Pexip Infinity

Hardware Resource Allocation Guide

Software Version 23

Document Version 23.a

March 2020

]pexip[

Contents

Introduction

This guide describes how hardware resources are allocated in a Pexip Infinity deployment. It includes the following sections:

- Hardware resource allocation rules: how hardware resources are allocated and consumed in a Pexip Infinity deployment.
- Resource allocation examples: how resources are used in a number of specific scenarios, such as distributed and non-distributed conferences, gateway calls, calls using standards-based endpoints, Skype for Business / Lync clients, or a combination of clients.
- Resource allocation case study: a walk-through of a fictitious deployment that at first resulted in lower than expected capacity, and how this was resolved.

Hardware resource allocation rules

A number of different types of connections to Transcoding Conferencing Nodes are required for a conference to take place, or for a gateway call to be made.

A connection can be a call or presentation from an endpoint to a Virtual Meeting Room or Virtual Auditorium, a backplane between Transcoding Conferencing Nodes, or a call into or out of the Pexip Distributed Gateway. In this context, a **connection** is analogous to a **port**.

When a connection is proxied via a Proxying Edge Node, the proxying node also consumes connection resources in order to forward the media streams on to a Transcoding Conferencing Node. A transcoding node always consumes the same amount of connection resources regardless of whether it has a direct connection to an endpoint, or it is receiving the media streams via a proxying node.

Each connection to a Transcoding Conferencing Node requires a certain amount of resource, or capacity.

In general, when compared to a single high definition **HD** 720p call:

- a **Full HD** 1080p call uses twice the resource
- an **SD** standard definition call uses half the resource
- an **audio-only** call uses one twelfth of the resource.

A WebRTC call using the **VP8** codec uses around 20% more resource than H.264, and the **VP9** codec uses around 50% more resource, so VP9 at 720p uses the equivalent of 1.5 HD resources, and VP9 at 1080p uses the equivalent of 3 HD resources.

If you want to limit video calls to specific resolutions (and limit the transcoding node resources that are reserved for calls), you should use the **Maximum call quality** setting.

The following rules determine how hardware resources are allocated and consumed by conference and gateway calls in a Pexip Infinity deployment. They are measured in terms of HD resources, and assume that the **Maximum call quality** of the conference is set to HD (more or less resources are used for Full HD and SD respectively, as described above):

- **Standards-based endpoints** (SIP or H.323) participants:
  - Each participant uses 1 HD resource.
  - They do not require an additional connection resource when sending or receiving presentation. Presentation is sent and received using the call connection.
- **Skype for Business / Lync** participants:
  - Each participant uses 1 HD resource.
  - They each require 1 additional HD resource when sending or receiving presentation.
- **WebRTC VP8/VP9** participants:
  - VP8 participants use 1.2 HD resources and VP9 participants use 1.5 HD resources for main video.
  - They use 0.5 HD resources for sending presentation content. This is always 0.5 HD resources, regardless of the **Maximum call quality** of the conference, the codec, and whether it is sending images, PDFs or screensharing.
  - They use no additional resources to receive normal (still images) presentation, and use 1 additional HD resource (regardless of codec) when receiving full motion presentation.
- If an endpoint is restricted due to bandwidth limitations to a lower resolution than the conference's maximum call quality, the transcoding node will use the appropriate lower level of resource for that endpoint's connection. For example, if it is limited to an SD connection it will use 0.5 HD resources.
- Each conference instance on each Transcoding Conferencing Node reserves a backplane connection at a resource level corresponding to the conference's **Maximum call quality** setting, to allow the conference to become geographically distributed if required. The exceptions to this are:
  - Deployments with a single Conferencing Node. In such cases, no backplanes will ever be required, so capacity is not reserved.
  - Conferences that are audio-only (in other words, where the conference has its **Conference capabilities** set to ***Audio-only***). In such cases, capacity equivalent to one audio connection is reserved for the backplane.
- Only one backplane connection is used for each conference on each Transcoding Conferencing Node, regardless of the number of other transcoding nodes that are involved in the conference. Note that on the Administrator interface, backplanes are displayed as unidirectional, so a single link between two nodes is shown with two backplanes, one in each direction. Likewise, a

conference between three nodes is shown with six backplanes. However, for resource allocation purposes each node would require just one backplane connection.

- Pexip Infinity always tries to optimize **gateway calls**:
  - A gateway call does not reserve resource for a backplane, but will use one if required (for example, if the participants are connected via different Transcoding Conferencing Nodes).
  - If both of the participants in a gateway call are connected to the same node, and at least one of the participants is either a WebRTC VP8/VP9 client or a Skype for Business / Lync client, and there is currently no presentation being sent by either participant, Pexip Infinity reserves a backplane connection on the node to which the participants are connected. This is in case either participant starts sending presentation and that presentation is handled on a different node to the node to which the participants are connected, thus requiring a backplane to be created between the two nodes.
  - For a gateway call to **Google Hangouts Meet**, the connection to Hangouts Meet always uses 1.2 HD resources (it uses VP8) for main video. The resources required for the VTC leg of the connection depend upon the type of endpoint and the **Maximum call quality** setting. If the VTC endpoint starts to present content then an extra 1.2 HD resources are used for the connection from Pexip Infinity to Hangouts Meet. However, no additional resources are required if presentation content is sent from Hangouts Meet.
  - For a gateway call to a **Microsoft Teams** meeting, the connection to Teams uses 1.5 HD of resource if **Maximum call quality** is SD or HD, otherwise it uses 1.5 Full HD resources. The resources required for the VTC leg of the connection depend upon the **Maximum call quality** setting. No additional resources are required for the connection from Pexip Infinity to Teams for presentations to or from the Teams meeting.
  - For a gateway call to a **Skype for Business meeting**, the connection to SfB uses 1 HD of resource for main video and will use another 1 HD of resource if either side starts presenting. The resources required for the VTC leg of the connection depend upon the **Maximum call quality** setting.
- If an API participant is the first participant to join a conference, it will reserve a backplane for the conference.

Proxying Edge Node resource requirements

When a connection is proxied via a Proxying Edge Node, the proxying node also consumes connection resources in order to forward the media streams on to a Transcoding Conferencing Node.

A proxying node uses approximately the equivalent of 3 audio-only resources to proxy a video call (of any resolution), and 1 audio-only resource to proxy an audio call.

