GTC 2019
S9883: POWERING INTELLIGENT VIDEO ANALYTICS WITH NVIDIA VIRTUAL GPU

Vinay Bagade – vGPU Technical Marketing

Charlie Huang – vGPU Product Marketing

AGENDA

- Background
  - Intelligent Video Analytics and Deep Stream
  - What is Virtual GPU
- Intelligent Video Analytics using Virtual GPU - Proof of Concept
- What's Next
- Resources

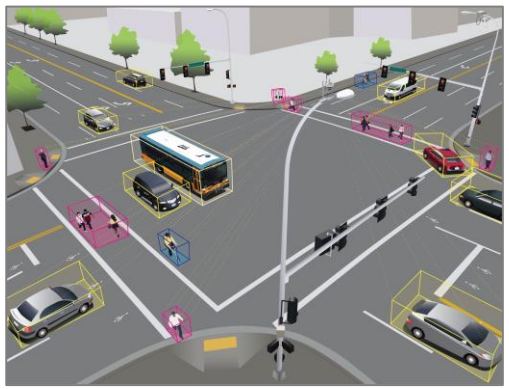
BACKGROUND

IVA FOR EFFICIENY AND SAFETY

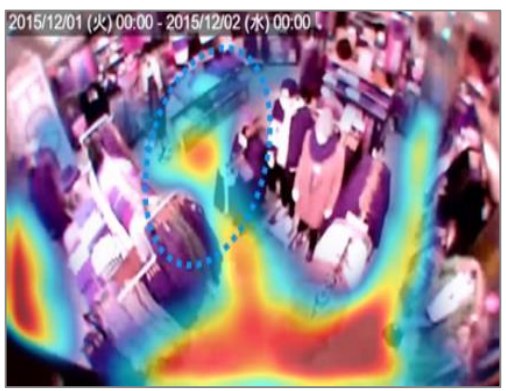
Access Control

Managing operations

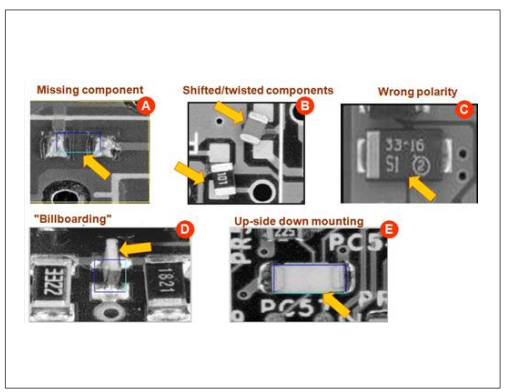
Parking Management



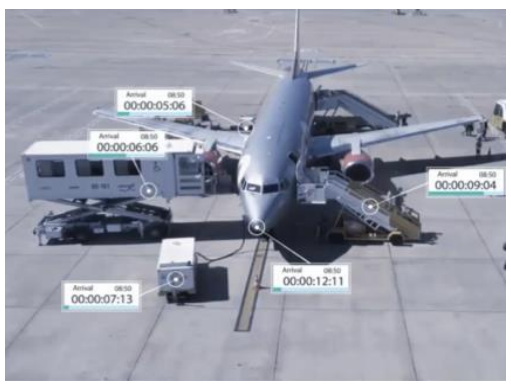
Traffic Engineering



