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
Agenda

Define the Technologies
Why do This?
Environment Overview
Deployment
Testing
Questions
Resources

#include <iostream>
#include <math.h>
// Kernel Function to add the elements
__global__
#define add(int n, float *x, float *y)
{
  int index = blockIdx.x * blockDim.x * gridDim.x;
  int stride = blockDim.x * gridDim.x;
  for (int i = index; i < n; i += 
    y[i] += x[i] + y[i];
}

int main(void)
{
  int N = 1<<20;
  float *x, *y;

  // Allocate Unified Memory - 
  cudaMallocManaged(&x, N*sizeof(float));
  cudaMallocManaged(&y, N*sizeof(float));

  // initialize x and y arrays
  for (int i = 0; i < N; i++)
    {
      x[i] = 1.0f;
      y[i] = 2.0f;
    }

  // Run kernel on 1M elements
  int blockSize = 256;
  int numBlocks = (N + blockSize) / blockSize;
  add<<<numBlocks, blockSize>>>(x, y);

  // Wait for GPU to finish before
  cudaDeviceSynchronize();

#define GTC18 S8483 @wonder_nerd
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 underlying physical GPU to the virtual machine, such as graphics processing or CUDA.
Why I Did This

• 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
- Security
- Resource Sharing
- Multiple Workspaces
- Version Control
- Backup / DR
- New Workspace
- Automated Delivery

#GTC18 #S8483 @wonder_nerd
Environment Overview
Requirements

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

Important
My Virtual Environment

- vCenter Server Appliance
- Connection Server
- Active Directory
- NVIDIA License Server

Virtual Desktops

VMware vCenter Server

Virtual Appliance

VM

"Lab"

Office
Scaling to the Organization

Remote Workers

VMware vSphere

Centralized Virtual Desktops

Virtualized Environment

Data Lakes
Hardware Specs

- 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

• 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?

VM

Virtual Display Console

VM

Virtual Display Console

Horizon

#GTC18 #S8483 @wonder_nerd
Preparing Hosts & VM

GTC17 Session S7349

S7349: Getting Started with GPUs for Linux Virtual Desktops on VMware Horizon
Trey Johnson — Sr. Architect, Lincare, Inc.
Tony Foster — Sr. Advisor, Technical Marketing, Dell Technologies
NVIDIA GRID Community Advisor

1drnrd.me/S7349

VMworld Session VMTN6636U

VMTN6636U: GPU Enabled Linux VDI
Tony Foster — Sr. Advisor, Technical Marketing, Dell Technologies
VMware vExpert, VMware EUC Champion
NVIDIA GRID Community Advisor
@wonder_nerd www.wondernerd.net

1drnrd.me/VMTN6636U

#GTC18 #S8483 @wonder_nerd
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

<table>
<thead>
<tr>
<th>Profile</th>
<th>Frame Buffer (Mbytes)</th>
<th>Maximum vGPUs per Board</th>
<th>License Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>P40-24Q</td>
<td>24576</td>
<td>1</td>
<td>Quadro vDWS</td>
</tr>
<tr>
<td>P40-12Q</td>
<td>12288</td>
<td>2</td>
<td>Quadro vDWS</td>
</tr>
<tr>
<td>P40-8Q</td>
<td>8192</td>
<td>3</td>
<td>Quadro vDWS</td>
</tr>
<tr>
<td>P40-6Q</td>
<td>6144</td>
<td>4</td>
<td>Quadro vDWS</td>
</tr>
<tr>
<td>P40-4Q</td>
<td>4096</td>
<td>6</td>
<td>Quadro vDWS</td>
</tr>
<tr>
<td>P40-3Q</td>
<td>3072</td>
<td>8</td>
<td>Quadro vDWS</td>
</tr>
<tr>
<td>P40-2Q</td>
<td>2048</td>
<td>12</td>
<td>Quadro vDWS</td>
</tr>
<tr>
<td>P40-1Q</td>
<td>1024</td>
<td>24</td>
<td>Quadro vDWS</td>
</tr>
</tbody>
</table>

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

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.

