ALCF Getting Started Videoconference January 2013

Yuri Alexeev

Graham Fletcher

Marta García

Ray Loy

Tim Williams

And the ALCF team

Agenda

- Blue Gene/P hardware overview
- Building your code
- Considerations before you run
- Queuing and Running
- After your job is submitted
- Potential problems
- Performance Tuning
- Backups and Tape Archival
- Getting Help

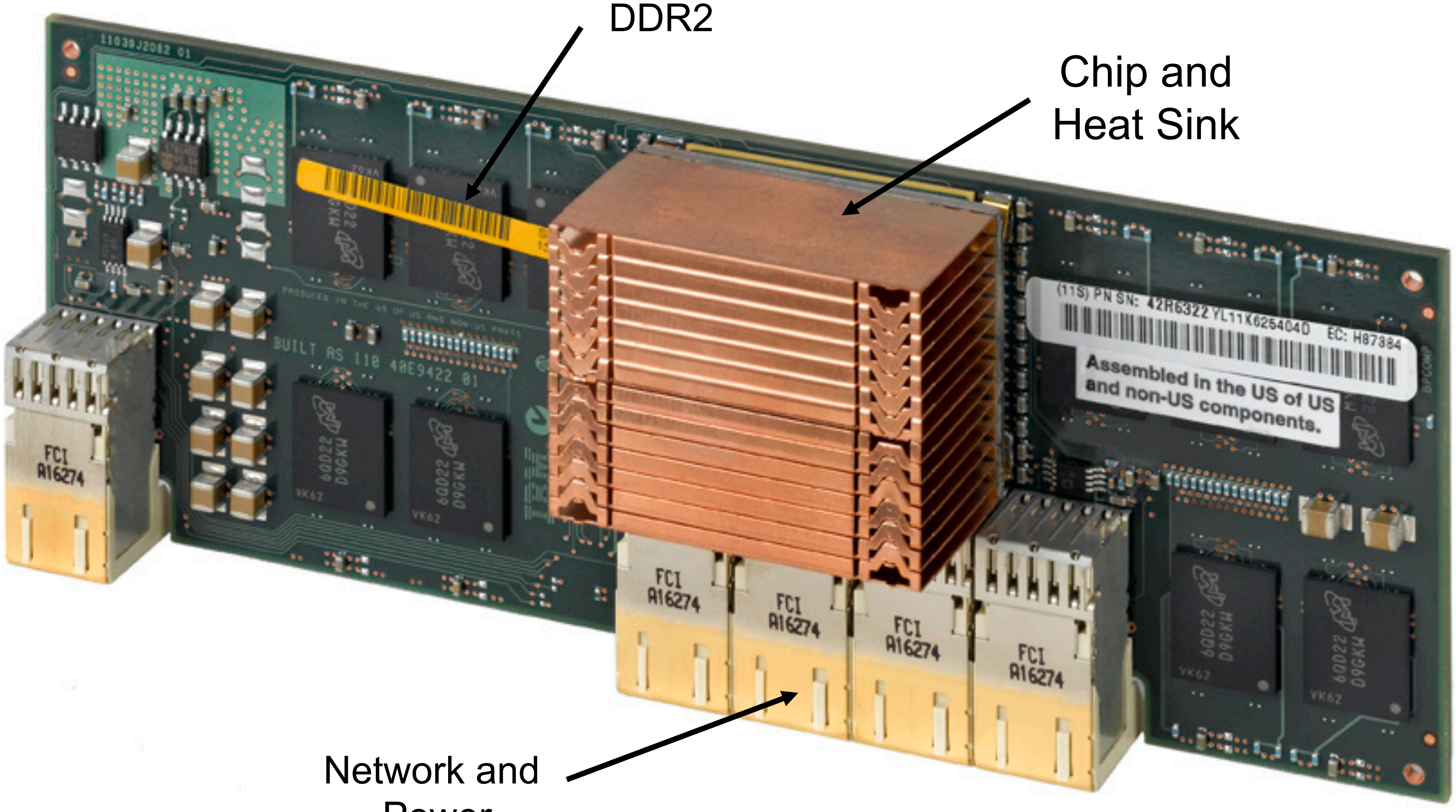
Section:

Blue Gene/P hardware overview

Chip: PowerPC 450 Processor

- A branch of PowerPC 440 Processor
- Dual-issue single-threaded embedded 32 bit processor @ 850 MHz
- Single integer unit, single load/store unit, special double FPU
- Three execution pipes and a two-way F-pipe
  - complex integer I-pipe for arithmetic, logic, and system management
  - simple integer J-pipe for arithmetic and logic instructions
  - L-pipe for loads, stores, and cache management
- Double FPU supports
  - standard PowerPC instructions (executed on fpu0)
  - SIMD instructions for 64-bit fp-numbers (fpadd, fpmul, fpmadd, fpre, …)
  - FP pipeline latency 5 cycles (fadd,fmadd,fpmadd)
- L1 cache: 32KB+32KB, 32 Byte line size, coherent across cores
- L2 cache: prefetch buffer with 16 128-byte lines (2KB)

Blue Gene/P Compute Card


(11S) PN SN: 42R6322 YL11K625404D EC: H87384
Assembled in the US of US and non-US components.
BUILT AS 110 40E9422 01
FCI A16274
FCI A16274
FCI A16274
FCI A16274
FCI A16274