Retail Analytics



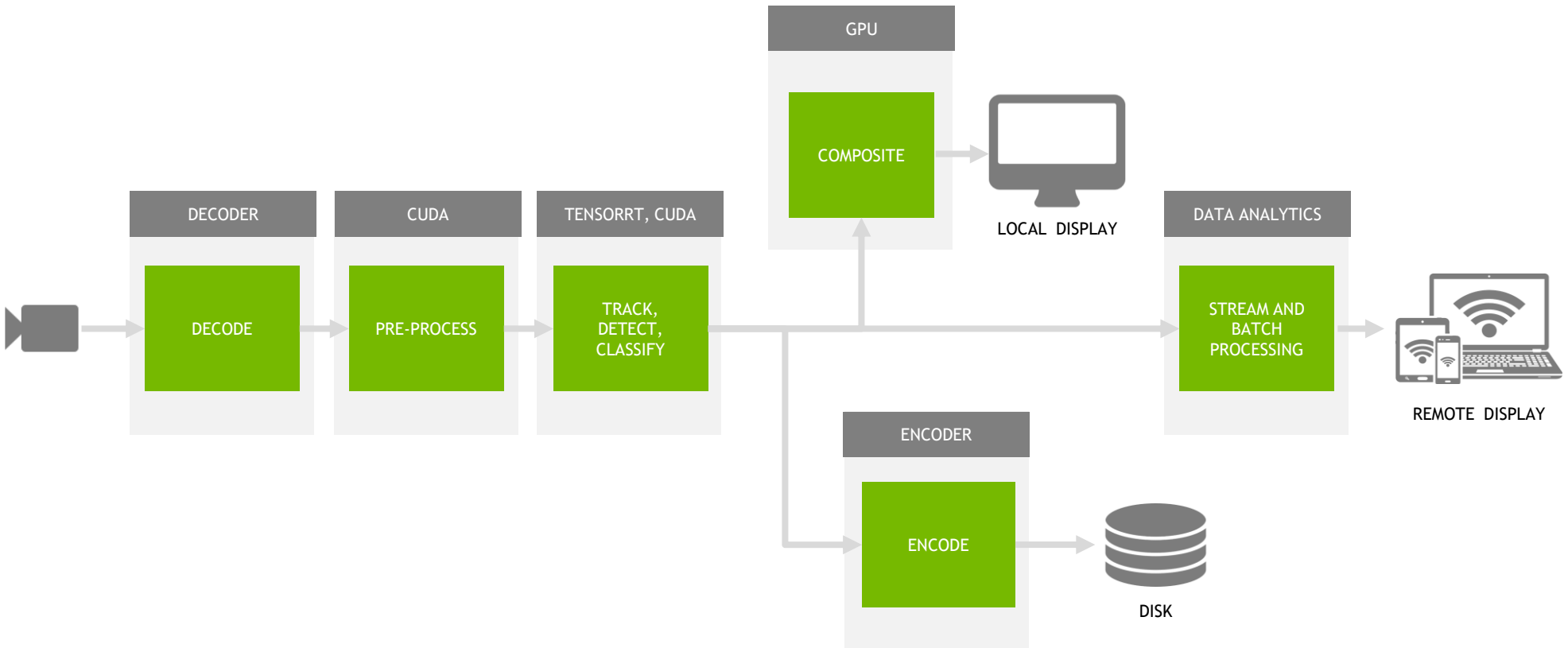
Optical Inspection



Managing Logistics

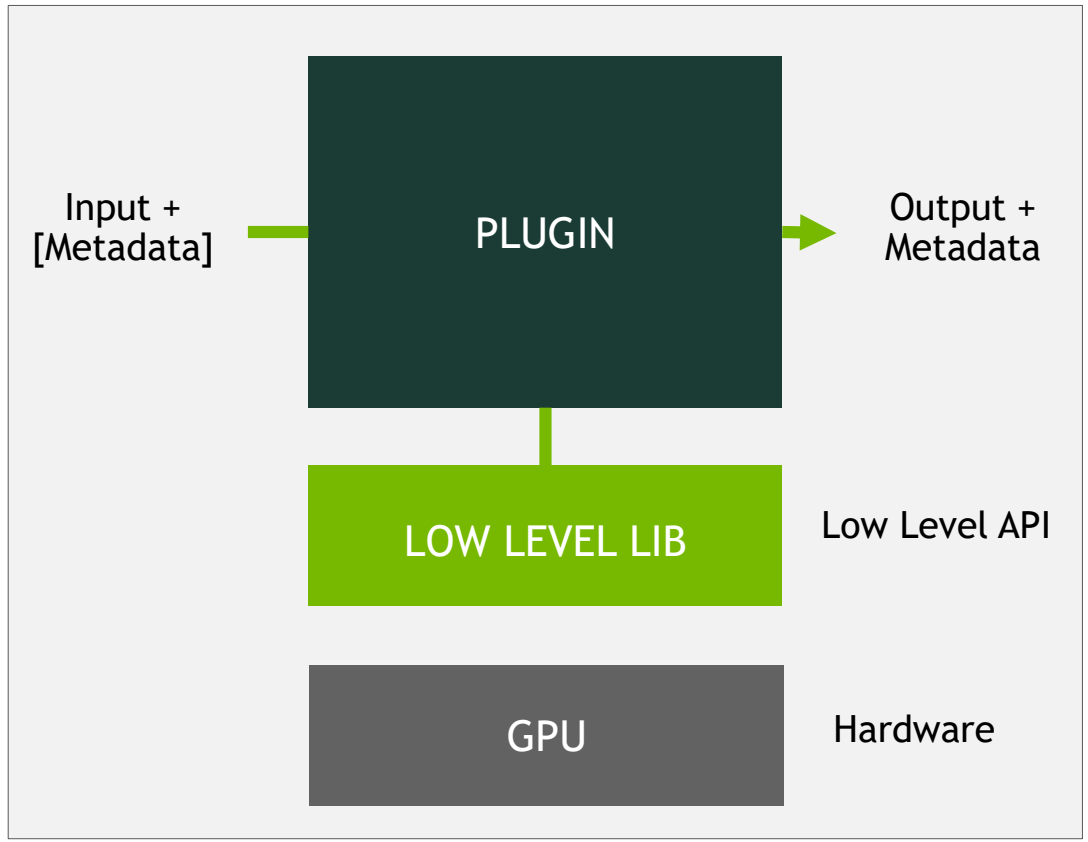
Content Filtering

REALTIME VIDEO UNDERSTANDING



DEEPSTREAM BUILDING BLOCK

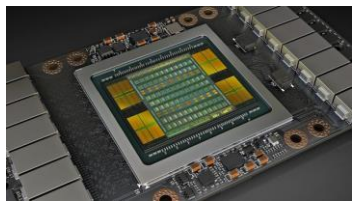
- A plugin model based pipeline architecture
- Graph-based pipeline interface to allow high-level component interconnect
- Heterogenous processing on GPU and CPU
- Hides parallelization and synchronization under the hood
- Inherently multi-threaded



WHAT'S NEW IN DEEPSTREAM 3.0

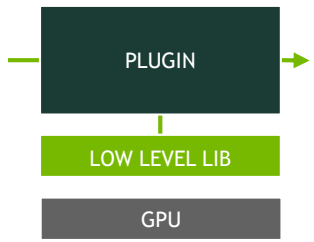
S9545 - Using the DeepStream SDK for AI-Based Video Analytics

