Using LLNL's BlueGene/Q Systems (Sequoia, rzUSeq)

SLURM batch system [MOAB wrappers]

Useful Batch & Info Commands (from logins)

sbatch/msub *scriptfile* - submit batch script job
checkjob [-v] *job-id* - see detailed job state/info
mshow [-a|-j [*job-id*] - partition/job(s) status
mdiag -j|-b|-f|-L|-p|-t|-c [-v] - batch resource info
showstart *job-id*|*procs@secs* - when can job run
showstate - sys state   showstats [-q] - resources
showbf [-r *procs*|-n *nodes*|-d *secs*]-resource check
mjobctl *attr* is [wclimit|class|account|qos|depend]
pdsh - execute commands on remote nodes
scontrol show job - view SLURM configuration
salloc -N *nodes* -p pdebug - interactive allocation
mxterm *nodes* 0 *mins* -q pdebug - like sxterm

mjstat - jobs status & machine availability info pgrep - find job
smap -c -D[b|j] - see blocks|jobs usage [graphical view if no -c]
sinfo [-s|-l] - show partition info   news job.lim.? - limits
squeue [-l|-i #] - show job details [updated every # seconds]
showq [-b|-i|-r|-c] - see blocked|eligible|running|done jobs info
pstat [-A|-R|-f *job-id*|-m *host*] - your batch jobs status [+ details]
mjobctl -m *attr=val job-id* or palter -n *job-id -attr val* - alter job
mjobctl -h|-u *job-id* or phold|prel -n *job-id* - hold/release job
mjobctl -c *jobid* or canceljob *jobid* or prm -n *jobid* - remove job
ps -lu *userid*; kill -9 *pid* - force job quit   scancel - kill batch job
pkill *string* or killall *string* - kill processes matching string
mshare -p ALL -u *userid* - see resource allocation and usage
sxterm *nodes #MPITasks mins* -p pdebug - interactive xterm

Interactive Sessions & TotalView Usage

(Can't do much from the compute nodes since uses lightweight kernel)
To grab cpus in interactive xterm: sxterm *num_nodes #tasks minutes* [-p *pool <other_msub_args>*] {or mxterm}
or can run in current terminal: salloc [-N *num_nodes*] -n *#MPItasks* -p *pool*   [Must grab a full partition]
Then execute your program in parallel using srun: (for available partitions: smap -c -Db, use READY & slurm)

    srun [-N nodes] -n #MPItasks <executable> <args>   [nodes can be less than allocated num_nodes]
or totalview srun -a [-verbose 1] <all_other_srun_args_above>

pool = pool name [i.e. pbatch (or pdebug or others listed by the sinfo command)], for partition usage: smap -c -Db
#MPItasks = 1-6,291,456 (can't exceed 16 cores x *#HardwareThreads*[1-4] x *num_nodes*) [rzuseq 32,768]
Each core can employ 4 hardware threads (but has only 2 floating-point units) which can help with latency issues
All node allocations must be consecutive in the matrix (torus), and only whole partitions of them can be obtained
For batch jobs (use #msub inside a script, or instead these arguments can be given after msub submit command)

    #msub -l nodes=num_nodes[:ppn=#]   (#MPItasks is set on the srun execution line using its -n option)

num_nodes = a full partition [...|16384|18432|24576|27648|32768|36864|41472|49152|55296|65536|73728|98304]
procs_per_node# = tasks per node [16|32|64] there are 16 cores, each with up to 4 hardware threads available
For OpenMP programs with TotalView compiled with IBMs compiler, use options: -qsmp=omp:noauto:noopt
Other options: LLNL_TV_PATCH_SPACE=Add To attach to a running job, use totalview -pid *<srunPid>* srun

Batch Scripts (MOAB)

