# Getting Started on ThetaGPU

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

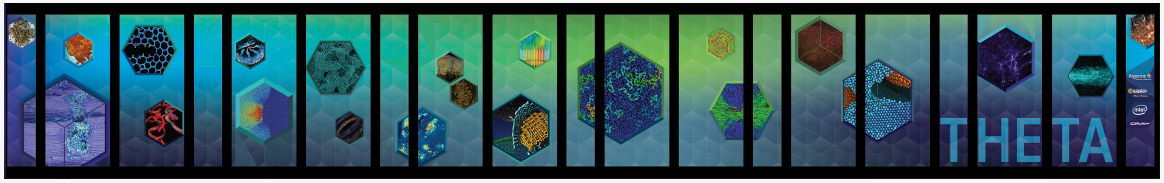
# Outline

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 to support coronavirus research that is now open 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)

- Dedicated Compute Fabric
  - 20 Mellanox QM9700 HDR200 40-port switches in fat-tree topology

- Project filesystem is Theta's 10 PB Lustre with 210 GB/s throughput

# 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

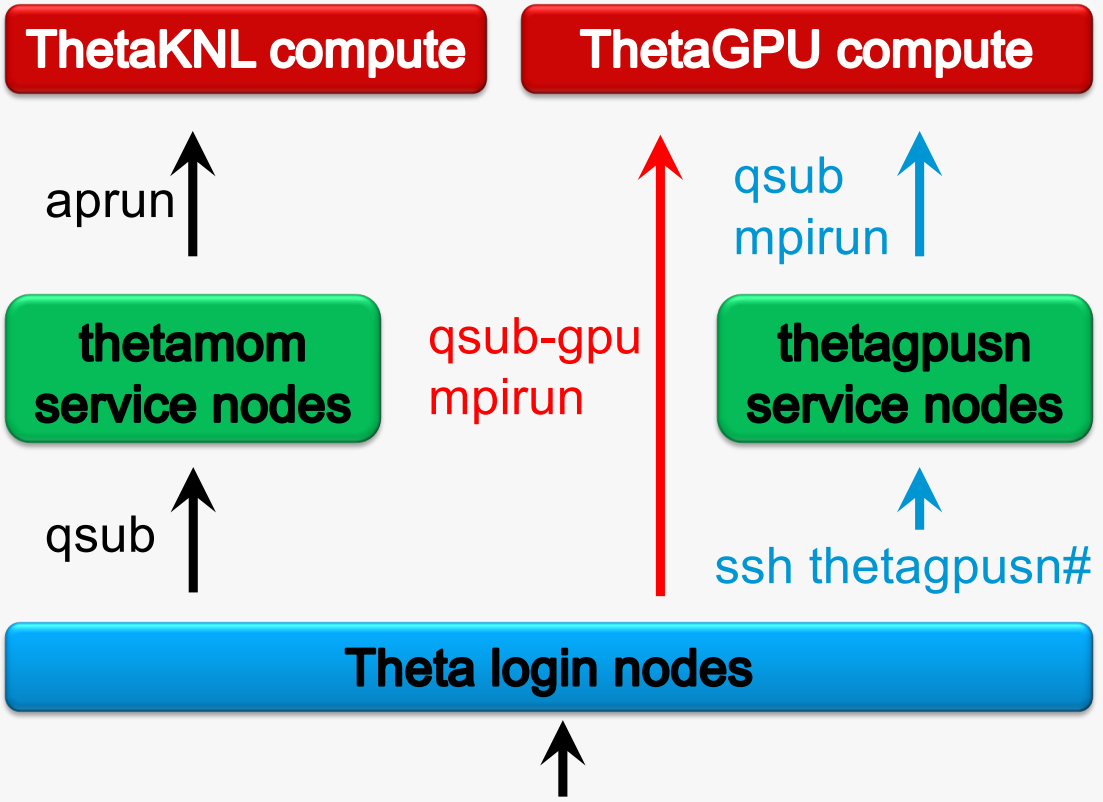
- Use ThetaGPU compute nodes for building and development

