

Argonne Training Program on Extreme-Scale Computing (ATPESC)

Quick Start on ATPESC Computing Resources

Ray Loy

ATPESC 2019 Deputy Program Director

Q Center, St. Charles, IL (USA)

Date 07/28/2019



EXASCALE COMPUTING PROJECT

OUTLINE

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

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).



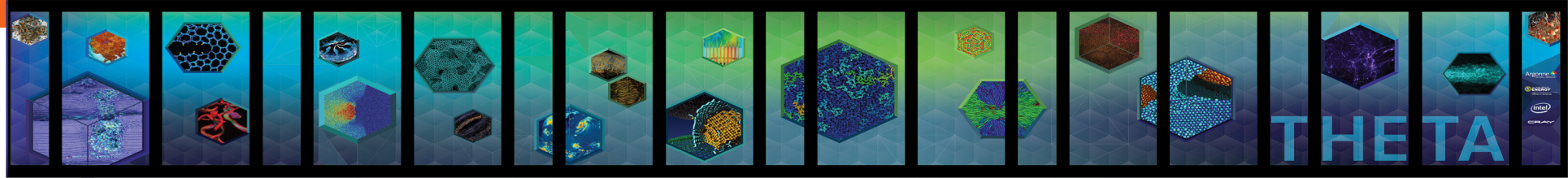
Leadership Computing Facility System

	Argonne LCF		Oak Ridge LCF
System	Cray XC40	IBM Blue Gene/Q	IBM
Name	Theta	Mira	Summit
Compute nodes	4,392	49,152	4608
Node architecture	Intel Knights Landing, 64 cores	PowerPC, 16 cores	2 x IBM POWER9 22 cores 6 x NVIDIA V100 (Volta) GPU
Processing Units	281,088 Cores	786,432 Cores	202,752 POWER9 Cores + 27648 GPUs
Memory per node, (gigabytes)	192 DDR4 + 16 MCDRAM	16	512 DDR4 + 96 HBM2 + 1600 NVM
Peak performance, (petaflops)	11.69	10	200

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 / 65,536 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





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
 - 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

Theta - 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/ATPESC2019)
 - **CREATE A SUBDIRECTORY** /projects/ATPESC2019/your_username
 - Access controlled by unix group of your project
 - Default quota 1TiB
 - NOT backed up
- ⦿ With large I/O, be sure to consider **stripe width**

- ⦿ **RECOMMENDATION**

- ⦿ Source code and compilation in your home directory. Data files and execution in project directory.