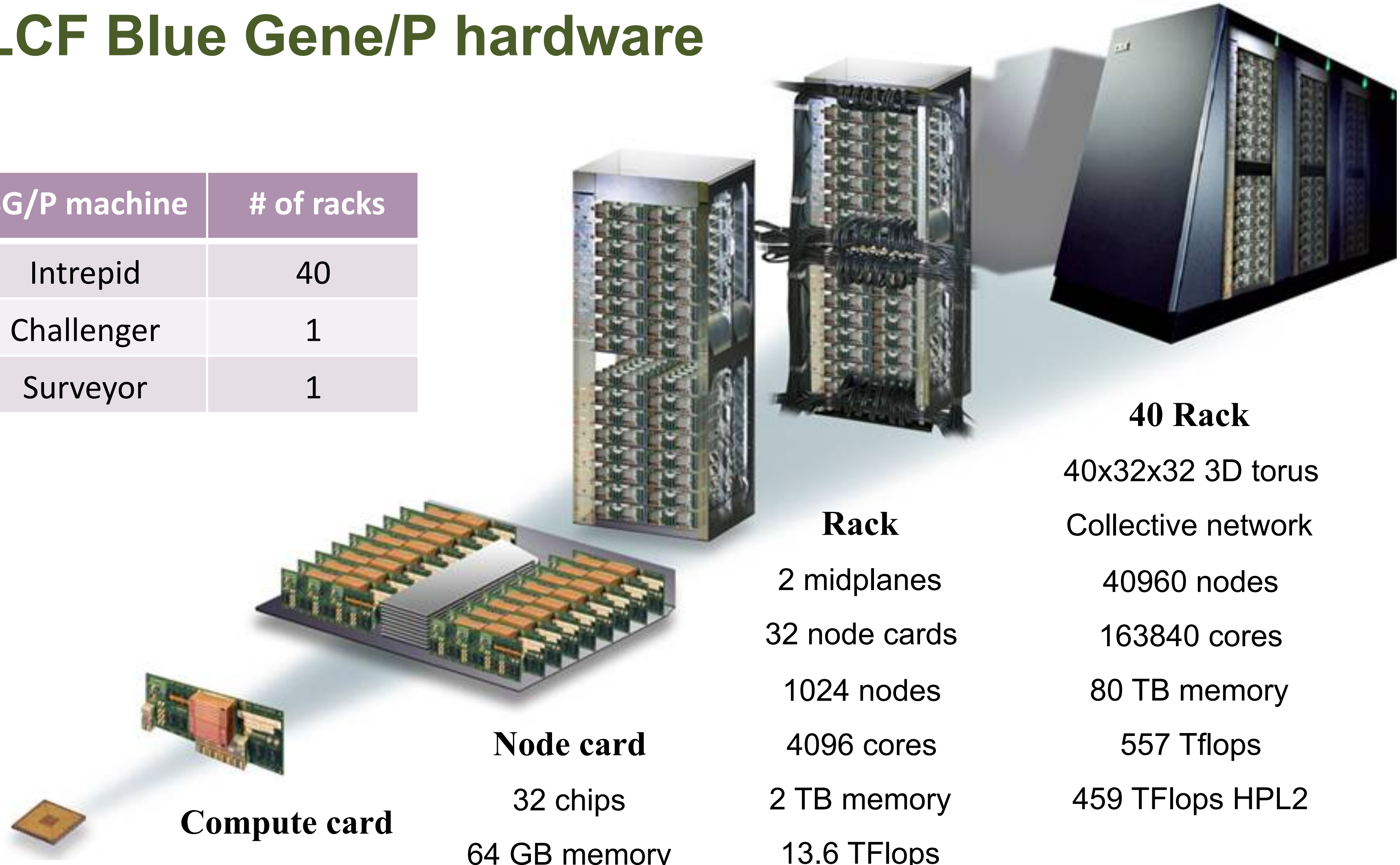
DDR2

Chip and Heat Sink

Network and Power

ALCF Blue Gene/P hardware

BG/P machine	# of racks
Intrepid	40
Challenger	1
Surveyor	1



Chip

4xPPC450 cores

Compute card

4 cores

2 GB memory

13.6 GFlops

Node card

32 chips

64 GB memory

435 GFlops

Rack

2 midplanes

32 node cards

1024 nodes

4096 cores

2 TB memory

13.6 TFlops

40 Rack

40x32x32 3D torus

Collective network

40960 nodes

163840 cores

80 TB memory

557 Tflops

459 TFlops HPL2

ALCF Resources connected to BG/P

INCITE

Intrepid

557 TF
40K 850 MHz nodes
160K cores
80TB RAM

Challenger

13.9 TF
1K 850 MHz nodes
4K cores
2TB RAM

Eureka

111TF (SP)
(100) 2.0 GHz nodes
800 Cores
3.2 TB RAM
200 Nvidia FX5600 GPUs

USER TEAMS
(via ESnet, Internet2, MREN)

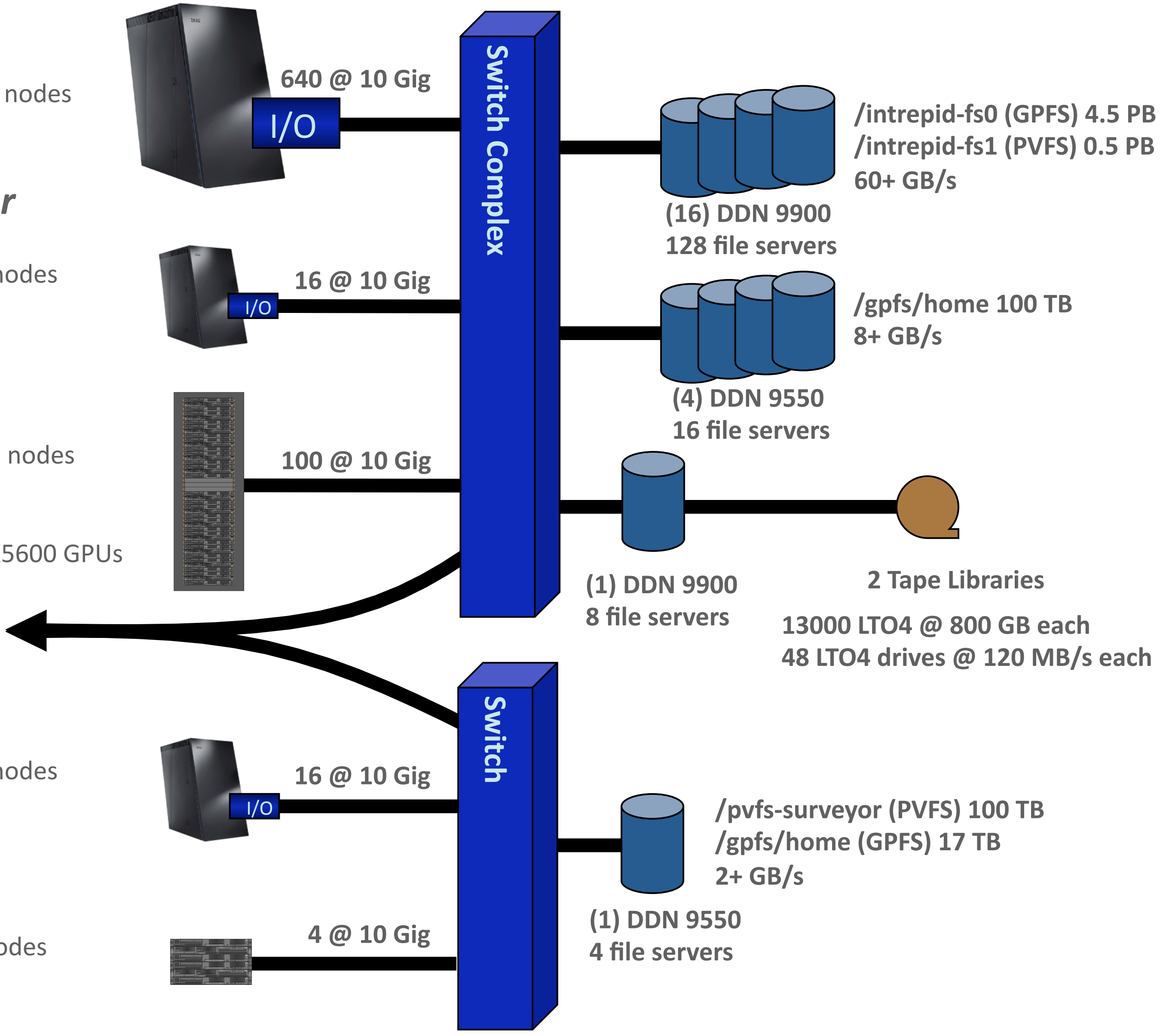
T&D

Surveyor

13.9 TF
1K 850 MHz nodes
4K cores
2TB RAM

Gadzooks

4.4 TF (SP)
(4) 2.0 GHz nodes
32 Cores
128 GB RAM
8 Nvidia FX5600 GPUS



/intrepid-fs0 (GPFS) 4.5 PB
/intrepid-fs1 (PVFS) 0.5 PB
60+ GB/s

(16) DDN 9900
128 file servers

/gpfs/home 100 TB
8+ GB/s

(4) DDN 9550
16 file servers

(1) DDN 9900
8 file servers

2 Tape Libraries

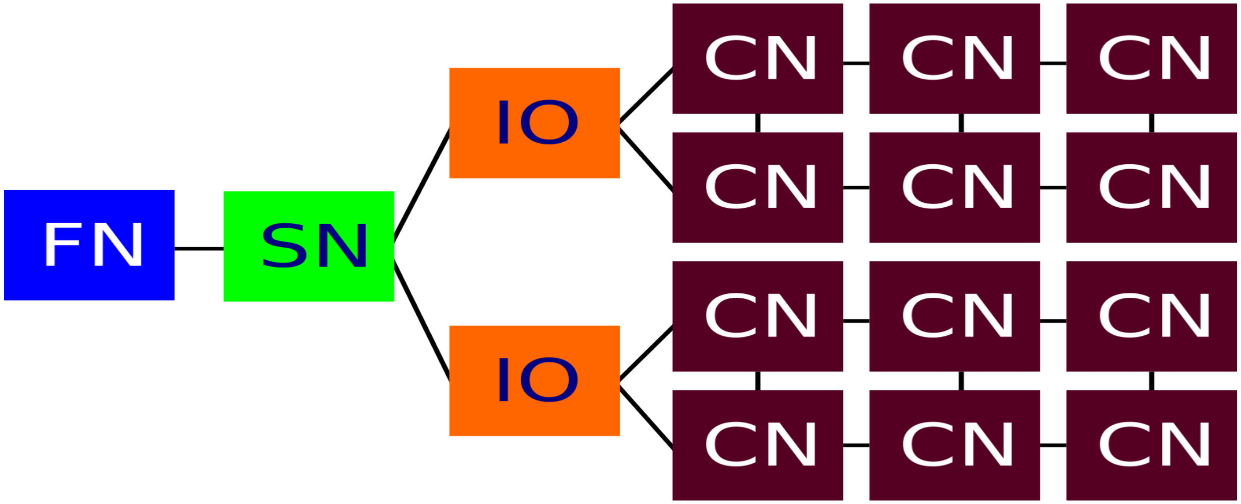
13000 LTO4 @ 800 GB each
48 LTO4 drives @ 120 MB/s each

/pvfs-surveyor (PVFS) 100 TB
/gpfs/home (GPFS) 17 TB
2+ GB/s

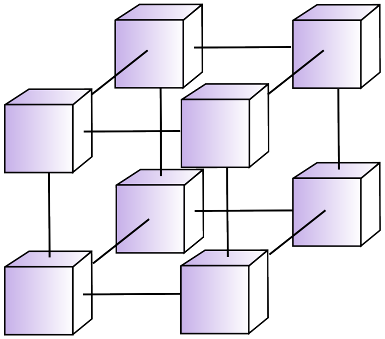
(1) DDN 9550
4 file servers

Blue Gene/P Hierarchical Organization

- Front-end nodes – dedicated for user's to login, compile programs, submit jobs, query job status, debug applications. **Standard Linux OS.**
- Compute nodes – run user applications, use simple **compute node kernel (CNK)** operating system, ships I/O-related system calls to I/O nodes.
- I/O nodes – provide a number of Linux/Unix typical services, such as files, sockets, process launching, signals, debugging; run Linux.
- Service nodes – perform partitioning, monitoring, synchronization and other system management services. Users do not run on service nodes directly.