  $ 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

- List available modules on ThetaGPU compute node

  thetagpu##$ module avail

```
------------------------------------------------------------------------ /usr/local/lmod/lmod/modulefiles -------------------------------------------------------------------------
   Core/lmod    Core/settarg

------------------------------------------------------------ /lus/theta-fs0/software/environment/thetagpu/lmod/modulefiles -------------------------------------------------------------
   Core/StdEnv                 (L,D)    conda/tensorflow/2020-12-23        llvm/release-12.0.0       (D)    nvhpc-nompi/21.3
   aocl/blis-3.0                        conda/tensorflow/2021-01-08        nccl/nccl-v2.8.4-1_CUDA11        nvhpc/20.9                       (D)
   cmake/3.19.5                         conda/tensorflow/2021-03-02 (D)    nvhpc-byo-compiler/20.9   (D)    nvhpc/21.2
   conda/pytorch/2020-11-25             hdf5/1.8.13                        nvhpc-byo-compiler/21.2          nvhpc/21.3
   conda/pytorch/2021-03-02    (D)      hdf5/1.12.0                (D)     nvhpc-byo-compiler/21.3          openmpi/openmpi-4.0.5            (L)
   conda/tensorflow/2020-11-11          llvm/main-20210112                 nvhpc-nompi/20.9          (D)    openmpi/openmpi-4.1.0_ucx-1.10.0
   conda/tensorflow/2020-12-17          llvm/main-20210426                 nvhpc-nompi/21.2                 openmpi/openmpi-4.1.0            (D)

------------------------------------------------------- /lus/theta-fs0/software/spack/share/spack/modules/linux-ubuntu18.04-x86_64 --------------------------------------------------------
   autoconf-2.69-gcc-7.5.0-wmttzuv                gdbm-1.18.1-gcc-10.2.0-ia4egqb       mpfr-4.0.2-gcc-7.5.0-mpv2v7v         readline-8.0-gcc-10.2.0-ephdh34
   autoconf-archive-2019.01.06-gcc-7.5.0-bdyarrk  gdbm-1.18.1-gcc-7.5.0-4av4gyw        ncurses-6.2-gcc-10.2.0-qjpgcs6       readline-8.0-gcc-7.5.0-t54jzdy
   automake-1.16.3-gcc-7.5.0-stmktof              gmp-6.1.2-gcc-7.5.0-3ol3tld          ncurses-6.2-gcc-7.5.0-crhlefo        zlib-1.2.11-gcc-10.2.0-glt2u7u
   berkeley-db-18.1.40-gcc-10.2.0-fle5h4p         libiconv-1.16-gcc-7.5.0-jearpk4      openssl-1.1.1j-gcc-10.2.0-3g4hmwz    zlib-1.2.11-gcc-7.5.0-smoyzzo
   berkeley-db-18.1.40-gcc-7.5.0-vd7vwr5          libsigsegv-2.12-gcc-7.5.0-lbrx7ln    perl-5.32.1-gcc-10.2.0-grji3ix       zstd-1.4.5-gcc-7.5.0-rnf7xyj
   cmake-3.19.5-gcc-10.2.0-felctqr                libtool-2.4.6-gcc-7.5.0-jdxbjft      perl-5.32.1-gcc-7.5.0-op6xocu
   diffutils-3.7-gcc-7.5.0-otkkten                m4-1.4.18-gcc-7.5.0-mkc3u4x          pkgconf-1.7.3-gcc-10.2.0-4aysapw
   gcc-10.2.0-gcc-7.5.0-jj2fh4j                   mpc-1.1.0-gcc-7.5.0-pj4yncj          pkgconf-1.7.3-gcc-7.5.0-4sh6pym

  Where:
   L:  Module is loaded
   D:  Default Module

Use "module spider" to find all possible modules and extensions.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".
```

# ThetaGPU - GNU Compilers

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

- GCC w/ OpenMPI
  - Default environment
  - GPU Programming Models: CUDA, OpenCL
  - Use C/C++ wrappers: mpicxx, mpicc
  - Fortran is not supported (gfortran not installed)

# ThetaGPU - NVIDIA Compilers

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

- NVIDIA HPC SDK
  - Load module

```
$ module use /lus/theta-fs0/software/environment/thetagpu/lmod/tmp
$ module swap openmpi openmpi-4.1.0_nvhpc-21.3
$ module list
Currently Loaded Modules:
1) Core/StdEnv  2) nvhpc-nompi/21.3   3) openmpi-4.1.0_nvhpc-21.3
```

  - GPU Programming Models: CUDA, OpenCL, OpenACC, OpenMP
  - Use pgc++, pgcc, pgf90, etc…
  - Use mpicxx, mpicxx, mpif90, etc…

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

# ThetaGPU - NVIDIA Compilers

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

- NVIDIA HPC SDK modules
  - Adds NVIDIA SDK compilers, libraries, and tools to paths

```
ftware/environment/thetagpu/lmod/modulefiles ------------------------------------------
   llvm/release-12.0.0       (D)    nvhpc-nompi/21.3
   nccl/nccl-v2.8.4-1_CUDA11        nvhpc/20.9                        (D)
   nvhpc-byo-compiler/20.9   (D)    nvhpc/21.2
   nvhpc-byo-compiler/21.2          nvhpc/21.3
   nvhpc-byo-compiler/21.3          openmpi/openmpi-4.0.5             (L)
   nvhpc-nompi/20.9          (D)    openmpi/openmpi-4.1.0_ucx-1.10.0
   nvhpc-nompi/21.2                 openmpi/openmpi-4.1.0             (D)
```