LATEST GPUs - TESLA T4



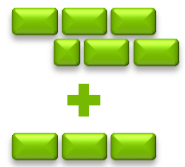
TensorRT 5, CUDA 10

NEW PLUGINS



Increased capability and throughput

DYNAMIC STREAM MANAGMENT



Add, remove, modify streams on the fly

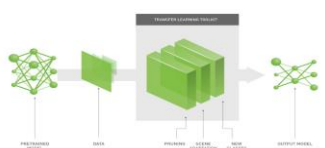
CONNECT EDGE TO CLOUD

Stream and Batch Analytics on Metadata

EASY TO SCALE AND MANAGE

Deploy in Docker Containers

HIGH EFFIENCY AND THROUGHPUT WITH TLT

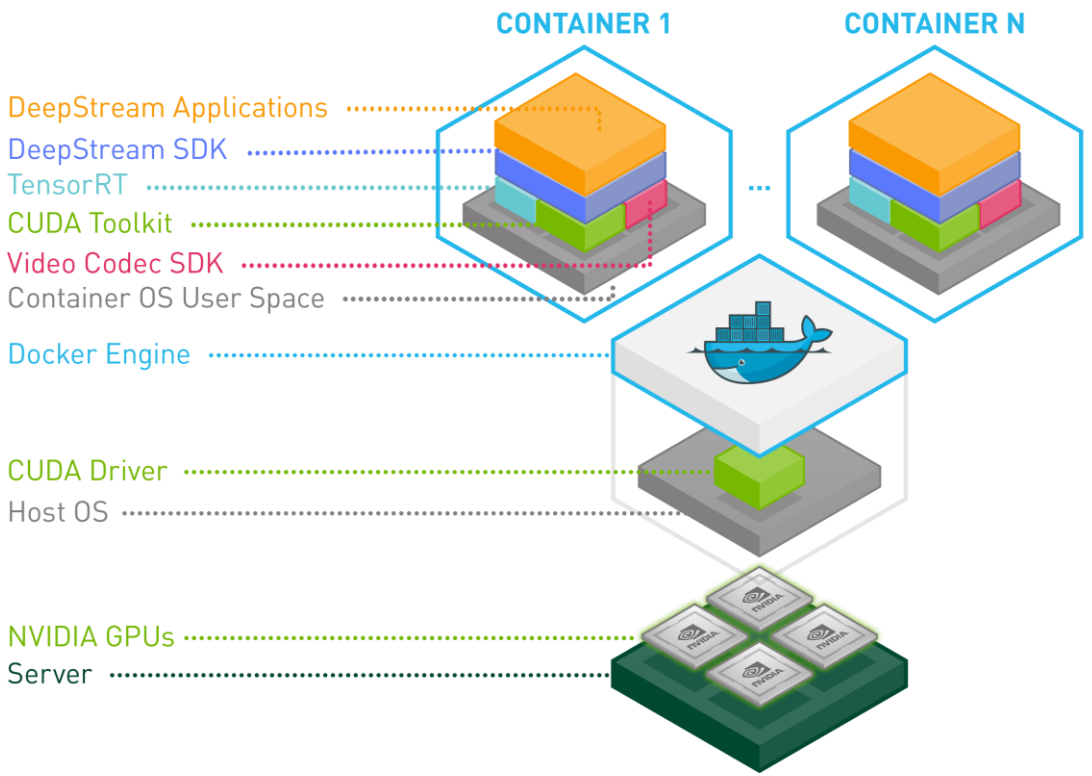


TLT model files are plug-n-play

https://developer.nvidia.com/deepstream-sdk

NVIDIA DEEPSTREAM IN DOCKER

IVA in Containers



https://developer.nvidia.com/deepstream-sdk

GPUS IN DATACENTERS

VFIO Passthrough

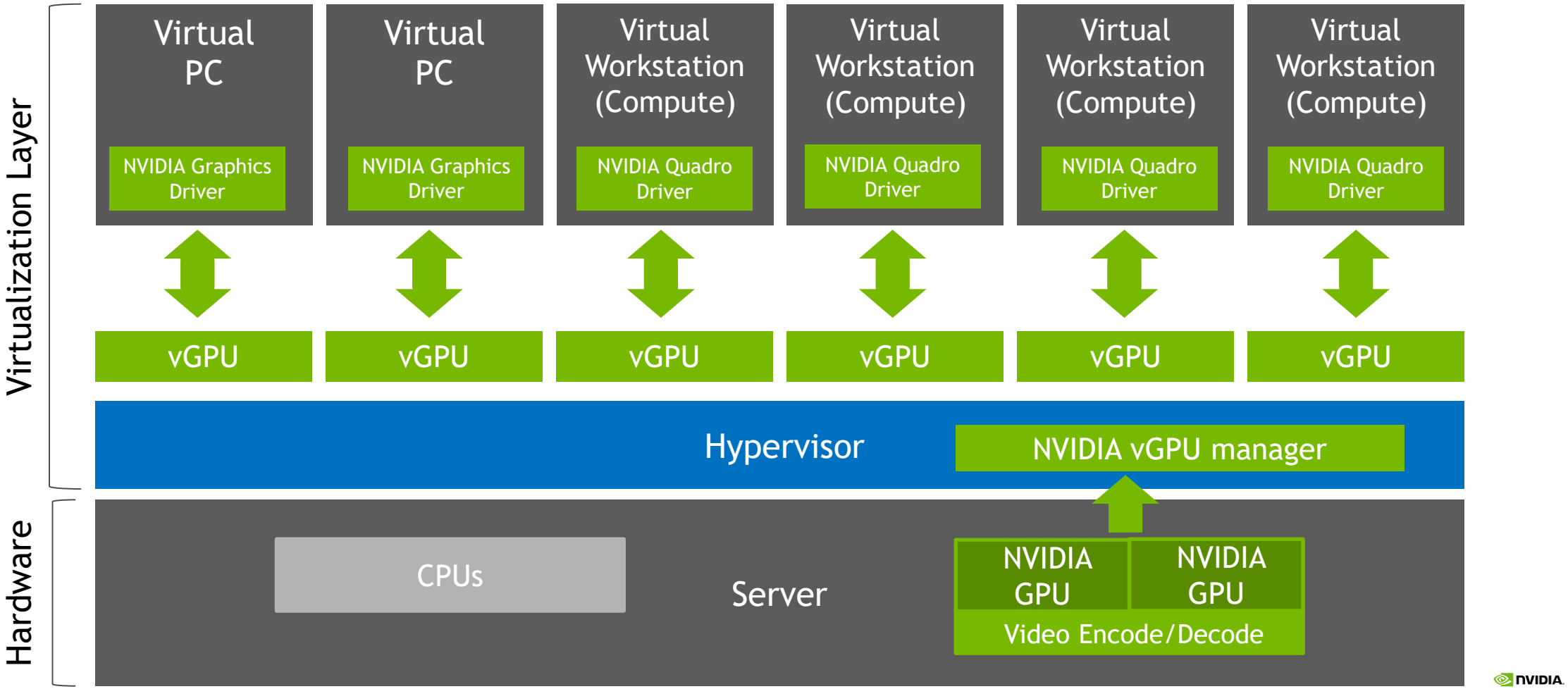
- Allows VM complete access to the PCIe hardware
- Best Performance available to the Virtual Machine
- Poor user density - Limited by the number of PCIe devices on the Server
- Under utilization of resources - No flexibility

