usenix THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

gScale: Scaling up GPU Virtualization with Dynamic Sharing of Graphics Memory Space

Mochi Xue

Shanghai Jiao Tong University

Intel Corporation

GPU in Cloud

- **GPU Cloud** is introduced to meet the high demand of GPU resource.
- As a key enabling technology of GPU Cloud, **GPU virtualization** is intended to provide flexible and scalable GPU resources for multiple instances.
- Some GPU virtualization solutions spring out recently:

GPUvm	gVirt
USENIX ATC'14	USENIX ATC'14

gVirt (iGVT-g)

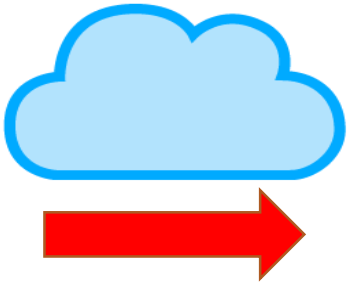
A full GPU virtualization solution with mediated pass-through for Intel GPU.

- Full Features
- High Performance
- Open Source

However, gVirt suffers from its limited scalability.

Scalability Issue

Scalability is an indispensable feature which ensures high resource utilization by hosting dense VM instances on cloud servers.



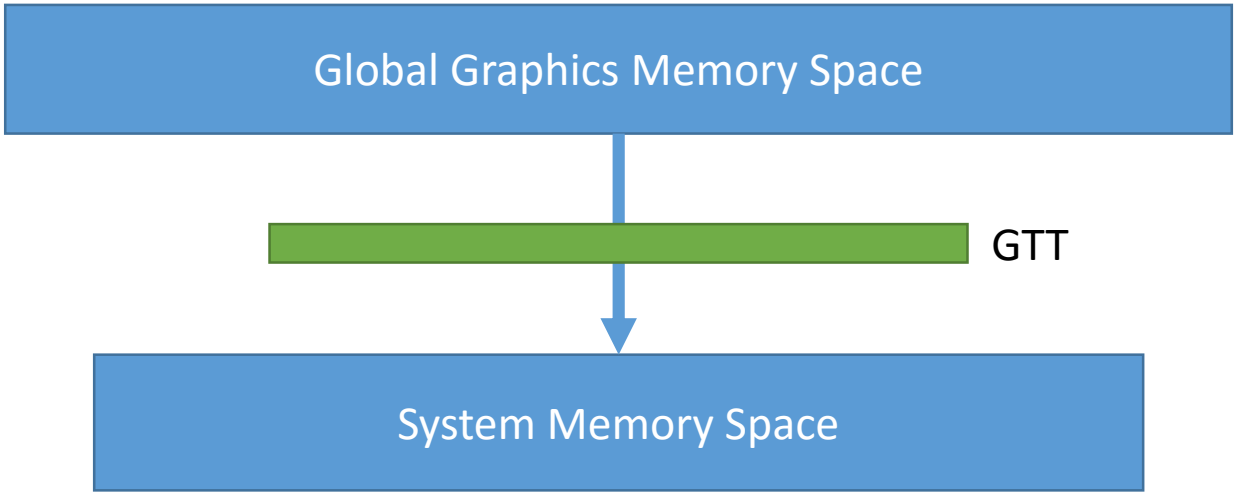
Limited Scalability → Resource Underutilization

For GPU embedded in Intel HASWELL CPU, gVirt is only able to host 3 guest VM instances maximally on one physical server.

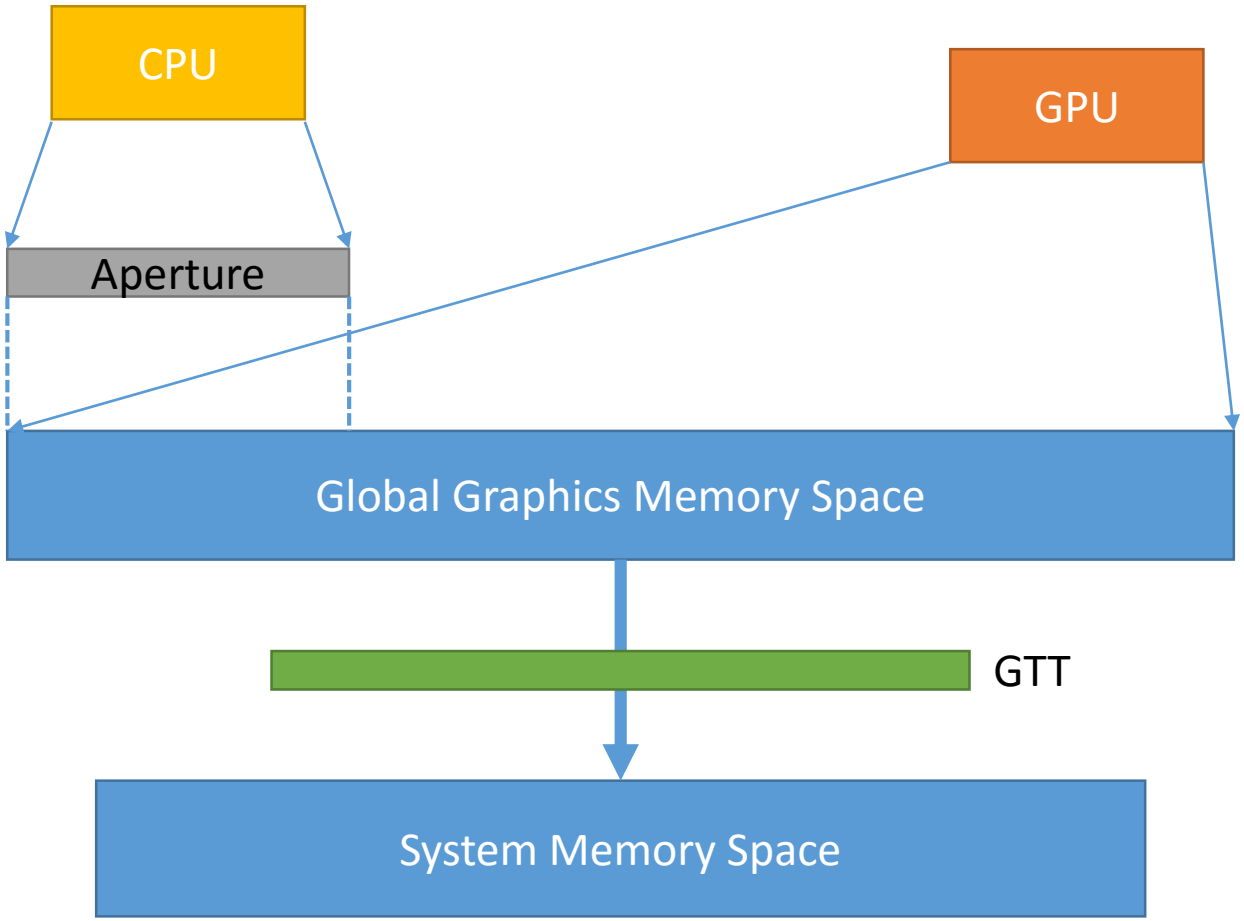
Background

Graphics Translation Table (GTT)

