Introduction to GPU/AI Resources on Minerva

Minerva Scientific Computing Environment

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

Yiyuan Liu, PhD
The Minerva HPC Team

March 5, 2026

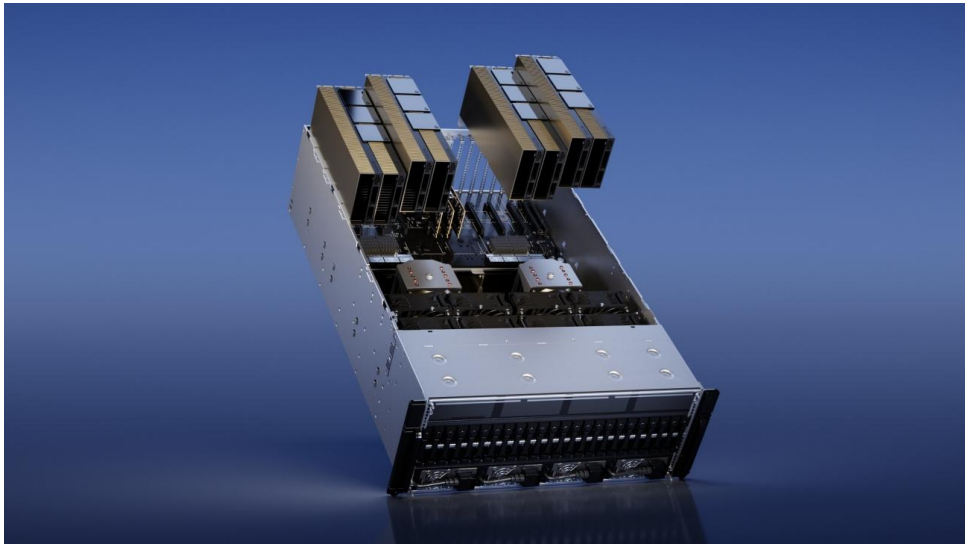
Icahn
School of
Medicine at
Mount
Sinai

Outline

- What Is GPU?
- CPU vs GPU
- GPU Resources on Minerva
- Choosing the Right GPU on Minerva
- User GPU Software Environment on Minerva
- Run GPU Jobs in LSF

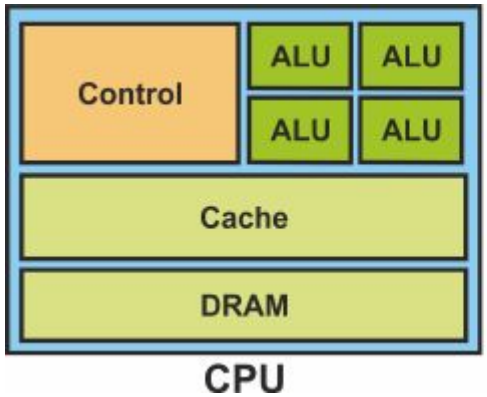
What Is GPU?

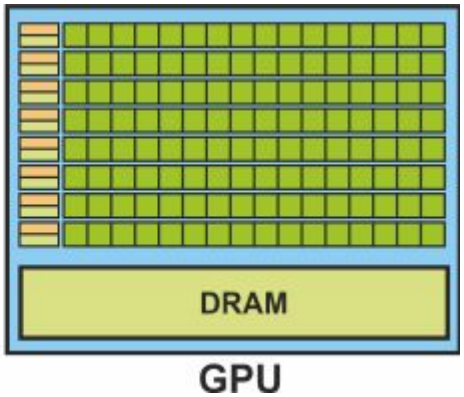
- A graphics processing unit (GPU) is a specialized electronic circuit initially designed to accelerate computer graphics and image processing.
- GPUs can be used across a wide range of compute-intensive applications:
  - AI/Machine Learning
  - Simulations
  - Professional visualization
  - Gaming



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, low latency	More cores than CPUs, but less powerful than CPU cores, high throughput





reference

DOI:10.1117/12.892282

GPU Resources on Minerva

- GPU resources on Minerva
  - Current:
    - Interactive queue (1 GPU node)
    - gpu and gpuexpress queues for batch and interactive jobs (80 GPU nodes)
    - 364 GPUs in total

GPU model	V100	A100	A100-80GB	H100	H100-NVlink	L40S	B200
# of nodes	12	8	2	2	47	4	6
GPU card/node	4	4	4	4	4	8	8
CPU cores/node	32	48	64	64	96	96	112
Host memory/node	384 GB	384 GB	2 TB	512 GB	1.5 TB	1.5 TB	2 TB
GPU memory	16 GB	40 GB	80 GB	80 GB	80 GB	48 GB	192 GB
NVLink	❌	❌	✅	❌	✅	❌	NVSwitch

Choosing the GPU on Minerva

- The B200 model introduces new format FP4 (4-bit floating point), enabling nearly an exaflop (one quintillion, i.e. 1,000,000,000,000,000,000, floating-point operations performed per second). Best suited for AI research.
- For smaller jobs (1–2 GPUs or light GPU memory), H100 or A100 GPUs are usually sufficient.
- Choose resources based on the model size, the dataset size, GPU memory requirements.
- Usage of GPU nodes is monitored, and job adjustments will be suggested if necessary.

