Pexip Infinity
Server Design Guide

Introduction

This document describes the recommended specifications and deployment for servers hosting the Pexip Infinity platform. It starts with a Summary of recommendations and some Example Conferencing Node server configurations, which are supplemented by further details and explanations in the following Appendices:

- Appendix 1: Detailed server hardware requirements provides a more detailed breakdown of the minimum and recommended hardware requirements for servers hosting the Management Node and Conferencing Nodes respectively.
- Appendix 2: Achieving high density deployments with NUMA provides details on NUMA architecture and how this impacts server architecture and overall performance of Conferencing Nodes.
- Appendix 3: VMware NUMA affinity and hyperthreading is for administrators with advanced VMware knowledge. It explains how to experiment with VMware NUMA affinity and make use of hyperthreading for Pexip Infinity Conferencing Node VMs, in order to achieve up to 50% additional capacity.
- Appendix 4: Hyper-V NUMA affinity and hyperthreading is for administrators with advanced Hyper-V knowledge. It explains how to experiment with Hyper-V NUMA affinity and make use of hyperthreading for Pexip Infinity Conferencing Node VMs, in order to achieve up to 50% additional capacity.

Summary of recommendations

This section summarizes the terminology, recommended specifications and deployment guidelines for servers hosting the Pexip Infinity platform.

Terminology

The table below provides descriptions for the terms used in this guide, in the context of a Pexip Infinity deployment.

Term	Description
Processor	The hardware within a computer that carries out the basic computing functions. It can consist of multiple cores.
Core	One single physical processing unit. Intel Xeon E5 typically has 8 cores (10, 12 or more in newer versions)
Socket	The socket on the motherboard where one processor is installed.
RAM	Also referred to as "memory modules". The hardware that stores data which is accessed by the processor core while executing programs.
Virtual CPU (vCPU)	The VM's understanding of how many CPU cores it requires. Each vCPU appears as a single CPU core to the guest operating system. When configuring a Conferencing Node, you are asked to enter the number of virtual CPUs to assign to it. We recommend no more than one virtual CPU per physical core, unless you are making use of CPUs that support hyperthreading — see NUMA affinity and hyperthreading for more details.
NUMA node	The combination of a processor (consisting of one or more cores) and its associated memory.

Management Node

Recommended host server specifications

- 4 cores* (most modern processors will suffice)
- 4 GB RAM*
- 100 GB SSD storage
- The Pexip Infinity VMs are delivered as VM images (.ova etc.) to be run directly on the hypervisor. No OS should be installed.

* Sufficient for deployments of up to 30 Conferencing Nodes. For larger deployments, you will need to increase the amount of RAM and number of cores. For smaller test and development deployments, 2 cores will suffice. For guidance on Management Node sizing, consult your Pexip authorized support representative or your Pexip Solution Architect.

Conferencing Node

Below are our general recommendations for Conferencing Node (Proxying Edge Nodes and Transcoding Conferencing Nodes) servers. For some specific examples, see Example Conferencing Node server configurations.

Recommended host server specifications

- AVX instruction set required. We recommend Intel Xeon Scalable Processors (Skylake) Gold 61xx generation or E5-2600 v3/v4 Haswell/Broadwell architecture from 2014 or later. Also works with Xeon E5-2600 v1/v2 processors (Sandy Bridge/Ivy Bridge from 2012 or later).
- 2.3 GHz (or faster) clock speed
- 10-12 physical cores per socket
- 1 GB RAM for each vCPU that is allocated to the Conferencing Node
- 4 memory modules per processor socket, with all memory channels populated
- Storage: 50 GB minimum per Conferencing Node; 500 GB total per server (to allow for snapshots etc.)
- RAID 1 mirrored storage
- Hypervisors: VMware ESXi 5.5 or 6.0; KVM; Xen; Microsoft Hyper-V 2012 or later
- The Pexip Infinity VMs are delivered as VM images (.ova etc.) to be run directly on the hypervisor. No OS should be installed.

Transcoding Conferencing Nodes versus Proxying Edge Nodes

The specifications and guidelines shown below apply to Transcoding Conferencing Nodes.

The servers hosting Proxying Edge Nodes do not need to have as high a specification as those servers hosting Transcoding Conferencing Nodes. This is because proxying nodes are not as processor intensive as transcoding nodes. However, you still need multiple proxying nodes for resilience and capacity.

We recommend allocating 4 vCPU and 4 GB RAM to each Proxying Edge Node (which must be dedicated resource).

General deployment recommendations for Transcoding Conferencing Nodes

Cores, CPU and RAM

- Prefer processors with high core count (10 cores or more per CPU).
- Prefer processors with a high clock speed (2.3 GHz and higher).
- Prefer a smaller number of large Conferencing Nodes (e.g. 4 x 10-core nodes), rather than large number of small Conferencing Nodes (e.g. 10 x 4-core nodes).
- A single Conferencing Node must not be assigned more vCPU than the amount of physical cores on each processor socket. (An exception to this rule is when NUMA affinity is enabled.)
- For each physical CPU core (or logical thread, if employing NUMA affinity):
  - configure 1 vCPU
  - assign at least 1 GB RAM

  For example, on an E5-2680v2 CPU with 10 physical cores (i.e. 20 logical threads) per CPU, either
  - assign 10 vCPU (one per physical core) and 10 GB of RAM, or
  - enable NUMA affinity, and assign 20 vCPU (one per logical thread) and 20 GB of RAM
- A Conferencing Node must have 4 vCPU and 4 GB RAM as an absolute minimum.
- Do not over-commit either RAM or CPU resources on hardware hosts. In other words, the Conferencing Node and Management Node each must have dedicated access to their own RAM and CPU cores. Pexip Conferencing Nodes use real-time media, which needs dedicated capacity.
- We recommend 8 memory modules for a dual E5-2600 configuration, as each CPU has 4 memory channels. 8 x 4 GB should be sufficient for most deployments as we recommend 1 GB RAM per vCPU. Some vendors do not provide modules smaller than 8 GB, so in that case we suggest 8 x 8 GB. (This is more than required, but it could be useful if the server is repurposed in the future.)
- Populate memory equally across all NUMA nodes/sockets on a single host server. All memory channels (typically 4 per CPU for E5-2600) must be populated.
- For high performance clusters dedicated to Pexip Infinity, you can achieve 30-50% additional performance by using NUMA affinity and taking advantage of hyperthreading (for CPUs supporting this). For more information, see NUMA affinity and hyperthreading.

Hyperthreading

- Hyperthreading (also referred to as Hyper-Threading Technology), if supported, should always be left enabled by default.

BIOS performance settings

- Ensure all BIOS settings pertaining to power saving are set to maximize performance rather than preserve energy. (Setting these to an energy-preserving or balanced mode may impact transcoding capacity, thus reducing the total number of HD calls that can be provided.) The actual settings depend on the hardware vendor; some examples are given below:

  Typical HP settings
  - HP Power Profile: Maximum Performance
  - Power Regulator modes: HP Static High Performance mode
  - Energy/Performance Bias: Maximum Performance
  - Memory Power Savings Mode: Maximum Performance

  Typical Dell settings
  - System Profile: Performance Optimized

  Typical Cisco UCS B-Series settings
  - System BIOS Profile (Processor Configuration) - CPU Performance: Enterprise
  - System BIOS Profile (Processor Configuration) - Energy Performance: Performance
  - VMware configuration: Active Policy: Balanced

Network

- Although the Conferencing Node server will normally not use more than 1-2 Mbps per video call, we recommend 1 Gbps network interface cards or switches to ensure free flow of traffic between Pexip Infinity nodes in the same datacenter. We do not recommend 100 Mbps NIC.
- Redundancy: for hypervisors that support NIC Teaming (including VMware), you can configure two network interfaces for redundancy, connected to redundant switches (if this is available in your datacenter).

