# Getting Started on ThetaGPU

**Christopher Knight**
**Catalyst Team**

Argonne NATIONAL LABORATORY

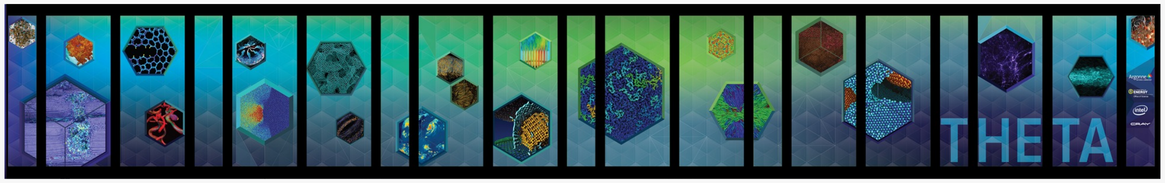
## Outline

https://www.alcf.anl.gov/user-guides

- ThetaGPU (DGX A100)
  - System Overview
  - Software & Environment Modules
  - Building your code
  - Data Science Software
  - Queuing and running jobs with qsub & mpirun

- Hands-on

## ThetaGPU



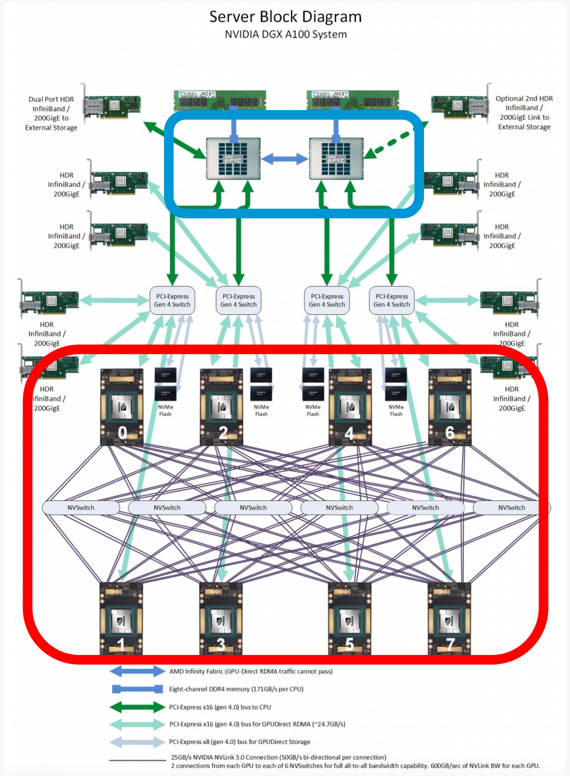
https://www.alcf.anl.gov/theta

- Theta expansion initially to support coronavirus research available for general use

- NVIDIA DGX A100 partition
  - 24 nodes each with
    - 8 NVIDIA A100 Tensor Core GPUs & 320 GB HBM memory
    - 2 AMD Rome 64-core CPUs & 1 TB DDR4
    - 15 TB SSD (4 x 3.84 TB), 25 Gb/s bandwidth
    - 8 HDR 200 NICs (compute network)
    - 2 HDR 200 NICs (storage network)

- 2 of 24 nodes have 2x memory (bigmem queue)
  - 2 TB DDR4 & 640 GB HBM

- Dedicated Compute Fabric (Mellanox in fat-tree topology)


Server Block Diagram
NVIDIA DGX A100 System
Dual Port HDR InfiniBand / 200GigE to External Storage
Optional 2nd HDR InfiniBand / 200GigE Link to External Storage
HDR InfiniBand / 200GigE
PCI-Express Gen 4 Switch
NVMe Flash
NVSwitch
0
1
2
3
4
5
6
7
AMD Infinity Fabric (GPU-Direct RDMA traffic cannot pass)
Eight-channel DDR4 memory (171GB/s per CPU)
PCI-Express x16 (gen 4.0) bus to CPU
PCI-Express x16 (gen 4.0) bus for GPUDirect RDMA (~24.7GB/s)
PCI-Express x8 (gen 4.0) bus for GPUDirect Storage
25GB/s NVIDIA NVLink 3.0 Connection (50GB/s bi-directional per connection)
2 connections from each GPU to each of 6 NVSwitches for full all-to-all bandwidth capability. 600GB/sec of NVLink BW for each GPU.

