Cray XK7 (Titan/Metis)
Quick Start

Node Types

- Batch/Service/Login
  - General interaction
  - Compiling
  - Batch scripts & batch-interactive sessions run here
- Compute
  - Where parallel jobs run
  - Only accessible via `aprun` command

Compiling

- Compiler is controlled by `PrgEnv-*` module
  - To change compilers, change this, not the individual compiler module
- Compilers are invoked the same way regardless of back-end compiler
  - C: `cc`
  - C++: `CC`
  - Fortran: `ftn`

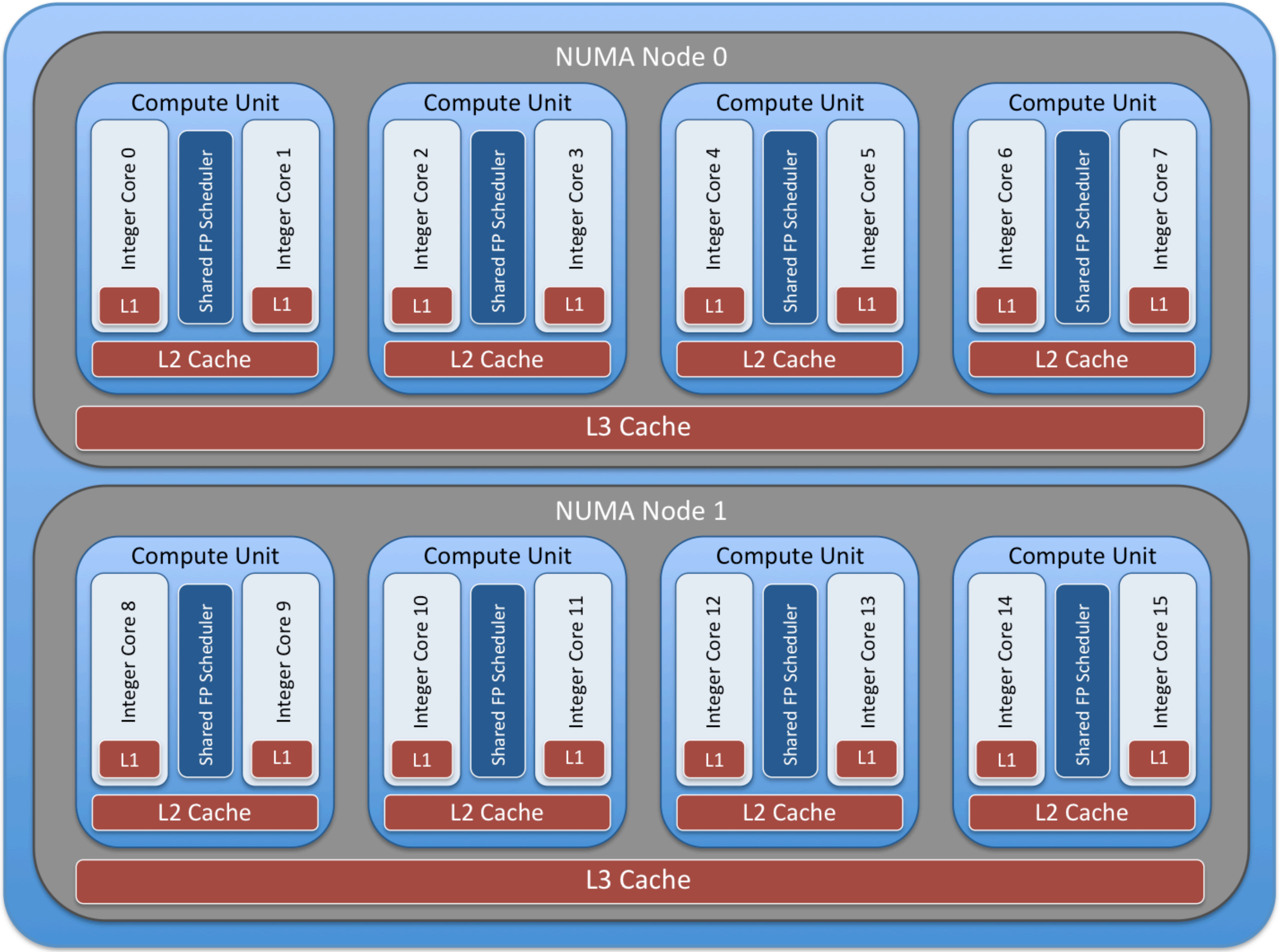
Compiling

- Many libraries (including accelerator libraries) automatically linked based on loaded modules
  - No need to add -l flags for CUDA, hdf5, mpi, etc.
  - Be sure to `module load cudatoolkit`
- You’re cross-compiling

