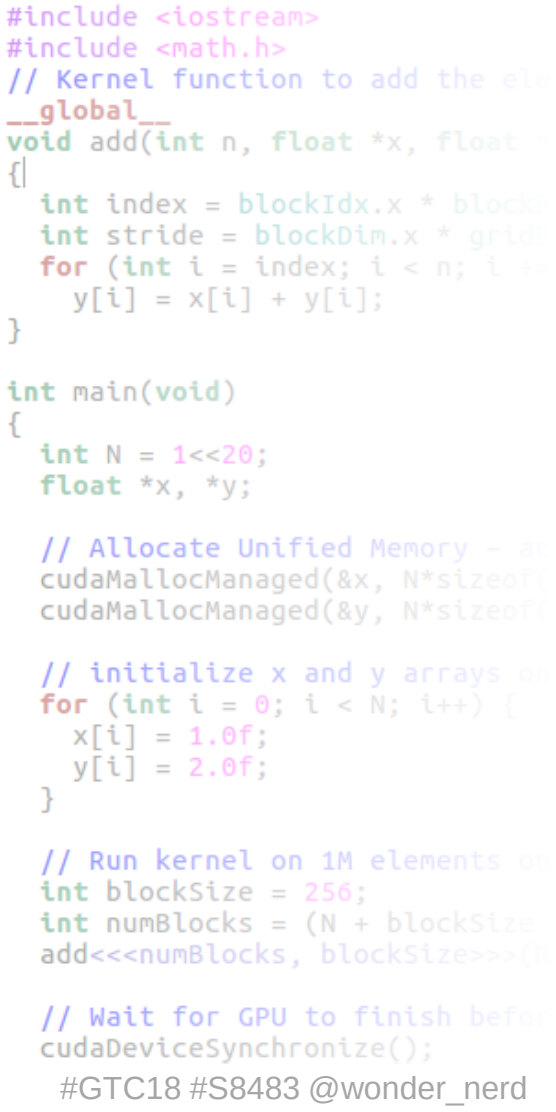
S8483 - Empowering CUDA Developers with Virtual Desktops

Tony Foster – Sr. Advisor, Technical Marketing, Dell EMC
VMware vExpert; VMware EUC Champion;
VMware Experts Program, BDSEW;
NVIDIA vGPU Community Advisor (NGCA)

@wonder_nerd www.wondernerd.net

V 4.0 Date:3-24-18

Agenda



Define the Technologies

Why do This?

Environment Overview

Deployment

Testing

Questions

Resources

1drnrd.me/blog More

Slides Available at:
www.wondernerd.net
(in 20 minutes)

What is CUDA and Virtualization

CUDA

- Provides a development environment for creating high performance GPU-accelerated applications.

Virtualization

- Takes physical computing resources and divides them up among virtual machines

Virtual GPU (vGPU)

- Provides a shared instance of a GPU to a virtual machine, delivering resources of the underling physical GPU to the virtual machine, such as graphics processing or CUDA.

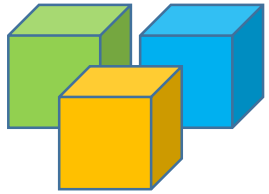
Why I Did This

1drnrd.me/blog More

- Cool part of the job – pushing technology further
- Limited resources in my home lab
  - 1 - P4 GPU
  - $1/Day power consumption
  - Happy Wife
- Multiple Code Branches
- Multiple Projects
- Easy to Change OS

In The Real World Why?

