Argonne Training Program on Extreme-Scale Computing (ATPESC)

Quick Start on ATPESC Computing Resources

Ray Loy
ATPESC 2018 Deputy Program Director

Q Center, St. Charles, IL (USA)
Date 07/29/2018

NNSA National Nuclear Security Administration
U.S. DEPARTMENT OF ENERGY | Office of Science
Argonne NATIONAL LABORATORY

OUTLINE

- ALCF Systems
  - KNL (Theta)
  - Blue Gene/Q (Mira, Cetus, Vesta)
  - x86+GPU (Cooley)
- ANL JLSE
- OLCF
  - Cray XK7 (Titan)
- NERSC
  - KNL (Cori, Edison)

The DOE Leadership Computing Facility

- Collaborative, multi-lab, DOE/SC initiative ranked top national priority in *Facilities for the Future of Science: A Twenty-Year Outlook.*
- Mission: Provide the computational and data science resources required to solve the most important scientific & engineering problems in the world.
- Highly competitive user allocation program (INCITE, ALCC).
- Projects receive 100x more hours than at other generally available centers.
- LCF centers partner with users to enable science & engineering breakthroughs (Liaisons, Catalysts).

Argonne NATIONAL LABORATORY
OAK RIDGE National Laboratory
NATIONAL LABORATORY

Leadership Computing Facility System

	Argonne LCF		Oak Ridge LCF
System	Cray XC40	IBM Blue Gene/Q	Cray XK7
Name	Theta	Mira	Titan
Compute nodes	4,392	49,152	18,688
Node architecture	Intel Knights Landing, 64 cores	PowerPC, 16 cores	AMD Opteron, 16 cores NVIDIA K20x (Kepler) GPU
Processing Units	281,088 Cores	786,432 Cores	299,008 x86 Cores + 18,688 GPUs
Memory per node, (gigabytes)	192 DDR4 + 16 MCDRAM	16	32 + 6
Peak performance, (petaflops)	11.69	10	27

ALCF

ALCF Systems

- ***Theta - Cray* XC40**
  - 4,392 nodes / 281,088 cores
- ***Mira* – BG/Q**
  - 49,152 nodes / 786,432 cores
  - 786 TB of memory
  - Peak flop rate: 10 PetaFLOPs
  - 3,145,728 hardware threads
- ***Vesta** (T&D)* **- BG/Q**
  - 2,048 nodes / 32,768 cores
- ***Cetus** (debug)* **- BG/Q**
  - 4,096 nodes / 65,5368 cores
- ***Cooley (visualization & data analysis)* – Cray CS**
  - 126 nodes, each with
    - Two Intel Xeon E5-2620 Haswell 2.4 GHz 6-core processors
    - NVIDIA Tesla K80 graphics processing unit with 24 GB memory
    - 384 GB DDR4 memory

Argonne NATIONAL LABORATORY

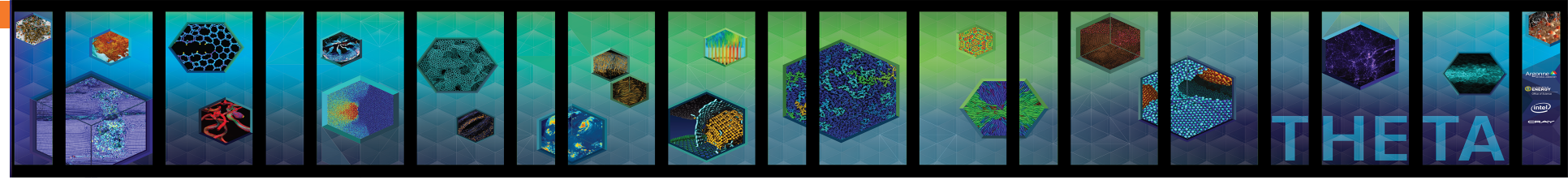
ECP EXASCALE COMPUTING PROJECT

JLSE

- Joint Laboratory for System Evaluation
  - ALCF / MCS evaluation of future HW and SW platforms
  - http://jlse.anl.gov
- Testbed systems
  - http://jlse.anl.gov/resources/hardware-under-development/
  - https://wiki.jlse.anl.gov/display/JLSEdocs/JLSE+Hardware
- Monday hands-on with FPGA system
- help@jlse.anl.gov

ALCF KNL (*Theta*)