Disk

- Pexip Infinity will work with higher speed SAS or Near-Line SAS drives but we recommend SSD drives for the Management Node. For Conferencing Nodes, SSDs are not a requirement, but general VM processes such as snapshots and backups will be faster with SSDs.
- Pexip Infinity can absorb and recover relatively gracefully from short bursts of I/O latency but sustained latency will create problems.
- Deployment on SAN/NAS storage should in most cases work well. Disk access is only required by the operating system and logs, so a normal fair performance is expected.
- Redundancy: For RAID 1 mirroring for disk redundancy, remember to use a RAID controller supported by VMware or your preferred hypervisor. Most vendors can advise which of the RAID controllers they provide are appropriate for your hypervisors.

Power

- Sufficient power to drive the CPUs. The server manufacturer will typically provide guidance on this.
- Redundancy: Dual PSUs.

Example Conferencing Node server configurations

The table below gives some example server configurations for Transcoding Conferencing Nodes, with an estimate of how many HD (720p) connections (or ports) you can expect to achieve with each. Note that the lower figure represents normal configuration, and the upper figure represents what you can expect with NUMA affinity enabled.

Recommended server sizes

For the Pexip Infinity platform, the following server configurations provide maximum performance for cost:

	Capacity (no. of connections)			
	100-132 HD / 200-280 SD	**55-75 HD / 100-150 SD**	**50-60 HD / 100-140 SD**	**40-50 HD / 80-100 SD**
Cores / Generation	40-core Skylake (Xeon Scalable Processors, launched 2017)	28-core Broadwell (E5-2600 v4 generation, launched 2016)	24-core Haswell (E5-2600 v3 generation, launched 2014)	20-core Ivy Bridge (E5-2600 v2 generation, launched 2013)
CPU	2 x Intel Xeon Gold 6148 • 20 core • 2.4 GHz • 27.5 MB cache	2 x Intel E5-2680v4 • 14 core • 2.4 GHz • 35 MB cache	2 x Intel E5-2680v3 • 12 core • 2.5 GHz • 30 MB cache	2 x Intel E5-2680v2 • 10 core • 2.8 GHz • 25 MB cache
RAM	12 x 8 GB (6 RAM modules per CPU)	8 x 8 GB (minimum 4 RAM modules per CPU for max memory bandwidth)		
Network	1 Gbps NIC (we recommend dual NIC for redundancy)			
Storage	• 2 x 50 GB (50 GB minimum per Conferencing Node) • 500 GB total per server (to allow for snapshots etc.) • RAID 1 mirror for redundancy			
Power	We recommend redundant power			
Example 1U rack servers	• HPE ProLiant DL360 Gen10 • Dell R640 • Cisco US-C220 M5 • Lenovo ThinkSystem SR630 • Supermicro 6019P	• HPE ProLiant DL360 Gen9 • Dell R430/R630 • Cisco UCS C220 M4 • IBM x3550 M5 • Supermicro 6018R-WTR	• HPE ProLiant DL360 Gen9 • Dell R430/R630 • Cisco UCS C220 M4 • IBM x3550 M5 • Supermicro 6018R-WTR	• HPE DL360p Gen8 • Dell R620 • Cisco UCS C220 M3 • IBM x3550 M4 • Supermicro 6017R-WRF

Other processor examples

With specifications similar to the above table, using other processors you can expect to achieve the following capacity:

HD connections	Processor	Cores	Speed
45-70	2 x Intel E5-2690v3	12	2.6 GHz
36-55	2 x Intel E5-2660v3	10	2.6 GHz
32-50	2 x Intel E5-2650v3	10	2.3 GHz
36-52	2 x Intel E5-2630v4	10	2.2 GHz
50-88	2 x Intel E5-2695v4	18	2.4 GHz

Appendix 1: Detailed server hardware requirements

Host server hardware requirements

The following table lists the recommended hardware requirements for the Management Node and Conferencing Node (Proxying Edge Nodes and Transcoding Conferencing Nodes) host servers.

	Management Node	Conferencing Node ††
Server manufacturer	Any	Any
Processor make (see also Performance considerations)	Any	We recommend Intel Xeon Scalable Processors (Skylake) Gold 61xx generation or E5-2600 v3/v4 Haswell/Broadwell architecture from 2014 or later. Also works with Xeon E5-2600 v1/v2 processors (Sandy Bridge/Ivy Bridge from 2012 or later).
Processor instruction set	Any	AVX2 (AVX is also supported)‡
Processor architecture	64-bit	64-bit
Processor speed	2.0 GHz	2.3 GHz or faster
No. of physical cores*	4†	10-12 cores per socket
Processor cache	no minimum	20 MB or greater (2.5 MB L3 cache per core)
Total RAM*	4 GB†	1 GB RAM per vCPU, so either: • 1 GB RAM per physical core (if deploying 1 vCPU per core), or • 2 GB RAM per physical core (if using hyperthreading and NUMA affinity to deploy 2 vCPUs per core).
RAM makeup	Any	All channels must be populated with a DIMM, see Memory configuration below. (For example, E5-24xx supports 3 DIMMs per socket, E5-26xx supports 4 DIMMs per socket.)
Hardware allocation	The host server must not be over-committed in terms of either RAM or CPU. In other words, the Management Node and Conferencing Nodes each must have dedicated access to their own RAM and CPU cores.	
Storage space required	100 GB SSD	• 50 GB minimum per Conferencing Node • 500 GB total per server (to allow for snapshots etc.) SSDs are not a requirement, but general VM processes such as snapshots and backups will be faster with SSDs.
GPU	No specific hardware cards or GPUs are required.	
Network	Gigabit Ethernet connectivity from the host server.	
Operating System	The Pexip Infinity VMs are delivered as VM images (.ova etc.) to be run directly on the hypervisor. No OS should be installed.	

	Management Node	Conferencing Node ††
Hypervisor (see also Performance considerations)	• VMware ESXi 4.1**, 5.x or 6.x • Microsoft Hyper-V 2012 or later • Xen 4.2 or later • KVM (Linux kernel 3.10.0 or later, and QEMU 1.5.0 or later)	

* This does not include the processor and RAM requirements of the hypervisor.

** Support for ESXi 4.1 is being deprecated.

† Sufficient for deployments of up to 30 Conferencing Nodes. For larger deployments, you will need to increase the amount of RAM and number of cores. For smaller test and development deployments, 2 cores will suffice. For guidance on Management Node sizing, consult your Pexip authorized support representative or your Pexip Solution Architect.

‡ For VMware platforms, ESXi 6.x is required to make full use of the AVX2 instruction set. Note that AVX or AVX2 is required; older instruction sets are not supported.

†† The servers hosting Proxying Edge Nodes do not need to have as high a specification as those servers hosting Transcoding Conferencing Nodes. This is because proxying nodes are not as processor intensive as transcoding nodes. However, you still need multiple proxying nodes for resilience and capacity. We recommend allocating 4 vCPU and 4 GB RAM to each Proxying Edge Node (which must be dedicated resource).

Capacity

The number of calls (or ports) that can be achieved per server in a Pexip Infinity deployment will depend on a number of things including the specifications of the particular server and the bandwidth of each call.

As a general indication of capacity:

- When deployed on our recommended hardware (Intel Haswell, 10 cores, 2.3 GHz), Pexip Infinity can connect up to two High Definition 720p30 calls per CPU core. This is based on 1.1 GHz per HD call plus 20% headroom. Capacity for higher speeds can be linearly calculated based on these figures.
- The same recommended hardware can connect a higher number of lower-resolution calls per CPU core. For example, up to 20 audio-only AAC-LD calls at 64 kbps.
- Servers that are older, have slower processors, or have fewer CPUs, will have a lower overall capacity. Newer servers with faster processors will have a greater capacity.

Performance considerations

The type of processors and Hypervisors used in your deployment will impact the levels of performance you can achieve. Some known performance considerations are described below.

Intel AVX2 processor instruction set

As from software version 11, Pexip Infinity can make full use of the AVX2 instruction set provided by modern Intel processors. This increases the performance of video encoding and decoding. For VMware platforms, ESXi 6.x is required to enable this optimization.

AMD processors

We have observed during internal testing that use of AMD processors results in a reduction of capacity (measured by ports per core) of around 40% when compared to an identically configured Intel platform. This is because current AMD processors do not execute advanced instruction sets at the same speed as Intel processors.

