

# GPU nodes in Tetralith

## HARDWARE:

- 170 retrofitted thin nodes
- 96 GiB of primary memory (RAM)
- One NVIDIA Tesla T4 GPU (Turing)
- One NVMe SSD scratch disk of ~2TiB

Further reading: [www.nsc.liu.se/systems/tetralith/](http://www.nsc.liu.se/systems/tetralith/)

**TESLA T4 SPECIFICATIONS:**

compute capability	7.5
tensor cores	320
CUDA cores	2560
single precision performance	8.1 TFLOPS
memory	16 GB
power	70 W

**PRIMARILY SUITABLE FOR:**

- Machine learning
- Single precision FP (*e.g.* MD)
- Hardware accelerated graphics

Available to all projects with allocations on Tetralith!

# ALLOCATING A GPU NODE

Using `interactive`:

```
1 [x_torra@tetralith2]$ interactive -n 1 -c 32 --gpus-per-task=1 -t 60 -A snic2022-5-301 --reservation=devel
2 salloc: Pending job allocation 22767410
3 salloc: job 22767410 queued and waiting for resources
4 salloc: job 22767410 has been allocated resources
5 salloc: Granted job allocation 22767410
6 salloc: Waiting for resource configuration
7 salloc: Nodes n86 are ready for job
8 [x_torra@n86]$
```

`-n 1 -c 32` or `(-N 1)`

Allocate a complete compute node

`--gpus-per-task=1` or `(--gpus=1)`

Allocates the GPU

`-A "slurm account"`

Only needed if you are included in several projects

`--reservation=devel`

Is for short (*max* 60 min.) jobs

Don't use for longer jobs!

*Always allocate a full compute node when using a GPU on Tetralith!*

## ALLOCATING A GPU NODE

Batch script header:

```
#!/bin/bash

#SBATCH --ntasks=1
#SBATCH --cpus-per-task=32
#SBATCH --gpus-per-task=1
#SBATCH --time=24:00:00
#SBATCH --account=snic2022-5-301
:
```

Here I've used long options (e.g. `--ntasks=1`), but short options (e.g. `-n 1`) also work!

Further reading: [www.nsc.liu.se/support/systems/tetralith-GPU-user-guide/](http://www.nsc.liu.se/support/systems/tetralith-GPU-user-guide/)

```
[x_torra@tetralith2]$ interactive -n 1 -c 32 --gpus-per-task=1 -t 60 --reservation=devel
salloc: Pending job allocation 22767410
salloc: job 22767410 queued and waiting for resources
salloc: job 22767410 has been allocated resources
salloc: Granted job allocation 22767410
salloc: Waiting for resource configuration
salloc: Nodes n86 are ready for job
[x_torra@n86]$ nvidia-smi
Mon Nov 14 11:43:42 2022
```

```
+-----+
| NVIDIA-SMI 515.65.01      Driver Version: 515.65.01      CUDA Version: 11.7      |
+-----+-----+-----+
| GPU  Name          Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                               |                  |              MIG M. |
+-----+-----+-----+
|   0   Tesla T4             On      | 00000000:3B:00.0 Off  |          0          |
| N/A   44C    P8     16W /  70W |  2MiB / 15360MiB |      0%      Default |
|                               |                  |              N/A     |
+-----+-----+-----+
```

```
+-----+
| Processes:
| GPU   GI    CI          PID    Type    Process name          GPU Memory
|       ID    ID                                   Usage
+-----+
| No running processes found
+-----+
```

## ALLOCATING A GPU NODE FOR GRAPHICS

1. Login with ThinLinc!
2. Allocate a GPU node using `interactive.vgl`
3. Launch GUI with `vglrun gui_name`

```
[x_torra@tetralith1]$ interactive.vgl -t 60 --reservation=devel
Enabling VirtualGL mode.
Adding --exclusive option. Note: your project will be charged for full nodes!
Adding --constraint=virtualgl to enable VirtualGL.
Adding --gres=gpu to allocate GPU to job.
Allocating one GPU for the interactive shell to allow accelerated graphics. Note: GPU will not be available from e.g job step
Remember to use "vglrun <application>" to enable accelerated graphics for <application>.
salloc: Pending job allocation 22779726
salloc: job 22779726 queued and waiting for resources
salloc: job 22779726 has been allocated resources
salloc: Granted job allocation 22779726
salloc: Waiting for resource configuration
salloc: Nodes n1091 are ready for job
[x_torra@n1091]$
```