Theta



Theta serves as a bridge to the exascale system coming to Argonne

- Serves as a bridge between Mira and Aurora, transition and data analytics system
- Cray XC40 system. Runs Cray software stack
- 11.69 PF peak performance
- 4392 nodes with 2nd Generation Intel® Xeon Phi™ processor
  - codename Knights Landing (KNL), 7230 SKU 64 cores 1.3GHz
  - 4 hardware threads/core
- 192GB DDR4 memory 16GB MCDRAM on each node
- 128GB SSD on each node
- Cray Aries high speed interconnect in dragonfly topology
- Initial file system: 10PB Lustre file system, 200 GB/s throughput

Filesystems

- GPFS
  - Home directories (/home) are in /gpfs/mira-home
    - Default quota 50GiB
    - Your home directory is backed up

- Lustre
  - Project directories (/projects) are in /lus/theta-fs0/projects (e.g. /projects/ATPESC2018)
    - Access controlled by unix group of your project
    - Default quota 1TiB
    - NOT backed up
  - With large I/O, be sure to consider **stripe width**

Modules (Theta ONLY)

- 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*
  - *help*
  - *list ← what is currently loaded*
  - *avail*
  - *load*
  - *unload*
  - *switch|swap*
  - *use ← add a directory to MODULEPATH*
  - *display|show*

Compilers

- For all compilers (Intel, Cray, Gnu, etc):
  - **Use:** cc, CC, ftn
  - **Do not use** mpicc, MPICC, mpic++, mpif77, mpif90
    - *they do not generate code for the compute nodes*
- Selecting the compiler you want using **"module swap"** or **"module unload"** followed by **"module load"**
  - Intel
    - PrgEnv-intel *This is the default*
  - Cray
    - module swap PrgEnv-intel PrgEnv-cray
    - **NOTE:** links libsci by default
  - Gnu
    - module swap PrgEnv-intel PrgEnv-gnu
  - Clang/LLVM
    - module swap PrgEnv-intel PrgEnv-llvm

Theta Job script

```
#!/bin/bash
#COBALT -t 10
#COBALT –n 2
#COBALT -A  ATPESC2018

# Various env settings are provided by Cobalt
echo $COBALT_JOBID  $COBALT_PARTNAME  $COBALT_JOBSIZE

aprun -n 16 -N 8 -d 1 -j 1 -cc depth ./a.out
status=$?

# could do another aprun here...

exit $status
```

Aprun overview

- Options
  - -n *total_number_of_ranks*
  - -N *ranks_per_node*
  - -d *depth* [number of cpus (hyperthreads) per rank]
  - -cc **depth** [Note: **depth** is a keyword]
  - -j *hyperthreads* [cpus (hyperthreads) per compute unit (core)]

- Env settings you may need
  - -e OMP_NUM_THREADS=*nthreads*
  - -e KMP_AFFINITY=...

- See also **man aprun**

Submitting a Cobalt job

- qsub –A <project> -q <queue> -t <time> -n <nodes> ./jobscript.sh

  E.g.

  qsub –A Myprojname –q cache-quad t –t 10 –n 32 ./jobscript.sh

- If you specify your options in the script via #COBALT, then just:
  - qsub jobscript.sh
- Make sure jobscript.sh is executable
- Without "-q", submits to the queue named "default"
- **man qsub** for more options
- ***Theta default (production) queue has 128 node minimum job size***

Managing your job

- qstat – show what's in the queue
  - qstat –u <username> # Jobs only for user
  - qstat <jobid> # Status of this particular job
  - qstat –fl <jobid> # Detailed info on job
- qdel <jobid>
- showres – show reservations currently set in the system
- **man qstat** for more options

Cobalt files for a job

- Cobalt will create 3 files per job, the basename <prefix> defaults to the jobid, but can be set with “qsub -O myprefix”
  - jobid can be inserted into your string e.g. "-O myprefix_$jobid"

- **Cobalt log file**: **<prefix>.cobaltlog**
  - created by Cobalt when job is submitted, additional info written during the job
  - contains submission information from qsub command, runjob, and environment variables
- **Job stderr file**: **<prefix>.error**
  - created at the start of a job
  - contains job startup information and any content sent to standard error while the user program is running
- **Job stdout file**: **<prefix>.output**
  - contains any content sent to standard output by user program

Interactive job

- Useful for short tests or debugging
- Submit the job with –I (letter I for Interactive)
  - Default queue and default project
    - qsub –I –n 32 –t 30
  - Specify queue and project:
    - qsub –I –n 32 –t 30 –q training –A ATPESC2018
- Wait for job's shell prompt
  - *This is a new shell* with env settings e.g. COBALT_JOBID
  - Exit this shell to end your job
- From job's shell prompt, run just like in a script job, e.g.
  - aprun -n 512 -N 16 -d 1 -j 1 -cc depth ./a.out
- After job expires, apruns will fail. *Check* **qstat $COBALT_JOBID**