https://www.microway.com/hpc-tech-tips/dgx-a100-review-throughput-and-hardware-summary/

# ThetaGPU - Logging in and Environment

https://www.alcf.anl.gov/support-center/theta/theta-thetagpu-overview#theta-gpu

- Use Theta login nodes

```
$ ssh user@theta.alcf.anl.gov
```

- Load ThetaGPU scheduler

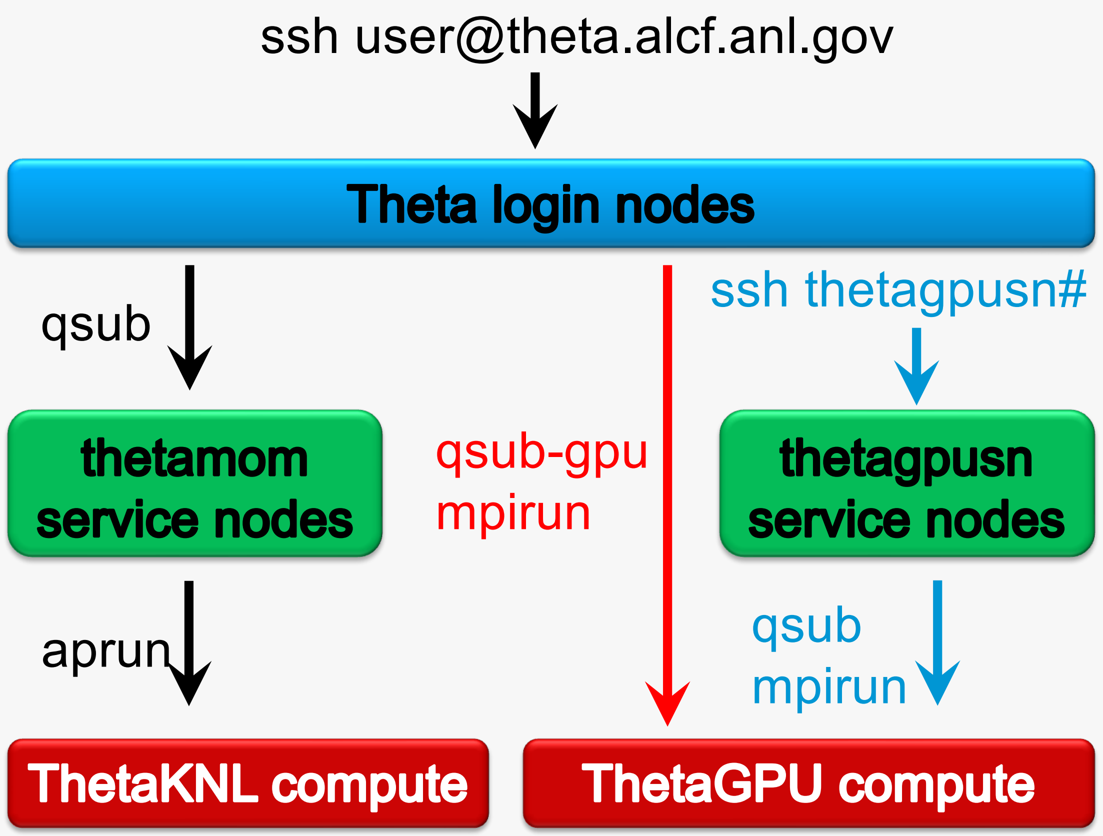
```
$ module load cobalt/cobalt-gpu
```

- Use ThetaGPU compute nodes for building and development

```
$ qsub -I -n 1 -t 60 -q full-node -A …
```

- Can also login to ThetaGPU service nodes, if needed

```
$ ssh thetagpusn1
$ qsub -I -n 1 -t 60 -q full-node -A …
```



