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

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

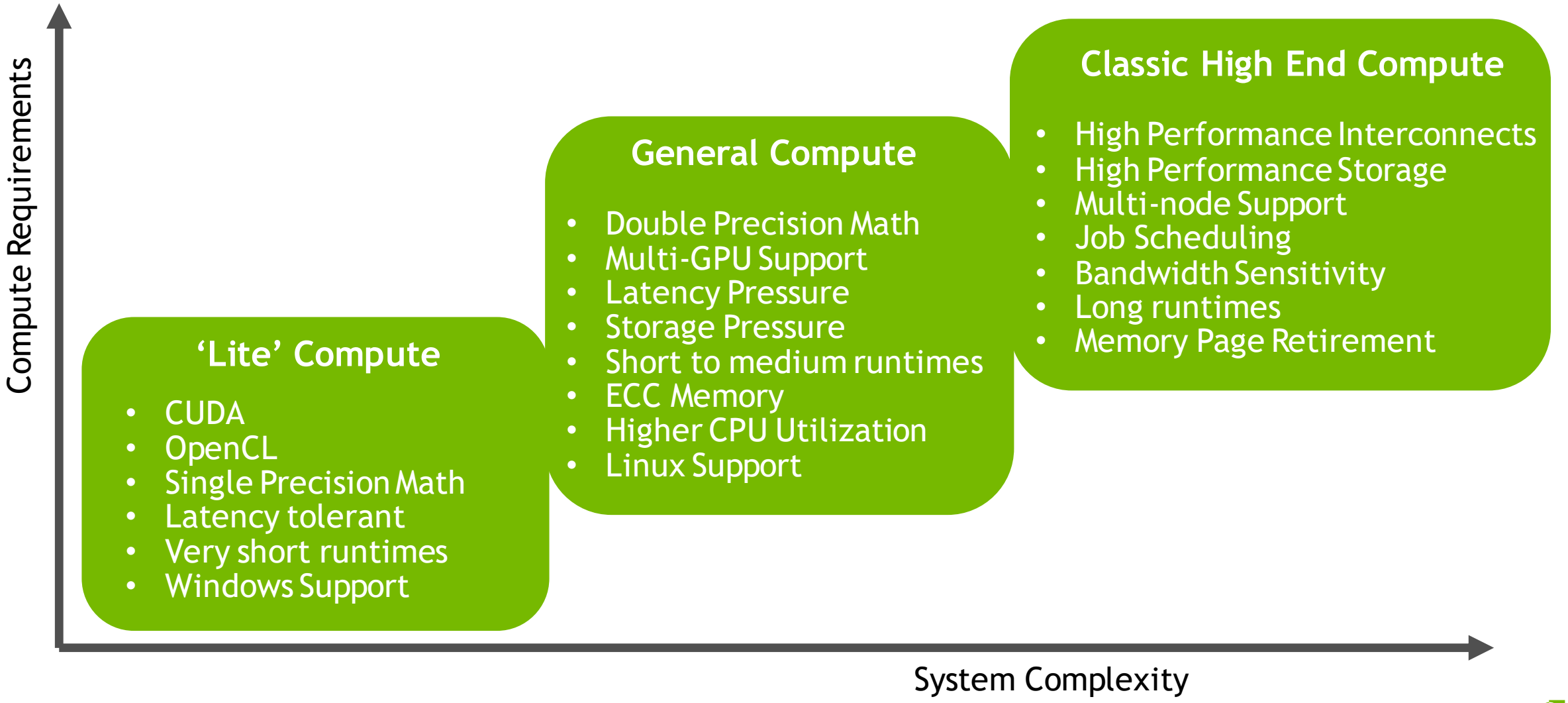
BENEFITS OF VIRTUAL DESKTOP INFRASTRUCTURE

- 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



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

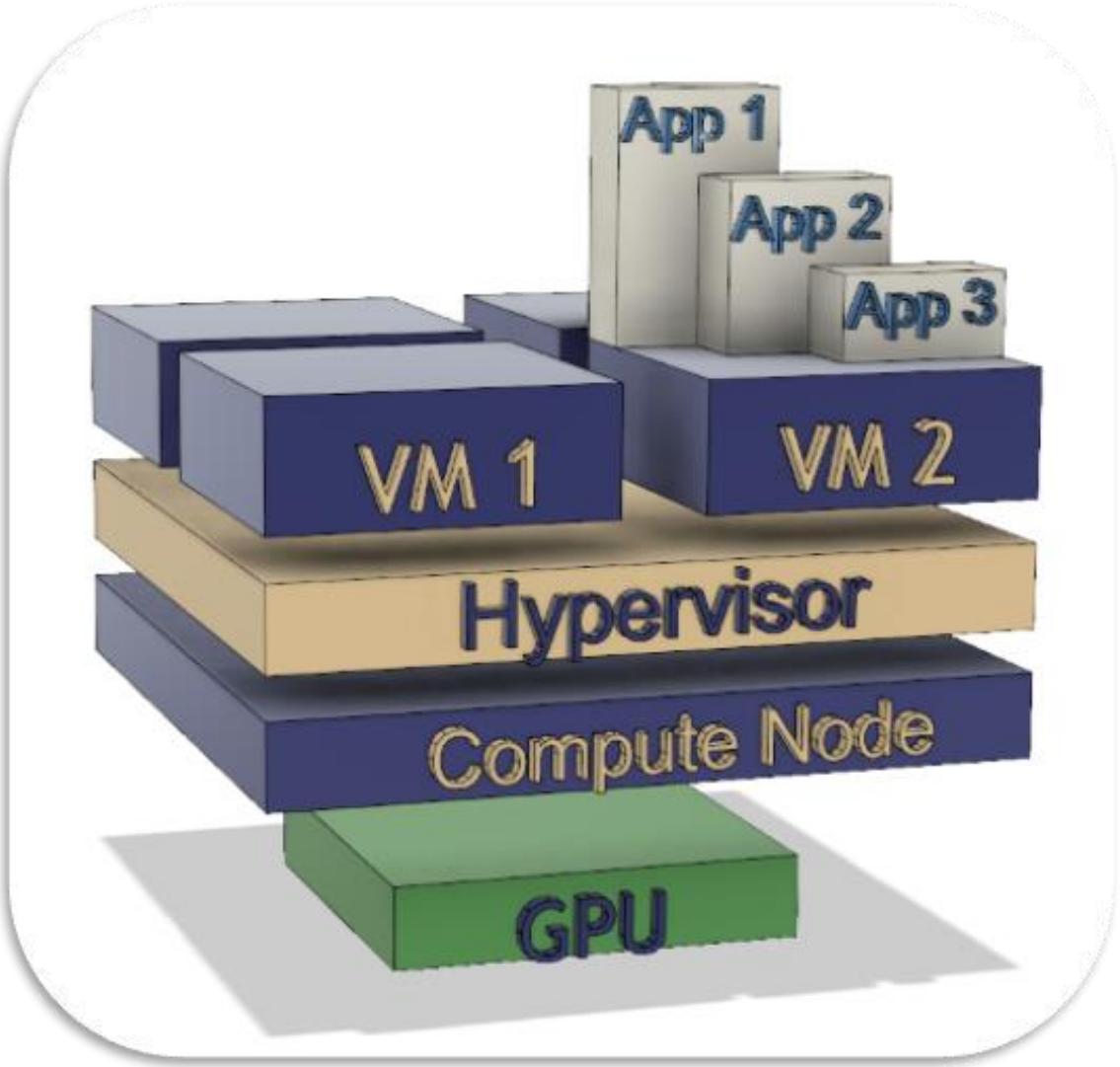
Many common building blocks available

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