Core files and debugging

- Abnormal Termination Processing (ATP)
  - Set environment **ATP_ENABLED=1** in your job script before aprun
  - On program failure, generates a merged stack backtrace tree in file **atpMergedBT.dot**
  - View the output file with the program **stat-view** (module load stat)

- Notes on linking your program
  - make sure you load the "atp" module before linking
    - to check, *module list*

- Other debugging tools
  - You can generate STAT snapshots asynchronously
  - Full-featured debugging with DDT
  - More info at
    - https://tinyurl.com/debugging-cpw-2018-05

ALCF Blue Gene/Q (*Mira*, *Cetus*, *Vesta*)

Softenv (BG/Q and Cooley ONLY)

- Similar to **modules** package
- Keys are read at login time to set environment variables like PATH.
  - Mira, Cetus, Vesta: ~/.soft
  - Cooley: ~/.soft.cooley
- To get started:

```
# This key selects XL compilers to be used by mpi wrappers
+mpiwrapper-xl
@default
# the end - do not put any keys after the @default
```

- After edits to .soft, type "resoft" or log out and back in again

Using compiler wrappers

- **IBM XL cross-compilers:**
  - SoftEnv key: +mpiwrapper-xl
  - Non-thread-safe: mpixlc, mpixlcxx, mpixlf77, mpixlf90, mpixlf95, mpixlf2003, etc.
  - **Thread-safe** (add _r suffix): mpixlc_r, mpixlcxx_r, mpixlf77_r, etc.
  - Example: mpixlc –O3 –o hellompi hellompi.c

- **GNU cross-compilers:**
  - SoftEnv key: +mpiwrapper-gcc
  - mpicc, mpicxx, mpif77, mpif90

- **CLANG cross-compilers:**
  - SoftEnv key: +mpiwrapper-bgclang
  - mpiclang, mpiclang++, mpiclang++11

http://www.alcf.anl.gov/user-guides/software-and-libraries

BG/Q Job script

- Sample:

```
#!/bin/bash
#COBALT -n 32 -t 30 -q training -A ATPESC2018
# -p is mode (how many ranks per node)
# --np is number of ranks
runjob -p 16 --np 32 --block $COBALT_PARTNAME  :  hellompi
# Note:  exit status of this script is runjob's status
```

- Some args use ***single*** dash and some ***double*** dash (**man runjob**)
- Don't forget --block. COBALT_PARTNAME is set automatically by Cobalt.
- You can do multiple runjobs in succession
  - Use normal shell redirection to separate output
- Must use --envs to pass environment variables into your program
- Output to <jobid>.{output,error,cobaltlog} (use -O to change prefix)

Submitting your job

- qsub –A <project> -q <queue> -t <time> -n <nodes> --mode script ./jobscript.sh

  E.g.

  qsub –A ATPESC2018 –q training –t 10 –n 32 –mode script ./jobscript.sh

  Note: runs on Mira should use "default" queue

- If you specify your options in the script via #COBALT, then just:
  - qsub jobscript.sh
- Make sure jobscript.sh is executable
- Without "-q", submits to the queue named "default"
- **man qsub** for more options

Interactive job

- Useful for short tests or debugging
- Submit the job with –I (letter I for Interactive)
  - Default queue and default project
    - qsub –I –n 32 –t 30
  - For the workshop:
    - qsub –I –n 32 –t 30 –q training –A ATPESC2018
- Wait for job's shell prompt
  - *This is a new shell* with settings COBALT_PARTNAME, COBALT_JOBID
  - Exit this shell to end your job
- **Run "wait-boot" ← Important!**
- From job's shell prompt, run just like in a script job:
  - runjob –block $COBALT_PARTNAME –p 16 –np 32 : hellompi
- After job expires, runjob will fail. *Check* **qstat $COBALT_JOBID**

About node count and mode

- Node count
  - Minimum physical partition sizes available depend on machine
    - Vesta: 32 Cetus: 128 Mira: 512
    - Your job will get the smallest available size >= what you ask for
      - It is reserved for you; you are charged for entire partition
- Mode
  - How many MPI ranks per node
    - Possible values: 1,2,4,8,16,32,64
  - A node has 16 cores, each can run 4 threads
    - For modes < 16, an MPI rank will be assigned more than one core
    - Example: "-p 4" can run up to 16 threads per MPI rank

