Improving VM Placement to Servers to Optimize Your GPU Usage in VMware vSphere 8 Update 2

VMware AI/ML

Improving VM Placement to Servers to Optimize Your GPU Usage in VMware vSphere 8 Update 2

Enterprises are deploying more and more virtualized GPUs (vGPUs) on their VMs to speed up their work on larger machine learning (ML) models and high-performance computing. One clear use case for this is the training and fine-tuning of Large Language Models (LLMs) that make high demands on GPUs.

These larger models often require more than one physical GPU, represented to vSphere as a full memory profile vGPU, to handle their high numbers of parameters and to execute the model training process within a reasonable time. The LLM training process can occupy the GPU for many hours or days.

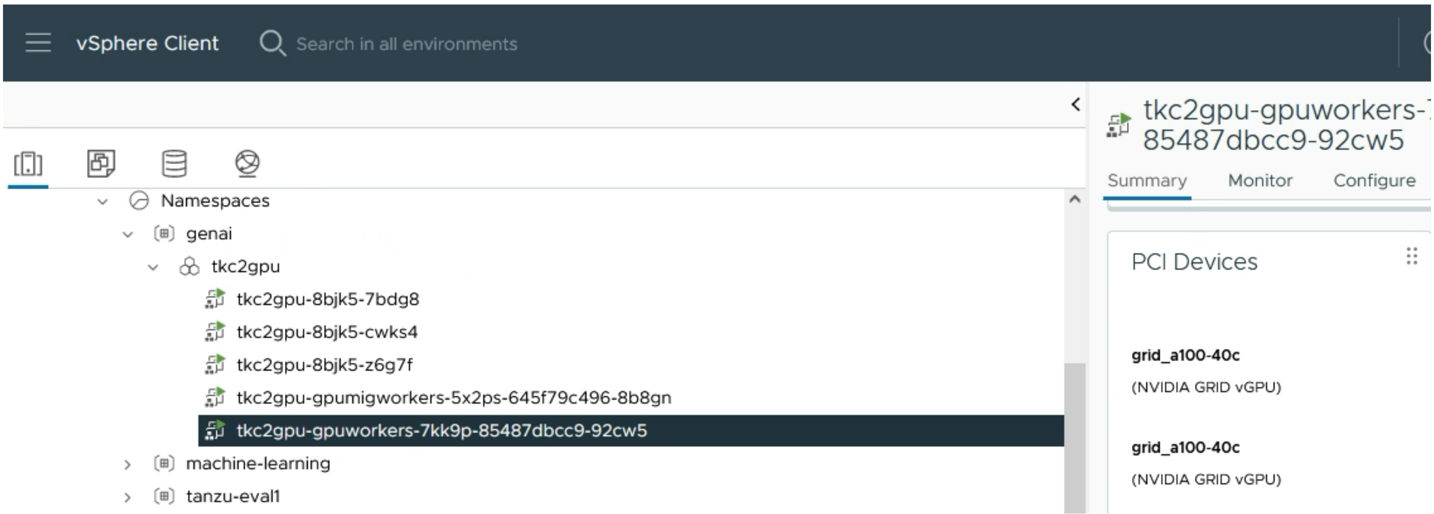

Figure 1: The vSphere Client view of a VM that has two full vGPU profiles consuming all of two A100 40GB GPUs

An example of a full memory vGPU profile is “grid_a100-40c”, indicating that the system administrator chose to give all 40 GB of the GPU’s memory to the VM. You can see two such vGPU profiles on one VM in the above setup. For newer A100 models, this profile would be “grid_a100_80c”, as another example.

As more vGPU power is needed, 2, 4 or 8 physical GPUs may be fully assigned to a VM on a host. Up to eight vGPU profiles can be assigned to a VM in vSphere 8 and in the vSphere 8.0 Update 2 release, this number will increase to 16.

When there are differently-sized VMs sharing a cluster of common host hardware, there can be situations where the placement of a particular size of VM (in terms of its vGPU profile) determines what other VMs would later fit on the same machine or cluster.