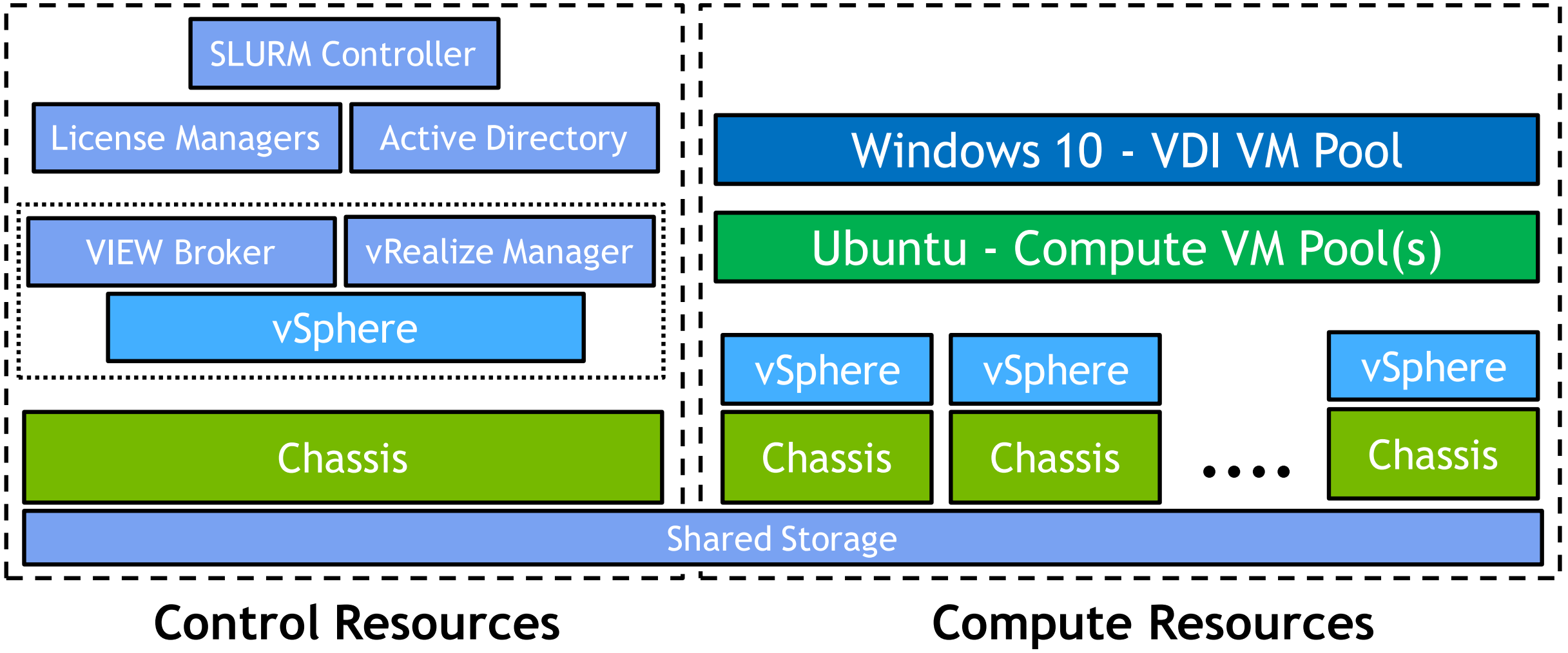
OPTION 1: SHUT DOWN / SWAP IN



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

(Partial Shutdown also applies)

ARCHITECTURE DIAGRAM



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

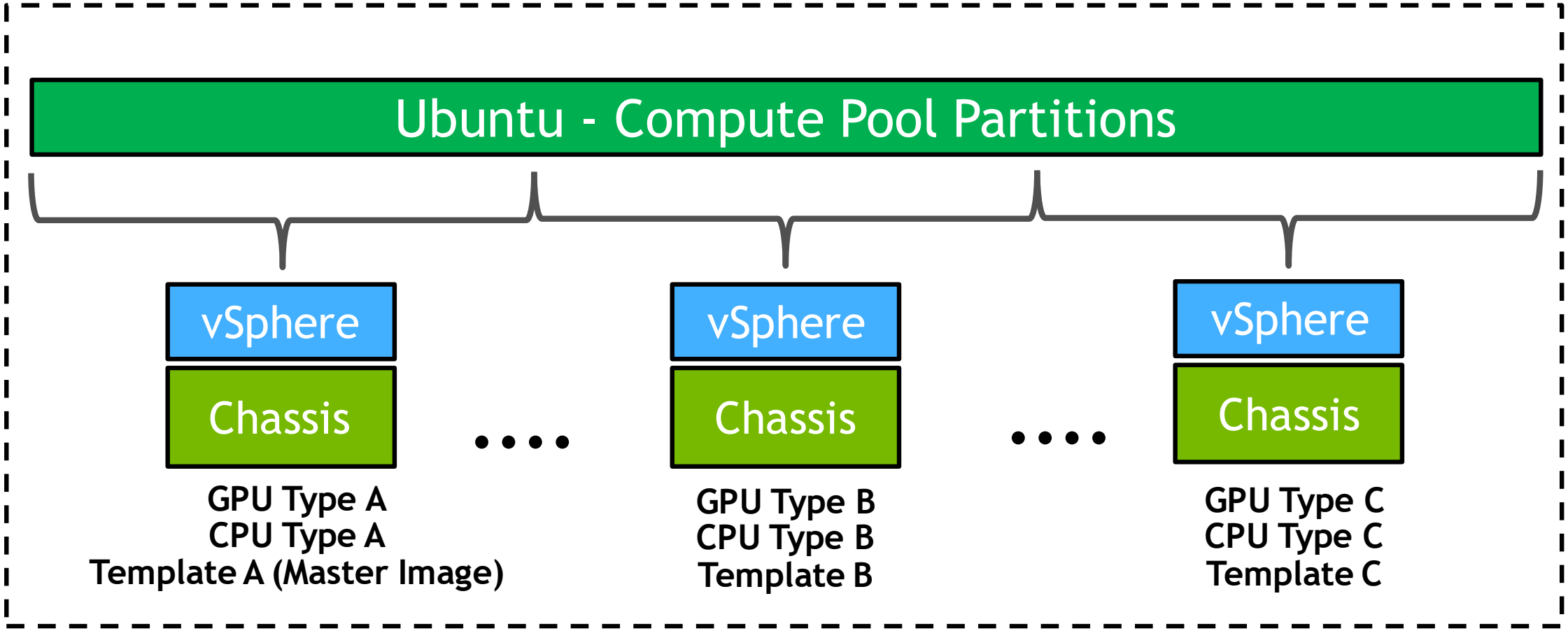
In this example:

- 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

COMPUTE PARTITION ORGANIZATION



Template Resource Partitions (SLURM)

SLURM COMPUTE PARTITION CONFIG

```
# COMPUTE NODES Partitions
NodeName=Compute[01-07] CPUs=2 Sockets=2 CoresPerSocket=8 RealMemory=14336 State=U
PartitionName=T4x16Q Nodes=Compute[01-04] Default=YES MaxTime=48:00:00 State=UP
PartitionName=V100x32Q Nodes=Compute[05] Default=YES MaxTime=48:00:00 State=UP
PartitionName=RTXx24Q Nodes=Compute[06-07] Default=YES MaxTime=48:00:00 State=UP
```