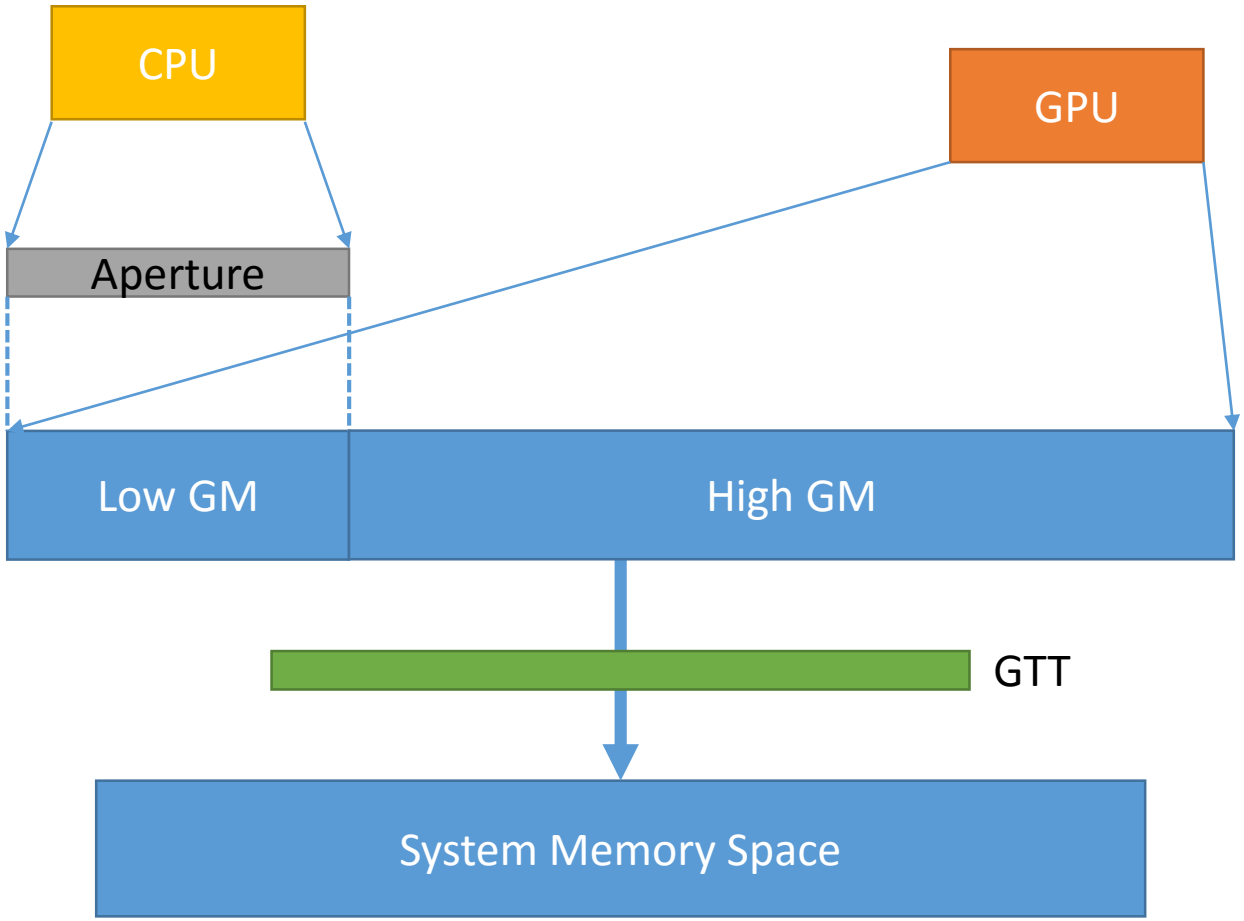
- Also known as *global graphics translation table*
- A memory-resident page table
- Global Graphics Memory Address -> System Memory Address.



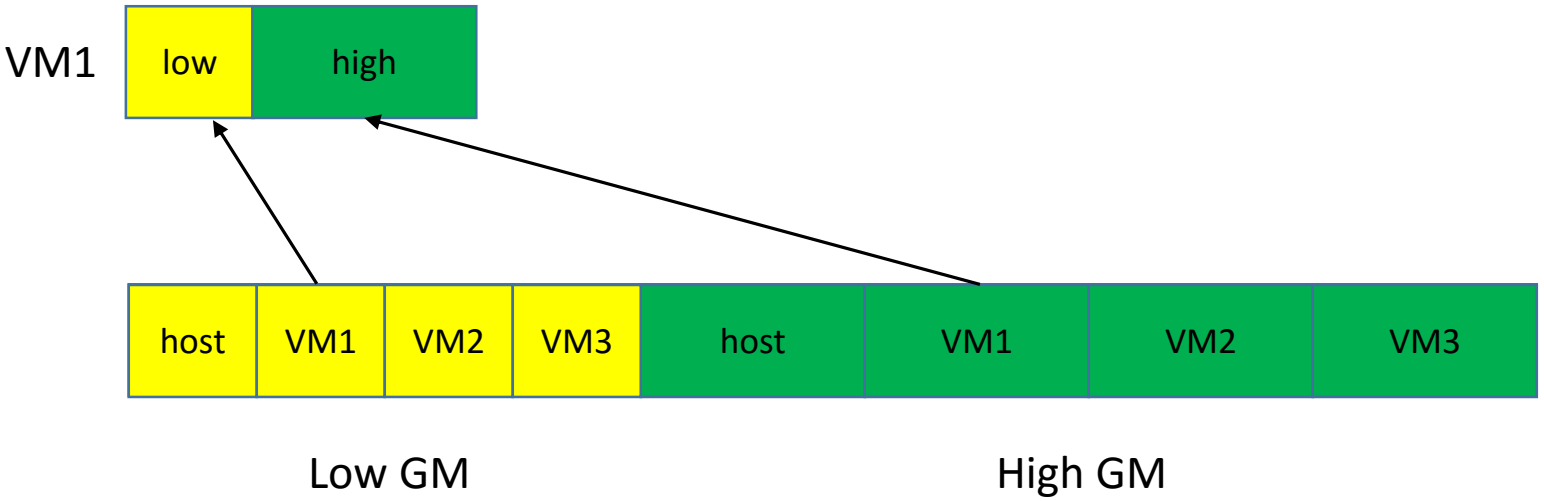
Background


CPU
GPU
Aperture
Global Graphics Memory Space
GTT
System Memory Space

Background



Static Partition



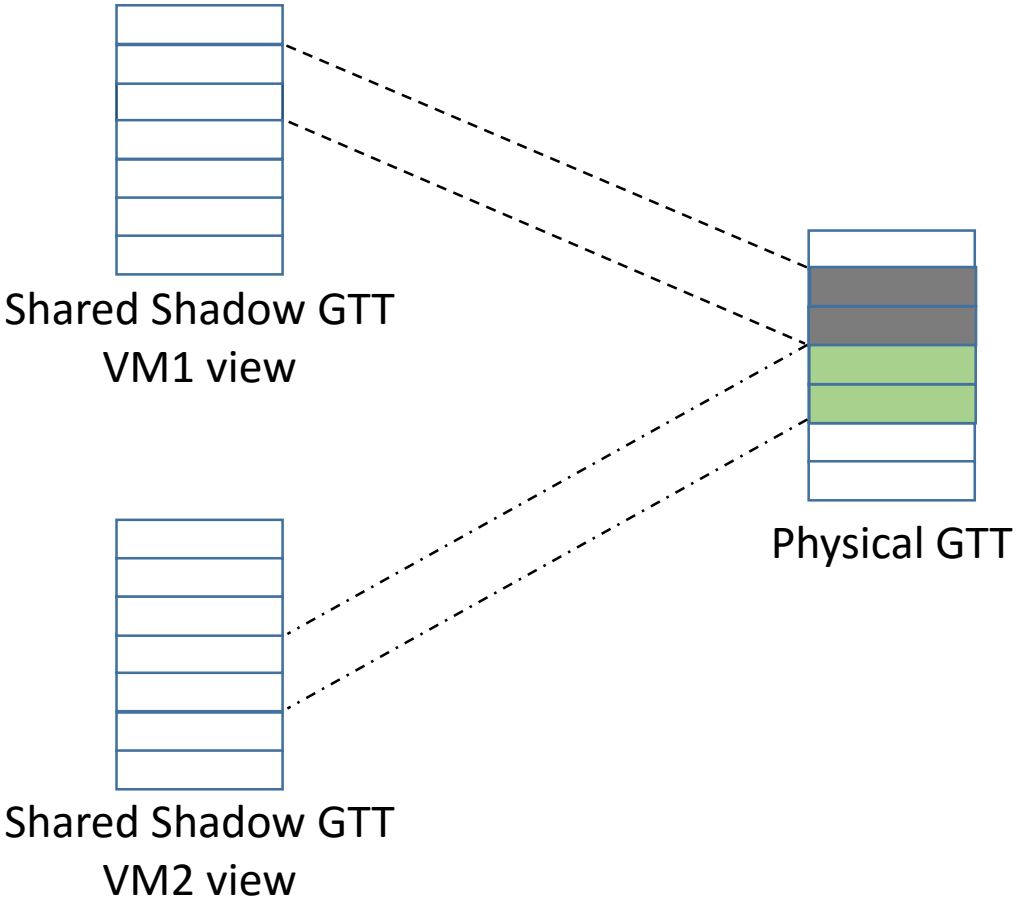
For Intel GPU (HASWELL CPU):
2MB GTT -> 2GB Graphics Memory Space
512KB Aperture -> 512MB low GM + 1536 MB high GM

Recommended VM settings:
128 MB low GM (512 / 128 = 4)
384 MB high GM (1536 / 384 = 4)
4 – 1 = 3 (guest VMs)

