Improving Quality of Service (QoS) of Multi-Tenant GPU Clouds Enabled by Passthrough

Youssef Elmougy and Professor Jianchen Shan Ph.D.

Fred DeMatteis School Of Engineering And Applied Science, Hofstra University

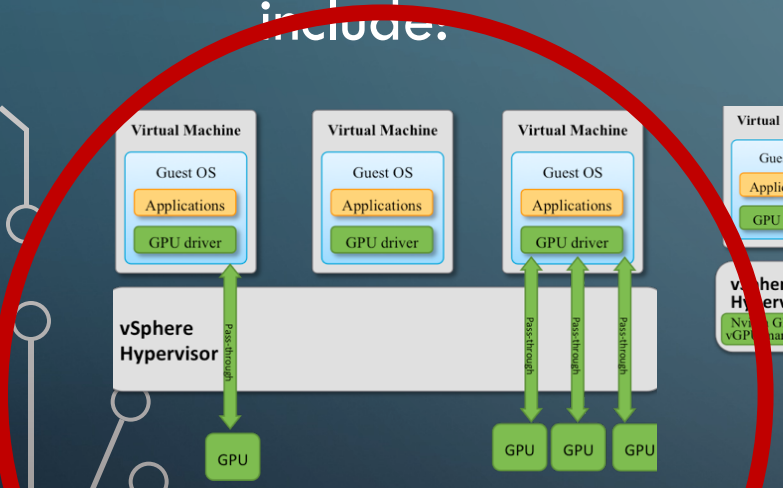
BACKGROUND

GPGPU (General Purpose GPU) greatly accelerates large scale data processing and parallel computing. Thus, most major cloud providers have introduced the GPU cloud by provisioning the GPU instances as compared to the already existing cloud-based CPUs.

Cloud Provider	vCPUs / instance	vGPUs / instance
Microsoft Azure	6, 12, 24	1, 2, 4
Amazon Web Services (AWS)	8, 32, 64	1, 4, 8
Google Cloud	16, 32, 64	1, 2, 4
IBM Cloud	16, 28, 36	1, 2
Oracle Cloud	12, 24, 28, 52	1, 2, 4, 8

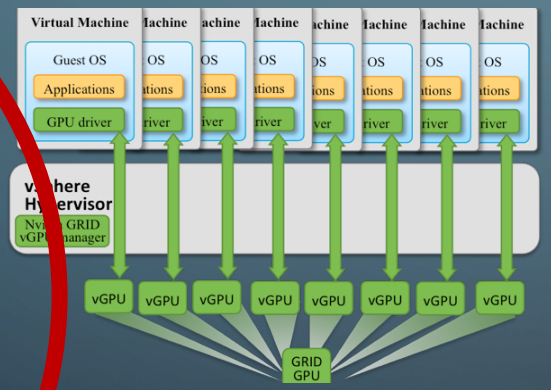
BACKGROUND continued…

GPU virtualization, as the key enabling technology, allows GPU instances to access either the shared or dedicated GPU devices. Several main techniques include:



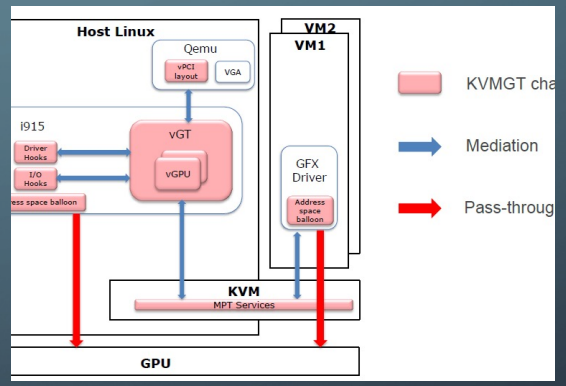
PCI Passthrough (DirectPath I/O)

(DEDICATED GPU)



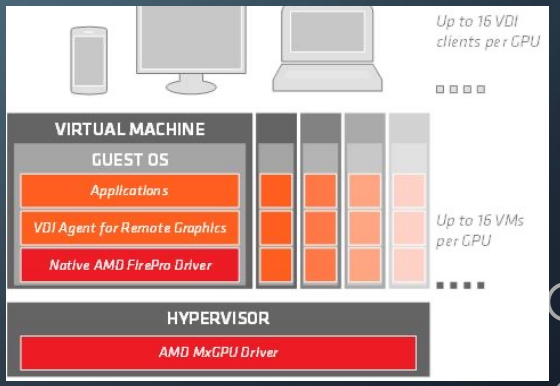
NVIDIA GRID vGPU

(SHARED GPU)



intel KVMGT

(SHARED GPU)