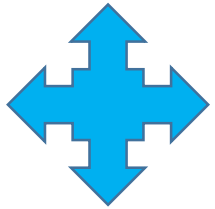
Resource Optimization



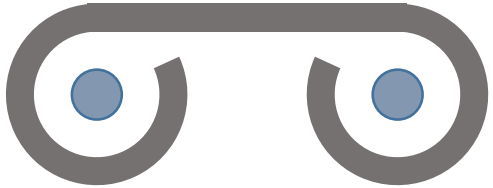
Multiple Workspaces

Security

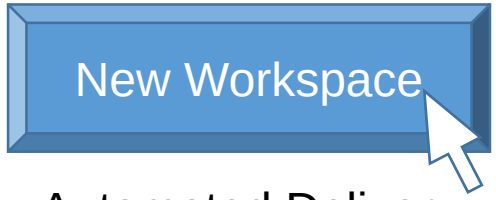
Version Control



Resource Sharing



Backup / DR



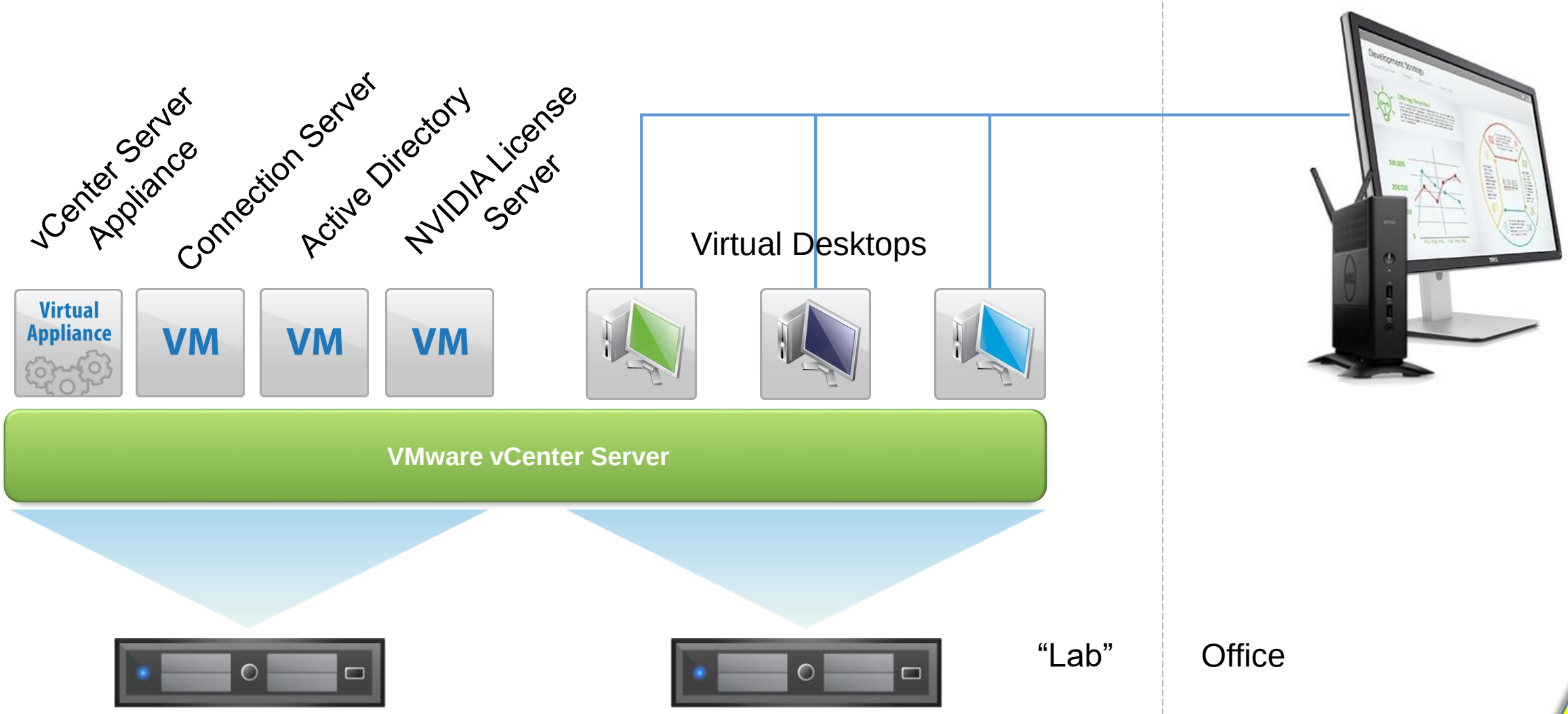
Automated Delivery

Environment Overview

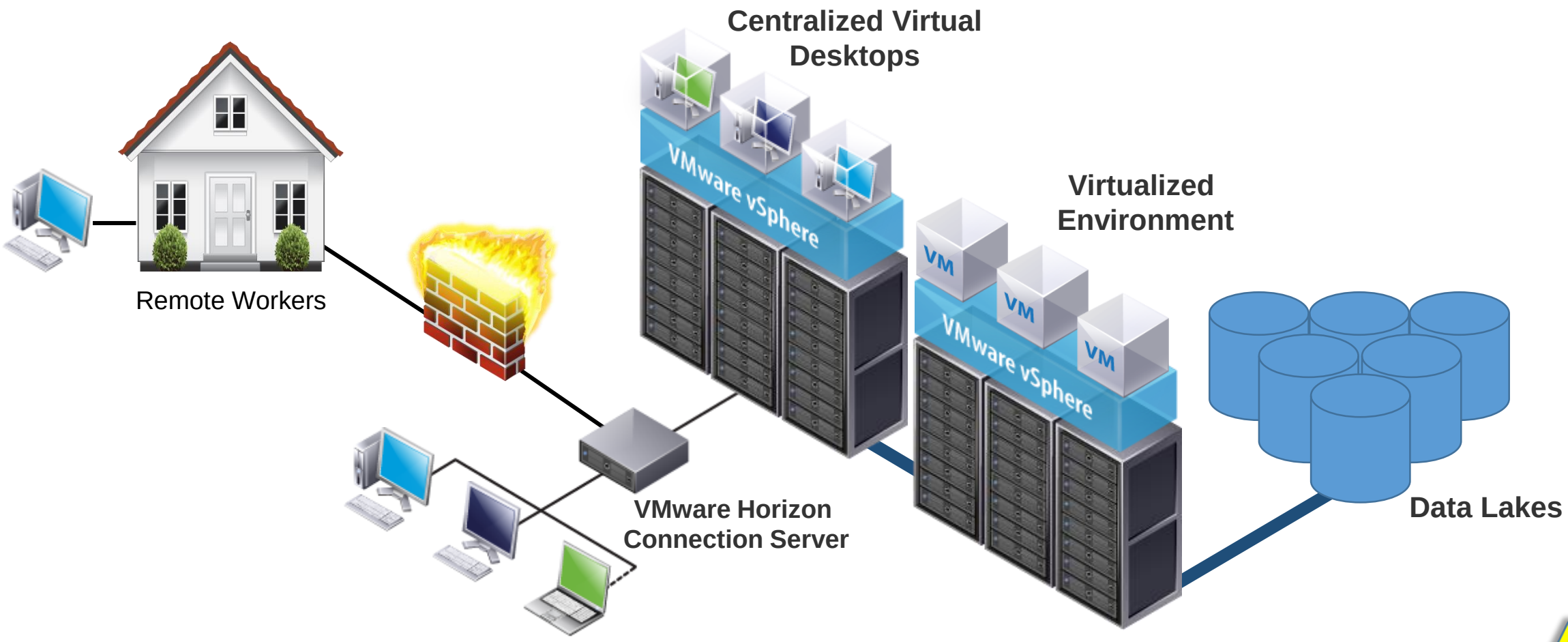
Requirements

- GPU (P4, P40, etc.)
- VMware Horizon
- Linux VM
- NVIDIA CUDA Toolkit
- NVIDIA Quadro vDWS, Virtual GPU Software License

My Virtual Environment



Scaling to the Organization



Hardware Specs

1drnrd.me/lab More

- Testing on 2U host
  - Dual E5-2640 – 6 Core Procs
  - 64GB of RAM
  - NVIDIA P4 @ 384.111
- VMware vSphere 6.5 (Build 7388607)
- vCenter Server Appliance 6.5.0 (Build 6.5.0.14100)
- VMware Horizon 7.4.0 (Build 7400497)
  - Basic Environment Only
  - *Sub-optimal*

- Management environment on separate 1U host
  - vCenter Appliance
  - AD/DNS (Windows 2k8 R2)
  - Jump Box (Windows 2k8 R2)
  - NVIDIA GRID License Server (CentOS7.1 & Windows 2k8 R2)
  - vSphere Connection Server (Windows 2k8 R2)
- Horizon View Client running on Jump box

Unsupported Lab Configuration

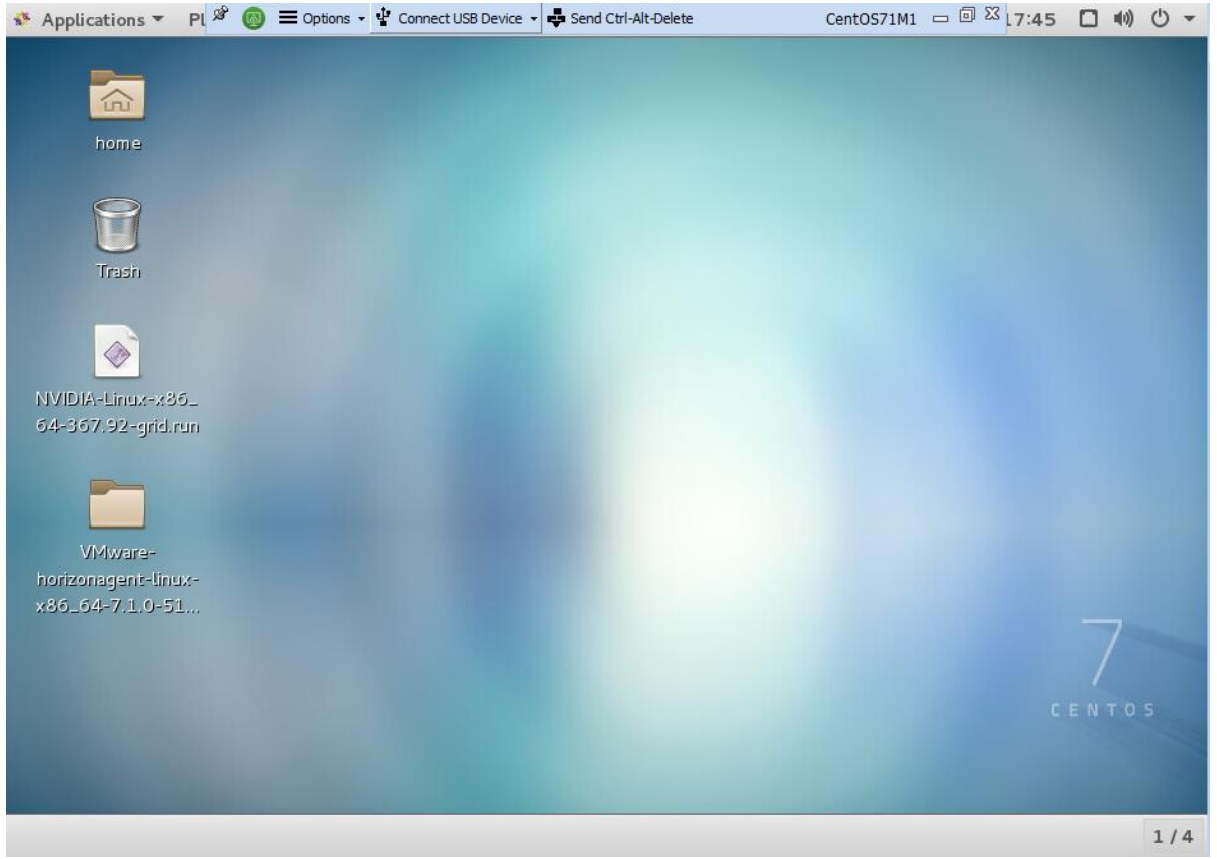
VM Specs

1drnrd.me/ubuntu More

- CentOS 7.1 (x64)
  - 4 vCPU
  - 12GB vRAM
  - VMware Blast Extreme protocol

  vGPU Profile
    - Quadro vDWS P4-4Q
    - Equal Share Scheduling
    - CUDA Toolkit 9.0.176