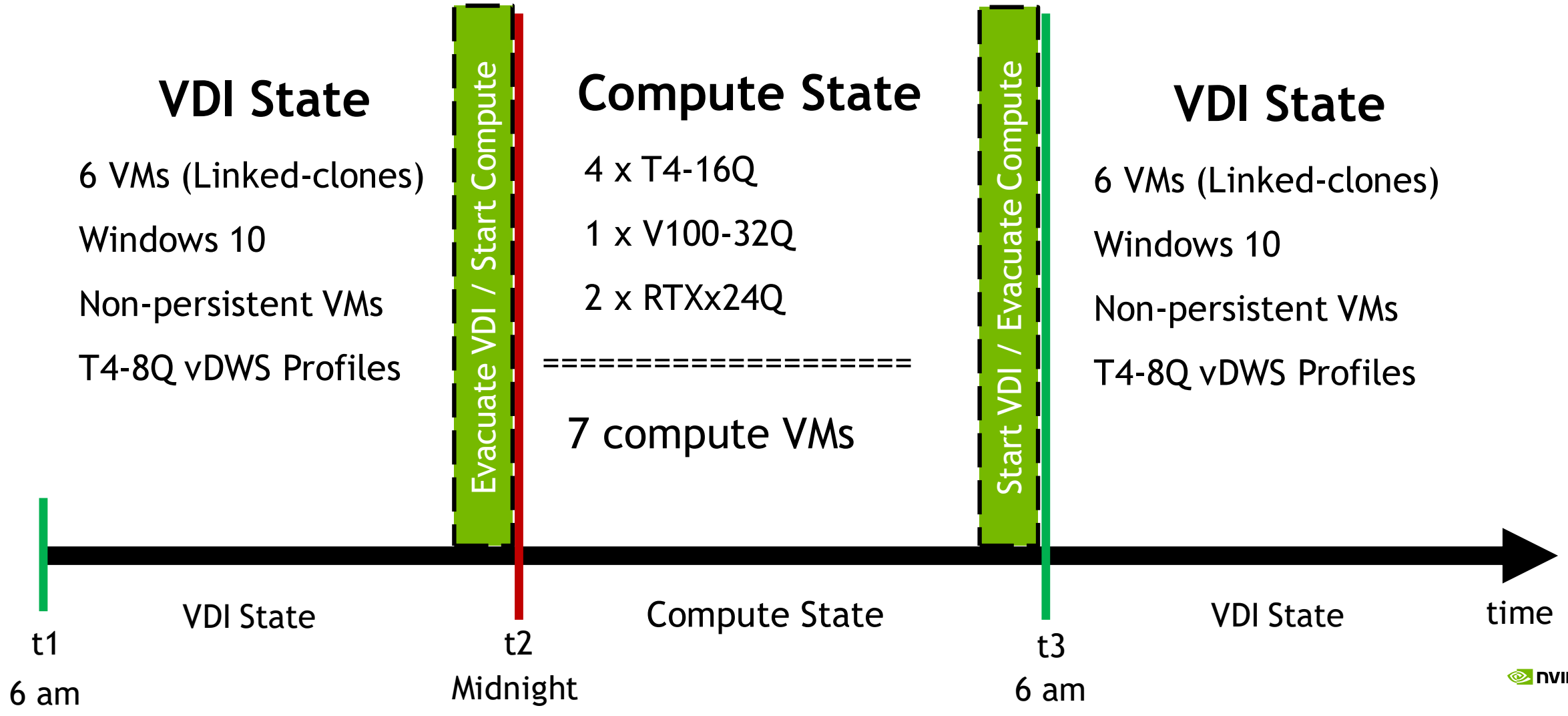
/etc/slurm/slurm.conf

```
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
T4x16Q       up 2-00:00:00      4   idle Compute[01-04]
V100x32Q     up 2-00:00:00      1   idle Compute05
RTXx24Q*     up 2-00:00:00      2   idle Compute[06-07]
```