Extra information

See Pexip Infinity license installation and usage for full information about how call licenses are consumed.

Resource allocation examples

The examples below are designed to give you an idea of how the Hardware resource allocation rules apply as the capabilities of the Pexip Infinity platform are extended in the various scenarios:

- Virtual Meeting Rooms
- Gateway calls — all clients
- Gateway calls to Skype for Business meetings
- Gateway calls to Hangouts Meet
- Gateway calls to Microsoft Teams

In each of these cases, the hardware capacity requirements are described in terms of the HD **resources** required for a connection. A connection can be a call or presentation from an endpoint to a Virtual Meeting Room or Virtual Auditorium, a backplane between Transcoding Conferencing Nodes, or a call into or out of the Pexip Distributed Gateway. In this context, a **connection** is analogous to a **port**.

All of these examples are based around using the default **Maximum call quality** of **HD**. A **Full HD** connection would use twice the resource of a HD connection and a standard definition SD connection would use half the resource. WebRTC connections are assumed to be using the VP9 codec. VP9 calls consume around 1.5 times the resource for calls for main video (i.e. 1.5 HD resources for 720p, and 3 HD for 1080p).

All of the examples assume that the endpoints are connecting directly to Transcoding Conferencing Nodes. However, the Teams examples also show the effect of routing those calls via a Proxying Edge Node to the Teams Connector. (When a connection is proxied via a Proxying Edge Node, the proxying node also consumes connection resources in order to forward the media streams on to a Transcoding Conferencing Node. A transcoding node always consumes the same amount of connection resources regardless of whether it has a direct connection to an endpoint, or it is receiving the media streams via a proxying node.)

Virtual Meeting Rooms

Single non-distributed VMR

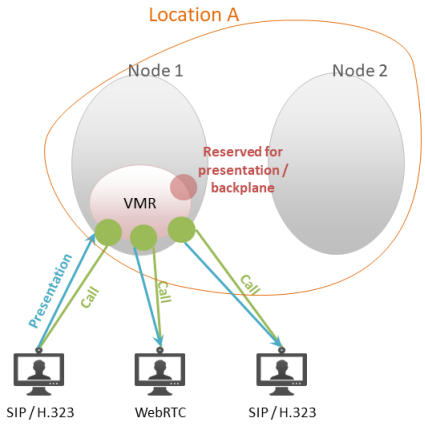
- There are multiple Conferencing Nodes in a single system location, but the conference is hosted on a single node.
- There are three endpoints (two standards-based and one WebRTC VP9*) connecting directly to the node.
- Each endpoint uses HD video and audio.

Pexip Infinity **requires 4.5 HD resources**. 3.5 HD resources are required by the endpoints (the two standards-based endpoints each use 1 HD resource and the WebRTC VP9 client uses 1.5 HD), and 1 HD resource is reserved by the backplane.

- A presentation is then sent from a standards-based endpoint (see diagram).

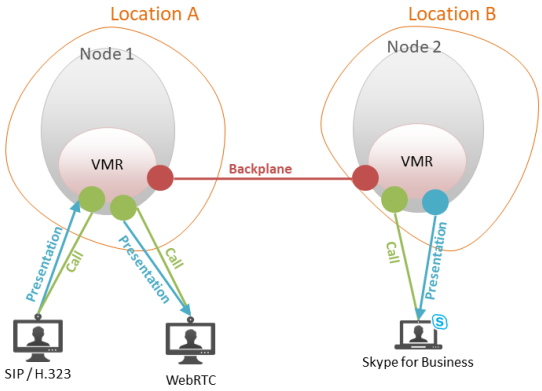
Pexip Infinity **requires no additional resources** (assuming the WebRTC client has not elected to receive full motion presentation).

The conference uses 3 concurrent call licenses.



Single distributed VMR (standards-based, WebRTC and SfB/Lync endpoints)

- There are two Conferencing Nodes in two different system locations.
- One standards-based endpoint and one WebRTC VP9* endpoint connect to the VMR in the Location A node.
- One SfB/Lync endpoint connects to the VMR in the Location B node.
- Each endpoint uses HD video and audio.



Pexip Infinity **requires 3.5 HD resources** on Node 1 (2.5 for the connected participants, and 1 for the backplane), **and 2 HD resources** on Node 2, (1 for the connected participant, and 1 for the backplane).

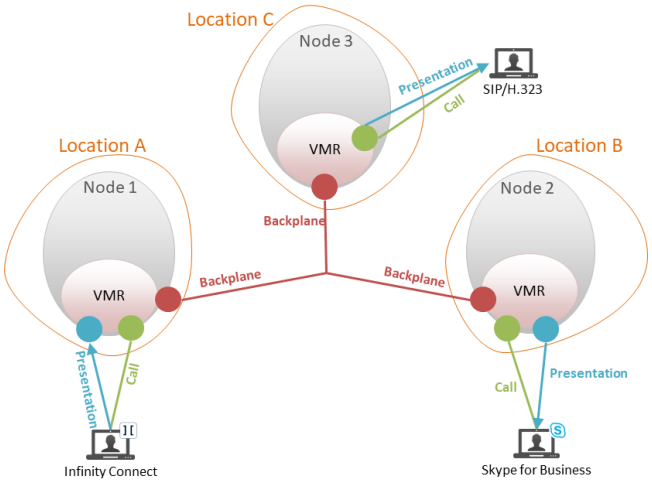
- A presentation is then sent from the standards-based endpoint (see diagram).

Pexip Infinity **requires no additional resources** on Node 1 (assuming the WebRTC client has not elected to receive full motion presentation). But it **requires 1 additional HD resource** on Node 2 for the SfB/Lync client to receive presentation, so the **total resources used on Node 2 is now 3 HD**. The same is true if the SfB/Lync client were to send presentation.

The conference uses 3 concurrent call licenses.

Single distributed VMR — 3 nodes

- There are three Conferencing Nodes in three different system locations.
- One WebRTC VP9* endpoint connects to the VMR in Location A.
- One SfB/Lync endpoint connects to the VMR in Location B.
- One standards-based endpoint connects to the VMR in Location C.
- Each endpoint uses HD video and audio.