AMD processors older than 2012 may not perform sufficiently and are not recommended for use with the Pexip Infinity platform.

VMware ESXi 4.1

We have observed during internal testing that use of VMware's ESXi 4.1 hypervisor may result in a reduction of performance of approximately 20% (as measured in number of ports per physical core allocated to the Conferencing Node) when compared to VMware ESXi 5.x. This is due to slower pass through execution of advanced processor instruction sets.

Memory configuration

Memory must be distributed on the different memory channels (i.e. 4 channels per socket on the E5-2600).

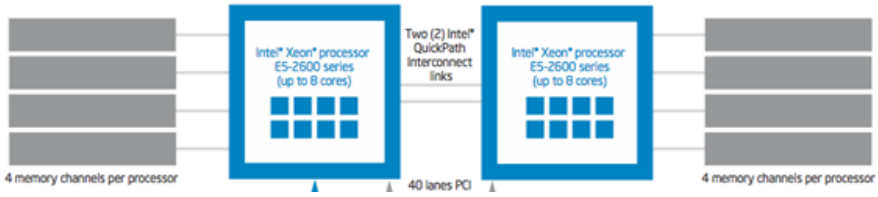
There must be an equal amount of memory per socket, and all sockets must have all memory channels populated (you do not need to populate all slots in a channel, one DIMM per channel is sufficient). Do not, for example, use two large DIMMs rather than four lower-capacity DIMMs - using only two per socket will result in half the memory bandwidth, since the memory interface is designed to read up from four DIMMs at the same time in parallel.

Example

E5-2600 dual socket system:

- Each socket has 4 channels
- All 4 channels must be populated with a DIMM
- Both sockets must have the same configuration

Therefore for a dual socket E5-2600 you need 8 identical memory DIMMs.



Appendix 2: Achieving high density deployments with NUMA

There are many factors that can affect the performance of Virtual Machines (VMs) running on host hardware. One of these is how the VM interacts with NUMA.

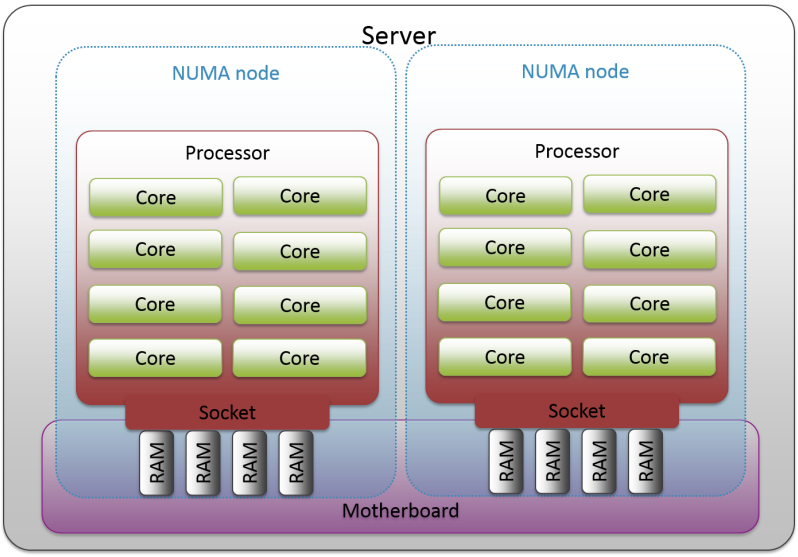
This section provides an overview of NUMA and how it applies to Pexip Infinity Conferencing Nodes. It summarizes our recommendations and suggests best practices for maximizing performance.

About NUMA

NUMA stands for non-uniform memory access.

It is an architecture that divides the computer into a number of nodes, each containing one or more processor cores and associated memory. A core can access its local memory faster than it can access the rest of the memory on that machine. In other words, it can access memory allocated to its own NUMA node faster than it can access memory allocated to another NUMA node on the same machine.

The diagram (right) outlines the physical components of a host server and shows the relationship to each NUMA node.



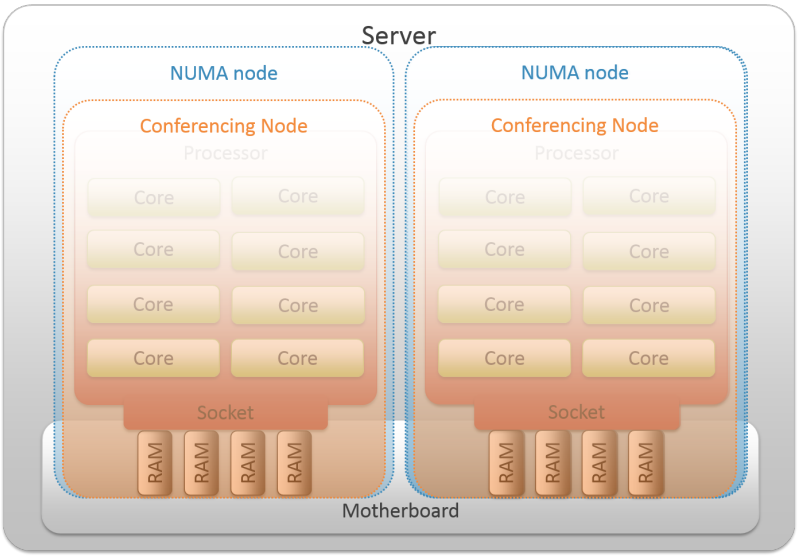
Conferencing Nodes and NUMA nodes

We strongly recommend that a Pexip Infinity Conferencing Node VM is deployed on a single NUMA node in order to avoid the loss of performance incurred when a core accesses memory outside its own node.

In practice, with modern servers, each socket represents a NUMA node. We therefore recommend that:

- one Pexip Infinity Conferencing Node VM is deployed per socket of the host server
- the number of vCPUs that the Conferencing Node VM is configured to use is the same as or less than the number of physical cores available in that socket (unless you are taking advantage of hyperthreading to deploy one vCPU per logical thread - in which case see NUMA affinity and hyperthreading).

This second diagram shows how the components of a Conferencing Node virtual machine relate to the server components and NUMA nodes.



You can deploy smaller Conferencing Nodes over **fewer** cores/threads than are available in a single socket, but this will reduce capacity.

Deploying a Conferencing Node over **more** cores (or threads when pinned) than provided by a single socket will cause loss of performance, as and when remote memory is accessed. This must be taken into account when moving Conferencing Node VMs between host servers with different hardware configuration: if an existing VM is moved to a socket that contains fewer cores/threads than the VM is configured to use, the VM will end up spanning two sockets and therefore NUMA nodes, thus impacting performance.

To prevent this occurring, ensure that either:

- you only deploy Conferencing Nodes on servers with a large number of cores per processor
- the number of vCPUs used by each Conferencing Node is the same as (or less than) the number of cores/threads available on each NUMA node of even your smallest hosts.

NUMA affinity and hyperthreading

It is possible to utilize the logical threads of a socket (hyperthreading) to deploy a Conferencing Node VM with two vCPUs per physical core (i.e. one per logical thread), in order to achieve up to 50% additional capacity.

However, if you do this you must ensure that all Conferencing Node VMs are **pinned** to their respective sockets within the hypervisor (also known as NUMA affinity). If you do not, the Conferencing Node VMs will end up spanning multiple NUMA nodes, resulting in a loss of performance.

Affinity does NOT guarantee or reserve resources, it simply forces a VM to use only the socket you define, so mixing Pexip Conferencing Node VMs that are configured with NUMA affinity together with other VMs on the same server is not recommended.

NUMA affinity is not practical in all data center use cases, as it forces a given VM to run on a certain CPU socket (in this example), but is very useful for high-density Pexip deployments with dedicated capacity.