Shared Shadow GTT

Shared Shadow GTT

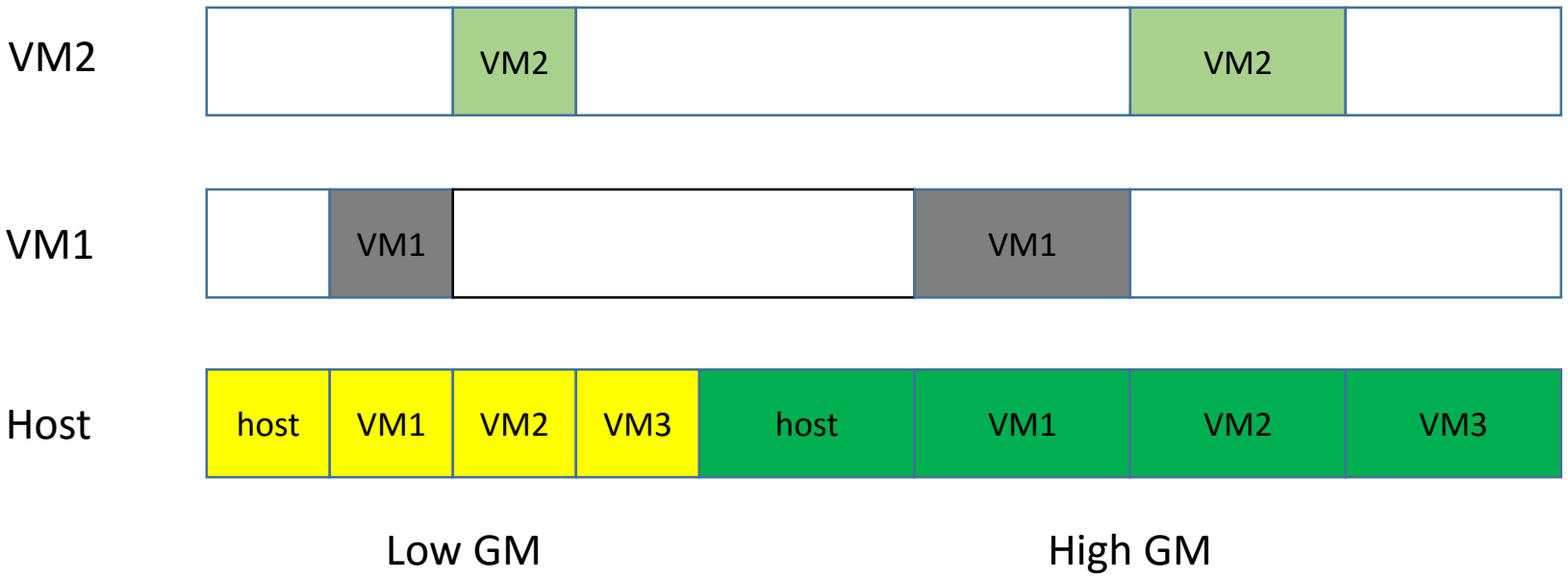
- A copy of physical GTT shared by all the guest VMs.
- It exposes different parts of GTT to each guest VM.



Shared Shadow GTT

Shared Shadow GTT

- VM can only access its assigned part of Graphics Memory Space.
- Graphics driver marks the rest part of Graphics Memory Space as inaccessible.



gScale

A scalable GPU virtualization solution with dynamic sharing of graphics memory space based on gVirt.

Private Shadow GTT

Ladder Mapping

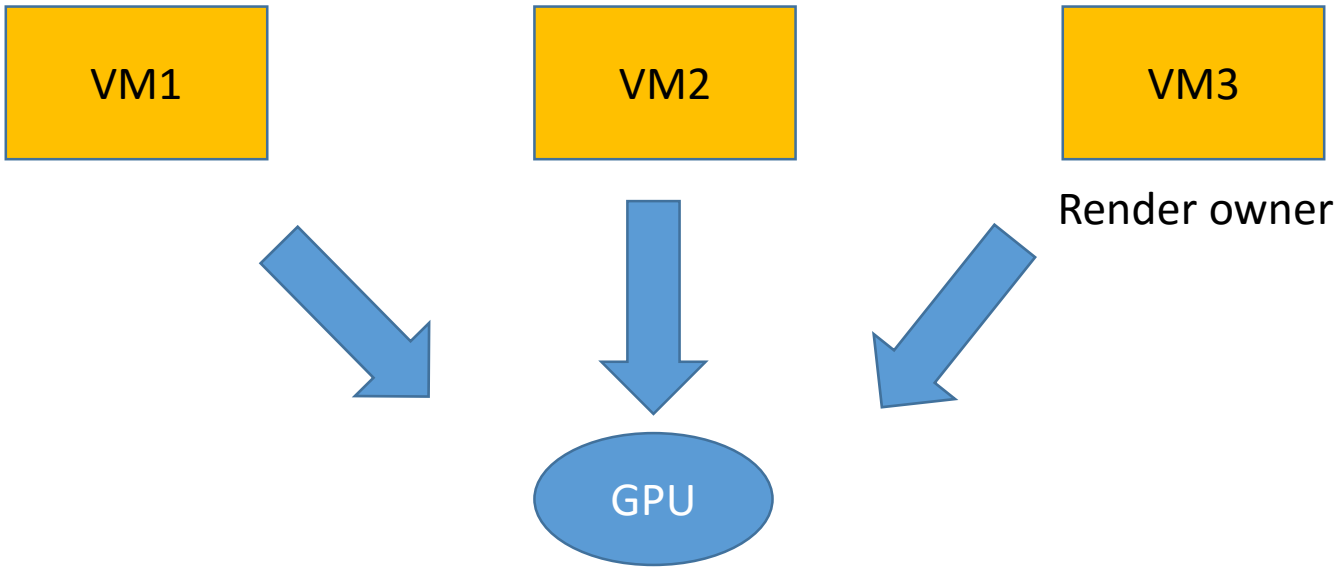
Slot Sharing

Up to 5x scalability of gVirt with slight runtime overhead.

Private Shadow GTT

GPU Scheduling Pattern

- VMs take turns to submit workloads to physical GPU.
- Each VM occupy the GPU for a time slice.

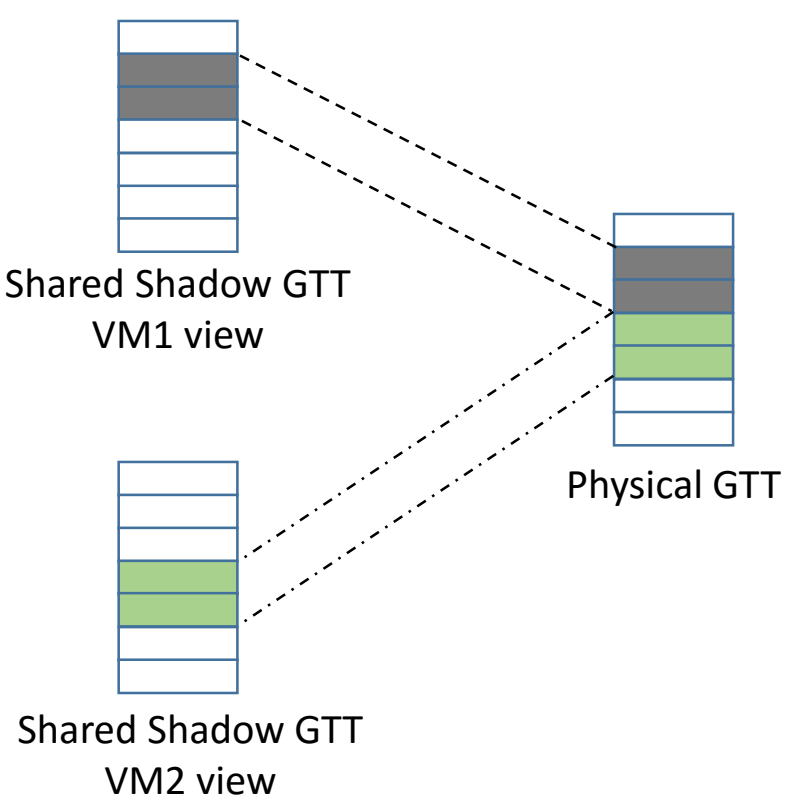


At the same time, only one VM is served by physical GPU.