WHAT IS VGPU?

- **Sharing GPU**: Sharing a GPU between multiple Virtual Machines
- **Direct Communication Path**: Provides a direct communication path from the drivers inside the guest Virtual Machine to the GPU
- **IO Device Virtualization**:
  - SR- IOV (Single-Root Input/Output Virtualization): HW-based partitioning of GPU using Virtual Functions
  - Mediated Devices (non SR-IOV): Requires vendor specific drivers to mediate sharing of resources between the Virtual Machines, which is what vGPU technology uses

HOW DOES NVIDIA VGPU WORK?

NVIDIA vGPU Architecture



WHY SHARE A GPU?

- **Utilization of Resources:**
  - GPU has traditionally been allocated to a particular user
  - Dedicating GPU leads to poor utilization of resources in datacenters
  - 100% of GPU resources are required for training in most networks
  - For inferencing, especially lower batch sizes, GPU is not utilized completely and sharing the GPU makes sense

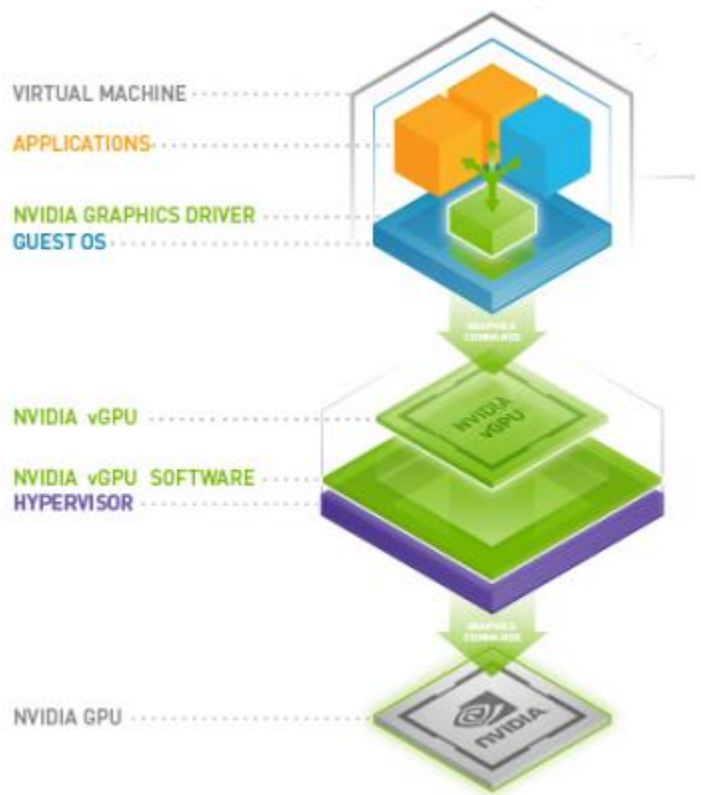
	Avg Images/s		
Tensor RT Inferencing - NGC 18.12	T4-16Q 410.49	T4 BM 410.79	T4 vGPU vs BM
Model:ResNet50 Python: 2 Precision: FP32 bs:1	309	337	-8%
Model:ResNet50 Python: 2 Precision: FP32 bs:2	394	427	-8%
Model:ResNet50 Python: 2 Precision: FP32 bs:4	444	484	-8%
Model:ResNet50 Python: 2 Precision: FP32 bs:8	482	521	-8%
Model:ResNet50 Python: 2 Precision: FP32 bs:16	505	561	-10%
Model:ResNet50 Python: 2 Precision: FP32 bs:32	495	553	-11%
Model:ResNet50 Python: 2 Precision: FP32 bs:64	518	575	-10%
Model:ResNet50 Python: 2 Precision: FP32 bs:128	515	580	-11%
Model:ResNet50 Python: 2 Precision: FP32 bs:7ms Target Latency	380	447	-15%

- **Idle Time**: It is estimated that GPUs are not being used as effectively because of idle time between two subsequent workload.