NUMA affinity for Pexip Conferencing Node VMs should only be used if the following conditions apply:

- The server/blade is used for Pexip Conferencing Node VMs only, and the server will have only one Pexip Conferencing Node VM per CPU socket (or two VMs per server in a dual socket CPU e.g. E5-2600 generation).
- vMotion (VMware) or Live Migration (Hyper-V) is NOT used. (Using these may result in having two nodes both locked to a single socket, meaning both will be attempting to access the same processor, with neither using the other processor.)
- You know what you are doing, and you are happy to revert back to the recommended settings, if requested by Pexip support, to investigate any potential issues that may result.

Step-by-step guides

For instructions on how to achieve NUMA pinning (also known as NUMA affinity) for your particular hypervisor, see:

- Appendix 3: VMware NUMA affinity and hyperthreading
- Appendix 4: Hyper-V NUMA affinity and hyperthreading

Summary of deployment recommendations

We are constantly optimizing our use of the host hardware and expect that some of this advice will change in later releases of our product. However our current recommendations are:

- Prefer processors with a high core count.
- Prefer a smaller number of large Conferencing Nodes rather than a larger number of smaller Conferencing Nodes.
- Deploy one Conferencing Node per NUMA node (i.e. per socket).
- Configure **one vCPU per physical core** on that NUMA node (without hyperthreading and NUMA pinning), or **one vCPU per logical thread** (with hyperthreading and all VMs pinned to a socket in the hypervisor).
- Populate memory equally across all NUMA nodes on a single host server.
- Do not over-commit resources on hardware hosts.

Appendix 3: VMware NUMA affinity and hyperthreading

Introduction

This topic explains how to experiment with VMware NUMA affinity and Hyper-Threading Technology for Pexip Infinity Conferencing Node VMs, in order to achieve up to 50% additional capacity.

If you are taking advantage of hyperthreading to deploy two vCPUs per physical core (i.e. one per logical thread), you must first enable NUMA affinity; if you don't, the Conferencing Node VM will end up spanning multiple NUMA nodes, resulting in a loss of performance.

Affinity does NOT guarantee or reserve resources, it simply forces a VM to use only the socket you define, so mixing Pexip Conferencing Node VMs that are configured with NUMA affinity together with other VMs on the same server is not recommended.

NUMA affinity is not practical in all data center use cases, as it forces a given VM to run on a certain CPU socket (in this example), but is very useful for high-density Pexip deployments with dedicated capacity.

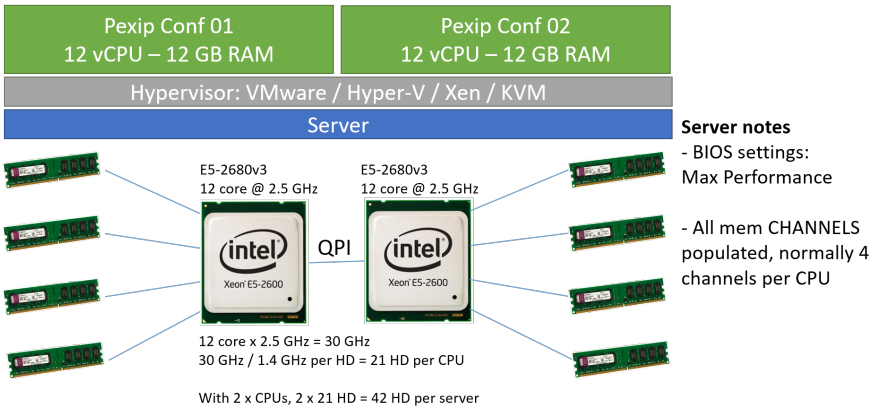
This information is aimed at administrators with a strong understanding of VMware, who have very good control of their VM environment, and who understand the consequences of conducting these changes.

Please ensure you have read and implemented our recommendations in Appendix 2: Achieving high density deployments with NUMA before you continue.

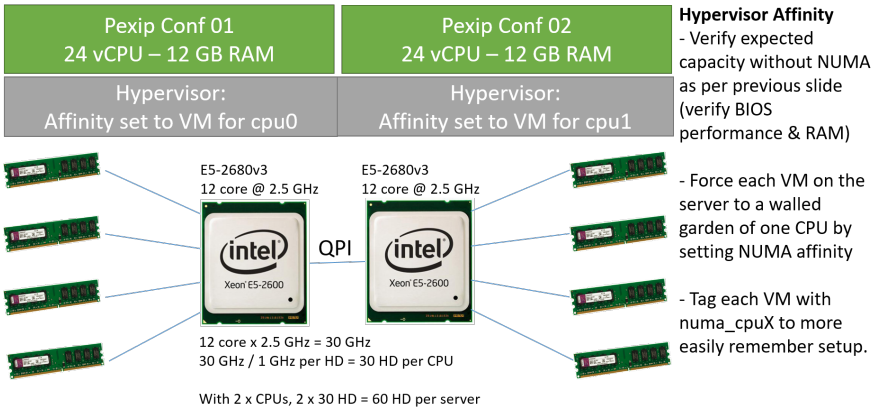
Prerequisites

VMware NUMA affinity for Pexip Conferencing Node VMs should only be used if the following conditions apply:

- The server/blade is used for Pexip Conferencing Node VMs only, and the server will have only one Pexip Conferencing Node VM per CPU socket (or two VMs per server in a dual socket CPU e.g. E5-2600 generation).
- vMotion (VMware) or Live Migration (Hyper-V) is NOT used. (Using these may result in having two nodes both locked to a single socket, meaning both will be attempting to access the same processor, with neither using the other processor.)
- You know what you are doing, and you are happy to revert back to the recommended settings, if requested by Pexip support, to investigate any potential issues that may result.



Example server without NUMA affinity - allows for more mobility of VMs



Example server with NUMA affinity - taking advantage of hyperthreading to gain 30-50% more capacity per server

Overview of process

We will configure the two Conferencing Node VMs (in this example, an E5-2600 CPU with two sockets per server) with the following advanced VMware parameters:

Conferencing Node A locked to Socket 0

- cpuid.coresPerSocket = 1
- numa.vcpu.preferHT = TRUE
- numa.nodeAffinity = 0

Conferencing Node B locked to Socket 1

- cpuid.coresPerSocket = 1
- numa.vcpu.preferHT = TRUE
- numa.nodeAffinity = 1

You must also double-check the flag below to ensure it matches the number of vCPUs in the Conferencing Node:

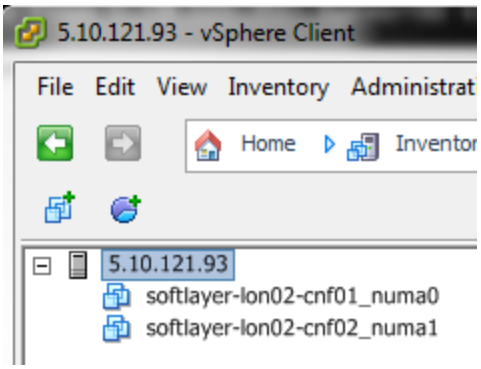
- numa.autosize.vcpu.maxPerVirtualNode

For example, it should be set to 24 if that was the number of vCPUs you assigned.