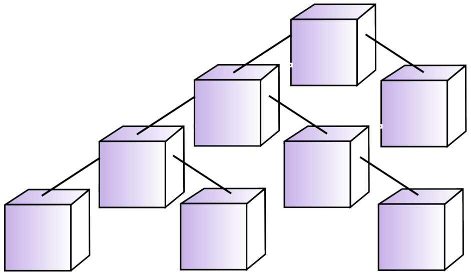


Interconnect Networks



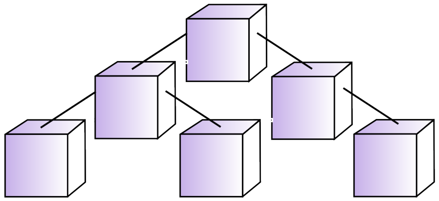
3-D Torus

- Basis for point-to-point communications
- Connects all compute nodes
- Supports virtual cut-through hardware routing
- 3.4 Gb/s on all 12 links (5.1GB/s per node)
- Hardware latency: 0.5 µs per hop, 5 µs farthest link
- MPI latency: 3 µs per hop, 10 µs farthest link



Global Collective Network

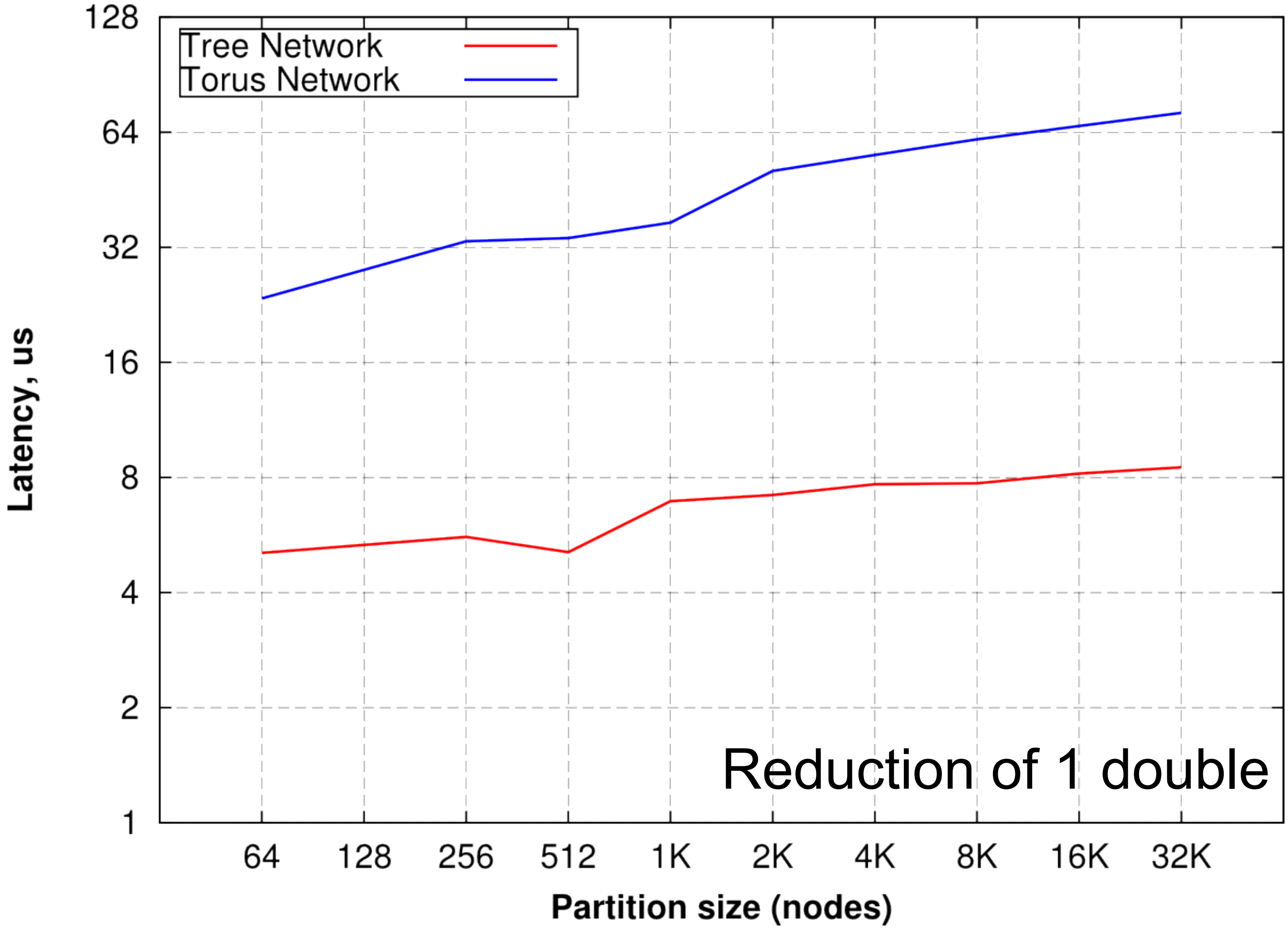
- Tree topology
- Basis for collective and I/O communications
- Connects all compute and I/O nodes
- Supports integer and double reductions
- 6.8 Gb/s of bandwidth per link per direction
- Hardware latency: 1.3 µs per tree traversal
- MPI latency: 5 µs per tree traversal



