Infor LN Deployment in a Virtualized Environment

Publication Information

Release: Infor LN

Publication date: December 6, 2012

Document code: B0073C US

Contents

About this guide ... 9

Intended audience ... 9

Terms used in this document ... 9

References ... 10

Infor ... 10

VMware vSphere 5 ... 10

Citrix XenServer 6 ... 10

Microsoft Hyper-V R2 ... 11

Contacting Infor ... 11

Chapter 1 Introduction to virtualization ... 13

What is virtualization? ... 13

How does virtualization work? ... 13

Advantages and disadvantages of virtualization ... 14

Types of virtualization ... 15

Chapter 2 Infor LN in a virtualized environment ... 17

Infor Solution License Manager ... 17

Hyper-threading technology ... 17

Infor recommendations for hyper-threading ... 17

Impact of hyper-threading on batches ... 18

NUMA systems ... 18

Affinity ... 19

Infor recommendations for NUMA systems ... 19

Example of maximum VM size on a NUMA system ... 19

Affinity ... 19

Example of using affinity on a NUMA system ... 20

Multiple VMs and over-committing resources ... 20

Example of an overcommitted server ... 21

Infor recommendations for multiple VM systems ... 21

Storage ... 22

Virtual storage types ... 22
Network settings in a 3-tier configuration ... 22
Virtual CPU sockets ... 23

Chapter 3 VMware vSphere ... 25
Infor LN benchmark results ... 25
Hyper-threading and VMware vSphere ... 26
Hyper-threaded core sharing options ... 26
Recommendations for core sharing ... 27
Benchmark note ... 27
VMware vSphere and NUMA ... 27
Single VM performance and scalability ... 28
Conclusions ... 28
Multiple VM performance and scalability ... 29
Conclusions ... 29
Configuration maximums ... 30
VMware vSphere 5 licensing model changes ... 30
VMware vMotion ... 31
Infor benchmark results ... 31
Requirements for running VMware vMotion on Infor LN ... 31
VMware ESX Fault Tolerance (FT) ... 32
Infor benchmark results ... 32
Requirements for running VMware Fault Tolerance on Infor LN ... 32

Chapter 4 Citrix XenServer ... 33
Introduction ... 33
Architecture ... 33
Managing XenServer ... 34
Configuration limits ... 34
XenServer High Availability ... 35
Performance tuning XenServer 6 ... 36
Important XenServer settings ... 36
Installation template ... 36
Install XenServer tools ... 37
Monitoring DOM0 and VM performance ... 37
Increase DOM0 memory configuration ... 37
XenServer CPU usage ... 38
Capping CPU ... 39

Setting CPU affinity (pinning) ... 39
Setting vCPU priority ... 39
CPU recommendations for Infor LN ... 40
Hyperthreading ... 40
XenServer memory usage ... 40
Memory allocation and Dynamic Memory Control ... 40
Memory recommendations for Infor LN ... 40
Storage in Citrix XenServer ... 41
Storage recommendations for Infor LN ... 41
Benchmark results ... 42
Infor LN OLTP Benchmark - Windows ... 42
Infor LN OLTP Benchmark - Linux ... 43
Single VM performance and scalability ... 44
Multiple VM ... 45

Chapter 5 Microsoft Hyper-V R2 ... 47
Hyper-V architecture ... 47
Integration components ... 47
Microsoft Hyper-V R2 versions ... 48
Live migration ... 48
Requirements for running live migration on Infor LN environments ... 48
Network settings in a 3-tier configuration ... 48

Chapter 6 Recommendations ... 49
BIOS settings ... 49
Windows settings ... 49
General VM settings ... 50
Storage ... 51
Networking ... 51
VMware vSphere ... 51
Citrix XenServer ... 52
Hyper-V ... 52

Chapter 7 Sizing information ... 53
Introduction ... 53
Sizing information ... 54
CPU overhead ... 54
Batches ... 54
Memory overhead ... 54

VMware ESX and ESXi 54
Citrix XenServer memory overhead 55
Microsoft Hyper-V R2 56
Sizing storage 56

Chapter 8 Performance troubleshooting 57
Performance tools 57
CPU 57
Hyper-threaded core-sharing mode 58
Memory 58
I/O 58
NUMA systems 58
Network 59
Single processor in XenServer SUSE Linux VM 59

Appendix A Benchmark server configuration 61

About this guide

This document contains guidelines for deploying Infor LN in a virtualized environment. The virtualization technologies covered in this document are VMware vSphere 5, Citrix XenServer 6 and Microsoft Hyper-V R2.

Intended audience

This guide is intended for Infor consultants, partners and customers who are responsible for implementing Infor LN in a virtualized environment. This document contains information that can help system administrators to optimize the performance of their virtualized Infor LN environment.

Terms used in this document

Some of the terms used in this document are defined in the following table:

Term	Definition
VM (virtual machine)	A software implementation of a machine, that is, a computer, which executes programs similar to a physical machine.
Hypervisor	Virtualization works by inserting a thin layer of software directly on the computer hardware or on a host operating system. This software contains a hypervisor, which allocates hardware resources dynamically and transparently.
NUMA (non-uniform memory access)	A computer memory design used in multiprocessors, where the memory access time depends on the memory location relative to a processor. Under NUMA, a processor can access its own local memory faster than non-local memory, that is, memory local to another processor or memory shared between processors.
NUMA node	A processor or a group of processors with local memory.

Term	Definition
VMware ESXi	ESXi is the name of the hypervisor used in the VMware vSphere products. VMware vSphere 5 contains ESXi version 5.
Intel EPT/AMD RVI	Processor capabilities to help minimize the overhead of memory virtualization.

References

The references below include performance tuning documentation from the tested virtualization solution providers.

Infor

Document	Link
Infor Solution License Manager 7.3 – Installation and Configuration Guide	InforXtreme document U9200 US

VMware vSphere 5

Document	Link
Performance Best Practices for VMware vSphere® 5.0	http://www.vmware.com/pdf/Perf_Best_Practices_vSphere5.0.pdf
VMware vSphere 5 editions	http://www.vmware.com/products/datacenter-virtualization/vsphere/compare-editions.html

Citrix XenServer 6

Document	Link
Configuration limits	http://www.citrix.com/site/resources/dynamic/salesdocs/Citrix_XenServer_6_Configuration_Limits.pdf
XenServer 6 editions	http://www.citrix.com/English/ps2/products/subfeature.asp?contentID=2300456

Microsoft Hyper-V R2

Document	Link
Performance Tuning Guidelines for Windows Server 2008 R2	http://www.microsoft.com/whdc/system/sysperf/Perf_tun_srv-R2.mspx
Virtualization with Hyper-V: FAQ	http://www.microsoft.com/windowsserver2008/en/us/hyperv-faq.aspx
Optimizing performance on Hyper-V	http://technet.microsoft.com/en-us/library/cc768529(BTS.10).aspx

Contacting Infor

If you have questions about Infor products, go to the Infor Xtreme Support portal at www.infor.com/inforxtreme.

If we update this document after the product release, we will post the new version on this Web site. We recommend that you check this Web site periodically for updated documentation.

If you have comments about Infor documentation, contact documentation@infor.com.

Chapter 1 Introduction to virtualization

1

What is virtualization?

Today's x86 computer hardware was designed to run a single operating system and a single application, leaving most machines vastly underutilized. Virtualization lets you run multiple virtual machines on a single physical machine, with each virtual machine sharing the resources of that one physical computer across multiple environments. Different virtual machines can run different operating systems and multiple applications on the same physical computer.

How does virtualization work?

To create a fully functional virtual machine that can run its own operating system and applications similar to a "real" computer, you can use software such as VMware vSphere, Citrix XenServer or Microsoft Hyper-V to transform or "virtualize" the hardware resources of an x86-based computer, including the CPU, RAM, hard disk, and network controller.

Virtualization works by inserting a thin layer of software directly on the computer hardware or on a host operating system. This software contains a virtual machine monitor or hypervisor that allocates hardware resources dynamically and transparently.

Multiple operating systems run concurrently on a single physical computer, and share hardware resources with each other. By encapsulating an entire machine - including the CPU, memory, operating system, and network devices - a virtual machine is completely compatible with all standard x86 operating systems, applications, and device drivers. You can safely run several operating systems and applications simultaneously on a single computer, with each having access to the resources it requires.

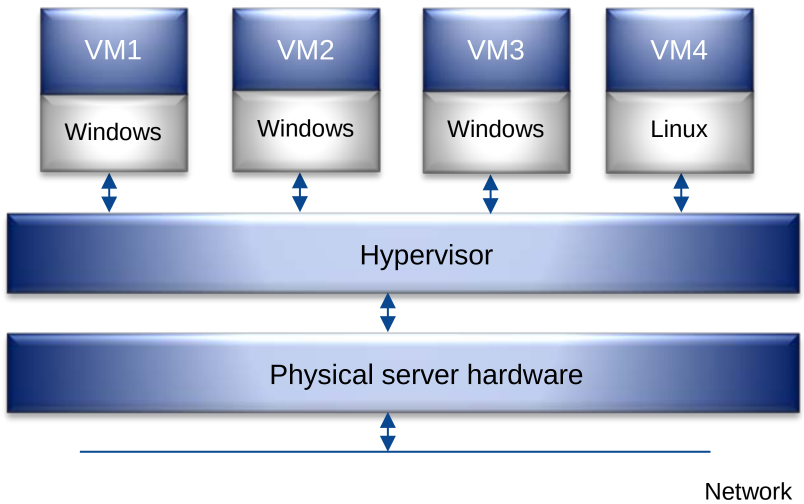


Figure 1: Architecture of a virtualized server

Advantages and disadvantages of virtualization

The advantages of virtualization include the following:

- You get more out of your existing resources. Pool common infrastructure resources and break the legacy “one application to one server” model with server consolidation.
- You can reduce datacenter costs by reducing your physical infrastructure and improving your server to admin ratio. Fewer servers and related IT hardware means reduced real estate and reduced power and cooling requirements. With better management tools, you can improve your server to admin ratio so personnel requirements are reduced.
- You can increase the availability of hardware and applications for improved business continuity.
- Securely back up and migrate entire virtual environments with no service interruptions. Eliminate planned downtime and recover immediately from unplanned issues.
- Gain operational flexibility. Respond to market changes with dynamic resource management, faster server provisioning, and improved application deployment.

The disadvantages of virtualization include the following:

- Virtualization adds overhead to the CPU, memory, IO, and network.
- Virtualization adds an additional layer to the hardware and software stack. Therefore, additional complexity is introduced in the following circumstances:
  - When sizing the physical server.
  - When planning VM capacity.
  - When planning multiple VMs on the same physical server.
  - When investigating performance issues.

Types of virtualization

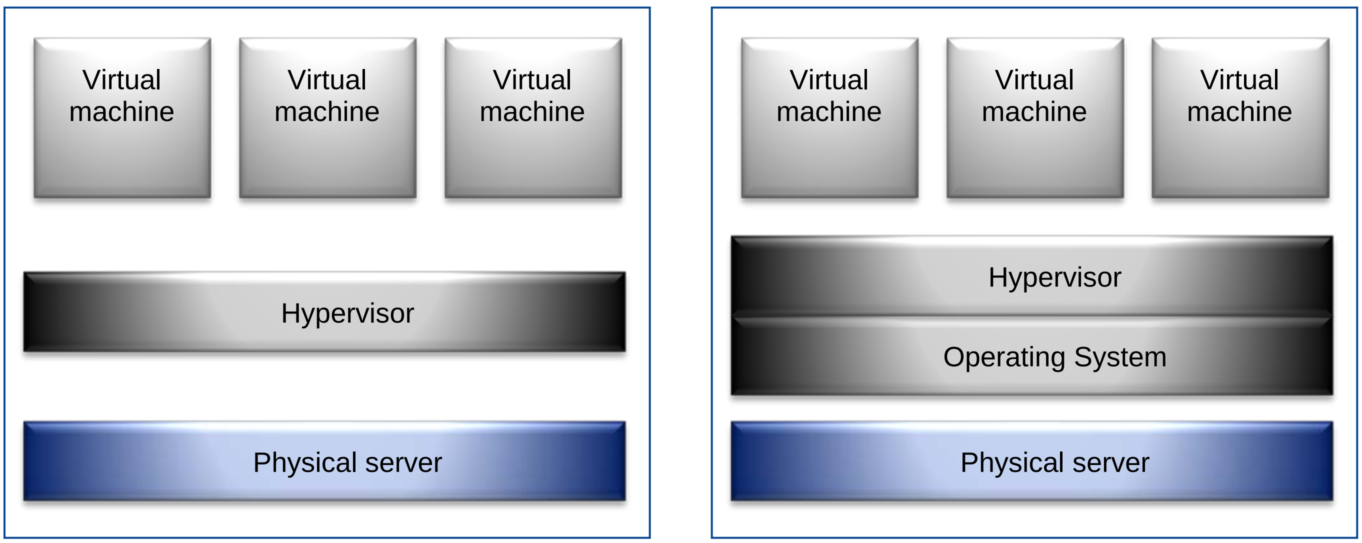
The two types of virtualization techniques available in the market are explained as follows:

- Type 1, or “bare metal” virtualization solutions; these solutions include VMware vSphere, Citrix XenServer and Microsoft Hyper-V.