## Theta - Modules

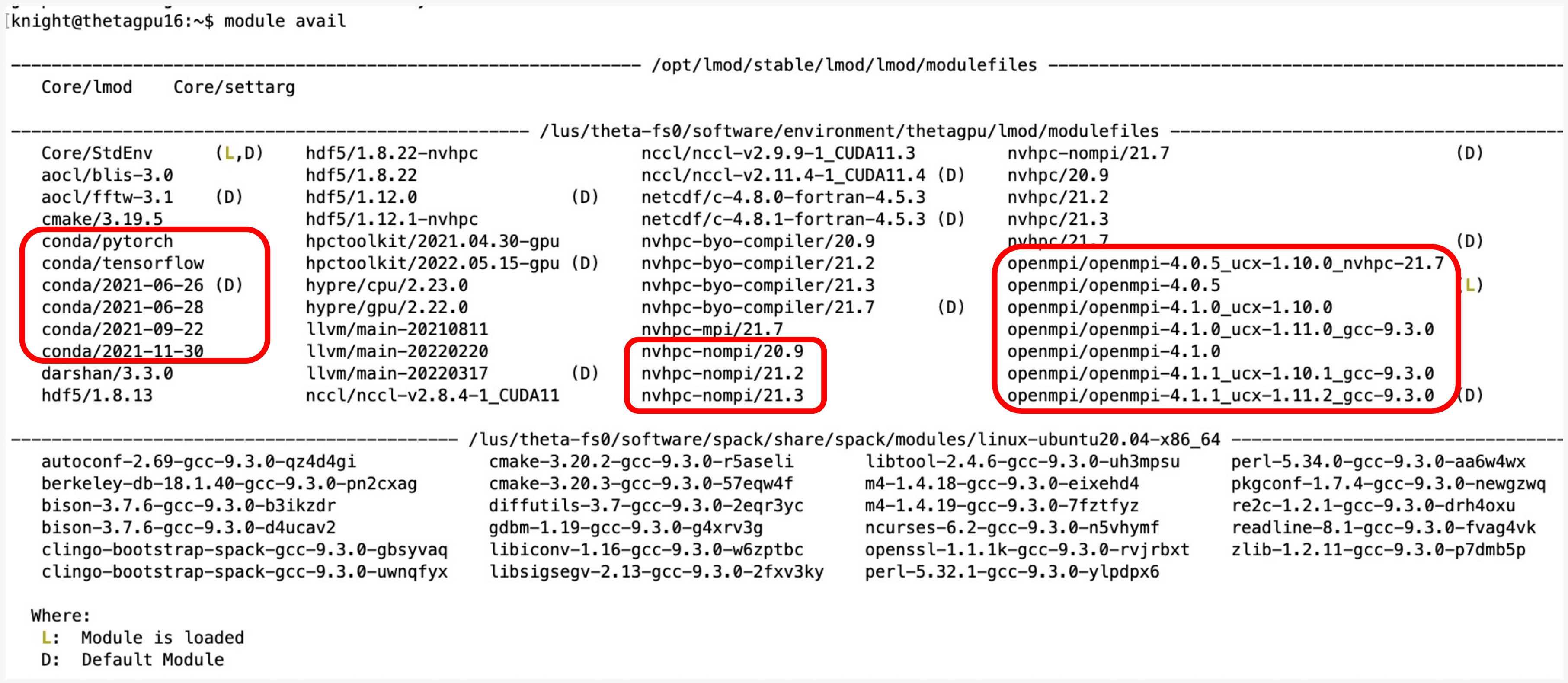
https://modules.sourceforge.net

- A tool for managing a user's environment
  - Sets your PATH to access desired front-end tools
  - Your compiler version can be changed here

- Module commands
  - List available module commands: module help
  - List currently loaded modules: module list
  - List all available modules: module avail
  - Add module to environment: module load <mod>
  - Remove module from environment: module unload <mod>
  - Swap loaded module with new one: module swap <mod_old> <mod_new>
  - List information about module: module show <mod>
  - Include additional modules: module use <path_to_extra_modules>