- **Best Effort**
  - VM1, 17%
  - VM2, 33%
  - VM3, 50%

- **Equal Share**
  - VM1, 33%
  - VM2, 33%
  - VM3, 33%

- **Fixed Share**
  - VM1, 25%
  - VM2, 25%
  - VM3, 25%
  - No VM, 25%

#GTC18 #S8483 @wonder_nerd
Configuring Scheduling
RmPVMRL Registry Key

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Best Effort Scheduler</td>
<td></td>
</tr>
<tr>
<td>0x01</td>
<td>Equal Share Scheduler (Default)</td>
<td>Enterprise</td>
</tr>
<tr>
<td>0x11</td>
<td>Fixed Share Scheduler</td>
<td>Service Provider</td>
</tr>
</tbody>
</table>

#GTC18 #S8483 @wonder_nerd
vGPU Driver Requirements

• **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 Virtual GPU P40**
  - GPU Driver X.Y.Z

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

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

1. NVIDIA GPU VIB

2. NVIDIA GPU Driver

3. VMware Horizon Agent

4. NVIDIA CUDA Toolkit

#GTC18 #S8483 @wonder_nerd
# Get the Right Installer

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.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Windows</th>
<th>Linux</th>
<th>Mac OSX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>x86_64</td>
<td>ppc64le</td>
<td></td>
</tr>
<tr>
<td>Distribution</td>
<td>Fedora</td>
<td>OpenSUSE</td>
<td>RHEL</td>
</tr>
<tr>
<td>Version</td>
<td>7</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Installer Type</td>
<td>runfile [local]</td>
<td>rpm [local]</td>
<td>rpm [network]</td>
</tr>
</tbody>
</table>

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

#GTC18 #S8483 @wonder_nerd
Using .run to Deploy CUDA Toolkit

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)|c|/(q)|uit: n

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

Install the CUDA 9.1 Samples?
|y|es/(n)|c|/(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 CUDA 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

Log files is /tmp/cuda_install_0041.log
[tony@centos050910 Downloads]# cd ..
Post Installation Steps

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

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

3. Install the writable samples
   ```bash
   cuda-install-samples-<version>.sh <dir>
   ```

4. Make the samples:
   ```bash
   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

deviceQuery part of NVIDIA CUDA Samples
Licensing or Insufficient vGPU Profile

GPU Device 0: "GRID P4-4Q" with compute capability 6.1

Device 0: "GRID P4-4Q"
> SM Capability 6.1 detected:
> [GRID P4-4Q] has 20 MP(s) x 128 (Cores/MP) = 2560 (Cores)
> Compute performance scaling factor = 1.00
CUDA error at transpose.cu:473 code=46(cudaErrorDevicesUnavailable) "cudaMalloc([void **] &d_data, mem_size)"
[tony@centos108 release]$  

... code=46(cudaErrorDevicesUnavailable) ...
Testing
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
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

1dmrd.me/CUDApasstP4
More

Single VM
Entire P4 GPU
Unified Memory

Appears to not work in Linux VMs, more testing required

```c
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));
```
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 scheduler API

- **/scheduling** NVIDIA Virtual GPU Documentation - Scheduling

- **/vCUDAp1** Wondernerd.net - Empowering CUDA Developers with Virtual Desktops (Part1)
  [https://www.wondernerd.net/blog/empowering-cuda-developers-with-virtual-desktops-part1/](https://www.wondernerd.net/blog/empowering-cuda-developers-with-virtual-desktops-part1/)

- **/getCUDA** NVIDIA CUDA Toolkit Page
Resources (3 of 5)

- **/CUDAGuide** NVIDIA CUDA Toolkit Documentation - Linux

- **/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

- **/CUDApassP4** Wondernerd.net – CUDA Examples Run on a Passthrough P4
Resources (4 of 5)

- An Even Easier Introduction to CUDA
  https://devblogs.nvidia.com/even-easier-introduction-cuda/
- /unimem Unified Memory for CUDA Beginners
- CUDA Profiling Tools
- CUDA LLVM Compiler
- CUDA Toolkit Documentation
  http://docs.nvidia.com/cuda/index.html
- CUDA Enabled Products
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

• Virtual GPU Software Documentation
  http://docs.nvidia.com/grid/latest/index.html

• Install Horizon Agent on a Linux Virtual Machine

• Using the Horizon Client for Linux (4.4)
END