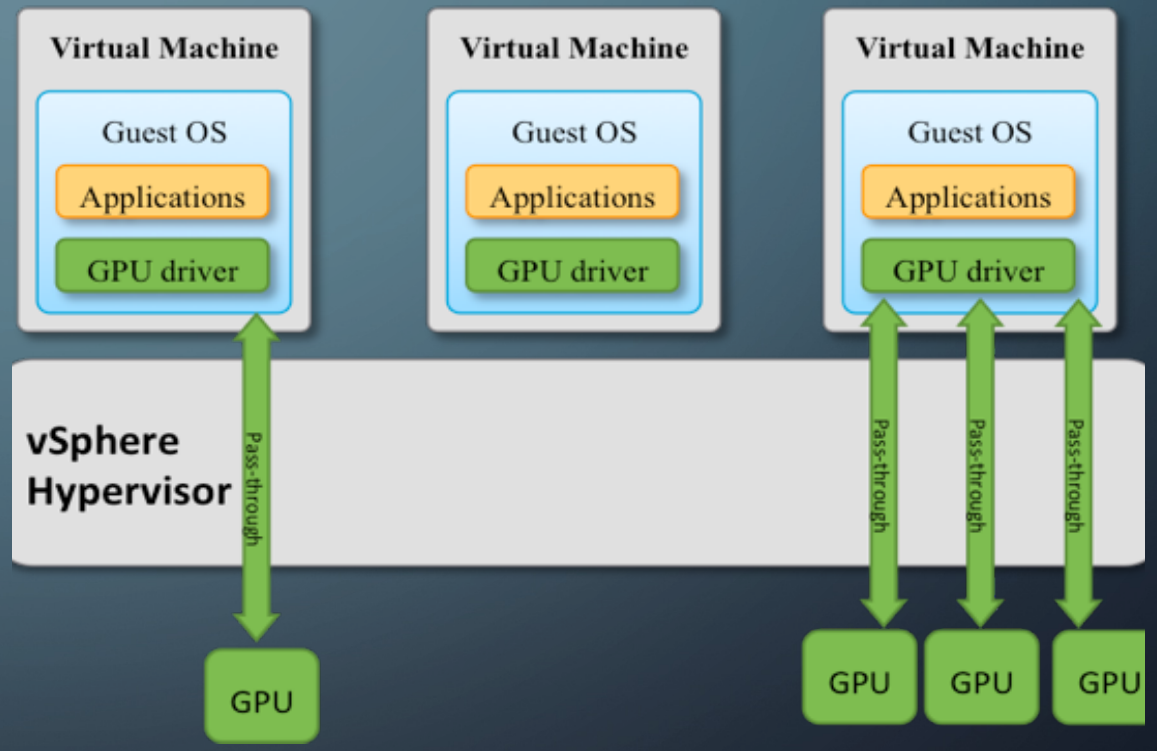


AMD MxGPU

(SHARED GPU)