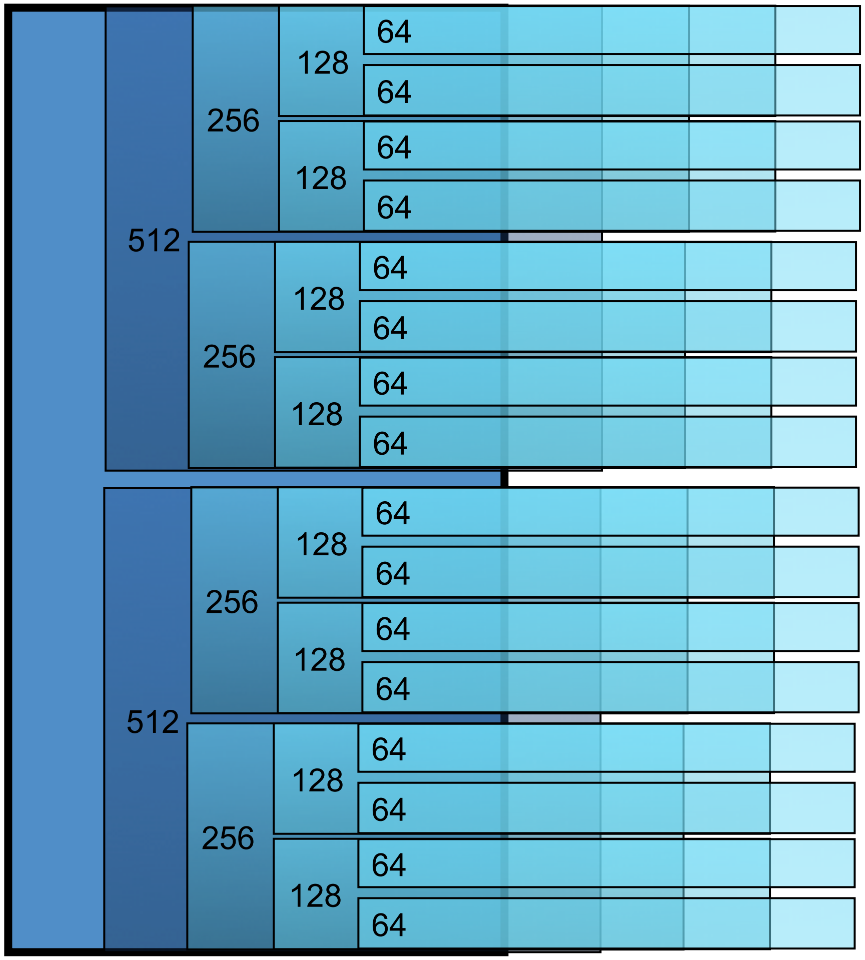
Global Barrier and Interrupt Network

- Hardware latency: 0.65 µs
- MPI latency: 1.6 µs

Collective interconnect performance



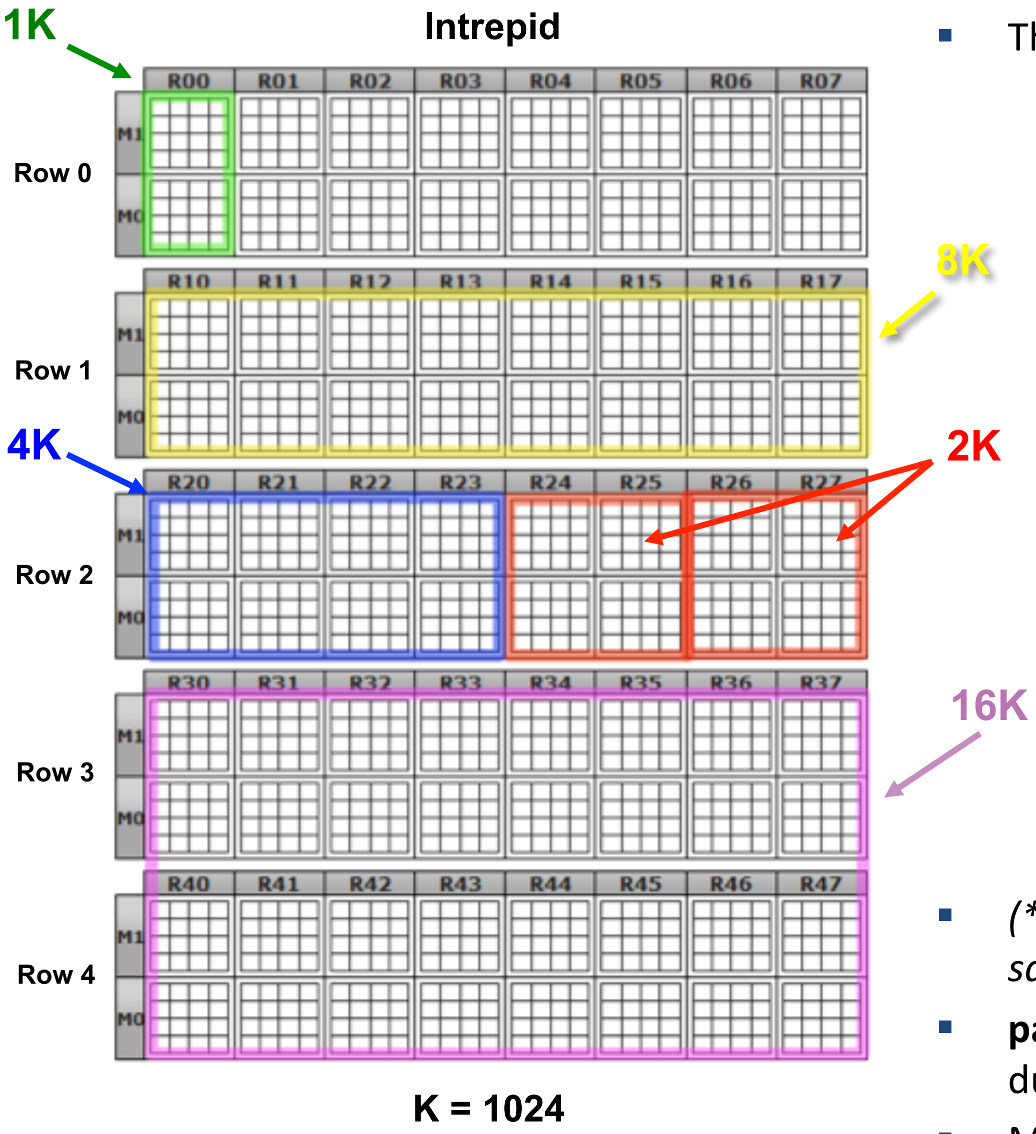
Blue Gene/P Single Rack Partitions (“blocks”)



- I/O node to compute node ratio of 1:64 on Intrepid and a ratio of 1:16 on Challenger.
- Partition sizes: 16*, 32*, 64, 128, 256, 512, 1024
  - *Any partition < 512 nodes will get a mesh network layout and not a torus.*
  - *Any partition <512 nodes will get a non-optimal I/O tree network.*
  - *Do not do performance testing on <512 nodes*
- Smaller partitions are enclosed inside of larger ones
  - *Not all partitions are available at all times*
  - *Once a job is running on one of the smaller partitions, no jobs can run on the enclosing larger partitions*
- Configuration changes frequently
  - ***bg-listblocks --all*** lists all defined partitions
    - E.g. ANL-R00-M0-N00-64
  - ***partlist*** shows partition state

* 16 and 32 nodes partitions only available on Challenger

Blue Gene/P Multiple Rack Partitions (“blocks”)


1K
Intrepid
R00 R01 R02 R03 R04 R05 R06 R07
R10 R11 R12 R13 R14 R15 R16 R17
R20 R21 R22 R23 R24 R25 R26 R27
R30 R31 R32 R33 R34 R35 R36 R37
R40 R41 R42 R43 R44 R45 R46 R47
M1
M0
Row 0
Row 1
Row 2
Row 3
Row 4
8K
4K
2K
16K
K = 1024

- The number of large block sizes possible are:

# of nodes	Possible #
40960	1
32768	1
24576	1
16384	2
8192	5
4096	5 (*)
2048	20
1024	20 (*)
512	80

- *(*) Not all possible blocks are available at the same time due to wiring dependencies.*
- **partlist** will show you if a large free block is busy due to a wiring dependency.
- Mesh partitions are available by reservation only.

Partition Dimensions

Challenger

Nodes	X	Y	Z	Torus
16	4	2	2	No
32	4	4	2	No
64	4	4	4	No
128	4	4	8	No
256	8	4	8	No
512	8	8	8	Yes
1024	8	8	16	Yes

Intrepid

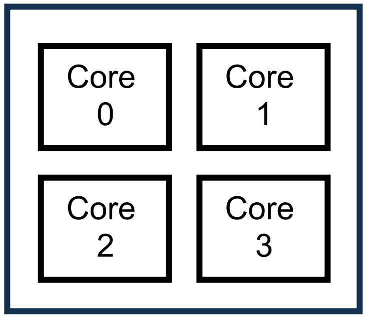
Nodes	X	Y	Z	Torus
512	8	8	8	Yes
1024	8	8	16	Yes
2048	8	8	32	Yes
4096	8	16	32	Yes
8192	8	32	32	Yes
16384	16	32	32	Yes
24576	24	32	32	Yes
32768	32	32	32	Yes
40960	40	32	32	Yes

<X,Y,Z,T> coordinates describe the location of a process within the torus network.

T being the core number

http://www.alcf.anl.gov/resource-guides/internal-networks/torus

Execution Modes in BG/P



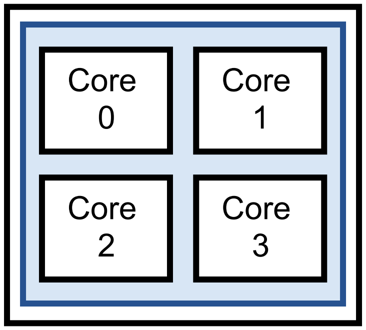
- Hardware elements (black)
- Software Abstractions (blue)
- qsub … --mode smp/dual/vn (default smp)

SMP Mode

1 Process (MPI rank)

1-4 Threads/Process

2 GB/Process

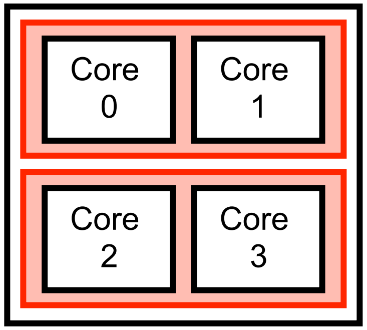


Dual Mode

2 Processes (MPI ranks)

1-2 Threads/Process

1 GB/Process

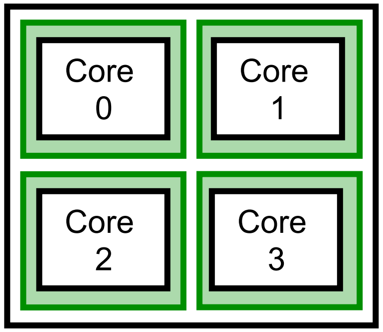


VN Mode

4 Processes (MPI ranks)

1 Thread/Process

512 MB/Process



Questions?