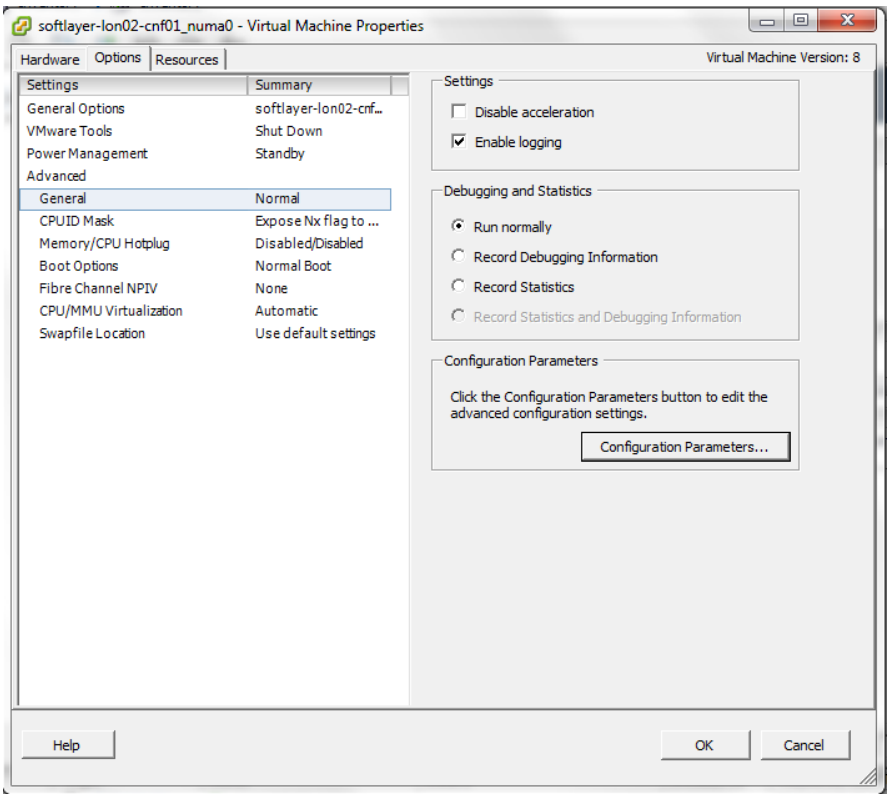
Setting NUMA affinity

Before you start, please consult your local VMware administrator to understand whether this is appropriate in your environment.

1. Shut down the Conferencing Node VMs, to allow you to edit their settings.
2. Give the Conferencing Node VMs names that indicate that they are locked to a given socket (NUMA node). In this screenshot the VM names are suffixed by **numa0** and **numa1**:



3. Right-click the first Conferencing Node VM in the inventory and select **Edit Settings**.
4. From the **Options** tab select **General** and then **Configuration Parameters...**:



5. At the bottom of the window that appears, select **Add Row**.
6. Enter the following data for the first VM, which should be locked to the first socket (**numa0**):
   - cpuid.coresPerSocket = 1
   - numa.vcpu.preferHT = TRUE
   - numa.nodeAffinity = 0

   It should now look like this in the bottom of the parameters list:

cpuid.coresPerSocket	1
numa.vcpu.preferHT	TRUE
numa.nodeAffinity	0

7. Select **OK** and **OK** again.

   Now our **conf-node_numa0** Virtual Machine is locked to **numa0** (the first socket).
8. Repeat the above steps for the second node, entering the following data for the second VM, which should be locked to the

second socket (**numa1**):

- cpuid.coresPerSocket = 1
- numa.vcpu.preferHT = TRUE
- numa.nodeAffinity = 1

It should now look like this in the bottom of the parameters list:

cpuid.coresPerSocket	1
numa.vcpu.preferHT	TRUE
numa.nodeAffinity	1

9. Select **OK** and **OK** again.

   Now our **conf-node_numa1** Virtual Machine is locked to **numa1** (the second socket).

It is very important that you actually set **numa.nodeAffinity** to **1** and not **0** for the second node. If both are set to **0**, you will effectively only use numa node 0, and they will fight for these resources while leaving numa node 1 unused.

Increasing vCPUs

You must now increase the number of vCPUs assigned to your Conferencing Nodes, to make use of the hyperthreaded cores. (Hyperthreading must always be enabled, and is generally enabled by default.)

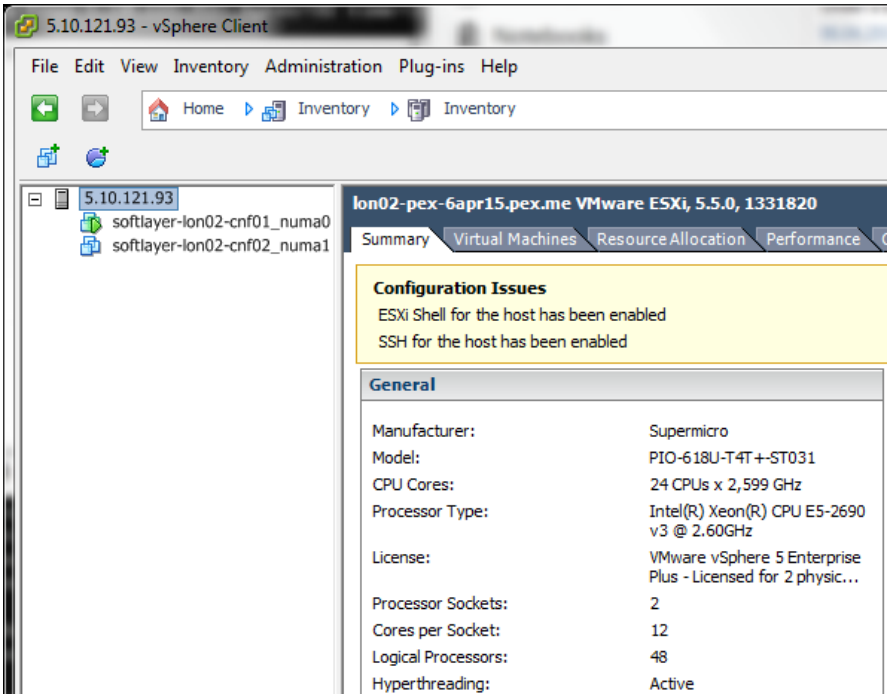
Count logical processors

First you must check how many logical processors each CPU has.

In the example screenshot below, the E5-2690 v3 CPU has **12** physical cores per CPU socket, and there are two CPUs on the server.

With hyperthreading, each physical core has **2** logical processors , so the CPU has **24** logical processors (giving us a total of 48 with both CPUs).

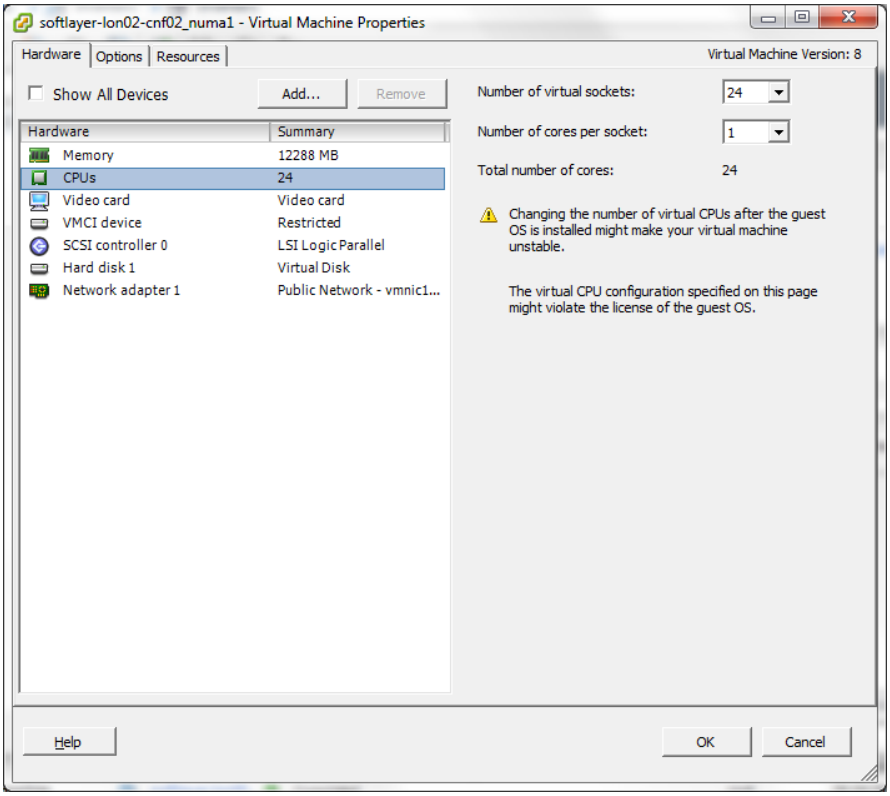
In this case **2 x 12 = 24** is the "magic number" we are looking for with our Conferencing Nodes - which is double the amount of **Cores per Socket**.



