

GPU NODES IN TETRALITH

HARDWARE:

- 170 retrofitted thin nodes
- 96 GiB of primary memory (RAM)
- One NVIDIA Tesla T4 GPU (Turing)
- 16GB GPU memory
- One NVMe SSD scratch disk of ~2TiB

Further reading: www.nsc.liu.se/systems/tetralith/

```
[torbenr@n1112 ~]$ nvidia-smi
Tue Dec  1 21:08:59 2020
+-----+
| NVIDIA-SMI 455.32.00      Driver Version: 455.32.00      CUDA Version: 11.1      |
+-----+-----+-----+
| GPU  Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|====+=====+====+=====+
|   0   Tesla T4              On      | 00000000:3B:00.0 Off  |                0    |
| N/A   29C    P8             9W / 70W |  0MiB / 15109MiB |         0%    Default |
|                                     |                    |                    N/A |
+-----+-----+-----+
+-----+
| Processes: |
| GPU  GI   CI          PID   Type   Process name          GPU Memory |
|      ID   ID                          |              Usage   |
+-----+
| No running processes found |
+-----+
```

PRIMARYLY 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:

```
[torbenr@tetralith1 ~]$ interactive -n 1 -c 32 --gpus-per-task=1 -t 60 -A snic2020-5-235 --reservation=devel
salloc: Pending job allocation 11187331
salloc: job 11187331 queued and waiting for resources
salloc: job 11187331 has been allocated resources
salloc: Granted job allocation 11187331
srun: Step created for job 11187331
[torbenr@n99 ~]$
```

- Generally allocate a complete node with `-n 1 -c 32` or `(-N 1)`
- `--gpus-per-task=1` allocates the GPU
- Only need `-A "slurm account"` if you are included in several projects
- `--reservation=devel` is for short (max 60 min.) jobs
- For longer jobs, don't add `--reservation=devel`

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=snic2020-5-235
:
```

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/

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"`