GPU model	Use Case
V100	Early GPU. Slower than newer GPUs.
A100	Much faster than V100. Widely used for deep learning.
A100-80GB	Same GPU as A100 but NVlinked with double memory, good for bigger models.
H100	Transformer Engine (optimized for LLMs). Much faster than A100.
H100-NVlink	The high-end version of H100. Connects to other GPUs very fast.
L40S	Best for running inference not model training.
B200	Latest model on Minerva. Fastest of all, high GPU memory. GPUs that are interconnected via NVSwitch.

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 12.9.1

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/4.3.0, R/4.3.3, R/4.3.3-intel-mkl, R/4.4.0, R/4.4.1, R/4.4.3-intel-mkl, R/4.5.1, R/4.5.2-intel-mkl
- 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

```
[liuy22@li04e01 ~]$ ml python
[liuy22@li04e01 ~]$ ml

Currently Loaded Modules:
  1) unixODBC/2.3.12   2) gcc/14.2.0   3) openBLAS/0.3.28   4) python/3.12.5



[liuy22@li04e01 ~]$ ml python/3.10.17

The following have been reloaded with a version change:
  1) python/3.12.5 => python/3.10.17

[liuy22@li04e01 ~]$ ml -python
[liuy22@li04e01 ~]$ ml
No modules loaded
```

User Software Environment - Anaconda Distribution

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

  $ml purge

  $ml anaconda3/2025.06

  $conda env list ***# get a list of the env available (Or*** *$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
```

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
  - B200: **-R** b200

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

```
[choh07@li04e03 ~]$ bhosts gpuexpress
HOST_NAME          STATUS       JL/U    MAX  NJOBS    RUN  SSUSP  USUSP    RSV
lg02e17            ok              -     64      4      4      0      0      0
lg03a02            ok              -     32      0      0      0      0      0
lg03a03            ok              -     32      1      1      0      0      0
lg03a04            ok              -     32     10     10      0      0      0
lg03a05            ok              -     32      4      4      0      0      0
lg03a06            ok              -     32      0      0      0      0      0
lg03a07            ok              -     32      5      5      0      0      0
lg03a08            ok              -     32      5      5      0      0      0
lg03a09            ok              -     32      6      6      0      0      0
lg03a10            ok              -     32      1      1      0      0      0
lg03a11            ok              -     32     10     10      0      0      0
lg03e18            ok              -     64     32     32      0      0      0
lg03e19            ok              -     64     36     36      0      0      0
lg05e01            ok              -     96     16     16      0      0      0
lg05e02            ok              -     96      4      4      0      0      0
lg05e03            ok              -     96      4      4      0      0      0
lg05e04            ok              -     96      4      4      0      0      0
lg05e05            ok              -     96      4      4      0      0      0
lg05e06            ok              -     96      4      4      0      0      0
lg05e07            ok              -     96     67     67      0      0      0
lg05e08            ok              -     96      7      7      0      0      0
lg05e09            ok              -     96     10     10      0      0      0
lg05e10            ok              -     96      4      4      0      0      0
lg05e11            ok              -     96      7      7      0      0      0
```

bhosts

- bhosts -R gpu_model

