Getting Started on Theta

Graham Fletcher
ALCF Catalyst Team

ALCF Computational Performance Workshop
May 4, 2021

www.anl.gov

Outline

http://www.alcf.anl.gov/presentations

- Theta Hardware
  - System overview
  - Processor
  - File systems
- Software
  - Operating System and Programming environment
  - Building Your Code
  - Tools
- Queuing and running jobs
  - Cobalt
  - aprun
  - Queues

Tips for troubleshooting

Theta- Hardware

Theta system overview

Architecture: Cray XC40

Aries interconnect with
Dragonfly configuration

Total **Nodes**: 4,392

Total **Cores**: 281,088

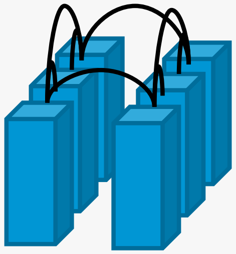
Total MCDRAM: 70 TB

Total DDR4: 843 TB

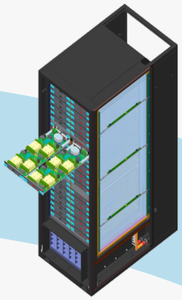
Total SSD: 562 TB

Peak performance: **11.69 PF**

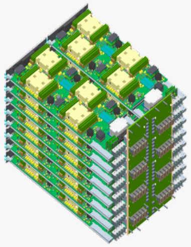
For more information, see
https://www.alcf.anl.gov/support-center/theta



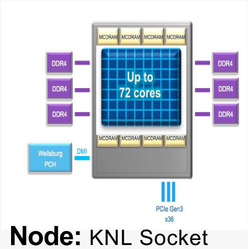
System: 24 Cabinets
4392 Nodes, 1152 Switches
Dual-plane, 12 groups, Dragonfly 7.2 TB/s Bi-Sec
11.69 PF Peak
70 TB MCDRAM, 843 TB DRAM



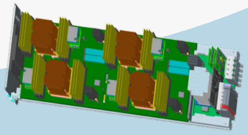
Cabinet: 3 Chassis, 75kW liquid/air cooled
510.72 TF 3TB MCDRAM, 36TB DRAM



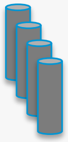
Chassis: 16 Blades, 16 Cards
64 Nodes, 16 Switches
170.24 TF 1TB MCDRAM, 12TB DRAM



Node: KNL Socket
192 GB DDR4 (6 channels) **2.66 TF** 16GB MCDRAM

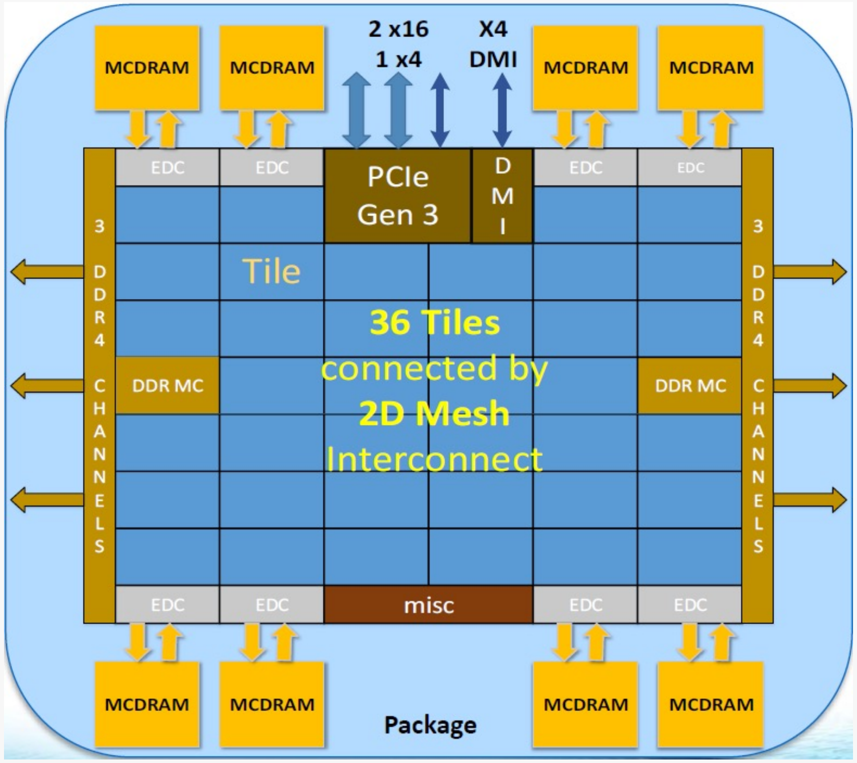


Compute Blade:
4 Nodes/Blade + Aries switch
128GB SSD
10.64 TF 64GB MCDRAM
768GB DRAM