Metis Compute Node

- AMD Opteron processor w/32GB memory
    - 16 integer cores
    - 8 FP units (pairs of integer cores share a floating point unit)
- NVIDIA Kepler K20X GPU w/6GB memory

AMD Opteron™ 6274 (Interlagos) CPU



NUMA Node 0

Compute Unit: Integer Core 0, L1 | Shared FP Scheduler | Integer Core 1, L1 | L2 Cache

Compute Unit: Integer Core 2, L1 | Shared FP Scheduler | Integer Core 3, L1 | L2 Cache

Compute Unit: Integer Core 4, L1 | Shared FP Scheduler | Integer Core 5, L1 | L2 Cache

Compute Unit: Integer Core 6, L1 | Shared FP Scheduler | Integer Core 7, L1 | L2 Cache

L3 Cache

NUMA Node 1

Compute Unit: Integer Core 8, L1 | Shared FP Scheduler | Integer Core 9, L1 | L2 Cache

Compute Unit: Integer Core 10, L1 | Shared FP Scheduler | Integer Core 11, L1 | L2 Cache

Compute Unit: Integer Core 12, L1 | Shared FP Scheduler | Integer Core 13, L1 | L2 Cache

Compute Unit: Integer Core 14, L1 | Shared FP Scheduler | Integer Core 15, L1 | L2 Cache

L3 Cache

https://www.olcf.ornl.gov/support/system-user-guides/titan-user-guide/

Running

- Parallel launcher is `aprun` (not `mpirun`)

```
aprun [options] program [program options]
```

- Numerous options

`-n`	Number of MPI tasks (up to 16 per node) **NOTE:** `-n`, not `-np`
`-N`	Number of tasks per node (1 – 16)
`-S`	Number of tasks per NUMA node (1 – 8)
`-j1`	Idles one integer core per Bulldozer module
`-d`	Number of cores to reserve (for threads) per MPI task

Running

- Must use both `OMP_NUM_THREADS` and the `-d` option to `aprun` for MPI+OpenMP codes
  - Set # threads via the variable (or calls in code)
  - Set aside cores with `-d`
- If doing module commands in batch job

```
. $MODULESHOME/init/bash
source $MODULESHOME/init/csh
```

- Multiple MPI tasks on a node can't all access the GPU by default

```
export CRAY_CUDA_MPS=1
setenv CRAY_CUDA_MPS 1
```

Batch System

- MOAB/Torque
  - PBS-like commands: `qsub`, `qstat`, `qdel`
  - Other commands: `showq`, `showstart`, `checkjob`
- Helpful directives

`#PBS -l walltime=HH:MM:SS`	Walltime request
`#PBS -l nodes=X`	Node request
`#PBS -j oe`	Send script STDOUT & STDERR to same file

- Directives to avoid

`#PBS -A`	Account (we're not using them)
`#PBS -q debug`	You'll be limited to 1 task

Sample Batch Script

```
#!/bin/bash
#PBS -l nodes=2,walltime=30:00
#PBS -j oe
## Remember, no #PBS -A or #PBS -q

. $MODULESHOME/init/bash
module load cudatoolkit
cd /whatever/directory
aprun -n16 -S4 -j1 ./a.out
```

Sample Batch Jobs

- Batch-interactive

```
$ qsub –I –lnodes=2,walltime=30:00
```

- Via script

```
$ qsub myscript.pbs
```

Errors From `aprun`

- `aprun` gives you lots of control over job layout which means it’s easy to make an invalid request
- The error message you get is dependent on the particular reason the request is invalid
- Just ask us if you get any of these

Errors From `aprun`

- Trying to run on more cores than are available

  ```
  apsched: claim exceeds reservation's node-count
  ```

- Intra-node layout problem

  ```
  apsched: claim exceeds reservation's CPUs
  ```

- Too many cores per NUMA node requested

  ```
  apsched: -S value cannot exceed max CPUs/NUMA node

  apsched: -S times -d cannot exceed max CPUs/NUMA
  node
  ```

Getting Data on Compute Nodes

- Typically you'll just use one of the filesystems that are common to all nodes
- If you need data in a non-shared directory like `/tmp`, copy w/`aprun`

```
aprun -n4 -N1 cp /local/file /tmp/remote/file
```

More Info

- This information comes from
  - *Titan User Guide* at https://www.olcf.ornl.gov
  - *How to OLCF* & *Best Practices* presentations
    - https://www.olcf.ornl.gov/training-event/how-to-olcf/
    - https://www.olcf.ornl.gov/training-event/olcf-users-webinar-how-to-olcf/
    - https://www.olcf.ornl.gov/wpcontent/uploads/2016/07/Best-Practices-v7.pdf
- These aren’t entirely relevant but system details should be similar