# ThetaGPU - Software & Libraries

https://www.alcf.anl.gov/support-center/theta-gpu-nodes

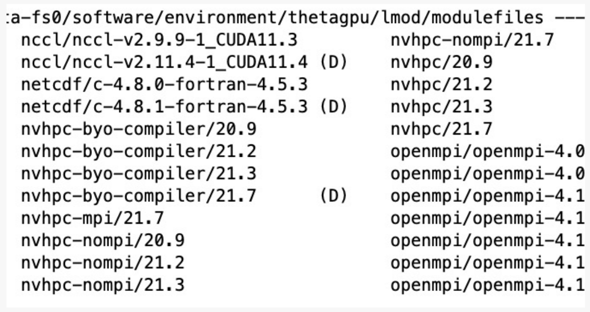


```
knight@thetagpu16:~$ module avail

------------------------------------------------------------ /opt/lmod/stable/lmod/lmod/modulefiles -------------------------------------------------
   Core/lmod    Core/settarg

---------------------------------------------------- /lus/theta-fs0/software/environment/thetagpu/lmod/modulefiles --------------------------------------
   Core/StdEnv      (L,D)    hdf5/1.8.22-nvhpc                nccl/nccl-v2.9.9-1_CUDA11.3       nvhpc-nompi/21.7                               (D)
   aocl/blis-3.0             hdf5/1.8.22                      nccl/nccl-v2.11.4-1_CUDA11.4 (D)  nvhpc/20.9
   aocl/fftw-3.1    (D)      hdf5/1.12.0               (D)    netcdf/c-4.8.0-fortran-4.5.3      nvhpc/21.2
   cmake/3.19.5              hdf5/1.12.1-nvhpc                netcdf/c-4.8.1-fortran-4.5.3 (D)  nvhpc/21.3
   conda/pytorch             hpctoolkit/2021.04.30-gpu        nvhpc-byo-compiler/20.9           nvhpc/21.7                                     (D)
   conda/tensorflow          hpctoolkit/2022.05.15-gpu (D)    nvhpc-byo-compiler/21.2           openmpi/openmpi-4.0.5_ucx-1.10.0_nvhpc-21.7
   conda/2021-06-26 (D)      hypre/cpu/2.23.0                 nvhpc-byo-compiler/21.3           openmpi/openmpi-4.0.5                          (L)
   conda/2021-06-28          hypre/gpu/2.22.0                 nvhpc-byo-compiler/21.7      (D)  openmpi/openmpi-4.1.0_ucx-1.10.0
   conda/2021-09-22          llvm/main-20210811               nvhpc-mpi/21.7                    openmpi/openmpi-4.1.0_ucx-1.11.0_gcc-9.3.0
   conda/2021-11-30          llvm/main-20220220               nvhpc-nompi/20.9                  openmpi/openmpi-4.1.0
   darshan/3.3.0             llvm/main-20220317        (D)    nvhpc-nompi/21.2                  openmpi/openmpi-4.1.1_ucx-1.10.1_gcc-9.3.0
   hdf5/1.8.13               nccl/nccl-v2.8.4-1_CUDA11        nvhpc-nompi/21.3                  openmpi/openmpi-4.1.1_ucx-1.11.2_gcc-9.3.0    (D)

------------------------------------------- /lus/theta-fs0/software/spack/share/spack/modules/linux-ubuntu20.04-x86_64 --------------------------------
   autoconf-2.69-gcc-9.3.0-qz4d4gi                cmake-3.20.2-gcc-9.3.0-r5aseli        libtool-2.4.6-gcc-9.3.0-uh3mpsu    perl-5.34.0-gcc-9.3.0-aa6w4wx
   berkeley-db-18.1.40-gcc-9.3.0-pn2cxag          cmake-3.20.3-gcc-9.3.0-57eqw4f        m4-1.4.18-gcc-9.3.0-eixehd4        pkgconf-1.7.4-gcc-9.3.0-newgzwq
   bison-3.7.6-gcc-9.3.0-b3ikzdr                  diffutils-3.7-gcc-9.3.0-2eqr3yc       m4-1.4.19-gcc-9.3.0-7fztfyz        re2c-1.2.1-gcc-9.3.0-drh4oxu
   bison-3.7.6-gcc-9.3.0-d4ucav2                  gdbm-1.19-gcc-9.3.0-g4xrv3g           ncurses-6.2-gcc-9.3.0-n5vhymf      readline-8.1-gcc-9.3.0-fvag4vk
   clingo-bootstrap-spack-gcc-9.3.0-gbsyvaq       libiconv-1.16-gcc-9.3.0-w6zptbc       openssl-1.1.1k-gcc-9.3.0-rvjrbxt   zlib-1.2.11-gcc-9.3.0-p7dmb5p
   clingo-bootstrap-spack-gcc-9.3.0-uwnqfyx       libsigsegv-2.13-gcc-9.3.0-2fxv3ky     perl-5.32.1-gcc-9.3.0-ylpdpx6

  Where:
   L:  Module is loaded
   D:  Default Module
```