Further reading: [www.nsc.liu.se/support/graphics/](http://www.nsc.liu.se/support/graphics/)



torbenr@tetralith2.nsc.liu.se - ThinLinc Client

MATLAB R2020b - academic use

19:51 Torben Rasmussen

HOME PLOTS APPS LIVE EDITOR INSERT VIEW

Current Folder: /proj/nsc/users/torbenr/jobs/matlab/CardiacMRI-master/Part02\_Modeling.mlx

### Semantic Segmentation Transfer Learning

We are now ready to actually train our network. Let's set up some training options and get to work. Our network was trained on four NVIDIA® V100 Tensor Core GPUs on the cloud, taking approximately one hour of training time.

```

64 if doTraining
65     options = trainingOptions('sgdm', ...
66         'Momentum', 0.9, ...
67         'InitialLearnRate', 0.0002, ...
68         'L2Regularization', 0.0005, ...
69         'MaxEpochs', 100, ...
70         'MiniBatchSize', 4, ...
71         'Shuffle', 'every-epoch', ...
72         'VerboseFrequency', 100, ...
73         'ValidationData', valds, ...
74         'ValidationPatience', 5, ...
75         'Plots', 'training-progress', ...
76         'ExecutionEnvironment', 'gpu');
77     tic
78     [net, info] = trainNetwork(trainds, lgraph, options);
79     toc
80 else
81     imshow(fullfile(prj.RootFolder, "HelperFunctions", "Images", "SegnetTrainingProgressPlot.png"));
82 end

```

Initializing input data normalization.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Base Learning Rate
1	1	00:00:28	63.13%	63.11%	0.9535	0.9476	0.0002
1	50	00:01:02	63.59%	63.27%	0.9687	0.9473	0.0002
1	100	00:01:34	63.25%	63.47%	0.9245	0.9468	0.0002
1	150	00:02:07	63.98%	63.62%	0.9672	0.9466	0.0002
2	200	00:02:38	64.04%	63.79%	0.9513	0.9463	0.0002
2	250	00:03:11	63.65%	63.94%	0.9282	0.9461	0.0002
2	300	00:03:43	63.92%	64.08%	0.9314	0.9459	0.0002
3	350	00:04:15	64.06%	64.19%	0.9385	0.9458	0.0002
3	400	00:04:48	64.23%	64.32%	0.9356	0.9456	0.0002
3	450	00:05:20	64.34%	64.45%	0.9456	0.9455	0.0002
4	500	00:05:52	64.27%	64.56%	0.9270	0.9454	0.0002

Workspace: ans 1x1 struct

Command Window: New to MATLAB? See resources for [Getting Started](#).