The discussion here covers full allocation of one or more **full** physical GPUs to a VM, via a **full memory** vGPU profile. The current implementation of the new feature for VM consolidation does not take partial-memory vGPU profiles into account. The kinds of vGPU profiles that are applicable here are time-slicing full memory allocation profiles (occupying all the memory of one or more GPUs)

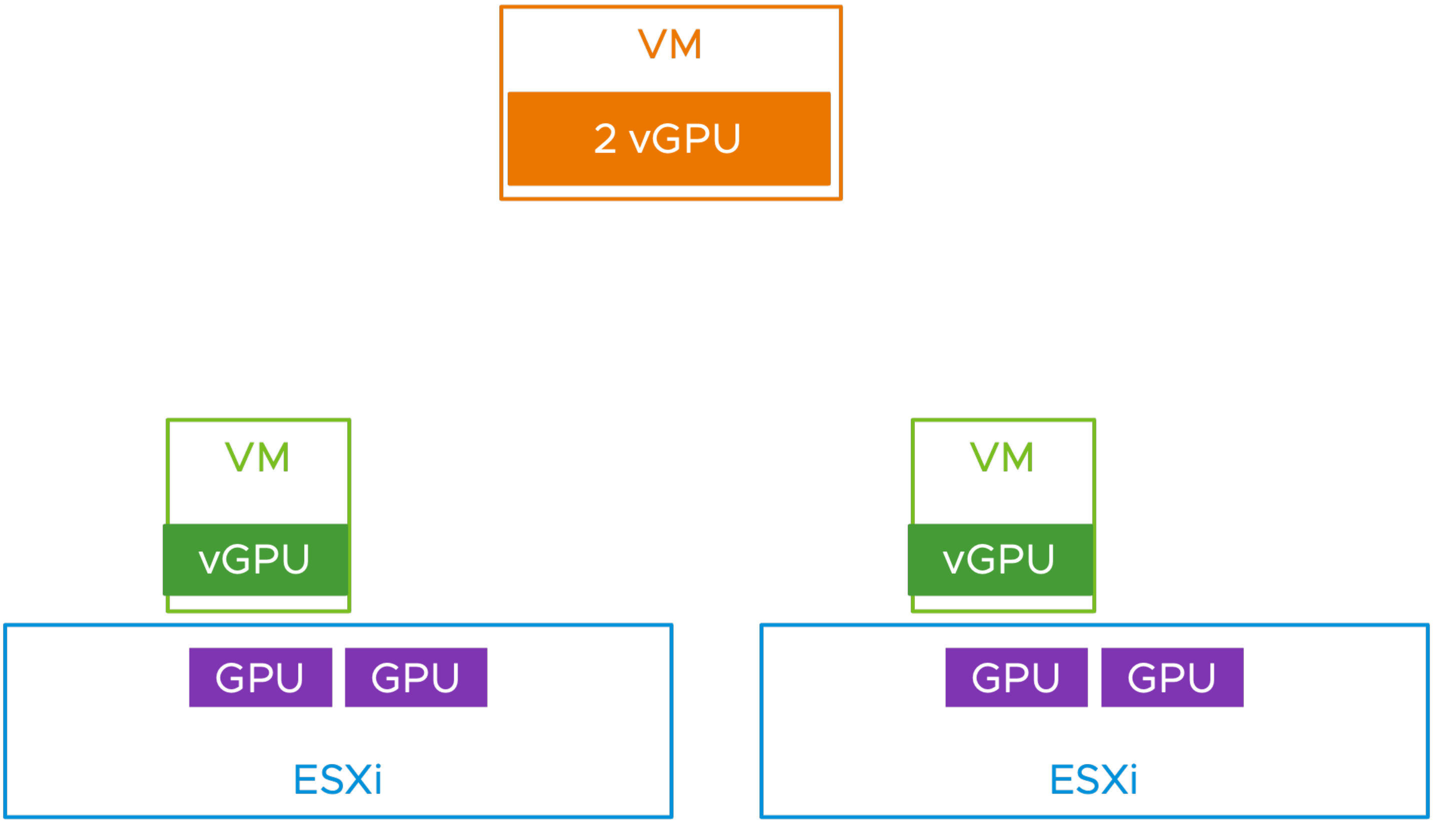

Figure 2 - Two single vGPU VMs obstructing 1 larger 2 vGPU VM from bring placed onto a host server