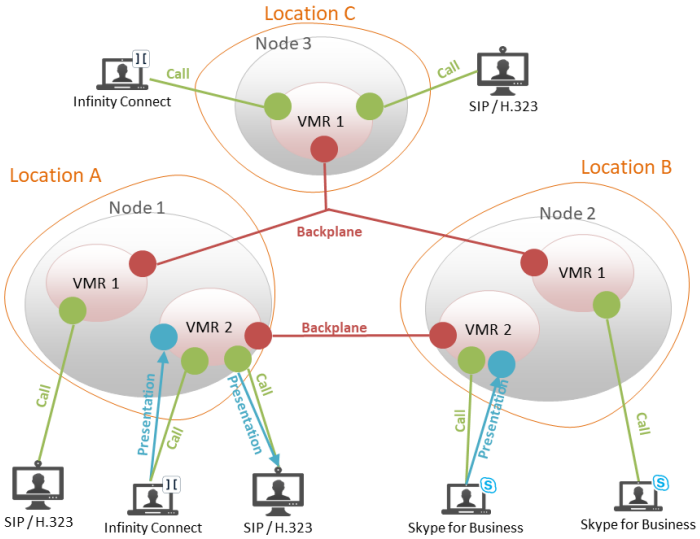
Pexip Infinity **requires 2.5 HD resources** on node 1 (1.5 for the WebRTC participant, and 1 for the backplane) and **2 HD resources** on nodes 2 and 3 (1 for the connected participants, and 1 for the backplane).

- The WebRTC participant then shares their screen (see diagram).

Pexip Infinity **requires an additional 0.5 HD resources** on Node 1 for the WebRTC endpoint to send presentation, and **1 additional HD resource** on Node 2 for the Lync / Skype for Business client to receive presentation, so the **total resources used on Node 1 is now 3 HD and Node 2 is also 3 HD**.

The conference uses 3 concurrent call licenses.

Multiple distributed VMRs



- There are three Conferencing Nodes in three different system locations.
- There are two VMRs currently being used. VMR 1 is being hosted over three nodes, and VMR 2 is being hosted over two nodes.
- A total of seven SfB/Lync, standards-based and WebRTC VP9* endpoints connect to the VMRs over all three nodes, as per the diagram.
- Each endpoint uses HD video and audio.

Pexip Infinity **requires 5.5 HD resources** on node 1, **4 HD resources** on node 2, and **3.5 HD resources** on node 3.

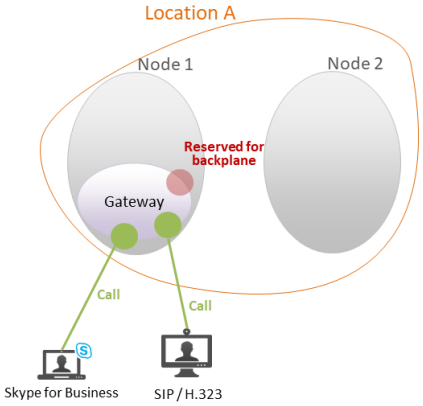
- The WebRTC participant in VMR 2 then shares their screen (see diagram).

Pexip Infinity **requires an additional 0.5 HD resources** on node 1 (for the WebRTC participant to send presentation) and **1 additional HD resource** on node 2 (for the Lync / Skype for Business participant to receive presentation), so the **total resources used on node 1 is now 6 HD** and the **total resources on node 2 is now 5 HD**.

The conferences use 7 concurrent call licenses.

Gateway calls — all clients

Non-distributed gateway call (standards-based endpoint to SfB/Lync client)



- There are multiple Conferencing Nodes in a single system location, but the gateway call is located on a single node.
- There is one standards-based endpoint connecting via the node to a SfB/Lync client.
- Each endpoint uses HD video and audio.

Pexip Infinity **requires 2 HD resources**: 1 for the inbound leg (endpoint to Pexip Infinity) and 1 for the outbound leg (Pexip Infinity to recipient). It also **reserves 1 HD resource** for a presentation backplane (in case either participant starts sending presentation and it is handled on a different node), so a **total of 3 HD resources** are used (see diagram).

- If a presentation is sent from the standards-based endpoint, Pexip Infinity will use the 1 reserved HD resource for the SfB/Lync client to receive presentation.
- If a presentation is sent from the SfB/Lync client, Pexip Infinity will use the 1 reserved HD resource for the SfB/Lync client to send presentation.

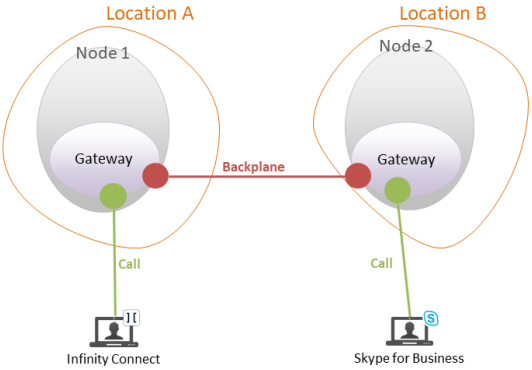
  However, if the RDP or VBSS presentation call from the SfB/Lync client lands on a different Conferencing Node to the video call, then the call will require an additional 2 HD resources (as a backplane is required in each direction between the node handling the video call and the node handling the presentation call, making 5 HD resources in total — 2 video + 2 backplane + 1 presentation).

The conference uses 2 concurrent call licenses in both cases.

Distributed gateway call (WebRTC to SfB/Lync client)

- There are two Conferencing Nodes in two different system locations.
- A gateway call is placed from a SfB/Lync client on node 2, to a WebRTC VP9* Infinity Connect desktop client registered to node 1.
- Each endpoint uses HD video and audio.

Pexip Infinity **requires 4.5 HD resources**: 2.5 HD on node 1 (1.5 for the connected participant and 1 for the backplane), and 2 HD on node 2 (1 for the connected participant and 1 for the backplane).

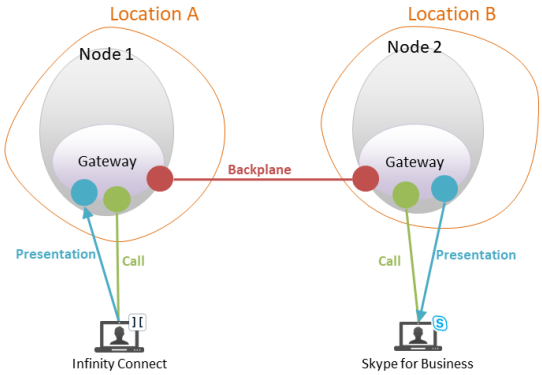


- A presentation (screensharing) is then sent from the Infinity Connect desktop client to the SfB/Lync client.