Sonexion Storage
4 Cabinets
Lustre file system
10 PB usable
210 GB/s

Theta node: Intel Xeon Phi 7230 SKU (‘Knights Landing’)



Theta node: 64 Cores

- 32 tiles
- 2 cores per tile
- 2.1 TF per node

Cores

- 1.3 GHz
- 1MB L2 shared
- 2 VPU with AVX-512
- 512-bit (8 DP)
- 1, 2, or 4 h/w threads

On-Package Memory

- 16 GB MCDRAM
- ~450 GB/s bandwidth

Memory/node: 192 GB
DDR4 SDRAM

SSD/node: 128 GB

File systems

GPFS

– Home directories are located on /gpfs/mira-home/

– Default quota 100GB

Lustre

– Project directories (/projects) are in /lus/theta-fs0/projects, also Eagle (/eagle or /lus/eagle/projects) and Grand (/grand or /lus/grand/projects)

  - Access controlled by unix group of your project
  - Default quota 1TiB
  - NOT backed up

– With large I/O, be sure to consider stripe width

For more information, see https://www.alcf.anl.gov/support-center/theta/theta-file-systems

Theta- Software

Operating System

Login nodes: SUSE Enterprise Linux based full Cray Linux Environment (CLE) OS
Compute nodes: Compute Node Linux (CNL)

Cray Programming Environment

Languages: Fortran, c, c++, Python

Programming Models: (Distributed Memory) MPI … (Shared) OpenMP, OpenACC; (PGAS) UPC, CAF …

Compilers: Cray, GNU, (3rd party) PGI …

Tools: (debuggers) DDT … (debugging tools) ATP … (performance analysis) CrayPAT … (porting) Reveal …

Optimized Libraries: BLAS, LAPACK, ScaLAPACK; Cray PETSc …

For more information, see https://www.alcf.anl.gov/support-center/theta

Logging into Theta

```
ssh [your username]@theta.alcf.anl.gov      ⏎
```

press the button on your cryptocard

enter your 4-digit PIN followed by the cryptocard sequence (upper-case!) ⏎

Building Your Code

Compiler wrappers

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)

Select the compiler you want using "module swap" or "module unload" followed by "module load”, eg.

– 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

…

For more info, see https://www.alcf.anl.gov/support-center/theta/compiling-and-linking-overview-theta-thetagpu

Tools: performance, profiling, debugging

Non-system libraries and tools – /soft/

Debuggers (eg. DDT) – /soft/debuggers

- For more information, see https://www.alcf.anl.gov/support-center/theta/introduction-debugging

Performance tools (eg. TAU, hpctoolkit, darshan, memlog) – /soft/perftools

- For more information, see https://www.alcf.anl.gov/support-center/theta

Installed compilers (including llvm and intel beta releases) – /soft/compilers

Applications – /soft/applications

Libraries (eg. argobots, bolt, breakpad) – /soft/libraries

Queuing and Running Jobs

Queuing a job

On Theta the job scheduler is called **Cobalt**
Executables are invoked within a **script** (bash, csh, …)

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

Make sure jobscript.sh is executable.
Without "-q", submits to the queue named "default”.

For more information, see https://www.alcf.anl.gov/support-center/theta/submit-job-theta

Within the script jobs are launched using ***aprun*** ...

Launching executables with *aprun*

(on compute nodes)
aprun <options> <executable> <args>

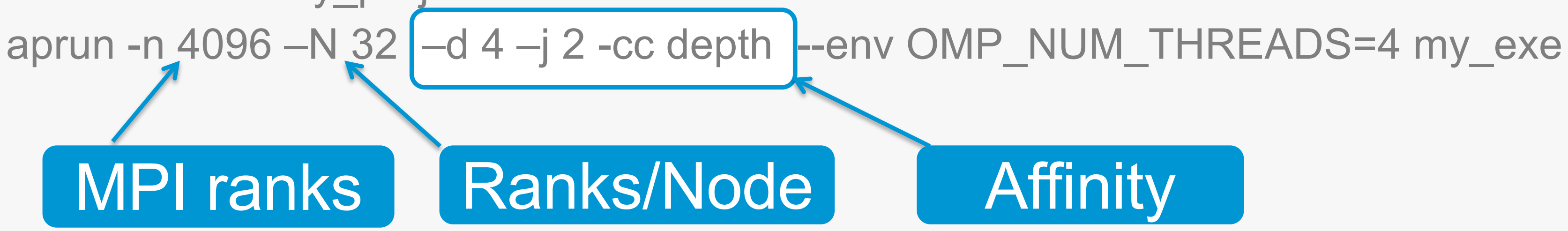
Options

– Total number of MPI ranks:	–n <total_num_ranks>
– Number of MPI ranks per node:	–N <num_ranks_per_node>
– Number of hyperthreads per core:	–j <num_threads>
– Number of hyperthreads per MPI rank (depth):	–d <num_threads>
– MPI rank and thread placement:	-cc depth
– Environment variables:	--env <env_var>