  Type 1 hypervisors run directly on the system hardware. Figure 2 shows one physical system with a type 1 hypervisor running directly on the system hardware, and three virtual systems using virtual resources provided by the hypervisor.

- Type 2, or “hosted” virtualization solutions; these solutions include VMware Player (formerly named VMware Server), VMware Workstation and Microsoft Virtual Server.

  Type 2 hypervisors run on a host operating system that provides virtualization services, such as I/O device support and memory management. Figure 3 shows one physical system with a type 2 hypervisor running on a host operating system, and three virtual systems using the virtual resources provided by the hypervisor.



Type 1 virtualization

Type 2 virtualization

Figure 2: Type 1 versus type 2 virtualization

Type 1 virtualization solutions are the preferred approach, because they can achieve higher virtualization efficiency by dealing directly with the hardware. Type 1 virtualization provides higher performance efficiency, availability, and security than type 2. Therefore, type 2 virtualization solutions are not suitable (and not supported) for Infor LN.

This document is based on the use of the type 1 virtualization solutions VMware vSphere 5 (ESXi 5 hypervisor), Citrix XenServer 6 (Xen 4.1 hypervisor) and Microsoft Hyper-V R2.

Chapter 2 Infor LN in a virtualized environment

2

This chapter describes several key concepts you must understand when deploying Infor LN in a virtualized environment. The information in this chapter applies to VMware vSphere and Microsoft Hyper-V R2.

Infor Solution License Manager

The Infor Solution License Manager (SLM) is supported in the VMware vSphere and Microsoft Hyper-V R2 virtualized environment starting from version 7.2.0.4.

The Infor Solution License Manager (SLM) is supported in the Citrix XenServer virtualized environment starting from version 7.3.0.2.

Please refer to document U9200 US (Infor Solution License Manager 7.3 – Installation and Configuration Guide) for more information.

Hyper-threading technology

Intel developed hyper-threading technology to increase the performance of its Xeon processors. By using hyper-threading technology, a single processor core behaves similar to two logical processors.

Hyper-threading does not double the performance of a processor core, but can increase the performance by better utilizing idle resources. Hyper-threading performance improvements are application dependent, and some applications may see performance degradation with hyper-threading because many processors' resources, such as cache, are shared between the two logical processors.

Hyper-threading technology is not available in AMD processors.

Infor recommendations for hyper-threading

When using the latest virtualization solutions and using modern Intel Nehalem CPU series (Xeon 5500 series or newer), we recommend that you enable hyper-threading on the host.

Enabling hyper-threading results in an improved system performance for almost all workloads, including Infor LN.

For Infor LN workloads, the additional available CPU capacity created by enabling hyper-threading is approximately 20% (1.2 x). This additional CPU capacity is possible because, with hyper-threading, the CPU resources can be better utilized.

Impact of hyper-threading on batches

Virtualization solutions are hyperthreading aware and will try to schedule a task to a dedicated CPU core if sufficient capacity is available on the host. The hypervisor will also recognize that a task runs on a thread instead of a full CPU core, and compensate this by giving more time to the task. Therefore the enabling of hyper-threading should not cause the batch performance to decrease.

NUMA systems

NUMA (non-uniform memory access) is a technology in which nodes are linked using a high-performance connection. Each node contains a processor (with multiple CPU cores) and has local memory. A node can use “remote memory” from other nodes by using an advanced memory controller. In this case, data must be transferred over the NUMA connection, which is slower than accessing local memory.

NUMA is an alternative approach to SMP (symmetric multi-processing) systems. SMP systems have many processors which must compete for bandwidth on the same system bus, for example, to access memory.

Microsoft Windows, VMware vSphere, Citrix XenServer and Hyper-V are all “NUMA aware” and try to balance the processor and memory load in such a way that accessing remote memory is not required.

The Intel Xeon (5500 series or newer) processors and AMD Opteron processors are using NUMA technology.

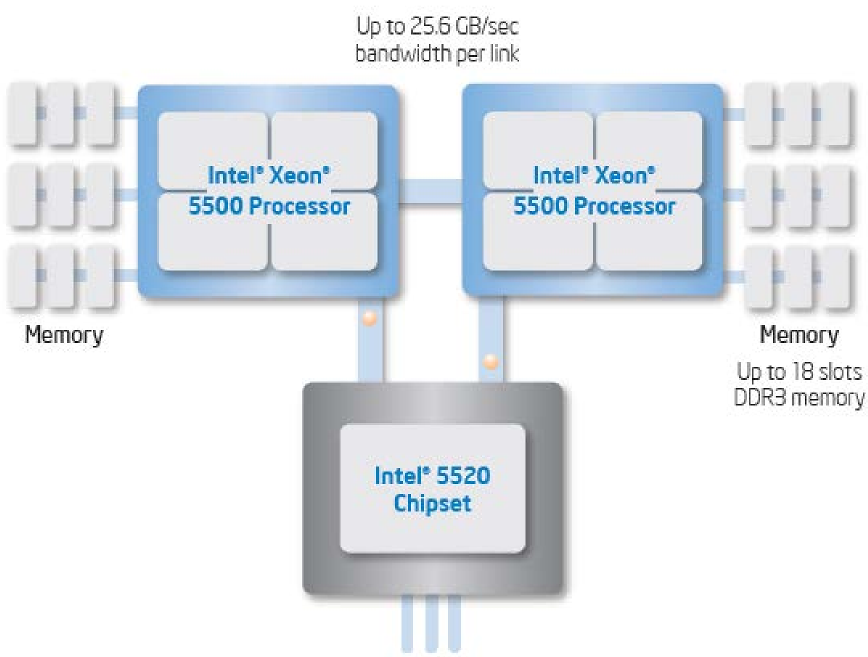


Figure 3: NUMA architecture of a system with 2 Quad core Intel Xeon 5500 series processors; this configuration has two nodes with each node having 4 CPU cores.

Affinity

On NUMA systems which run multiple memory-intensive and/or CPU-intensive VMs on the same server, you can use affinity to place these VMs on different NUMA nodes.

The following table shows the two affinity options:

Affinity setting	Description
CPU affinity	A VM uses only the processors on the specified NUMA node.
Memory affinity	The memory of the VM is allocated only on the specified NUMA node.

Manual affinity settings interfere with the standard resource management algorithms of the hypervisor, which tries to give each VM a fair share of the available processor resources. For example, if 10 VMs are assigned to the first node and two VMs are assigned to the second node, you cannot give all VMs an equal share of the system resources.

Affinity is not supported in Microsoft Hyper-V.

Infor recommendations for NUMA systems

We recommend that, for NUMA systems, you limit the size of a VM to the memory and physical processor capacity of a single NUMA node. Limiting the size avoids the performance penalties that come with using remote memory.

To investigate the number of physical cores available in each NUMA node, see the processor documentation. At the time of writing this document, the number of physical cores per NUMA node may vary from two up to ten.

Example of maximum VM size on a NUMA system

Take an Intel Xeon 5500 series server with 2 * quad core hyper-threaded CPU and 64 GB of internal memory. This server has two NUMA nodes, with each node having the following:

- One quad core hyper-threaded CPU
- Four physical processor cores and eight logical processor threads
- 32 GB of memory

The recommended maximum size of a VM is four virtual CPUs and 32 GB of memory.

Affinity

Infor recommends that you only use affinity on NUMA servers which run multiple memory-intensive and/or CPU-intensive VMs on the same server, and you experience performance problems due to the incorrect scheduling of these VM’s.

In most situations the hypervisor is able to schedule the VM’s correctly and setting affinity is not required and might even reduce performance.

Infor LN and the database used by Infor LN are examples of memory-intensive and CPU-intensive workloads. In case of performance issues, affinity can be used to place the Infor LN application VM and the Infor LN database VM on different NUMA nodes.

Example of using affinity on a NUMA system

Take an Intel Xeon 5500 series server with 2 * quad core hyper-threaded CPU and 64 GB of internal memory. This server has two NUMA nodes, with each node having the following:

- One quad core hyper-threaded CPU
- Four physical processor cores and eight logical processor threads
- 32 GB of memory

Assume that multiple VMs are running on this server; amongst them are the Infor LN application server VM and the Infor LN database VM.

Both the Infor LN VMs are running memory-intensive workloads. To ensure these VMs are scheduled on different NUMA nodes, use the following settings:

VM	CPU affinity	Memory affinity
Infor LN application	Logical processors 0-7	Node 0
Infor LN database	Logical processors 8-15	Node 1
Other VMs	No settings	No settings

Multiple VMs and over-committing resources

Virtualization does not increase server capacity. Virtualization has an overhead which results in less server capacity or less server throughput. The benefit of virtualization is that idle resources can be used more efficiently.

By running multiple VMs on the same host (physical server), overcommitment of the physical resources (CPU, Memory etc.) can occur. In this case all VMs on this host will suffer performance problems.

The following are factors involved in the processing capacity assigned to a VM:

- Number of vCPUs
- Memory assigned
- CPU affinity
- Memory affinity
- CPU Shares (also called relative weight), reservations, and limits
- Memory Shares (also called relative weight), reservations, and limits
- Hyper-threading and core sharing

Example of an overcommitted server

Take a server with 2 * quad core CPUs and 64 GB of internal memory. This server has 8 physical cores.

In this example the server has 8 VMs installed, with each VM having the following:

- Two vCPUs
- 12 GB of internal memory assigned; 6 GB of this memory is reserved. A reservation is the guaranteed minimum amount of memory available to a VM.

This results in the following:

- The maximum CPU usage of these 8 VMs is 16 vCPU. Because the server has eight physical cores, there is CPU overcommitment. This overcommitment will result in a performance penalty if the total CPU load of all VMs exceeds the capacity of the 8 physical cores.
- The initial memory capacity of the VMs is 48 GB, based on the memory reservation. The maximum memory usage of these VMs is 96 GB. Because the server has 64 GB of physical memory, the memory is overcommitted. This overcommitment will result in a performance penalty if the actual total memory usage of the VMs exceeds 64 GB.
- The performance penalty due to the overcommitment is visible in all VMs, except where different reservations or shares settings between the VMs have been used.

Infor recommendations for multiple VM systems

When you deploy Infor LN on a virtual server which hosts other VMs, Infor recommends that you do the following:

- Use reservations and shares, which are also called relative weight, for the CPU to ensure that the Infor LN application and database VMs obtain the required CPU capacity.
- Use reservations for memory to ensure that the Infor LN application and database VMs have sufficient memory.

A reservation is the guaranteed minimum amount of resources available to a VM. Note that by using reservations you make sure that the VM always gets the required resources (CPU and memory) when needed. However in case the VM does not use all of the reserved resources these unused resources cannot be used by other VMs on the same host. Effectively making reservations will reduce the number of VMs that can run on a host. Therefore only use reservations for performance critical VMs.

Storage

The general rule for virtualization is to not make use of local storage. By using local storage, virtualization technologies such as live migration and dynamic resource balancing (DRS) are not available. Local storage does not provide flexibility in reallocating storage between servers.

Using shared storage is the correct way to implement a virtualized infrastructure. Implementing shared storage provides maximum flexibility and supports virtualization features such as vMotion and live migration.

The following are the shared storage options:

- SAN using iSCSI or Fiber channel
- NAS using NFS/NTFS

Infor recommends using a Fiber channel SAN to implement Infor LN in virtualized environments for optimal performance.

Virtual storage types

Virtualization solutions support the following types of storage:

- Virtual disks (VMDK or VHD), which are the most commonly used storage type.
- Raw device mappings or pass-through disks, which allow direct access to SCSI disks or LUNs.

There are several configuration options when using virtual disks. In general, always use pre-allocated types, which are also called “thick” or fixed size types, for I/O intensive disks; for example, disks hosting the database.

Thin provisioned disks, also called dynamically expanding disks, do not pre-allocate all space in advance, but only when it is required. Thin provisioned disks incur a performance penalty in case a new block must be allocated. Additionally, a performance penalty is included because of the continuous updating of VM metadata.