Pexip Infinity requires **0.5 additional HD resources** for node 1 for the Infinity Connect client to share its screen (a total of **3 HD resources on node 1**). Likewise, **1 additional HD resource** is used on node 2 for the SfB/Lync client to receive presentation (a total of **3 HD resources on node 2**).

If the SfB/Lync client was to send presentation rather than receive, we would still require the same 3 HD resources on node 2, but as the Infinity Connect client does not consume a HD resource to receive content (as still images), we still only use 2.5 HD resources on node 1.

In both cases, the conference uses 2 concurrent call licenses.



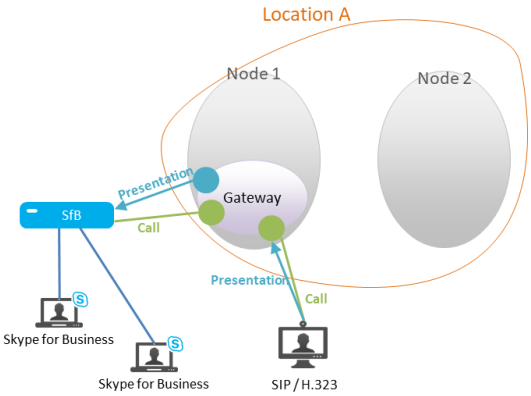
Gateway calls to Skype for Business meetings

Single non-distributed gateway call to SfB/Lync meeting

- There are multiple Conferencing Nodes in a single system location, but the gateway call is located on a single node.
- We have one standards-based endpoint connecting to a SfB/Lync meeting via the node.
- Two SfB clients are also connected directly to the SfB/Lync meeting.
- Each endpoint uses HD video and audio.

Pexip Infinity **requires 2 HD resources**. The standards-based endpoint uses 1 HD resource, and the connection to the SfB/Lync meeting uses 1 HD resource.

- A presentation is then sent from the standards-based endpoint (see diagram).



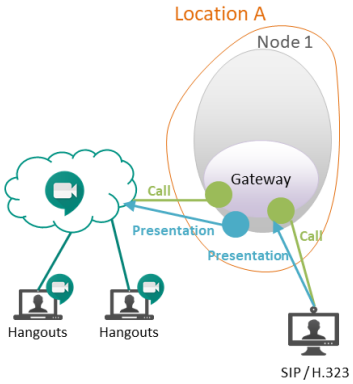
Pexip Infinity **requires 1 additional HD resource** for the SfB/Lync meeting to receive presentation, so a total of 3 HD resources are used. The same is true if a SfB/Lync user were to send a presentation within the SfB/Lync meeting.

The call uses 2 concurrent call licenses in both cases.

Gateway calls to Hangouts Meet

Single non-distributed gateway call (standards-based endpoint to Hangouts)

- We have one standards-based endpoint connecting via a single node in a single location to a Google Hangouts Meet meeting.
- Two Hangouts Meet clients are also connected to the meeting.
- Each endpoint uses HD video and audio.



Pexip Infinity **requires 2.2 HD resources**. The standards-based endpoint uses 1 HD resource. The connection to Hangouts Meet uses 1.2 HD resources (it uses VP8).

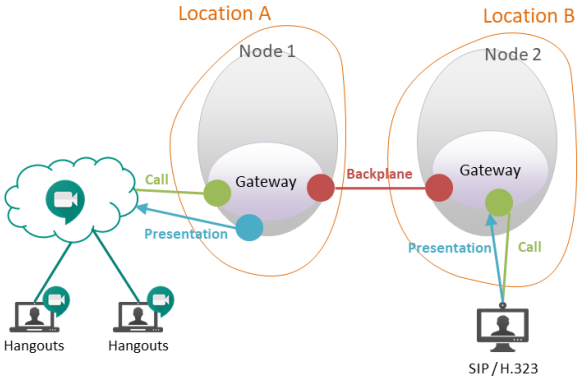
- A presentation is then sent from the standards-based endpoint (see diagram).

Pexip Infinity **requires an additional 1.2 HD resources** for Hangouts Meet to receive presentation. However, no additional resources are required if presentation content is sent from Hangouts Meet.

The call uses 2 concurrent call licenses in both cases.

Single distributed gateway call (standards-based endpoint to Hangouts)

- There are two Conferencing Nodes in two different system locations.
- A gateway call is placed from a standards-based endpoint connected via node 2 to a Google Hangouts Meet meeting connected via node 1.
- Two Hangouts Meet clients are also connected to the meeting.
- Each endpoint uses HD video and audio.



Pexip Infinity **requires 2.2 HD resources** on node 1 (1.2 for the connection to Hangouts Meet and 1 for the backplane), and **2 HD resources** on node 2 (1 for the standards-based endpoint and 1 for the backplane), so **4.2 HD resources in total**.

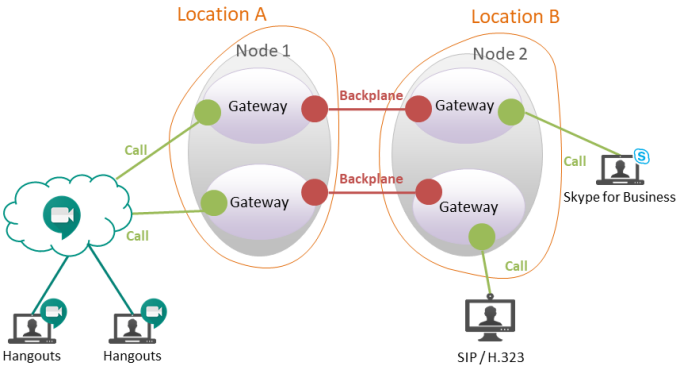
- A presentation is then sent from the standards-based endpoint (see diagram).

Pexip Infinity **requires an additional 1.2 HD resources** for Hangouts Meet to receive presentation. However, no additional resources are required if presentation content is sent from Hangouts Meet.

The call uses 2 concurrent call licenses in both cases.