Let's look at an example scenario. Let’s say you have 2 host servers in a cluster and each host server has two physical GPUs. The GPUs are in time sliced mode, which is a requirement for multiple vGPUs on a VM.

You want some of your VMs to have 1 full memory profile vGPU and other VMs to have 2 full memory profile vGPUs - for differently sized ML or HPC jobs, for example.

Your users create and power on their VMs and the first two VMs happen to be 1 vGPU profile VMs (occupying a full GPU each). These will placed, pre-vSphere 8 Update 2, onto each of your two hosts - effectively dispersing them to each host rather than bin-packing them tightly together.

Now, if you want to provision a new 2-vGPU VM, you have no room for it - although there are two free GPUs available in aggregate across your hosts. We are not making full use of the hardware here. Of course, you can vMotion one of the VMs over, but that is extra work for the system administrator, especially if you have higher numbers of VMs and hosts.

VMware vSphere 8 Update 2 addresses this concern by adding an Advanced Feature to your cluster that causes similar-sized vGPU profile VMs to be held together on one or more hosts. You would set the advanced parameter in the vSphere Client for your cluster as:

VgpuVmConsolidation = 1

With this new consolidation feature, the first two 1-vGPU VMs will be packed onto one host at initial VM placement time. The DRS placement process takes the presence of the vGPU profile and its sizing into account. This now gives you the second host onto which your third VM, that requires 2 full profile vGPUs, can be provisioned.

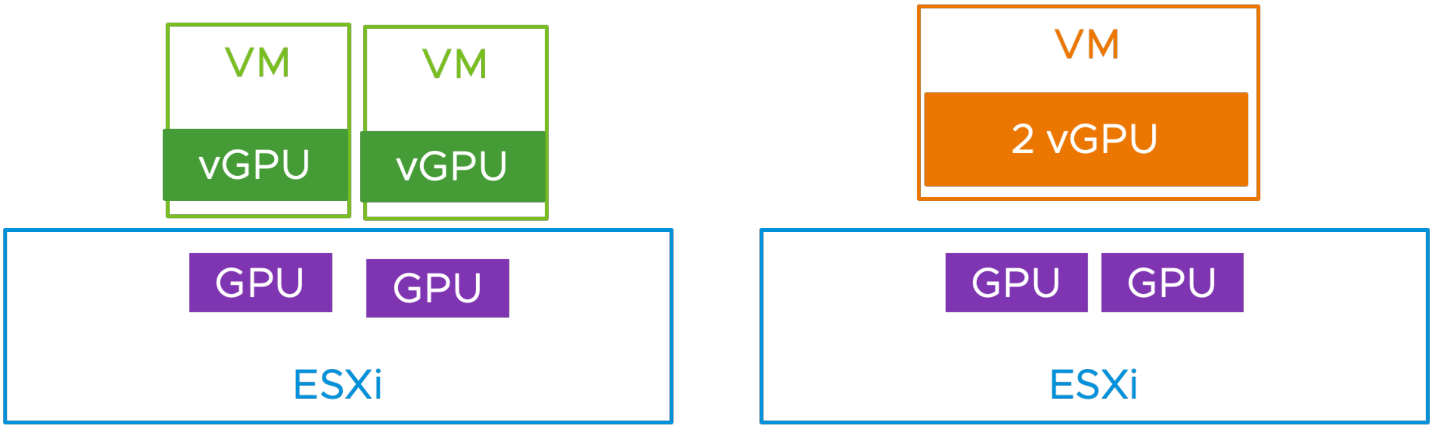

Figure 3: The Two vGPU VM is now placed onto second GPU host

This vSphere 8 Update 2 feature allows DRS to select the best host to achieve consolidation (that is, ideally bin-packed VMs) if you want that approach to be used. The recommendation will result in a vMotion if DRS automation is enabled for vGPU VMs.