UTF-8 script Ln 83 Col 1

```

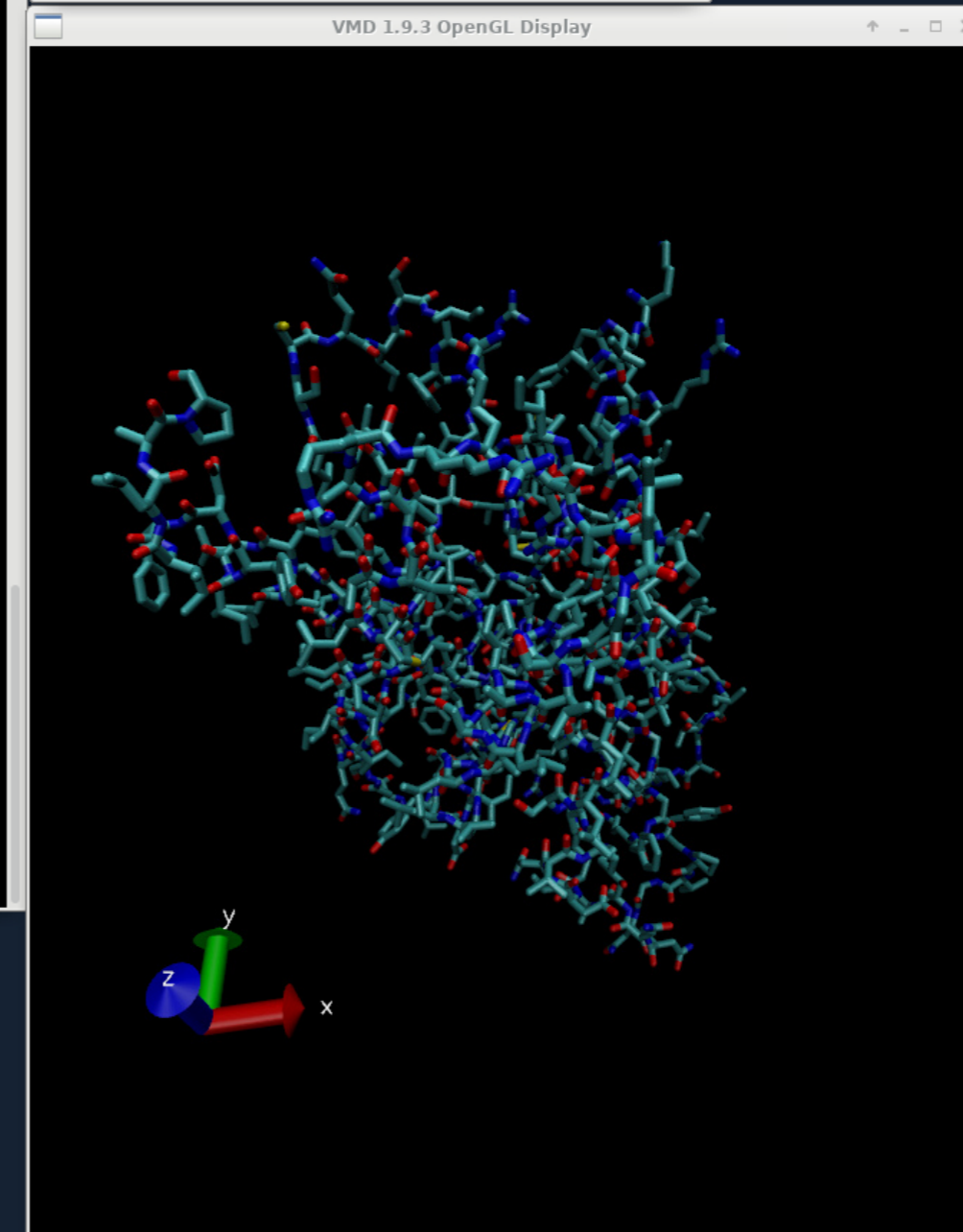
Terminal - torbenr@n99: vmd
File Edit View Terminal Tabs Help
Info) http://www.ks.uiuc.edu/Research/vmd/
Info) Email questions and bug reports to vmd@ks.uiuc.edu
Info) Please include this reference in published work using VMD:
Info)   Humphrey, W., Dalke, A. and Schulten, K., 'VMD - Visual
Info)   Molecular Dynamics', J. Molec. Graphics 1996, 14.1, 33-38.
Info) -----
Info) Multithreading available, 32 CPUs detected.
Info) CPU features: SSE2 AVX AVX2 FMA KNL:AVX-512F+CD+ER+PF
Info) Free system memory: 88GB (94%)
Info) Creating CUDA device pool and initializing hardware...
Info) Detected 1 available CUDA accelerator:
Info) [0] Tesla T4          40 SM 7.5 @ 1.59 GHz, 15GB RAM, KTO, AE3, ZCP
Warning) Detected X11 'Composite' extension: if incorrect display occurs
Warning) try disabling this X server option. Most OpenGL drivers
Warning) disable stereoscopic display when 'Composite' is enabled.
Info) OpenGL renderer: Tesla T4/PCIe/SSE2
Info) Features: STENCIL MSAA(4) MDE CVA MTX NPOT PP PS GLSL(OVFGS)
Info) Full GLSL rendering mode is available.
Info) Textures: 2-D (32768x32768), 3-D (16384x16384x16384), Multitexture (4)
Info) Detected 1 available TachyonL/OptiX ray tracing accelerator
Info) Compiling 1 OptiX shaders on 1 target GPU...
Info) Dynamically loaded 2 plugins in directory:
Info) /software/sse/manual/vmd/1.9.3/nsc1/lib/vmd/plugins/LINUXAMD64/molfile
vmd > Info) Using plugin pdb for structure file /proj/nsc/users/torbenr/jobs/vmd/1fqy.pdb
Info) Using plugin pdb for coordinates from file /proj/nsc/users/torbenr/jobs/vmd/1fqy.pdb
Info) Determining bond structure from distance search ...
Info) Analyzing structure ...
Info)   Atoms: 1661
Info)   Bonds: 1693
Info)   Angles: 0 Dihedrals: 0 Improper: 0 Cross-terms: 0
Info)   Bondtypes: 0 Angletypes: 0 Dihedraltypes: 0 Improper types: 0
Info)   Residues: 226
Info)   Waters: 0
Info)   Segments: 1
Info)   Fragments: 1 Protein: 1 Nucleic: 0
Info) Finished with coordinate file /proj/nsc/users/torbenr/jobs/vmd/1fqy.pdb.