Section:

Building your code

Softenv

- A tool for managing a user' s environment
  - Sets your PATH to access desired front-end tools
  - *Your compiler version can be changed here*
- Settings:
  - Maintained in the file ~/.softenvrc
  - Add/remove keywords from ~/.softenvrc to change environment
  - ***Make sure @default is at the very end***
- Commands:
  - softenv
    - a list of all keywords defined on the systems
  - resoft
    - reloads initial environment from ~/.softenvrc file
  - soft add|remove keyword
    - Temporarily modify environment by adding/removing keywords

http://www.mcs.anl.gov/hs/software/systems/softenv/softenv-intro.html

Use Compiler Wrappers

- MPI wrappers for IBM XL cross-compilers:

Wrapper	Thread-Safe Wrapper	Underlying Compiler	Description
mpixlc	mpixlc_r	bgxlc	IBM BG C Compiler
mpixlcxx	mpixlcxx_r	bgxlC	IBM BG C++ Compiler
mpixlf77	mpixlf77_r	bgxlf	IBM BG Fortran 77 Compiler
mpixlf90	mpixlf90_r	bgxlf90	IBM BG Fortran 90 Compiler
mpixlf95	mpixlf95_r	bgxlf95	IBM BG Fortran 95 Compiler
mpixlf2003	mpixlf2003_r	bgxlf2003	IBM BG Fortran 2003 Compiler

- MPI wrappers for GNU cross-compilers:

Wrapper	Underlying Compiler	Description
mpicc	powerpc-bgp-linux-gcc	GNU BG C Compiler
mpicxx	powerpc-bgp-linux-g++	GNU BG C++ Compiler
mpif77	powerpc-bgp-linux-gfortran	GNU BG Fortran 77 Compiler
mpif90	powerpc-bgp-linux-gfortran	GNU BG Fortran 90 Compiler

- “-show” option: shows complete command used to invoke compiler

ex: mpixlc –show sum.c

```
/opt/ibmcmp/vacpp/bg/9.0/bin/bgxlc sum.c -I/bgsys/drivers/V1R3M0_460_2008-081112P/ppc/comm/default/include -I/bgsys/drivers/
V1R3M0_460_2008-081112P/ppc/comm/sys/include -L/bgsys/drivers/V1R3M0_460_2008-081112P/ppc/comm/default/lib -Wl,-rpath,/bgsys/
drivers/V1R3M0_460_2008-081112P/ppc/comm/default/lib -lmpich.cnk -L/bgsys/drivers/V1R3M0_460_2008-081112P/ppc/comm/sys/lib -
Wl,-rpath,/bgsys/drivers/V1R3M0_460_2008-081112P/ppc/comm/sys/lib -ldcmfcoll.cnk -ldcmf.cnk -lpthread -L/bgsys/drivers/
V1R3M0_460_2008-081112P/ppc/runtime/SPI -Wl,-rpath,/bgsys/drivers/V1R3M0_460_2008-081112P/ppc/runtime/SPI -lSPI.cna -lrt
```

IBM XL Optimization Settings Options

Level	Implies	Description
-O0	-qstrict -qfloat=nofltint:norsqrt:rngchk -qstrict_induction	Minimal optimization, preserves program semantics, best for debugging
-O2 (*or -O*)	-qstrict -qfloat=nofltint:norsqrt:rngchk -qnostrict_induction -qmaxmem=8192	Preserves program semantics, eliminates redundant code, basic loop optimization
-O3	-qnostrict -qfloat=fltint:rsqrt:norngchk -qnostrict_induction -qmaxmem=-1 -qhot=level=0	High order loop analysis and transformations, better loop scheduling, inlining, in depth memory access analysis, *can alter program semantics*
-O4	*All -O3 options plus* -qhot=level=1 -qhot=vector -qipa=level=1	Additional loop analysis, basic interprocedural optimization, *can alter program semantics*
-O5	*All -O4 options plus* -qipa=level=2	Advanced interprocedural analysis, *can alter program semantics*

Hierarchy of Optimization Levels

- Suggested set of optimization levels from least to most optimization:
  - -O0 # best level for use with a debugger
  - -O2 # good level for verifying correctness, baseline perf
  - -O2 -qmaxmem=-1 -qhot=level=0
  - -O3 -qstrict (preserves program semantics)
  - -O3
  - -O3 -qhot=level=1
  - -O4
  - -O5
- Tips:
  - -qlistopt generates a listing with all flags used in compilation
  - -qreport produces a listing, shows how code was optimized
  - Performance can decrease at higher levels of optimization, especially at -O4 or -O5
  - May specify different optimization levels for different routines/files

Sample BG/P makefile

```
CC = mpixlc
CXX = mpixlcxx
FC = mpixlf90
OPTFLAGS = -O3
CFLAGS = $(OPTFLAGS) -qlist -qsource -qreport -g
FFLAGS = $(OPTFLAGS) -qlist -qsource -qreport -g

myprog: myprog.c
      $(CC) $(CFLAGS) -o myprog myprog.c
```

Threading

- OpenMP is supported
  - IBM XL compilers: -qsmp=omp
  - GNU: add softenv key +gcc-4.3.2-gomp
- pthreads is supported
  - NPTL pthreads implementation in glibc requires no modifications
- Compiler auto thread parallelization is available
  - use -qsmp=auto
  - not always effective
- The job mode will determine maximum total number of threads (including the master thread)
  - smp=4, dual=2, vn=1
  - Maximum one thread per core, no oversubscription, no thread scheduling
  - All possible threads need not be used (but cores will be idle)

OpenMP

- Shared-memory parallelism is supported on single node
- Hybrid programming model
  - MPI at outer level, across compute nodes
  - OpenMP at inner level, within a compute node
- **Thread-safe compiler version should be used** (mpixlc_r etc.) with any threaded application (either OMP or pthreads)
- OpenMP 2.5 standard directives are supported:
  - parallel, for, parallel for, sections, parallel sections, critical, single
  - #pragma omp <rest of pragma> for C/C++
  - !$OMP <rest of directive> for Fortran
- Compiler functions
  - omp_get_num_procs, omp_get_num_threads
    omp_get_thread_num, omp_set_num_threads
- Number of OpenMP threads
  - set using environment variable OMP_NUM_THREADS
  - must be exported to the compute nodes using qsub --env flag (note 2 dashes)

Software Libraries

- ALCF Supports two sets of libraries:
  - IBM system and provided libraries: /bgsys/drivers/ppcfloor
    - glibc
    - mpi
    - DCMF (Deep Computing Messaging Framework)
    - SPI (System Programming Interface)
    - UPC (Universal Performance Counters)
    - BG/P Personality
  - Site supported libraries and programs: /soft/apps/current
    - PETSc
    - FFTW
    - HDF5
    - *And many others (see also http://www.alcf.anl.gov/resource-guides/software-and-libraries)*

Questions?

Section:

Considerations before you run

Transferring Data To/From ALCF

- **sftp** and **scp (**for "small" transfers)
  - If you must use scp, eureka is a better system to scp to/from. All paths will be the same as they are on Intrepid.
  - Eureka is also a better host for compressing and uncompressing large file archives
- **GridFTP** (for large transfers)
  - Other site must accept our CA
  - ssh / cryptocard access available
- **Globus Online** (for large transfers)
  - *Globus Online* addresses the challenges faced by researchers in moving, sharing, and archiving large volumes of data among distributed sites.
  - ALCF BG/P endpoints: alcf#dtn_intrepid, alcf#dtn_surveyor, alcf#dtn
  - Check if your laboratory, university or research center has already an endpoint.
  - *Globus Connect* to transfer files to and from your local machine.

globus online

http://www.alcf.anl.gov/resource-guides/data-transfer

Table of BG/P File Systems on ALCF Resources

System	Type	Path	Production	Backed up	Visible to BG/P Jobs	Uses
Surveyor	GPFS	/home	Yes	No	Yes	General use
Surveyor	PVFS	/pvfs-surveyor	Yes	No	Yes	Storage, large file I/O, high performance I/O
Intrepid	GPFS	/home	Yes	Yes	Yes	General use
Intrepid	GPFS	/intrepid-fs0	Yes	No	Yes	Storage, large file I/O, high performance I/O
Intrepid	PVFS	/intrepid-fs1	Yes	No	Yes (*)	Storage, large file I/O, high performance I/O
Surveyor/ Intrepid	Local	/scratch	No	No	No	Storage space local to the login machine that should be reasonably fast and will allow you to store large files on a temporary basis

(*) /intrepid-fs1 requires using the '--kernel pvfs' option on qsub in order to be visible by BG/P jobs.

Allocation Management

- Every user must have at least one Project they are assigned to.
  - Use ‘projects’ command to query.
- Projects are then given allocations
  - Allocations have an amount, start, and end date and are tracked separately; Charges will cross allocations automatically. The allocation with the earliest end date will be charged first, until it runs out, then the next, and so on
- Charges are based on the partition size, NOT the # of nodes or cores used!
- Reservations are charged for the full time they are active
- Use ‘cbank’ command to query allocation, balance
  - cbank -l charge -p <projectname>   # list all charges against a particular project
  - cbank -l allocation -p <projectname> # list all active allocations for a particular project
  - Other useful options:
    - -u <user> : show info for specific user(s)
    - -a <YYYY-MM-DD> : show info after date (inclusive)
    - -b <YYYY-MM-DD> : show info before date (exclusive)
    - --help
  - http://www.alcf.anl.gov/resource-guides/query-allocations-cbank

Questions?