Using OpenMP

- Shared-memory parallelism is supported within a single node
  - Use MPI across compute nodes, OpenMP within a compute node
- **For XL compilers, thread-safe compiler version should be used** (mpixlc_r etc.) with any threaded application (either OMP or Pthreads)
  - OpenMP standard directives are supported (version 3.1)
  - Compile with –qsmp=omp,noauto (Note: debugging use *noopt*)
  - Increase default thread stack size using environment value XLSMPOPTS=stack=NNN (value per thread, e.g. 10M)
- Setting number of OpenMP threads
  - set using environment variable OMP_NUM_THREADS
  - must be exported to the compute nodes using runjob –envs

- Example: 32 nodes / 512 ranks / 4 threads per rank:

```
#!/bin/bash
#COBALT –n 32 –t 30
runjob –block $COBALT_PARTNAME –p 16 --np 512 --envs OMP_NUM_THREADS=4 : a.out
```

ALCF x86+GPU (*Cooley*)

Cooley Job Script

- More like a typical Linux cluster
- Job script different than BG/Q.
  - Example test.sh:

```
#!/bin/sh
NODES=`cat $COBALT_NODEFILE | wc -l`
PROCS=$((NODES * 12))
mpirun -f $COBALT_NODEFILE -n $PROCS myprog.exe
```

  - Submit on 5 nodes for 10 minutes

```
qsub -n 5 -t 10 -q training -A ATPESC2018 ./test.sh
```

  - Refer to online user guide for more info

ATPESC Resources

ALCF – Mira, Cetus, Vesta, Cooley, and Theta

- **Project name: ATPESC2018**
- **Note:** use your ALCF Username. The password will be your old/newly established PIN + token code displayed on the token.
- **Support:** on-site ALCF staff available to help you!! and support@alcf.anl.gov
- **Reservations:** Please check the details of the reservations directly on each machine (**command**: showres)
- **Queue: training**

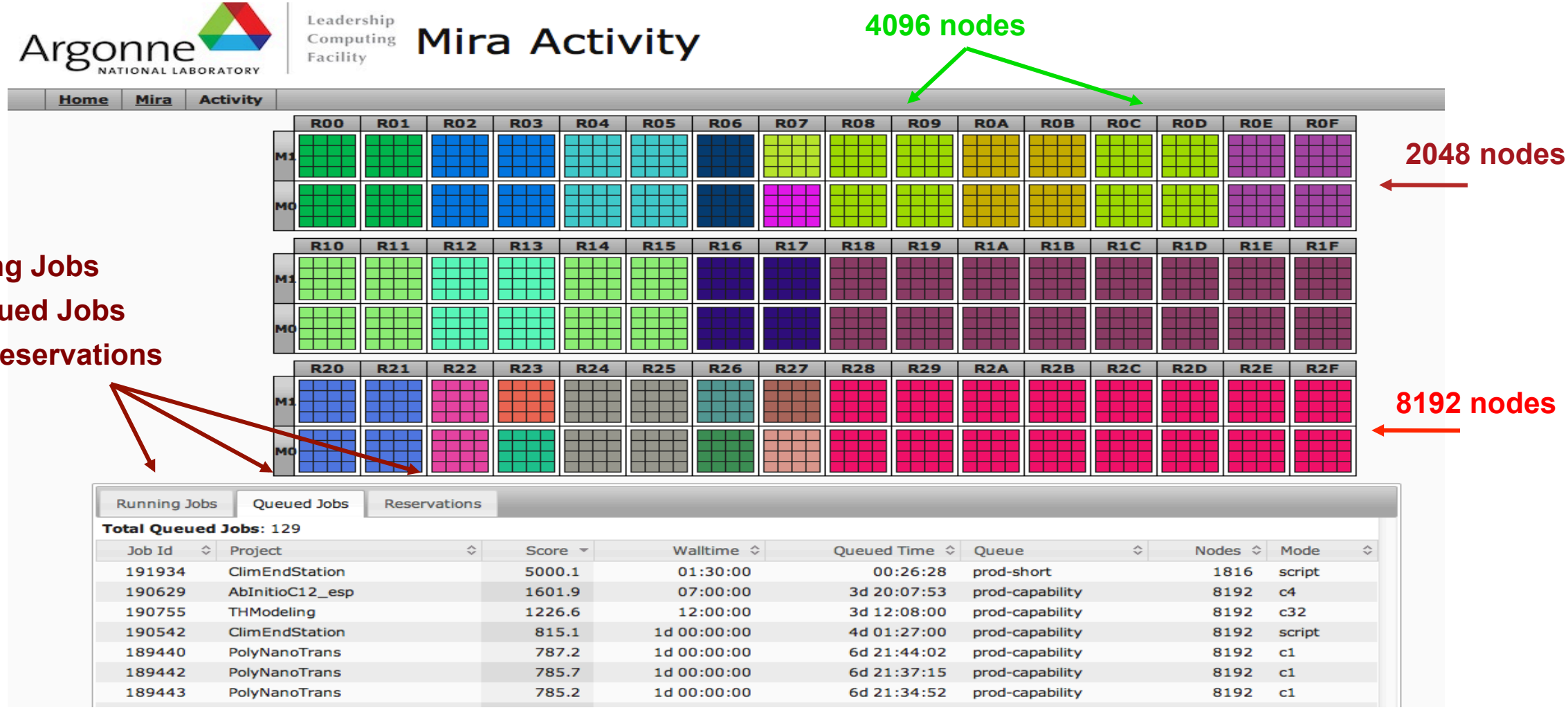
Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