```

VMD Main

ID	T	A	D	F	Molecule	Atoms	Frames	Vol
0	T	A	D	F	1fqy.pdb	1661	1	0

zoom  Loop step 1 speed





# USING SINGULARITY/APPTAINER AND NGC

**NGC Catalog:** Software Hub with containers with a range of GPU-accelerated software for NVIDIA GPUs

```
[torbenr@tetralith1]$ interactive -n 1 -c 32 --gpus-per-task=1 -t 60 -A nsc --reservation=devel
salloc: Pending job allocation 22784449
salloc: job 22784449 queued and waiting for resources
salloc: job 22784449 has been allocated resources
salloc: Granted job allocation 22784449
salloc: Waiting for resource configuration
salloc: Nodes n1130 are ready for job
[torbenr@n1130]$ export APPTAINER_BIND="/proj,/scratch/local,/software:/software:ro"
[torbenr@n1130]$ apptainer run --nv tf20.09_py3.v3.sif
Usage example: change_mofed_version.sh 4.5-1.0.1

=====
== TensorFlow ==
=====

NVIDIA Release 20.09-tf2 (build 16003717)
TensorFlow Version 2.3.0

Container image Copyright (c) 2020, NVIDIA CORPORATION. All rights reserved.
Copyright 2017-2020 The TensorFlow Authors. All rights reserved.

Various files include modifications (c) NVIDIA CORPORATION. All rights reserved.
NVIDIA modifications are covered by the license terms that apply to the underlying project or file.

Detected MOFED .

Singularity>
```

## Use the python environment in the container:

```
Singularity> jupyter-notebook --no-browser --ip=n1130
2022-11-15 14:37:00.421476: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library
[I 14:37:03.099 NotebookApp] jupyter_tensorboard extension loaded.
[I 14:37:03.494 NotebookApp] JupyterLab extension loaded from /usr/local/lib/python3.6/dist-packages/jupyterlab
[I 14:37:03.494 NotebookApp] JupyterLab application directory is /usr/local/share/jupyter/lab
[I 14:37:03.712 NotebookApp] [JupyterText Server Extension] NotebookApp.contents_manager_class is (a subclass of) jupyterlab.Text
[I 14:37:03.714 NotebookApp] Serving notebooks from local directory: /proj/nsc/users/torbenr/jobs/ngc
[I 14:37:03.714 NotebookApp] The Jupyter Notebook is running at:
[I 14:37:03.714 NotebookApp] http://n1130:8900/?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d
[I 14:37:03.714 NotebookApp] or http://127.0.0.1:8900/?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d
[I 14:37:03.714 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 14:37:03.801 NotebookApp]
```