# ThetaGPU - NVIDIA HPC SDK

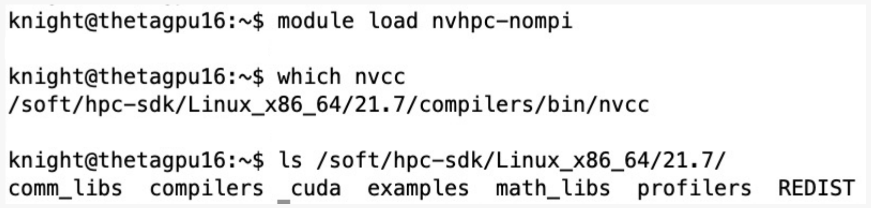
https://www.alcf.anl.gov/support-center/theta-gpu-nodes/compiling-and-linking-thetagpu

- Add NVIDIA SDK compilers, libraries, and tools to paths
  - nvhpc: adds all components to paths
  - nvhpc-byo-compiler: doesn't set compiler env. variables
  - nvhpc-nompi: excludes MPI libraries
    - Preferred module
    - Important to use ALCF-provided OpenMPI for multi-node runs



```
:a-fs0/software/environment/thetagpu/lmod/modulefiles ---
  nccl/nccl-v2.9.9-1_CUDA11.3              nvhpc-nompi/21.7
  nccl/nccl-v2.11.4-1_CUDA11.4 (D)         nvhpc/20.9
  netcdf/c-4.8.0-fortran-4.5.3             nvhpc/21.2
  netcdf/c-4.8.1-fortran-4.5.3 (D)         nvhpc/21.3
  nvhpc-byo-compiler/20.9                  nvhpc/21.7
  nvhpc-byo-compiler/21.2                  openmpi/openmpi-4.0
  nvhpc-byo-compiler/21.3                  openmpi/openmpi-4.0
  nvhpc-byo-compiler/21.7      (D)         openmpi/openmpi-4.1
  nvhpc-mpi/21.7                           openmpi/openmpi-4.1
  nvhpc-nompi/20.9                         openmpi/openmpi-4.1
  nvhpc-nompi/21.2                         openmpi/openmpi-4.1
  nvhpc-nompi/21.3                         openmpi/openmpi-4.1
```

- First time user of NVHPC SDK?
  - Commonly used libraries spread across directories
    - comm_libs: nccl, nvshmem, …
    - compilers/lib: blas, lapack, …
    - cuda/lib64: cudart, OpenCL, …
    - math_libs/lib64: cublas, cufft, …



```
knight@thetagpu16:~$ module load nvhpc-nompi

knight@thetagpu16:~$ which nvcc
/soft/hpc-sdk/Linux_x86_64/21.7/compilers/bin/nvcc

knight@thetagpu16:~$ ls /soft/hpc-sdk/Linux_x86_64/21.7/
comm_libs  compilers  cuda  examples  math_libs  profilers  REDIST
```

https://developer.nvidia.com/hpc-sdk

## ThetaGPU - Compilers

https://www.alcf.anl.gov/support-center/theta-gpu-nodes/compiling-and-linking-thetagpu

| Vendor | modules | mpiwrappers | Env. Var.* |
|---|---|---|---|
| GNU | openmpi | mpicc<br>mpicxx<br>mpif90 | OMPI_MPICC=gcc<br>OMPI_MPICXXX=g++<br>OMPI_MPIFC=gfortran |
| LLVM | llvm/main-20220317<br>openmpi | mpicc<br>mpicxx<br>mpif90 | OMPI_MPICC=clang<br>OMPI_MPICXXX=clang++<br>OMPI_MPIFC=gfortran |
| NVHPC | nvhpc-nompi<br>openmpi/openmpi-4.0.5_ucx-1.10.0_nvhpc-21.7 | mpicc<br>mpicxx<br>mpif90 | OMPI_MPICC=pgcc<br>OMPI_MPICXXX=pgc++<br>OMPI_MPIFC=pgf90 |