Multiple distributed gateway calls (mixed endpoints to Hangouts)

- There are two Conferencing Nodes in two different system locations.
- Two gateway calls are placed to the same Google Hangouts Meet meeting connected via node 1: one from a standards-based endpoint connected via node 2, and one from a Skype for Business client also connected via node 2.
- Two Hangouts Meet clients are also connected to the meeting.
- Each endpoint uses HD video and audio.

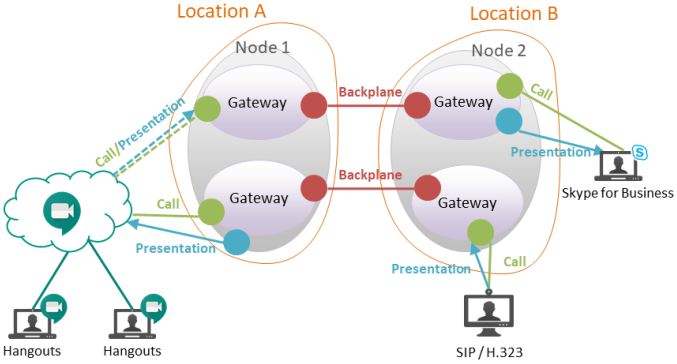


Pexip Infinity **requires 4.4 HD resources** on node 1 (each of the 2 gateway calls requires 1.2 for the connection to Hangouts Meet and 1 for the backplane). It also requires **4 HD resources** on node 2 (the standards-based endpoint requires 1 for its

connection to the gateway, and 1 for the gateway's backplane; the Skype for Business client requires 1 for its connection to the gateway, and 1 for the gateway's backplane), so **8.4 HD resources in total** (see diagram above).

- A presentation is then sent from the standards-based endpoint (see diagram below).

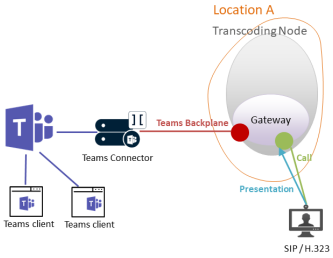
Pexip Infinity **requires an additional 1.2 HD resources** on node 1 to send the presentation to Hangouts Meet, but does not require additional resources on the other connection that is receiving the presentation from Hangouts Meet. The Skype for Business client **requires 1 additional HD resource** on node 2 to receive presentation, so **10.6 HD resources in total** across the two nodes.



Gateway calls to Microsoft Teams

Single non-distributed gateway call (standards-based endpoint to Teams)

- We have one standards-based endpoint connecting via a single transcoding node to a Microsoft Teams meeting via the Teams Connector.
- Two Teams clients are also connected to the meeting.
- Each endpoint uses HD video and audio.



Pexip Infinity **requires 2.5 HD resources**. The standards-based endpoint uses 1 HD resource, and the backplane connection to the Teams Connector uses 1.5 HD resources.

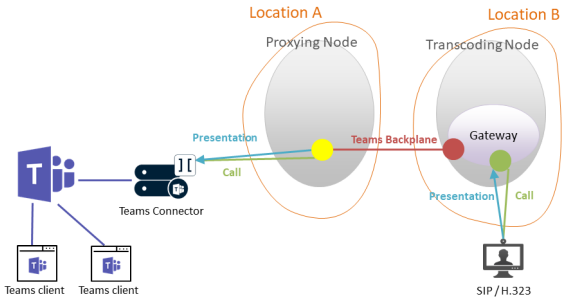
- A presentation is then sent from the standards-based endpoint (see diagram).

Pexip Infinity **requires no additional resources** for the Microsoft Teams meeting to receive presentation. The same would be true if one of the Teams clients sent presentation content.

The call uses 2 concurrent call licenses in both cases.

Single gateway call via Proxying Edge Node (standards-based endpoint to Teams)

- There is a proxying node in location A and a transcoding node in location B.
- A gateway call is placed from a standards-based endpoint connected to the transcoding node to a Microsoft Teams meeting. The call is routed via the proxying node and the Teams Connector.
- Two Teams clients are also connected to the meeting.
- Each endpoint uses HD video and audio.



Pexip Infinity **requires 0.2 HD resources** on the proxying node, and **2.5 HD resources** on the transcoding node (1 for the standards-based endpoint and 1.5 for the backplane connection).

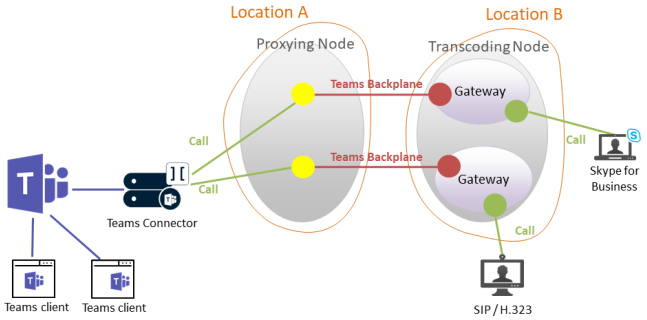
- A presentation is then sent from the standards-based endpoint (see diagram).

Pexip Infinity **requires no additional resources** for the Microsoft Teams meeting to receive presentation. The same would be true if one of the Teams clients sent presentation content.

The call uses 2 concurrent call licenses in both cases.

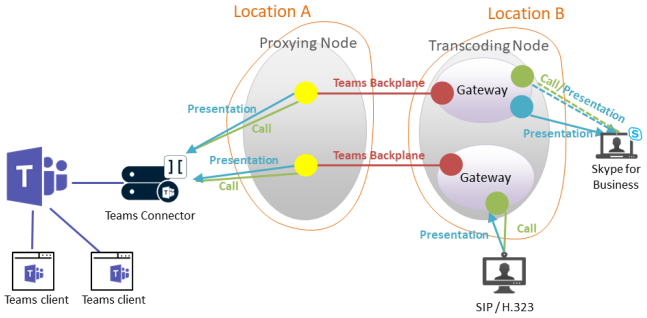
Multiple distributed gateway calls via Proxying Edge Node (mixed endpoints to same Teams meeting)

- There is a proxying node in location A and a transcoding node in location B.
- Two gateway calls are placed from endpoints connected to the transcoding node to a Microsoft Teams meeting. One endpoint is a standards-based endpoint and the other is a Skype for Business client. Both calls are routed via the proxying node and the Teams Connector.
- Two Teams clients are also connected to the Microsoft Teams meeting.
- Each endpoint uses HD video and audio.