We recommend that you do not use snapshots, they incur a performance penalty because of the use of a redo log mechanism. In case of multiple snapshots a chain of snapshots is created which further reduce performance.

Raw device mappings or pass-through disks should only being used in specific situations. Please consult the vendor manuals for more information.

Network settings in a 3-tier configuration

For 3-tier configurations, that is, where the Infor LN application and the database are on different servers, you can improve network performance by enabling options on the network adapter. The availability of these options depends on the brand and type of the network adapter. The network adapter options include the following:

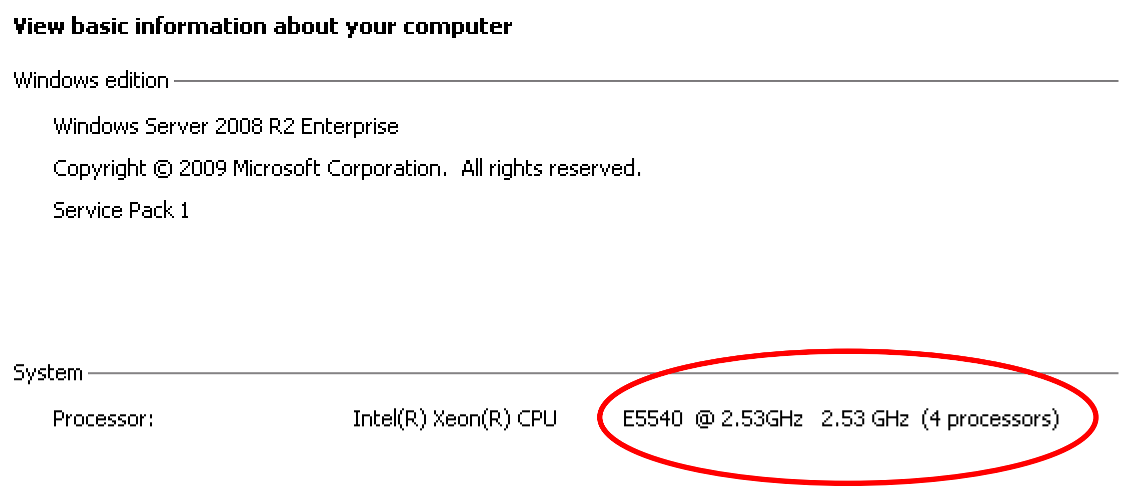
- Disable interrupt coalescing; this reduces CPU overhead for the virtualization of the network.
- Enable TCP offloading; this improves performance by allowing the VM to offload network processing to hardware.

- Enable Virtual Machine Queue (VMQ); which allows physical computer Network Interface Cards (NICs) to use Direct Memory Access to place the contents of packets directly into the VM memory.

In general, we recommend that you have at least a 1-Gbps network connection between virtual servers in an Infor LN 3-tier configuration.

Virtual CPU sockets

By default virtualization solutions map each virtual CPU to a CPU socket with a single core. For example when you create a 4 vCPU virtual machine, this will be configured as 4 virtual CPU sockets with 1 core each. In a Windows VM this is shown as 4 processors in the computer properties:



Some virtualization solutions have an option to increase the number of cores per CPU socket. This setting is mostly used for licensing reasons and has no significant performance impact. In case you have an application that is licensed per CPU socket it is recommended to make the virtual cores per socket match the physical server CPU configuration.

For example when configuring a 16 vCPU VM with Windows 2008 R2 x64 Enterprise Edition you will only see 8 vCPUs in the virtual machine. This is because the Enterprise Edition license is restricted to 8 CPU sockets. By increasing the cores per socket parameter you can make the full 16 vCPUs available in the VM.

Note on VMware vSphere

The setting does make a small difference when used with vNUMA; when used with vNUMA the core/socket setting that you set will be used in the vNUMA representation when presenting the NUMA topology to the VM. In most cases this settings will not make a performance difference.

The general recommendation is to configure it to match the physical resources in the host.

Chapter 3 VMware vSphere

3

This chapter contains specific information about the VMware vSphere 5 virtualization solution. The following topics are covered:

- Infor LN FP6 benchmark results
- Hyper-threading and NUMA
- Single and multiple VM performance and scalability based on Infor LN FP6 benchmarks
- VMware vMotion and Fault Tolerance (FT)

The measurements in this chapter are based on the server configurations detailed in appendix A and are based on VMware vSphere 5 (ESXi 5.0 build 515841). The benchmarked platform is Windows 2008 R2 x64 SP1 and SQL Server 2008 R2 x64 SP1.

Infor LN benchmark results

The graph below shows the maximum number of benchmark users in a VMware vSphere 5 VM. In the 2-tier configuration the Infor LN application and the database are running in a single VM. In the 3-tier configuration the Infor LN application runs in a VMware vSphere 5 VM and the database runs on a separate physical server.

The benchmark simulates Infor LN users. The number of users is increased up to pre-defined response time criteria. At this point the benchmark uses 100% CPU in the VM.

Hyper-threading and VMware vSphere

Hyper-threading technology (sometimes also called simultaneous multithreading, or SMT) allows a single physical processor core to behave like two logical processors, essentially allowing two independent threads to run simultaneously. Unlike having twice as many processor cores - that can roughly double performance - hyper-threading can provide anywhere from a slight to a significant increase in system performance by keeping the processor pipeline busier.

If the hardware and BIOS support hyper-threading, ESXi automatically makes use of it.

Notes when using hyper-threading:

- When ESXi is running on a system with hyper-threading enabled, it assigns adjacent CPU numbers to logical processors on the same core. Thus CPUs 0 and 1 are on the first core, CPUs 2 and 3 are on the second core, and so on. ESXi systems manage processor time intelligently to guarantee that load is spread smoothly across all physical cores in the system. If there is no work for a logical processor it is put into a halted state that frees its execution resources and allows the virtual machine running on the other logical processor on the same core to use the full execution resources of the core.
- Be careful when using CPU affinity on systems with hyper-threading. Because the two logical processors share most of the processor resources, pinning vCPUs, whether from different virtual machines or from a single SMP virtual machine, to both logical processors on one core (CPUs 0 and 1, for example) could cause poor performance.
- ESXi provides configuration parameters for controlling the scheduling of virtual machines on hyper-threaded systems (Edit virtual machine settings > Resources tab > Advanced CPU). These settings are described in the next paragraph.

Hyper-threaded core sharing options

VMware ESX has three options to specify how the virtual CPUs of a VM can share physical cores on a hyper-threaded system. Two virtual CPUs share a processor core if they are running simultaneously on the logical CPUs of this processor core.

The following table shows the hyper-threaded core sharing options in VMware vSphere:

Option	Description
Any	The default for all VMs running on a hyper-threaded system. The virtual CPUs of a VM with this setting can freely share cores with other virtual CPUs from this or any other VM at any time.
None	Virtual CPUs of a VM do not share cores with each other or with virtual CPUs from other VMs. By enabling this option, each virtual CPU of this VM is always assigned to a whole processor core, with the other logical CPU on that core being placed in the halted state. This option is like disabling hyper-threading for that one virtual machine.
Internal	Virtual CPUs from this VM cannot share cores with virtual CPUs from other VMs. They can share cores with the other virtual CPUs from the same VM.

By selecting the “None” or “Internal” hyper-threading option, the VM is isolated from other VMs. Therefore, the VM does not suffer from performance issues due to core sharing.

The trade-off for configuring hyper-threading constraints should also be considered. With this setting, there can be cases where there is no CPU core to which a descheduled virtual machine can be migrated, even though one or more logical cores are idle. As a result, it is possible that virtual machines with hyper-threading constraints can experience performance degradation, especially on systems with a limited number of CPU cores.

Recommendations for core sharing

For nearly all workloads, custom hyper-threading settings are not necessary. For VMware vSphere 5, we recommend that you leave the core sharing option to the default “Any”. Instead, use CPU reservations and CPU shares, which are also known as relative weight, to ensure the Infor LN VM acquires sufficient CPU resources.

Benchmark note

In our benchmarks with VMware vSphere on the Intel Xeon 5500 series NUMA platform, we did not see noticeable performance differences using the hyper-threaded core sharing modes “Any”, “None” or “Internal”.

If you experience performance issues when running multiple VMs on a hyper-threaded server, change the default setting from “Any” to “Internal” or “None” to check for performance gains. For example, even though the ESXi scheduler tries to dynamically run higher-priority virtual machines on a whole core for longer durations, you can further isolate a high-priority virtual machine from interference by other virtual machines by setting its hyper-threading sharing property to “None”.

VMware vSphere and NUMA

VMware vSphere uses a NUMA scheduler to dynamically balance processor load and memory locality:

1. Each VM is assigned one or more home nodes at startup.
2. When memory is allocated to a VM, the memory is preferentially allocated from the home node(s).
3. The NUMA scheduler can dynamically change a VM’s home node to respond to changes in system load.
4. The NUMA scheduler can migrate a VM’s memory to its new home node to improve memory locality.

Single VM performance and scalability

Figure 6 shows the single VM performance and scaling in VMware vSphere. The used workload is an Infor LN batch which takes, on average, one full CPU core.

The graph shows that the performance of this workload is stable and does not depend on the configured number of vCPUs.

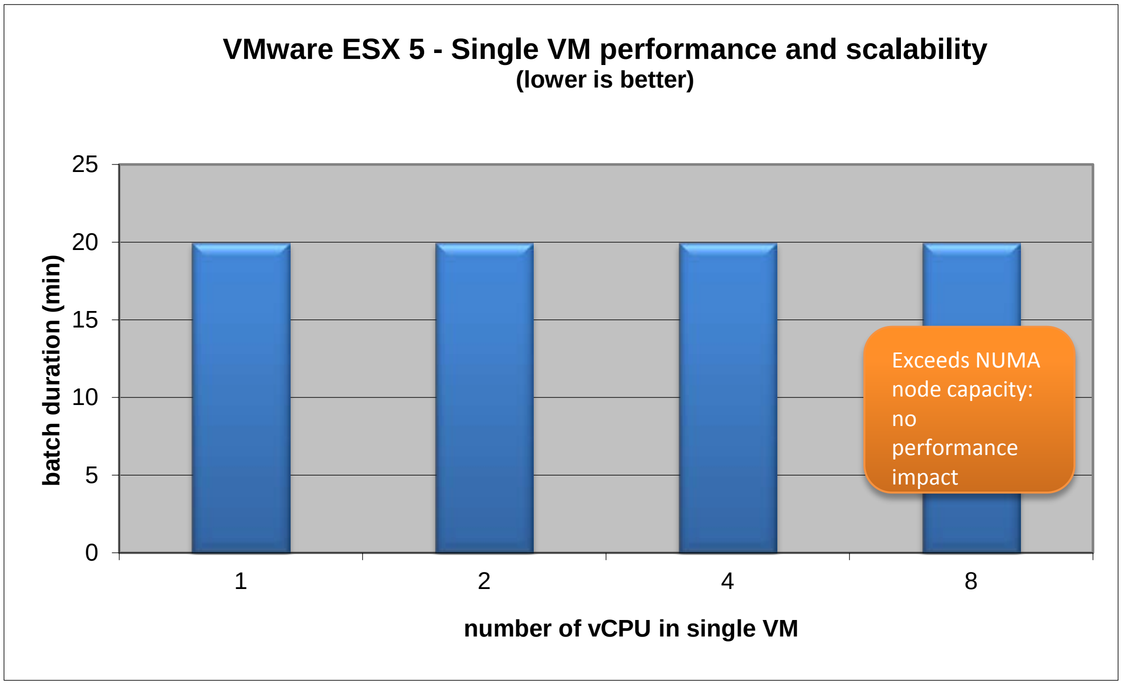


Figure 4: Single VM performance and scalability in Vmware ESX

When configuring the VM with 8 vCPU the NUMA node capacity of 4 cores on the benchmark server is exceeded. There is no performance impact because the scheduler can effectively manage the benchmarked workload (one full CPU core) on the capacity of a single NUMA node. There will be a performance penalty in case of larger workloads.

Conclusions

The performance of a VM does not suffer noticeably if more vCPUs are added to the VM (up to the capacity of the NUMA node). Configuring a VM with more vCPUs than required cause’s performance problems if there are more VMs running on the server and there is CPU contention.

Multiple VM performance and scalability

Figure 7 shows the multiple VM performance and scaling in Vmware vSphere. The used workload is an Infor LN batch which takes, on average, one full CPU thread. This workload is run in each VM participating in the test.

