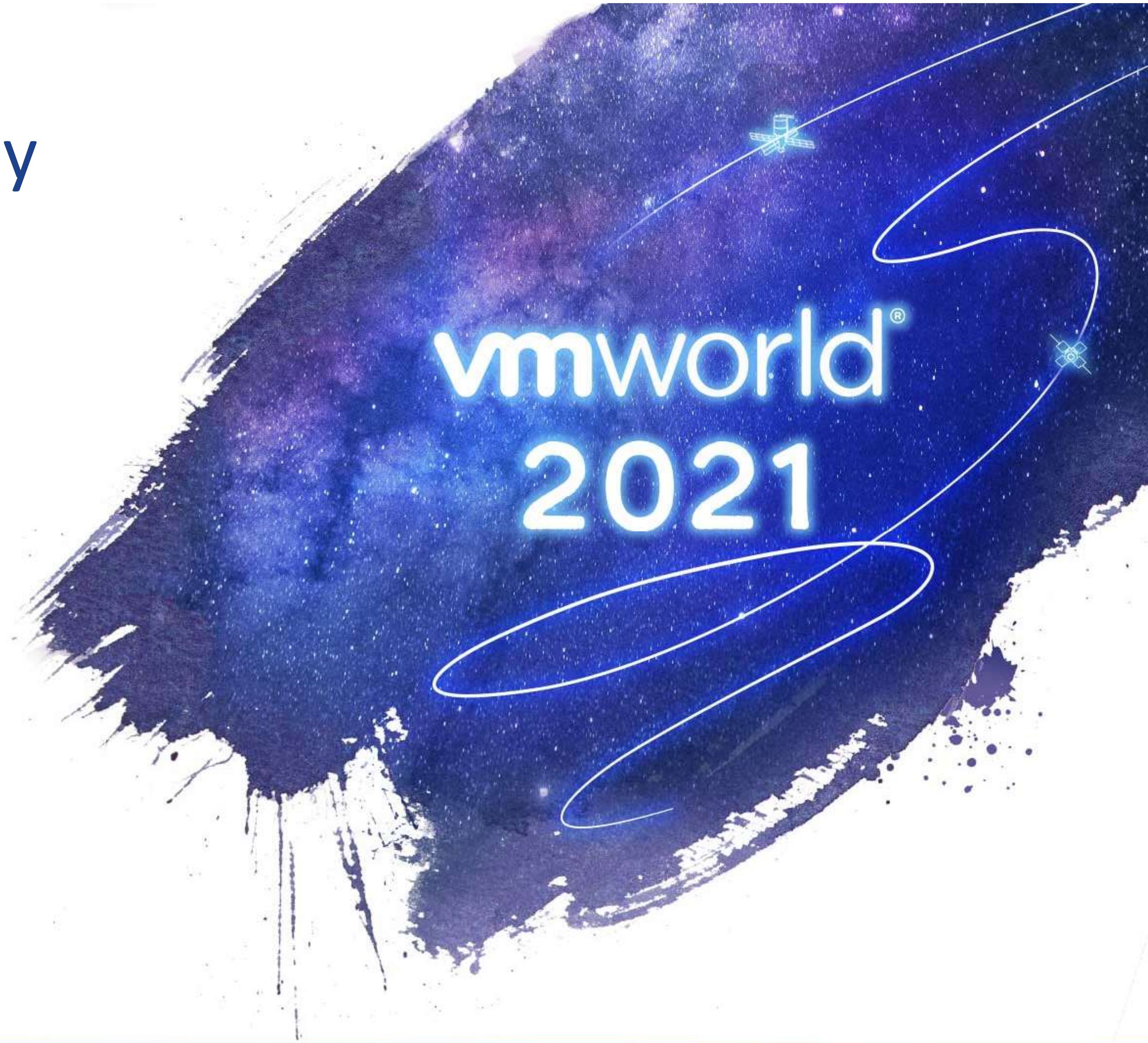
Update to VDI by Day Compute by Night

Now with more vGPUs!