  Passthrough
    - NVIDIA P4 GPU
    - CUDA Toolkit 9.1.85



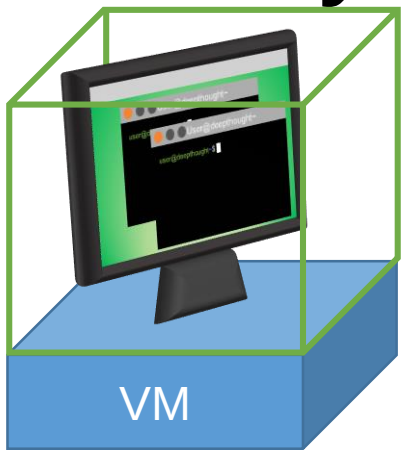
Flings

https://labs.vmware.com/flings/horizon-ova-for-ubuntu

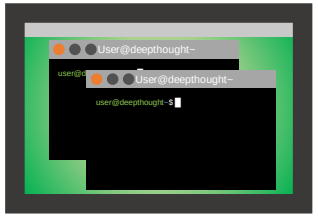
Deployment

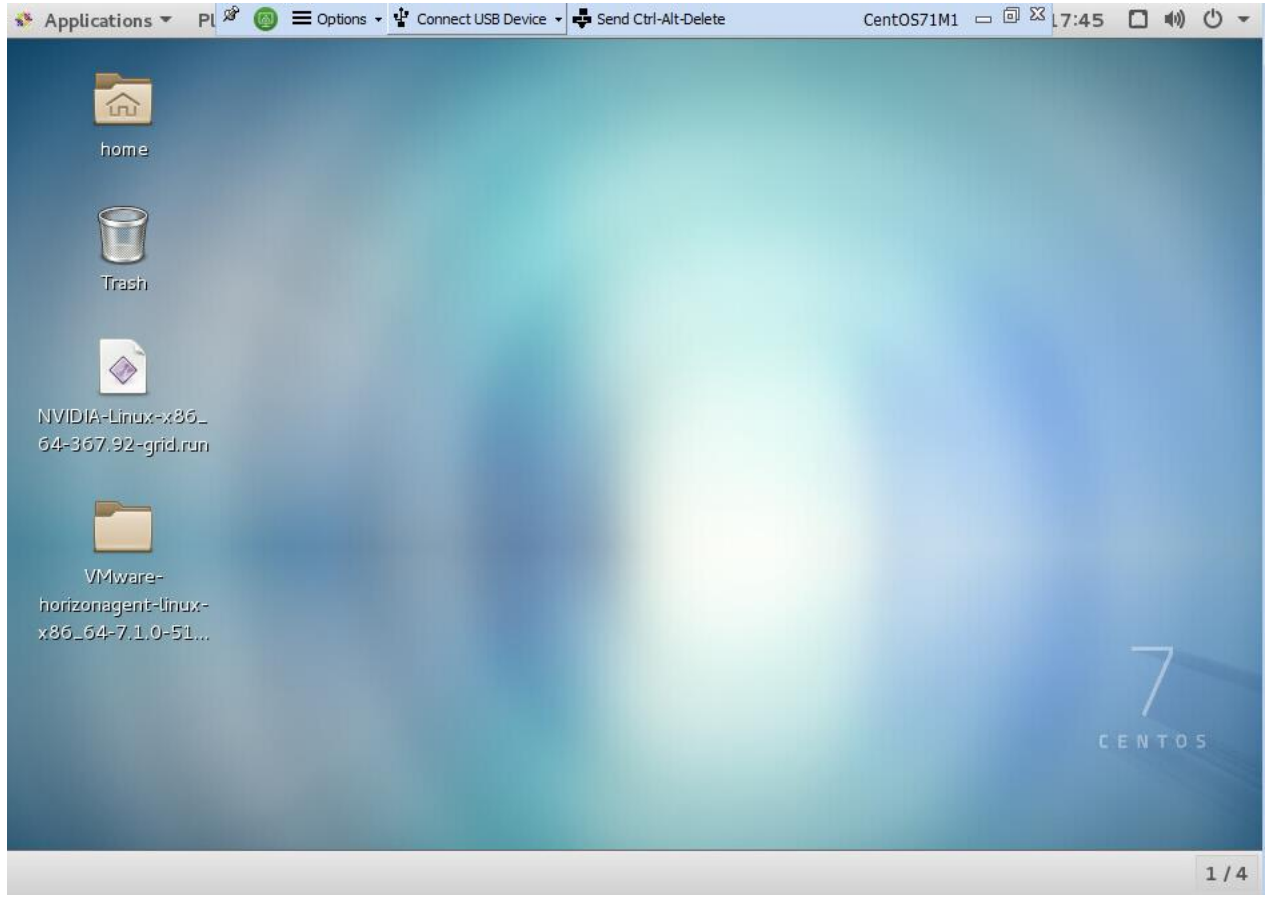
Why Horizon/VDI?