sinfo output

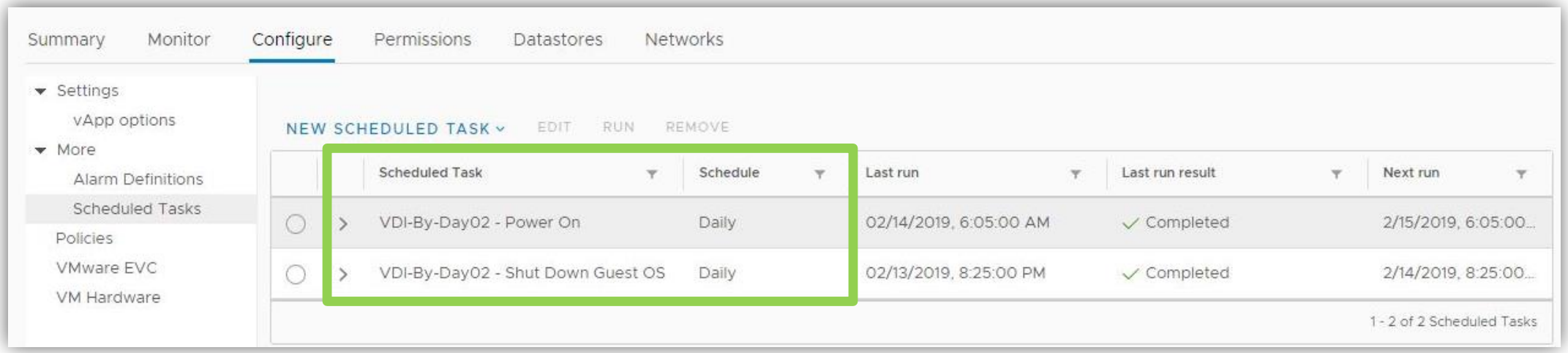
Linux VM Templates mapped to Compute Partitions

OPERATIONAL TIMELINE

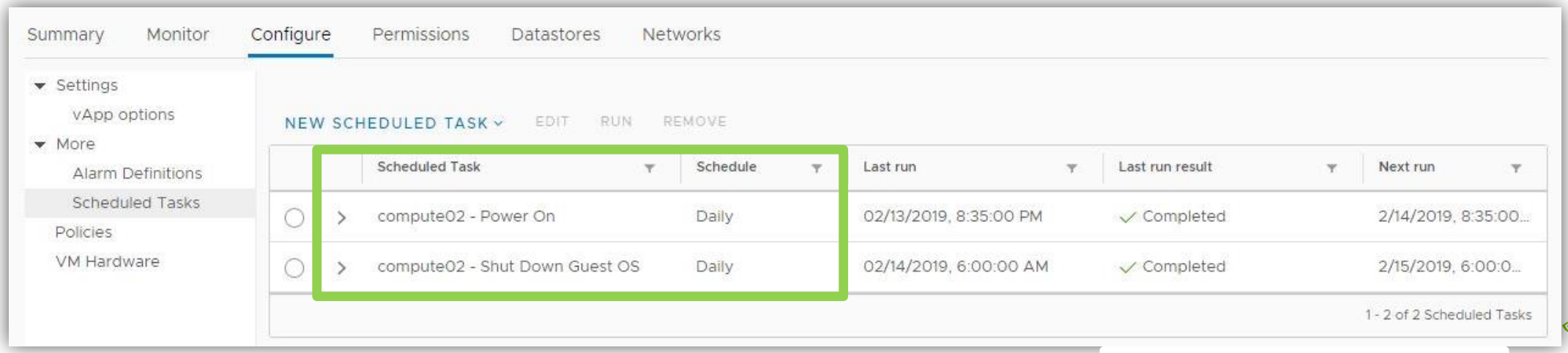


VCENTER INTERVAL SCHEDULING

VDI Interval:



Compute Interval:



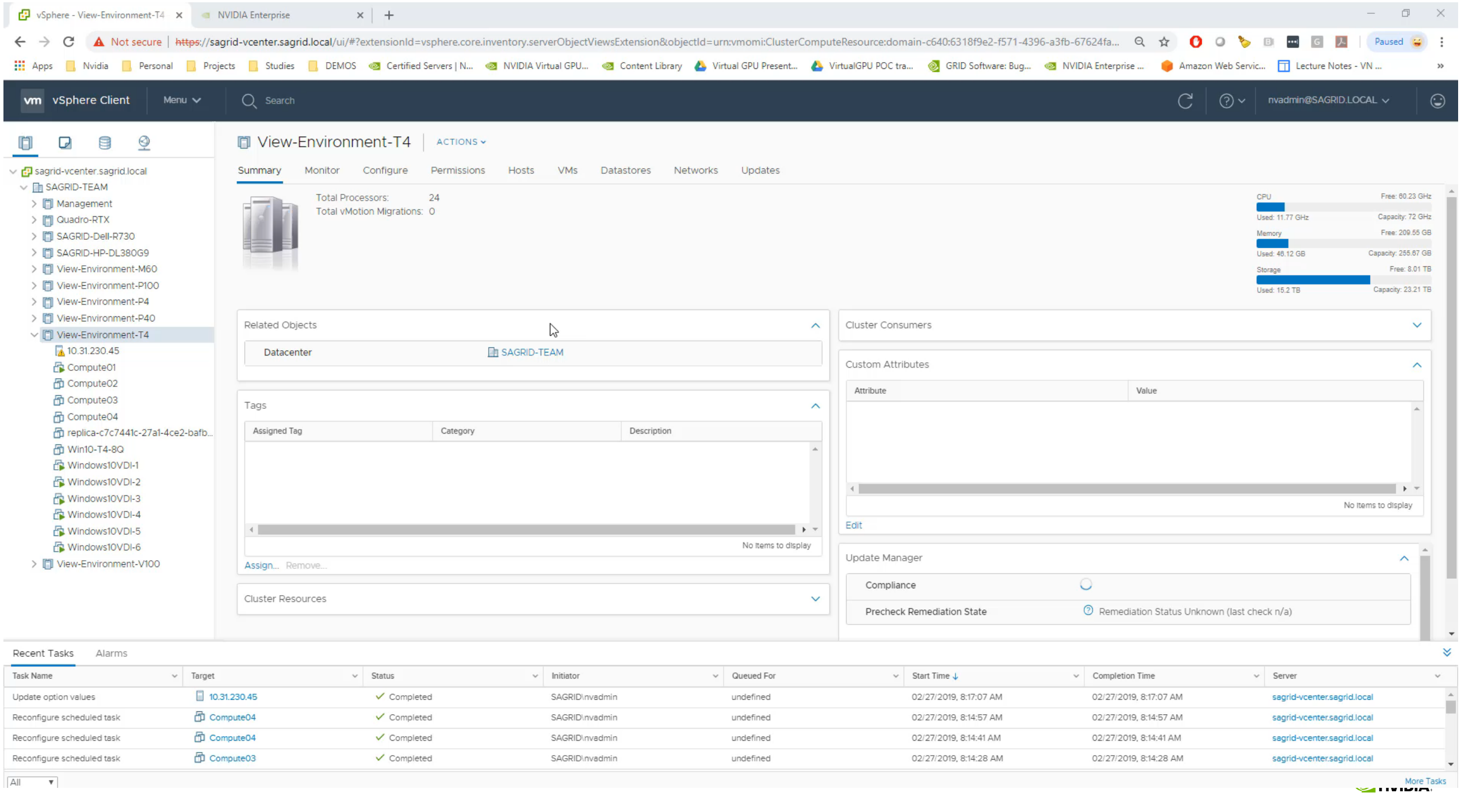
NVIDIA

SHUT DOWN / SWAP IN - HARDWARE

Component	Name
GPU	Tesla T4, V100, P40, RTX
Chassis	Supermicro 4029GP, Dell R740, HPDL380 Gen9
Storage	FA-M20R2 (Pure Storage)
Network	CISCO 10G
Endpoints	Various

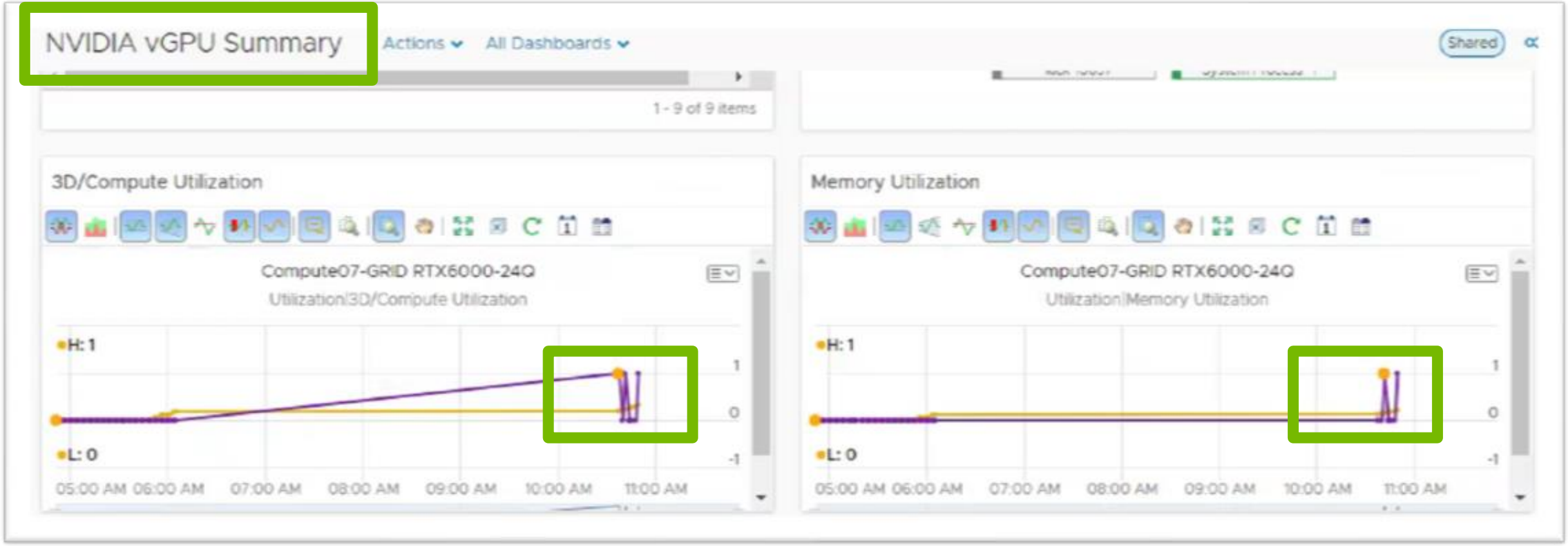
SHUT DOWN / SWAP IN - SOFTWARE

Component	Name
Hypervisor	vSphere 6.7u1
Hypervisor Manager	vCenter 6.7
Job Scheduler	Slurm 17.11.12
Interval Scheduler	vCenter 6.7
VDI Guest o/s	Windows 10
Compute Guest o/s	Ubuntu 16.04
NVIDIA vGPU Software	vGPU 7.1