*Set by user when compiling applications

- NVHPC SDK
  - PGI compilers are symlinks
    - pgcc → nvc
    - pgc++ → nvc++
    - pgf90 → nvfortran

- GPU Programming Models
  - CUDA
  - OpenACC
  - OpenCL
  - OpenMP

https://developer.nvidia.com/hpc-sdk

Argonne NATIONAL LABORATORY

## ThetaGPU - Data Science

https://www.alcf.anl.gov/support-center/theta-gpu-nodes

- Documentation available for Data & Learning workflows
  - Building Python Packages
  - Singularity containers
    - Launching container with MPI
    - Converting Docker images
  - Distributed training using data parallelism
  - Running PyTorch and Tensorflow with Conda

- Many good examples available from recent SDL workshop
  - https://www.alcf.anl.gov/events/2021-alcf-simulation-data-and-learning-workshop
  - https://github.com/argonne-lcf/sdl_ai_workshop/

## ThetaGPU - qsub attributes

https://www.alcf.anl.gov/user-guides/running-jobs-xc40

- Enable Multi-Instance GPU (MIG) mode
  - --attrs mig-mode=True

- Enable public network connectivity from compute nodes
  - --attrs=pubnet

- Specify required filesystems
  - --attrs=filesystems home,grand,eagle,theta-fs0

https://docs.nvidia.com/datacenter/tesla/mig-user-guide/

## ThetaGPU - Submitting Script Jobs

https://www.alcf.anl.gov/support-center/theta-gpu-nodes/running-jobs-thetagpu

- Executable is invoked within script (bash, csh, …)

- mpirun is used to launch executables on compute nodes

```
> cat myscript.sh
  #!bin/sh
  #COBALT -n 2 -t 15 -q full-node -A <project_name>
  #COBALT --attrs pubnet
  echo “Starting Cobalt job script”
  mpirun -hostfile ${COBALT_NODEFILE} –n 16 -N 8 <app> <app_args>
```

Cobalt Options

MPI Ranks

Ranks per node

```
> qsub ./myscript.sh
  123456
```

## ThetaGPU - mpirun Overview

https://www.alcf.anl.gov/support-center/theta-gpu-nodes/running-jobs-thetagpu

- mpirun options
  - Total number of MPI ranks: –n <total_number_ranks>
  - Number of MPI ranks per node: –N <number_ranks_per_node>
  - Environment variables: –x <VAR1=1> –x <VAR2=1>
  - Display MPI process map: -display-map
  - Display detected resource allocation: -display-allocation
  - Process binding: --bind-to <hwthread|core|socket|…>
  - Process mapping: --map-by ppr:<N>:<unit>

- Environment settings you may need
  - -x OMP_NUM_THREADS=<num_threads>

- See also man mpirun

# ThetaGPU - GPU Assignment

https://www.alcf.anl.gov/support-center/theta-gpu-nodes/gpu-monitoring

- Map processes to GPUs on each node
- Programming model and framework semantics (CUDA, Tensorflow, etc…)
  - Determine local MPI rank id

```
int local_rank_id =
            atoi( getenv("OMPI_COMM_WORLD_LOCAL_RANK") );
```

  - Round-robin assignment GPUs to ranks

```
cudaGetDeviceCount(&num_devices);
cudaSetDevice(local_rank_id % num_devices);
```

```
int local_rank_id;
MPI_Comm ncomm;
MPI_Comm_split_type(MPI_COMM_WORLD,
            MPI_COMM_TYPE_SHARED, 0,
            MPI_INFO_NULL, &ncomm);
MPI_Comm_rank(ncomm, &local_rank_id);
```

- Environment variables (e.g. in helper scripts)

```
export CUDA_VISIBLE_DEVICES=0,1,2,3
```

