S9670 VIRTUAL DESKTOPS BY DAY, COMPUTATIONAL WORKLOADS BY NIGHT - AN EXAMPLE INFRASTRUCTURE
AGENDA

• What We Will Discuss
• Benefits of VDI
• Computation Defined and Context
• Dual-Use and Workflow Scenarios
• Operational Challenges
• Solution Options
• Reference Architecture
• Demonstration
• Summary
WHAT WE WILL DISCUSS

A practical approach to configure intervals of VDI and Computational Resources on a daily basis - in an environment primarily designed for VDI - using commonly available tools.

More about perspective than technology
BENEFITS OF VIRTUAL DESKTOP INFRASTRUCTURE

• Enable flexible workflow scenarios
• Utilize centralized, shared, and protected storage
• Enable intellectual property protection
• Provide flexibility in configuration
• Enable user/workforce mobility
• Widely supported GPU acceleration

What you planned the system to do.
COMPUTATIONAL SPECTRUM

Additive Scale of Requirements

General Compute
- Double Precision Math
- Multi-GPU Support
- Latency Pressure
- Storage Pressure
- Short to medium runtimes
- ECC Memory
- Higher CPU Utilization
- Linux Support

‘Lite’ Compute
- CUDA
- OpenCL
- Single Precision Math
- Latency tolerant
- Very short runtimes
- Windows Support

Classic High End Compute
- High Performance Interconnects
- High Performance Storage
- Multi-node Support
- Job Scheduling
- Bandwidth Sensitivity
- Long runtimes
- Memory Page Retirement
WHY DUAL USE?

• Cost and/or space savings
• Variable usage trends/rates
• Desire for on-prem elasticity
• Unpredictable user community
• Provide more workflow options to more users
• Effective cost justification (capital/operational)

Make best use of available resources
SCENARIO CONSIDERATIONS FOR DUAL USE

- Creative Studio - Artists go home during late hours
- Architecture Firm - Engineers/Designers work daylight hours
- University/College - Lower utilization during summer sessions
- Financial Services Firm - Lower utilization when markets are closed
- Gov’t Agency - Multiple programs, duplicate (idle) resources

*Primary goal is user experience*
WORKFLOW CONSIDERATIONS FOR DUAL USE

- Creative Studio - Create during day / Render by Night
- Architecture Firm - Design during day / Render-Compute by Night
- University/College - Sell cycles or run experiments during Summer
- Financial Services Firm - Traders by day / Numerical analysis by night
- Gov’t Agency - Analysis work by Day / Image processing at Night

Get creative with workflow overlap
OPERATIONAL CHALLENGES

• What to do with our user VMs?
• How do we best provision user VMs?
• How do we monitor utilization?
• How do we orchestrate user VM state, migration, and timing?
• How do we manage compute jobs, and be ready for user VM restart?
• How will users be productive in a scheduled environment?

*Manage Users, balanced with Compute Productivity*
VECTORS FOR SUCCESS

• User policies - reboot per day or week
• Single precision math jobs
• Single GPU compute jobs
• Jobs that may be coalesced
• Excess capacity
• Stakeholder buy-in
• Skilled admin staff
COMMON VDI INFRASTRUCTURE ASSETS

- Hypervisor(s) - vSphere, AHV, RHVH, XenServer
- vGPU Software
- Compute cluster of nodes (chassis)
- CPUs, GPUs, Storage, Network Assets
- Monitoring Tools
- Orchestration / Layering Tools
- Containers
- Job Schedulers

Many common building blocks available
SOLUTION VECTORS

• Shut down (all users) and swap (in all the compute)
• Shut down (some users) and swap in (some) compute
• Migrate/degrade (users) to fewer hosts, swap (in some/all) compute
• Shut down (all users) and reprovision (to bare metal) nodes
• Keep all users intact; initiate a cycle harvester
• Some mixture of the above
• Other options...

GOAL = Use common and available tools
OPTION 1: SHUT DOWN / SWAP IN

- Shut Down User Pool
- Spin up compute Pool
- Run Scheduled Jobs
- Spin down compute Pool
- Restart User Pool

(Partial Shutdown also applies)
SLURM WORKLOAD MANAGER

"Slurm is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters."

Source: https://slurm.schedmd.com/overview.html

Components:

- Centralized Manager: `slurmctld` - monitors resources and work
- Compute Node daemon: `slurmd` - waits for and executes work, returns work status

In this example:

- Slurm-ctrl = cluster controller VM
- Compute[01-07] = compute VMs (nodes)
ANATOMY OF A COMPUTE VM

- Ubuntu 16.04/18.04
- Docker, nv-docker, Anaconda, Python3-pip, ipython-notebook
- vGPU 7.1
- CUDA 10, toolkit, and samples
- SLURM
- VMware VIEW agent
- DHCP per Active Directory DNS
- Packaged as a VM template
NVIDIA CONFIDENTIAL. DO NOT DISTRIBUTE.