vSphere - View-Environment-T4
NVIDIA Enterprise
Not secure
https://sagrid-vcenter.sagrid.local/ui/#?extensionId=vsphere.core.inventory.serverObjectViewsExtension&objectId=urn:vmomi:ClusterComputeResource:domain-c640:6318f9e2-f571-4396-a3fb-67624fa...
Paused
Apps
Nvidia
Personal
Projects
Studies
DEMOS
Certified Servers | N...
NVIDIA Virtual GPU...
Content Library
Virtual GPU Present...
VirtualGPU POC tra...
GRID Software: Bug...
NVIDIA Enterprise ...
Amazon Web Servic...
Lecture Notes - VN ...
vSphere Client
Menu
Search
nvadmin@SAGRID.LOCAL
sagrid-vcenter.sagrid.local
SAGRID-TEAM
Management
Quadro-RTX
SAGRID-Dell-R730
SAGRID-HP-DL380G9
View-Environment-M60
View-Environment-P100
View-Environment-P4
View-Environment-P40
View-Environment-T4
10.31.230.45
Compute01
Compute02
Compute03
Compute04
replica-c7c7441c-27a1-4ce2-bafb...
Win10-T4-8Q
Windows10VDI-1
Windows10VDI-2
Windows10VDI-3
Windows10VDI-4
Windows10VDI-5
Windows10VDI-6
View-Environment-V100
View-Environment-T4
ACTIONS
Summary
Monitor
Configure
Permissions
Hosts
VMs
Datastores
Networks
Updates
Total Processors: 24
Total vMotion Migrations: 0
CPU
Free: 60.23 GHz
Used: 11.77 GHz
Capacity: 72 GHz
Memory
Free: 209.55 GB
Used: 46.12 GB
Capacity: 255.67 GB
Storage
Free: 8.01 TB
Used: 15.2 TB
Capacity: 23.21 TB
Related Objects
Datacenter
SAGRID-TEAM
Tags
Assigned Tag
Category
Description
No items to display
Assign...
Remove...
Cluster Resources
Cluster Consumers
Custom Attributes
Attribute
Value
No items to display
Edit
Update Manager
Compliance
Precheck Remediation State
Remediation Status Unknown (last check n/a)
Recent Tasks
Alarms
Task Name
Target
Status
Initiator
Queued For
Start Time
Completion Time
Server
Update option values
10.31.230.45
Completed
SAGRID\nvadmin
undefined
02/27/2019, 8:17:07 AM
02/27/2019, 8:17:07 AM
sagrid-vcenter.sagrid.local
Reconfigure scheduled task
Compute04
Completed
SAGRID\nvadmin
undefined
02/27/2019, 8:14:57 AM
02/27/2019, 8:14:57 AM
sagrid-vcenter.sagrid.local
Reconfigure scheduled task
Compute04
Completed
SAGRID\nvadmin
undefined
02/27/2019, 8:14:41 AM
02/27/2019, 8:14:41 AM
sagrid-vcenter.sagrid.local
Reconfigure scheduled task
Compute03
Completed
SAGRID\nvadmin
undefined
02/27/2019, 8:14:28 AM
02/27/2019, 8:14:28 AM
sagrid-vcenter.sagrid.local
All
More Tasks