Traditional VMs

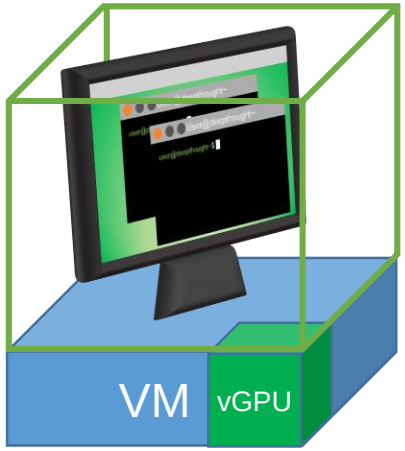


Virtual Display Console

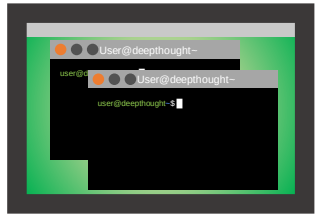




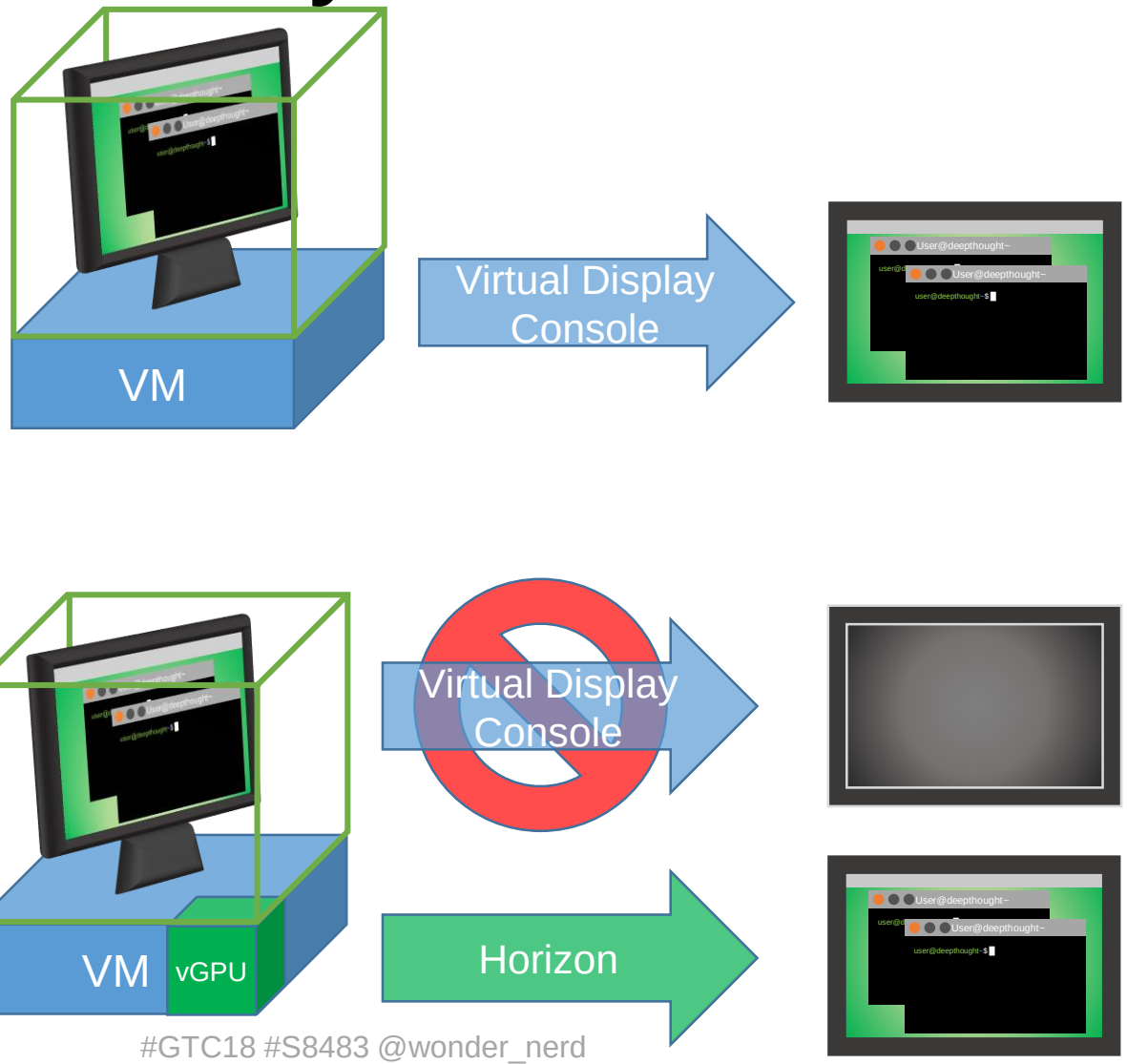
GPU Enabled VMs

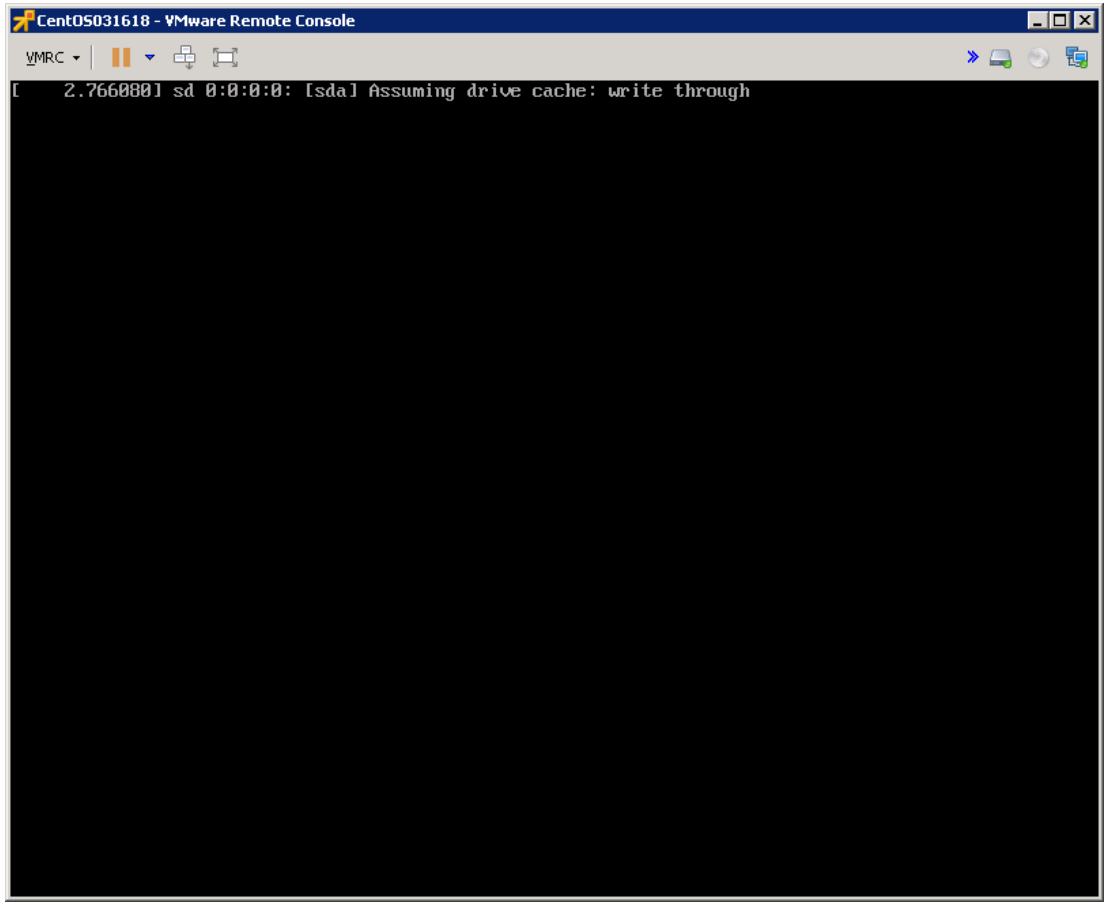


Horizon



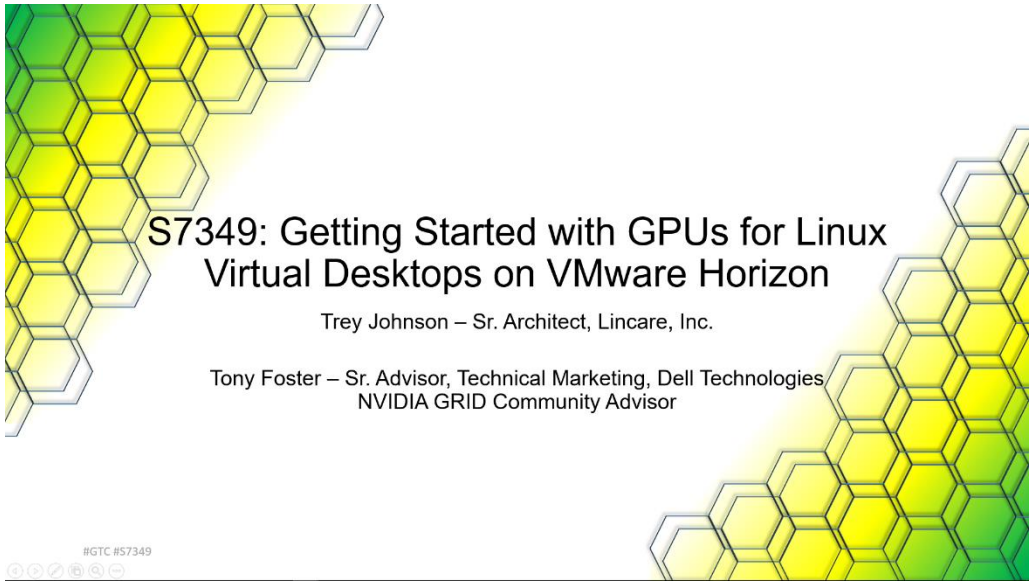
Why Horizon/VDI?





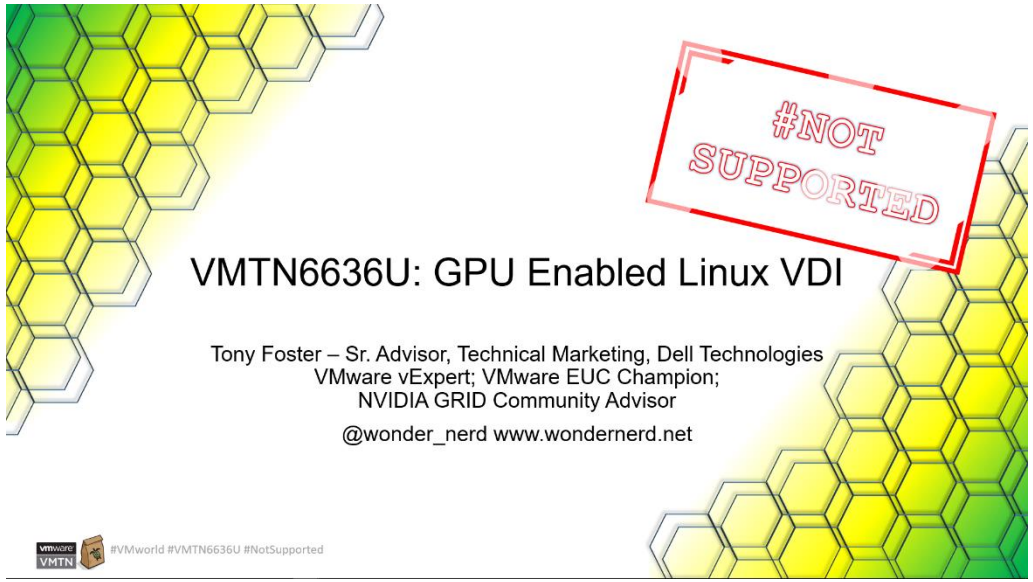
Preparing Hosts & VM

GTC17 Session S7349



1drnrd.me/S7349 More

VMworld Session VMTN6636U



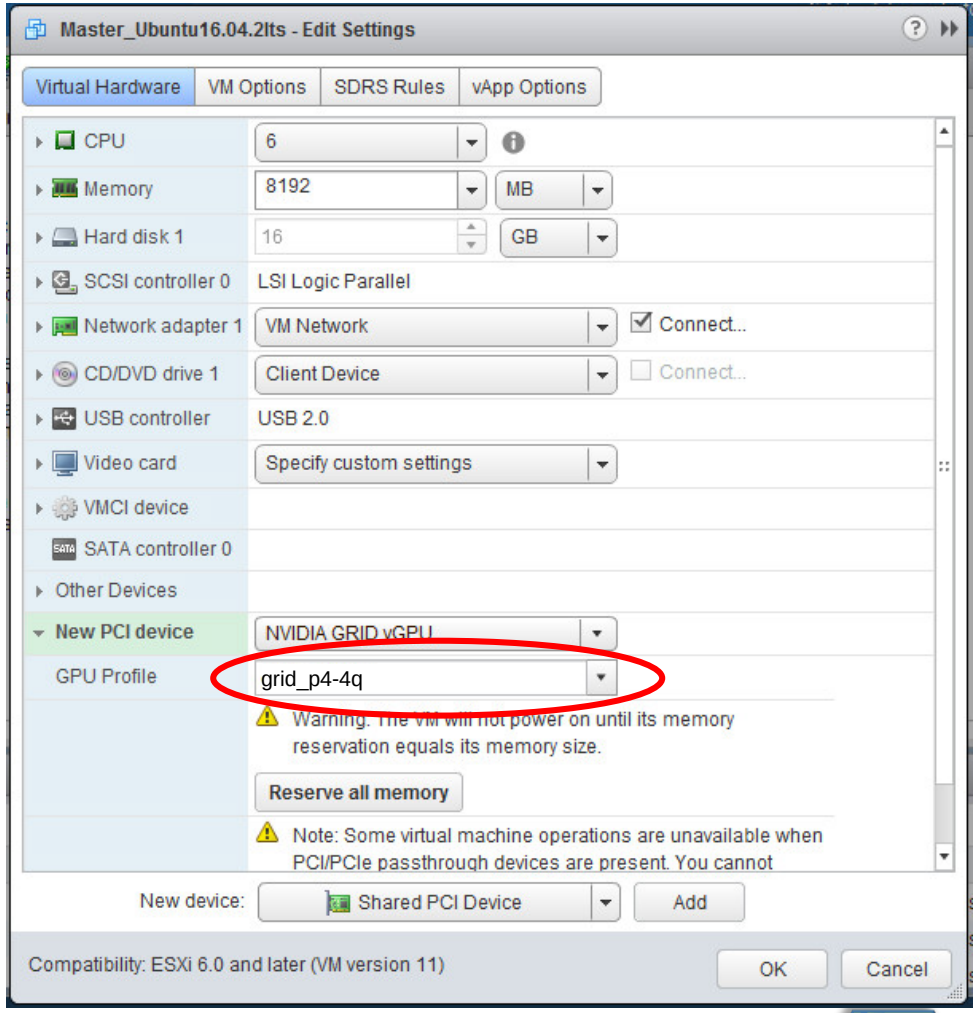
1drnrd.me/VMTN6636U More

Licensing

Requires NVIDIA Quadro vDWS

Examples:

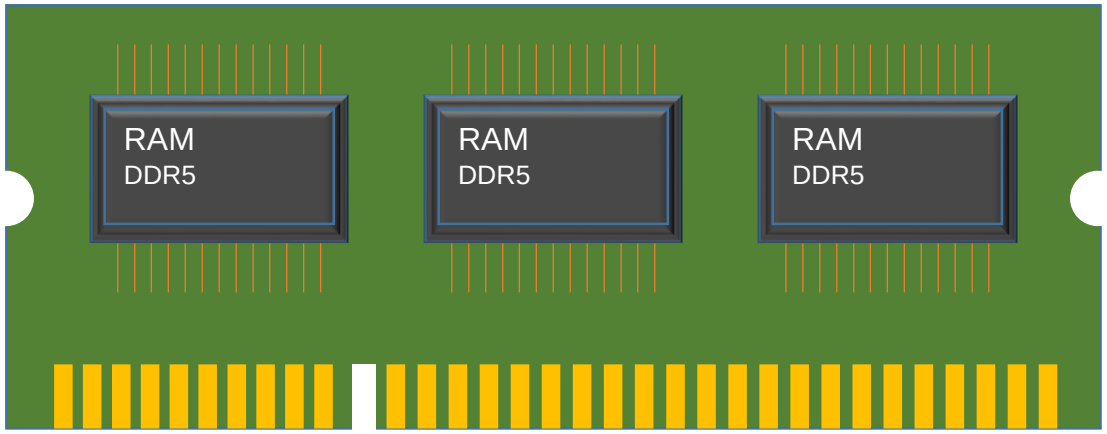
- P4
  - P4-8Q; P4-4Q; P4-2Q; P4-1Q
- P40
  - P40-24Q; P40-12Q; P40-8Q;
- P100
  - P100-16Q; P100-8Q
  - P100C-12Q; P100C-6Q



Two Parts of a vGPU

Memory

- “Frame Buffer”
  - vGPU Profiles



Streaming Multiprocessor (SM)

- Does the computation

vGPU Profiles

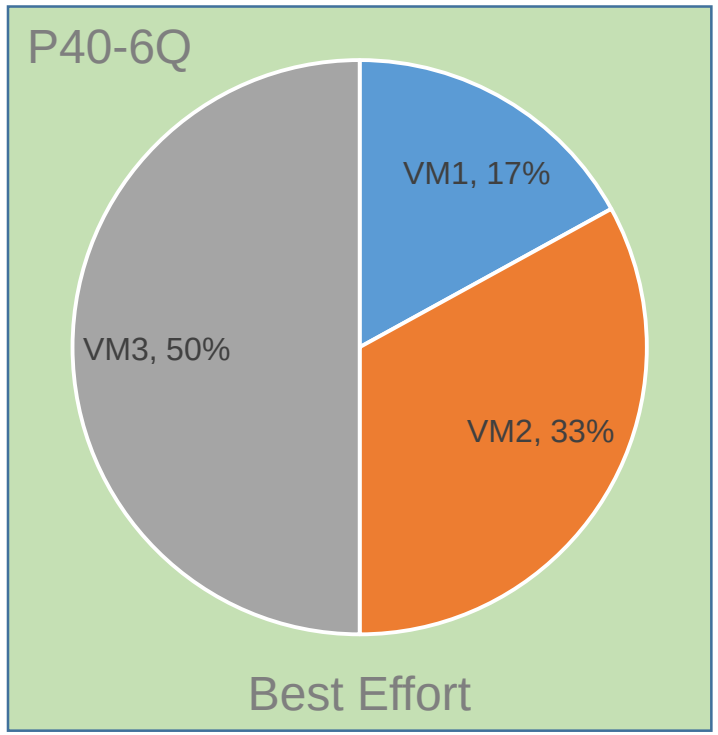
Profile	Frame Buffer (Mbytes)	Maximum vGPUs per Board	License Required
P40-24Q	24576	1	Quadro vDWS
P40-12Q	12288	2	Quadro vDWS
P40-8Q	8192	3	Quadro vDWS
P40-6Q	6144	4	Quadro vDWS
P40-4Q	4096	6	Quadro vDWS
P40-3Q	3072	8	Quadro vDWS
P40-2Q	2048	12	Quadro vDWS
P40-1Q	1024	24	Quadro vDWS

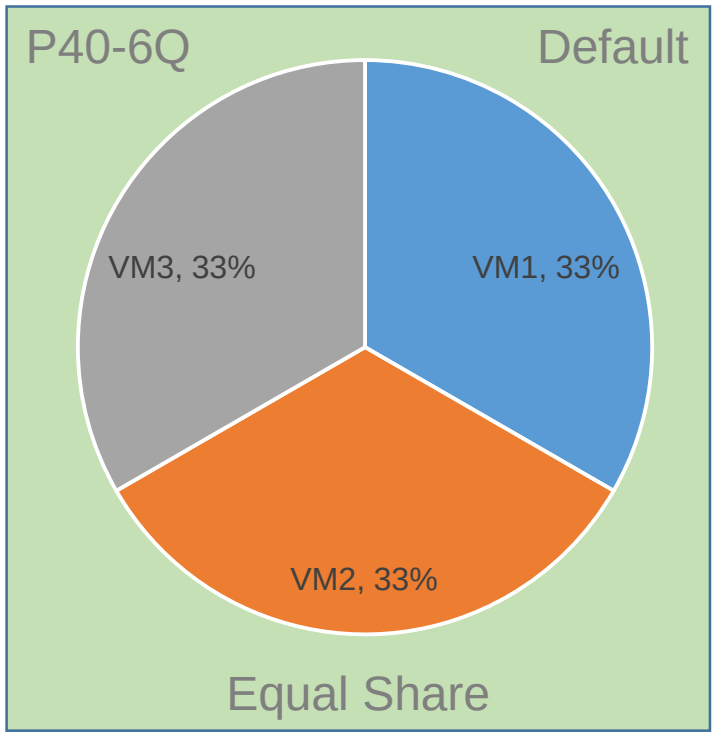
Frame Buffer = GPU Card Memory (24GB) ÷ vGPUs per Card

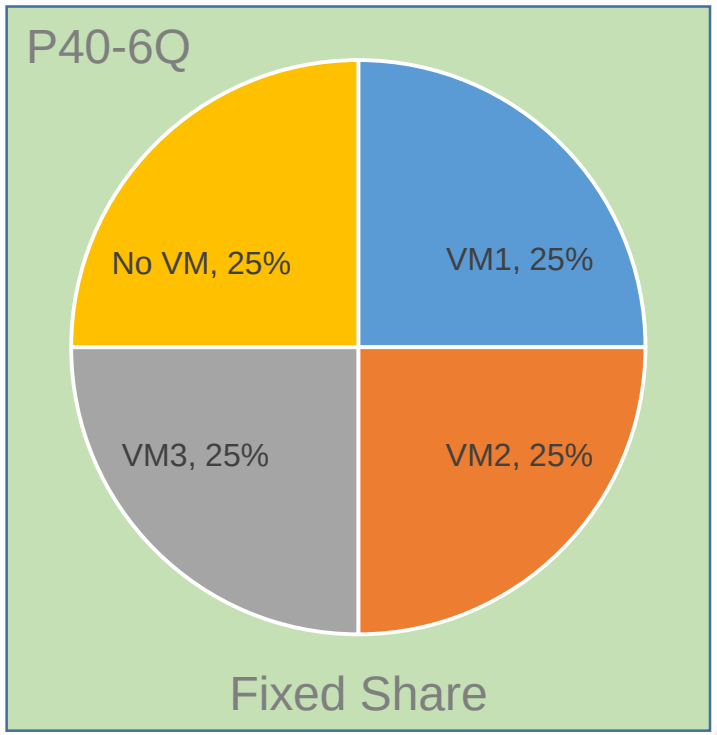
Scheduling vGPUs

1drnrd.me/GPUQoS More

Schedulers impose a limit on GPU processing cycles used by a vGPU, which prevents vGPU-intensive applications running in one VM from affecting the performance of vGPU-light applications running in other VMs. On GPUs based on the Pascal architecture, you can select the vGPU scheduler to use.







Configuring Scheduling

RmPVMRL Registry Key

1drnrd.me/scheduling More

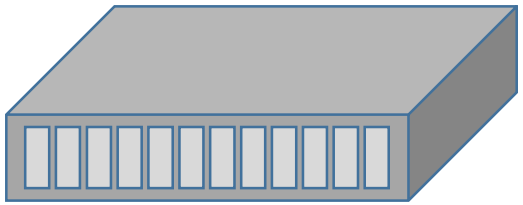
Value	Meaning	Usage
0x00	Best Effort Scheduler	
0x01	Equal Share Scheduler (Default)	Enterprise
0x11	Fixed Share Scheduler	Service Provider