Cryptocard tips

- The displayed value is a hex string. Type your PIN followed by all letters as CAPITALS.
- If you fail to authenticate the first time, you may have typed it incorrectly
  - Try again with the **same crypto string** (do NOT press button again)
- If you fail again, try a different ALCF host with a fresh crypto #
  - A successful login resets your count of failed logins
- Too many failed logins → your account locked
  - Symptom: You get password prompt but login denied even if it is correct
- Too many failed logins from a given IP → the IP will be blocked
  - Symptom: connection attempt by ssh or web browser will just time out

Machine status web page



Job Id	Project	Score	Walltime	Queued Time	Queue	Nodes	Mode
191934	ClimEndStation	5000.1	01:30:00	00:26:28	prod-short	1816	script
190629	AbInitioC12_esp	1601.9	07:00:00	3d 20:07:53	prod-capability	8192	c4
190755	THModeling	1226.6	12:00:00	3d 12:08:00	prod-capability	8192	c32
190542	ClimEndStation	815.1	1d 00:00:00	4d 01:27:00	prod-capability	8192	script
189440	PolyNanoTrans	787.2	1d 00:00:00	6d 21:44:02	prod-capability	8192	c1
189442	PolyNanoTrans	785.7	1d 00:00:00	6d 21:37:15	prod-capability	8192	c1
189443	PolyNanoTrans	785.2	1d 00:00:00	6d 21:34:52	prod-capability	8192	c1

http://status.alcf.anl.gov/mira/activity (a.k.a. The Gronkulator)

References

- Sample files
  - On Theta or Cooley
    - /projects/ATPESC2018/ALCF_Getting_Started
  - On Vesta, Mira, Cetus:
    - /projects/ATPESC2018/getting-started
- Online docs
  - www.alcf.anl.gov/user-guides
  - Theta/Cooley Getting Started: https://tinyurl.com/theta-gsv-2018-07
  - Mira Getting Started: https://tinyurl.com/mira-gsv-2018-07
  - Debugging: https://tinyurl.com/debugging-cpw-2018-05

Additional resources: OLCF and NERSC

ATPESC Resources

OLCF – Titan

- **Project name: TRN001**
- **Note:** use the Username printed on the envelope the token came in. It will be csep01, csep02, etc. The password will be your newly established PIN + token code displayed on the token.
- **Support:** help@olcf.anl.gov or call 1-865-241-6536
- See documents in your Argonne Folder for additional information

ATPESC Resources

NERSC – Edison and Cori

- **Project name:** ntrain
- **Note:** *ssh machine_name.nersc.gov*
- **Support:** accounts@nersc.gov or call 1-800-666-3772

Edison (Cray XC30):

http://www.nersc.gov/users/computational-systems/edison/running-jobs/

Cori (Cray XC40):

http://www.nersc.gov/users/computational-systems/cori/running-jobs/

Questions?