To force the over-commitment of the system, the VMs were affinitized to a single Quad Core CPU.

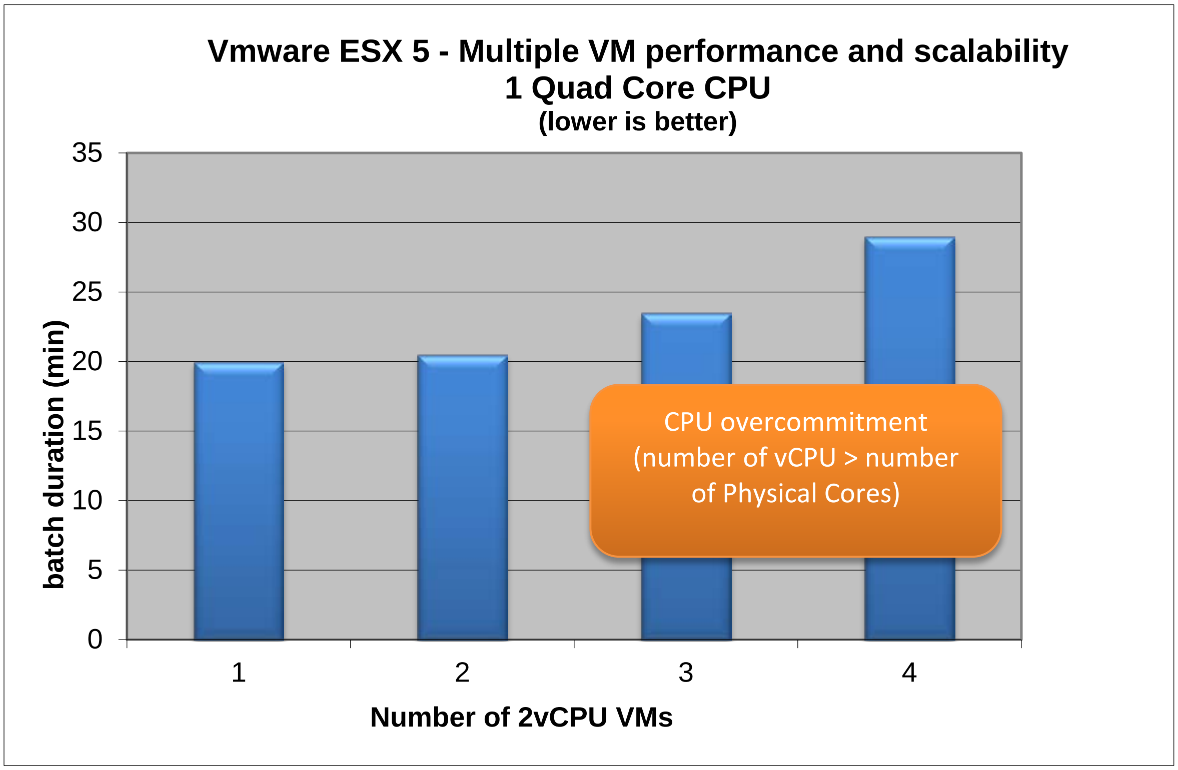


Figure 5: Multiple VM performance and scalability in Vmware ESXi

Starting with three VMs, the total number of vCPUs exceeds the number of physical cores (four) in the benchmark server configuration.

Conclusions

The results show that over-committing a server decreases the performance of all VMs in the test. In the 3 VM configurations, processing takes approximately 20 percent longer; in the 4 VM configurations, processing takes approximately 45 percent longer.

Configuration maximums

Before you select a virtualization product, check the configuration maximums of the product. There are limitations with the maximum size of a Virtual Machine and the size of the host. Examples of configuration maximums in vSphere 5.0 compared to vSphere 4.1 are shown below. Consult the product documentation for a full list of configuration maximums.

The table below shows some maximum configurations of a virtual machine:

VM configuration maximums	Virtual CPU	Virtual Memory
vSphere 4.1	8 vCPU	255 GB vRAM
vSphere 5.0	32 vCPU	1 TB vRAM
vSphere 5.1	64 vCPU	1 TB vRAM

The table below shows some host configuration maximums:

Host configuration maximums	VMs per host	RAM per host
vSphere 4.1	320	1 TB
vSphere 5.0 / 5.1	512	2 TB

VMware vSphere 5 licensing model changes

Note that in VMware vSphere 5 the licensing model has changed from a processor based license to a combined processor and memory license model. The vSphere virtual machine memory (vRAM) entitlement per processor license is 32 – 96 GB depending on the vSphere edition.

VMware vMotion

By using VMware vMotion, you can enable the live migration of running VMs from one physical server to another with zero downtime, and have continuous service availability and complete transaction integrity. Consult the VMware product manuals for more information on VMware vMotion.

Infor benchmark results

Infor has tested vMotion while running benchmarks of up to 200 concurrent Infor LN users. This benchmark has been run using VMware ESXi 4.1.

Figure 8 shows the results for a 4 vCPU VM with different memory configurations. The "Time" is the total time it takes for vSphere to report the vMotion as completed. The "blackout" period is the period in which the server does not respond to network traffic. Benchmark response times only increased during this blackout period.

To avoid timeouts during the "blackout" period, a TCPIP parameter has been increased in the registry. Without setting this parameter, Infor LN users receive a lost connection error.

The table shows that the total vMotion time depends on the memory size of the VM and the number of concurrent users.

	No users running		**100 users running**		**200 users running**	
Memory (GB)	**vMotion time (sec)**	**Blackout (sec)**	**vMotion time (sec)**	**Blackout (sec)**	**vMotion time (sec)**	**Blackout (sec)**
7.5	37	4	-	-	-	-
15	71	4	236	25	-	-
30	136	4	289	28	462	41
60	204	4	422	28	504	41

Figure 6: Total vMotion time for a single 4 vCPU VM when running an Infor LN benchmark

Requirements for running VMware vMotion on Infor LN

To run VMware vMotion on Infor LN environments, you must have/complete the following:

- At least a 1-gigabit Ethernet connection between the servers performing the vMotion.
- Increase the TCPMaxDataRetransmissions key in the registry to avoid TCPIP timeouts during the transition period.

  Add a DWORD TcpMaxDataRetransmissions registry key with a decimal value of 30 in HKLM\System\CurrentControlSet\Services\Tcpip\Parameters

  For more information, see http://technet.microsoft.com/en-us/library/cc938210.aspx

  After you change this parameter, reboot the server.

VMware ESX Fault Tolerance (FT)

vSphere Fault Tolerance (FT) provides continuous availability for applications in the event of server failures by creating a live shadow instance of a virtual machine that is in virtual lockstep with the primary instance. By allowing instantaneous failover between the two instances in the event of hardware failure, FT eliminates even the smallest chance of data loss or disruption.

VMware Fault Tolerance is supported by VMs with only a single virtual processor, that is, one vCPU.

Infor benchmark results

Infor has tested Fault Tolerance while running benchmarks of up to 50 concurrent Infor LN users. Fault Tolerance was activated by removing the network cable from the VM. This action resulted in a small blackout period; during this period, benchmark response times increased. This benchmark has been run using VMware ESXi 4.1.

To avoid timeouts during the blackout period, a TCPIP parameter has been increased in the registry. If this parameter is not set, Infor LN users receive a lost connection error.

The VM involved in the fault tolerance configuration is continuously updated by the VMware Fault Tolerance. This updating has a bandwidth usage and a time lag, also known as a vLockstep interval, which was measured in the Infor LN benchmark (one vCPU VM with 16 GB of internal memory).

The table below shows time lag and the bandwidth usage between the primary VM and the Fault Tolerance VM for a number of benchmark scenarios.

Benchmark scenario	VM CPU usage	VM Memory usage (MB)	vLockStep interval (seconds)	Log bandwidth (Kbps)
Idle	15 %	2457	0.01	261
25 users	60 %	3508	0.008	839
50 users	85 %	5024	0,01	1292

Requirements for running VMware Fault Tolerance on Infor LN

In general, Infor does not recommend that you run Infor LN with a VM with less than two vCPU. We therefore do not recommend that you use Fault Tolerance in VMware vSphere on Infor LN environments, because it only supports a single vCPU.

If you decide to run Fault Tolerance on Infor LN environments, you must do the following:

- Use at least a 1-gigabit Ethernet connection between the servers involved.
- Increase the TcpMaxDataRetransmissions registry key to avoid TCPIP timeouts during the transition period.

  Add a DWORD TcpMaxDataRetransmissions registry key with a decimal value of 30 in HKLM\System\CurrentControlSet\Services\Tcpip\Parameters

For more information, see http://technet.microsoft.com/en-us/library/cc938210.aspx

Chapter 4 Citrix XenServer

4

Introduction

Citrix XenServer 6.0 is a complete virtual infrastructure solution that includes a 64-bit hypervisor, an intuitive virtualization management console, live migration capabilities, and the tools necessary to convert existing workloads from a physical to virtual environment.

The Advanced, Enterprise and Platinum Editions of XenServer offer rich management and automation capabilities that provide full datacenter automation, advanced integration and management, and key performance optimization features.

Included features are:

- Centralized management console
- Dynamic workload balancing
- High availability
- Host power management
- Disaster recovery
- Intellicache (used for XenDesktop)
- XenConvert
- XenMotion

Citrix XenServer 6.0 is based on the open source Xen 4.1 Hypervisor (www.xen.org). It supports both Windows and Linux virtual machines.

Architecture

XenServer is a thin hypervisor running directly on the hardware. It includes a Linux based service console or domain (referred to as DOM0) which is required to run before any virtual machine can be started.

The XenServer Tools, containing high performance drivers and a management agent, must be installed in each VM.

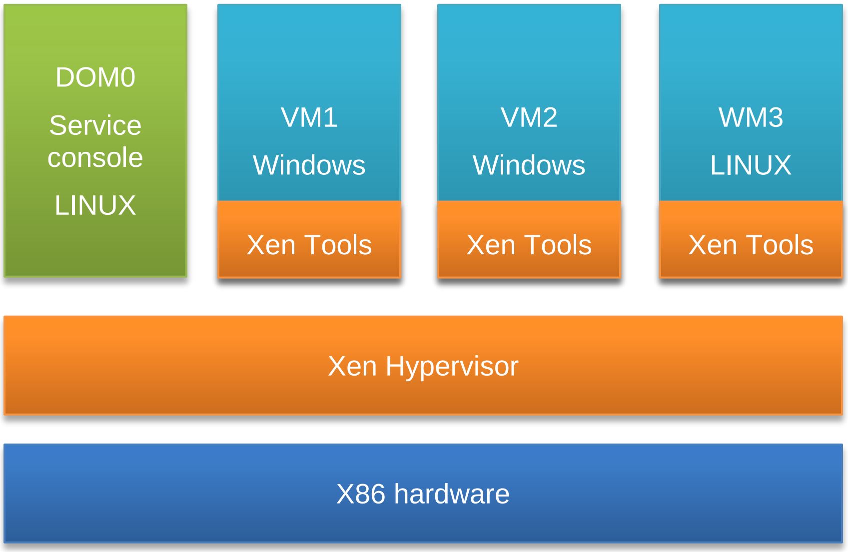


Figure 7: Citrix XenServer architecture

Managing XenServer

You can manager a XenServer using the free XenCenter management tool. More advanced options are available in the XenServer CLI (command line interface). You can use the XenServer CLI by logging in to the XenServer console using ssh.

A full list of available XenServer CLI commands is available in the XenServer administration guide.

Configuration limits

The configuration limits for the physical host and the virtual machines are described in this document:

http://www.citrix.com/site/resources/dynamic/salesdocs/Citrix_XenServer_6_Configuration_Limits.pdf

For Citrix XenServer 6 the maximum configuration of a Virtual Machine is:

Configuration item	Maximum value
Number of vCPUs	16
Virtual memory	128 GB

XenServer High Availability

With High Availability enabled, Virtual machines from a failed server are automatically redistributed and restarted on other physical servers within a designated resource pool according to priority and resource availability.

Notes:

- To configure High Availability, all hosts in the pool must have licenses for Citrix XenServer Advanced Edition or higher.
- High Availability requires the use of shared storage for the VMs; additionally shared storage device is required for the heartbeat.

Performance tuning XenServer 6

Important XenServer settings

This chapter contains important XenServer settings.

Installation template

Make sure to select the correct template when installing a new VM, for example when installing Windows 2008 R2 x64 select the template shown below.