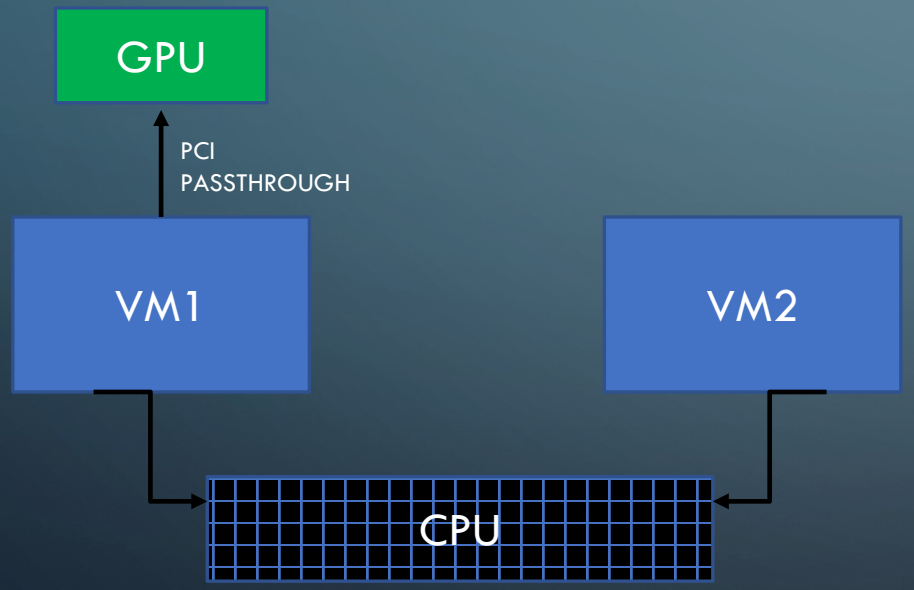
BACKGROUND continued…

PCI Passthrough is the most widely used technique which supports high performance cloud infrastructure. It is the method by which a Virtual Machine (VM) gains direct access to the dedicated GPU, permitting the VM to fully benefit from the GPU's processing capability.



PROBLEM: GPU UNDERUTILIZATION IN MULTI-TENANT CLOUDS

A multi-tenant cloud is an architecture that allows multiple consumer instances to operate and share computing resources on a server.



- Both VM1 and VM2 share the CPU resource because cloud providers aim to increase utilization of available physical devices as well as reduce waste cost on the cloud.
- VM1 has a dedicated GPU as it allows for bare-metal performance which assists in increased device usage efficiency.

MOTIVATING EXPERIMENT

To show the GPU underutilization problem, experiments were conducted on the **JUPITER server**, which contains **4 Intel Xeon Gold 6138 CPUs with 80 cores**@2.00GHz and **2 NVIDIA Corporation Tesla P100 PCIe 16GB GPGPUs**. Two Virtual Machines (VMs) are setup as a representation of the multi-tenant GPU cloud.

VM1 – GPU Instance	VM2 – CPU Instance (corunner VM)
• 16 vCPU • Dedicated GPU attached by PCI Passthrough	• 16 vCPU • No GPU

GPU Intensive Benchmarks {CAT1}		CPU Intensive Benchmarks {CAT2} (used in corunner)	
Benchmark name	**Field**	**Benchmark name**	**Field**
NAMD	Scientific Simulation	Streamcluster	Data Mining
Gromacs	Biochemical Molecular Dynamics	matmul	Matrix Multiplication
DQN	Artificial Intelligence	Bodytrack	Computer Vision

