

Introduction to GPU/AI resources on Minerva

Minerva Scientific Computing Environment

<https://labs.icahn.mssm.edu/minervalab>

Jielin Yu, PhD

The Minerva HPC Team

February 27 2025



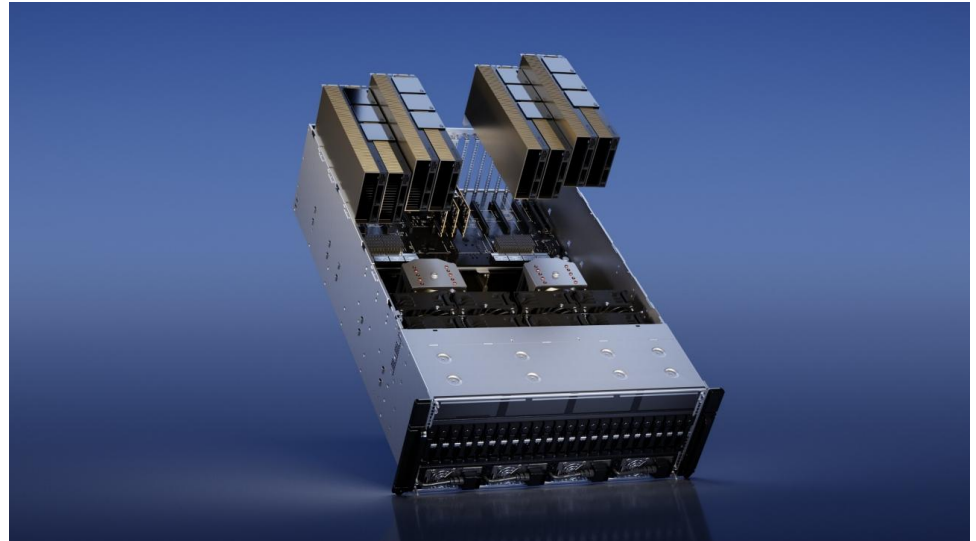
Icahn
School of
Medicine at
**Mount
Sinai**

Outline

- **What is GPU?**
- **GPU resources on Minerva**
- **User GPU Software Environment on Minerva**
- **Run GPU jobs in LSF**

What is GPU?

- A graphics processing unit (GPU) is a specialized electronic circuit initially designed to accelerate computer graphics and image processing.
- GPUs can be used across a wide range of compute-intensive applications:
 - AI/Machine Learning
 - Simulations
 - Professional visualization
 - Gaming



CPU vs GPU

CPU vs GPU		
	CPU	GPU
Function	Generalized component that handles main processing functions of a server	Specialized component that excels at parallel computing
Processing	Designed for serial instruction processing	Designed for parallel instruction processing
Design	Fewer, more powerful cores	More cores than CPUs, but less powerful than CPU cores

[reference](#)

GPU resources on Minerva

- GPU resources on Minerva
 - Current:
 - Interactive queue (1 GPU node)
 - gpu queue for batch (74 GPU nodes)
 - 316 GPUs in total

GPU model	V100	A100	A100-80GB	H100	H100-NVlink	L40S
# of nodes	12	8	2	2	47	4
GPU card/node	4	4	4	4	4	8
CPU cores	32	48	64	64	96	96
host memory	384 GB	384 GB	2 TB	512 GB	1.5 TB	1.5 TB
GPU memory	16 GB	40 GB	80 GB	80 GB	80 GB	48 GB

User GPU Software Environment - Major packages

OS: Rocky 9.4 with glibc-2.34(GNU C library) available

- Packages with GPU support:
 - Schrödinger Suite, Amber tools, NAMD, Gromacs, Alpha Fold2, etc.
- AI tools with python/3.12.5
 - CuPy, cuDF, cuML, Numba, scikit-learn, Scanpy, Squidpy, etc.
 - [Minerva Python instruction](#)
- AI tools with conda
 - MONAI, Rapids, NVFlare, tensorflow, pytorch, etc.
 - [Minerva conda instruction](#)
- AI tools with singularity
 - Holoscan, BioNeMo, Parabricks, DeepVariant, Alpha Fold3, etc.
 - [Minerva singularity instruction](#)
 - [Minerva Singularity training](#)
- Cuda toolkit versions up to 12.4.0
- Nsight Systems