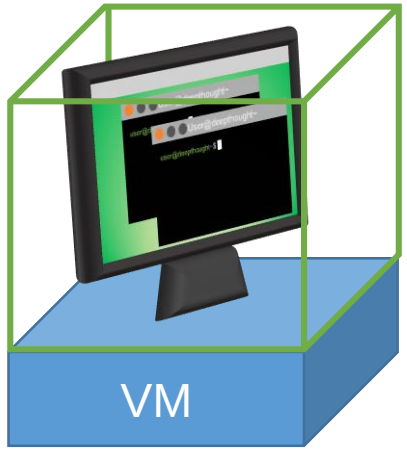
1. SSH to the ESXi host
2. Issue the following
   1. For all cards on a host:
      ```
      esxcli system module parameters set -m nvidia -p
      "NVreg_RegistryDwords=RmPVMRL=<value>"
      ```
   2. For individual cards on a host:
      1. List the GPUs in the host: `lspci | grep NVIDIA`
         Results in: `0000:85:00.0 VGA compatible…`
      2. Set the policy per card:
         ```
         esxcli system module parameters set -m nvidia \ -p
         "NVreg_RegistryDwordsPerDevice=pci=<pci-domain:pci-
         bdf>;RmPVMRL=<value>[;pci=<pci-domain:pci-
         bdf>;RmPVMRL=<value>][;...]"
         ```
   3. Reboot

vGPU Driver Requirements

1drnrd.me/vCUDAp1 More

- ***Must*** match between host and VM

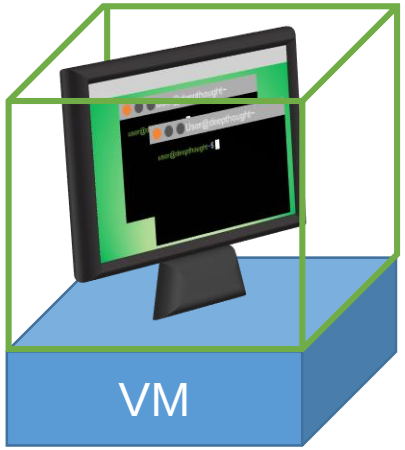




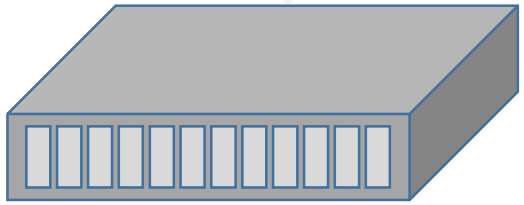
ESXi Host
- ✓ NVIDIA GPU P40
- ❑ GPU VIB **X.Y.Z**

Virtual Machine (Linux)
- ✓ NVIDIA Virtual GPU P40-8Q
- ❑ GPU Driver **X.Y.Z**

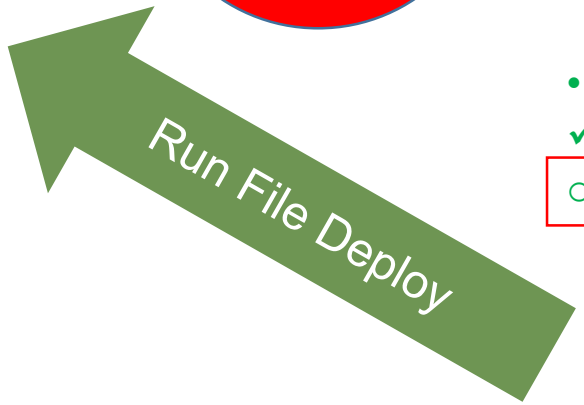
Two Methods to Install the CUDA Toolkit

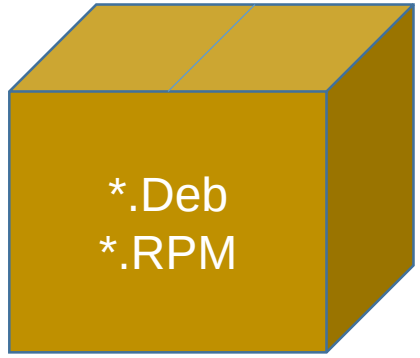


- Virtual Machine (Linux)
- ✓ NVIDIA Virtual GPU P40-8Q
- ❑ GPU Driver **X.Y.Z**

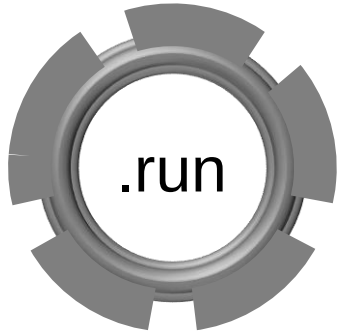


- ESXi Host
- ✓ NVIDIA GPU P40
- ❑ GPU VIB **X.Y.Z**



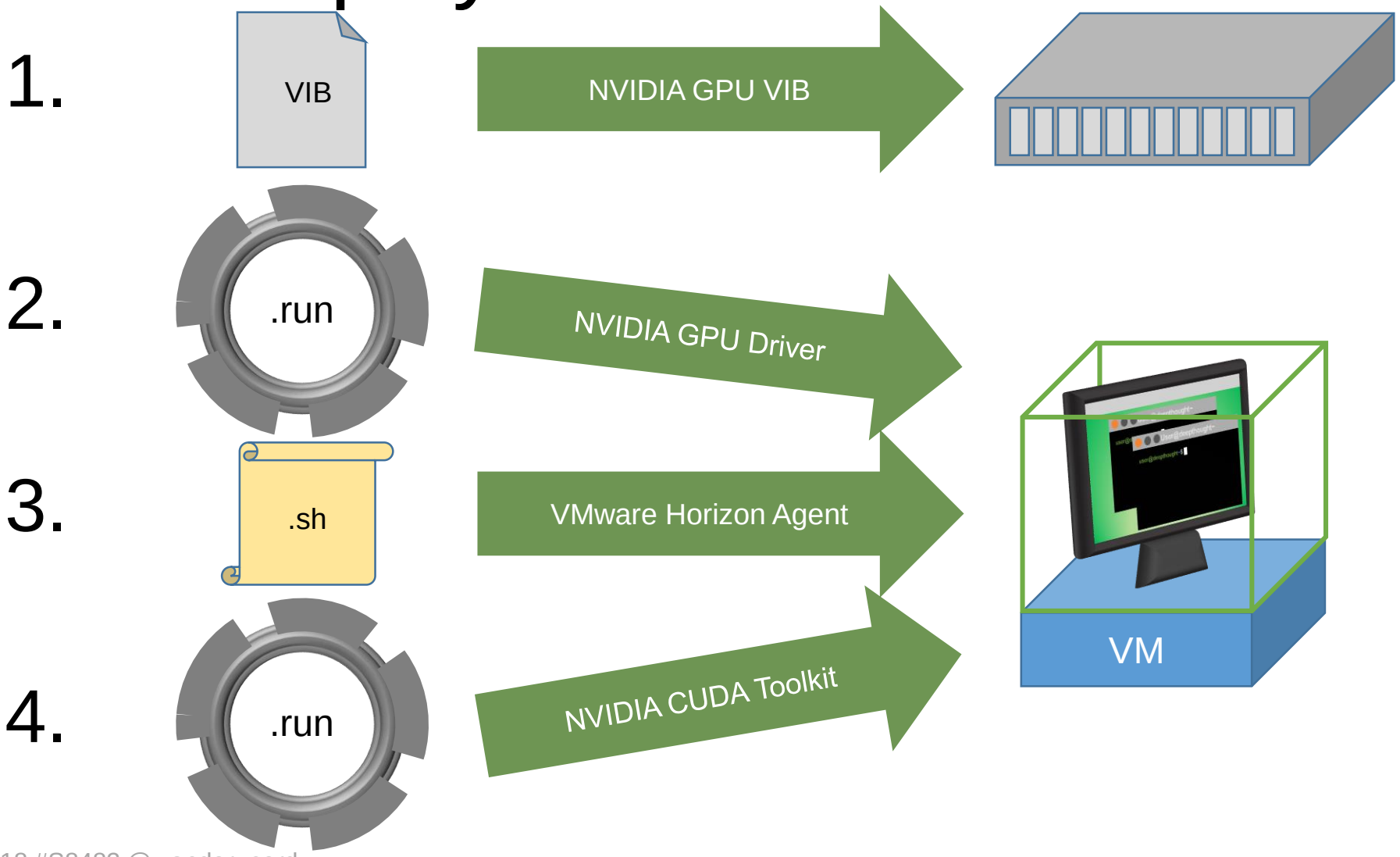


- NVIDIA CUDA Toolkit Deb/RPM
- ✓ CUDA Compatible GPU
- ○ GPU Driver **A.B.C**



- NVIDIA CUDA Toolkit (run)
- ✓ CUDA Compatible GPU
- ✓ GPU Driver **configurable**

CUDA Deployment Overview



Get the Right Installer

1drnrd.me/getCUDA More

CUDA Toolkit 9.1 Download

Home > ComputeWorks > CUDA Toolkit > CUDA Toolkit 9.1 Download

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System	Windows, Linux, Mac OSX
Architecture	x86_64, ppc64le
Distribution	Fedora, OpenSUSE, RHEL, CentOS, SLES, Ubuntu
Version	7, 6
Installer Type	runfile (local), rpm (local), rpm (network)

Select appropriate installer

Before installing the CUDA Toolkit on Linux, please ensure that you have the latest NVIDIA driver R390 installed. The latest NVIDIA R390 driver is available at: www.nvidia.com/drivers

Using .run to Deploy CUDA Toolkit

1drnrd.me/CUDAguide More

1. Disable Nouveau (varies per OS)
2. Switch runlevel 3 (text mode) – *when you do this the virtual console will be functional again until you exit the run level*
3. Execute the run file: **`sudo sh ./cuda_<version>_linux.run`**
    1. Follow the prompts on screen
    2. When asked to install the GPU driver enter **`No (N)`**, **this is the most important part of this process**.
    3. If you select yes, the file will overwrite the already installed driver with the driver included in the CUDA package
4. Finish answering the prompts and complete the installation of the run file
5. Apply any patches
6. Complete Post-Installation Actions
    1. Mandatory Actions
    2. Recommended Actions
    3. Optional Actions