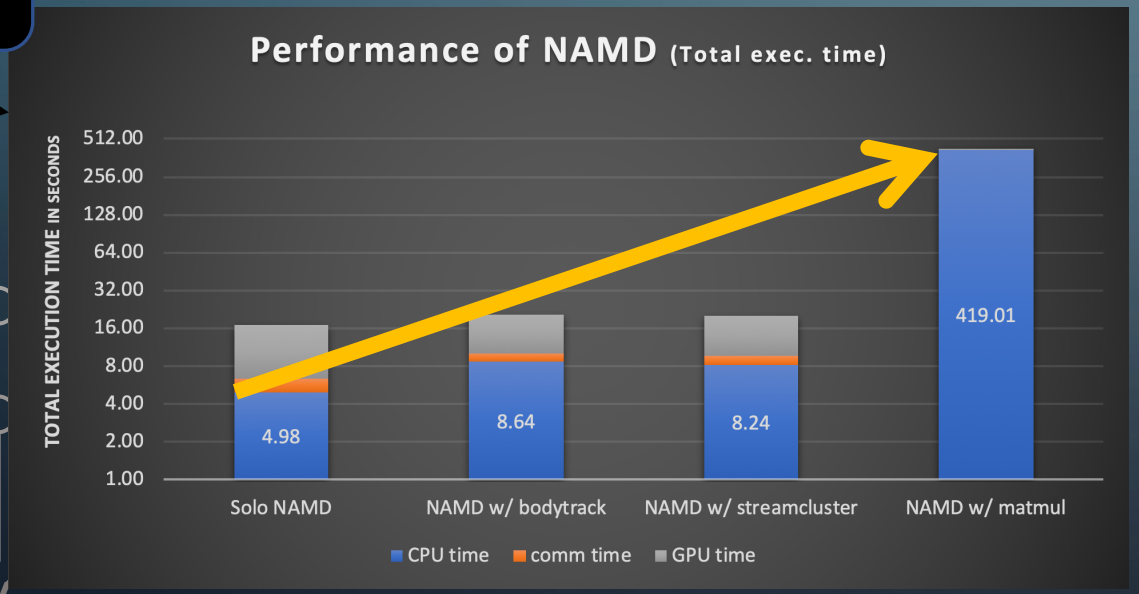
MOTIVATING EXPERIMENT continued…

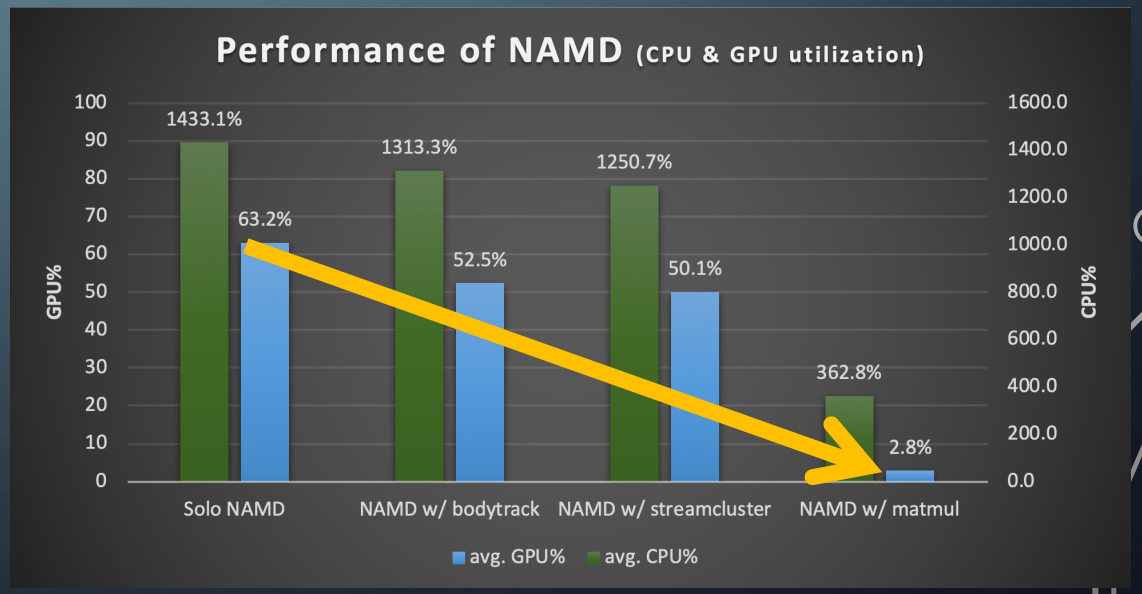
Each benchmark in CAT1 was evaluated in the following two scenarios:

1. when the benchmark was running in the first VM without corunner VM running
2. when the benchmark was running in the first VM with the corunner VM executing each benchmark in CAT2

Log graph

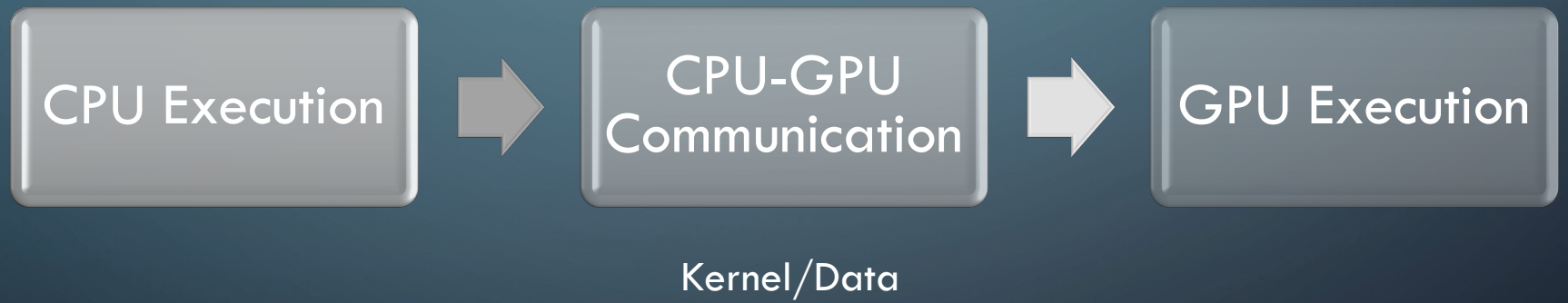
The following shows the results of the NAMD benchmark executed in the two scenarios:





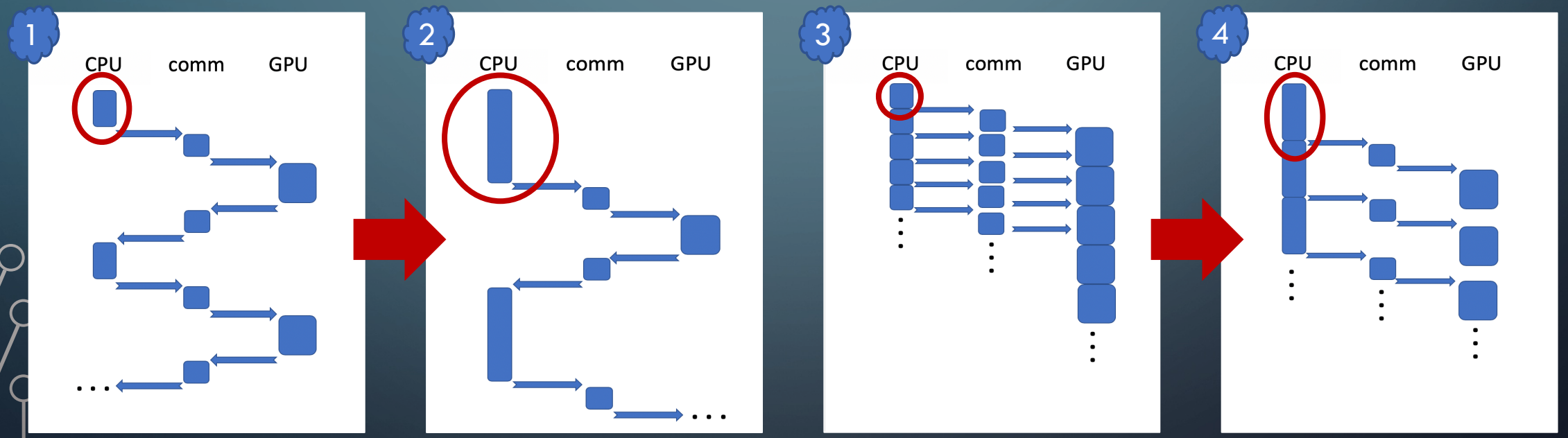
PROBLEM ANALYSIS

The dividing and issuing of tasks between the CPU and the GPU is an essential asset in program execution. This involves 3 major phases:



PROBLEM ANALYSIS continued…

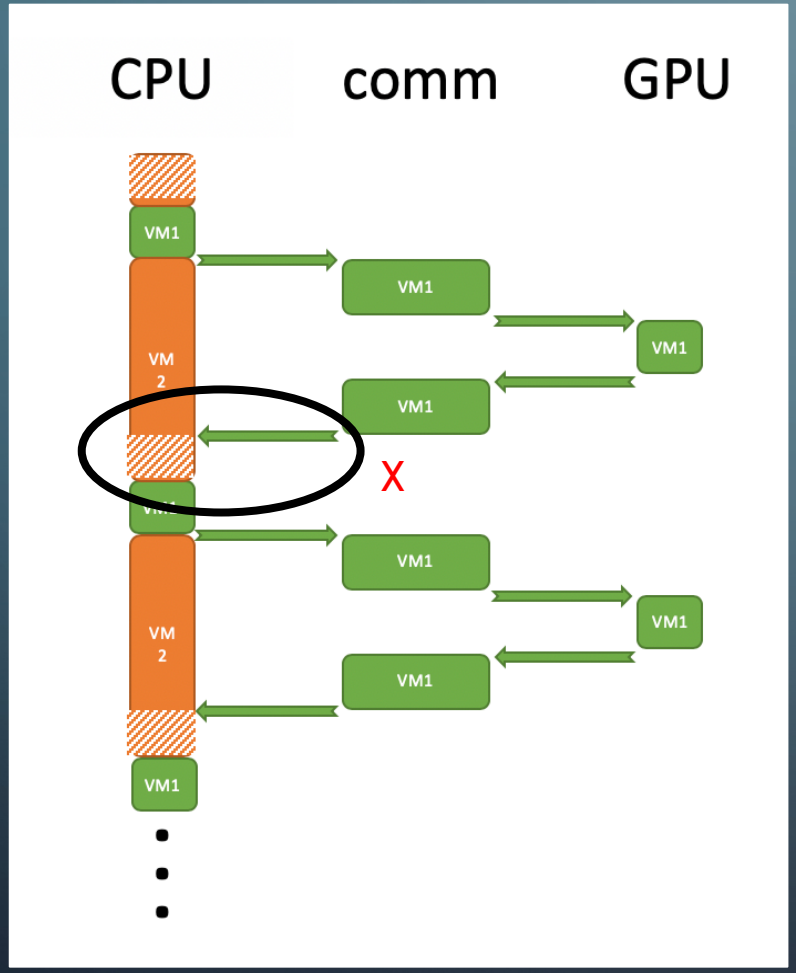
The following flowcharts demonstrate the different scenarios as to which the 3 phases may be executed:

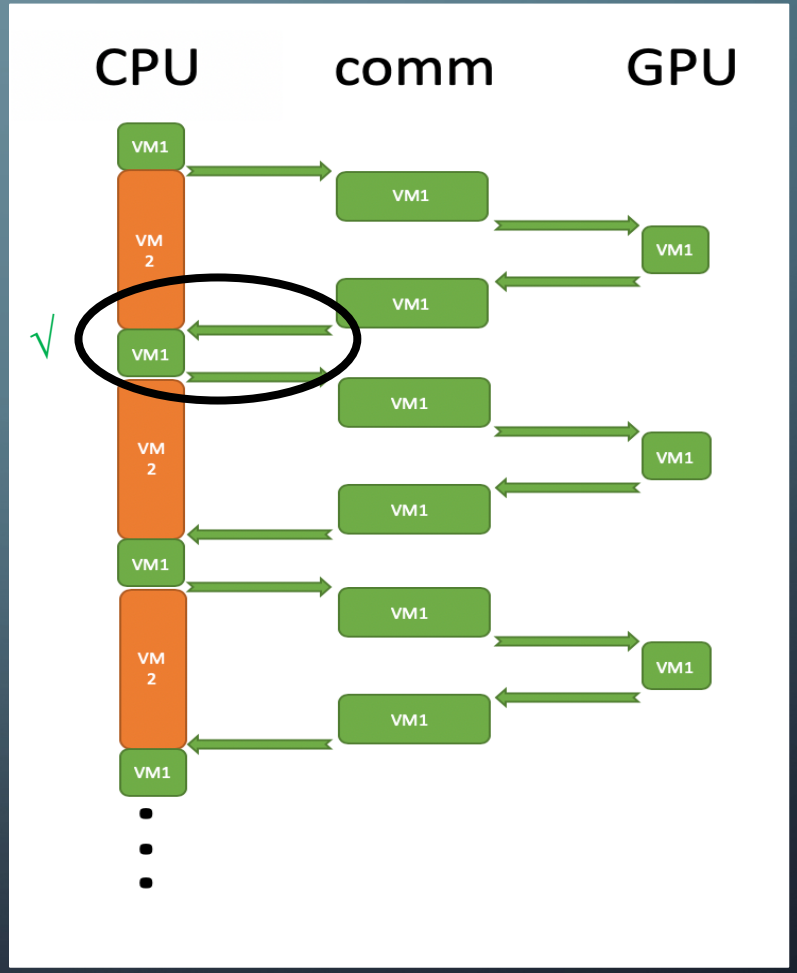


Synchronized

Asynchronized

PROPOSED SOLUTION (work in progress)





FUTURE WORK

The following ideas and experiments could be done:

◎ deeper analysis of already chosen benchmarks and the acquiring of more benchmarks in common fields of study

◎ more studying of CUDA programming to implement application-level modifications to benchmark program code

◎ producing and proposing a resilient system-level solution to lessen or eliminate the effects of the problem

THANK YOU FOR LISTENING!!

Questions?