Minerva GPU (graphics processing unit)

Minerva Scientific Computing Environment

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

Patricia Kovatch
Lili Gai, PhD

Hyung Min Cho, PhD
Jielin Yu, PhD
Wei Guo, PhD
Kali Mclennan
Yiyuan Liu, PhD
Eric Rosenberg
April 10th, 2024

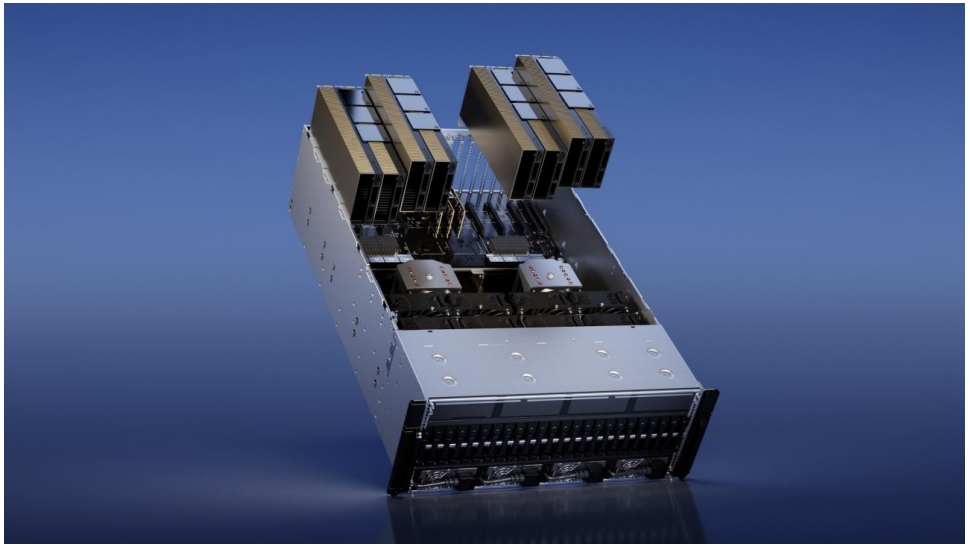
Icahn
School of
Medicine at
Mount
Sinai

Outline

- What is GPU?
- GPU resources on Minerva
- User GPU Software Environment on Minerva
- Run GPU jobs in LSF
- Reminder and future training sessions

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 (23 GPU nodes)
  - By Jan. 2025:
    - 51 additional nodes with 220 GPUs in total.

	Current				Jan. 2025	
	V100	A100	A100-80GB	H100	H100-NVlink	L40S
# of nodes	12	8	2	2	47	4
GPU card/node	4 V100	4 A100	4 A100	4 H100	4 H100	8 L40S
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: **Centos 7.9** with **glibc-2.17(GNU C library) available**

- Packages with GPU support:
  - Schrödinger Suite, Amber tools, NAMD, Gromacs, Alpha Fold2, etc.
- AI tools with python/3.7.3
  - 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, etc.
  - Minerva singularity instruction
  - Minerva Singularity training
- Cuda toolkit versions up to 12.1.1
- Nsight Systems

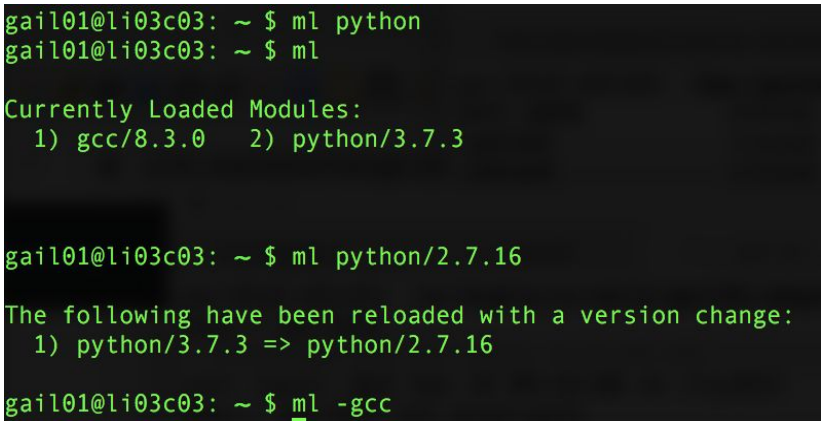
User Software Environment: Lmod

> 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/2.7.16** (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