CUDA Toolkit Install

```
    LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
    OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
    THE SOFTWARE.

-----------------
     Do you accept the previously read EULA?
accept/decline/qaccept

Install NVIDIA Accelerated Graphics Driver for Linux-x86_64 387.26?
(y)es/(n)o/(q)uit: n

Install the CUDA 9.1 Toolkit?
(y)es/(n)o/(q)uit: y

Enter Toolkit Location
 [ default is /usr/local/cuda-9.1 ]:

Do you want to install a symbolic link at /usr/local/cuda?
(y)es/(n)o/(q)uit: y

Install the CUDA 9.1 Samples?
(y)es/(n)o/(q)uit: y

Enter CUDA Samples Location
 [ default is /root ]: /home/tony

Installing the CUDA Toolkit in /usr/local/cuda-9.1 ...
```

CUDA Toolkit Install - Complete

```
===========
= Summary =
===========

Driver:   Not Selected
Toolkit:  Installed in /usr/local/cuda-9.1
Samples:  Installed in /home/tony, but missing recommended libraries

Please make sure that
 -   PATH includes /usr/local/cuda-9.1/bin
 -   LD_LIBRARY_PATH includes /usr/local/cuda-9.1/lib64, or, add /usr/local/cuda-9.1/lib64 to /etc/ld.so.conf and run ldconfig as root

To uninstall the CUDA Toolkit, run the uninstall script in /usr/local/cuda-9.1/bin

Please see CUDA_Installation_Guide_Linux.pdf in /usr/local/cuda-9.1/doc/pdf for detailed information on setting up CUDA.

***WARNING: Incomplete installation! This installation did not install the CUDA Driver. A driver of version at least 384.00 is required for C
UDA 9.1 functionality to work.
To install the driver using this installer, run the following command, replacing <CudaInstaller> with the name of this run file:
    sudo <CudaInstaller>.run -silent -driver

Logfile is /tmp/cuda_install_9941.log
[root@centos030918 Downloads]# cd ..
```

Post Installation Steps

1. Add /usr/local/cuda-<*version*>/bin to the PATH variable:
   `export PATH=/usr/local/cuda-``<version>`**`/bin${PATH:+:${PATH}}`**
   (Non persistent)

2. We then need to add the 64bit library to the the LD_LIBRARY_PATH variable:
   `export LD_LIBRARY_PATH=/usr/local/cuda-`
   `<version>`**`/lib64\${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}`**
   (Non persistent)

3. Install the writable samples
   `cuda-install-samples-``<version>`**`.sh <dir>`**
4. Make the samples:
   `cd ~/NVIDIA_CUDA-``<version>`**`_Samples`**
   `make`
   This can take a while to run, you may want to do this over lunch
5. Reboot your VM

Validating CUDA Functionality

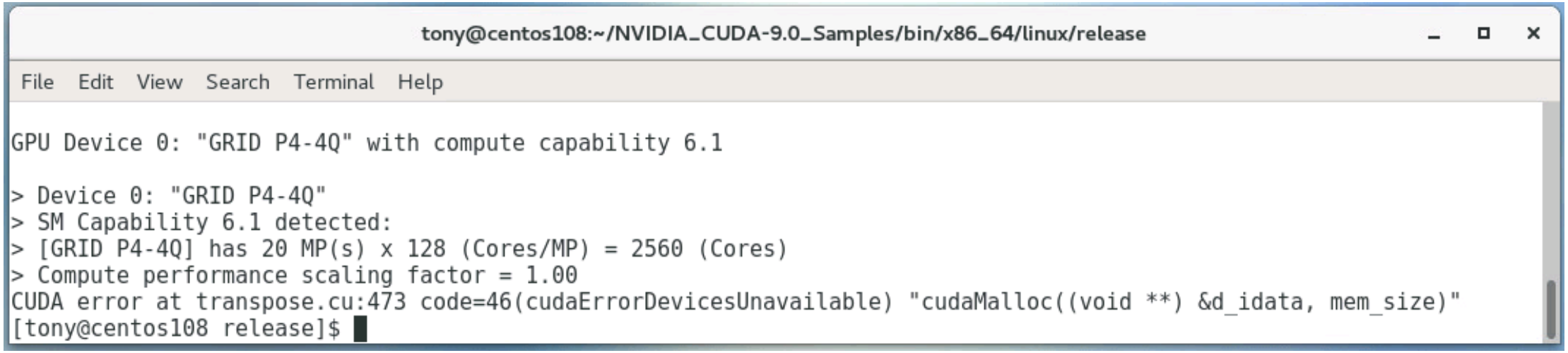
1drnrd.me/CUDAtest More

deviceQuery part of
NVIDIA CUDA Samples

tony@centos108:~/NVIDIA_CUDA-9.0_Samples/bin/x86_64/linux/release

```
[tony@centos108 release]$ ./deviceQuery
./deviceQuery Starting...

 CUDA Device Query (Runtime API) version (CUDART static linking)

Detected 1 CUDA Capable device(s)

Device 0: "GRID P4-4Q"
  CUDA Driver Version / Runtime Version          9.0 / 9.0
  CUDA Capability Major/Minor version number:    6.1
  Total amount of global memory:                 4096 MBytes (4294705152 bytes)
  (20) Multiprocessors, (128) CUDA Cores/MP:     2560 CUDA Cores
  GPU Max Clock rate:                            1114 MHz (1.11 GHz)
  Memory Clock rate:                             3003 Mhz
  Memory Bus Width:                              256-bit
  L2 Cache Size:                                 2097152 bytes
  Maximum Texture Dimension Size (x,y,z)         1D=(131072), 2D=(131072, 65536), 3D=(16384, 16384, 16384)
  Maximum Layered 1D Texture Size, (num) layers  1D=(32768), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers  2D=(32768, 32768), 2048 layers
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  2048
  Maximum number of threads per block:           1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size    (x,y,z): (2147483647, 65535, 65535)
  Maximum memory pitch:                          2147483647 bytes
  Texture alignment:                             512 bytes
  Concurrent copy and kernel execution:          Yes with 2 copy engine(s)
  Run time limit on kernels:                     Yes
  Integrated GPU sharing Host Memory:            No
  Support host page-locked memory mapping:       Yes
  Alignment requirement for Surfaces:            Yes
  Device has ECC support:                        Disabled
  Device supports Unified Addressing (UVA):      Yes
  Supports Cooperative Kernel Launch:            Yes
  Supports MultiDevice Co-op Kernel Launch:      Yes
  Device PCI Domain ID / Bus ID / location ID:   0 / 2 / 2
  Compute Mode:
     < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 9.0, CUDA Runtime Version = 9.0, NumDevs = 1
Result = PASS
[tony@centos108 release]$
```

Licensing or Insufficient vGPU Profile



… code=46(cudaErrorDevicesUnavailable) …

Testing

P4-4Q – MC_EstimatePiP

```
Monte Carlo Estimate Pi (with batch PRNG)
=========================================

Estimating Pi on GPU (GRID P4-4Q)

Precision:       single
Number of sims: 100000
Tolerance:       1.000000e-02
GPU result:      3.136320e+00
Expected:        3.141593e+00
Absolute error: 5.272627e-03
Relative error: 1.678329e-03

MonteCarloEstimatePiP, Performance = 565585.27 sims/s,
Time = 176.81(ms), NumDevsUsed = 1, Blocksize = 128
```

1drnrd.me/CUDA4Q More

Single VM
Equal Share Scheduling

Passthrough P4 – MC_EstimatePiP

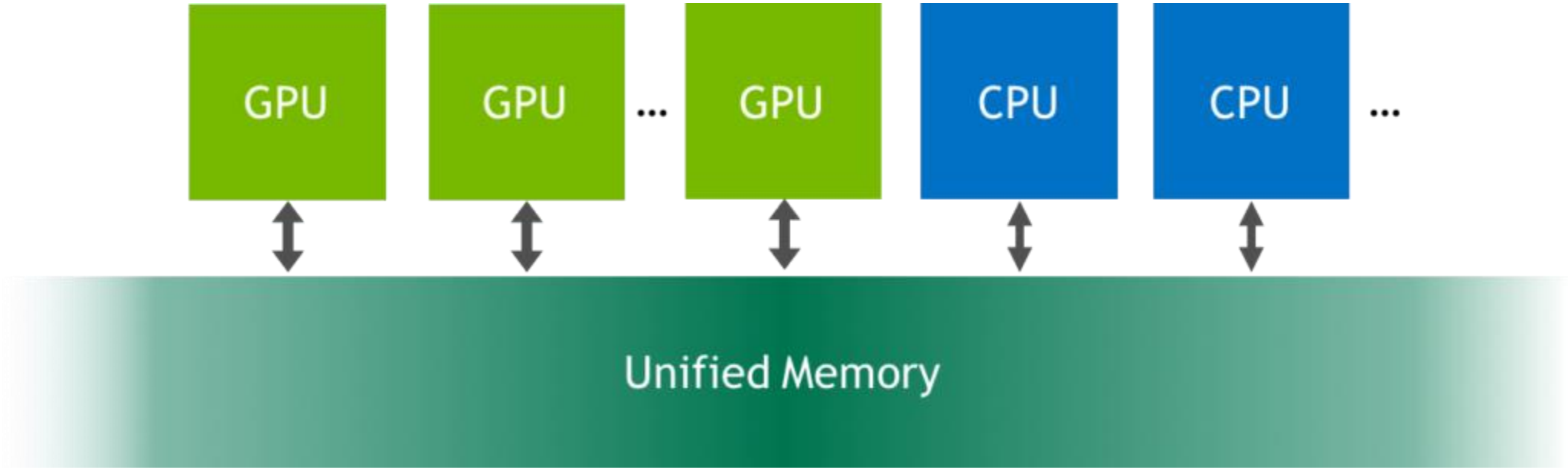
```
Monte Carlo Estimate Pi (with batch PRNG)
=========================================

Estimating Pi on GPU (Tesla P4)

Precision:      single
Number of sims: 100000
Tolerance:      1.000000e-02
GPU result:     3.136320e+00
Expected:       3.141593e+00
Absolute error: 5.272627e-03
Relative error: 1.678329e-03

MonteCarloEstimatePiP, Performance = 1100097.88 sims/s,
Time = 90.90(ms), NumDevsUsed = 1, Blocksize = 128
```