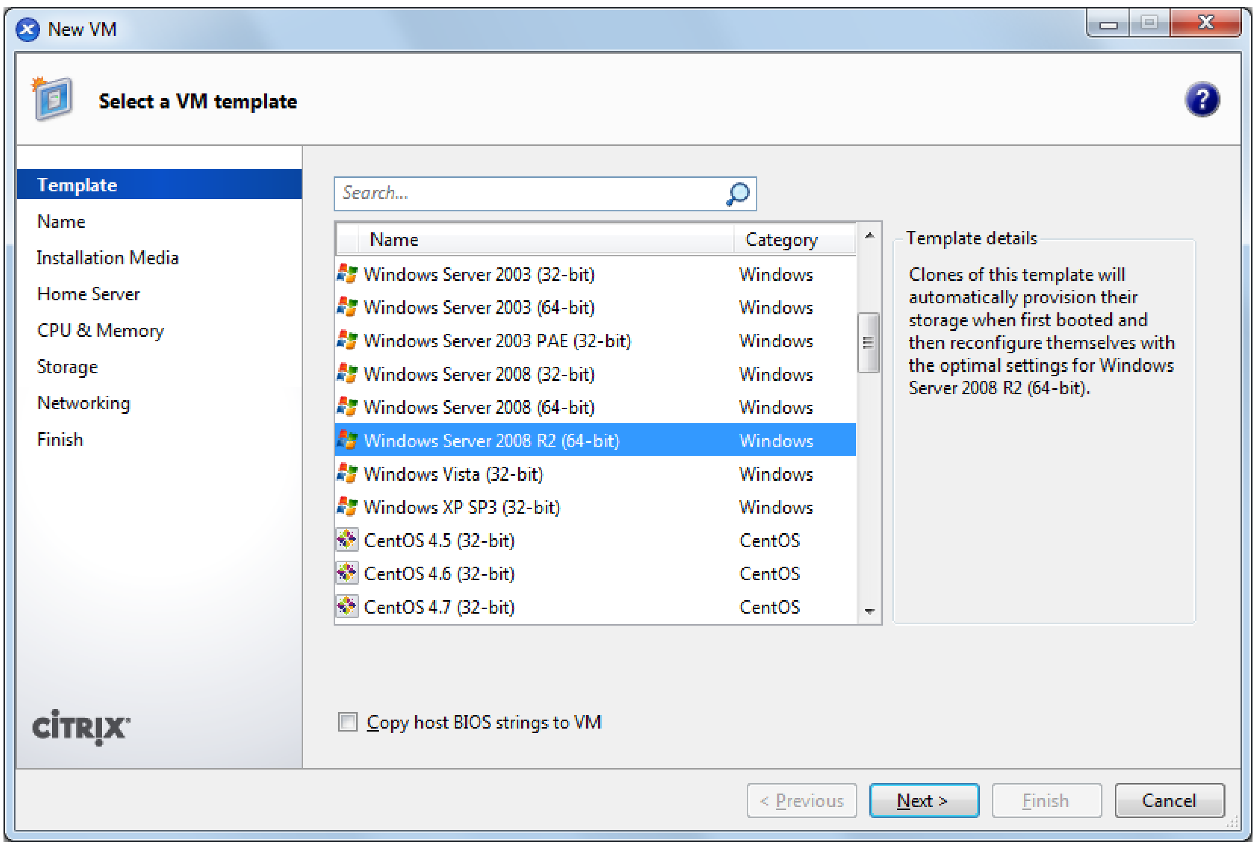


Figure 8: Template selection in Citrix XenServer 6

Install XenServer tools

Make sure to install the latest tools in each VM. The VM general properties should show the state optimized as shown below.

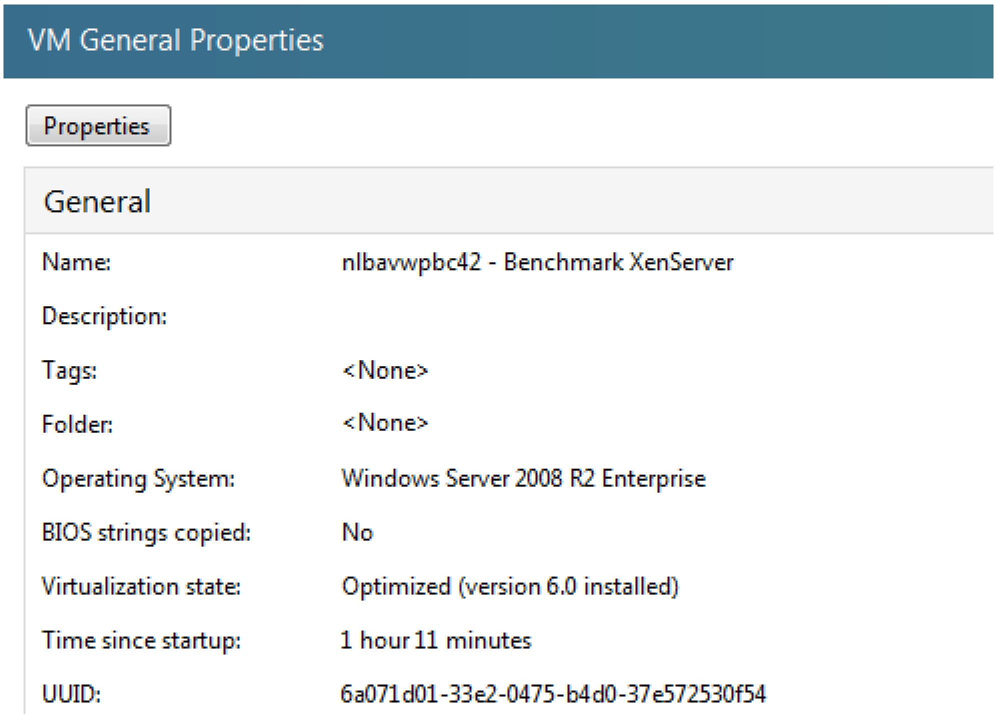


Figure 9: Properties of a VM with XenServer tools installed showing state optimized

Monitoring DOM0 and VM performance

You can monitor performance of a VM using the XenCenter management tool. It is also possible to login to the service console with ssh and use these tools:

- xentop
- sar
- vmstat

Increase DOM0 memory configuration

In case multiple VMs are deployed the DOM0 default memory size of 752MB can be insufficient. The below article describes how to increase this to a maximum of 2.94GB.

http://support.citrix.com/article/CTX126531

XenServer CPU usage

In the default setup of XenServer, physical CPU resources are split between virtual machines using a fair-share balancing algorithm. This ensures all of the virtual machines get their share of the system's CPU resources. XenServer has a smart CPU load balancing scheduler that automatically moves the virtual machine CPUs between physical CPU cores to provide the best performance for all virtual machines. In some situations it might be necessary to modify the default setup to grant virtual machines more or less resources to optimize the use of processor resources.

XenServer enumerates physical CPUs in a depth-first fashion. For a system with both hyperthreading and multiple cores, this would be all the hyperthreads on a given core, then all the cores on a given socket, and then all sockets. For example: On a two socket, dual-core, hyperthreaded Xeon, the CPU order would be:

socket0				socket1			
core0		core1		core0		core1	
ht0	ht1	ht0	ht1	ht0	ht1	ht0	ht1
#0	#1	#2	#3	#4	#5	#6	#7

Controlling CPU usage in the XenCenter management tool is limited to setting the number of vCPUs in a VM and determining the vCPU priority.

The vCPU priority determines the priority in case there is CPU contention on the XenServer host. Setting priority to the highest priority determines that a VM gets twice as much CPU compared to a normal priority VM.

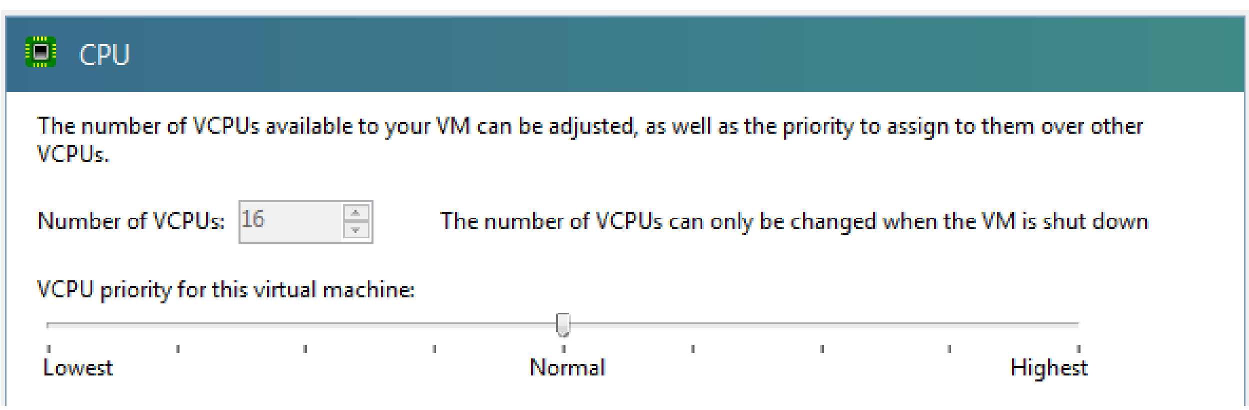


Figure 10: Setting number of vCPU and vCPU priority in XenCenter

There are more advanced options available through the CLI, these include:

Capping CPU

The CPU cap optionally fixes the maximum amount of CPU a VM can use.

To retrieve the list of installed virtual machines with their UUIDs:

```
xe vm-list
```

To cap a VM:

```
xe vm-param-set uuid=<VM UUID> VCPUs-params:cap=80
```

The VM from the above example with a cap of 80 will only be able to use 80 percent of one physical CPU even if the XenServer Host has idle CPU cycles.

The cap is expressed in percentage of one physical CPU: 100 is one physical CPU, 50 is half a CPU, 400 is 4 CPUs, and so on. The default, 0 (zero), means there is no upper cap.

Setting CPU affinity (pinning)

Setting CPU affinity is not supported by the XenCenter management tool. To assign a VM to specific CPUs you need to login to the service console of your XenServer using ssh.

To retrieve the list of installed virtual machines with their UUIDs:

```
xe vm-list
```

Affinitize the CPUs of a VM to the first 8 CPUs of a system:

```
xe vm-param-set uuid=<VM UUID> VCPUs-params:mask=0,1,2,3,4,5,6,7
```

Setting vCPU priority

The VCPU priority weight parameters can also be modified to grant a specific VM more CPU time than others. This can be done using XenCenter or by using the CLI.

To retrieve the list of installed virtual machines with their UUIDs:

```
xe vm-list
```

To change the weight:

```
xe vm-param-set uuid=<VM UUID> VCPUs-params:weight=512
```

The VM from the above example with a weight of 512 will get twice as much CPU as a domain with a weight of 256 on a busy XenServer Host where all CPU resources are in use.

Valid weights range from 1 to 65535 and the default is 256.

CPU recommendations for Infor LN

When you deploy Infor LN on a virtual server which hosts other VMs, Infor recommends that you use vCPU priority to make sure the Infor LN application and database obtain the required CPU capacity.

XenServer does not support the concept of reserving CPU capacity for a VM. Pinning vCPUs to physical CPUs is the only way to make dedicated CPU capacity available to a VM. Using pinning has some disadvantages, therefore using this is only recommended for experienced system administrators.

Hyperthreading

Infor recommends enabling Hyperthreading. Hyperthreading in modern Intel Nehalem CPU series (Intel Xeon 5500 series and later models) does increase the overall CPU capacity of a server.

Virtualization solutions are hyperthreading aware and will try to schedule a task to a dedicated CPU core if sufficient capacity is available on the host. The hypervisor will also recognize that a task runs on a thread instead of a full CPU core, and compensate this by giving more time to the task.

XenServer memory usage

Memory allocation and Dynamic Memory Control

When you first create a VM, XenServer allocates it a fixed amount of memory. You can improve how XenServer uses physical memory by using Dynamic Memory Control (DMC), a memory management feature that enables dynamic reallocation of memory between VMs.

Without DMC, when a server is full, starting further VMs will fail with "out of memory" errors. To reduce the existing VM memory allocation and make room for more VMs you must edit each VM's memory allocation manually and then reboot the VM. With DMC enabled, even when the server is full, XenServer attempts to reclaim memory by automatically reducing the current memory allocation of running VMs within their defined memory ranges. When successful, XenServer reclaims memory without needing to reboot.

Note: The DMC feature is available in Citrix XenServer Advanced Edition or higher.

Memory recommendations for Infor LN

When you deploy Infor LN on a virtual server which hosts other VMs, Infor recommends that you allocate dedicated memory to the Infor LN application and database VM. Dedicated memory can be assigned by using the fixed memory setting or setting the minimum memory to the required value when using Dynamic Memory Control.