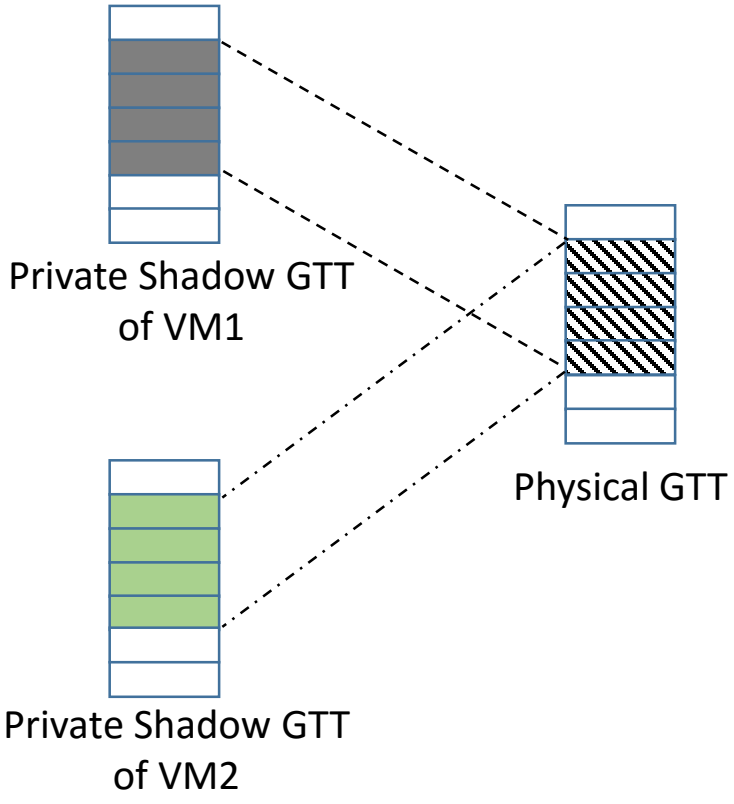
Private Shadow GTT

Private Shadow GTT

- A specific copy of physical GTT for each guest VM.
- It could expose same part of GTT to guest VMs.



Shared Shadow GTT

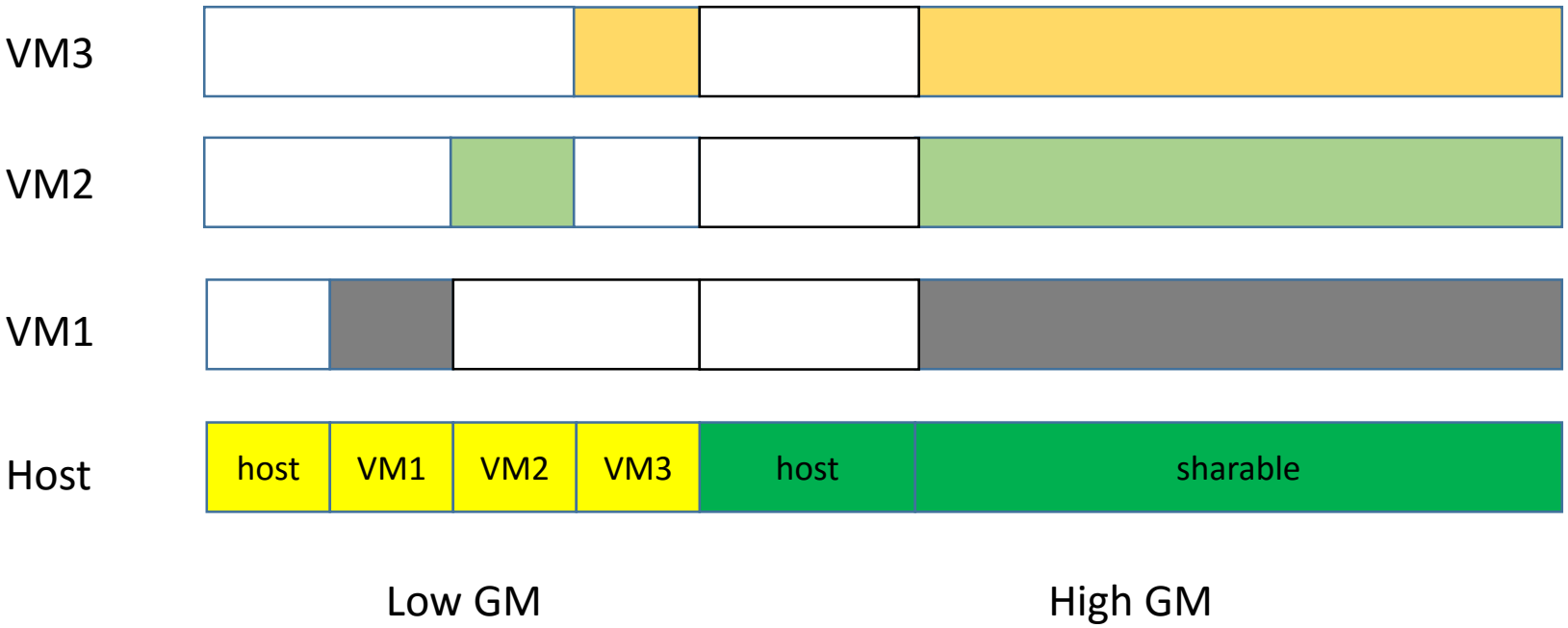


Private Shadow GTT

Private Shadow GTT

Shared Shadow GTT

- When context switch happens, gScale writes VM's private shadow GTT onto physical GTT.
- Guest VMs could use the same range of Graphics Memory Space.
- High GM space now is sharable among the guest VMs.

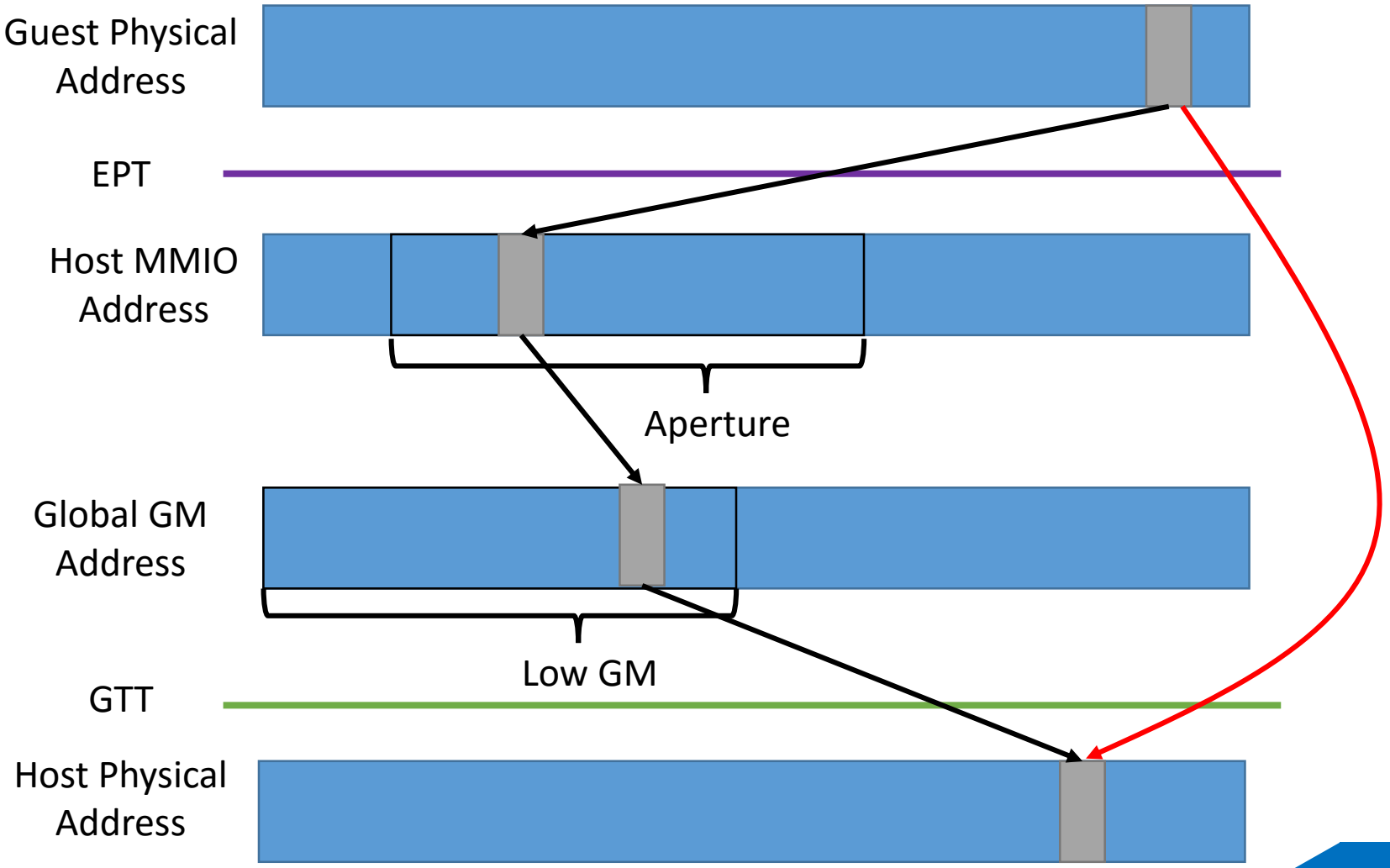


Ladder Mapping

- High GM space is only accessible by GPU.
- Low GM space is also accessible by CPU.
- Multiple VMs could access the low GM simultaneously.