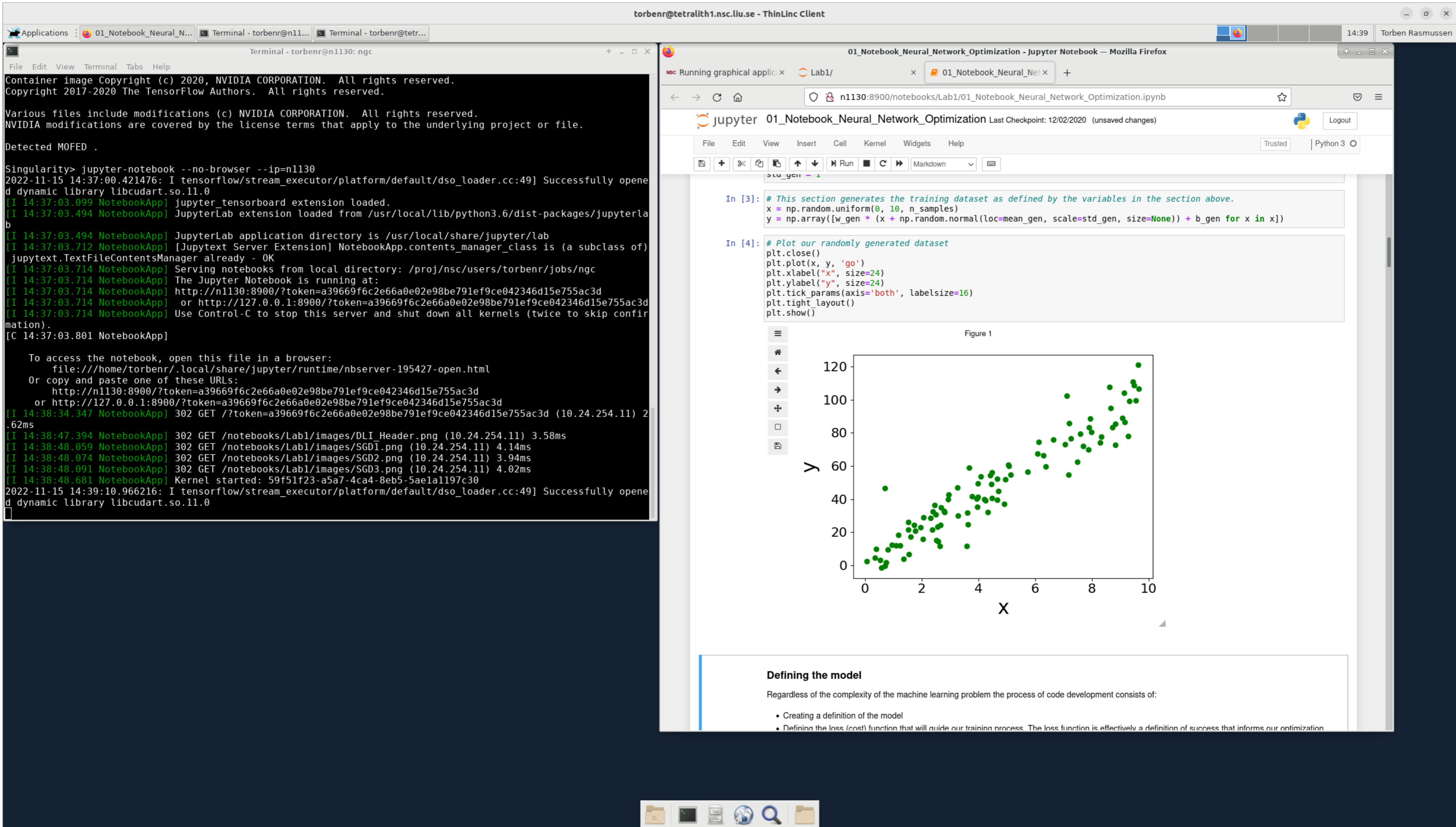
To access the notebook, open this file in a browser:

```
file:///home/torbenr/.local/share/jupyter/runtime/nbserver-195427-open.html
```