- **Efficiency**: Even in use cases where two people need concurrent access to one GPU, the resource utilization can be made much more effective

NVIDIA DEEPSTREAM IN VGPU

Benefits of IVA in VMs



Security: workload and user isolation

Maximum GPU utilization

- Fractional GPU
- Flexible vGPU Profile Configurations: GPU memory statically partitioned & given to VM
- Flexible vGPU Scheduler Configurations:
  - Fixed share scheduler
  - Equal share
  - Best effort

Multi-tenancy

Multiple vGPU

Mixed workloads

IVA USING VGPU – PROOF OF CONCEPT

OVERVIEW

Objective: Measure the scale performance of an inference based deep learning application in a virtualized setup using NVIDIA vGPU

Glossary:

- **GPU Inference Engine(GIE)**: TensorRT optimized pretrained neural networks.
- **Stream:** Source of images into the network - camera, video files, etc.
- **Configuration File:** Metadata information about the DeepStream Pipeline - # of GIEs, # of streams, etc.
- **Tracker:** DeepStream module with Kanade–Lucas–Tomasi feature tracker implementation - used for motion tracking of object in scenes

Setup:

System Configuration	Specification
CPU	Intel Xeon® E5-2620 v4 @ 2.10GHz x 32
GPU	NVIDIA Tesla P4
System Memory	256 GB DDR4, 2400 MHz
OS	Ubuntu 16.04
GPU Driver	411.95
CUDA	10.0
TensorRT	5.0 RC
GPU Clock Frequency	1113 MHz

NVIDIA VGPU PROFILES EXPLAINED

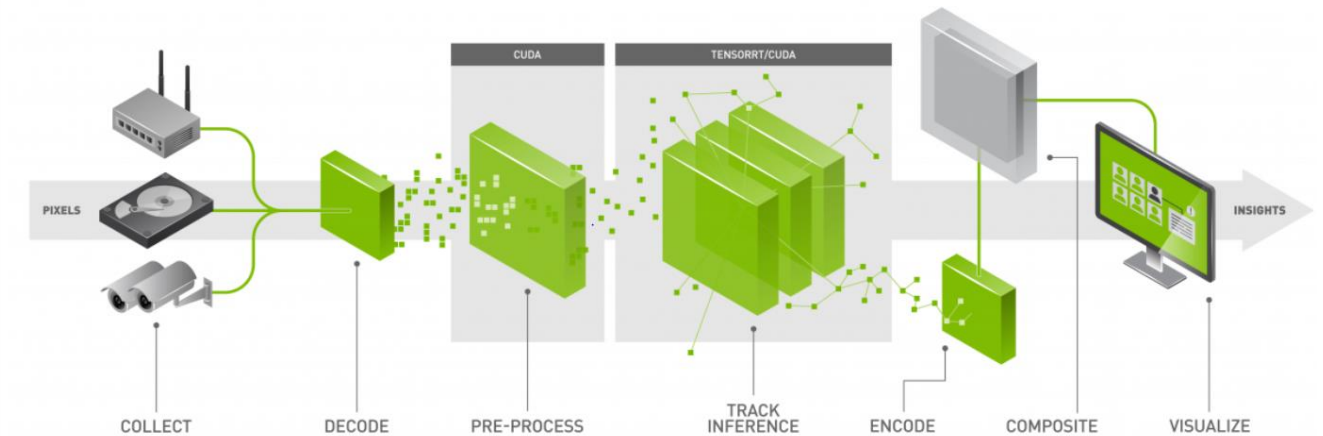
Q - Profiles	Quadro Features Enabled
B - Profiles	Basic Display Driver
A - Profiles	Application Profile

1Q is 1 Gig of Frame Buffer
2B is 2 Gig of Frame Buffer
8A is 8 Gig of Frame Buffer

Xq/b/a → “X” is the amount of Frame Buffer on the Nvidia GPU card

EXPERIMENT DETAILS

End-to-end Workflow



- # of streams (camera sources) per VM decreases with respect to the scale on a single GPU due to vGPU memory bottleneck
  Examples:
  - VM with an 8Q profile (8GB Frame Buffer) can run 26 streams before vGPU memory becomes a bottleneck
  - VM with a 2Q profile (2GB Frame Buffer) can run a maximum of 6 streams before vGPU memory becomes a bottleneck.

TEST PLAN

- **Compare:**
  - **Performance of the inference application on one, two, and four VMs** with 8Q, 4Q, and 2Q profiles respectively on a single Tesla P4
    VS.
  - **Performance of the application on a bare-metal server**
- **Scale:**
  - **Experiment on six Tesla P4 GPUs** to explore any CPU or memory bottlenecks to come up with a reference architecture number of Virtual Machines on a Inference server
- **GIEs:**
  - For the benchmarks use Resnet-10 and Resnet-18 backends
  - Batch size of each GIE set equal to the number of streams to ensure the output has an ideal 30 FPS
  - Any experiment which outputs the inference at less than 30 FPS was considered a degradation to ideal performance
- **Tracker Module:**
  - Ran tests with tracker module ON for CPU and turned OFF to avoid high CPU utilization on the host

TEST PLAN
(continued)

- A Deep Stream application on a VM with a 2Q vGPU profile can run inference on a maximum of 6 streams with 4 GIEs.
- If total number of GIEs per VM were fixed, regardless of the scale, it would then have a multiplicative effect on the number of GIE processes at scale, while keeping the total batch size on all GIEs across all the VMs

vGPU Profile (one GPU)	# of VMs	GIEs per VM	Streams per VM	Total Batch Size of all GIEs Summed up Across all VMs	Total # of GIE Processes Summed Across all VMs
P4-8Q	1	4	26	104	4
P4-4Q	2	4	12	96	8
P4-2Q	4	4	6	96	16

RESULTS

Performance Tracker ON

Bare-metal Baseline showed a performance of Real Time 30 FPS on all 26 streams with about 27% CPU utilization and approximately 63% GPU utilization