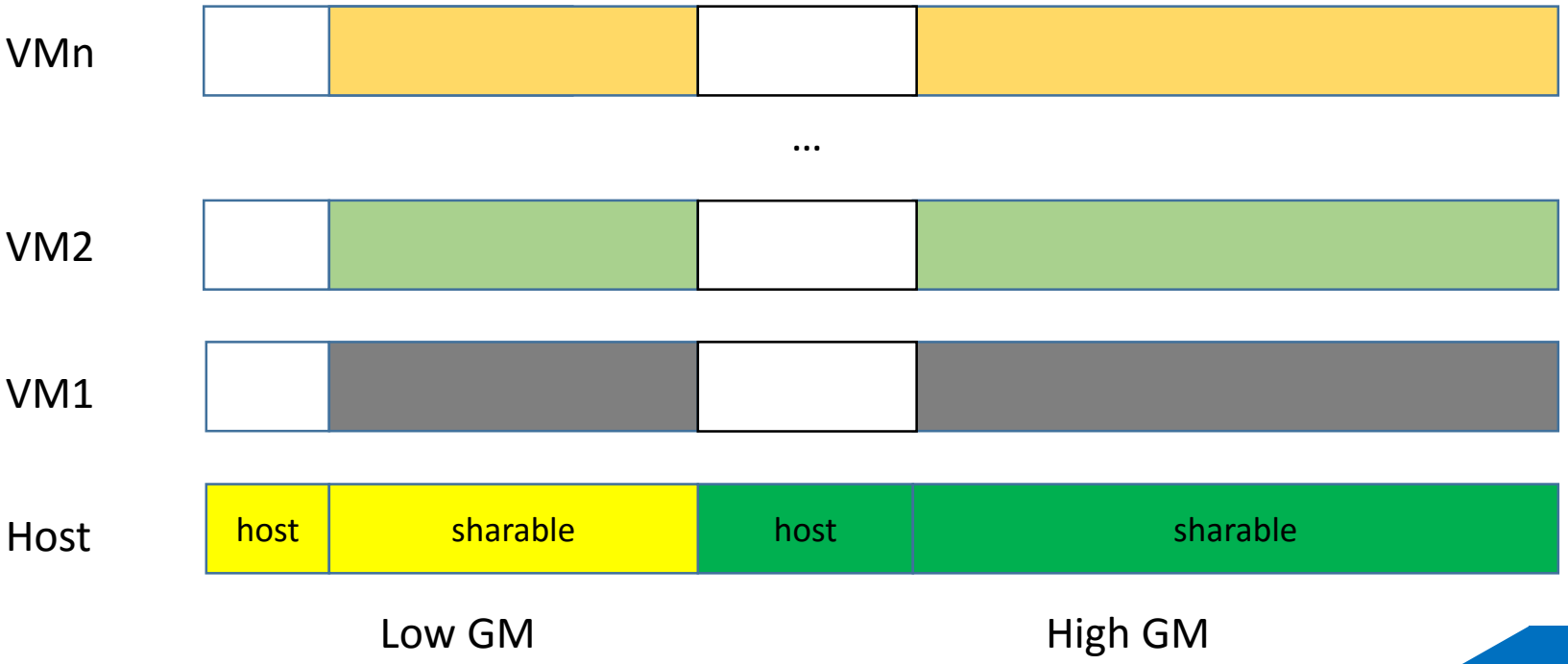
Private shadow GTT doesn’t work for low GM.

Ladder Mapping

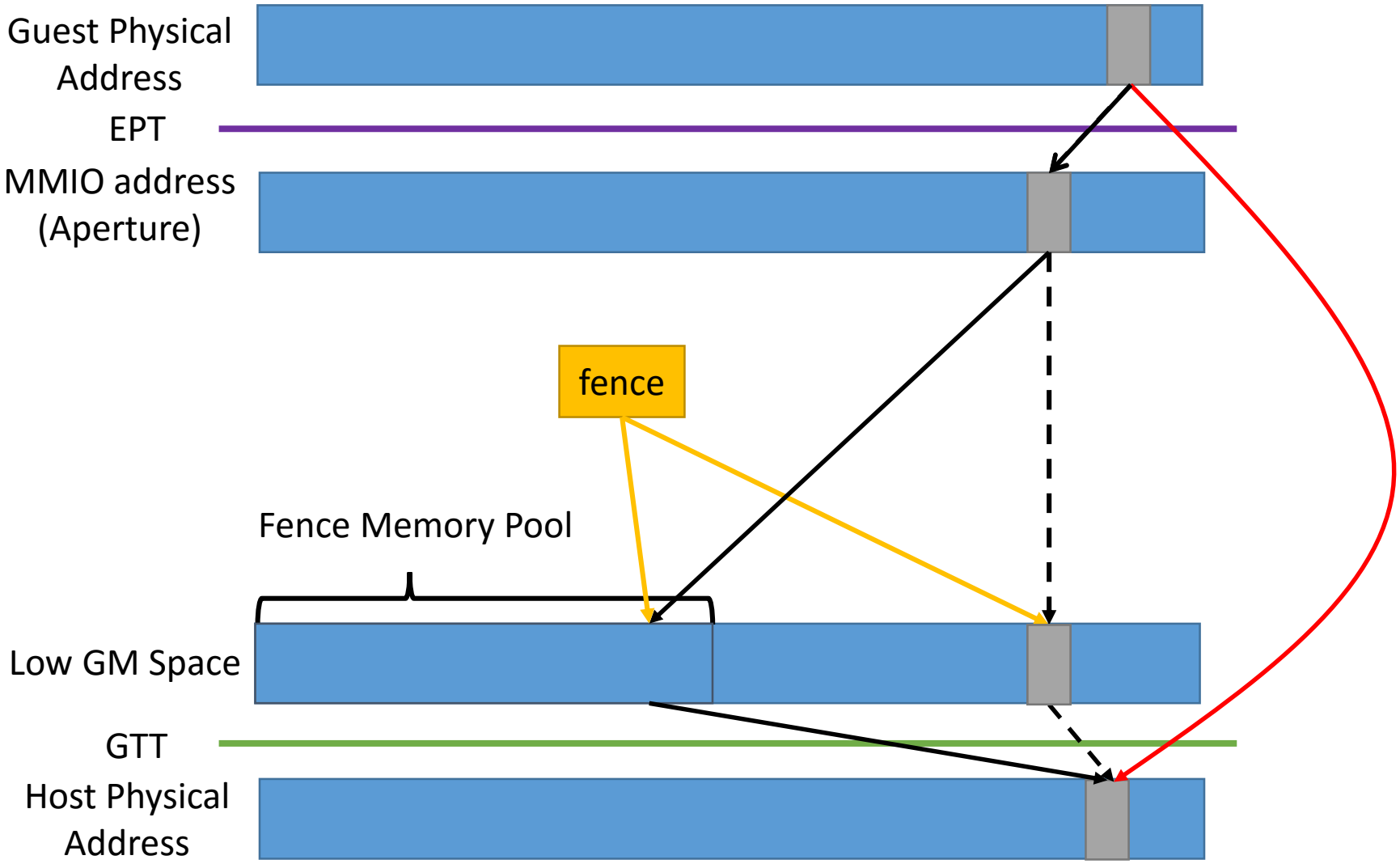


Ladder Mapping

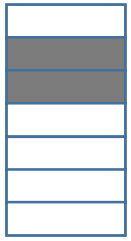
- Force CPU access system memory space bypassing the low GM space. Low GM now is not accessible for CPU.
- We could enable dynamic sharing for the whole graphics memory space.