[submit with msub [*<additional_msub_args>*] *scriptfile*]

```
#!/bin/csh -f                  # Sets your shell (#MSUB -S <shell>, ksh uses export <var>=<val>)
#MSUB -V -j oe                 # Export env & join out+err; Also: -h [or -a hhmm] holds run [until time]
#MSUB -q pdebug                # Dont use -l partition=, list by smap -c -Db; also -l qos=[standby|expedite]
#MSUB -l nodes=#k[:ppn=#]      # Nodeblock [...|12k|16k|24k|32k|36k|48k|64k|72k|96k][:tasks/node=16|64]
#MSUB -l walltime=3600         # Set duration limit in seconds; 1 hour, or use format 01:00:00 or 00:60:00
#MSUB -A bdiv                  # Sets bank account to use (or bdev); (mdiag -u userid gives bank info [-a])
setenv OMP_NUM_THREADS 4  # For OpenMP threaded runs, set this env var
date ; cd /p/lscratchrza/collette  # Change into appropriate directory to execute code in & store output
srun [-N nodes] -n #MPITasks <exec> <args>
```

Compiling (from front-end)

- Cross-compile code from LAC login nodes (front-end) [objdump -d *file.o* - asmblr]
Use IBM's compilers mpixlc, mpixlcxx, or mpixlf90 (for OpenMP, use their _r threaded versions & -qsmp=omp)
Tuning: -qhot=novector -qsimd=auto [-qdebug=diagnostic for simd success] [-O3 -qfloat=maf -qstrict=precision]
Useful options: -qipa, -qhot, -O[0|2|3|4|5], -g[#], -qlist[opt], -qreport, -qsource, -qlistfmt=html, -qnostaticlink
Use -qtm [-qthreaded] for transactional memory; Use -qsmp=speculative for thread-level speculative execution
For GNU compilers: mpicc, mpicxx, or mpif90 [or /bgsys/drivers/ppcfloor/gnu-linux/powerpc64-bgq-linux/bin]
Available libraries: ESSL/BLAS/LAPACK/SCALAPACK/FFTW numerical libs are in /usr/local/tools/*<libname>*
MASS[V|_SIMD]: -L/opt/ibmcmp/xlmass/bg/7.3/bglib64 -lmass[v|_simd];
Largepages and 64-bit compilations are defaults, no -qarch/-qtune options necessary.
LNAD login nodes l8r with compute-node-like hardware for small compile & serial tests.

Summary sheet by
Mike Collette 12/2012

BGQ Environment Variables [options] [Desired settings are in **bold**] Summary by Mike Collette 12/2012

BG_SHAREDMEMSIZE=[32|**64**] MB for shared mem pool
BG_MAPPING=? 5D torus layout
BG_PERSISTMEMSIZE=[] MB for persistent mem pool
BG_PERSISTMEMRESET=[0|**1**] clear b4 run?
PAMID_VERBOSE=[0|**1**] For some execution debug info
BG_MAPALIGN16=[1] force TLB 16MB align
PAMID_CONTEXT_MAX=[1]
PAMID_THREAD_MULTIPLE=
BG_COREDUMPDISABLED=[0|**1**] allow cores
BG_COREDUMPONEXIT =[0|**1**] make cores
BG_COREDUMP_FILEPREFIX=*str* name of core
BG_COREDUMP_PATH=*dir* where cores go
PAMID_*<collective>*=various collective adjustments
BG_COREDUMPBINARY=[rank#s] pick cores
BG_LTMDISABLE=[ON|**OFF**] Livermore thread model
BG_THREADLAYOUT=[**1**|2] round-robin|fill
BG_MAPCOMMONHEAP=[0|**1**] bigger (but shared) task mem
BG_THREADMODEL=[0|**1**] 1 omp thrd/HWT
BG_SMP_FAST_WAKEUP=[**YES**] reduce nonthreaded impact
BG_MAPNOALIASES=[1] disables alias mode
OMP_WAIT_POLICY=[**ACTIVE**] faster thread performance
OMP_STACKSIZE=[16MB]
OMP_NUM_THREADS = OpenMP # of threads per MPI task
XLSMPOPTS = [stack=8000000] OMP stack sz
For threaded runs: set OMP_NUM_THREADS else unused hardware threads will NOT idle & may slow the code