- *Use this presentation as a reference during ATPESC!*

- Supplemental info will be posted as well

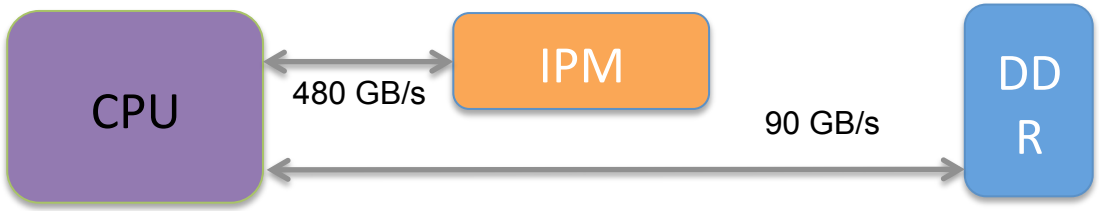
Supplemental Info

Theta Memory Modes - IPM and DDR

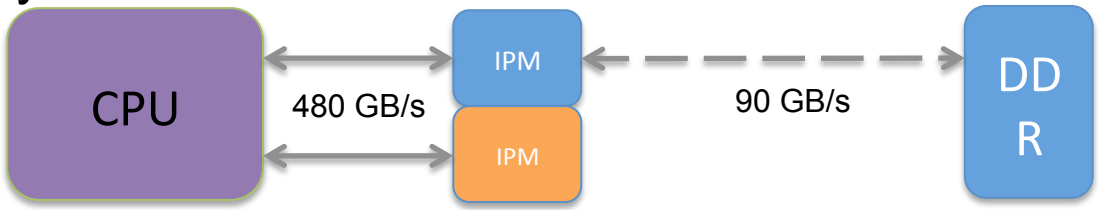
Selected at node boot time

Cache

Flat



Hybrid

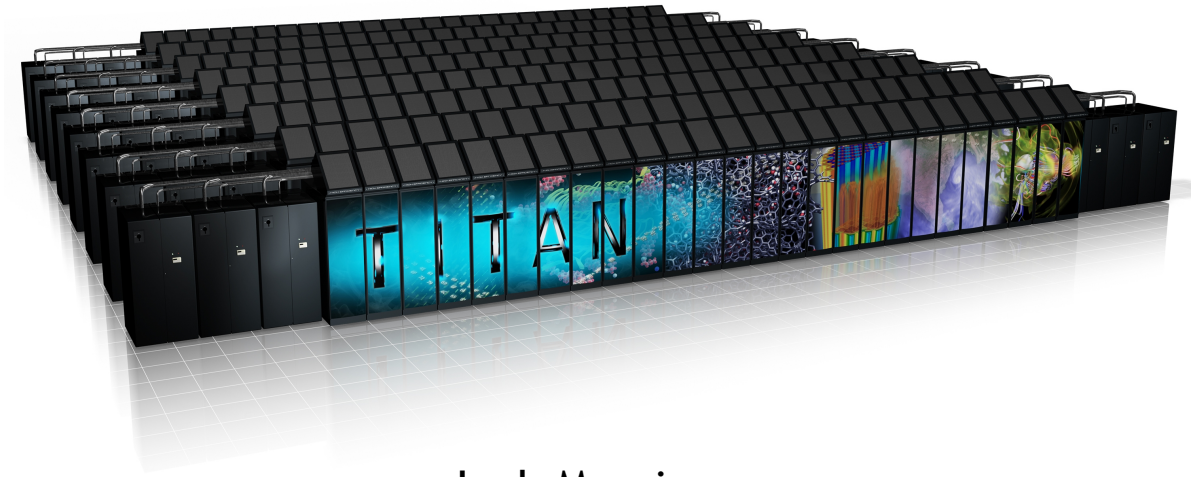


- **Two memory types**
- In Package Memory (IPM)
  - 16 GB MCDRAM
  - ~480 GB/s bandwidth
- Off Package Memory (DDR)
  - Up to 384 GB
  - ~90 GB/s bandwidth
- **One address space**
- Possibly multiple NUMA domains
- **Memory configurations**
- Cached: DDR fully cached by IPM
  - Flat: user managed
- Hybrid: ¼, ½ IPM used as cache
  - **Managing memory:**
- jemalloc & memkind libraries
- Pragmas for static memory allocations

Theta queues and modes

- MCDRAM and NUMA modes can only be set by the system when nodes are rebooted. *Users cannot directly reboot nodes.*
- Submit job with the --attrs flag to get the mode you need. E.g.
  - qsub −n 32 −t 60 −attrs mcdram=cache:numa=quad ./jobscript.sh
- Other mode choices
  - mcdram: cache, flat, split, equal
  - numa: quad, a2a, hemi, snc2, snc4
- Queues
  - Normal jobs use queue named "default"
  - Debugging: debug-cache-quad, debug-flat-quad
    - Note: pre-set for mcdram/numa configuration
  - "qstat −Q" lists all queues