For more information, see https://www.alcf.anl.gov/support-center/theta/running-jobs-and-submission-scripts

Example Submission

```
> cat my_script.sh
#!/bin/sh
#COBALT –A my_project –t 60 –n 128
aprun -n 4096 –N 32 –d 4 –j 2 -cc depth --env OMP_NUM_THREADS=4 my_exe
```



In this example the qsub options are specified in the script via the #COBALT syntax, then just type:

```
> qsub my_script.sh
```

For further information on bundling multiple jobs concurrently, simultaneously, and using workflow tools, see https://www.alcf.anl.gov/support-center/theta/running-jobs-and-submission-scripts

Production Queues, policy

There is a single submission queue for the entire system: default

Priority is given to jobs using at least 20% of Theta (802 nodes)

There is a global limit of ten (10) jobs running per user

There is a global limit of twenty (20) jobs in queue per user

There is a minimum job time of thirty (00:30:00) minutes for the default queue

There is a minimum allocation of 8 nodes

For more information, see https://www.alcf.anl.gov/support-center/theta/job-scheduling-policy-theta

Production Queues, policy

Wall-clock limits are a step-wise function designed to encourage scaling:

– node count >= 8 nodes : maximum 2:00:00 hours

– node count >= 16 nodes : maximum 4:00:00 hours

– node count >= 128 nodes : maximum 6:00:00 hours

– node count >= 384 nodes : maximum 12:00:00 hours

– node count >= 648 nodes : maximum 24:00:00 hours

For more information, see https://www.alcf.anl.gov/support-center/theta/job-scheduling-policy-theta

Debugging Queues, policy

There are two 16-node debugging queues:
debug-cache-quad
debug-flat-quad
(for more on memory modes, see https://www.alcf.anl.gov/support-center/theta/theta-memory-modes)

Hardware is dedicated to each queue (nodes are not rebootable to another mode).

Limits:

- Maximum 1 hour wall-clock time
- Maximum 1 job per user
- Maximum 8 nodes

For more information, see https://www.alcf.anl.gov/support-center/theta/job-scheduling-policy-theta

Cobalt files for a job

Cobalt will create 3 files per job

Cobalt log file: <prefix>.cobaltlog

– created by Cobalt when job is submitted, additional info written during the job
– contains submission information from qsub command, aprun, 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

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"

Managing your job

Chain your jobs by specifying dependencies:
qsub --dependencies <jobid1>:<jobid2> …

qstat – show what's in the queue

```
– qstat –Q                  # Check available queues
– qstat –u <username>       # Jobs only for user
– qstat <jobid>             # Status of this particular job
– qstat –fl <jobid>         # Detailed info on job
```