Or copy and paste one of these URLs:

```
http://n1130:8900/?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d
```

```
or http://127.0.0.1:8900/?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d
```



```
Container image Copyright (c) 2020, NVIDIA CORPORATION. All rights reserved.
Copyright 2017-2020 The TensorFlow Authors. All rights reserved.

Various files include modifications (c) NVIDIA CORPORATION. All rights reserved.
NVIDIA modifications are covered by the license terms that apply to the underlying project or file.

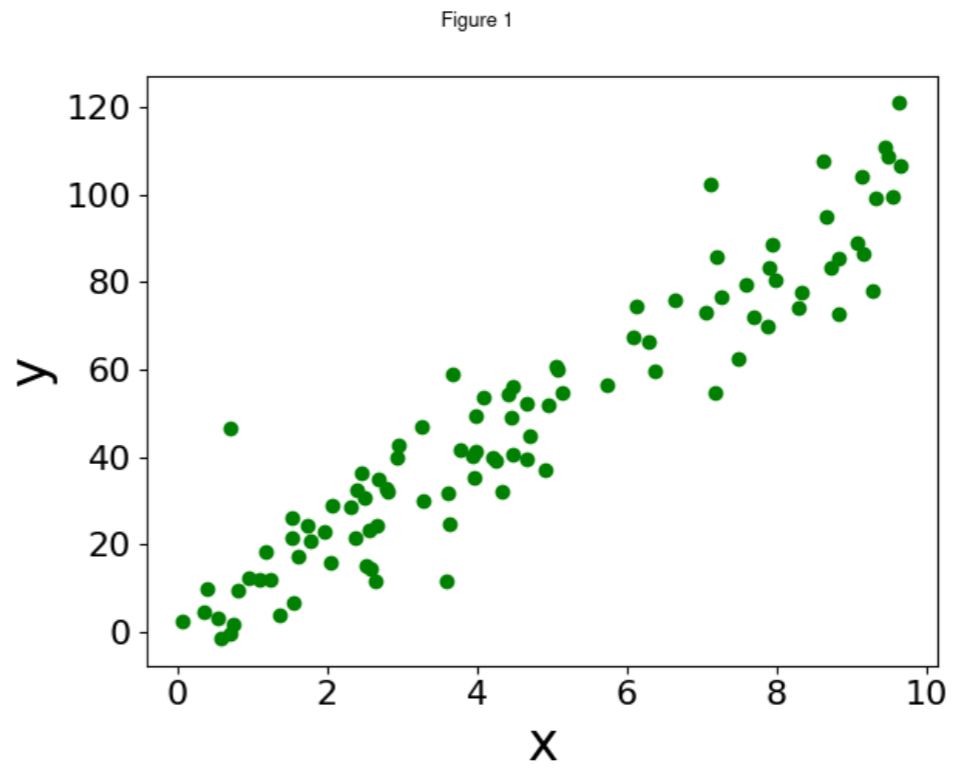
Detected MOFED .

Singularity> jupyter-notebook --no-browser --ip=n1130
2022-11-15 14:37:00.421476: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library libcudart.so.11.0
[I 14:37:03.099 NotebookApp] jupyter_tensorboard extension loaded.
[I 14:37:03.494 NotebookApp] JupyterLab extension loaded from /usr/local/lib/python3.6/dist-packages/jupyterlab
[I 14:37:03.494 NotebookApp] JupyterLab application directory is /usr/local/share/jupyter/lab
[I 14:37:03.712 NotebookApp] [Jupyter Server Extension] NotebookApp.contents_manager_class is (a subclass of)
jupyterlab.TextFileContentsManager already - OK
[I 14:37:03.714 NotebookApp] Serving notebooks from local directory: /proj/nsc/users/torbenr/jobs/ngc
[I 14:37:03.714 NotebookApp] The Jupyter Notebook is running at:
[I 14:37:03.714 NotebookApp] http://n1130:8900/?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d
[I 14:37:03.714 NotebookApp] or http://127.0.0.1:8900/?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d
[I 14:37:03.714 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 14:37:03.801 NotebookApp]

To access the notebook, open this file in a browser:
file:///home/torbenr/.local/share/jupyter/runtime/nbserver-195427-open.html
Or copy and paste one of these URLs:
http://n1130:8900/?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d
or http://127.0.0.1:8900/?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d
[I 14:38:34.347 NotebookApp] 302 GET /?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d (10.24.254.11) 2.62ms
[I 14:38:47.394 NotebookApp] 302 GET /notebooks/Lab1/images/DLI_Header.png (10.24.254.11) 3.58ms
[I 14:38:48.059 NotebookApp] 302 GET /notebooks/Lab1/images/SGD1.png (10.24.254.11) 4.14ms
[I 14:38:48.074 NotebookApp] 302 GET /notebooks/Lab1/images/SGD2.png (10.24.254.11) 3.94ms
[I 14:38:48.091 NotebookApp] 302 GET /notebooks/Lab1/images/SGD3.png (10.24.254.11) 4.02ms
[I 14:38:48.681 NotebookApp] Kernel started: 59f51f23-a5a7-4ca4-8eb5-5ae1a1197c30
2022-11-15 14:39:10.966216: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library libcudart.so.11.0
```

```
In [3]: # This section generates the training dataset as defined by the variables in the section above.
x = np.random.uniform(0, 10, n_samples)
y = np.array([w_gen * (x + np.random.normal(loc=mean_gen, scale=std_gen, size=None)) + b_gen for x in x])

In [4]: # Plot our randomly generated dataset
plt.close()
plt.plot(x, y, 'go')
plt.xlabel("x", size=24)
plt.ylabel("y", size=24)
plt.tick_params(axis='both', labels=16)
plt.tight_layout()
plt.show()
```



### Defining the model

Regardless of the complexity of the machine learning problem the process of code development consists of:

- Creating a definition of the model
- Defining the loss (cost) function that will guide our training process. The loss function is effectively a definition of success that informs our optimization

## Building an Apptainer container from an NGC Docker image:

```
[torbenr@tetralith1]$ interactive -n 1 -c 32 --gpus-per-task=1 -t 60 -A snic2020-5-235 --reservation=devel
⋮
[torbenr@n1112]$ cat sourceme.txt
export SINGULARITY_DOCKER_USERNAME='$oauthtoken'
export SINGULARITY_DOCKER_PASSWORD="long-pw-private-string"
export SINGULARITY_BIND="/proj,/scratch/local,/software:/software:ro"
[torbenr@n1112]$ . ./sourceme.txt
[torbenr@n1112]$ singularity build tf20.09_py3.sif docker://nvcr.io/nvidia/tensorflow:20.09-tf2-py3
⋮
[torbenr@n1112]$ singularity run --nv tf20.09_py3.sif
Singularity>
```