```
[torbenr@tetralith2 ~]$ interactive.vgl -N 1 -t 60 -A snic2020-5-235 --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 11193190
salloc: job 11193190 queued and waiting for resources
salloc: job 11193190 has been allocated resources
salloc: Granted job allocation 11193190
srun: Step created for job 11193190
[torbenr@n99 ~]$
```

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(trains, 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](#).

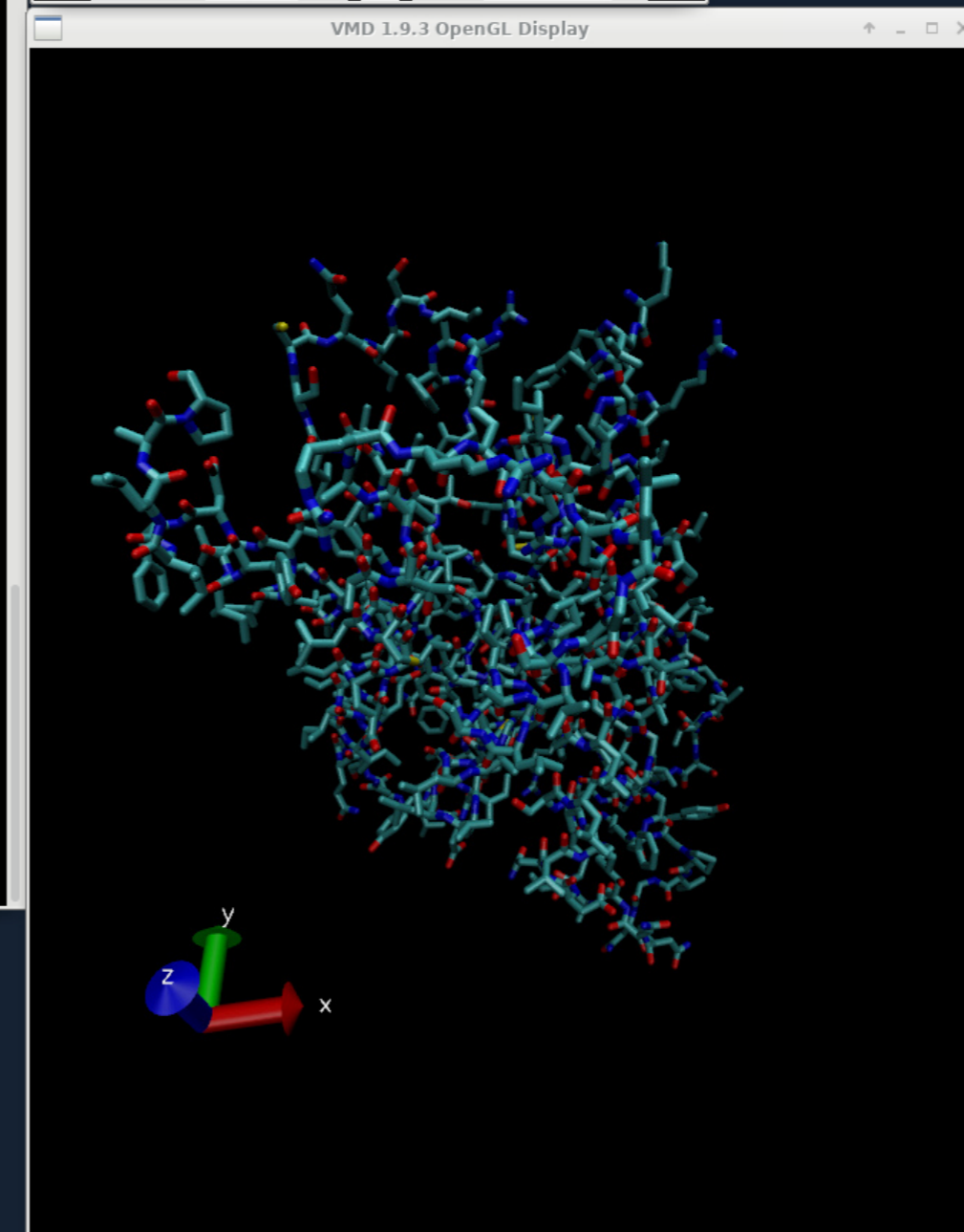
UTF-8 script Ln 83 Col 1

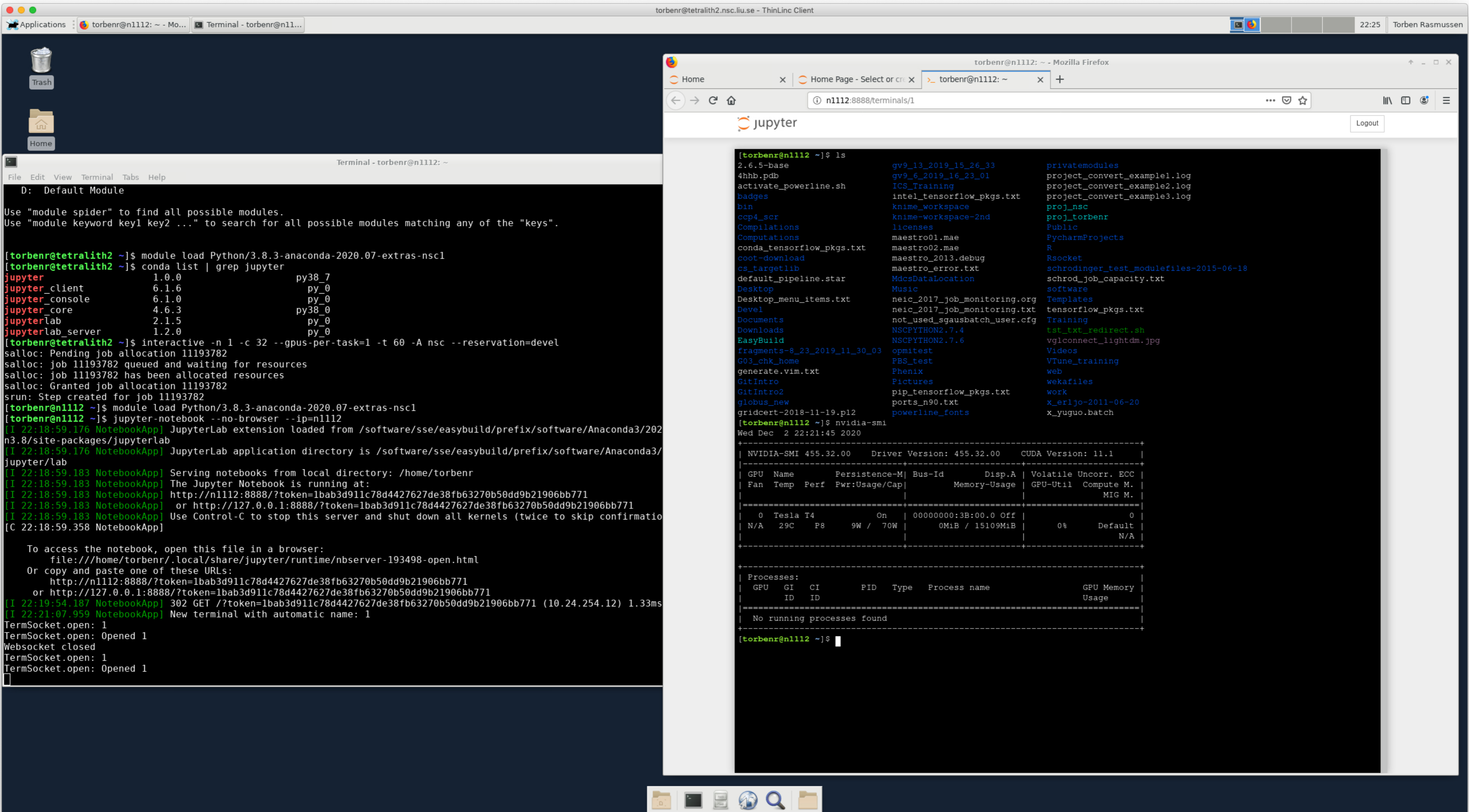