Storage in Citrix XenServer

Citrix XenServer 6 supports these storage types:

Virtual disk format	Storage type	Table heading
File-based VHD	EXT3	Full thin provisioning
Logical volume based VHD (LVHD)	LVM	Thick provisioned, thin provisioning for snapshots and cloning
LUN per VDI	Netapp, EqualLogic, StorageLink	Most high-performance storage method

By default the LVM storage type is being used. This is the most commonly used storage type. LVM is a blockdevice-based storage model which has better performance compared to the file-based EXT3 storage type.

When installing XenServer with the ‘Enable thin provisioning’ option the default storage type changes from LVM to EXT3. For using the ‘IntelliCache’ option in XenDesktop it is required to use the EXT3 storage type.

LUN per VDI directly uses the interface of the disk arrays and has less overhead compared to the other storage types. This format directly maps a raw LUN as a virtual disk image and is the most high-performance storage method.

The storage type is shown in the ‘type’ column in the XenCenter storage tab.

Storage recommendations for Infor LN

When you deploy Infor LN on a virtual server, Infor recommends that you use the LVM storage type for performance reasons.

Benchmark results

The benchmark results in this document are based on the server configuration detailed in appendix A and on Citrix XenServer 6.0.2 (build 53456p). Two platforms are benchmarked: Windows 2008 R2 x64 SP1 with SQL Server 2008 R2 x64 SP1 and SUSE Linux 11 SP1 x64 with Oracle 11.2.0.3.

Infor LN OLTP Benchmark - Windows

The graph below shows the maximum number of benchmark users in a XenServer VM. In the 2-tier configuration the Infor LN application and the database are running in a single VM. In the 3-tier configuration the Infor LN application runs in a VM and the database runs on a separate physical server.

The benchmark simulates Infor LN OLTP users. The number of users is increased up to pre-defined response time criteria. At this point the benchmark uses 100% CPU in the VM.

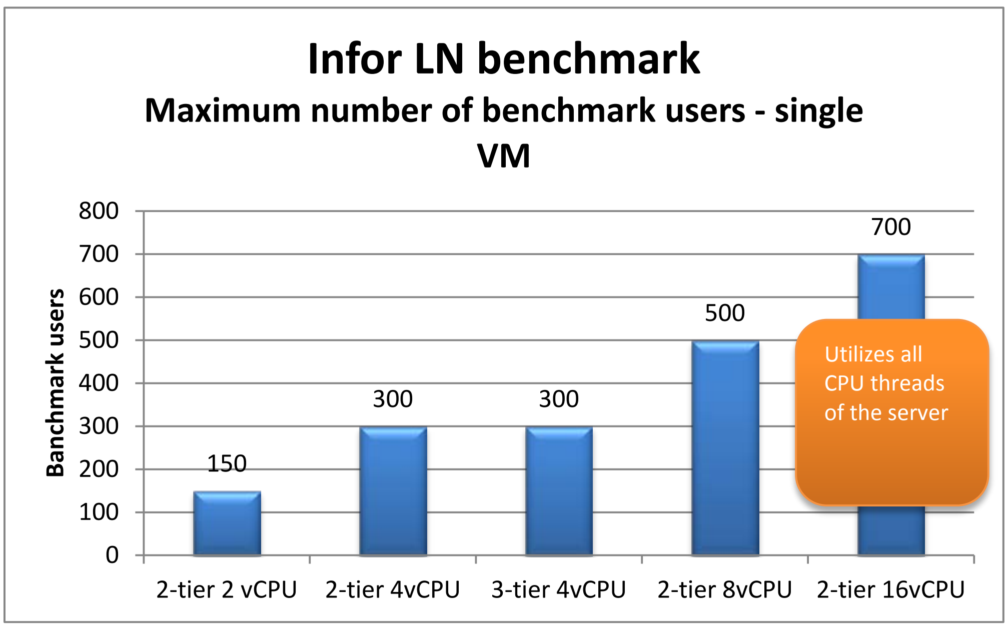


Figure 11: Maximum number of Infor LN benchmark users - Windows

Benchmark note

This benchmark shows that Citrix XenServer performance scales reasonable well. With the 16 vCPU all threads in the benchmark server are in use (including the hyperthreads, the benchmark server has 2 CPUs, 8 cores and 16 threads). This test shows that hyperthreading increases the capacity of the server.

Infor LN OLTP Benchmark - Linux

The graph below shows the maximum number of benchmark users in a XenServer VM. In the 2-tier configuration the Infor LN application and the database are running in a single VM. In the 3-tier configuration the Infor LN application runs in a VM and the database runs on a separate physical server.

The benchmark simulates Infor LN OLTP users. The number of users is increased up to pre-defined response time criteria. At this point the benchmark uses 100% CPU in the VM.

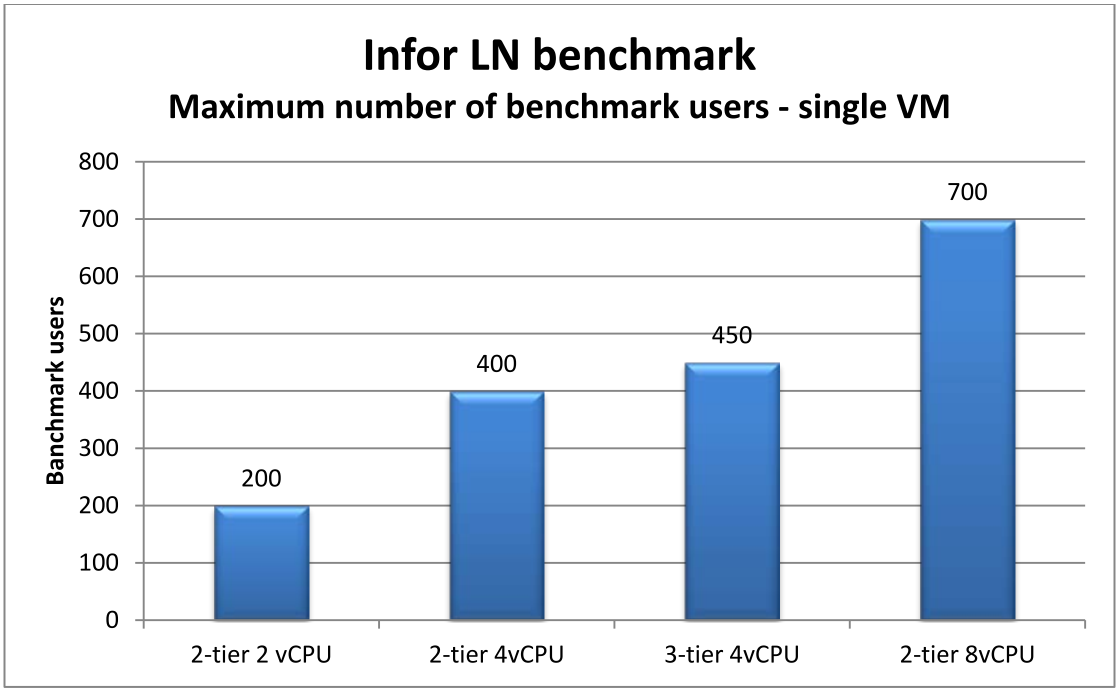


Figure 12: Maximum number of Infor LN benchmark users – SUSE Linux

Benchmark note

This benchmark shows that Citrix XenServer performance scales well. It was not possible to run the 16 vCPU benchmark due to insufficient memory in the server.

Single VM performance and scalability

The graph below shows the single VM performance and scaling in XenServer. The used workload is an Infor LN batch which takes, on average, one full CPU core.

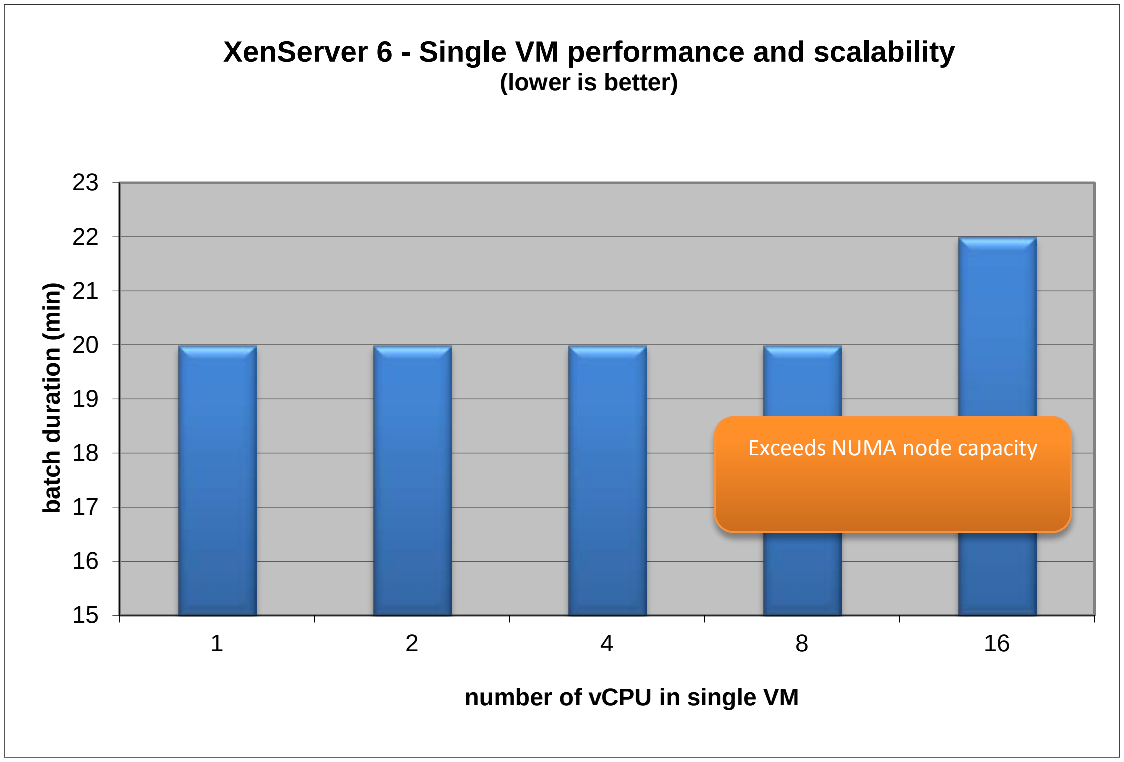


Figure 13: Single VM performance and scalability in Citrix XenServer 6

Benchmark note

The graph shows that the batch performance does not change up to 8 vCPU. With the 16vCPU VM the batch duration increased with 10%.

The 8vCPU and 16vCPU exceed the NUMA node capacity of 4 cores. There is no performance impact because the CPU scheduler can effectively manage the benchmarked workload (one full CPU core) on the capacity of a single NUMA node. There might be a performance penalty in case of larger workloads.

Multiple VM

The graph below shows the multiple VM performance and scaling in XenServer. The used workload is an Infor LN batch which takes, on average, one full CPU thread. This workload is run in each VM participating in the test. Each VM is configured with 2 vCPU.

To force the over-commitment of the system, the VMs were affinitized to a single Quad Core CPU (1 CPU, 4 cores and 8 logical threads).