# of VMs	# of GPUs	vGPU Profile	CPU Util	GPU Util	GPU Mem Util per VM	FPS (Ideal 30)	# of Streams per VM	# of GIEs per VM	Batch Size Across GIEs-VMs	Max # of Streams per VM (GPU Mem Bottleneck)
1	1	P4-2Q	9.26	27	1505	30	6	4	24	6
2	1	P4-2Q	38.12	55	1505	30	6	4	48	6
3	1	P4-2Q	36.00	87	1505	30	6	4	72	6
4	1	P4-2Q	71.53	100	1505	23	6	4	96	6
1	1	P4-4Q	16.83	48	2338	30	12	4	48	16
2	1	P4-4Q	46.86	99	2338	30	12	4	96	16
1	1	P4-4Q	16.59	51	2612	30	14	4	48	16
2	1	P4-4Q	36.12	99	2612	26	14	4	96	16
1	1	P4-4Q	23.56	76	5077	30	26	4	104	26

RESULTS - OBSERVATIONS

Performance Tracker ON (continued)

- **Performance held to a steady 30 FPS when GPU was shared between 2 VMs** (each with a 4Q vGPU profile)
- Therefore **able to run 24 streams with 8 GIEs between the 2 VMs**, which was a step above running the workload on 1 VM using the entire GPU (with an 8Q vGPU profile with 26 streams and 4 GIEs)
- Resulting in a scale sweet spot for this application configuration

- **Observed frame drop when scaled to 4 VMs** (each with a 2Q profile) on a single GPU because 16 GIEs were running on 4 VMs on the GPU as opposed to 4 GIEs on a single VM with the whole GPU (on an 8Q profile), although the total batch size for all the GIE summed up across VMs remains the same

RESULTS

Performance Tracker OFF

- CPU utilization much lower with the tracker OFF
  - For all purposes of scale is recommended to be turned OFF
- No difference in GPU utilization as tracker process currently runs on CPU

# of VMs	# of GPUs	vGPU Profile	CPU Util	GPU Util	GPU Mem Util per VM	FPS (Ideal 30)	# of Streams per VM	# of GIEs per VM	Batch Size Across GIEs-VMs	Max # of Streams per VM (GPU Mem Bottleneck)
1	1	P4-2Q	4.5	35	1505	30	6	4	24	6
2	1	P4-2Q	10.56	68	1505	30	6	4	48	6
3	1	P4-2Q	22.33	86	1505	30	6	4	72	6
4	1	P4-2Q	49.41	99	1505	25-27	6	4	96	6
1	1	P4-4Q	6.2	45	2338	30	12	4	48	16
2	1	P4-4Q	15.44	99	2338	30	12	4	96	16
1	1	P4-4Q	7.63	53	2612	30	14	4	48	16
2	1	P4-4Q	19.2	99	2612	23-26	14	4	96	16
1	1	P4-4Q	20.13	76	5077	30	26	4	104	26

PERFORMANCE AT SCALE

- Recommendation: Best vGPU profile for this video streaming workload is P4-4Q based on prior test
- Scaling Trial: Run 12 DeepStream Virtual Machines (each 4Q) to test CPU and memory bottlenecks

Setup:

vGPU Profile	P4-4Q
# of Streams (720p 30fps)	12
# of GIEs	1* Primary (Resnet-10) + 3* Secondary (Resnet-18)
Batch Size	12 (Primary GIE – 640x368 Images) (Secondary GIE – 224*224 Images)
Tracker	OFF
GPU Mem Util	2146 MB

Results:

# of VMs	# of GPUs	CPU Util	GPU Util	FPS (Ideal 30)
12	6	92	93,91,92,95,91,92	29-30
11	6	84	93,91,54,90,95,90	29-30
10	5	75	90,91,87,90,90	29-30
9	5	60	90,91,48,90,90	29-30
8	4	49	90,91,84,90	29-30
7	4	38	90,91,44,90	29-30
6	3	27	92,93,91	29-30
5	3	21	90,86,46	29-30
4	2	19	93,97	29-30
3	2	16	92,47	29-30
2	1	14	96	29-30
1	1	11	57	29-30

CONCLUSION

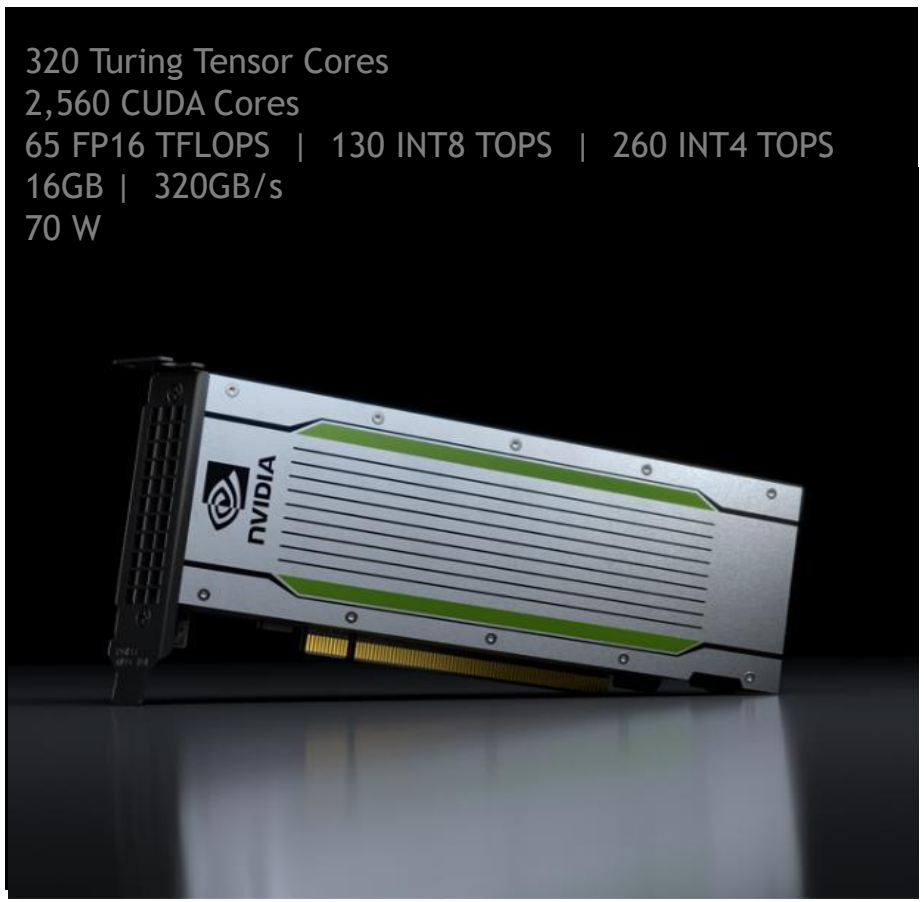
Virtualization Benefits with Minimum Performance Impact