It may take multiple DRS passes to approach an ideally bin-packed state, but each pass should not make fragmentation worse.

To avoid multiple vMotion events happening from any one host, when the VgpuVmConsolidation feature is on, we also set a second DRS advanced configuration option:

LBMaxVmotionPerHost = 1

A Second Example

Here is a second scenario where this consolidation of VMs to fewer hosts would be a better option, if consolidation is a requirement. This time we have three host servers, each with four physical GPUs, all of the same model, and we want to place a mixture of 2 vGPU VMs and a 4 vGPU VM onto these hosts. This scenario has come up in customer deployments.

In the pre 8.0 Update 2 versions of vSphere, we can be subject to the same placement conundrum as seen above with the default dispersal of smaller VMs to multiple hosts. Even though there are four GPUs available in aggregate across two of our servers, that 4 x vGPU VM at the top of the diagram below cannot be placed onto a host. We cannot split the VM across two host servers.

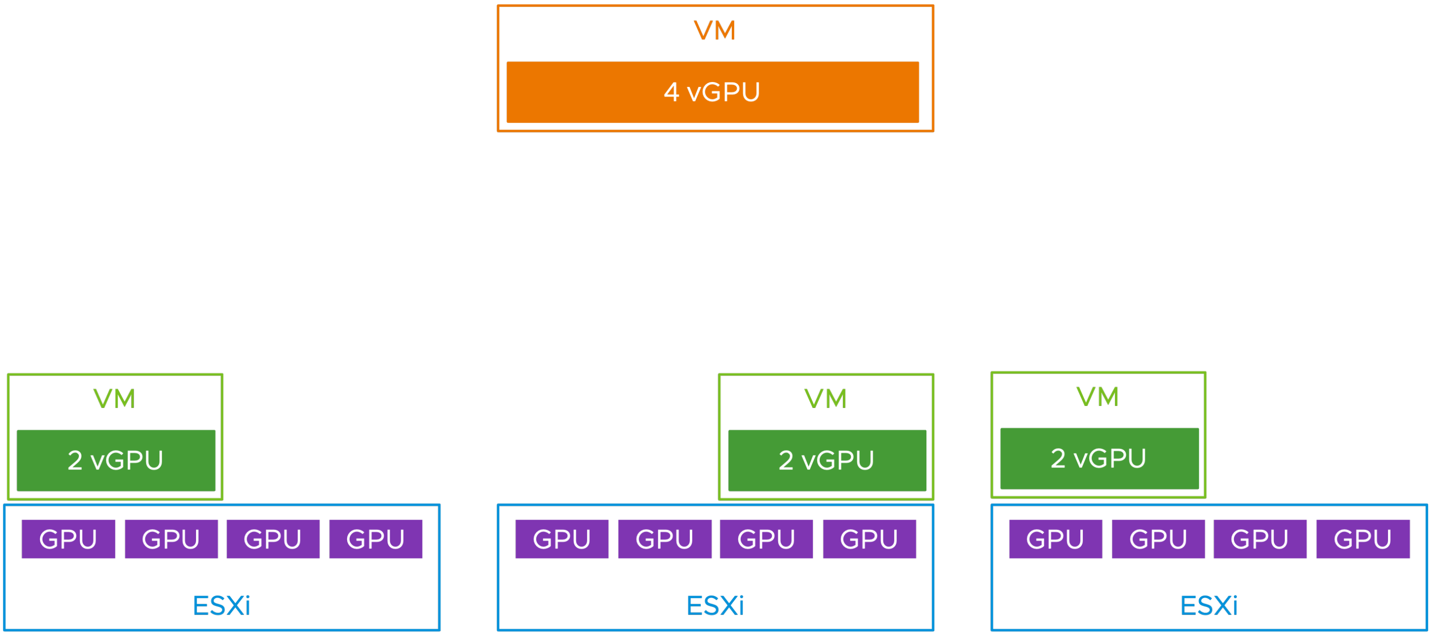

Figure 4: A 4 full vGPU VM is waiting for a host to be placed onto

However, using the consolidation feature of vSphere 8 Update 2 by setting VgpuVmConsolidation = 1 at the DRS cluster level, we can instead have the scenario seen below, with a satisfactory placement of that large VM onto a host of its own.

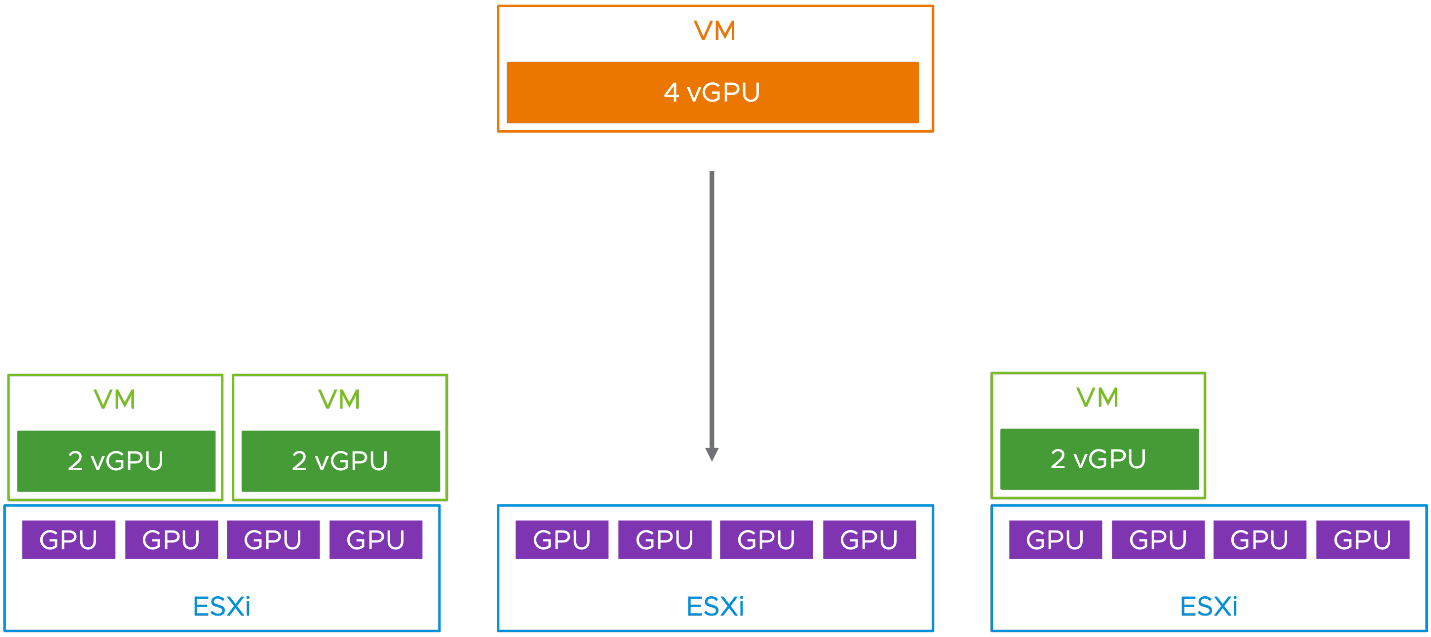

Figure 5: The full profile four vGPU VM gets assigned to a host successfully

We can see now that we are making much better use of our GPU hardware and servers using this VM consolidation option. For those situations where more than one full GPU, represented by its vGPU profile, is required for a number of differently-sized VMs, this new consolidation option will be very useful to those who want to make full use of their GPU hardware resources.