Assign vCPU and RAM

Next, you must edit the settings on the Virtual Machines to assign 24 vCPU and 24 GB RAM to each of the two Conferencing Nodes.

Ensure that the server actually has 24 GB of RAM connected to each CPU socket. Since all four memory channels should be populated with one RAM module each, you will normally require 4 x 8 GB per CPU socket.



Reboot

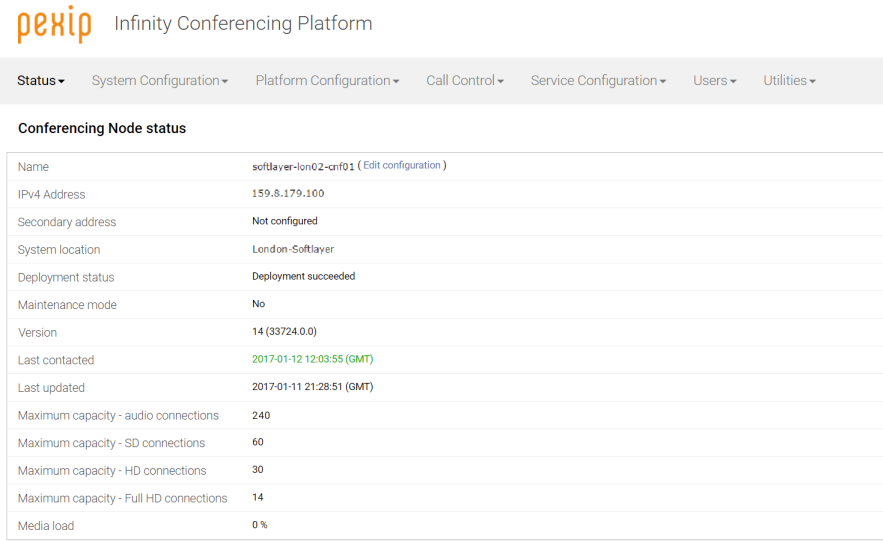
Finally, save and boot up your virtual machines. After about 5 minutes they should be booted, have performed their performance sampling, and be available for calls.

Viewing updated capacity

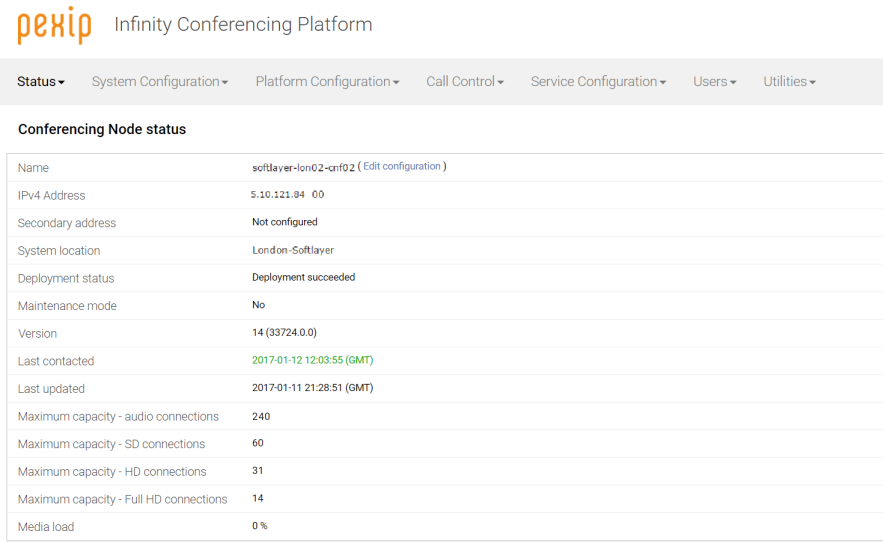
To view the updated capacity of the Conferencing Nodes, log in to the Pexip Management Node, select **Status > Conferencing Nodes** and then select one of the nodes you have just updated. The **Maximum capacity - HD connections** field should now show slightly less than one HD call per GHz (compared to the previous one HD call per 1.41 GHz).

In our example, 12 physical cores x 2.6 GHz = 31.2 GHz, so the Conferencing Node should show around 30 or 31 HD calls, assuming a balanced BIOS power profile. With maximum performance BIOS power profiles, the results could be up to 33-34 HD calls per Conferencing Node VM.

Our first VM:



Our second VM:



Checking for warnings

You should check for warnings by searching the administrator log (**Status > Administrator Log**) for "`sampling`".

A successful run of the above example should return something like:

```
2015-04-05T18:25:40.390+00:00 softlayer-lon02-cnf02 2015-04-05 18:25:40,389 Level="INFO"
Name="administrator.system" Message="Performance sampling finished" Detail="HD=31 SD=60 Audio=240"
```

An unsuccessful run, where VMware has split the Conferencing Node over multiple NUMA nodes, would return the following warning in addition to the result of the performance sampling:

```
2015-04-06T17:42:17.084+00:00 softlayer-lon02-cnf02 2015-04-06 17:42:17,083 Level="WARNING"
Name="administrator.system" Message="Multiple numa nodes detected during sampling" Detail="We
strongly recommend that a Pexip Infinity Conferencing Node is deployed on a single NUMA node"
```

```
2015-04-06T17:42:17.087+00:00 softlayer-lon02-cnf02 2015-04-06 17:42:17,086 Level="INFO"
Name="administrator.system" Message="Performance sampling finished" Detail="HD=21 SD=42 Audio=168"
```

If you have followed the steps in this guide to set NUMA affinity correctly and you are getting the warning above, this could be due to another VMware setting. From VMware, select the Conferencing Node and then select **Edit Settings > Options > General > Configuration Parameters...**). The **numa.autosize.vcpu.maxPerVirtualNode** option should be set to your "magic number". For example, 24 is our "magic number" - the number of logical processors, or vCPUs, assigned in our example.

If this option is set to anything lower, e.g. 8, 10 or 12, VMware will create two virtual NUMA nodes, even if locked on one socket.

VMware and NUMA

As well as the physical restrictions discussed above, the hypervisor can also impose restrictions. VMware provides virtual NUMA nodes on VMs that are configured with more than 8 CPUs. This default value can be altered by setting **numa.vcpu.min** in the VM's configuration file.

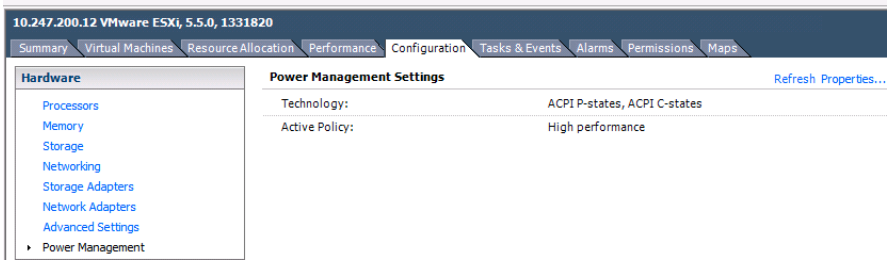
BIOS settings

Ensure all BIOS settings pertaining to power saving are set to maximize performance rather than preserve energy. (Setting these to an energy-preserving or balanced mode may impact transcoding capacity, thus reducing the total number of HD calls that can be provided.) While this setting will use slightly more power, the alternative is to add another server in order to achieve the increase in capacity, and that would in total consume more than one server running in high performance mode.

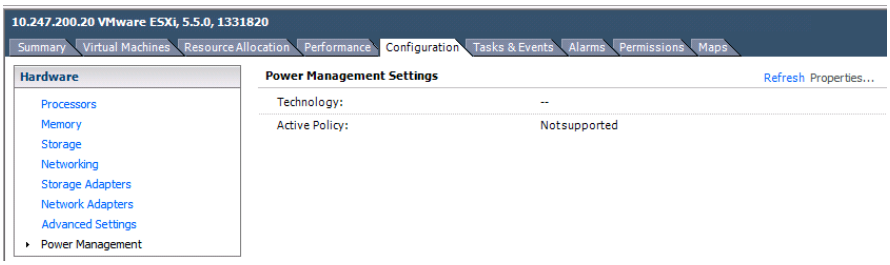
The actual settings will depend on the hardware vendor; see BIOS performance settings for some examples.