## ThetaGPU - Queues

https://www.alcf.anl.gov/support-center/theta-gpu-nodes/queue-policy-thetagpu

- Three queues currently available with simple First-In First-Out (FIFO) policy
  - full-node: request entire node
  - bigmem: require entire node with 2x memory (2 nodes @ 2 TB CPU & 650 GB GPU)
  - single-gpu: request single gpu
    - Other node resources shared by other users
    - Analogous to debug queue to build applications and debug

| queue | full-node | bigmem | single-gpu |
|---|---|---|---|
| MinTime | 5 minutes | 5 minutes | 5 minutes |
| MaxTime | 12 hours | 12 hours | 1 hour |
| MaxQueued | 20 jobs | 2 jobs | 1 job |
| MaxRunning | 10 jobs | 1 job | 1 job |

Check website for current policies

## ThetaGPU - Profiling

https://www.alcf.anl.gov/support-center/theta-gpu-nodes/nvidia-nsight

- NVIDIA NSight Systems: system-wide profile of application

```
$ nsys profile -o <output_filename> --stats=true <app> <app_args>
$ nsys stats <output_filename>.qdrep
```

- NVIDIA NSight Compute: GPU kernel-level profiler

```
$ ncu --set detailed -o <output_filename> <app> <app_args>
$ ncu -i <output_filename>.ncu-rep
```

- Post-processing via GUI
  - Recommend downloading desktop target to view results locally
  - https://developer.nvidia.com/tools-overview

## Theta/ThetaGPU - Summary

| | **Theta** | **ThetaGPU** |
|---|---|---|
| Login | ssh user@theta.alcf.anl.gov | |
| Submit jobs from login node | qsub<br>-q default<br>-q debug-cache-quad<br>-q debug-flat-quad | module load cobalt/cobalt-gpu<br>qsub<br>-q single-gpu<br>-q full-node<br>-q bigmem |
| Compilation | On Theta login node<br>*cc, CC, ftn* | On ThetaGPU compute node<br>*mpicc, mpicxx, mpif90* |
| Launch executable | aprun | mpirun |

## ANY QUESTIONS?

## HANDS-ON SESSION

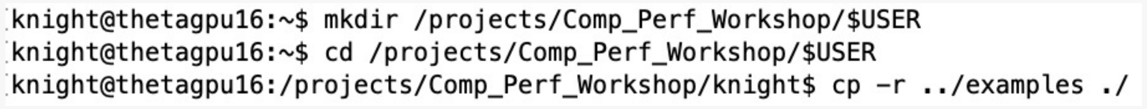
## Hands-on session

- Some examples from prior events available:
  - https://github.com/argonne-lcf/GettingStarted
  - https://github.com/argonne-lcf/sdl_ai_workshop

- GitHub repo for current workshop: https://github.com/argonne-lcf/CompPerfWorkshop

- Remember to use Workshop allocation and queue!
  - Theta: -A Comp_Perf_Workshop -q comp_perf_workshop
  - ThetaGPU:
    - -A Comp_Perf_Workshop -q single-gpu
    - -A Comp_Perf_Workshop -q full-node

- Some examples from repos available for convenience

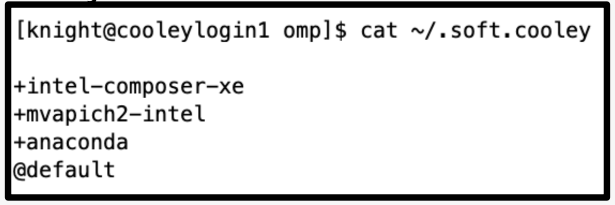


```
knight@thetagpu16:~$ mkdir /projects/Comp_Perf_Workshop/$USER
knight@thetagpu16:~$ cd /projects/Comp_Perf_Workshop/$USER
knight@thetagpu16:/projects/Comp_Perf_Workshop/knight$ cp -r ../examples ./
```

## Cooley Examples

- Example of an OpenMP job submission
  - Change to directory, compile, and submit