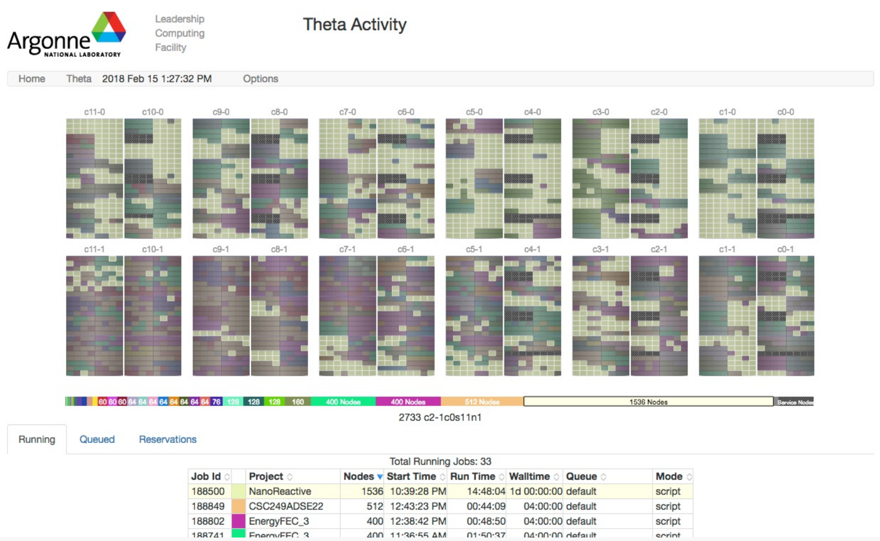
man qstat for more options

Other commands

Check available nodes-
nodelist

Show reservations currently set in the system-
showres

http://status.alcf.anl.gov/theta/activity



Theta Activity

2733 c2-1c0s11n1

Total Running Jobs: 33

Job Id		Project	Nodes	Start Time	Run Time	Walltime	Queue	Mode
188500	🟨	NanoReactive	1536	10:39:28 PM	14:48:04	1d 00:00:00	default	script
188849	🟧	CSC249ADSE22	512	12:43:23 PM	00:44:09	04:00:00	default	script
188802	🟪	EnergyFEC_3	400	12:38:42 PM	00:48:50	04:00:00	default	script
188741	🟩	EnergyFEC_3	400	11:36:55 AM	01:50:37	04:00:00	default	script

Managing your job

Other Cobalt commands

To delete a job from the queue-

```
qdel <jobid>
```

Alter parameters of a queued job-

```
qalter [most qsub options] <jobid1> …
```

, except the queue itself-

```
qmove <destination_queue> <jobid>
```

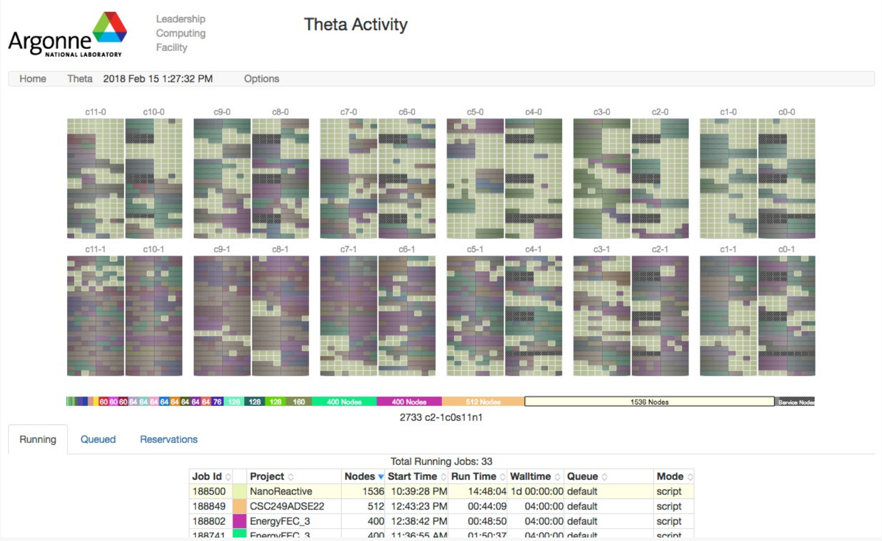
Place a hold on a job-

```
qhold <jobid>
```

Release a job-

```
qrls <jobid>
```

http://status.alcf.anl.gov/theta/activity



Theta Activity

2018 Feb 15 1:27:32 PM

2733 c2-1c0s11n1

Total Running Jobs: 33

Job Id		Project	Nodes	Start Time	Run Time	Walltime	Queue	Mode
188500	🟨	NanoReactive	1536	10:39:28 PM	14:48:04	1d 00:00:00	default	script
188849	🟧	CSC249ADSE22	512	12:43:23 PM	00:44:09	04:00:00	default	script
188802	🟪	EnergyFEC_3	400	12:38:42 PM	00:48:50	04:00:00	default	script
188741	🟩	EnergyFEC_3	400	11:36:55 AM	01:50:37	04:00:00	default	script

Interactive job

Useful for short tests or debugging

Submit the job with –I (letter I for Interactive)

Example:

- qsub –I –n 32 –t 30 –q cache-quad –A Myprojname

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

Reservations

Reservations allow exclusive use of a set of nodes for a specified group of users for a specific period of time

- a reservation prevents other users' jobs from running on that resource

- often used for system maintenance or debugging
- R.pm (preventive maintenance), R.hw* or R.sw* (addressing HW or SW issues)
- maintenance reservations appear idle

Requesting
- See: http://www.alcf.anl.gov/user-guides/reservations
- Email reservation requests to support@alcf.anl.gov

When things go wrong…

Examine core files using 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)

Retain all job information

- Jobid, copy/location of all files (*.cobaltlog, *.error, *.output), exact error message

Contact us

- Your ALCF contact
- Email: support@alcf.anl.gov
- Call the ALCF Help Desk
  - Hours: Monday-Friday, 9am-5pm CT
  - Phone: 630-252-3111 or 866-508-9181 (toll-free,US only)

HAPPY COMPUTING!