1drnrd.me/CUDApassP4 More

Single VM
Entire P4 GPU

Unified Memory

1drnrd.me/unimem More

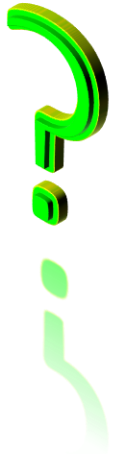


Appears to not work in Linux VMs, more testing required

```
17.  float *x, *y;
18.
19.  // Allocate Unified Memory -- accessible from CPU or GPU
20.  cudaMallocManaged(&x, N*sizeof(float));
21.  cudaMallocManaged(&y, N*sizeof(float));
```

```
tony@centos0322:~/Documents
File  Edit  View  Search  Terminal  Help
[tony@centos0322 Documents]$ ./unified
Segmentation fault (core dumped)
[tony@centos0322 Documents]$
```

Questions



Thank you for attending

Please complete the session survey in the mobile app

Catch me after the session or at the Dell booth (815)

Tony Foster
@wonder_nerd
Tony.Foster@wondernerd.net
https://wondernerd.net

Resources (1 of 5)

- /blog Wondernerd.net – Personal Blog https://www.wondernerd.net/blog/
- /lab Wondernerd.net - My Home Lab https://www.wondernerd.net/blog/about/my-home-lab/
- /ubuntu VMware Flings – Ubuntu OVA https://labs.vmware.com/flings/horizon-ova-for-ubuntu
- /S7349 GTC17 Session – Getting Started with Linux VMs http://on-demand-gtc.gputechconf.com/gtc-quicklink/hFm3d
- /VMTN6636U VMworld 2017 vBrownBag Tech Talk – GPU Enabled Linux VDI https://youtu.be/RuZK-X4LQiQ

Resources (2 of 5)

- /GPUQoS NVIDIA Forums – vGPU management and QoS schduler API https://devtalk.nvidia.com/default/topic/1023524/-vgpu-management-and-qos-schduler-api-pascal-preemption-api-/
- /scheduling NVIDIA Virtual GPU Documentation - Scheduling http://docs.nvidia.com/grid/latest/grid-vgpu-user-guide/index.html#changing-vgpu-scheduling-policy
- /vCUDAp1 Wondernerd.net - Empowering CUDA Developers with Virtual Desktops (Part1) https://www.wondernerd.net/blog/empowering-cuda-developers-with-virtual-desktops-part1/
- /getCUDA NVDIA CUDA Toolkit Page https://developer.nvidia.com/cuda-downloads

Resources (3 of 5)

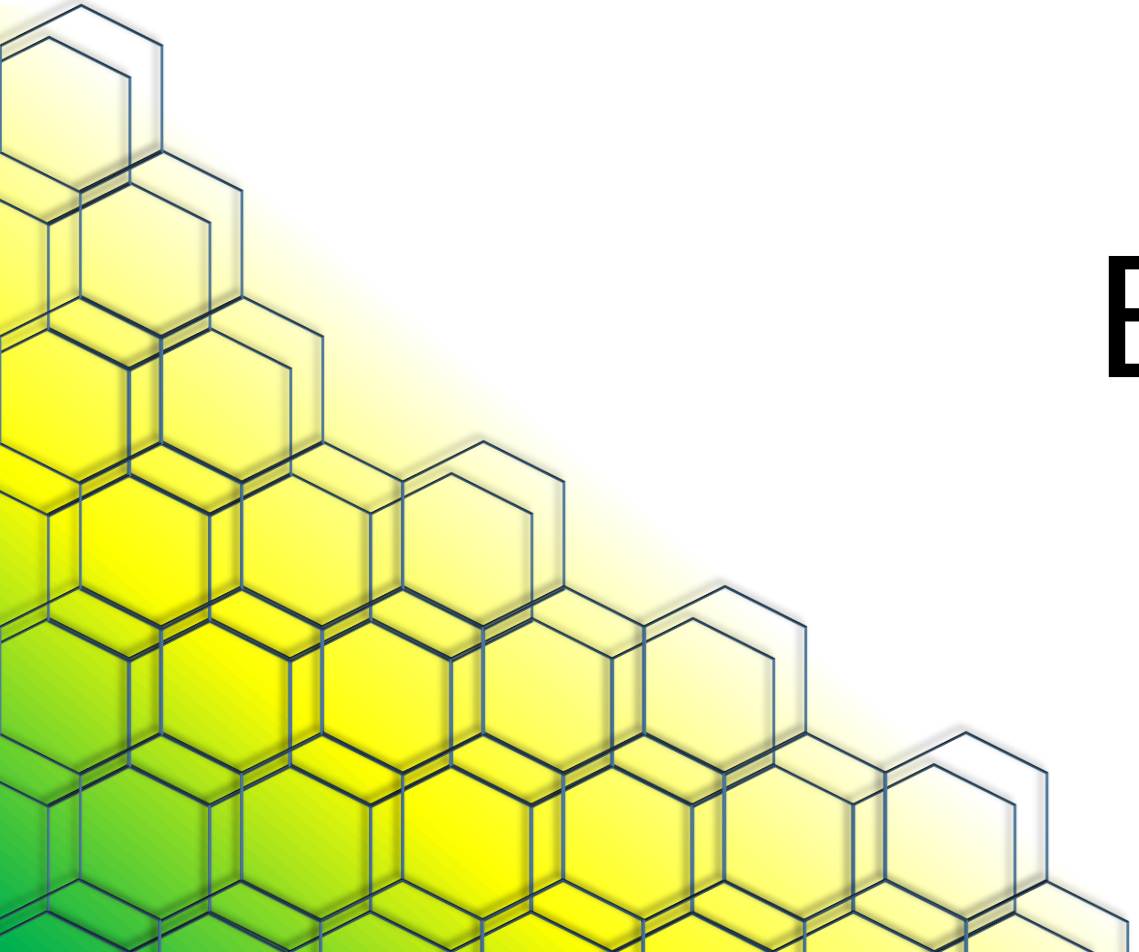
- /CUDAguide NVIDIA CUDA Toolkit Documentation - Linux https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html
- /CUDAtest NVIDIA CUDA Toolkit Documentation - Linux – Verify the Installation https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#running-binaries
- /CUDA4Q Wondernerd.net – CUDA Examples Run on P4-4Q https://www.wondernerd.net/blog/wp-content/uploads/2017/10/CUDAtoolkitResults.htm.html
- /CUDApassP4 Wondernerd.net – CUDA Examples Run on a Passthrough P4 https://www.wondernerd.net/blog/wp-content/uploads/2018/03/testresults_03-17-18_13-23_P4.htm

Resources (4 of 5)

- An Even Easier Introduction to CUDA
  https://devblogs.nvidia.com/even-easier-introduction-cuda/
- /unimem Unified Memory for CUDA Beginners
  https://devblogs.nvidia.com/unified-memory-cuda-beginners/
- CUDA Profiling Tools
  https://developer.nvidia.com/cuda-profiling-tools-interface
- CUDA LLVM Compiler
  https://developer.nvidia.com/cuda-llvm-compiler
- CUDA Toolkit Documentation
  http://docs.nvidia.com/cuda/index.html
- CUDA Enabled Products
  https://developer.nvidia.com/cuda-gpus

Resources (5 of 5)

- White Paper – NVIDIA Tesla P100 https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf
- Product Brief – NVIDIA Tesla P40 http://images.nvidia.com/content/tesla/pdf/Tesla-P40-Product-Brief.pdf
- Virtual GPU Software Documentation http://docs.nvidia.com/grid/latest/index.html
- Install Horizon Agent on a Linux Virtual Machine https://docs.vmware.com/en/VMware-Horizon-7/7.3/linux-desktops-setup/GUID-F06FF1A7-BDEF-4269-B2AB-C62819D4FCCD.html
- Using the Horizon Client for Linux (4.4) https://docs.vmware.com/en/VMware-Horizon-Client-for-Linux/4.4/horizon-client-linux-44-document.pdf



END