## Monitoring GPU usage:

```
[torbenr@tetralith1]$ interactive -N 1 --gpus=1 -t 60 -A snic2020-5-235 --reservation=devel
:
[torbenr@n75]$ tmux
# ^b c
[torbenr@n75]$ aptainer run --nv -B $PWD:/host_pwd --pwd /host_pwd relion_3.1.3.sif ./ngc_run_relion.sh
# ^b 0
[torbenr@n75]$ nvidia-smi
Mon Apr 24 15:17:39 2023
+-----+
| NVIDIA-SMI 515.105.01    Driver Version: 515.105.01    CUDA Version: 11.7    |
+-----+-----+-----+-----+-----+-----+
| GPU   Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                                           |              MIG M. |
+-----+-----+-----+-----+-----+-----+
|    0   Tesla T4              On          | 00000000:3B:00.0 Off  |           0          |
| N/A   40C    P0     34W / 70W | 233MiB / 15360MiB |      0%      Default |
|                                           |              N/A     |
+-----+-----+-----+-----+-----+

+-----+
| Processes: |
| GPU   GI   CI          PID    Type   Process name          GPU Memory |
|       ID   ID              |          |         | Usage              |
+-----+-----+-----+-----+-----+
|    0   N/A  N/A       66383     C   relion_refine_mpi     103MiB |
|    0   N/A  N/A       66384     C   relion_refine_mpi     103MiB |
|    0   N/A  N/A       66392     C   nvidia-cuda-mps-server 23MiB  |
+-----+-----+-----+-----+-----+

```

## Monitoring GPU usage: (load)

```
[torbenr@n75]$ nvidia-smi dmon
# gpu    pwr gtemp mtemp    sm    mem    enc    dec    mclk    pclk
# Idx    W     C     C     %     %     %     %     MHz    MHz
  0     68    67    -    94    1     0     0    5000   1155
  0     71    67    -    83    1     0     0    5000   1230
  0     67    67    -    87    1     0     0    5000   1170
  0     69    67    -    94    1     0     0    5000   1170
  0     68    67    -    93    1     0     0    5000   1155
  0     65    67    -    94    1     0     0    5000   1155
  0     69    67    -    81    1     0     0    5000   1200
  0     71    67    -    95    1     0     0    5000   1140
  0     72    67    -    89    1     0     0    5000   1140
  0     65    67    -    94    1     0     0    5000   1140
  0     70    67    -    93    1     0     0    5000   1170
  0     69    67    -    93    1     0     0    5000   1185
  :
```



# GPU nodes in Sigma

## HARDWARE:

- Two new nodes
- **Note:** 36 CPU cores
- 384 GiB of primary memory (RAM)
- Four NVIDIA Tesla V100 SXM2 GPUs (Volta)
- Two 7680GB NVMe SSD scratch disks (~14TiB total)

Further reading: [www.nsc.liu.se/systems/sigma/](http://www.nsc.liu.se/systems/sigma/)

**TESLA V100 SXM2 SPECIFICATIONS:**

compute capability	7.0
tensor cores	640
CUDA cores	5120
double precision performance	7.8 TFLOPS
single precision performance	15.7 TFLOPS
memory	32 GB
power	300 W

- Access currently restricted to Machine learning projects
- Apply using the [LiU Local GPU 2022](#) round in SUPR
- Can be used for both single and double precision FP

# ALLOCATING ONE GPU

Using `interactive`:

```
[torbenr@sigma]$ interactive -n 1 -c 9 --gpus-per-task=1 -t 60 -A LiU-gpu-2022-4 --reservation=gpu
salloc: Granted job allocation 3330796
salloc: Waiting for resource configuration
salloc: Nodes n2017 are ready for job
[torbenr@n2017]$
```

`-n 1`

Allocate 1 task

`-c 9`

Generally allocate 9 CPU-cores per GPU

`--gpus-per-task=1`

Allocates the GPU

`-A "slurm account"`

Needed unless you are only included in a LiU-gpu-202X-Y project

`--reservation=gpu`

Required flag!