COMPUTE PARTITION ORGANIZATION

Ubuntu - Compute Pool Partitions

vSphere
Chassis
GPU Type A
CPU Type A
Template A (Master Image)

vSphere
Chassis
GPU Type B
CPU Type B
Template B

vSphere
Chassis
GPU Type C
CPU Type C
Template C

Template Resource Partitions (SLURM)
SLURM COMPUTE PARTITION CONFIG

/etc/slurm/slurm.conf

<table>
<thead>
<tr>
<th>PARTITION</th>
<th>AVAIL</th>
<th>TIMELIMIT</th>
<th>NODES</th>
<th>STATE</th>
<th>NODELIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>T4x16Q</td>
<td>up</td>
<td>2-00:00:00:00</td>
<td>4</td>
<td>idle</td>
<td>Compute[01-04]</td>
</tr>
<tr>
<td>V100x32Q</td>
<td>up</td>
<td>2-00:00:00:00</td>
<td>1</td>
<td>idle</td>
<td>Compute05</td>
</tr>
<tr>
<td>RTXx24Q*</td>
<td>up</td>
<td>2-00:00:00:00</td>
<td>2</td>
<td>idle</td>
<td>Compute[06-07]</td>
</tr>
</tbody>
</table>

sinfo output

Linux VM Templates mapped to Compute Partitions
OPERATIONAL TIMELINE

**VDI State**
- 6 VMs (Linked-clones)
- Windows 10
- Non-persistent VMs
- T4-8Q vDWS Profiles

**Compute State**
- 4 x T4-16Q
- 1 x V100-32Q
- 2 x RTXx24Q

**VDI State**
- 6 VMs (Linked-clones)
- Windows 10
- Non-persistent VMs
- T4-8Q vDWS Profiles

---

**t1**
6 am
VDI State

**t2**
Midnight
Evacuate VDI / Start Compute

**t3**
6 am
Start VDI / Evacuate Compute
VCENTER INTERVAL SCHEDULING

VDI Interval:

Compute Interval:
## SHUT DOWN / SWAP IN - HARDWARE

<table>
<thead>
<tr>
<th>Component</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU</td>
<td>Tesla T4, V100, P40, RTX</td>
</tr>
<tr>
<td>Chassis</td>
<td>Supermicro 4029GP, Dell R740, HPDL380 Gen9</td>
</tr>
<tr>
<td>Storage</td>
<td>FA-M20R2 (Pure Storage)</td>
</tr>
<tr>
<td>Network</td>
<td>CISCO 10G</td>
</tr>
<tr>
<td>Endpoints</td>
<td>Various</td>
</tr>
</tbody>
</table>
# SHUT DOWN / SWAP IN - SOFTWARE

<table>
<thead>
<tr>
<th>Component</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypervisor</td>
<td>vSphere 6.7u1</td>
</tr>
<tr>
<td>Hypervisor Manager</td>
<td>vCenter 6.7</td>
</tr>
<tr>
<td>Job Scheduler</td>
<td>Slurm 17.11.12</td>
</tr>
<tr>
<td>Interval Scheduler</td>
<td>vCenter 6.7</td>
</tr>
<tr>
<td>VDI Guest o/s</td>
<td>Windows 10</td>
</tr>
<tr>
<td>Compute Guest o/s</td>
<td>Ubuntu 16.04</td>
</tr>
<tr>
<td>NVIDIA vGPU Software</td>
<td>vGPU 7.1</td>
</tr>
</tbody>
</table>
ENVIRONMENT MONITORING
FUTURE NEEDS AND ASKS

• Multiple GPUs per VM - limited availability today
• Dynamic vGPU assignment per Template provisioning
• Dynamic vGPU on live migration
• vGPU + GPU ECC + UVM + P2P - supports relevant compute
• vGPU + GPU memory Page retirement
• VM snapshots and user sessions
• Storage optimizations
• Live migration integration - exists today
IMPORTANT: VGPU VM DEPLOYMENT POLICY (VMWARE / CITRIX)

VMware vSphere Hypervisor (ESXi) by default uses a breadth-first allocation scheme for vGPU-enabled VMs; allocating new vGPU-enabled VMs on an available, least loaded physical GPU. **We need to change that.**

For Citrix, it's easy
FINDINGS

- At least 1 vCenter VM powered on in a pool (20/80 best practice)
- Unify the storage for users and data - both VDI and Linux
- Alert users when jobs don’t start properly - SLURM
- Care for permissions - SLURM, containers, renderers, storage
- SLURM is very powerful and potentially complex - understand it
- Manage user VDI logistics and operations
- Keep the UX paramount
S9670 VIRTUAL DESKTOPS BY DAY, COMPUTATIONAL WORKLOADS BY NIGHT - AN EXAMPLE INFRASTRUCTURE

Shailesh Deshmukh
Senior Solution Architect

Konstantin Cvetanov
Senior Solution Architect

Eric Kana
Senior Solution Architect

GPU Technology Conference 2019