Fence Memory Pool



Optimization: Partial Copying



Private Shadow GTT of VM1

Physical GTT

Low GM

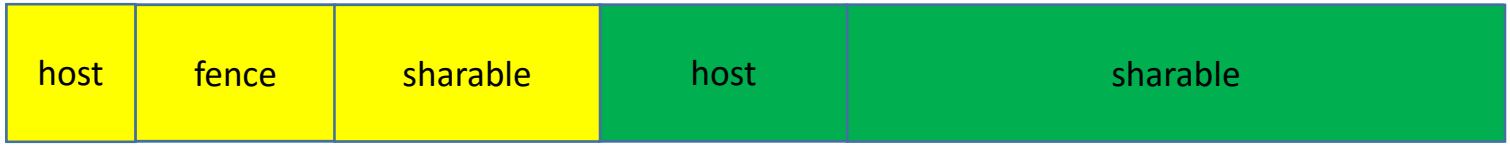
High GM

gScale only copies sharable part of GTT

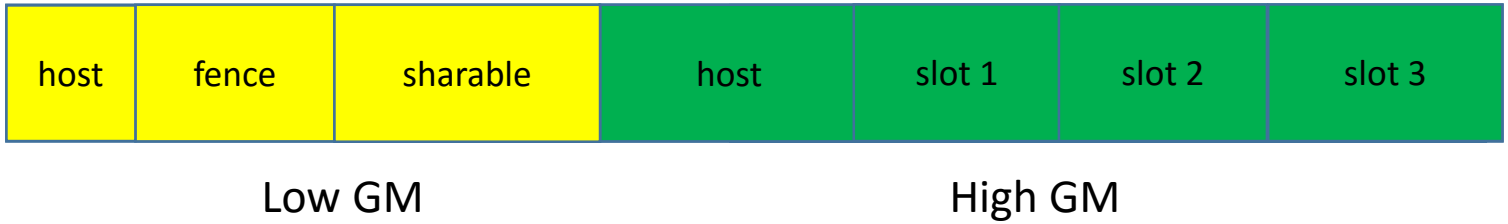
Private Shadow GTT of VM1

Physical GTT

Slot Sharing



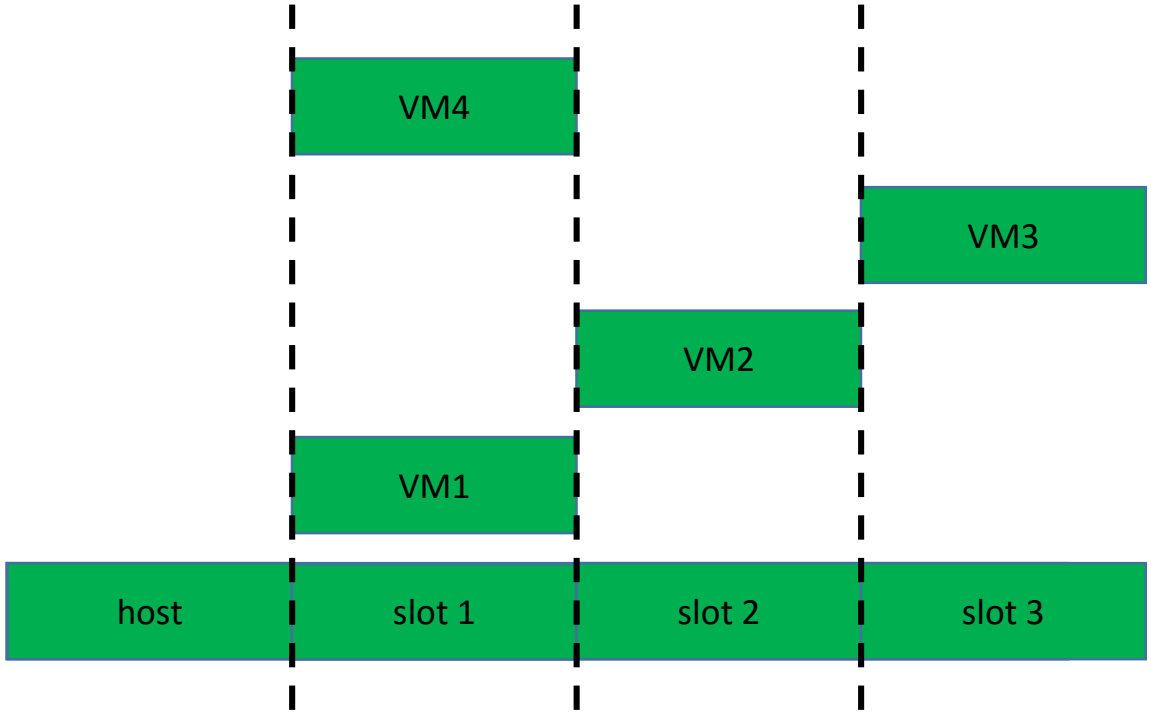
More high GM space than standard configuration does not increase the performance of VM.



- gScale divides sharable high GM space into 3 slots.
- Each slot could hold one VM's high GM.

Slot Sharing

- Optimized Scheduler: gScale does not switch GTT of idle VM.
- Put idle VM with busy VM in one slot.



Evaluation

1. Performance comparison with gVirt
- Up to 99% performance of gVirt.
- The overhead is less than 5% of gVirt.

2. Scalability of gScale (Linux & Windows)
- For Linux, gScale hosts up to 15 guest VMs. (5x)
- For Windows, gScale hosts up to 12 guest VMs. (4x)

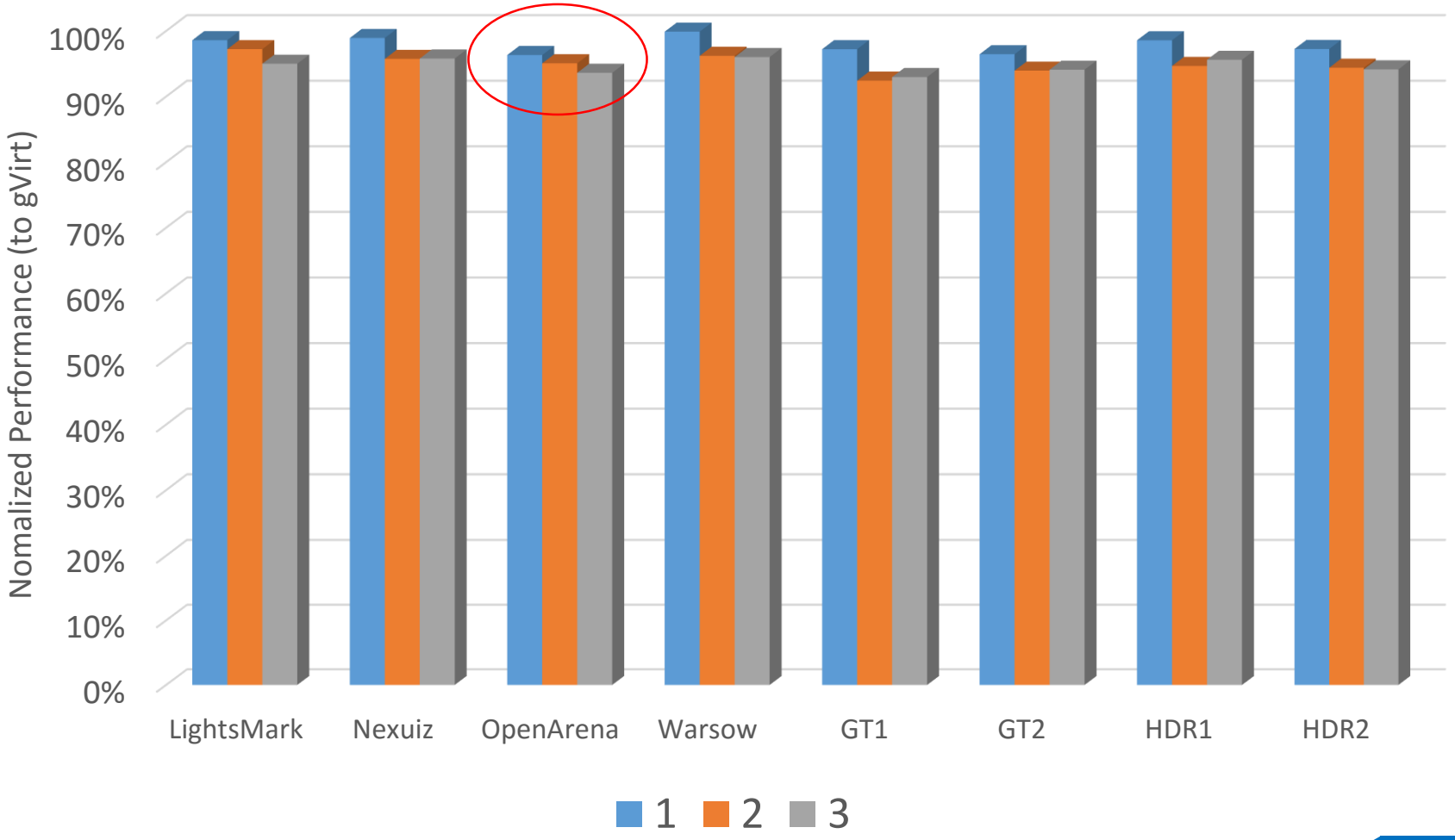
3. Performance impact of slot sharing
- Slot sharing could reduce up to 80% overhead of GTT switch.