Interactive access to GPU resources

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

```
bsub -P acc_hpcstaff -q interactive -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
- GPU option specification:
  - V100: **-gpu num**=Ngpus **-R** v100
  - A100: **-gpu num**=Ngpus **-R** a100
  - A100-80G: **-gpu num**=Ngpus **-R** a10080g
  - H100: **-gpu num**=Ngpus **-R** h10080g

Minerva LSF queues with GPUs

Queue structure in Minerva		
Queue	**Wall time limit**	**available resources**
interactive (Dedicated to interactive jobs)	12 hours	4 nodes+1 V100 GPU nodes
gpu	6 days	40 V100*, 16 A100, 4 A100-80GB, 4 H100-80GB
gpuexpress	15 hours	40 V100*, 16 A100, 4 A100-80GB, 4 H100-80GB

bhosts

- bhosts queue_name
  - bhosts gpu
  - bhosts gpuexpress

```
[yuj25@li03c03 ~]$ bhosts gpu
HOST_NAME          STATUS       JL/U    MAX  NJOBS    RUN  SSUSP  USUSP    RSV
dg-p100            ok              -     20     19      8      0      0     11
lg03a03            ok              -     32      4      4      0      0      0
lg03a04            ok              -     32      5      5      0      0      0
lg03a07            ok              -     32      4      4      0      0      0
lg03a08            ok              -     32      8      8      0      0      0
lg03a10            ok              -     32      8      8      0      0      0
lg03a11            ok              -     32      8      8      0      0      0
lg03a12            ok              -     32      0      0      0      0      0
lg06g28            ok              -     64     13     13      0      0      0
lg07c05            ok              -     48     37     37      0      0      0
lg07c06            ok              -     48     29     29      0      0      0
lg07c07            ok              -     48     17     17      0      0      0
lg07c08            ok              -     48     11     11      0      0      0
lg07c09            ok              -     64     24     24      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                                  # submit to gpu queue
#BSUB -n Ncpu                                 # Ncpu is 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 v100                   # request specified gpu node v100, change to a100, a10080g, or h10080g
#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.

Current	**A100**	**1.8 TB SATA SSD**
	A100-80GB	**7.0 TB NVMe PCIe SSD**
	H100	**3.84 TB NVMe PCIe SSD**
Jan. 2025	**L40S**	**3.84 TB NVMe PCIe SSD**
	H100	**3.84 TB NVMe PCIe SSD**

```
#BSUB -q gpu
#BSUB -gpu num=2
#BSUB -R v100
#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 | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA A100-PCI...  On   | 00000000:06:00.0 Off |                    0 |
| N/A   74C    P0   242W / 250W |    537MiB / 40960MiB |     80%   E. Process |
|                               |                      |              Disabled |
+-------------------------------+----------------------+----------------------+
|   1  NVIDIA A100-PCI...  On   | 00000000:2F:00.0 Off |                    0 |
| N/A   49C    P0   126W / 250W |    747MiB / 40960MiB |     45%   E. Process |
|                               |                      |              Disabled |
+-------------------------------+----------------------+----------------------+
|   2  NVIDIA A100-PCI...  On   | 00000000:86:00.0 Off |                    0 |
| N/A   28C    P0    32W / 250W |    425MiB / 40960MiB |      0%   E. Process |
|                               |                      |              Disabled |
+-------------------------------+----------------------+----------------------+
|   3  NVIDIA A100-PCI...  On   | 00000000:D8:00.0 Off |                    0 |
| N/A   28C    P0    33W / 250W |   7261MiB / 40960MiB |      0%   E. Process |
|                               |                      |              Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    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@li03c04
  - 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
- 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/

Minerva Seven Training Sessions Spring 2024

Seven training sessions in person/Zoom this Spring with more info here

Session	Topic	Date	Time
1	Minerva Intro	27-Mar	1PM
2	LSF Job Scheduler	3-Apr	1PM
3	**Intro to GPU resources on Minerva**	**10-April**	**1PM**
4	5 Ways to Get Started with GPUs*	12-Apr	1PM
5	Accelerated general data science in medicine with RAPIDS, CuPy and Numba*	17-Apr	1PM
6	Data Ark	24-Apr	1PM
7	How to Accelerate Genome Analysis Toolkit (GATK) by using Parabricks*	1-May	1PM

Last but not Least

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

hpchelp@hpc.mssm.edu

Acknowledgements

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

Thank you!