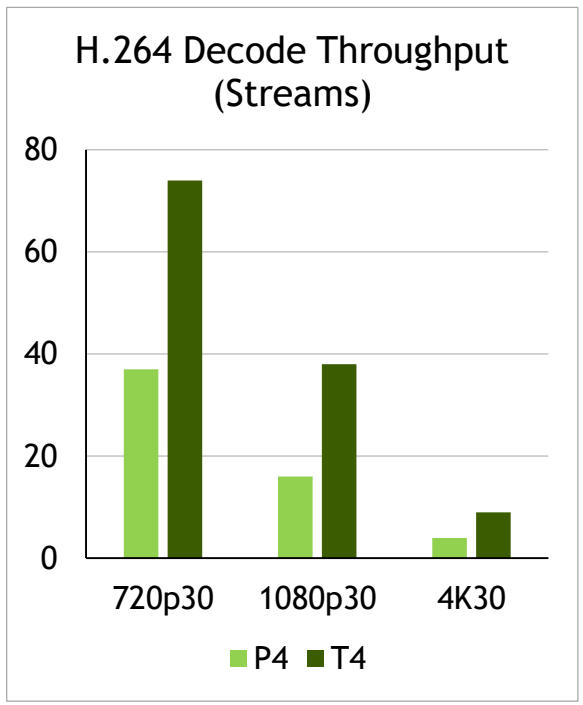
- **Minor to No Performance Impact:** Virtualizing the example inference workload had no performance impact on an 8Q profile
  - 8Q profile with 26 streams with the 4 GIEs specified in the application configuration ran inference at 30 FPS
  - Minor performance impact of about two streams at approximately 10% when workload split between 2 VMs on a 4Q profile, with trade-off of multi tenancy
- **Multi-tenancy:** Running inferencing on a virtualized environment can provide multi tenancy on a single GPU
- **Efficient use of resources:** If inference process does not use entire GPU then multiple processes can be run on separate VMs sharing the same GPU
  - However, increasing scale beyond a point where the GPU utilization hits the maximum would increase inference times on each VM sharing the GPU