## ALLOCATING ONE GPU

Multiple ranks per GPU:

```
[torbenr@sigma]$ interactive -n 2 -c 4 --gpus=1 -t 60 -A LiU-gpu-2022-4 --reservation=gpu
salloc: Granted job allocation 3330801
salloc: Waiting for resource configuration
salloc: Nodes n2017 are ready for job
[torbenr@n2017]$
```

# ALLOCATING ONE GPU

Batch script header:

```
#!/bin/bash
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=9
#SBATCH --gpus-per-task=1
#SBATCH --time=24:00:00
#SBATCH --account=LiU-gpu-2022-4
#SBATCH --reservation=gpu
:
```

Here I've used long options (e.g. `--ntasks=1`), but short options (e.g. `-n 1`) also work!

Further reading: [www.nsc.liu.se/support/systems/sigma-GPU-user-guide/](http://www.nsc.liu.se/support/systems/sigma-GPU-user-guide/)

```
[torbenr@sigma]$ interactive -n 1 -c 9 --gpus-per-task=1 -t 60 -A LiU-gpu-2022-4 --reservation=gpu
salloc: Granted job allocation 3330796
salloc: Waiting for resource configuration
salloc: Nodes n2017 are ready for job
[torbenr@n2017]$ nvidia-smi
Tue Nov 15 10:56:00 2022
+-----+
| NVIDIA-SMI 515.65.01      Driver Version: 515.65.01      CUDA Version: 11.7      |
+-----+-----+-----+
| GPU  Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                               |                  |              MIG M. |
+-----+-----+-----+
|    0  Tesla V100-SXM2...  On          | 00000000:61:00.0 Off  |                0     |
| N/A   39C    P0     41W / 300W |  0MiB / 32768MiB |    0%      Default  |
|                               |                  |              N/A    |
+-----+-----+-----+

+-----+
| Processes: |
| GPU   GI    CI          PID    Type    Process name          GPU Memory |
|       ID    ID                          |            Usage     |
+-----+
| No running processes found |
+-----+
[torbenr@n2017]$
```



# ALLOCATING TWO GPUS

Using `interactive`:

```
[torbenr@sigma]$ interactive -n 1 -c 18 --gpus-per-task=2 -t 60 -A LiU-gpu-2022-4 --reservation=gpu
salloc: Granted job allocation 3330818
salloc: Waiting for resource configuration
salloc: Nodes n2017 are ready for job
[torbenr@n2017]$ nvidia-smi
Tue Nov 15 12:02:15 2022
+-----+
| NVIDIA-SMI 515.65.01      Driver Version: 515.65.01      CUDA Version: 11.7      |
+-----+-----+-----+-----+-----+-----+
| GPU   Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                                           MIG M.         |
+-----+-----+-----+-----+-----+-----+
|    0   Tesla V100-SXM2...    On          | 00000000:61:00.0 Off  |
| N/A   39C    P0     41W / 300W | 0MiB / 32768MiB |      0%      Default |
|                                           N/A           |
+-----+-----+-----+-----+-----+-----+
|    1   Tesla V100-SXM2...    On          | 00000000:62:00.0 Off  |
| N/A   39C    P0     41W / 300W | 0MiB / 32768MiB |      0%      Default |
|                                           N/A           |
+-----+-----+-----+-----+-----+-----+

+-----+
| Processes:                                     |
|  GPU   GI    CI          PID    Type    Process name                        GPU Memory |
|          ID    ID                                   |          Usage |
+-----+-----+-----+-----+-----+-----+
| No running processes found |
+-----+
[torbenr@n2017]$
```

## USING SINGULARITY AND NGC

[NGC Catalog](#): Software Hub with containers with a range of GPU-accelerated software for NVIDIA GPUs

```
[torbenr@sigma]$ interactive -n 1 -c 18 --gpus-per-task=v100:2 -t 60 -A LiU-gpu-2022-4 --reservation=gpu
salloc: Granted job allocation 1512491
srun: Step created for job 1512491
[torbenr@n2017]$ . ./sourceme.txt
[torbenr@n2017]$ singularity run --nv tf20.09_py3.v3.sif

=====
== TensorFlow ==
=====

NVIDIA Release 20.09-tf2 (build 16003717)
TensorFlow Version 2.3.0

Container image Copyright (c) 2020, NVIDIA CORPORATION. All rights reserved.
Copyright 2017-2020 The TensorFlow Authors. All rights reserved.

Various files include modifications (c) NVIDIA CORPORATION. All rights reserved.
NVIDIA modifications are covered by the license terms that apply to the underlying project or file.

Detected MOFED 5.0-0.

NOTE: MOFED driver was detected, but nv_peer_mem driver was not detected.
Multi-node communication performance may be reduced.

Singularity>
```