Tony Foster
Principal Technical Marketing Engineer
Dell Technologies – Integrated Solutions Group

VMTN2835
#vmworld #VMTN2835

#vBrownBag

Required Disclaimer for All Presentations

- This presentation may contain product features or functionality that are currently under development.
- This overview of new technology represents no commitment from VMware to deliver these features in any generally available product.
- Features are subject to change, and must not be included in contracts, purchase orders, or sales agreements of any kind.
- Technical feasibility and market demand will affect final delivery.
- Pricing and packaging for any new features/functionality/technology discussed or presented, have not been determined.

At a glance

Overview of VDI by Day Compute by Night

New Features

- Dynamic vGPU Detection
- Support for most NVIDIA Ampere GPUs
- Parameter adjustments

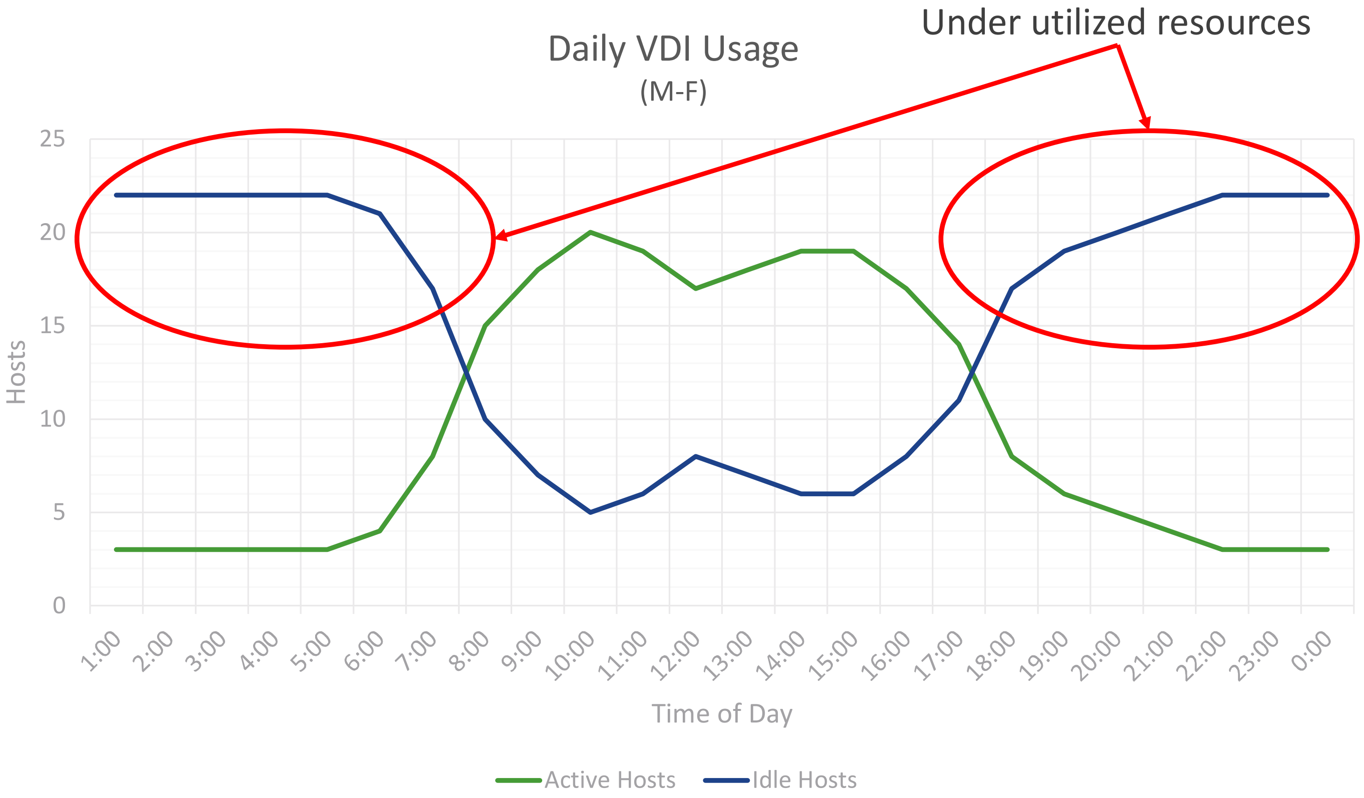
What's Next

- Speed it up
- Support for NVIDIA A16 GPU
- A real application

Demo Time

Resources

Overview – What Spare Resources?