    ```
    $ cd /projects/Comp_Perf_Workshop/$USER/examples/cooley/omp
    $ make
    $ qsub ./submit.sh
    ```
  - Remember to edit your ~/.soft.cooley file and add compiler & MPI keys.
  - Note, @default should be the last line in your file.



```
[knight@cooleylogin1 omp]$ cat ~/.soft.cooley

+intel-composer-xe
+mvapich2-intel
+anaconda
@default
```

- Example of a Python job submission
  - Edit your ~/.soft.cooley and add “+anaconda” before @default
  - Update your environment to include python paths

    ```
    $ resoft
    ```
  - Change to directory, compile, and submit

    ```
    $ cd /projects/Comp_Perf_Workshop/$USER/examples/cooley/python
    $ qsub ./submit.sh
    ```

## Theta OpenMP Example

- Compile OpenMP example using default Intel compiler

```
$ cd /projects/Comp_Perf_Workshop/$USER/examples/theta/affinity
$ make
```

- Submit job and check output

```
$ qsub ./submit.sh
JobID
$ qstat -u $USER
$ cat <JobID>.output
```

- qsub echos a cobalt JobID to the screen. In the absence of a -o argument, three files are created (say JobID was 123456):

  123456.cobaltlog, 123456.error, 123456.output (replaced by hellompi.output with -o)

- Remember that thread affinity is controlled by aprun settings

# Theta Python Example

- Example of a Python job submission
  - Change to directory, compile, and submit

```
$ cd /projects/Comp_Perf_Workshop/$USER/examples/theta/python
$ qsub ./submit.sh
```

  - Examine submit.sh script for loading python environment on Theta

```
$ module load miniconda-3
```

  - Additional documentation here: https://www.alcf.anl.gov/user-guides/conda

## ThetaGPU MPI+OpenMP Example

- Submit interactive job from Theta login node

```
$ module load cobalt/cobalt-gpu
$ qsub -I -n 2 -t 15 -q training -A Comp_Perf_Workshop
```

- Compile using default GNU compiler on ThetaGPU compute node

```
$ cd /projects/Comp_Perf_Workshop/$USER/examples/thetagpu/affinity
$ make
```

- Launch executable across two nodes binding threads to cores

```
mpirun -n 32 -N 16 -hostfile ${COBALT_NODEFILE} -x OMP_PLACES=cores ./hello_affinity
```

## ThetaGPU MPI+OpenMP Example

- Submit job and check output

```
$ ./submit.sh
To affinity and beyond!! nname= thetagpu07  rnk= 0  tid= 0: list_cores= (0,128)
…
To affinity and beyond!! nname= thetagpu07  rnk= 15  tid= 0: list_cores= (112,240)
To affinity and beyond!! nname= thetagpu01  rnk= 16  tid= 0: list_cores= (0,128)
…
To affinity and beyond!! nname= thetagpu01  rnk= 31  tid= 0: list_cores= (112,240)
```

## ThetaGPU CUDA Compilation Example

- Submit interactive job from Theta login node

```
$ module load cobalt/cobalt-gpu
$ qsub -I -n 1 -t 15 -q training -A Comp_Perf_Workshop
```

- Compile using default GNU compiler on ThetaGPU compute node

```
$ cd /projects/Comp_Perf_Workshop/$USER/examples/thetagpu/vecadd_mpi
$ make
```

- Submit job and check output

```
$ ./submit.sh
```

- Compile using NVIDIA compiler w/ ALCF provided OpenMPI

```
$ module load nvhpc-nompi
$ make -f Makefile.nvhpc clean ; make -f Makefile.nvhpc
$ ./submit.sh
```

## ThetaGPU CUDA Fortran Compilation Example

- Submit interactive job from Theta login node

```
$ module load cobalt/cobalt-gpu
$ qsub -I -n 1 -t 15 -q training -A Comp_Perf_Workshop
```

- Compile using NVIDIA compiler w/ ALCF provided OpenMPI
  - Need matching compiler and OpenMPI library for correct mpi.mod

```
$ module load nvhpc-nompi
$ module swap openmpi openmpi/openmpi-4.0.5_ucx-1.10.0_nvhpc-21.7

$ make -f Makefile.nvhpc
$ ./submit.sh
```

# HAPPY COMPUTING!