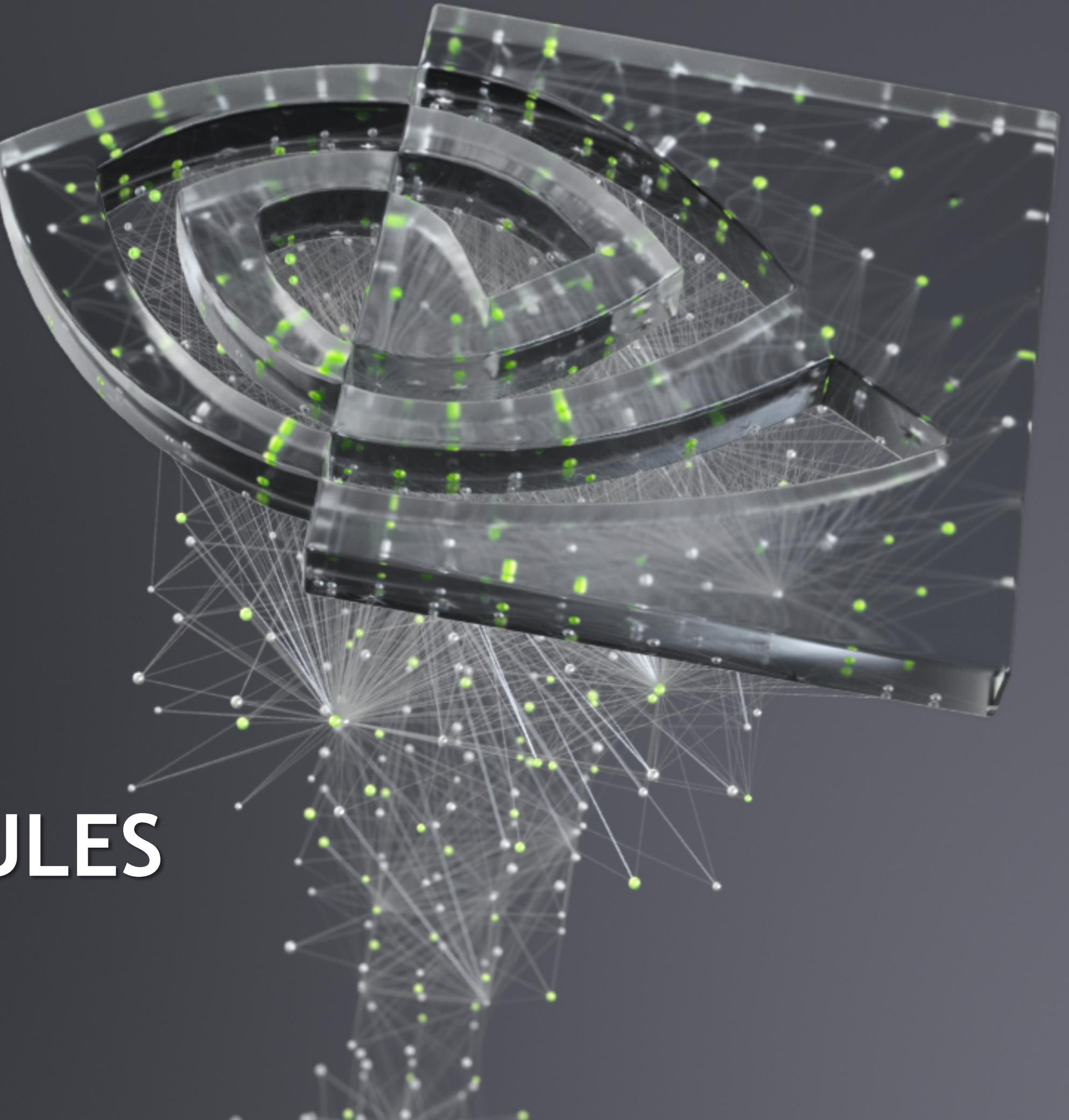


NVIDIA®

NGC CONTAINER ENVIRONMENT MODULES

Scott McMillan



ENVIRONMENT MODULES + CONTAINERS

- Environment modules are the de facto method for managing HPC software environments
- Containers can complement traditional software deployment methods by bundling the software and all its dependencies in a single image for enhanced portability and reproducibility
 - The NVIDIA NGC Catalog has up-to-date, GPU optimized containers for HPC, Deep Learning, and AI software
- But ... containers introduce a new workflow for users
- What if containers were available using the environment modules interfaces that users and systems professionals are already familiar with?

NGC Container Environment Modules are an environment modules interface to containers

HOW IT WORKS

Prerequisites: Lmod and Singularity

User Workflow

```
$ module load tensorflow
```

```
local image = "..."  
load("Singularity")  
  
if not (isFile(image)) then  
    image = "docker://nvcr.io/nvidia/tensorflow:20.06-tf1-py3"  
end  
  
set_shell_function("python", "singularity run --nv \" ..  
image .. \" $@")
```

```
$ python model.py
```

 singularity run --nv docker://nvcr.io/nvidia/tensorflow:20.06-tf1-py3 python model.py

TRANSPARENT CONTAINERS

Basic

```
$ module load gromacs/2020.2  
$ gmx mdrun -ntmpi 1 -ntomp 40 -v -pin on -nb gpu --pme gpu --resetstep 12000 -nsteps 20000 -nstlist 400 -noconfout -s topol.tpr
```

Interactive

```
$ module load pytorch/20.02-py3  
$ python3  
->>> import torch  
->>> x = torch.randn(2,3)
```

Transparent access to host data files:
Singularity automatically binds \$HOME,
/tmp, and the current working directory
into the container