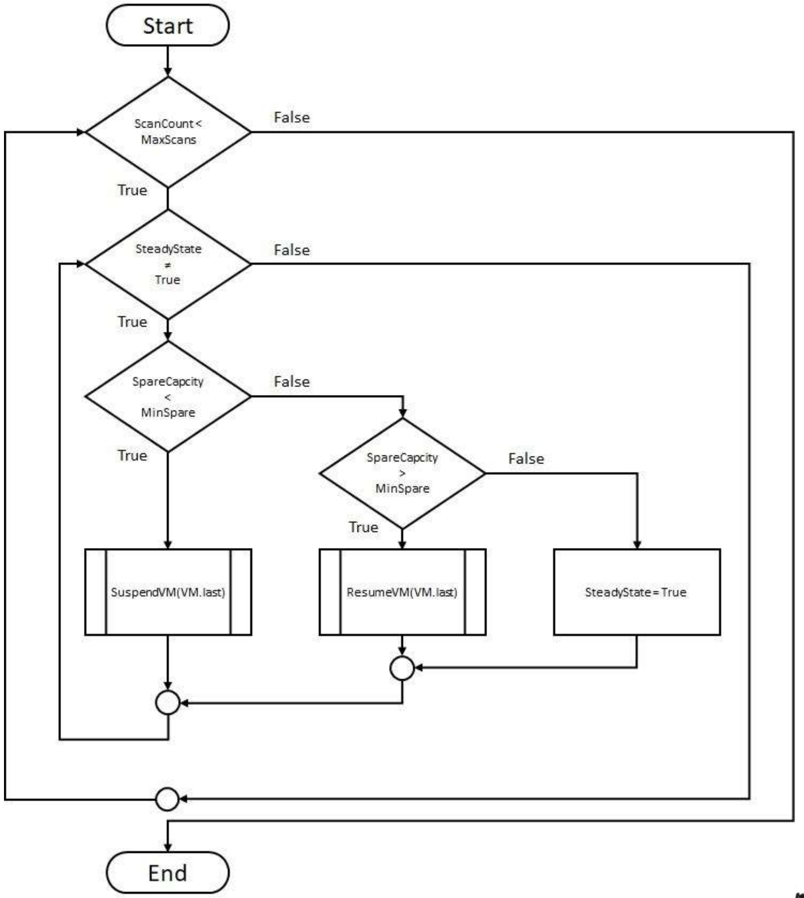


Time	Active Hosts
1:00	3
4:00	3
8:00	15
12:00	17
13:00	18
16:00	17
20:00	6
0:00	3

Overview – How it works

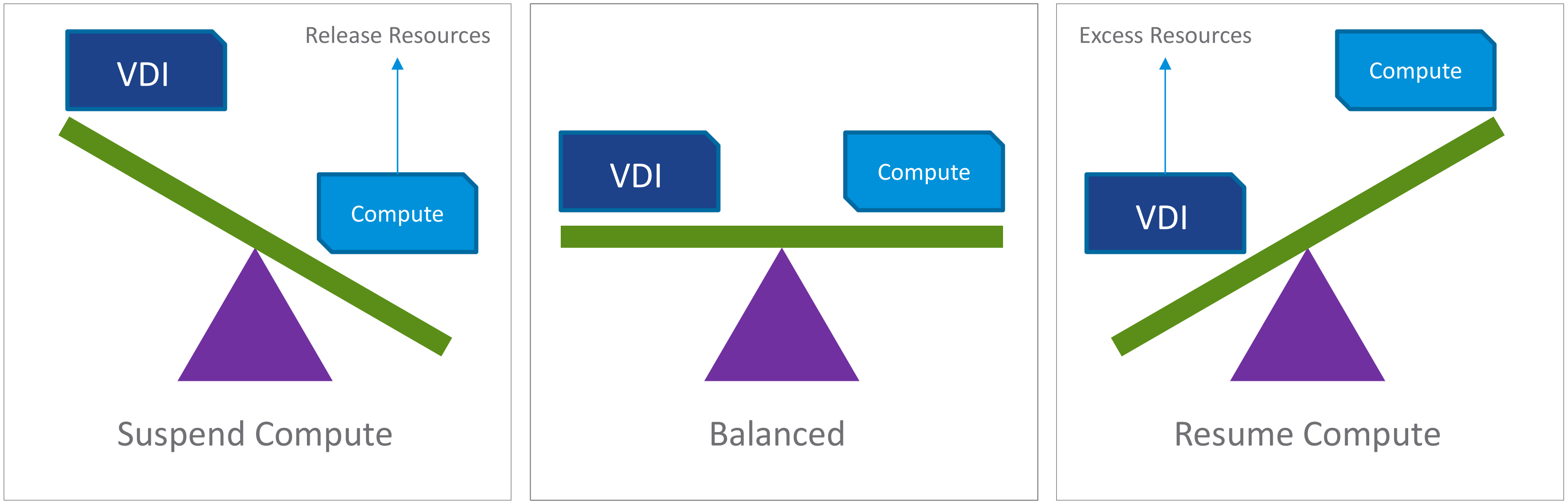
Simple approach:

- If there are free resources use them for AI Resume AI VM n
- If there are not enough resources for VDI Suspend AI VM n
- Maintain a steady state
- Repeat



#vmworld #VMTN2835

Overview – What it is Doing



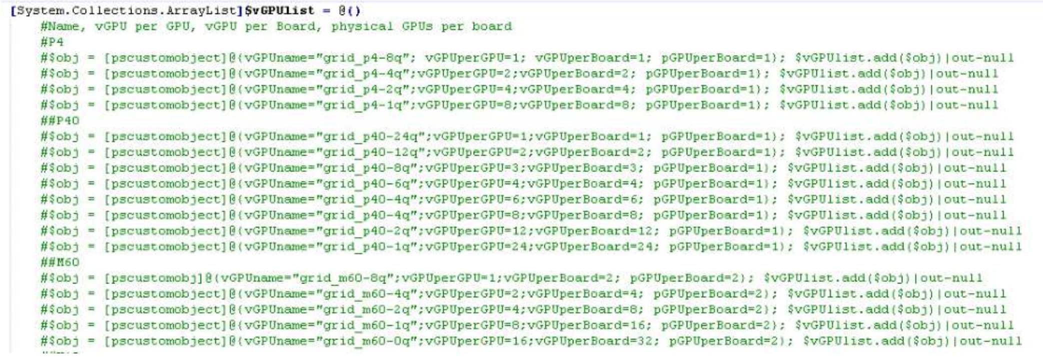
New Features – Dynamic vGPU Detection

Past:

- All vGPU profiles placed in an array

Results:

- Slower runtime
- Extra lines of code
- Manual additions of GPUs



```
[System.Collections.ArrayList]$vGPUlist = @()
    #Name, vGPU per GPU, vGPU per Board, physical GPUs per board
    #P4
    #$obj = [pscustomobject]@{vGPUname="grid_p4-8q"; vGPUperGPU=1; vGPUperBoard=1; pGPUperBoard=1}; $vGPUlist.add($obj)|out-null
    #$obj = [pscustomobject]@{vGPUname="grid_p4-4q";vGPUperGPU=2;vGPUperBoard=2; pGPUperBoard=1}; $vGPUlist.add($obj)|out-null
    #$obj = [pscustomobject]@{vGPUname="grid_p4-2q";vGPUperGPU=4;vGPUperBoard=4; pGPUperBoard=1}; $vGPUlist.add($obj)|out-null
    #$obj = [pscustomobject]@{vGPUname="grid_p4-1q";vGPUperGPU=8;vGPUperBoard=8; pGPUperBoard=1}; $vGPUlist.add($obj)|out-null
    ##P40
    #$obj = [pscustomobject]@{vGPUname="grid_p40-24q";vGPUperGPU=1;vGPUperBoard=1; pGPUperBoard=1}; $vGPUlist.add($obj)|out-null
    #$obj = [pscustomobject]@{vGPUname="grid_p40-12q";vGPUperGPU=2;vGPUperBoard=2; pGPUperBoard=1}; $vGPUlist.add($obj)|out-null
    #$obj = [pscustomobject]@{vGPUname="grid_p40-8q";vGPUperGPU=3;vGPUperBoard=3; pGPUperBoard=1}; $vGPUlist.add($obj)|out-null
    #$obj = [pscustomobject]@{vGPUname="grid_p40-6q";vGPUperGPU=4;vGPUperBoard=4; pGPUperBoard=1}; $vGPUlist.add($obj)|out-null
    #$obj = [pscustomobject]@{vGPUname="grid_p40-4q";vGPUperGPU=6;vGPUperBoard=6; pGPUperBoard=1}; $vGPUlist.add($obj)|out-null
    #$obj = [pscustomobject]@{vGPUname="grid_p40-4q";vGPUperGPU=8;vGPUperBoard=8; pGPUperBoard=1}; $vGPUlist.add($obj)|out-null
    #$obj = [pscustomobject]@{vGPUname="grid_p40-2q";vGPUperGPU=12;vGPUperBoard=12; pGPUperBoard=1}; $vGPUlist.add($obj)|out-null
    #$obj = [pscustomobject]@{vGPUname="grid_p40-1q";vGPUperGPU=24;vGPUperBoard=24; pGPUperBoard=1}; $vGPUlist.add($obj)|out-null
    ##M60
    #$obj = [pscustomobj]@{vGPUname="grid_m60-8q";vGPUperGPU=1;vGPUperBoard=2; pGPUperBoard=2}; $vGPUlist.add($obj)|out-null
    #$obj = [pscustomobject]@{vGPUname="grid_m60-4q";vGPUperGPU=2;vGPUperBoard=4; pGPUperBoard=2}; $vGPUlist.add($obj)|out-null
    #$obj = [pscustomobject]@{vGPUname="grid_m60-2q";vGPUperGPU=4;vGPUperBoard=8; pGPUperBoard=2}; $vGPUlist.add($obj)|out-null
    #$obj = [pscustomobject]@{vGPUname="grid_m60-1q";vGPUperGPU=8;vGPUperBoard=16; pGPUperBoard=2}; $vGPUlist.add($obj)|out-null
    #$obj = [pscustomobject]@{vGPUname="grid_m60-0q";vGPUperGPU=16;vGPUperBoard=32; pGPUperBoard=2}; $vGPUlist.add($obj)|out-null
```

Today:

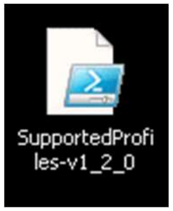
- Profiles are dynamically detected

Results:

- Quicker overall execution
- Fewer lines of code
- New GPUs easily supported

Supported Profiles Module

- Module available on https://github.com/wondernerd/
- Incorporated into script



New Features – Ampere & Parameters

Support for Ampere GPUs

- A100*, A40, & A10
- A16 not generally available

*A100 support is for compute workloads only

Updated Parameters

- State commands only accepts a single state argument
- `$vGPUSystemCapacity $vGPUtype $Cluster "connected"`
- Functions default to “`connected`” state
- Accepted Values are:
  - `connected, disconnected, notresponding`
- Due to changes in `Get-VMhost` command in PowerCLI

What’s Next

Speeding up for bigger environments

- Currently using `Get-VMhost`
- 20+ hosts can take 5+ minutes

- Code re-written with `Get-View`
- Expected to reduce cycles to 1 minute or less

Support for NVIDIA A16 GPU

- Waiting on release & final documentation
- May require some additional code
- Not ideal for Compute workloads

Dynamic Resource Optimizer

- VM based
- Python based infrastructure
- Robust command set

Demo

Resources

Details on VDI by Day Compute by Night: www.VDIbyDayComputeByNight.com

My blog: www.wondernerd.net

Get the code: www.github.com/wondernerd

Join the community http://code.vmware.com

Reach out:

Twitter @wonder_nerd

LinkedIn.com/in/wondernerd

Sessions you don’t want to miss!

[CODE2778] Talk Nerdy to Me, Using Python to Create VMs with vGPUs for AI Workloads

[EUS1289] VDI Nerdfest 2021: Demos That Make Admins Drool

[EUS3107] Nerd Tours: A Tech Deep Dive of the VDI NerdFest 2021 Extravaganza

[VI2222] Got GPUs? Learn How to Set Up Self-Service Access for AI/ML.

[VI1459] Best Practices for Running AI Workloads in VMs on VMware vSphere

[VI1559] vSphere Admin’s Guide to Virtual AI Infrastructure for Modern Data Science

Please take your survey.

Thank you!