AI Resources on Sapelo2 Cluster

Georgia Advanced Computing Resource Center (GACRC)

Enterprise Information Technology Services(EITS)

The University of Georgia

Outline

- GACRC
- Software and Frameworks for ML/DL
- Hardware for ML/DL

Georgia Advanced Computing Resource Center (GACRC)

- A high-performance-computing (HPC) center at the UGA
- Provide to the UGA research and education community an advanced computing environment:
  - HPC computing and networking infrastructure located at the Boyd Data Center
  - Comprehensive collection of scientific, engineering and business applications
  - Consulting and training services
- Wiki: http://wiki.gacrc.uga.edu

  Support: http://help.gacrc.uga.edu

  Web Site: http://gacrc.uga.edu

Popular Programming Languages for ML/DL

Python

The most popular programming language for ML/DL due to its simple syntax, vast libraries, and active community. Its wide range of libraries makes it an ideal choice for ML/DL.

R

A statistical programming language widely used for data analysis, visualization, and ML. It offers powerful packages for statistical modeling, data manipulation, and ML algorithms.

Java

A choice for building large-scale ML applications due to its performance, scalability, and robust libraries. It offers ML/DL frameworks like Apache Spark and Deeplearning4j.

C++

A high-performance language suitable for developing computationally intensive ML algorithms. It provides control over memory management, making it efficient for resource-intensive tasks.

Python and its Role in ML/DL

Why Python?

- Readability
- Comprehensive libraries
- Flexibility with dynamic typing and on-the-fly modification
- Code clarity to enhance developer productivity

Essential Libraries

- NumPy: Numerical computation
- Pandas: Data manipulation and analysis
- Matplotlib: Data visualization
- Scikit-learn: ML algorithms
- TensorFlow and PyTorch: DL frameworks

Ecosystem Benefits

- Vast ecosystem of libraries and tools provides everything needed for building, training, and deploying ML/DL models.
- It facilitates collaboration and sharing of knowledge within the community.

Overview of Popular ML/DL Frameworks

TensorFlow

- Developed by Google
- Popular open-source framework
- Excels in large-scale ML/DL applications
- Efficient parallel processing on CPUs, GPUs, and TPUs

PyTorch

- Developed by Facebook
- Widely used for research and development
- Offere flexibility and a Pythonic interface
- Popular for NL processing
- Known for ease of use

Keras

- High-level neural network API on top of TensorFlow and Theano
- Simplifies the process of building and training ML/DL models, making it user-friendly for beginners

Scikit-Learn

- Widely used ML library for traditional ML algorithms
- Comprehensive suite of tools for classification, regression, clustering, and dimensionality reduction
- Suitable for beginners and experienced users

Overview of Popular ML/DL Frameworks

XGBoost

- Widely used, highly efficient and optimized ML library for building predictive models
- Sequentially build of an ensemble of decision trees (Gradient Boosting Decision Trees (GBDT))

LightGBM

- Newer implementation of GBDT, designed with a focus on efficiency and scalability
- Well-suited for large datasets with millions of rows and a high number of features where faster training time is essential

NLTK

- Popular Python library for natural language processing (NLP)

Transformers

- Foundational model architecture in NLP
- High effectiveness in handling tasks like text classification, machine translation, and question answering
- Popular transformer-based models: BERT and GPT

ML/DL Frameworks on Sapelo2

TensorFlow with Keras	v2.11.0	: CPU and GPU with CUDA 11.7.0, Keras 2.11.0
	v2.8.4	: CPU and GPU with CUDA 11.4.1, Keras 2.8.0
	v2.13.0	: CPU, with Keras 2.13.1
PyTorch	v2.1.2	: CPU and GPU with CUDA 12.1.1*
	v1.12.1	: CPU and GPU with CUDA 11.7.0
	v1.3.1	: CPU and GPU with CUDA 11.1.1
	v2.0.1	: CPU
Scikit-Learn	v1.2.1 v1.1.2 v1.0.1 v0.23.2	: CPU

* supports NVIDIA H100 and L4 GPUs

ML/DL Frameworks on Sapelo2

XGBoost	`v1.7.2 : CPU and GPU with CUDA 11.7.0`
LightGBM	`v4.0.0 : CPU and GPU with CUDA 11.7.0`
NLTK	`v3.8.1 : CPU`

Transformers	`v4.41.2 v4.37.0 : CPU and GPU with CUDA 12.1.1`
	`v4.34.0 : CPU and GPU with CUDA 11.7.0`

MXNet	`v1.9.1 : CPU`
Theano	`v1.1.2 : CPU and GPU with CUDA 11.7.0`

Hardware for ML/DL on Sapelo2

Sapelo2 GPU Hardware

Number of Nodes	CPU Processor	CPU cores	Host Memory (GB)	Host lscratch (TB NVMe)	NVIDIA	CUDA Capability	Devices/ node	Device Memory (GB)	CUDA Cores	Tensor Cores	FP64 | TF32 [2] (TFLOPS)
12	Intel Sapphire Rapids	64	1024	3.5	H100	9.0	4	80	16,896	528	60|1000
12	AMD Genoa	128	745	3.5	L4	8.9	4	24	7,424	232	N/A|120
14	AMD Milan	64	1024	3.5	A100	8.0	4	80	6,912	432	9.7|312
13	Various	Various	Various	Various	V100 [1]	7.0	1 or 2	16 or 32	5,120	640	7.8|N/A
2	Intel Skylake	32	192	0.87	P100	6.0	1	16	3,584	N/A	4.7|N/A

[1] Buy-in nodes [2] Sparsity enabled

Request GPU Resources for ML/DL

Sapelo2 GPU

Specifying a particular GPU model using the --gres (Generic Resource) option in Slurm can be crucial for ensuring that your job is scheduled on nodes with the desired type of GPU, which can impact performance and compatibility.

```
#SBATCH --partition=gpu_p
#SBATCH --gres=gpu:H100:1
#SBATCH --time=7-00:00:00
```

```
#SBATCH --partition=gpu_p
#SBATCH --gres=gpu:L4:4
#SBATCH --time=7-00:00:00
```

```
#SBATCH --partition=gpu_p
#SBATCH --gres=gpu:A100:2
#SBATCH --time=7-00:00:00
```

```
#SBATCH --partition=batch
#SBATCH --gres=gpu:V100:1
#SBATCH --time=4:00:00
```

The newly built central modules, which are based on CUDA 12.1.1, now support CUDA capability up to 8.9 and 9.0 (L4 and H100 GPUs). This enhancement ensures compatibility with the latest GPU hardware, allowing for more efficient use of advanced GPU features for ML and DL on the GACRC Sapelo2 cluster. Examples:

- `PyTorch/2.1.2-foss-2023a-CUDA-12.1.1`
- `GROMACS/2023.3-foss-2023a-CUDA-12.1.1-PLUMED-2.9.0`
- `GROMACS/2023.4-foss-2023a-CUDA-12.1.1`
- `magma/2.7.2-foss-2023a-CUDA-12.1.1`
- `NCCL/2.18.3-GCCcore-12.3.0-CUDA-12.1.1`

Sapelo2 software modules with CUDA 12