Pexip Infinity **requires 0.4 HD resources** on the proxying node and **4 HD resources** on the transcoding node (the standards-based endpoint requires 1 for its connection to the gateway, and 1 for the backplane; the Skype for Business client requires 1 for its connection to the gateway, and 1 for the backplane) (see diagram above).

- A presentation is then sent from the standards-based endpoint (see diagram below).



Pexip Infinity **requires no additional resources** on the proxying node. The Skype for Business client **requires 1 additional HD resource** on the transcoding node to receive presentation.

* WebRTC connections are assumed to be using the VP9 codec. VP9 calls consume around 1.5 times the resource for calls for main video (i.e. 1.5 HD resources for 720p, and 3 HD for 1080p).

Resource allocation case study

This is a fictitious example of a deployment that did not meet initial expectations, the reasons why, and the steps that were taken to rectify the issues that were encountered. This case study takes you through the following steps:

Requirements and initial server/virtualization specification

- Example Corp (our fictitious company) wanted to deploy Pexip Infinity as a proof of concept, with a requirement it would handle **either** a single video conference for 20 users, or up to 150 simultaneous, audio-only calls across 10 different conferences.
- They wanted to use an off the shelf, dual-CPU server - Intel E5-2660 v3 – which has 10 physical cores per socket, and a Processor Base Frequency of 2.6 GHz and 32 GB RAM (consisting of 2 x 16 GB DIMMs).
- As Pexip Infinity is a virtualized platform, Example Corp wanted to run multiple Virtual Machines (VMs) on the same host server, as they do for other VMs in their datacenter. They currently use VMware 5.5 as their virtualization platform.

Pre-deployment

Memory configuration

The memory configuration is not ideal for this server and CPU combination. We say as part of our server design guidelines that a Conferencing Node VM should be deployed with 1 vCPU and 1 GB vRAM per physical core, and we also state that the physical RAM should be deployed across all channels accessed by the physical CPU. From the Intel Ark database we see that the specified E5-2660 v3 CPU has 4 memory channels, so for a dual CPU server, you should populate 8 DIMM slots (consult motherboard/server documentation as to which 8 channels should be populated if more exist), rather than the 2 slots currently occupied.

Pexip Infinity is an application that requires a high amount of compute power, with intensive calculations on data, so requires fast memory access. The more memory channels that a CPU can use, the better overall efficiency of the application.

In this case, assuming that Pexip Conferencing Nodes could utilize all available 20 cores (2 x 10-core CPUs), we would need a minimum of 20GB of physical RAM for the server. Given that DIMM modules usually are sized in 2, 4, 8, 16 etc. GB sticks, and that we need 8 modules (and assuming the same specification module is used in each slot), the ideal memory allocation is 8 x 4GB = 32 GB.

Hardware over-committing

Pexip Infinity does not support over-committing of RAM and CPU resources within a host server, i.e. it does not co-exist well with other VMs running on the same host that may use the resources allocated to Pexip Infinity VMs. Running additional VMs on host cores that are in use by Pexip Infinity results in the hypervisor time-slicing the physical host resources between the VMs. For normal applications (e.g. email, database applications, directory services, file server etc.) that may exist in a virtual environment, the time-slicing by the hypervisor makes no perceivable difference to the running application. However, because Pexip Infinity is a real-time application, this will result in poor performance, such as stuttering video and audio. Pexip Infinity is a very CPU-intensive platform, and the reason for virtualization in our case is more related to hardware abstraction rather than the traditional sharing of resources. Further information can be found in Configuring VMware and Advanced VMware ESXi administration. In particular, if you use vMotion with EVC, this can also cause issues and is also covered in this documentation.

The specified host server contains 20 cores. We must ensure that all the VMs that are running on this one host do not consume more than the 20 cores available. The specified CPU supports hyperthreading (effectively allowing 2 vCPUs per physical core);

however, for Pexip Infinity to make use of the hyperthreading capabilities we need to adjust how the hypervisor works, by ensuring that we lock Conferencing Nodes to exact physical sockets. This is known as NUMA pinning, and is discussed further later on.

It is also important to note that the overall CPU utilization figure of the host server, as reported by the hypervisor, may still seem to be within reasonable tolerance of the entire CPU capacity, even if many VMs are running on the host and they consume more cores than are physically available. However, if this is true, the hypervisor will time-slice these resources, and you will still notice poor performance on Pexip Infinity. As such, if you see a low overall host CPU utilization level, it does not necessarily mean that you are NOT over-committing.

For all of these reasons, we would generally recommend specifying dedicated hosts to run Pexip Conferencing Nodes.

Rule of thumb capacity calculation

Examples of the number of connections (also referred to as ports) you may expect to achieve for various CPU types is given in Example Conferencing Node server configurations. As a rule of thumb, for a normal deployment (not NUMA pinned), you may expect to see the following:

- Utilize approximately 1.4 to 1.5 GHz CPU capacity per HD (720p) connection
- 1 HD connection = approximately 2 SD connections
- 1 HD connection = approximately 12 audio connections
- 1 Full HD (1080p) connection = approximately 2 HD (720p) connection (assuming Full HD is enabled on the deployment)

So, looking at the Intel Ark database for the CPU specified for this server, we can see the core speed of the processor is 2.6 GHz, each CPU contains 10 cores, and there are 2 physical CPU sockets. The calculation we can use to work out an approximate connection (port) capacity is:

$$\left(\frac{\text{Processor base frequency} \times \text{Number of cores}}{\text{CPU capacity per HD connection}}\right) \times \text{Number of CPU sockets}$$

So,

$$\left(\frac{2.6 \times 10}{1.4}\right) \times 2 = 37 \text{ HD connections (rounded down)}$$

Different hardware specifications may result in slightly lower numbers, hence, in our example documentation we have specified 35 HD connections for this CPU. This also assumes Pexip Infinity will consume all cores on the host.