User Software Environment - Anaconda Distribution

- <https://labs.icaahn.mssm.edu/minervalab/documentation/conda/>
- **To avoid incompatibilities with other python, clear your environment with module purge before loading Anaconda**

`$ml purge`

`$ml anaconda3/2024.06`

`$conda env list` # get a list of the env available (Or `$conda info --envs`)

- User should install their own envs locally, (see more guide [here](#))
 - Use option -p PATH, --prefix PATH Full path to environment location (i.e. prefix).
`$conda create python=3.x -p /sc/arion/work/gail01/conda/envs/myenv`
`$conda env create -p myenv -f environment.yml`
 - Set envs_dirs and pkgs_dirs in .condarc file, specify directories in which environments and packages are located
`$conda create -n myenv python=3.x`
- Set conda base auto-activation false
`conda config --set auto_activate_base false`

More at [Conda config guide](#)

```
$ cat ~/.condarc file
envs_dirs:
- /sc/arion/work/gail01/conda/envs
pkgs_dirs:
- /sc/arion/work/gail01/conda/pkgs
conda config --set auto_activate_base false
```

User Software Environment: Lmod

> 1000 modules, and different versions are supported on Minerva

Lmod Software Environment Module system implemented:

- Search for module: `$module avail` or `$ module spider`

Check all available R versions `$ ml spider R`

```
.....R/3.3.1, R/3.4.0-beta, R/3.4.0, R/3.4.1, R/3.4.3_p, R/3.4.3, R/3.5.0, R/3.5.1_p, R/3.5.1, R/3.5.2, R/3.5.3
```

- To check the detailed PATH setting in module files: `$ml show R`
- Load module: `$ml python` or `$module load python` or `$ml python/3.12.5` (for a specific version)
- Unload module `$ml -gcc` or `$module unload gcc`
- List loaded modules: `$ml` or `$module list`
- Purge ALL loaded modules `$ ml purge`
- Autocompletion with tab
- More at:
 - [Minerva Lmod guide](#)
 - [Lmod user guide](#)

```
gail01@li03c03: ~ $ ml python
gail01@li03c03: ~ $ ml

Currently Loaded Modules:
  1) gcc/8.3.0  2) python/3.7.3

gail01@li03c03: ~ $ ml python/2.7.16

The following have been reloaded with a version change:
  1) python/3.7.3 => python/2.7.16

gail01@li03c03: ~ $ ml -gcc
```


Ollama

- Ollama is a platform that enables users to interact with Large Language Models (LLMs) via an Application Programming Interface (API)

<https://github.com/ollama/ollama>

- It is a powerful tool for generating text, answering questions, and performing complex natural language processing tasks. It provides access to various fine-tuned LLMs.
- We provide an Ollama wrapper script that allows you to start an Ollama server on Minerva's compute node and access it from your local machine through an API endpoint:

<https://labs.ica hn.mssm.edu/minervalab/documentation/ollama/>

Interactive access to GPU resources

- Set up an interactive environment on compute nodes
- Useful for testing and debugging jobs

```
bsub -P acc_hpcstaff -q gpu -n 4 -W 2:00 -R rusage[mem=4000] -R span[hosts=1] -gpu num=1 -R v100 -XF -Is /bin/bash
```

- Interactive, gpu, gpuexpress can be specified for -q.
- **-Is**: Interactive terminal/shell
- **-XF**: X11 forwarding
- /bin/bash : the shell to use
- If GPU model flag is not specified, your job will start on the earliest available GPU nodes.
- GPU option specification:
 - V100: **-R v100**
 - A100: **-R a100**
 - A100-80G: **-R a10080g**
 - H100: **-R h10080g**
 - NVLinked H100: **-R h100nvl**
 - L40S: **-R l40s**