```

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.

```

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





USING SINGULARITY AND NGC

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

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)
- 32GB GPU memory pr card
- Two 7680GB NVMe SSD scratch disks (~14TiB total)

Further reading: www.nsc.liu.se/systems/sigma/

```
[torbenr@n2017 ~]$ nvidia-smi
Wed Dec  2 11:01:25 2020
+-----+
| NVIDIA-SMI 455.32.00      Driver Version: 455.32.00      CUDA Version: 11.1      |
+-----+-----+-----+
| GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|====+=====+====+=====+====+=====+=====+=====+=====+=====+=====+
|    0  Tesla V100-SXM2...  On         | 00000000:61:00.0 Off  |           0         |
| N/A   41C    P0     41W / 300W |  0MiB / 32510MiB |    0%         Default |
|                                     |                       |               N/A   |
+-----+-----+-----+
|
| Processes:
| GPU   GI    CI          PID    Type    Process name          GPU Memory
|       ID    ID                                   Usage
|=====+=====+=====+=====+=====+=====+=====+=====+=====+
| No running processes found
+-----+-----+-----+
```

- Access currently restricted to Machine learning projects
- Apply using the [LiU Local GPU 2020](#) 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=v100:1 -t 60 -A LiU-gpu-2020-4 --reservation=gpu
salloc: Granted job allocation 1133249
srun: Step created for job 1133249
[torbenr@n2017 ~]$
```

- Generally allocate 9 cores per GPU with `-c 9`, `-c 18` etc.
- `--gpus-per-task=v100:1` allocates the GPU
- You need to add `-A "slurm account"` unless you are only included in a LiU-gpu-202X-Y project
- `--reservation=gpu` is required

ALLOCATING ONE GPU

Batch script header:

```
#!/bin/bash
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=9
#SBATCH --gpus-per-task=v100:1
#SBATCH --time=24:00:00
#SBATCH --account=LiU-gpu-2020-4
#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/

ALLOCATING TWO GPUS

Using interactive:

```
[torbenr@sigma ~]$ interactive -n 1 -c 18 --gpus-per-task=v100:2 -t 60 -A LiU-gpu-2020-4 --reservation=gpu
salloc: Granted job allocation 1133252
srun: Step created for job 1133252
[torbenr@n2017 ~]$ nvidia-smi
Thu Dec  3 08:26:33 2020
+-----+
| NVIDIA-SMI 455.32.00      Driver Version: 455.32.00      CUDA Version: 11.1      |
+-----+-----+-----+
| GPU   Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|=====+=====+=====+
|    0   Tesla V100-SXM2...    On          | 00000000:61:00.0 Off  |          0          |
| N/A   39C    P0     41W / 300W |  0MiB / 32510MiB |    0%      Default  |
|                                           |                      | N/A           |
+-----+-----+-----+
|    1   Tesla V100-SXM2...    On          | 00000000:62:00.0 Off  |          0          |
| N/A   39C    P0     41W / 300W |  0MiB / 32510MiB |    0%      Default  |
|                                           |                      | N/A           |
+-----+-----+-----+

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