A quick way to verify that BIOS has been set appropriately is to check the hardware's **Power Management Settings** in VMware. In most cases, the ACPI C-states should **not** be exposed to VMware when BIOS is correctly set to maximize performance.

If the **ACPI C-states are showing** in VMware (as shown below), the BIOS has most likely **not been set to maximize performance** :



When BIOS has been correctly set to **maximize performance**, it should in most cases look like this:



If your server is set to maximize performance, but VMware still shows ACPI C-states, change it to balanced (or similar), and then change back to maximize performance. This issue has been observed with some Dell servers that were preconfigured with maximize performance, but the setting did not take effect initially.

Appendix 4: Hyper-V NUMA affinity and hyperthreading

Introduction

This topic explains how to experiment with NUMA pinning and Hyper-Threading Technology for Pexip Infinity Conferencing Node VMs, in order to achieve up to 50% additional capacity.

If you are taking advantage of hyperthreading to deploy two vCPUs per physical core (i.e. one per logical thread), you must first enable NUMA affinity; if you don't, the Conferencing Node VM will end up spanning multiple NUMA nodes, resulting in a loss of performance.

Affinity does NOT guarantee or reserve resources, it simply forces a VM to use only the socket you define, so mixing Pexip Conferencing Node VMs that are configured with NUMA affinity together with other VMs on the same server is not recommended.

NUMA affinity is not practical in all data center use cases, as it forces a given VM to run on a certain CPU socket (in this example), but is very useful for high-density Pexip deployments with dedicated capacity.

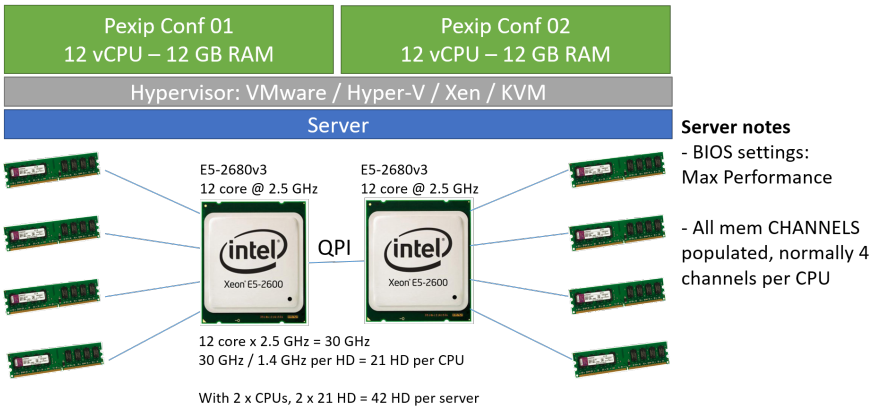
This information is aimed at administrators with a strong understanding of Hyper-V, who have very good control of their VM environment, and who understand the consequences of conducting these changes.

Please ensure you have read and implemented our recommendations in Appendix 2: Achieving high density deployments with NUMA before you continue.

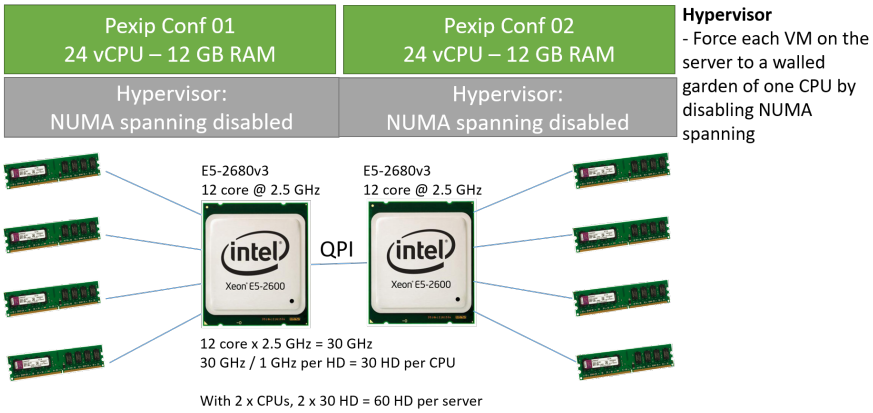
Prerequisites

NUMA affinity for Pexip Conferencing Node VMs should only be used if the following conditions apply:

- The server/blade is used for Pexip Conferencing Node VMs only, and the server will have only one Pexip Conferencing Node VM per CPU socket (or two VMs per server in a dual socket CPU e.g. E5-2600 generation).
- vMotion (VMware) or Live Migration (Hyper-V) is NOT used. (Using these may result in having two nodes both locked to a single socket, meaning both will be attempting to access the same processor, with neither using the other processor.)
- You know what you are doing, and you are happy to revert back to the recommended settings, if requested by Pexip support, to investigate any potential issues that may result.



Example server without NUMA affinity - allows for more mobility of VMs



Example server with NUMA affinity - taking advantage of hyperthreading to gain 30-50% more capacity per server

Example hardware

In the example given below, we are using a SuperMicro SuperServer with dual Intel Xeon E5-2680-v3 processors, 64GB RAM, and 2 x 1TB hard drives.

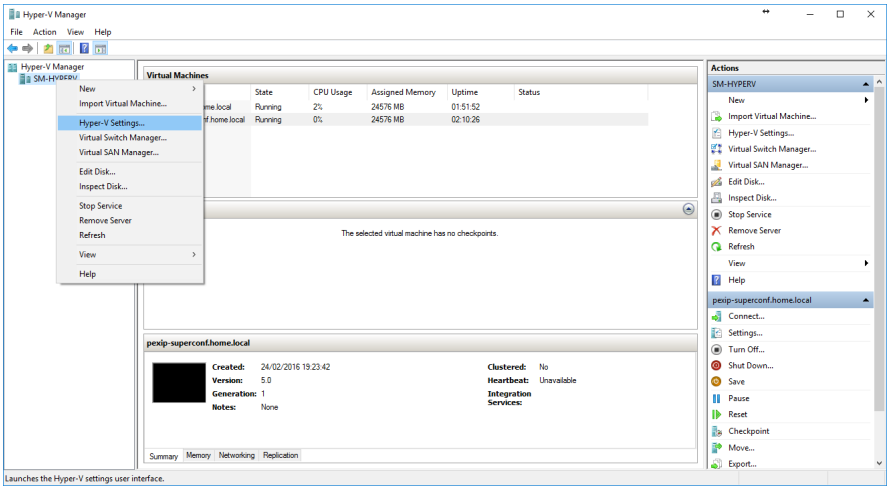
On this server:

- we deploy one Conferencing Node VM per processor/socket, so two Conferencing Nodes in total
- we disable NUMA spanning, so each Conferencing Node VM runs on a single NUMA node/processor/socket
- each processor has 12 physical cores
- we use hyperthreading to deploy 2 vCPUs per physical core
- this gives us 24 vCPUs / 24 threads per Conferencing Node
- therefore we get 48vCPUs / 24 threads in total on the server.