Login Node Info
stat - display file status | ps -fu *<user>* - process status | pstree - display process tree
uname -a - machine name and info | quota -a [-v], du -k, df - disk usage | hostname - node name
cat /proc/meminfo - see memory info | cat /proc/cpuinfo - see CPU info | cat /proc/version - see OS info
netstat & vmstat [-P] - net & memory info | iostat, mpstat - see CPU & I/O info | uptime - duration and load
free - see used, swap, and free memory | id [*<user>*] - list UID's and GID's | watch - periodic cmnd execution
limit - resource limits (unlimit will max) | ypmatch & ypcat - see NIS values | w, who - show users and activity
pgrep - find processes by name/attributes | pkill, skill - send signal to a process | snice, renice - fix process priority
nslookup, dig, host - name server info | stty - set/show terminal settings | ac -p[d] - show user's usage time
showrgb, xlsfonts, fc-list - colors & fonts | xfd -fn *<font>* - see font characters | groups [*<user>*] - list groups
printenv, env - see environment variables | top, xload, tload - load level | last - see previous users

Document Viewers
man, apropos, info, whatis - command info | evince, gs - view .ps & .pdf files
[s|vim]diff, diff3, cmp, comm - diff files | pod2man, perldoc - view .pod docs | zcmp|zdiff - diff compressed files
more, less, cat, head, tail, vi, emacs - text | whereis, which - command location | readelf - view ELF files

File Handlers
cpio - copy to archive | pftp - parallel ftp | strip - remove exe's symbol table
nl, wc - list with line numbers / count lines | g[un]zip - make/expand .[g]z files, or | tar -[c|x]f - groups files together
sort, uniq - sort / filter out repeated lines | basename, dirname - strip off suffixes | rsync - faster transfers than rcp
join - combine lines with common field | sed/awk - stream editor/pattern scan | tr - translate/squeeze/delete chars
expand - convert tabs to spaces | hsi - like ftp but faster | htar - for HPSS archive tar files

Directory Spaces
/p/ls1/*<user>* - parll IO | /g/g##/*<user>* - home dirs | /tmp/*<user>* - login-local dirs
/p/lscratchrza/*<user>* - parallel IO space | /[usr|var]/tmp - (same) temp space | /usr/local/docs - documentation

Debug & Optimization Tools (link/load tools - ar, nm, ldd, ldconfig) tracing: STATview, mpitrace
corefile debuging: gdb, addr2line, core_stack_merge, locate_rank
TotalView GUI: totalview srun -a *<sargs>* *<exec>* CLI: tvcli *<exec>* *<core>* [then 'dwhere' traceback cmnd]
profiling compiler options: -p, -pg profiling tools: gprof, cprof, mpiP memory tools: memP
hardware performance & others: PAPI (use v5), TAU, valgrind, Open|SpeedShop timing: time

Architecture of Sequoia (rzuseq has 512 nodes) 6,291,456 MPItasks
98,304 IBM BlueGene/Q nodes, each with a 16-core 1.6MHz PowerPC A2 64-bit processor
1,572,864 total cores with 4 hardware threads & 2-FPUs each; Memory/node:16GB, Total:1.57PB
Peak 20PFlops; Byte order: Big-Endian (high-order byte is lowest address). CNK liteweight kernel
5D-torus interconnect, Lustre&SLURM. Has 4 login front-end/Power7/24core/4GHz/64GB nodes
Nodes-16cores w/4threads/core; Supports SIMD, Transactional Memory & Speculative Execution
LAC login nodes: 24-core 3.7MHz Power7 64-bit processors w/64GB mem (front-end)

Help/Info lc-hotline@llnl.gov (925) 422-4531 http[s]://[www|lc].llnl.gov/computing
Docs /usr/local/docs/[rzuseq.basics] https://lc.llnl.gov/confluence/display/BGQ

M Collette 12/2012
collette1@llnl.gov