CUDA on HPC cluster:

What is CUDA?

Cuda is a parallel computing interface, developed by NVIDIA, for general computing on graphical processing units (GPU). It provides a software layer that can directly tap into the power of GPU, and dramatically decrease the runtime of a parallel algorithm.

Users can use most of the programming language including python to use CUDA, but it is best to learn it in its native form in C/C++.

Users can use this powerful programming interface to design fast algorithms for data science, ML, and simulations.

Additional Information on:

NVIDIA CUDA documentation

Basics of CUDA

Version Available:

- compilers/nvidia/20.7_gcc485_cuda110
- compilers/nvidia/20.7_gcc540_cuda110
- compilers/nvidia/22.3_gcc540_cuda116
- compilers/nvidia/toolkit_10.0

How to load a version of CUDA toolkit?

To load a version of CUDA on the HPC, use the following command:

```
# Select a version of cuda
module load compilers/nvidia/20.7_gcc485_cuda110
```

Verify if the module is currently loaded by using this command:

```
module list
```

If the module is loaded successfully, users can access the nvcc compiler. Test by using the following command,

```
which nvcc
```

Nvcc is a compiler which can compile CUDA codes, C as well as C++ codes.

How to use CUDA on the cluster?

After loading the module, users have access to cuda.h header file which has all the functions to access and program in the GPU pipeline.

Note:

Users can write codes without logging into the GPU nodes. However, to run the code, users must request a GPU compute node and run the compiled cuda code on the node.

GPU Access:

UA currently has 5 GPU nodes on UAHPC cluster and 3 GPU Nodes on CHPC cluster. Users can use the following command to request GPU access on an interactive bash.

```
srun -p gpu --qos gpu -n 1 -c 12 -t 100 --mem 20G  --gres gpu:1  --pty
bash
```

To use the gpu on sbatch script, use the following sbatch directive,

```
#!/bin/bash
#SBATCH -p gpu
#SBATCH -q gpu
#SBATCH -n 1
#SBATCH -c 12
#SBATCH --mem=45G
#SBATCH --gres=gpu:v100:1
```

Sample CUDA code:

Download the sample cuda code that adds two vectors of integer,

```
#Download sample cuda code
wget http://web.mit.edu/pocky/www/cudaworkshop/Matrix/VectorAdd.cu
```

Users need to compile the code using the nvcc compiler,

```
#Compile the code using nvcc compiler
nvcc VectorAdd.cu
```

Now run the code in a gpu session,

```
# Calculates the addition of 64 *64 vector
echo "64 64" | ./a.out
```

Users can use cuda alongside MPI and OpenMPI, to gain both host and device parallelism. It is highly recommended to use such powerful programming interface in research to fully utilize the potential of HPC.

Where to find help?

If you are stuck on some part or need help at any point, please contact OIT at the following address.

https://ua-app01.ua.edu/researchComputingPortal/public/oitHelp