Minerva LSF queues with GPUs

Queue structure in Minerva	
Queue	Wall time limit
interactive (Dedicated to interactive jobs)	12 hours
gpu	6 days/144 hours
gpuexpress	15 hours

bhosts

- bhosts queue_name
 - bhosts gpu
 - bhosts gpuexpress

```
[[yuj25@li04e02 ~]$ bhosts gpu
HOST_NAME      STATUS      JL/U      MAX  NJOBS  RUN  SSUSP  USUSP  RSV
lg02e01        ok          -         96   16     16    0      0      0
lg02e02        ok          -         96    3     3     0      0      0
lg02e03        ok          -         96    8     8     0      0      0
lg02e04        ok          -         96    4     4     0      0      0
lg02e05        ok          -         96    0     0     0      0      0
lg02e06        ok          -         96   35    35    0      0      0
lg02e07        ok          -         96    0     0     0      0      0
lg02e08        ok          -         96    0     0     0      0      0
lg02e09        ok          -         96   27    27    0      0      0
lg02e10        ok          -         96    6     6     0      0      0
lg02e11        ok          -         96   50    50    0      0      0
lg02e12        ok          -         96   31    31    0      0      0
lg02e13        ok          -         96   26    26    0      0      0
lg02e14        ok          -         96    0     0     0      0      0
lg02e15        ok          -         96   30    30    0      0      0
```

Batch job submission example

```
$ cat myfirst.lsf
```

```
#!/bin/bash
#BSUB -J myfirstjob           # Job name
#BSUB -P acc_hpcstaff        # REQUIRED; To get allocation account, type "mybalance"
#BSUB -q premium            # queue; default queue is premium
#BSUB -n 1                   # number of compute cores (job slots) needed, 1 by default
#BSUB -W 6:00                # REQUIRED; walltime in HH:MM
#BSUB -R rusage[mem=4000]    # 4000 MB of memory request per "-n"; 3000 MB by default
#BSUB -oo %J.stdout          # output log (%J : JobID)
#BSUB -eo %J.stderr          # error log
#BSUB -L /bin/bash           # Initialize the execution environment

ml gcc                       # Commands that you need to run
cd /sc/arion/work/MyID/my/job/dir/
../mybin/serial_executable < testdata.inp > results.log
```

```
$ bsub < myfirst.lsf
```

```
Job <87426883> is submitted to queue <premium>.
```

GPGPU - batch jobs

```
#BSUB -q gpu  
#BSUB -n Ncpu
```

```
#BSUB -gpu num=4  
#BSUB -R a100  
#BSUB -R span[hosts=1]
```

```
module purge  
module load anaconda3 ( or 2)  
module load cuda  
source activate tfGPU
```

```
python -c "import tensorflow as tf"
```

```
# submit to gpu queue  
# Ncpu is 1~48 on A100
```

```
# request 4 GPUs per node on A100 node  
#  
# request all gpu cards on the same node
```

```
# to access tensorflow  
# to access the drivers and supporting  
subroutines
```

GPGPU - batch jobs (continue)

- LSF will set `CUDA_VISIBLE_DEVICES` to the list of GPU cards assigned to the job.
E.g: 2,1,3 Most standard packages honor these assignments
 - **DO NOT MANUALLY CHANGE THE VALUE OF `CUDA_VISIBLE_DEVICES`.**
- Multiple GPU cards can be requested across different GPU nodes

```
#BSUB -q gpu                # submit to gpu queue
#BSUB -n 8                  # 8 compute cores requested
#BSUB -R span[ptile=2]     # 2 cores per node, so 4 nodes in total requested
#BSUB -R v100              # request specified gpu node v100, change to a100,
#BSUB -gpu num=2           # 2 GPUs requested per node
```

Note that 2 GPU cards will be reserved on each of 4 nodes for your job. If your job cannot /does not run in distributed mode, you will still lock these resources on the nodes that you are not using and prevent others from being dispatched to those node.

`CUDA_VISIBLE_DEVICES` may be defined differently on each of the nodes allocated to your job.

