load ranges.
- Increase in the possible number of logical processors on the host.

As a rule of thumb it can be deduced that an existing server (between two and three years old) can usually be consolidated on a current system with Hyper-V without any loss in throughput (if the necessary number of logical cores for the guest operating system is supported by Hyper-V).

If you have the opportunity of installing a newer operating system, such as Windows Server 2008, in the VM, you should take it. Such operating systems actively support virtualization and consequently enable even higher performance values.

Desirable improvements in future versions of Hyper-V: further extend the list of supported guest operating systems and versions and lift the existing limitations to only one or two virtual CPUs in the VMs.

Literature

[L1]	General information about products from Fujitsu Technology Solutions http://www.ts.fujitsu.com
[L2]	General information about the PRIMERGY product family http://www.primergy.com
[L3]	PRIMERGY Benchmark - Performance Reports and Sizing Guides http://www.ts.fujitsu.com/products/standard_servers/primergy_bov.html
[L4]	Intel Virtualization Technology http://www.intel.com/technology/platform-technology/virtualization/index.htm
[L5]	AMD-V http://www.amd.com/us-en/Processors/ProductInformation/0,,30_118_8796_14287,00.html
[L6]	Netperf http://www.netperf.org/netperf/
[L7]	Iometer http://www.iometer.org/
[L8]	vConsolidate http://www.intel.com/technology/iti/2006/v10i3/7-benchmarking/6-vconsolidate.htm
[L9]	vServCon - Benchmark Overview http://docs.ts.fujitsu.com/dl.aspx?id=b953d1f3-6f98-4b93-95f5-8c8ba3db4e59
[L10]	SPECjbb2005 http://www.spec.org/jbb2005
[L11]	SPECjbb2005 - Benchmark Overview http://docs.ts.fujitsu.com/dl.aspx?id=5411e8f9-8c56-4ee9-9b3b-98981ab3e820
[L12]	SysBench http://sysbench.sourceforge.net/
[L13]	WebBench http://www.lionbridge.com/lionbridge/en-us/services/outsource-testing/benchmark-software.htm
[L14]	Hyper-V Guide http://msdn.microsoft.com/en-us/library/cc768532.aspx
[L15]	Hyper-V: Support of Guest Operating Systems http://technet.microsoft.com/en-us/library/cc794868(WS.10).aspx
[L16]	Windows Server 2008 R2: High-Speed Networking Features http://download.microsoft.com/download/8/E/D/8EDE21BC-0E3B-4E14-AAEA-9E2B03917A09/HSN_Deployment_Guide.doc
[L17]	Intel: Series 5500 Turbo Mode http://software.intel.com/sites/oss/pdfs/power_mgmt_intel_arch_servers.pdf
[L18]	Windows Server 2008 R2: Power Management Features http://download.microsoft.com/download/3/0/2/3027D574-C433-412A-A8B6-5E0A75D5B237/ProcPowerMgmtWin7.docx

Contact

PRIMERGY Hardware

PRIMERGY Product Marketing

<mailto:Primergy-PM@ts.fujitsu.com>

PRIMERGY Performance and Benchmarks

<mailto:primergy.benchmark@ts.fujitsu.com>