Configuration

Hardware	
CPU	Intel E3-1285 v3 (4 cores, 3.6Ghz)
RAM	32GB
HDD	SAMSUNG 850Pro 512GB * 3
GPU	Intel HD Graphics P4700
Linux/Windows Guest VM	
vCPU	2
Memory	1800/2048 MB
Low GM	64/128 MB
High GM	384 MB
OS	Ubuntu 14.04/Windows 7
Benchmarks	
Phoronix Test Suit 3D Marks	
3DMark06	

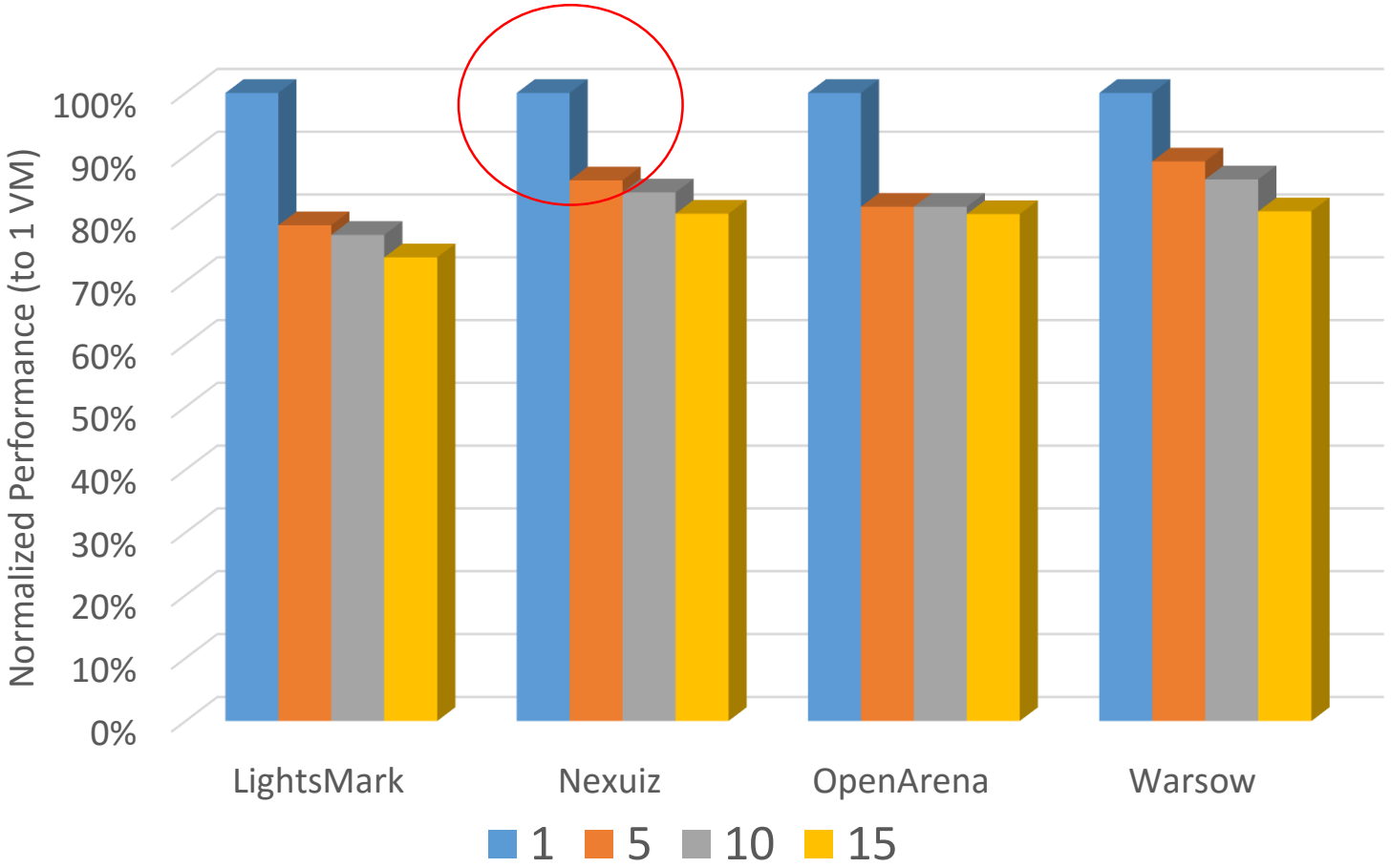
Compare with gVirt

The performance is up to 99% of gVirt, while the overhead is less than 5% of gVirt's performance.



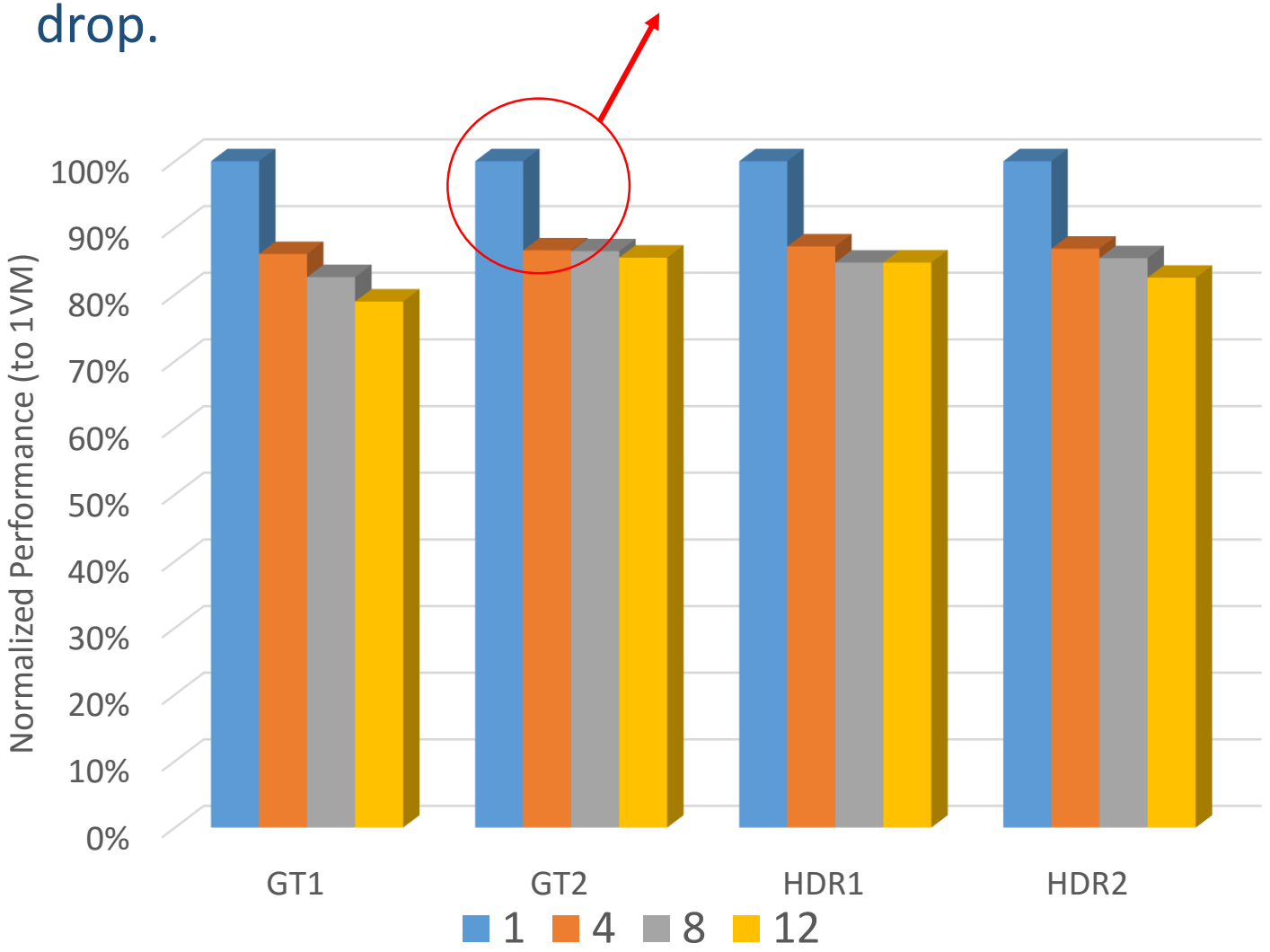
Scalability of Linux VM

When VM is over 1, GTT copying causes the performance drop.