GPGPU - Local SSD

- Make your own directory under **/ssd** and direct your temporary files there.
- Clean up your temporary files after completion.

A100	1.8 TB SATA SSD
A100-80GB	7.0 TB NVMe PCIe SSD
H100	3.84 TB NVMe PCIe SSD
L40S	3.84 TB NVMe PCIe SSD

```
#BSUB -q gpu
#BSUB -gpu num=2
#BSUB -R a100
#BSUB -R span[hosts=1]
#BSUB -E "mkdir /ssd/YourID_$(LSB_JOBID)"
#BSUB -Ep "rm -rf /ssd/YourID_$(LSB_JOBID)"
```


nvidia-smi

- A monitoring and management command line utility, nvidia-smi
- Only available on nodes with GPUs

NVIDIA-SMI 525.105.17 Driver Version: 525.105.17 CUDA Version: 12.0							
GPU	Name	Persistence-M	Bus-Id	Disp.A	Memory-Usage	Volatile	Uncorr. ECC
Fan	Temp	Perf	Pwr:Usage/Cap			GPU-Util	Compute M. MIG M.
0	NVIDIA A100-PCI...	On	00000000:06:00.0	Off	537MiB / 40960MiB	80%	E. Process Disabled
N/A	74C P0	242W / 250W					
1	NVIDIA A100-PCI...	On	00000000:2F:00.0	Off	747MiB / 40960MiB	45%	E. Process Disabled
N/A	49C P0	126W / 250W					
2	NVIDIA A100-PCI...	On	00000000:86:00.0	Off	425MiB / 40960MiB	0%	E. Process Disabled
N/A	28C P0	32W / 250W					
3	NVIDIA A100-PCI...	On	00000000:D8:00.0	Off	7261MiB / 40960MiB	0%	E. Process Disabled
N/A	28C P0	33W / 250W					

Processes:							
GPU	GI	CI	PID	Type	Process name	GPU Memory	Usage
	ID	ID					
0	N/A	N/A	13439	C	...macs/bin.AVX2_256/gmx_mpi	534MiB	
1	N/A	N/A	113104	C	...ill/progs/bin/gmx_24_tmpi	744MiB	
2	N/A	N/A	13439	C	...macs/bin.AVX2_256/gmx_mpi	422MiB	
3	N/A	N/A	115131	C	python	7258MiB	

Minerva Ticket Submission

- Send an email to: hpchelp@hpc.mssm.edu to start a ticket.
- Information to include:
 - The error message or the location of the log file on Minerva.
 - use command “pwd” to check the current working directory.
 - All the commands you used to get the error message.
 - The location of the scripts used and how did you run the script.
 - Which node.
 - yuj25@li04e04
 - The job ID and location of the job script.
 - Job <123456789> is submitted to queue <premium>.
- Please do not include any “<” in the email.

Friendly Reminder

- Never run jobs on login nodes
 - For file management, coding, compilation, etc., purposes only
- Never run jobs outside LSF
 - Fair sharing
 - Scratch disk not backed up, efficient use of limited resources
 - Job temporary dir configured to /local/JOBS instead of /tmp.
- Logging onto compute nodes is no longer allowed
- **WE DO NOT BACKUP USER FILES. PLEASE ARCHIVE/BACKUP YOUR IMPORTANT FILES.**
- Follow us by visiting <https://labs.ica hn.mssm.edu/minervalab>
- Acknowledge Scientific Computing at Mount Sinai and NIH funding in your publications with the template
<https://labs.ica hn.mssm.edu/minervalab/policies/acknowledge-scientific-computing-at-mount-sinai/>

Acknowledgements

- ▶ Supported by the Clinical and Translational Science Awards (CTSA) grant UL1TR004419 from the National Center for Advancing Translational Sciences, National Institutes of Health.

CTSA Clinical & Translational[®]
Science Awards

Last but not Least

- ▶ Got a problem? Need a program installed? Send an email to:

hpchelp@hpc.mssm.edu

Thank you!