- nvhpc: adds all NVIDIA SDK compilers, libraries, and tools to paths

- nvhpc-byo-compiler: identical to nvhpc, but doesn't set compiler environment variables

- nvhpc-nompi: excludes MPI libraries
  - Preferred module
  - Important to use ALCF provided OpenMPI modules for multi-node runs

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

# ThetaGPU - LLVM Compilers

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

- LLVM w/ OpenMP offload
  - Load module

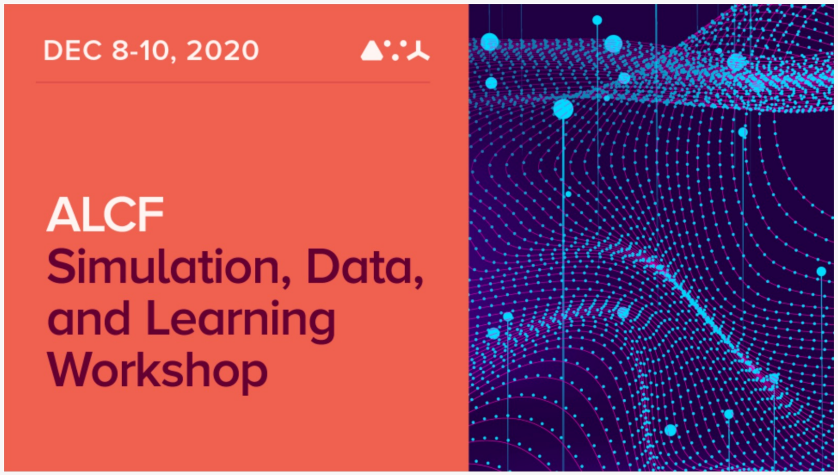
```
$ module load llvm
$ module list
Currently Loaded Modules:
1) openmpi/openmpi-4.0.5  2) Core/StdEnv  3) llvm/release-12.0.0
```

  - GPU Programming Models: CUDA, OpenCL, OpenMP
  - Use clang, clang++
  - Use mpicxx, mpicxx

# 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/2020-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

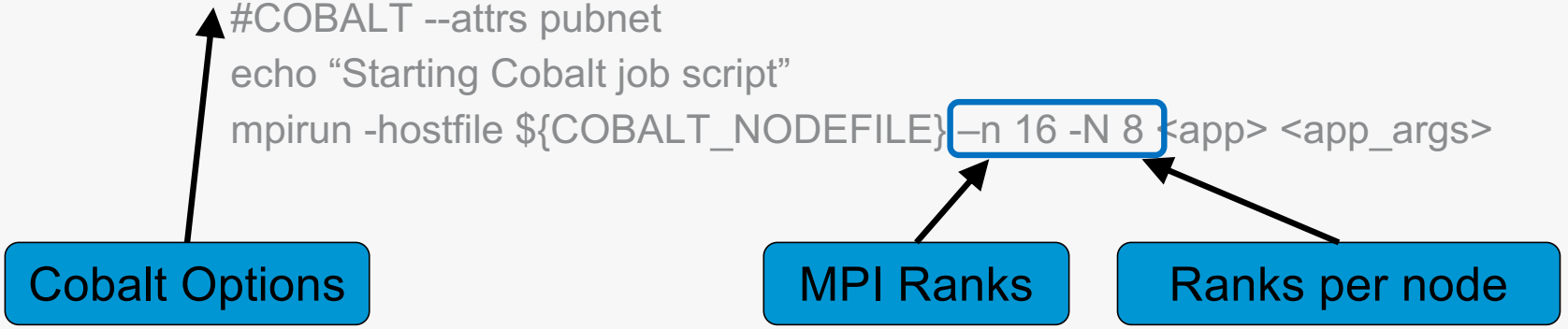
# ThetaGPU - Submitting Script Jobs

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>
```




```
> 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|…>

- 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…)

```
MPI_Comm_rank(MPI_COMM_WORLD, &me)
cudaGetDeviceCount(&num_devices);
cudaSetDevice(me % num_devices);
```

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

```
export CUDA_VISIBLE_DEVICES=4
```

# ThetaGPU - Queues

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

- Two queues currently available with simple First-In First-Out (FIFO) policy
  - full-node: request entire node
  - single-gpu: request single gpu; node resources shared by other users
    - analogous to debug queue to build applications and debug