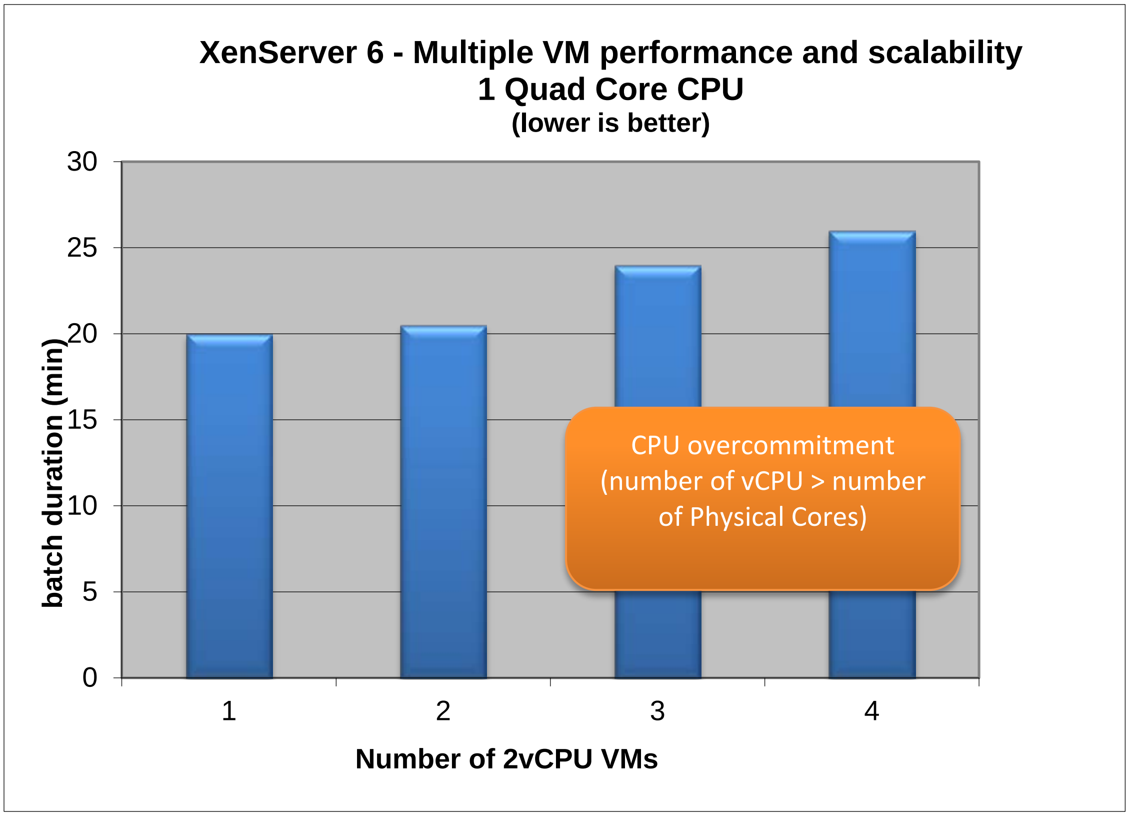


Figure 14: Multiple VM performance and scalability in Citrix XenServer 6

Benchmark note

The test shows that overcommitting CPU decreases performance in all VMs. In the 3 VM test the batch duration increases with 20%; in the 4 VM test the batch duration increases with 30%.

Chapter 5 Microsoft Hyper-V R2

5

This chapter contains specific information about the Microsoft Hyper-V virtualization solution.

Hyper-V architecture

Hyper-V features a hypervisor-based architecture, which is shown in the figure below. The hypervisor virtualizes processors and memory, and provides mechanisms for the virtualization stack in the root partition to manage child partitions (VMs) and expose services such as I/O devices to the VMs.

The root partition owns and has direct access to the physical I/O devices. The virtualization stack in the root partition provides a memory manager for VMs, management APIs, and virtualized I/O devices. The virtualization stack implement emulated devices such as Integrated Device Electronics (IDE), but supports synthetic devices for increased performance and reduced overhead.

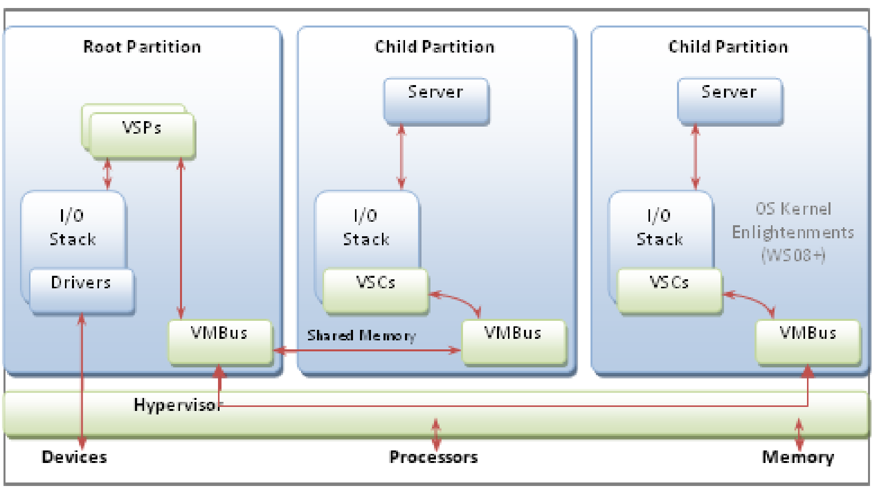


Figure 15: Hyper-V architecture

Integration components

The Microsoft integration components contain drivers and services to maximize the performance of Hyper-V virtual machines (VMs). These integration components include the synthetic device drivers, which give optimal performance and the lowest overhead.

The integration components must be manually installed on each VM. By default, the VM uses emulated devices for IO and networking. You must replace these devices with the synthetic devices that are included in the integration components.

Microsoft Hyper-V R2 versions

To install Microsoft Hyper-V R2, select one of the following options:

- Enable the Hyper-V role in the server manager of a Windows 2008 R2 installation. Infor does not recommend this option, because it installs a full Windows installation next to the VMs. Therefore, this option has the most CPU and memory overhead.
- Install the Windows 2008 R2 server core with the Hyper-V role. This option is a minimal installation of Windows 2008, after which the Hyper-V role can be added. This option has some overhead in CPU and memory.
- Install Microsoft Hyper-V server 2008 R2. Infor recommends this option, because the Hyper-V server is a dedicated stand-alone product, which contains only the Windows Hypervisor, Windows server driver model, and the virtualization components. This option provides a small footprint and minimal overhead.

Live migration

Hyper-V™ live migration is designed to move running VMs and have no effect on VM availability to users. By pre-copying the memory of the migrating VM to the destination physical host, live migration minimizes the transfer time of the VM. A live migration is deterministic, meaning that the administrator, or script, that initiates the live migration can control which computer is the destination for the live migration. The guest operating system in the migrating VM is unaware that the migration is occurring. Therefore, no special configuration for the guest operating system is required.

Live migration is comparable to the VMware vMotion functionality.

Requirements for running live migration on Infor LN environments

Infor did not benchmark Microsoft Hyper-V R2 live migration on Infor LN environments; because it is expected that live migration is comparable to VMware vMotion, which has been benchmarked.

It is expected that the requirements for VMware vMotion, which are given in the “Requirements for running VMware vMotion on Infor LN environments” section of the “VMware vSphere 4” chapter, apply to live migration.

Network settings in a 3-tier configuration

For the Hyper-V 3-tier benchmark, that is, where the Infor LN application and database are on different physical servers, some network changes were required to acquire sufficient networking throughput in the benchmark. Therefore, to acquire sufficient networking throughput, disable TCP Chimney Offloading on the network interface.

For more information refer to http://support.microsoft.com/kb/951037

Chapter 6 Recommendations

6

This chapter contains a summary of the recommendations for using Infor LN in a virtualized environment.

BIOS settings

The recommended BIOS settings for optimal performance are as follows:

- Enable Turbo Mode in the BIOS when available.
- Enable maximum performance or high performance mode in the BIOS.
- Make sure all CPU cores and sockets are enabled in the BIOS.
- Enable “Virtualization support” in the BIOS. Virtualization support for Intel includes Intel VT-x and Intel EPT, and for AMD it includes AMD-V and AMD RVI.
- Enable “Hyper-threading” in the BIOS when available.
- Disable node interleaving to enable the NUMA architecture of the system when available.
- Disable the C1E halt state and any other power-saving mode in the BIOS. Power-saving technologies may reduce the system’s performance in some situations, so consider disabling them when performance considerations outweigh power considerations.
- Disable any unused devices in the BIOS, such as serial ports.

For more information on these settings, refer to the “References” section of the “Introduction to Virtualization” chapter.

Windows settings

- Enable the use of Large Pages in the guest OS. For more information, refer to: http://www.vmware.com/files/pdf/large_pg_performance.pdf.
- Enable power plan “High performance”, to make sure the system delivers the maximum performance possible.

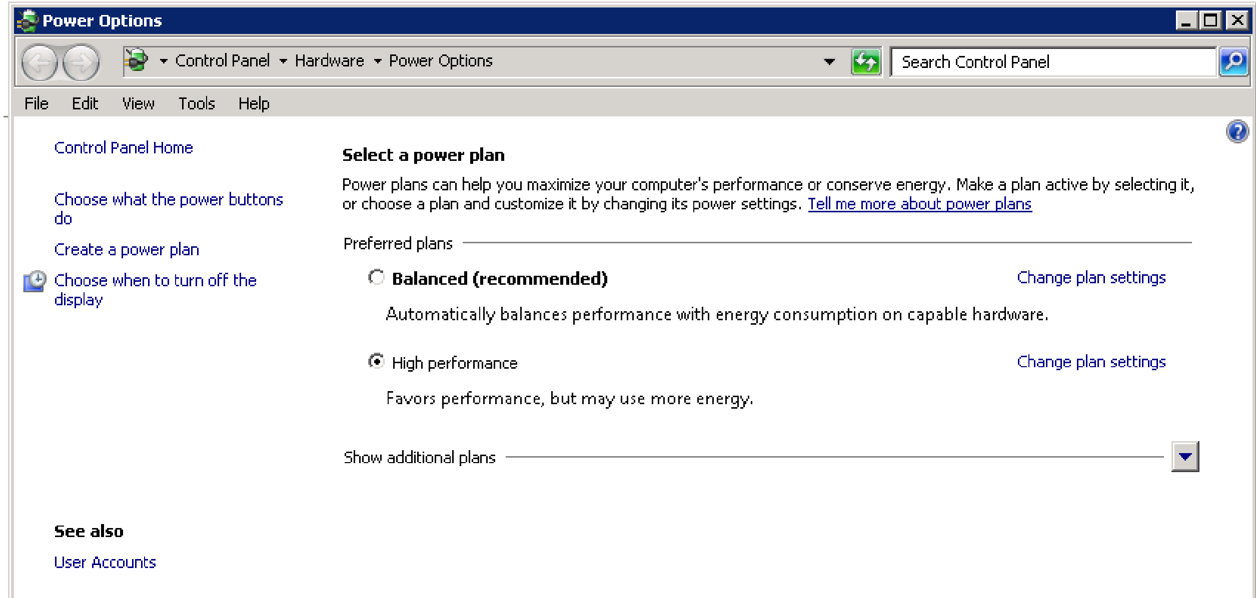


Figure 16: Windows 2008 R2 Power Options

General VM settings

The recommended virtual machine (VM) settings for optimal performance are as follows:

- Disconnect or disable unused hardware devices in the VM, for example COM ports and CD-ROM drives.
- Do not assign more vCPUs to the VM than are required. Assigning more vCPUs than required to a VM can incur a performance penalty. For example, a 4 vCPU VM using on average 60 percent CPU may see a performance drop when configured with eight vCPUs and if there is CPU contention on the system.
- Assign sufficient vCPUs to the VM. A rule of thumb is that a VM should not exceed an average of 80 percent CPU usage. If there is more CPU usage, the number of vCPUs must be increased.
- Be aware of over commitment of the host resources.
- Use CPU and memory reservations for business critical VMs to ensure they obtain the resources they require. Reserve the memory required for the application to avoid performance problems due to insufficient memory (paging). Reserve the CPU capacity required for the application to avoid performance problems due to other VMs taking CPU capacity from the host.
- Use CPU shares, known as relative weight, to allocate resources fairly across VMs.
- If possible do not configure a VM with more vCPUs or memory than available on a single NUMA node. For instance, if your physical server has 6 cores per NUMA node, size your VMs with 2, 3, or 6 vCPUs.
- Do not overload the CPU capacity of the physical host. The host is overloaded if the average CPU usage exceeds 80 percent. By overloading a host, all VMs see a performance drop.

Storage

The storage recommendations for Infor LN VMs and database VMs are as follows:

- Use shared storage solutions like SAN or ISCSI.
- Use pre-allocated or “thick” disks; dynamically growing or “thin” disks are not recommended because of performance reasons.
- Do not use snapshots for VMs with intensive disk IO; for example a database VM. When a snapshot is created, the VMware product produces an additional delta file. Each successive snapshot produces an additional file. When a disk operation is performed within the guest, the disk I/O is recreated by parsing each snapshot delta file in the chain. This produces additional disk overhead on the host because more than one file must be opened and processed to recreate the I/O data for the guest operating system. For best performance, remove all snapshots in the guest operating system or store performance-sensitive data on an independent virtual disk.

Networking

The networking recommendations are as follows:

- For 3-tier configurations, that is, when an Infor LN application and database are on different physical servers, you can improve network communication by using VMQ or enabling TCP offloading.
- In the event of Hyper-V network issues, disable TCP Chimney offloading options on the network adapter.

  See http://support.microsoft.com/kb/951037
- To establish a network connection between two VMs that reside on the same physical system, connect both VMs to the same virtual switch. If the VMs are connected to different virtual switches, traffic will go through the network stack and create unnecessary CPU and network overhead.