Theta - 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*

Theta - 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 ATPESC2019

# 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
```

Theta - aprun overview

- ⦿ Start a parallel execution (equivalent of *mpirun*, *mpiexec* on other systems)
 - ⦿ *Must be invoked from within a batch job that allocates nodes to you!*
- ⦿ 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 default -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"**
 - ⊙ For ATPESC reservations, specify e.g. `"-q R.ATPESC2019"` (see *showres* output)
 - ⊙ For small tests outside of reservations, use e.g. `"-q debug-cache-quad"`
- ⊙ ***Theta "default" (production) queue has 128 node minimum job size***
 - ⊙ The ATPESC reservation does not have this restriction
- ⊙ `man qsub` for more options

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://www.alcf.anl.gov/files/Loy-comp_perf_workshop-debugging-2019-v1.2.pdf

Machine status web page

Running Jobs
Queued Jobs
Reservations



<http://status.alcf.anl.gov/theta/activity> (a.k.a. The Gronkulator)

ALCF Cooley (x86+GPU)

Cooley - Softenv (Cooley and BG/Q 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

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

ALCF References

- Sample files (Theta, Cooley, Mira, Vesta, Cetus)
 - /projects/ATPESC19_Instructors/ALCF_Getting_Started/examples
- Online docs
 - www.alcf.anl.gov/user-guides
 - Getting Started Presentations (*slides and videos*)
 - Theta and Cooley ; BG/Q (Mira, Vesta, Cetus)
 - <https://www.alcf.anl.gov/workshops/2019-getting-started-videos>
 - Debugging:
 - https://www.alcf.anl.gov/files/Loy-comp_perf_workshop-debugging-2019-v1.2.pdf

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

ATPESC Resources



- ◉ *Joint Laboratory for System Evaluation*
 - ◉ ALCF / MCS evaluation of future HW and SW platforms
 - ◉ <http://jlse.anl.gov>
- ◉ Systems
 - ◉ <https://wiki.jlse.anl.gov/display/JLSEdocs/JLSE+Hardware>
- ◉ Monday hands-on with FPGA system
- ◉ Email: help@jlse.anl.gov

ATPESC Resources



- **Project name:** **ATPESC2019** (qsub –A ATPESC2019 ...)
- **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:** **R.ATPESC2019** (check *showres*) or **default**

ATPESC Resources



- Summit User Guide <https://www.olcf.ornl.gov/for-users/system-user-guides/summit/>
- Tools to learn how to use the `jsrun` job launcher
 - [Hello jsrun](#) – A “Hello, World!”-type program to help understand resource layouts on Summit/Ascent nodes.
 - [jsrunVisualizer](#) – A web-based tool to learn the basics of `jsrun`.
 - [Jsrun Quick Start Guide](#) – A very brief overview to help get you started
- [OLCF Tutorials at https://github.com/olcf-tutorials](https://github.com/olcf-tutorials)
- See documents in your Argonne Folder for additional information
- For token issues, call: 865.241.6536 (24x7). For other questions, email: help@olcf.ornl.gov

ATPESC Resources



- **9688 KNL (68-core) nodes + 2388 Haswell (16-core) nodes**
- **Logging in:** `ssh trainNNN@cori.nersc.gov`
- **Project (repository) name:** *ntrain*
- **Submit jobs**
<http://www.nersc.gov/users/computational-systems/cori/running-jobs/>
- **Reservations**
`#SBATCH –reservation=nameofreservation ./mybatchscript`
- **Support:** accounts@nersc.gov or call 1-800-666-3772

Reservations (Check *showres* for updates)

TRACK 1 Hardware Architectures	TRACK 2 Programming Models and Languages		TRACK 3 Data Intensive Computing and I/O	TRACK 4 Visualization and Data Analysis
No reservations	Tue 7/30 12h @ 9:30 AM Theta (40 nodes) Cooley (40 nodes)	Wed 7/31 & Thu 8/1 13h @ 8:30 AM Cooley (80 nodes)	Fri 8/2 12h @ 9:00 AM Theta (79 nodes)	Mon 8/5 10.5h @ 10:30 AM Cooley (78 nodes)
TRACK 5 Numerical Algorithms and Software for Extreme-Scale Science	TRACK 6 Performance Tools and Debuggers	TRACK 7 Software Productivity	TRACK 8 Machine Learning and Deep Learning for Science	
Tue 8/6 9h @ 8:30 AM Cooley (80 nodes)	Wed 8/7 10h @ 8:30 AM Theta (1K nodes) 4.5h @ 6:30 PM Theta (2K nodes)	Thu 8/8 No reservations	Fri 8/9 TBD	

Questions?

- *Use this presentation as a reference during ATPESC!*
- Supplemental info will be posted as well

Hands-on exercise

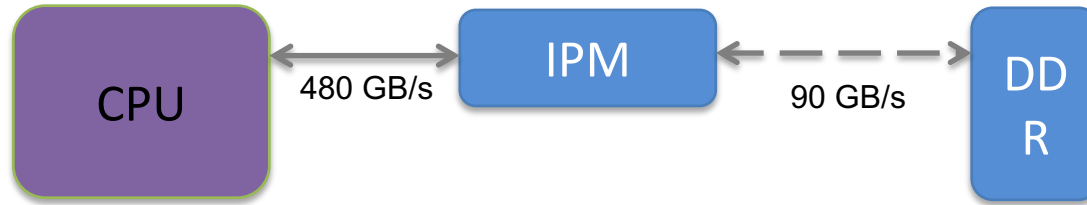
- Theta
 - `/projects/ATPESC19_Instructors/ALCF_Getting_Started/examples/theta/compilation`
 - Cooley
 - `/projects/ATPESC19_Instructors/ALCF_Getting_Started/examples/cooley/compilation`
-
1. Log in
 2. Create your own subdirectory in `/projects/ATPESC2019`
 3. Copy example code to your directory
 4. Compile (Makefile) and submit job (`jobscripts.sh`)

Supplemental Info

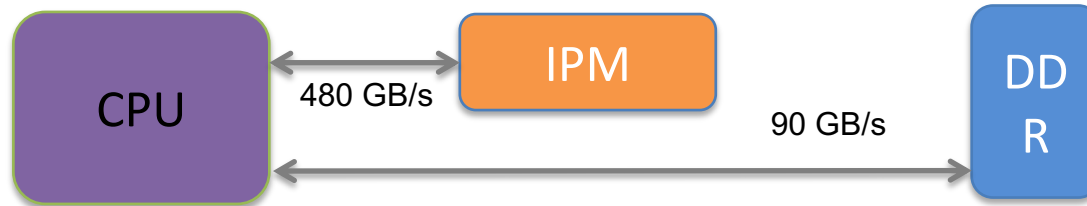
Theta Memory Modes - IPM and DDR

Selected at node boot time

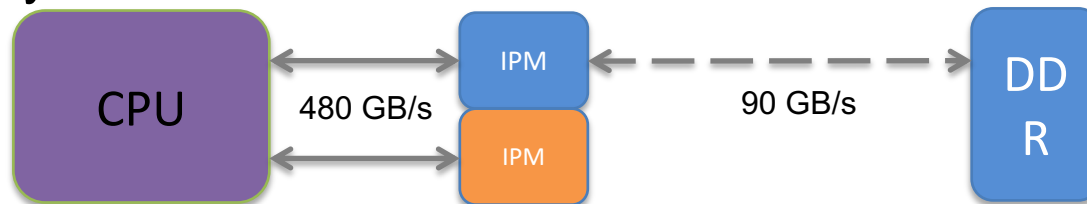
Cache



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: $\frac{1}{4}$, $\frac{1}{2}$ 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