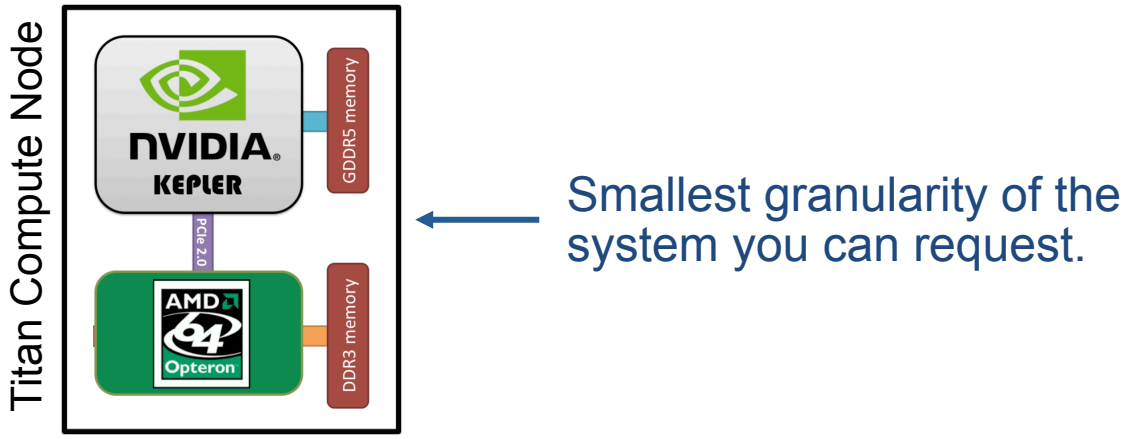
Titan
(Cray XK7, OLCF)



Jack Morrison
HPC Engineer
User Assistance and Outreach Group
Oak Ridge National Laboratory
morrisonjc@ornl.gov

Titan Overview

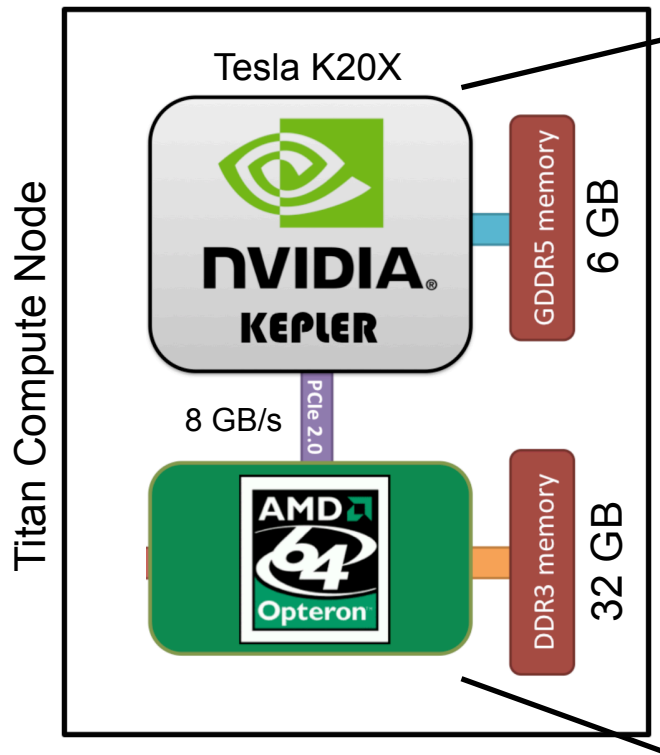
- Cray XK7 – Hybrid Architecture (CPUs + GPUs)
  - CPUs: several cores optimized for serial work
  - GPUs: thousands of core optimized for parallel work
  - Offload compute-intensive regions of code to run on GPUs
- 27 PF (peak performance)
- 18,688 compute nodes