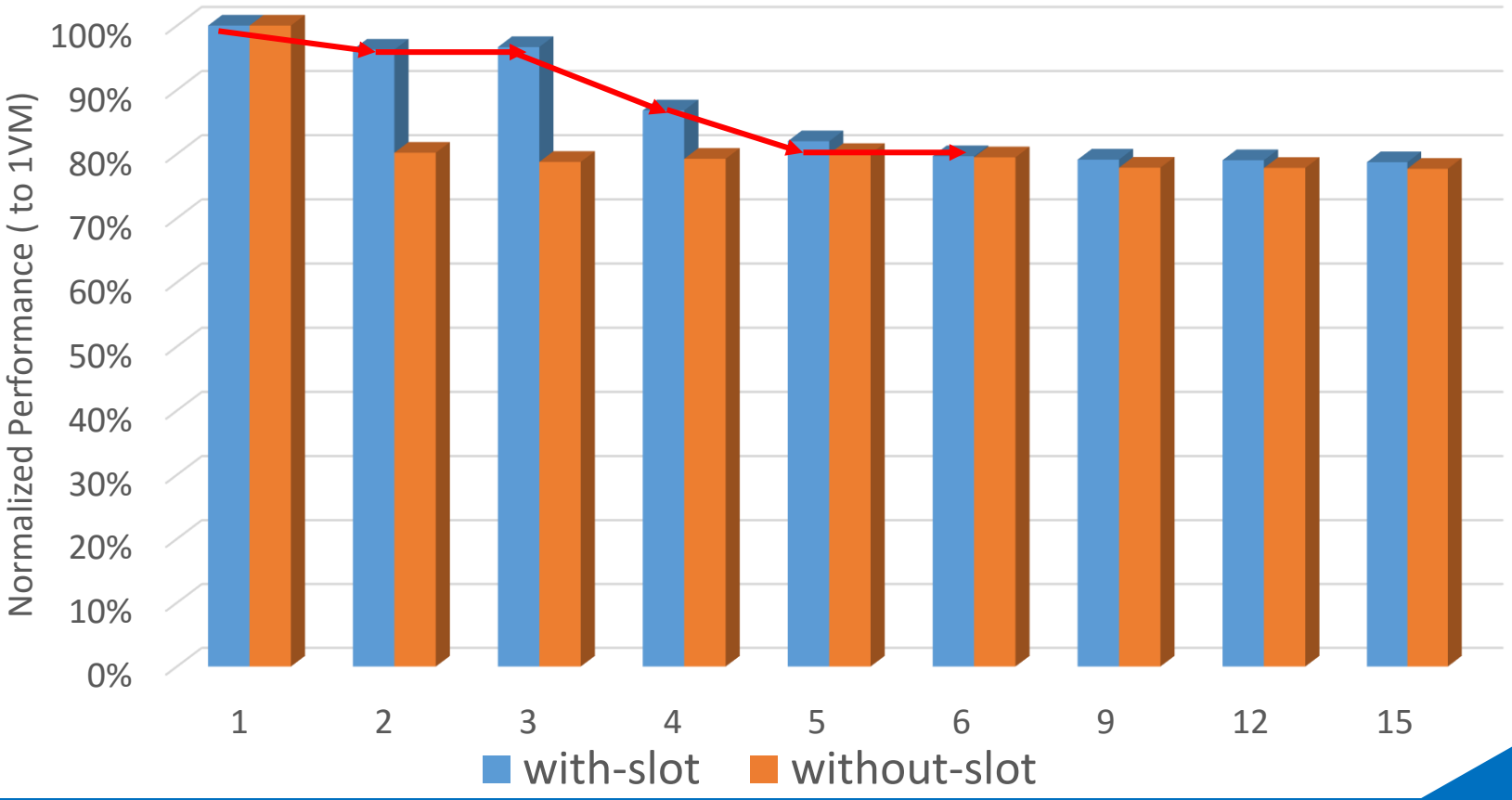
Scalability of Windows VM

When VM is over 1, GTT switch causes the performance drop.

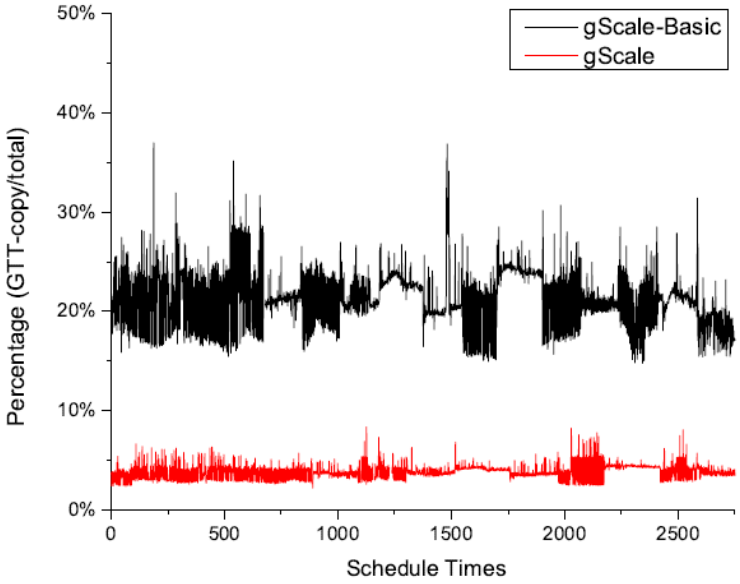


Performance with Slot Sharing

Under a high density of VM, slot sharing could mitigate the performance drop caused by shadow GTT copying.

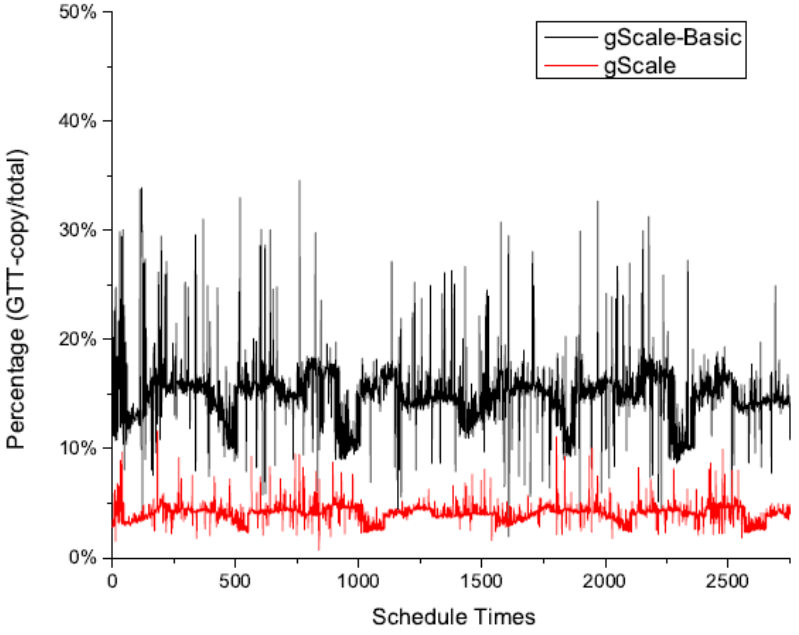


Overhead Impact of Slot Sharing



For Linux VM, the slot sharing reduces the overhead of shadow GTT copying by 83%.

For Windows VM, the slot sharing reduces the overhead of shadow GTT copying by 73%.



Summary

A scalable GPU virtualization solution with dynamic sharing of graphics memory space based on gVirt.

Up to 5x Scalability of gVirt.

Source code will soon be available

https://01.org/igvt-g

usenix
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION

Q&A

gScale: Scaling up GPU Virtualization with Dynamic Sharing of GraphicsMemory Space

Q&A