Jupyter Notebooks

```
$ module load rapidsai  
$ jupyter notebook --ip 0.0.0.0 --no-browser --notebook-dir /rapids/notebooks
```

Multi-node MPI

```
$ module load lammps/15Jun2020  
$ mpirun -n 2 lmp -in in.lj.txt -var x 8 -var y 8 -var z 8 -k on g 2 -sf kk -pk kokkos cuda/aware on neigh full comm device binsize 2.8
```

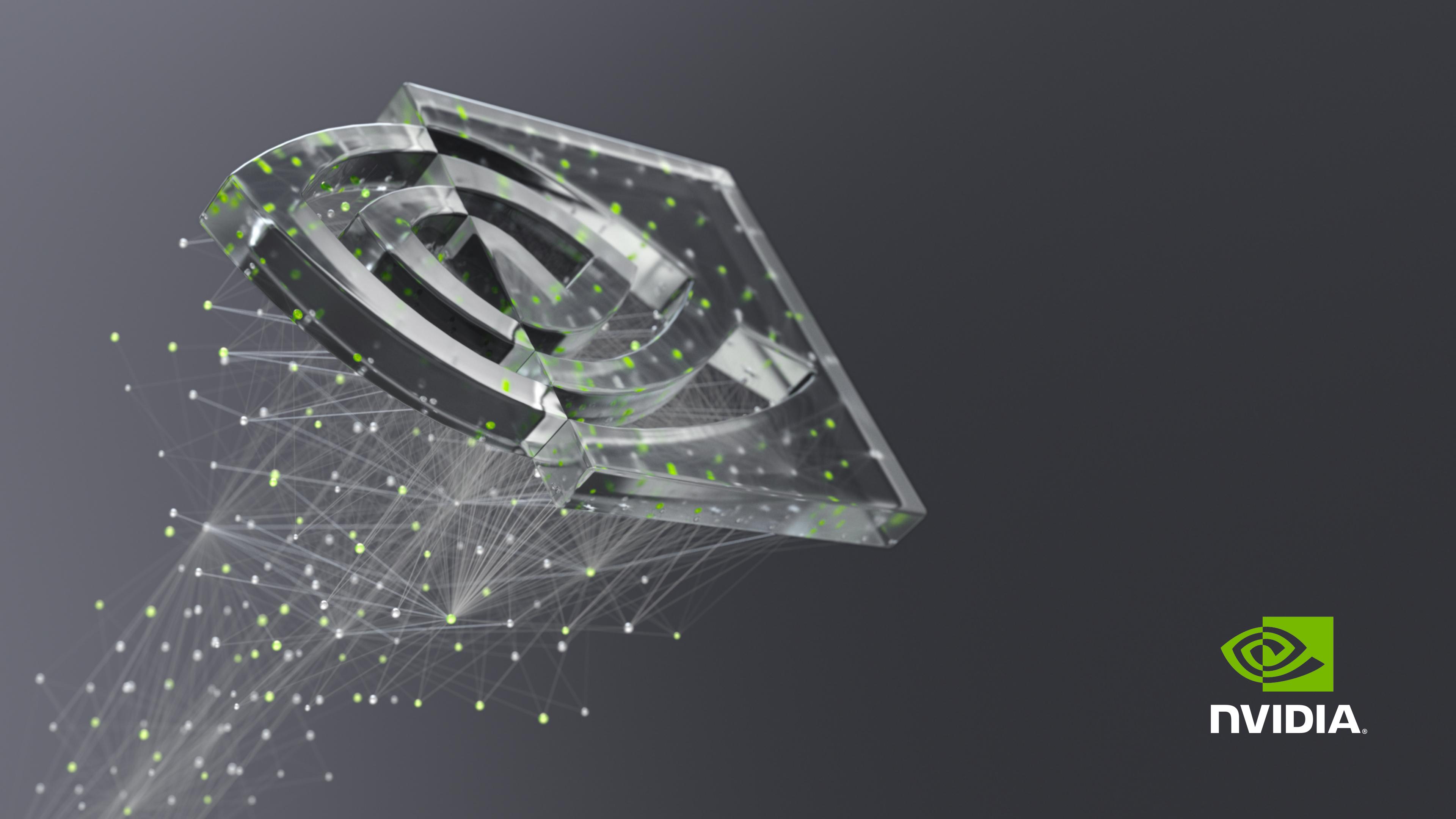
Use notebooks on host filesystem or embedded in container

No dependency on host MPI runtime and transparent launch of MPI ranks on remote nodes by setting
`OMPI_MCA_orte_launch_agent="singularity run --nv image orted"`

NGC CONTAINER ENVIRONMENT MODULES

github.com/nvidia/ngc-container-environment-modules

- NGC container environment modules are open, lightweight wrappers that make it possible to transparently use NGC containers as environment modules
 - Ready to download and add to MODULEPATH
 - Expected site specific customizations:
 - The name of the Singularity module
The container environment modules try to load the Singularity module (note the capital 'S'). Modify this if the local Singularity module is named differently.
 - Module conflicts
The container environment modules set module conflicts based on the commands mapped into the container. Sites may want to modify the list of conflicting modules to prevent conflicts between containers or other environment modules.
 - Container image cache
The container environment modules can either pull NGC container images as needed or use a library of pre-downloaded container images. Sites may wish to modify the modules to only support one of these modes. Sites may also want to hard-code the path to the library of pre-downloaded container images rather than using the `NGC_IMAGE_DIR` environment variable.
 - Mount additional directories into the containers
Sites may have filesystems that should be visible to all user processes, such as a `/scratch` filesystem or a set of shared datasets. In this case, set `SINGULARITY_BINDPATH`, either globally, or in the container environment modules.



NVIDIA®