Deployment 1 – making Conferencing Nodes too big

Additional deployment scope

- Example Corp is using the host server (dual CPU, 10 physical cores per socket) to run multiple VMs, although they have read our guidance and understand that they must not over-commit resources.
- They already have 2 standard IT infrastructure VMs consuming 4 cores of host resource each (8 cores in total) running on the host.
- They want to use the remaining 12 cores for Pexip Infinity. Thus, they deployed a single Conferencing Node VM with 12 vCPUs and 10 GB of vRAM.
- On boot, the administrator noted a very low HD and audio connection count (HD = 4, audio = 32).
- In the administrator log, the administrator noted the following entry:

```
Message="Multiple numa nodes detected during sampling" Detail="We strongly recommend that a Pexip Infinity Conferencing Node is deployed on a single NUMA node”
```

Understanding the deployment

Example Corp have used our calculation (with some mathematical transposition) to show that 12 of the 20 available cores could achieve at least the 20 HD connections and 150 audio connections required for the PoC (with a small amount of additional capacity).

$$\left(\frac{2.6 \times 12}{1.4}\right) = 22 \text{ HD connections (rounded down)}$$

And as they have seen that approximately 1 HD connection = 8 audio connections they have assumed they should get:

22 x 8 = 176 audio connections

So why is the connection count so low?

At first glance from the deployment notes above, it may seem that Example Corp has simply failed to follow our memory guidelines and has not allocated the full 12 GB of RAM to a Conferencing Node totaling 12 vCPUs. However, there is more to it than that.

The Pexip Conferencing Node is now using 10 cores on one socket, and 2 cores on the other. It may seem logical to simply increase the number of vCPUs assigned to a Conferencing Node in order to achieve more computational power and thus higher connection capacity. However, when the number of vCPUs on a node increases beyond the number of physical cores in a socket (for a normal deployment), the Conferencing Node VM is then hosted on two different physical CPUs and requires access to different banks of memory. This actually makes things quite inefficient, and results in poor connection capacity. NUMA nodes are described in more detail in Achieving high density deployments with NUMA. The warning log entry in the administrator log shows that the Conferencing Node has spanned NUMA nodes; this must be rectified, and the following examples show how this was done.

Deployment 2 – making Conferencing Nodes too small

Additional deployment scope

- Example Corp is using the host server (dual CPU, 10 physical cores per socket) to run multiple VMs, although they have read our guidance and understand that they must not over-commit resources.
- They are now only utilizing 5 cores on the host with their standard IT infrastructure VMs.
- Not wanting to make the same mistake as previously by spanning NUMA nodes, they decided to create smaller Conferencing Nodes to utilize the remaining 15 cores available on the host, so they deployed 3 Conferencing Node VMs with 5 vCPUs and 5 GB of vRAM each.
- On boot, the administrator saw reasonable HD and audio connection counts on each of the nodes:
  - Node 1: HD = 8, audio = 68
  - Node 2: HD = 7, audio = 65
  - Node 3: HD = 7, audio = 65
  - Total capacity: 22 HD, 198 audio

Understanding the deployment

The capacity figures are reasonable, if just a little lower than Example Corp hoped for. In this case, the Example Corp administrator had not tuned the BIOS settings on the server, but left them as the defaults, and these were configured by the manufacturer with energy saving in mind. Changing the relevant power setting to Maximum Performance and no power saving should further enhance the Pexip Infinity connection count.

However, given that the minimum requirements were met, Example Corp continued with their testing. Still, they were disappointed to find that they could not get all 20 users with HD video into a single videoconference. When the 20th user attempted to join the conference, they were disconnected with an announcement saying that capacity had been exceeded, and the Example Corp administrator saw the following entry in the administrator log:

```
Message="Alarm raised" Node="192.168.0.1" Name="capacity_exhausted" Time="1453719569.94" Details=""
```

```
Message="Participant failed to join conference." Conference="DB441" Participant="sip:+12345@10.0.0.1;user=phone"
Protocol="SIP" Participant-id="c09ee791-cd60-435e-b1f6-36c24e3dc9fc" Reason="Out of resource"
```

So, what had gone wrong?

To answer this, we need to look back at the resource allocation rules. For a multi-node deployment, each video-based VMR instance hosted on each Conferencing Node will reserve 1 HD of connection resource for a backplane. Because Example Corp had used 3 separate nodes, no single node had the capacity to host the entire conference, and so the conference was distributed among the nodes. Because a VMR instance was initiated on each node, Pexip Infinity immediately reserved 1 HD resource for the backplane for each instance. So, with 1 VMR instance running on each node, we can say:

Total node HD connection count – 1 HD backplane per VMR = Available node HD endpoint connections

Node 1: **8 – 1 = 7**
Node 2: **7 – 1 = 6**
Node 3: **7 – 1 = 6**

Total available endpoint connections = 7 + 6 + 6 = 19

While this was not ideal, Example Corp continued their testing, but this time for the audio conferences. However, they were somewhat puzzled in that they only reached a little over 50% of their requirement of 150 concurrent audio calls across 10 conferences, even though the raw calculation showed that the nodes could handle up to 196 audio calls.

Given that there are multiple calls and conferences occurring at the same time, it is useful to look at the data in a different view. Within Pexip support, we can create a single pivot table that shows the call load across all nodes at any single point in time. The data shows the conferences that are in operation, the number of participants, and the call media type (video, audio or presentation), across all the nodes. In the case of Example Corp and the example data shown below, these calls were all audio, so only audio call types are displayed for each node:

Count of Stream Type			
	Nodes and Media Types		
Conference (capability)	**192.168.0.1 audio**	**192.168.0.2 audio**	**192.168.0.3 audio**
IVR (audio-only)			1
VMR 1 (audio-only)	3		
VMR 2 (main video plus presentation)		6	
VMR 3 (main video plus presentation)			31
VMR 4 (main video plus presentation)	1		
VMR 5 (main video plus presentation)			9
VMR 6 (main video plus presentation)		4	
VMR 7 (main video plus presentation)			2
VMR 8 (main video plus presentation)		4	
VMR 9 (main video plus presentation)	10	3	
VMR 10 (main video plus presentation)	12	1	
Total	**26**	**18**	**43**

At first glance, we can see from the totals that only 26 audio calls were running on Node 1, 18 on Node 2 and 43 on Node 3, for a total of 87 concurrent calls. The audio participants are first connected to a Virtual Reception (IVR); from there they enter the conference ID they wish to join, and Pexip Infinity transfers them to the correct VMR. The user currently in the IVR wished to join VMR 4, however, after they entered the conference ID, they heard the capacity exceeded warning and the administrator saw a similar entry seen previously in the Administration Log.

Why did Example Corp achieve such a poor result?