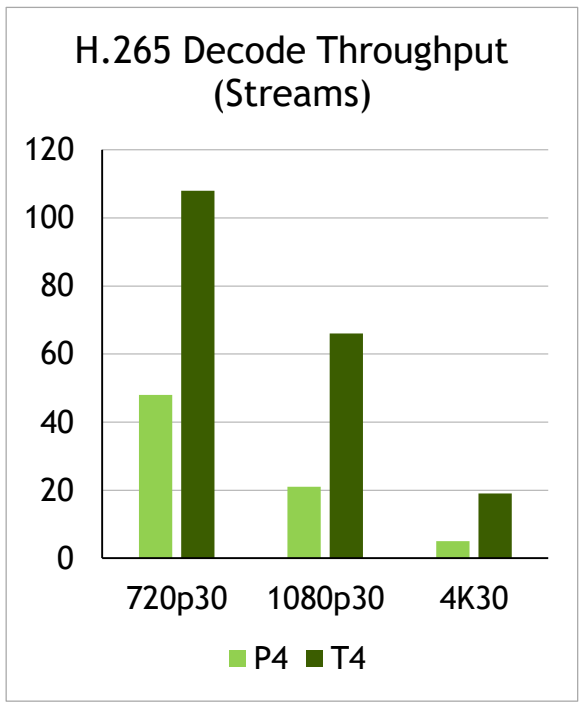
WHAT'S NEXT

NVIDIA T4 UNIVERSAL INFERENCE ACCELERATOR

320 Turing Tensor Cores
2,560 CUDA Cores
65 FP16 TFLOPS | 130 INT8 TOPS | 260 INT4 TOPS
16GB | 320GB/s
70 W

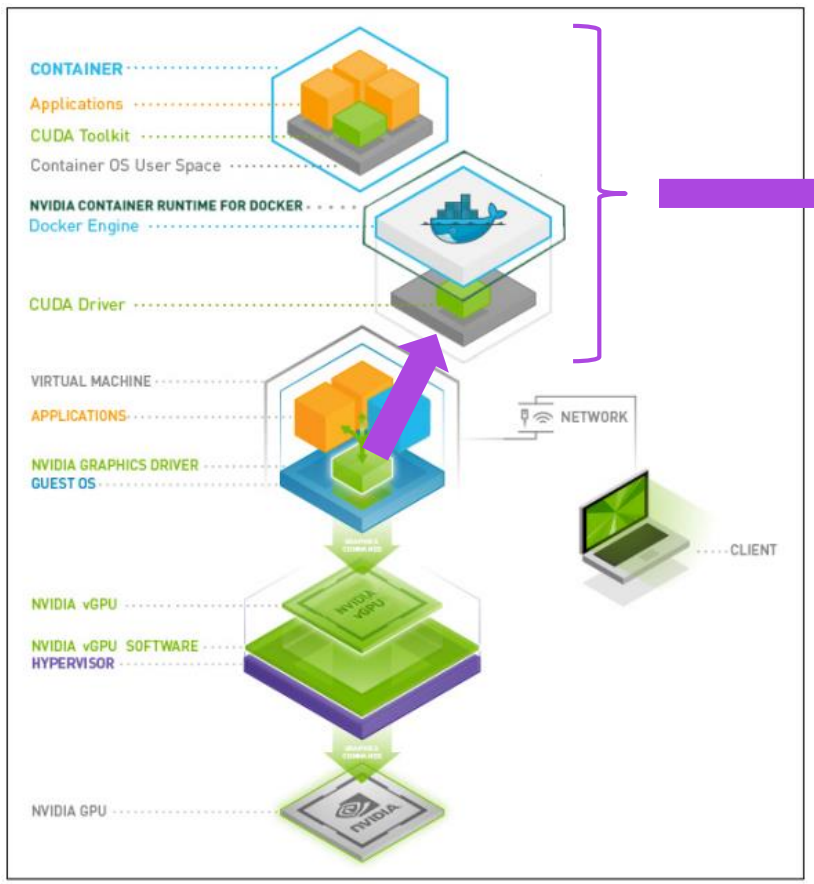


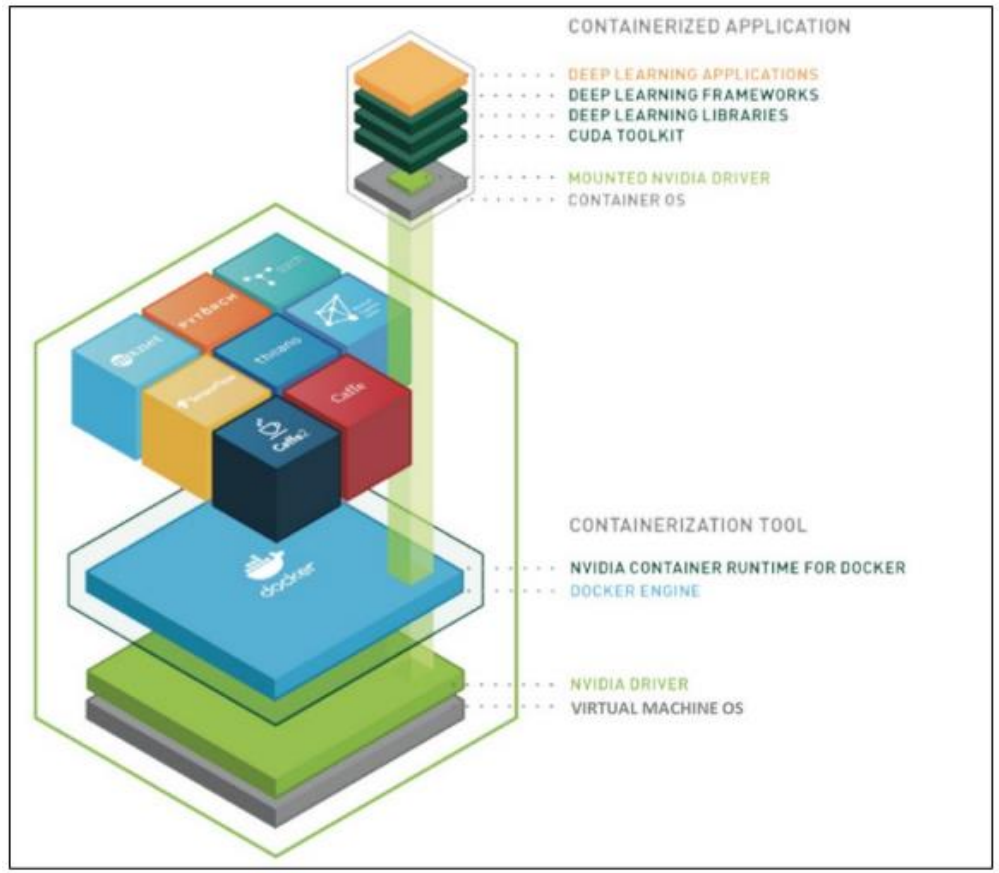




NVIDIA DEEPSTREAM IN DOCKER VIRTUALIZED

Mixed Workloads on vGPU - IVA in Containers Running in VMs?






RESOURCES

NVIDIA VIRTUAL GPU RESOURCES

Video Analytics Using NVIDIA vGPU Whitepaper
available upon request

Mixed Workloads Reference Design Guide
https://www.nvidia.com/content/dam/en-zz/Solutions/data-center/gated-resources/mixed-workloads-reference-design-guide.pdf

DeepStream
https://developer.nvidia.com/deepstream-sdk

Virtual GPU Test Drive
https://www.nvidia.com/tryvgpu

NVIDIA Virtual GPU Website
www.nvidia.com/virtualgpu

NVIDIA Virtual GPU YouTube Channel
http://tinyurl.com/gridvideos

Questions? Ask on our Forums
https://gridforums.nvidia.com

NVIDIA Virtual GPU on LinkedIn
http://linkd.in/QG4A6u

Follow us on Twitter
@NVIDIAVirt


NVIDIA