Section:

Queuing and Running

Cobalt resource manager and job scheduler

- Cobalt is used on all ALCF systems
  - *Similar to PBS but not the same*

- Job management commands:

  qsub: submit a job

  qstat: query a job status

  qdel: delete a job

  qalter: alter batched job parameters

  qmove: move job to different queue

  qhold: place queued (non-running) job on hold

  qrls: release hold on job

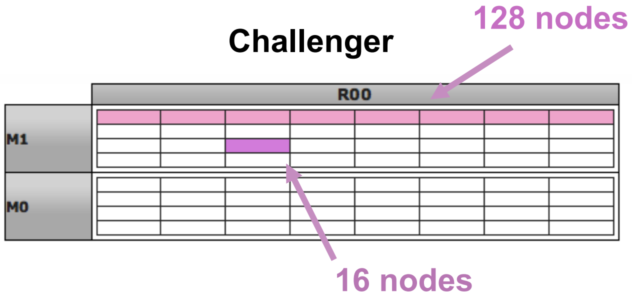
  showres: show current and future reservations

Challenger/Intrepid Queues

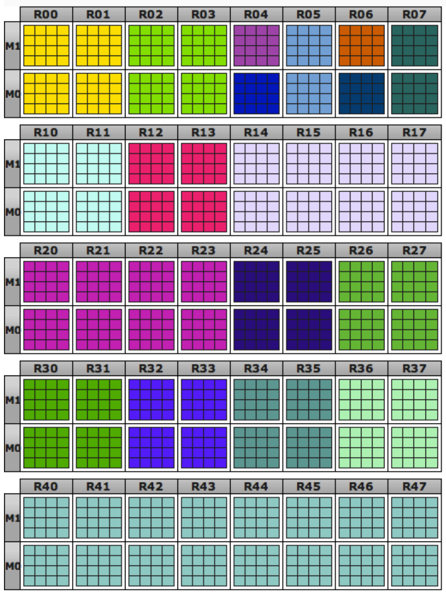
- “**prod-devel**” queue (Challenger)
  - For testing and debugging
  - Partition sizes: 16 – 512 nodes (in powers of 2)
  - Time limit: 1 hour
  - Max of 20 submitted jobs and 5 running jobs
  - Priority is given to small, short jobs

- “**prod**” queue (Intrepid)
  - For production compute jobs
  - Partition sizes: 512 – 32768 nodes (in powers of 2)
  - Time limit: 12 hours
  - Max of 20 submitted jobs and 5 running jobs
  - Priority is given to large jobs

Challenger



Intrepid

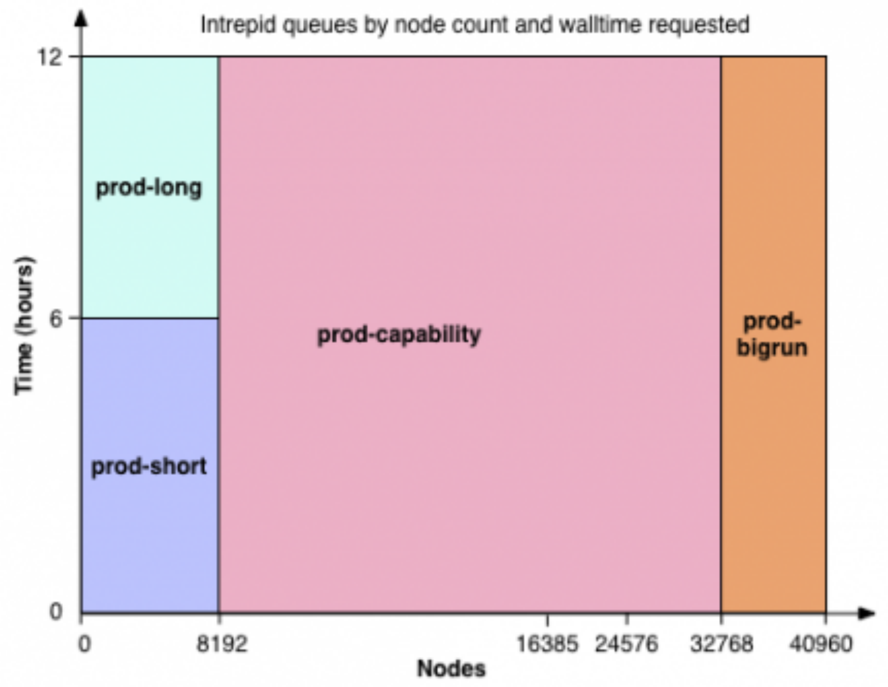

R00 R01 R02 R03 R04 R05 R06 R07
R10 R11 R12 R13 R14 R15 R16 R17
R20 R21 R22 R23 R24 R25 R26 R27
R30 R31 R32 R33 R34 R35 R36 R37
R40 R41 R42 R43 R44 R45 R46 R47
M1
M0

- Other special queues exist for mostly administrative purposes and are not generally available for running jobs (see all with qstat –Q)

Intrepid Job Scheduling

- **Big Run Monday**
  Every Monday, any jobs in the 'prod-capability' queue in the 'queued' state will be promoted to highest priority.

- **Restrictions in queues**
  - 'prod-long' restricted to rows 0 & 1.
  - 'prod-short', 'prod-capability', 'prod-bigrun' can run in the full machine.



http://www.alcf.anl.gov/resource-guides/job-scheduling-policy

User Queue	Underlying Queue	Nodes	Wall-clock Time (hours)	Maximum Jobs Per User	Maximum Jobs Per Project
prod	prod-short	512 - 4096	0 - ≤6	5	20
	prod-long	512 - 4096	>6 - 12	5	20
	prod-capability	4097 - 32768	0 - 12	2	2
	prod-bigrun	32769 - 40960	0 - 12	1	1
	backfill (*)	512 - 8192	0 - 6	5	10 (per user)

(*) This queue is automatically selected if a project's allocation is negative.

Intrepid Job boot times

- Each time a job is submitted using a standard qsub command all of the nodes in a partition are rebooted
- Boot times depend on the size of the partition

Nodes in Partition	Boot time (seconds)
512	80
1024	86
2048	105
4096	166
8192	256
16384	351
24576	532
32768	712

qsub options

- Syntax:

```
qsub [-d] [-v] -A <project name> -q <queue> --cwd <working directory>
          --env envvar1=value1:envvar2=value2 --kernel <kernel profile>
          -K <kernel options> -O <outputprefix> -t time <in minutes>
          -e <error file path> -o <output file path> -i <input file path>
          -n <number of nodes> -h --proccount <processor count>
          --mode <mode> -M <email> --dependencies <jobid1>:<jobid2> <command> <args>
```

- Standard options:

-A project	project to charge
-q queue	queue
-t <time_in_minutes>	required runtime
-n <number_of_nodes>	number of nodes
--proccount <number_of_cores>	number of CPUs
--mode <smp|dual|vn>	running mode
--env VAR1=1:VAR2=1	environment variables
<command> <args>	command with arguments
-O <output_file_prefix>	prefix for output files (default jobid)
-M <email_address>	e-mail notification of job start, end
--dependencies <jobid1>:<jobid2>	set the dependencies for the job being submitted

qsub: Examples of submitting a job

- Despite being redundant, we recommend to always specify the number of nodes, the number of processes (MPI ranks), and the mode of your run

- qsub -q prod-devel -t 10 -n 64 --proccount 64 --mode smp Hello
  - submits a job to a short queue
  - will run no longer than 10 minutes or when executable stops
  - will use smp-mode with 64 nodes, 64 CPUs

- qsub -q prod-devel -t 10 -n 4 --proccount 16 --mode vn -O My_Run My_Exe My_File
  - submits a job to a short queue and run no longer than 10 minutes
  - will use vn-mode with 4 nodes, 16 CPUs
  - will run program My_Exe with argument My_File
  - will create My_Run.output as stdout and My_Run.error as stderr files

Methods of submitting a job

- Directly submit an executable (no pre-processing or post-processing)
  - Run *qsub* from the command line (*not recommended*)

    ```
    qsub -q prod-devel -t 10 -n 64 --proccount 64 --mode smp Hello
    ```

  - Place the qsub command within a shell script

    ```
    #!/bin/bash
    # can do preprocessing here
      qsub -q prod-devel -t 10 -n 64 --proccount 64 --mode smp Hello
    ```

  - Note qsub is non-blocking so cannot do post-processing here

- Submit a job script to Cobalt

  ```
  #!/bin/bash
  qsub -q prod-devel -t 10 -n 64 --proccount 64 --mode script job.sh
  ```

  - Job script run by the scheduler only after the job starts
  - Job script runs on a dedicated node similar to the login node
  - Allows for pre-processing and post-processing at the end of the job
  - Call cobalt-mpirun in your script (example follows)

Cobalt Script Mode Job

- Sample script job.sh:

```
#!/bin/sh
echo "Starting Cobalt job script”
# Do pre-processing work here
…
# Run executable (important- do not use plain ‘mpirun’)
cobalt-mpirun –mode vn –np $NODES –cwd `pwd` -env “FOO=1 BAR=2” myprog1.exe args
# Do post-processing work here
…
```

- Submit using

```
#!/bin/bash
 qsub –A myproj -q prod-devel -t 10 -n 64 --proccount 64 --mode script job.sh
```

- Use cobalt-mpirun inside script, not ‘mpirun’ or ‘qsub’
- cobalt-mpirun blocks until run is complete
- The job is not complete until the script exits (you are charged for total time)

Advanced runs using script mode

- Multiple (consecutive) runs in a single job
- Multiple simultaneous runs in a single job
- Combinations of the above
- See: *http://www.alcf.anl.gov/resource-guides/running-jobs#advanced-job-patterns-using-scripts*

Questions?