The pivot table shows some very useful information, but does NOT show all the necessary connections reserved for backplanes for each VMR on each node, so we will need to account for these as well in our capacity calculations. In the resource allocation rules we have seen that each VMR in a multi-node deployment will reserve a connection for use by the backplane. However, we also noted that a Conference Capability (call type) can be defined on each VMR, so we can set a VMR to be audio-only. In this case the backplane will only reserve a single audio connection, rather than an HD connection as is the case with a video VMR.

We have added a label beside each VMR listed in the pivot table above to show how the Example Corp administrator had configured each of the Conference Capabilities for that VMR. You can see that apart from the IVR and VMR 1, all other VMRs have been left with the default Conference Capability of "main video plus presentation". As such, each of these VMRs will reserve an HD video resource, even if all the calls within it are audio-only.

So, in summary two mistakes were made here:

1. The Conferencing Nodes were deployed with a vCPU count that was too small.
2. The VMR Conference Capability type for audio-only VMRs was incorrectly set (i.e. left as the default rather than set to audio-only).

The necessity for reservation of the backplane connection is the reason why we recommend processors with high core count: so that we can deploy Conferencing Nodes with high number of vCPUs (as long as they do NOT span NUMA nodes), rather than a larger number of nodes with a smaller vCPU count. In this way, we can achieve higher concentration of capacity with the same resources. For further information, see our server design guidelines and Handling of media and signaling.

If Example Corp had deployed 2 Conferencing Nodes, one with 5 vCPUs and one with 10 vCPUs, they would have been able to achieve their requirement of creating a single videoconference with 20 HD video participants. In addition, if they had set the Conference Capability of the VMRs that were assigned to be used for audio-only conferences, they would have been able to achieve the audio capacity listed in their requirements.

Deployment 3 – correct deployment

Additional deployment scope

- Example Corp have decided to follow our recommended best practice (as per the server design guidelines) and use all the resources of this host server (dual CPU, 10 physical cores per socket) for the deployment of Conferencing Node VMs.
- They continue to utilize 8 x 4 GB RAM DIMMs (total of 32 GB RAM), thus have populated all 4 memory channels for each of the NUMA nodes (sockets).
- The administrator has set the BIOS performance levels to Maximum and switched off any Energy Saving settings.
- They deploy two Conferencing Nodes, each utilizing 10 vCPUs and 10 GB vRAM. In this way, both Conferencing Nodes will consume the resources of each NUMA node (socket) without spanning.
- On boot, the administrator saw good HD and audio connection counts on each of the nodes:
  - Node 1: HD = 18, audio = 144
  - Node 2: HD = 17, audio = 136
  - Total capacity: 35 HD, 280 audio
- They ensured that the IVR and VMRs that are specifically used for audio-only calls are configured with “Conference Capability” set to audio-only.

Understanding the deployment

The connection capacities are calculated during the boot phase of each Conferencing Node, when they simulate call loads. It is not uncommon for the nodes to give a subtle variance, even when they are running on the same host and configured in the same way.

Example Corp re-ran their tests and confirmed that the hardware could meet their initial requirements, i.e. they could host either:

- a single 20-user videoconference, where each participant was connected using HD video, or
- 150 concurrent audio calls spread across 10 simultaneous conferences.

They noted that the videoconferences were initiated on Node 1. This is because Node 1 had calculated a slightly higher capacity during its boot phase, and media is allocated according to which node has the least load. When additional participants are added to the same conference, Pexip Infinity attempts to keep all media on the same node. Example Corp then noted that as the load on Node 1 increased to its maximum, media for additional participants was allocated to Node 2. This is as per the distributed system design of Pexip Infinity (for more information, see Handling of media and signaling). The audio conferences appeared to be more evenly distributed across the two nodes. This is due to the variation in load on the nodes as new conferences are initiated, remembering that a node with the least load will be used to handle a new conference.

Example Corp were now satisfied that the hardware was sufficient and Pexip Infinity would perform well in their environment. But they wondered if there was even more they could do. After all, they had read in the resource allocation rules that Skype for Business / Lync calls consume additional HD resources if they are either sending or receiving content. In addition, they wondered if the hardware could support both elements of their original requirements concurrently, i.e. host a single 20-person HD videoconference, and the 150 audio calls across 10 separate audio conferences, all at the same time.

Deployment 4 – NUMA pinning to increase capacity

Additional deployment scope

- Example Corp followed our recommended best practice (as per the server design guidelines) and used all the resources of this host server (dual CPU, 10 physical cores per socket) for the deployment of Conferencing Node VMs.
- In addition, they made use of the hyperthreading capability of the Intel E5-2660 v3 CPU, so followed our additional guidance regarding Achieving additional performance with VMware NUMA affinity and hyperthreading.
- They changed the physical memory to utilize 8 x 8 GB RAM DIMMs (total of 64 GB RAM), and thus have populated all 4 memory channels for each of the NUMA nodes (sockets).
- The administrator set the BIOS performance levels to Maximum and switched off any Energy Saving settings, and ensured that hyperthreading is enabled.
- They deployed two Conferencing Nodes, each utilizing 20 vCPUs and 20 GB vRAM. In this way, both Conferencing Nodes will consume the resources of each NUMA node (socket), and because the Conferencing Nodes are now pinned to a NUMA node and make use of hyperthreading, they still do not span NUMA nodes.
- On boot, the administrator saw good HD and audio connection counts on each of the nodes:
  - Node 1: HD = 28, audio = 224
  - Node 2: HD = 27, audio = 216
  - Total capacity: 55 HD, 440 audio
- They ensured that the IVR and VMRs that are specifically used for audio-only calls are configured with “Conference capability” set to audio-only.

Example Corp ran their test again. They now find that they can run both scenarios set out in the initial requirements simultaneously.

However, if all 20 video participants in the VMR were Skype for Business / Lync endpoints (a potentially unlikely scenario), and they consumed both HD video and RDP presentation (perhaps in a multi-screen user environment), this in itself would consume at least 40 HD resources, and the conference would be split across both nodes, hence the VMR instance on each node would also require an HD resource. This would leave approximately 13 HD resources available on Node 2, and based on 1 HD resource = 8 audio resources, we would only have available 104 audio connections. This would not be enough to also support the 150 concurrent audio calls across 10 simultaneous audio conferences. At this point Example Corp would need to add an additional host and nodes to their Pexip Infinity deployment.