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

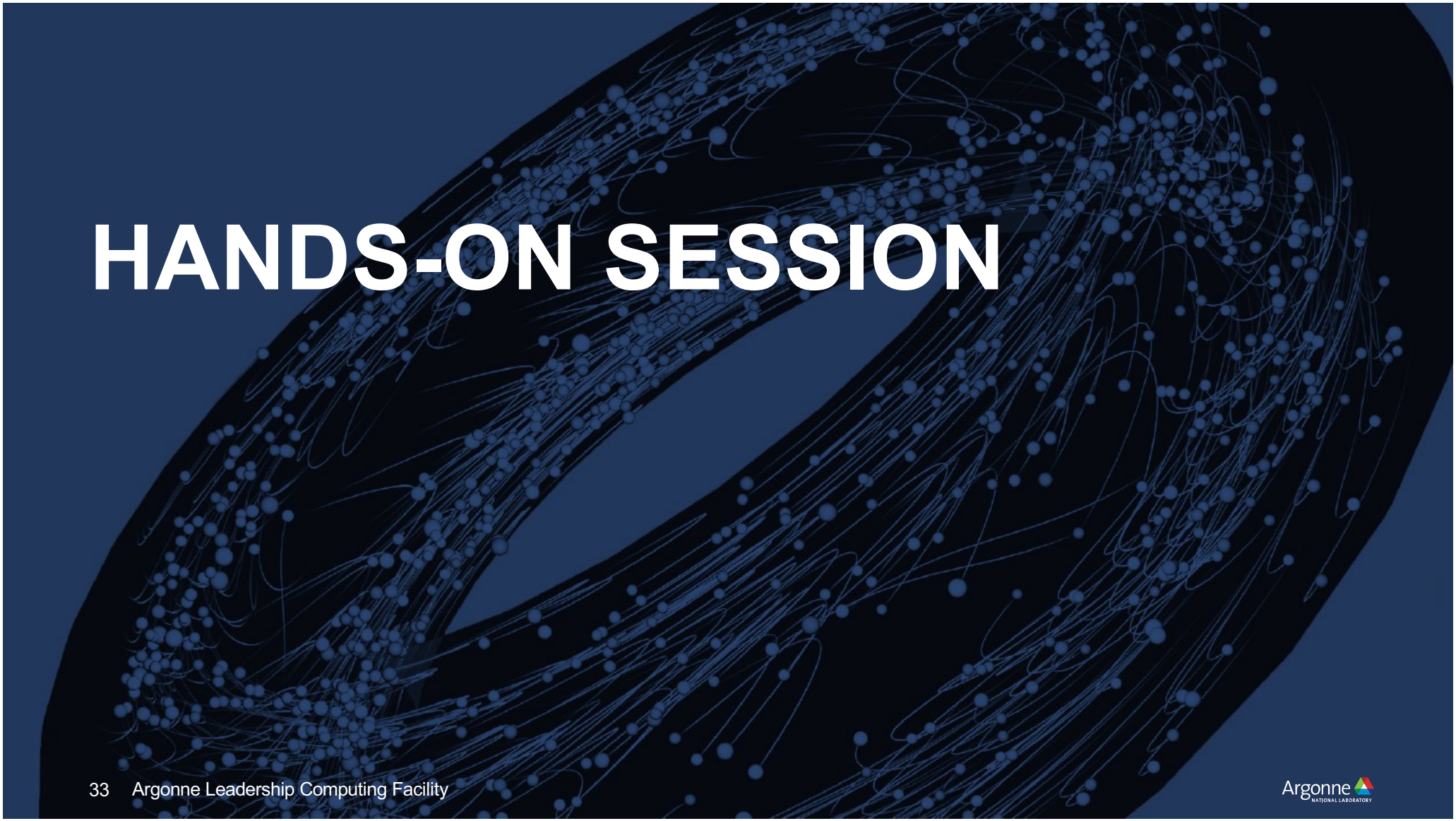
- NVIDIA NSight Compute: GPU kernel-level profiler

  $ 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


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 this workshop: https://github.com/argonne-lcf/CompPerfWorkshop-2021
- Remember to use Workshop allocation and queue!
  - Theta: -A Comp_Perf_Workshop -q comp_perf_workshop
  - ThetaGPU: -A Comp_Perf_Workshop -q training
- Copy of some examples from repos available for convenience

```
$ mkdir /projects/Comp_Perf_Workshop/$USER
$ cd /projects/Comp_Perf_Workshop/$USER
$ cp -r /projects/Comp_Perf_Workshop/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/21.3
  $ 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 use /lus/theta-fs0/software/environment/thetagpu/lmod/tmp
$ module swap openmpi openmpi-4.1.0_nvhpc-21.3
$ module list
Currently Loaded Modules:
  1) Core/StdEnv   2) nvhpc-nompi/21.3   3) openmpi-4.1.0_nvhpc-21.3

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


HAPPY COMPUTING!