Section:

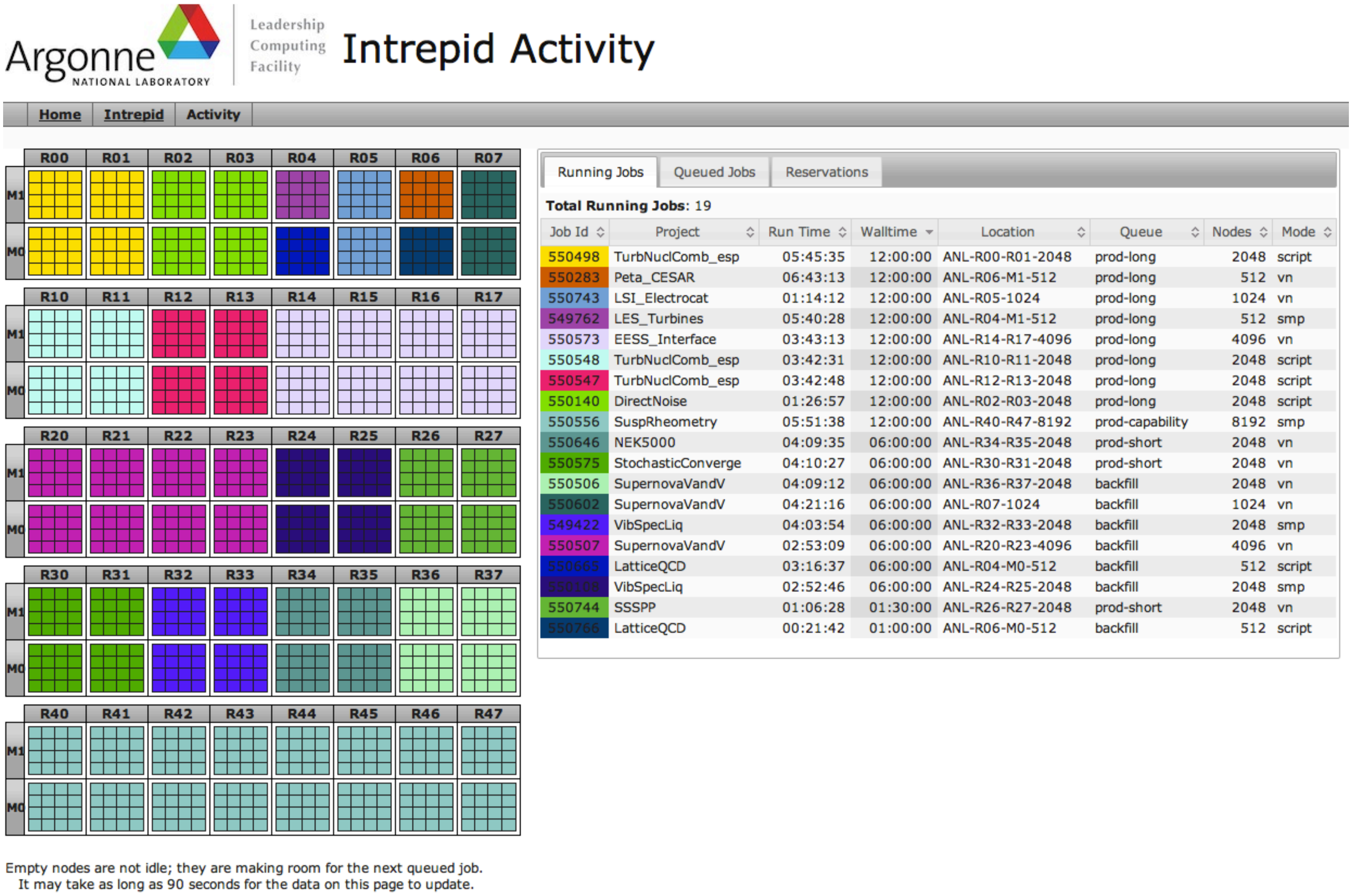
After your job is submitted

qstat: Show Status of a Batch Job(s)

- qstat # list all jobs

```
 JobID   User    WallTime  Nodes  State    Location
========================================================
301295  smith   00:10:00   16       queued  None
```

- About jobs
  - JobID is needed to kill the job or alter the job parameters
  - Common states: queued, running, user_hold, maxrun_hold
- qstat –f <jobid> # show more job details
- qstat -fl <jobid> # show all job details
- qstat -Q
  - instead of jobs, this shows information about the queues
  - will show all available queues and their limits
  - includes special queues, which we use to handle reservations



Argonne NATIONAL LABORATORY
Leadership Computing Facility
Home
Intrepid
Activity
R00 R01 R02 R03 R04 R05 R06 R07
R10 R11 R12 R13 R14 R15 R16 R17
R20 R21 R22 R23 R24 R25 R26 R27
R30 R31 R32 R33 R34 R35 R36 R37
R40 R41 R42 R43 R44 R45 R46 R47
M1
M0
Running Jobs
Queued Jobs
Reservations

Intrepid Activity

Total Running Jobs: 19

Job Id	Project	Run Time	Walltime	Location	Queue	Nodes	Mode
550498	TurbNuclComb_esp	05:45:35	12:00:00	ANL-R00-R01-2048	prod-long	2048	script
550283	Peta_CESAR	06:43:13	12:00:00	ANL-R06-M1-512	prod-long	512	vn
550743	LSI_Electrocat	01:14:12	12:00:00	ANL-R05-1024	prod-long	1024	vn
549762	LES_Turbines	05:40:28	12:00:00	ANL-R04-M1-512	prod-long	512	smp
550573	EESS_Interface	03:43:13	12:00:00	ANL-R14-R17-4096	prod-long	4096	vn
550548	TurbNuclComb_esp	03:42:31	12:00:00	ANL-R10-R11-2048	prod-long	2048	script
550547	TurbNuclComb_esp	03:42:48	12:00:00	ANL-R12-R13-2048	prod-long	2048	script
550140	DirectNoise	01:26:57	12:00:00	ANL-R02-R03-2048	prod-long	2048	script
550556	SuspRheometry	05:51:38	12:00:00	ANL-R40-R47-8192	prod-capability	8192	smp
550646	NEK5000	04:09:35	06:00:00	ANL-R34-R35-2048	prod-short	2048	vn
550575	StochasticConverge	04:10:27	06:00:00	ANL-R30-R31-2048	prod-short	2048	vn
550506	SupernovaVandV	04:09:12	06:00:00	ANL-R36-R37-2048	backfill	2048	vn
550602	SupernovaVandV	04:21:16	06:00:00	ANL-R07-1024	backfill	1024	vn
549422	VibSpecLiq	04:03:54	06:00:00	ANL-R32-R33-2048	backfill	2048	smp
550507	SupernovaVandV	02:53:09	06:00:00	ANL-R20-R23-4096	backfill	4096	vn
550665	LatticeQCD	03:16:37	06:00:00	ANL-R04-M0-512	backfill	512	script
550108	VibSpecLiq	02:52:46	06:00:00	ANL-R24-R25-2048	backfill	2048	smp
550744	SSSPP	01:06:28	01:30:00	ANL-R26-R27-2048	prod-short	2048	vn
550766	LatticeQCD	00:21:42	01:00:00	ANL-R06-M0-512	backfill	512	script

Empty nodes are not idle; they are making room for the next queued job.
It may take as long as 90 seconds for the data on this page to update.

http://status.alcf.anl.gov/intrepid/activity (beta, a.ka. The Gronkulator)

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"
- Cobalt log file: **<prefix>.cobaltlog**
  - first file created by Cobalt after a job is submitted
  - contains submission information from qsub command, mpirun, 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

qdel: Kill a Job

- qdel <jobid1> <jobid2>
  - delete the job from a queue
  - terminated a running job

qalter, qmove: Alter Parameters of a Job

- Allows to alter the parameters of queued jobs without resubmitting
  - *Most parameters may only be changed before the run starts*
- Type qalter to see