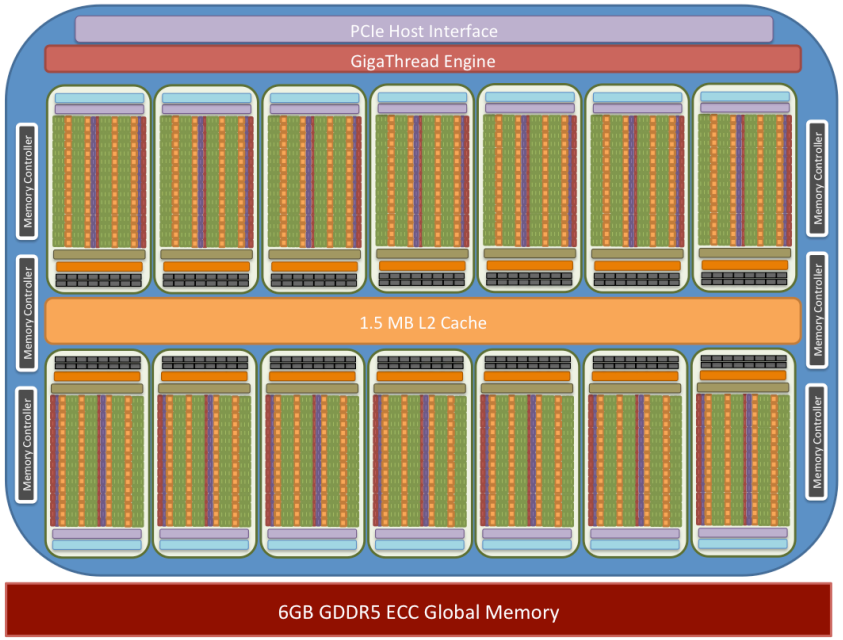


Smallest granularity of the system you can request.

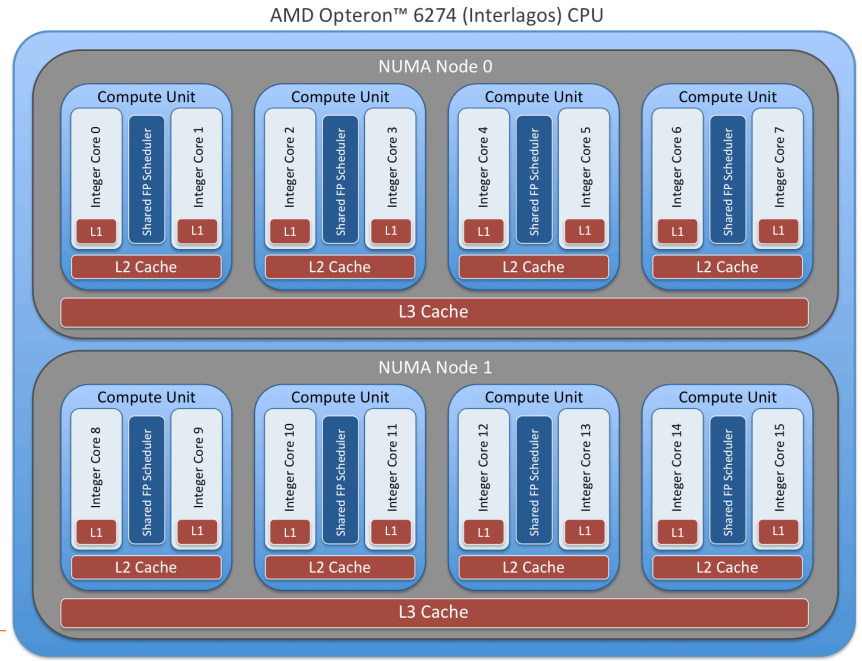
Feature	Titan
Number of Nodes	18,688
Node performance	1.4 TF
Memory per Node	32 GB DDR3 + 6 GB GDDR5
Total System Memory	710 TB
System Interconnect	Gemini (6.4 GB/s)
Interconnect Topology	3D Torus
Bi-Section Bandwidth	112 TB/s
Processors	1 AMD Opteron™ 1 NVIDIA Kepler™
File System	32 PB, 1 TB/s, Lustre®
Power Consumption	9 MW

Titan Compute Nodes





2688 cores



16 cores

Titan questions during ATPESC?

Jack Morrison
morrisonjc@ornl.gov

A few resources…

User Guide (start here!):
https://www.olcf.ornl.gov/for-users/system-user-guides/titan/

File Systems and Data Transfers: Recording https://vimeo.com/278737802
Slides https://www.olcf.ornl.gov/wp-content/uploads/2018/06/Intro_to_HPC_File_Systems_Data_Transfers.pdf

Programming Environment: Recording https://vimeo.com/279278490
Slides https://www.olcf.ornl.gov/wp-content/uploads/2018/07/titan_prog_environ_2018-copy.pdf

Batch Scheduler & Job Launcher: Recording https://vimeo.com/279278776
Slides https://www.olcf.ornl.gov/wp-content/uploads/2018/06/Intro_to_HPC_Titan-Job-Launch-Intro.pdf

Intro to GPU Computing: Recording https://vimeo.com/279319729
Slides https://www.olcf.ornl.gov/wp-content/uploads/2018/06/intro_to_HPC_gpu_computing.pdf