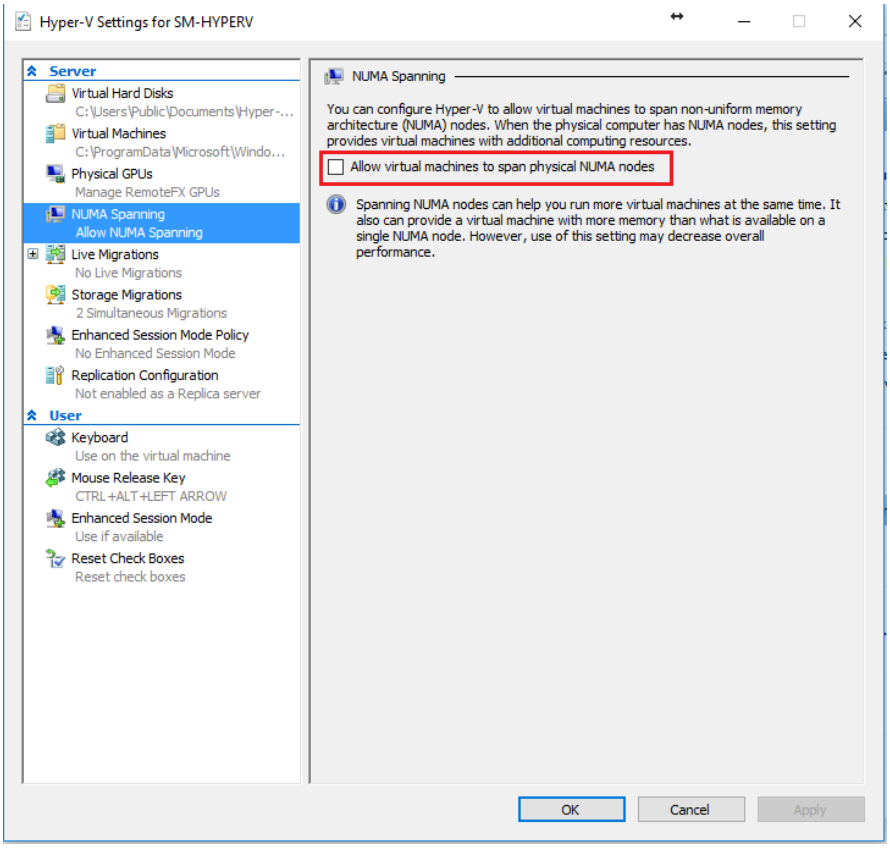
Disabling NUMA spanning on the server

Firstly, we must disable NUMA spanning on the server. To do this:

1. From within Hyper-V Manager, right-click on the server and select **Hyper-V Settings...**:



2. From the Server section, select **NUMA Spanning** and disable **Allow virtual machines to span physical NUMA nodes**. This ensures that all processing will remain on a single processor within the server:

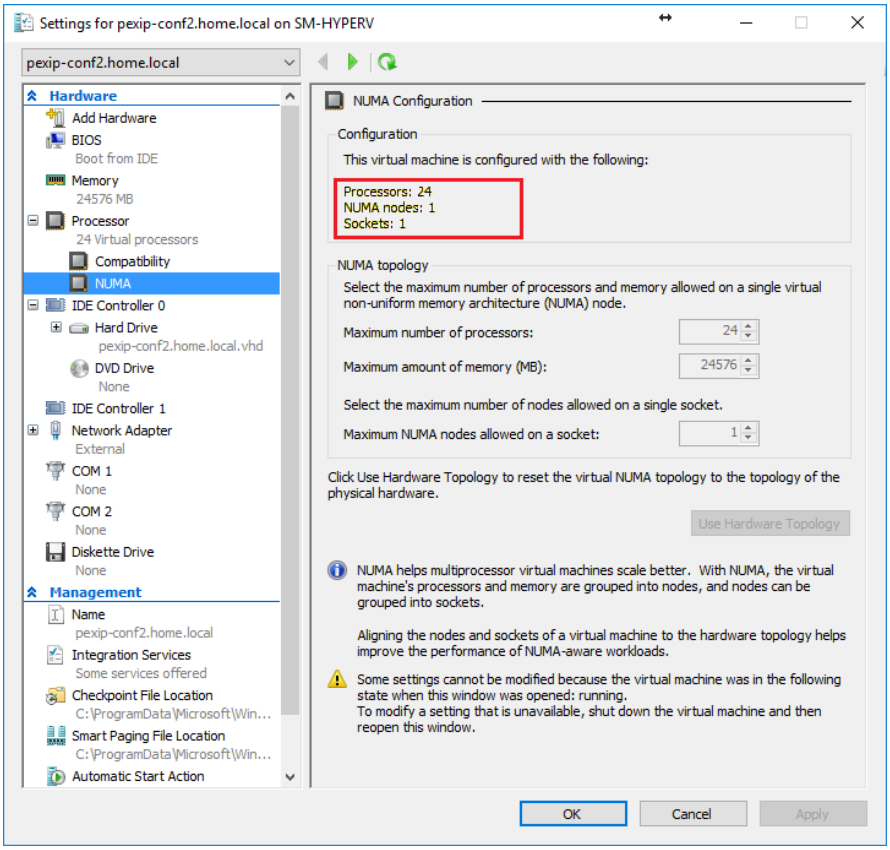


Disable NUMA spanning on the VM

Next we need to ensure the Conferencing Node VMs have the correct settings too, and do not span multiple processors.

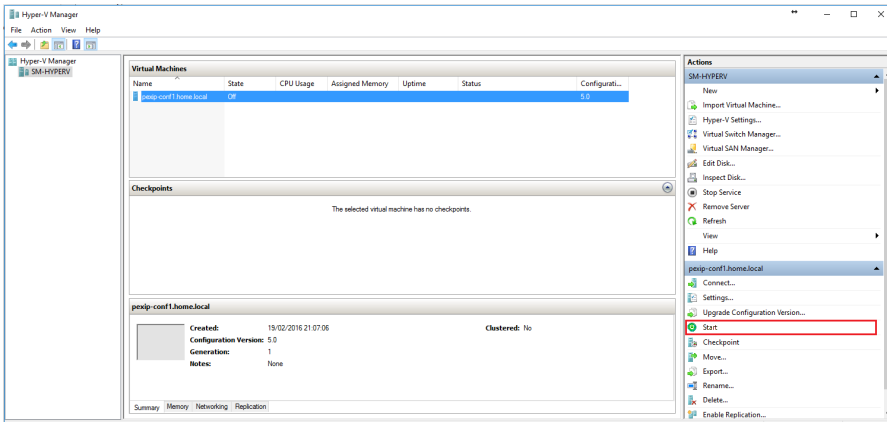
To do this:

1. From within Hyper-V, select the Conferencing Node VM, and then select **Settings > Hardware > Processor > NUMA**.
2. Confirm that only 1 NUMA node and 1 socket are in use by each Conferencing Node VM:



Starting the Virtual Machine

After the NUMA settings have been changed, you can start up each of the Conferencing Node VMs:



Viewing performance and checking for warnings

Every time a Conferencing Node is started up or rebooted, the Pexip Infinity Management Node will perform a sampling of the system to understand what capabilities it has. To view this information, go to the administrator log (**Status > Administrator Log**) and search for "`sampling`".

A successful run of the above example should return something like:

```
2015-04-05T18:25:40.390+00:00 softlayer-lon02-cnf02 2015-04-05 18:25:40,389 Level="INFO"
Name="administrator.system" Message="Performance sampling finished" Detail="FULLHD=17 HD=33 SD=74
Audio=296"
```

An unsuccessful run, where Hyper-V has split the Conferencing Node over multiple NUMA nodes, would return the following warning in addition to the result of the performance sampling:

```
2015-04-06T17:42:17.084+00:00 softlayer-lon02-cnf02 2015-04-06 17:42:17,083 Level="WARNING"
Name="administrator.system" Message="Multiple numa nodes detected during sampling" Detail="We
strongly recommend that a Pexip Infinity Conferencing Node is deployed on a single NUMA node"
```

```
2015-04-06T17:42:17.087+00:00 softlayer-lon02-cnf02 2015-04-06 17:42:17,086 Level="INFO"
Name="administrator.system" Message="Performance sampling finished" Detail="HD=21 SD=42 Audio=168"
```

Moving VMs

When moving Conferencing Node VMs between hosts, you must ensure that the new host has at least the same number of cores. You must also remember to disable NUMA spanning on the new host.

BIOS settings

Ensure all BIOS settings pertaining to power saving are set to maximize performance rather than preserve energy. (Setting these to an energy-preserving or balanced mode may impact transcoding capacity, thus reducing the total number of HD calls that can be provided.) While this setting will use slightly more power, the alternative is to add another server in order to achieve the increase in capacity, and that would in total consume more than one server running in high performance mode.

The actual settings will depend on the hardware vendor; see BIOS performance settings for some examples.