  Usage: qalter [-d] [-v] -A <project name> -t <time in minutes>

  -e <error file path> -o <output file path>

  -n <number of nodes> -h --proccount <processor count>

  -M <email address> --mode <mode smp/dual/vn> <jobid1> <jobid2>
- qalter cannot change the queue; use qmove instead
  - qmove <destination_queue> <jobid>

Holding and Releasing

- qhold - Hold a submitted job (will not run until released)

  qhold <jobid1> <jobid2>

- To submit directly into the hold state, use qsub -h
- qrls - Release a held job (in the *user_hold* state)

  qrls <jobid1> <jobid2>

- Jobs in the dep_hold state may be released by removing the dependency

  qalter --dependencies none <jobid>

- Jobs in the *admin_hold* state may only be released by a system administrator

Possibilities why a job is not running yet

- there is a reservation, which interferes with your job
  - showres shows all reservations currently in place
- There are no available partitions
  - partlist shows all partitions marked as functional
  - partlist shows the assignment of each partition to a queue

```
Name                Queue           State
=================================================================
ANL-R00-1024        default:spruce  blocked (ANL-R00-M0-N00-256)
ANL-R00-M0-512      default:spruce  blocked (ANL-R00-M0-N00-256)
ANL-R00-M1-512      default:spruce  idle
ANL-R00-M0-N00-256  default:spruce  busy
```

- wrong queue
  - the job submitted to a queue, which is restricted to run at this time

Optimizing for queue throughput

- Target prod-short
  - I.e. Small (<8K) jobs <= 6h
- Shotgun approach
  - If your code is amenable, submit a mix of job sizes and lengths
- Check for drain windows
  - qavail <partition_size>
  - E.g. qavail 2048

```
Name             State  Backfill  busy_for
==============================================
ANL-R20-R21-2048  idle   0:23      None
ANL-R24-R25-2048   idle   0:23      None
```

In this case, a job submitted for 2048 nodes can run immediately if its time is < 23 minutes.

Questions?

Section:

Potential problems

When things go wrong… Logging in

- Check to make sure it's not maintenance
  - Often login nodes on both BG/P and data analytics systems are closed off during maintenance to allow for activities that would impact users
  - There should be a mention in the bi-weekly maintenance announcement and the pre-login banner message
  - An all-clear will be sent out at the close of maintenance

- Remember that CRYPTOCard passwords
  - Require a pin at the start
  - Are all all hexadecimal characters (0-9, A-F). *Letters are all* ***UPPER CASE***.

- On failed login, try in this order:
  - Just try typing PIN+password again (without generating new password).
  - Try a different ALCF host to rule out login node issues (e.g. maintenance)
  - Push cryptocard button to generate new password and try that
  - Walk through the unlock and resync steps at: http://www.alcf.anl.gov/resource-guides/using-cryptocards#troubleshooting-your-cryptocard
  - Still can't login in? Connect with **ssh –vvv** and record the output, your IP address, hostname, and the time that you attempted to connect. Send this information in your e-mail to support@alcf.anl.gov

When things go wrong… running

- Cobalt jobs, by default, produce three files (.cobaltlog, .error, .output)
- Only .cobaltlog is generated at submit time, the others at runtime
- At boot, the .error file will have a non-zero size
  - Most of the messages are related to booting, look here to follow startup progress
  - *Note: If your script job redirects the stderr of cobalt-mpirun, it will not end up in the job's .error file*
- If you think there is an issue, it's best to save all three files
  - Send the jobid, and a copy of the files to support

When things go wrong… running

- You'll see RAS events appear in your .error file it's not always the sign of trouble
  - RAS stands for Reliability, Availability, and Serviceability

- Few are a sign of a serious issue, most are system noise
  - Messages have a severity associated with them
    - INFO
    - WARN
    - ERROR
    - FATAL
  - Only **FATAL** RAS events will lead to the termination of your application
    - ***ERROR may degrade performance but do NOT kill your job.***
  - Still worth watching as they may be the sign of an application performance issue

- If you run exits abnormally, the system will list the last RAS event encountered in the run. ***This RAS event did not necessarily cause the run to die.***

Core Files

- Jobs experiencing fatal errors will general produce a core file for each process
- Examining core files:
  - Core files are in text format, readable with the "`more`" command
  - bgp_stack command provides call stack trace from a core file:
    - `Ex: bgp_stack <executable> <corefile>`
    - Command line interface
    - Can only examine one core file at a time
  - coreprocessor command provides call stack trace from multiple cores
    - Ex: `coreprocessor`
    - GUI interface requires X11 forwarding (ssh -X intrepid.alcf.anl.gov)
    - Provides information from multiple core files
- Environment variables control core dump behavior:
  - BG_COREDUMPONEXIT: core dump when application exits
  - BG_COREDUMPDISABLED: disable core dumps

Can’t run what you need? Reservations

- Reservations allow exclusive use of a partition for a specified group of users for a specific period of time
  - A reservation blocks other users jobs from running on that partition
  - Often used for system maintenance or debugging
    - **R.pm** (preventative maintenance), **R.hw*** or **R.sw*** (addressing HW or SW issues)
  - Reservations are sometimes idle, but still block other users jobs from running on a partition
  - Should be the exception not the rule

- Requesting
  - See: http://www.alcf.anl.gov/resource-guides/reservations
  - Email reservation requests to ***support@alcf.anl.gov***
  - View reservations with **showres**
  - Release reservations with **userres**

- When working with others in a reservation, these qsub options are useful:
  - **--run_users <user1>:<user2>:…**   All users in this list can control this job
  - **--run_project <projectname>**   All users in this project can control this job

Questions?

Section:

Performance Tuning

Tools: Improved Performance, Profiling, Debugging …

- Most tools are under
  - /soft/apps/current (back-end libraries) or
  - /soft/apps/fen (front-end tools)
- Improved performance with optimized libraries
  - BLAS/LAPACK versus LibGOTO/LAPACK
  - BlueGene optimized Mass, MassV, ESSL libraries from IBM
- Practical Optimization
  - compiler switches
  - profiling and profiling tools: HPCT, Profiling "-pg", "-qdebug=function_trace", TAU
- Tracing MPI_Barrier/printf/exit/abort standard debugging methods
- GDB / Allinea DDT / Rogue Wave Totalview
  - Your choice of debuggers. *GDB not recommended for more than a single rank.*

MPI Mapping

- Default XYZT mapping
  - (XYZ) are torus coordinates, T is a CPU number
  - X-coordinate is increasing first, then Y, then Z
  - All XYZT permutations are possible
- qsub --env BG_MAPPING=TXYZ --mode vn …
  - This puts MPI task 0,1,2,3 to Node 0 CPU0, CPU1, CPU2, CPU3; MPI tasks 4,5,6, and 7 to Node2 CPU0,CPU1,CPU2,CPU3
  - Typically, default XYZT is less efficient than TXYZ mapping
- qsub --BG_MAPPING=<FileName> --mode smp …
  - use high-performance toolkits to determine communication pattern
  - optimize mapping by custom mapfile
  - mapfile: each line contains 4 coordinates to place the task, first line for task 0, second line for task 1…
  - avoid conflict in mapfiles (no verification)

Parallel I/O in HPC

- Applications want to achieve scalability, parallelism, high bandwidth, and usability
- Applications require more software than just a parallel file system
- Multiple layers are provided with distinct roles:
  - Parallel file system
    - maintains logical space, provides efficient access to data (PVFS, GPFS)
  - I/O forwarding
    - assists with I/O scaling issues, load balance for I/O servers
  - Middleware
    - organizes access by many processes (MPI-IO)
  - High-level I/O library
    - maps application abstractions to a structured portable data format (HDF5, Parallel netCDF)

Section:

Backups and Tape Archival

Backups and Tape Archival

- Backups
  - On-disk snapshots of /home directories are done nightly
    - Check ~/.snapshots if you delete a file accidentally
  - **Only home directories** are backed up to tape
  - *Data directories are not backed up*
    - E.g. /intrepid-fs0 and /intrepid-fs1

- Manual Data Archiving to Tape (**HPSS**)
  - HSI is an interactive client
  - Use HTAR for lots of small files
    - Path name is limited to 155 chars in the prefix and 100 bytes for the name (prefix/name)
    - File size is limited to 64 GB.
  - GridFTP access to HPSS is available
    - Should be significantly faster
  - See http://www.alcf.anl.gov/resource-guides/data-archive-hpss

Getting Help

Online resources:

- ALCF web pages:
  - http://www.alcf.anl.gov
  - http://www.alcf.anl.gov/resource-guides
- Intrepid Status: http://status.alcf.anl.gov/intrepid/activity
  (beta, a.k.a. The Gronkulator)

Contact:

e-mail: **support@alcf.anl.gov**

Help Desk: **630-252-3111** or **866-508-9181** (toll-free)

Your catalyst