```
[choh07@li04e03 ~]$ bhosts -R v100
HOST_NAME          STATUS       JL/U    MAX  NJOBS    RUN  SSUSP  USUSP    RSV
lg03a01            ok              -     32     12     12      0      0      0
lg03a10            ok              -     32      1      1      0      0      0
lg03a02            ok              -     32      0      0      0      0      0
lg03a06            ok              -     32      0      0      0      0      0
lg03a03            ok              -     32      1      1      0      0      0
lg03a08            ok              -     32      5      5      0      0      0
lg03a05            ok              -     32      4      4      0      0      0
lg03a09            ok              -     32      6      6      0      0      0
lg03a07            ok              -     32      5      5      0      0      0
lg03a04            ok              -     32     10     10      0      0      0
lg03a11            ok              -     32     10     10      0      0      0
```

Batch CPU 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                          # submit to gpu queue
#BSUB -n Ncpu                         # number of CPU cores (1~48 on A100)

#BSUB -gpu num=4                      # request 4 GPUs per node on A100 node
#BSUB -R a100                         #
#BSUB -R span[hosts=1]                # request all gpu cards on the same node

module purge
module load anaconda3 ( or 2)         # to access tensorflow
module load cuda                      # to access the drivers and supporting
source activate tfGPU                 subroutines

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

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 h100nvl                # request specified gpu node h100nvl
#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

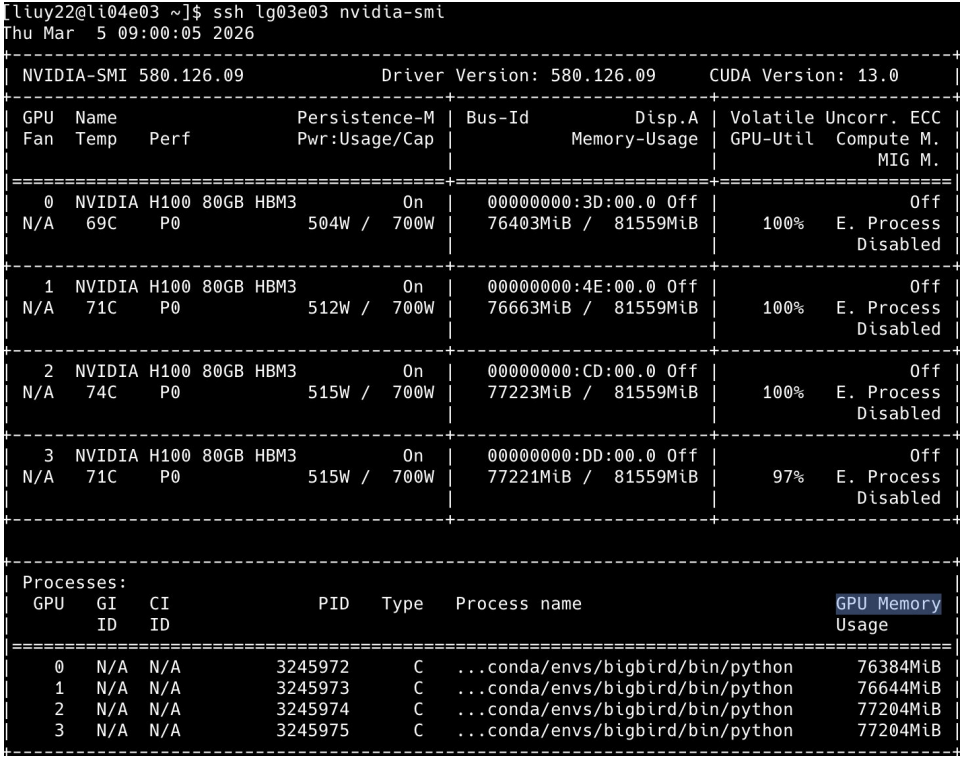
- Make your own directory under /ssd (or /local_nvme on B200) for your applications and direct your temporary files there.
- *Clean up your temporary files after completion.*

GPU Model	Total SSD	Available Under /ssd (or /local_nvme on B200)
A100	1.8 TB SATA SSD	715 GB
A100-80GB	7.0 TB NVMe PCIe SSD	358 GB
H100	3.84 TB NVMe PCIe SSD	1.4 TB
L40S	3.84 TB NVMe PCIe SSD	1.4 TB
B200	25 TB NVMe PCIe SSD	3.0 TB

```
#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 during job execution



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
- **WE DO NOT BACKUP USER FILES. PLEASE ARCHIVE YOUR IMPORTANT FILES.**
- Follow us by visiting https://labs.icahn.mssm.edu/minervalab
- Acknowledge Scientific Computing at Mount Sinai and NIH funding in your publications with the template https://labs.icahn.mssm.edu/minervalab/policies/acknowledge-scientific-computing-at-mount-sinai/

Important: Onboarding Minerva's New Multi-Factor Authentication by March 12 2026

- Digital and Technology Partners (DTP) is phasing out the Symantec VIP.
- Minerva will switch to Microsoft Authenticator as the new MFA (multi-factor authentication).
- All Minerva users are required to enroll with the new method by **March 12, 2026**.
- User announcement emails sent on 02/17/26 and 02/20/26. Instructions are also available on the Minerva login webpage. (https://labs.icahn.mssm.edu/minervalab/documentation/logging-in/)
  - Install and register Microsoft Authenticator on your phone for Microsoft MFA with your **school network account**.
  - **Make your default sign-in method to Microsoft Authenticator**.
  - Through campus network or VPN, SSH login to the Minerva test server **mfa-test.hpc.mssm.edu** with your Sinai **school password only**.
  - Wait for Microsoft Authenticator push notification dialogue on your phone. Select **Approve** to complete login.

Acknowledgements

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

Last but not Least

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

hpchelp@hpc.mssm.edu

Thank you!