nvadmin@Compute01: ~
00: T4-16Q 02:02.0 SM[0%] FB[7%] CL[
PID GPU S SM% FB% MC% EN% DE% PROCESS
1766 0 R 0.0 0.0 0.0 0.0 0.0 /usr/lib/xorg/
nvadmin@Compute02: ~
00: T4-16Q 02:02.0 SM[0%] FB[7%] CL[
PID GPU S SM% FB% MC% EN% DE% PROCESS
1823 0 R 0.0 0.0 0.0 0.0 0.0 /usr/lib/xorg/
nvadmin@Compute03: ~
00: T4-16Q 02:02.0 SM[0%] FB[7%] CL[
PID GPU S SM% FB% MC% EN% DE% PROCESS
1824 0 R 0.0 0.0 0.0 0.0 0.0 /usr/lib/xorg/
nvadmin@Compute04: ~
00: T4-16Q 02:02.0 SM[0%] FB[7%] CL[
PID GPU S SM% FB% MC% EN% DE% PROCESS
F1Help F2Setup F3Search F4Filter F5Start F6Stop
nvadmin@Compute05: ~
00: V100D-32Q 02:02.0 SM[0%] FB[7%] CL[-MHz
PID GPU S SM% FB% MC% EN% DE% PROCESS
1779 0 R 0.0 0.0 0.0 0.0 0.0 /usr/lib/xorg/Xorg -c
nvadmin@Compute06: ~
top - 11:39:15 up 1:02, 2 users, load average: 0.15, 0.57, 0.50
Tasks: 258 total, 1 running, 257 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.5 us, 0.8 sy, 0.2 ni, 98.5 id, 0.0 wa, 0.0 hi, 0.0 si,
KiB Mem : 16432396 total, 14121776 free, 707096 used, 1603524 buff/ca
KiB Swap: 998396 total, 998396 free, 0 used. 15321856 avail M
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMM
1558 root 20 0 4051932 194228 19572 S 0.7 1.2 0:17.52 java
14061 root 30 10 3666232 205088 19780 S 0.7 1.2 0:12.10 java
2573 nvadmin 20 0 41904 3804 3096 R 0.3 0.0 0:01.23 top
nvadmin@Compute07: ~
00: RTX6000-24Q 02:02.0 SM[0%] FB[7%] CL[
PID GPU S SM% FB% MC% EN% DE% PROCESS
1882 0 R 0.0 0.0 0.0 0.0 0.0 /usr/lib/xorg/Xorg -co
F1Help F2Setup F3Search F4Filter F5Start F6Stop
slurm@slurm-ctrl: ~/jobs
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages erver
Quick connect...
5. /home/moba
6. slurm@slurm
slurm@slurm-ctrl:~/jobs$

ENVIRONMENT MONITORING



FUTURE NEEDS AND ASKS

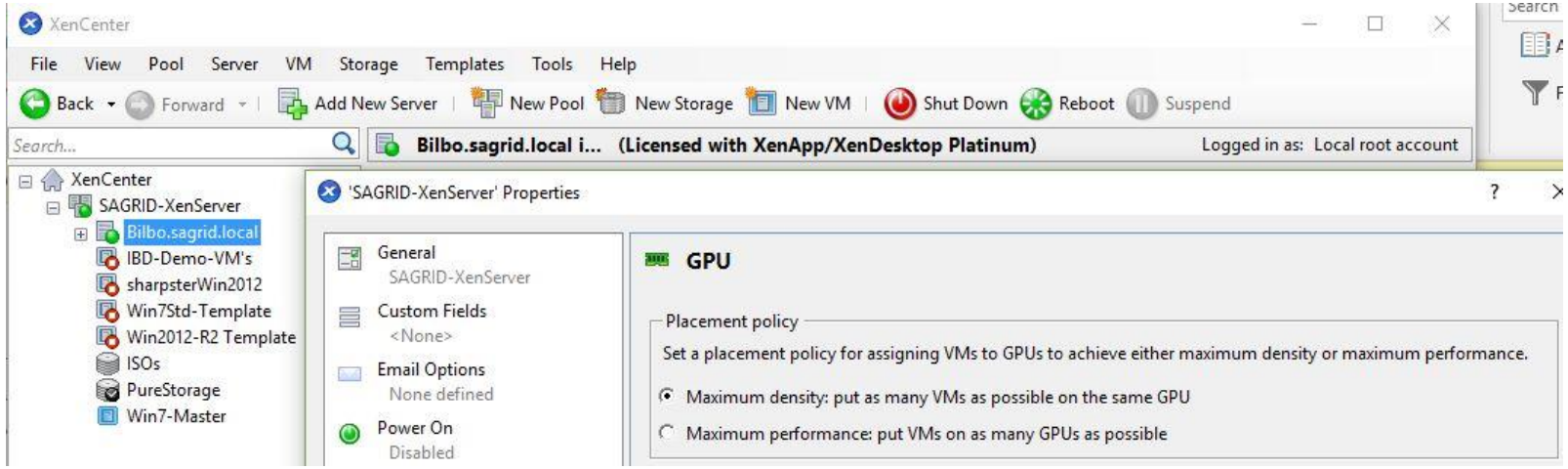
- Multiple GPUs per VM – limited availability today
- Dynamic vGPU assignment per Template provisioning
- Dynamic vGPU on live migration
- vGPU + GPU ECC + UVM + P2P – supports relevant compute
- vGPU + GPU memory Page retirement
- VM snapshots and user sessions
- Storage optimizations
- Live migration 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 ..**

```
[root@dc1vmhvesx45:~] cat /etc/vmware/config
.encoding = "UTF-8"
libdir = "/usr/lib/vmware"
authd.proxy.nfc = "vmware-hostd:ha-nfc"
authd.proxy.nfcssl = "vmware-hostd:ha-nfcssl"
authd.proxy.vpxa-nfcssl = "vmware-vpxa:vpxa-nfcssl"
authd.proxy.vpxa-nfc = "vmware-vpxa:vpxa-nfc"
authd.fullpath = "/sbin/authd"
vGPU.consolidation = "TRUE"
[root@dc1vmhvesx45:~]
```

For Citrix, its 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