VMware vSphere

The VMware recommendations are as follows:

- Use the latest VMware ESX version. Infor LN environments require VMware ESX 3.5 or later. We recommend that you use VMware ESX 5.0 or later.
- Use the latest VM version, version 7 in ESX 4 and version 8 in ESX 5.0.
- Use the latest VMFS version, VMFS 3 in ESX 4 and VMFS 5 in ESX 5.0.
- Install the latest version of the VMware tools in each VM.
- Use the VMXNET3 driver for network adapters, available in ESX 4 and later.
- Disable VMXNET3 virtual interrupt coalescing for the network adapter in 3-tier configurations (separate database server).

- Use the LSI logic driver for direct attached storage.
- Use the Paravirtualized SCSI driver only for high IO workloads (10-50k IOPS), available in ESX 4 and later.
- Use independent persistent disks (pre-allocated) as much as possible. At a minimum, independent persistent disks should be used for the disks hosting the database.

Citrix XenServer

The recommendations for XenServer are:

- Use the latest XenServer version, Infor recommends using version 6.0 or later.
- Install XenServer tools in each VM; the VM properties should show virtualization state optimized.

Hyper-V

The Hyper-V recommendations are as follows:

- Use the latest version of the Microsoft Hyper-V technology.
- Use the standalone hypervisor; at the time of writing this document, the product name is “Microsoft Hyper-V server 2008 R2.”
- Install the latest version of the Microsoft Integration Components in each VM.
- Use synthetic devices instead of emulated devices. Synthetic devices are installed with the integration components. Synthetic devices provide the best performance with the lowest amount of CPU overhead.
- Do not use the default IDE controller, which is an emulated device; use the SCSI controller, which is a synthetic device.
- Use either pass-through disks or fixed VHDs (pre-allocated) for the guest VM storage. These disks and VHDs are the best options to improve performance, and should at least be used for the VM hosting the database. We recommend that you do not use dynamic VHDs because of performance reasons. Note that pass-through disk do not support all Hyper-V functionality.

For 3-tier workloads making heavy use of the network, refer to the networking section.

Chapter 7 Sizing information

7

This chapter contains sizing information for Infor LN in a virtualized environment.

Use the information in this chapter to correctly size a physical server hosting virtual machines (VMs).

Alternatively, use the following rule of thumb: plan to allocate 120 percent of the hardware resources (CPU, Memory, Network, and Storage) required by a physical hardware solution to the VM used for the solution.

By allocating additional resources, you will ensure that it can provide performance on par with physical hardware while accommodating any overhead required by the virtualization technology.

Introduction

Infor has benchmarked the overhead of virtualization solutions for Infor LN. In each benchmark, the virtualized environment is compared to the native environment. The overhead has been measured for OLTP users and for batches. The benchmarked OLTP scenarios are as follows:

- Run Infor LN with a fixed number of benchmark users and investigate the CPU increase of a virtualized environment over a native environment.
- Run the maximum number of Infor LN benchmark users on a native environment and a virtualized environment with the same CPU configuration.

Separate benchmarks have been done to investigate the overhead on Infor LN batches.

Sizing information

This section provides sizing information for CPU, memory, and storage.

CPU overhead

The rule of thumb for sizing a virtualized environment is as follows:

> A virtualized environment running Infor LN takes 20 percent more CPU compared to a native environment.
>
> Or
>
> A virtualized environment has 20 percent less throughput (transactions/hour or concurrent users) compared to a native environment.

Batches

Batch throughput is almost the same in virtualized environments and native environments. Batches in Infor LN always use a full CPU thread in virtualized and non-virtualized environments.

Memory overhead

Virtualization incurs a memory overhead. There is a fixed memory overhead for the hypervisor. Additionally, there is memory overhead for each VM that is running.

VMware ESX and ESXi

The memory overhead of the ESX hypervisor is approximately 200 MB. The service console included with VMware ESX 4 is an additional 300 MB of memory. This service console is not included with the VMware ESXi product. In VMware vSphere5 only the ESXi 5 hypervisor without service console is available.

The overhead of each VM depends on the vCPU and memory configuration, as shown in the following table:

Memory (MB)	1 VCPU	2 VCPUs	3 VCPUs	4 VCPUs	5 VCPUs	6 VCPUs	7 VCPUs	8 VCPUs
256	113.17	159.43	200.53	241.62	293.15	334.27	375.38	416.50
512	116.68	164.96	206.07	247.17	302.75	343.88	385.02	426.15
1024	123.73	176.05	217.18	258.30	322.00	363.17	404.34	445.52
2048	137.81	198.20	239.37	280.53	360.46	401.70	442.94	484.18
4096	165.98	242.51	283.75	324.99	437.37	478.75	520.14	561.52
8192	222.30	331.12	372.52	413.91	591.20	632.86	674.53	716.19
16384	334.96	508.34	550.05	591.76	900.44	942.98	985.52	1028.07
32768	560.27	863.41	906.06	948.71	1515.75	1559.42	1603.09	1646.76
65536	1011.21	1572.29	1616.19	1660.09	2746.38	2792.30	2838.22	2884.14
131072	1912.48	2990.05	3036.46	3082.88	5220.24	5273.18	5326.11	5379.05
262144	3714.99	5830.60	5884.53	5938.46	10142.83	10204.79	10266.74	10328.69

Figure 17: VMware vSphere 5 memory overhead of a virtual machine

For example, a server installed with VMware ESXi and 4 running VMs (each of which has 2vCPUs and 8 GB of memory) has the following memory overhead:

- Hypervisor: 200 MB
- VMs: 4 * 331.12 MB
- Total overhead: 1525 MB

Citrix XenServer memory overhead

The memory overhead of a VM can be determined using the CLI:

```
Xe vm-param-list uuid=<VM UUID> | grep memory-overhead
```

The table below shows the memory overhead of a number of VM configurations on the benchmark server:

vCPU	vMem	VM memory overhead
2	16 GB	132 MB
2	30 GB	240 MB
2	30 GB	473 MB
4	60 GB	475 MB

vCPU	vMem	VM memory overhead
8	60 GB	479 MB

The total memory overhead is determined by memory for the service console (DOM0) and the memory of all running VM's:

Memory overhead = Service console memory + VM memory overhead

Example:

The virtualization memory overhead for a system with 4 VMs with each 2 vCPU and 16 GB internal memory is:

Memory overhead = 752 MB + (4 * 132 MB) = 1.25 GB

Microsoft Hyper-V R2

The approximate memory costs associated with running a guest operating system on a Hyper-V VM are as follows:

- 300 MB for the hypervisor
- 512 MB for the root partition
- 32 MB for the first GB of RAM allocated to each VM
- 8 MB for every additional GB of RAM allocated to each VM

For example, a server installed with Hyper-V and 4 VMs (each of which has 2 vCPUs and 8 GB of memory) has the following memory overhead:

- Hypervisor: 300 MB
- Root partition: 512 MB
- VMs: 4 * (1* 32 MB) + 4 * (7 * 8 MB)
- Total overhead: 1164 MB

Sizing storage

For sizing storage, add together the individual storage requirements of each VM on the same physical server.

For example, running an Infor LN VM and a database VM on the same physical machine will give the following storage requirement: Infor LN storage requirement + database storage requirement.

As a rule of thumb, allocate 120 percent of the disk storage required by the physical environment for the VM. This allocation includes room for some overhead and future growth.

Chapter 8 Performance troubleshooting

8

This chapter provides some basic techniques to troubleshoot a virtualized environment.

Performance tools

To find the resource usage and potential bottlenecks of the server, use a performance tool. These tools include Windows Perfmon, VMware esxtop, vSphere client, XenCenter, and Hyper-V client.

VMware tools include performance counters, which are available in Perfmon in the guest partition:

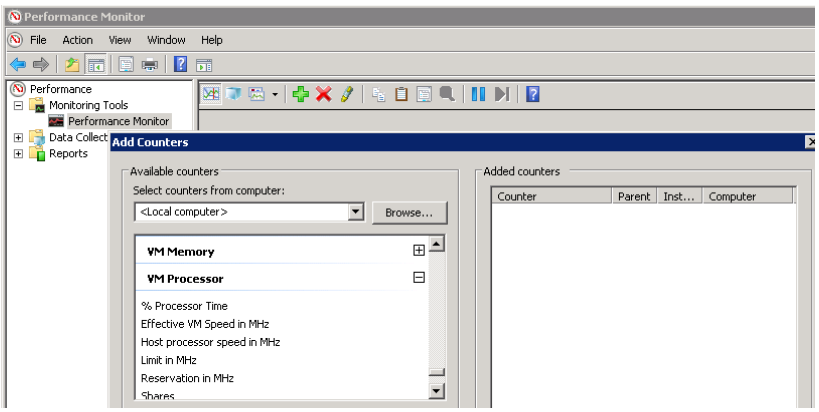


Figure 18: Perfmon counters for VMware available in the guest partition

For Hyper-V, the performance counters are available in Perfmon on the root partition.

CPU

To troubleshoot CPU-related performance issues, complete the following steps:

- Check the CPU load of the virtual machine (VM). If the average VM CPU load exceeds 80 percent, additional vCPUs must be added to the VM.
- Check the CPU load of the physical server. If the average CPU load of the physical server exceeds 80 percent, the server is overloaded. In this case, move VMs to different machines to decrease the load or increase the physical server CPU capacity.
- Use CPU reservations to ensure the VM acquires the necessary minimal CPU capacity.
- Use CPU shares (also named relative weight) to allocate resources fairly across VMs.

Hyper-threaded core-sharing mode

In our benchmarks on the Intel Xeon 5500 series platform, there were no noticeable performance differences by using the hyper-threaded core-sharing modes “Any,” “None,” or “Internal.”

Generally, we recommend that you use CPU shares, which are known as relative weight and reservations to ensure a VM acquires sufficient capacity.

In case of performance issues when running multiple VMs on a hyper-threaded server, change the default core-sharing setting from “Any” to “Internal” or “None” and check for performance gains.

Memory

To troubleshoot memory-related performance issues, complete the following steps:

- A VM should be configured with sufficient memory to hold the working set of the application. If a VM has insufficient memory, it will start paging or swapping.
- Use memory reservations to ensure the VM acquires the necessary memory.

I/O

Follow these steps to troubleshoot IO related performance issues:

- Ensure the total I/O requirement of all running VMs does not exceed the shared storage capacity of the physical server.
- Use fixed-size disks or pass-through disks for IO intensive workloads, for example, databases.

NUMA systems

To troubleshoot NUMA-related performance issues, complete the following steps:

- If possible create a VM configuration that does not exceed the physical CPU and memory capacity of a single NUMA node.
- Ensure that there are no frequent NUMA node migrations. A NUMA node migration happens when a VM is moved from one node to another.
- We recommend that you use affinity on NUMA servers which run multiple memory-intensive and/or CPU-intensive VMs on the same server. With affinity, these VMs can be placed on different NUMA nodes.

Network

To troubleshoot network-related performance issues, complete the following steps:

- Disable interrupt coalescing for the network adapter. This reduces CPU overhead for the virtualization of the network.
- Use a network adapter that supports hardware TCP offloading. Ensure the TCP offloading options are enabled.
- In the event of network performance issues with 3-tier Hyper-V environments, disable TCP Chimney offloading.

  See http://support.microsoft.com/kb/951037.

Single processor in XenServer SUSE Linux VM

When deploying a SUSE Linux VM using the XenServer template the resulting VM may be limited to only 1 CPU core. Check the number of configured processors in your Linux VM by entering:

Cat /xproc/cpuinfo | grep processor

If this command only shows 1 processor, and you have configured the VM with multiple vCPUs, change this setting:

Edit /boot/grub/menu.lst

Remove the “maxcpus=1” entry

Reboot the VM

Appendix A Benchmark server configuration

A

The benchmark results in this document are based on the server configurations shown below.

Physical host running the virtualization solution

- Dell R710
- 2 * Quad Core CPU Intel Xeon E5540 2.53 GHz
- 64 GB of internal memory
- 8 * disk 146 GB SAS, 15k, 3,5" configured in RAID0
- PERC6E SAS Raid Controller, PCIe, 512MB cache

Physical host running the database in 3-tier scenarios

- Dell R610
- 2 * Quad Core CPU Intel Xeon E5540 2.53 GHz
- 32 GB of internal memory
- 8 * disk 146 GB SAS, 15k, 3,5" configured in RAID0
- PERC6E SAS Raid Controller, PCIe, 512MB cache