Using Moab on LC's Aztec and Inca Clusters

The installation of Moab on Aztec and Inca is configured to schedule processors. On the Moab-scheduled, high-speed interconnected machines so far, Moab schedules whole nodes to the job, and at most one job could run on a node. On Aztec and Inca, multiple jobs can be scheduled on one node. Users will request a processor count for their job, and Moab will launch the job on a node with that many idle processors. Users will be able to ask for 1 to 12 processors for Aztec or Inca.

Users will use a new `msub -l` option, `ttc` (total task count) to specify the number of processors for the job, e.g., `msub -l ttc=4`. Jobs with `ttc` values in excess of 12 will be indefinitely deferred. psub wrapper users will use the `psub -np` option to specify the number of processors required.

Requesting a node count (either `msub -l nodes=<count>` or `psub -ln <count>`) will be irrelevant. However, if the msub or psub request includes a node request for more than one node, the (msub or psub) request will be rejected when invoked.

Another difference is that Aztec and Inca will be running the SLURM resource manager (not LCRM's own, TBS). While existing job scripts that invoke the user's codes will work fine, users who run MPI jobs will see the changes described below.

Aztec Time Limits

The wall-clock time limit of jobs running on Aztec will be 200 hours for all Aztec nodes. Under LCRM, there were shorter time limits for certain Aztec nodes.

LCRM had no means for dedicating a processor to a machine. Consequently, the `psub -tM` parameter allowed one to set a time limit on the number of cycles actually delivered to the job. Under Moab, all the requested processors will be exclusively allocated to the job; therefore, `msub -l walltime=<value>` should be used to specify the wall-clock time limit for the job. The equivalent psub wrapper option is `-tW`.

Memory on Aztec Nodes

All Aztec nodes have 48 GB of memory.

MPI on Aztec and Inca with Moab

With the switch to Moab on Aztec and Inca, SLURM will be used to launch MPI jobs. Although **mpirun** will continue to function, LC recommends that you use **srun** in place of mpirun. SLURM and the srun command provide a number of benefits over mpirun, for example:

1. SLURM runs the job script within a SLURM allocation, which provides improved control over processor binding and process cleanup. Both of these features will help ensure jobs get full access to the resources they were allocated.

2. You may now launch your MPI job as you do on other LC systems like Zeus or Atlas, e.g., you can use:

   ```
   srun -n 8 ...
   ```

   in place of

   ```
   mpirun -np 8 ...
   ```

3. You don't want (or need) to use `-A` or `-N` srun options. The job script starts inside of its own allocation, so `-A` is unnecessary. Because Aztec and Inca do not have an interconnect switch, a single job cannot run on more than one node, so `-N` is unnecessary.

Debugging

For debugging MPICH-based MPI executables with TotalView, the proper invocation of TotalView is still

```
mpirun -np <process count> -tv <executable>
```

For memory debugging, use the command below (all on one line; csh shell assumed):

```
setenv TOTALVIEW "totalview -e 'dheap -enable'"
```

Then run mpirun.

With the switch to CHAOS 4, shared-memory MVAPICH 0.9.9, and Moab on Aztec and Inca, TotalView invocation on MPI executables will be the same as on the parallel clusters, except that you are limited to one node:

```
totalview srun -a -n8 -ppdebug -l <code> <code args>
```

Use `mtv` for memory debugging:

```
mtv srun -a -n8 -ppdebug -l <code> <code args>
```

Use `mxterm` to launch a batch xterm:

```
mxterm 1 8 30 -q pdebug
```

Multiprocess Jobs Not Using MPI

If your job is a multiprocess job but does not use MPI, you will need to use the `--mpi=none` flag with srun. For example:

```
srun --mpi=none -n8 hostname
```

Using the New msub/#MSUB -l ttc= Option

Because the Aztec and Inca clusters do not have a switch, job scheduling is done by processor requirements instead of node requirements. The new Moab msub/#MSUB `-l ttc=` option is now required when you submit jobs to these clusters (where `ttc` stands for "total task count" and refers to the number of processors your job will require).

It is important that you match the new `-l ttc=` option with appropriate srun (or mpirun) options. The table below shows examples using srun. Note that the same will hold true if you are using `mpirun -np` instead of `srun -n`.

`#MSUB -l ttc=8` `srun -n8 a.out`	Correct. Uses 8 processes on 8 processors.

`#MSUB -l ttc=4` `srun -n8 a.out`	Incorrect. Uses 8 processes on only 4 processors.
`#MSUB -l ttc=4` `srun -n8 -O a.out`	This will work because the -O option permits SLURM to oversubscribe a node's processors. However, using more tasks than processors is not recommended and can degrade performance.
`#MSUB -l ttc=16` `srun -n16 a.out`	Incorrect. ttc is more than the physical number of processors on a node.
# (no specification for -l ttc) `srun -n2 a.out`	Incorrect. The #MSUB -l ttc= option isn't specified, so the default of 1 is less than required by the -n2 tasks specified with srun.
`#MSUB -l ttc=8` `srun -n2 a.out`	This is OK and might be used if each of 2 tasks spawns 4 threads.
`#MSUB -l ttc=8` `srun -n8 --mpi=none hostname`	Correct way to launch a non-MPI, multiprocess job.
`#MSUB -l ttc=8` `srun -n8 hostname`	Incorrect way to launch a non-MPI, multiprocess job. It is missing the --mpi=none flag.

For additional information about using Moab in general, including this new option, please see the Moab tutorial available online at https://computing.llnl.gov/tutorials/moab/.

December 10, 2010
LLNL-MI-401783

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.