

Porting to Columbia: With MPI and OpenMP

Category: Porting to Columbia

Columbia Phase Out:

As of Feb. 27, 2013, the Columbia21, Columbia23, and Columbia24 nodes have been taken offline as part of the [Columbia phase out process](#). Columbia22 is still available. If your script requires a specific node, please make the appropriate changes in order to ensure the success of your job.

Building Applications

To build a hybrid MPI+OpenMP application, you need to compile your code with the `-openmp` compiler flag and link in both the Intel OpenMP and the SGI MPT library:

```
%ifort -o your_executable -openmp program.f -lmpi
```

Running Applications

Process/thread placement is critical to the performance of MPI+OpenMP hybrid codes. Two environment variables should be set to get the proper placement:

MPI_DSM_DISTRIBUTE

Activates NUMA job placement mode. This mode ensures that each MPI process gets a unique CPU and physical memory on the node with which that CPU is associated. Currently, the CPUs are chosen by simply starting at relative CPU 0 and incrementing until all MPI processes have been forked.

MPI_OPENMP_INTEROP

Setting this variable modifies the placement of MPI processes to better accommodate the OpenMP threads associated with each process. For this variable to take effect, you must also set **MPI_DSM_DISTRIBUTE**.

Also note that **OMP_NUM_THREADS** is set to 1 by default for PBS jobs. Reset it to the number of threads that you want.

Here is a sample PBS script for running MPI+OpenMP hybrid (two MPI processes, four OpenMP threads per MPI process) applications on Columbia:

```
#PBS -lncpus=8,walltime=1:00:00

setenv MPI_DSM_DISTRIBUTE
setenv MPI_OPENMP_INTEROP
setenv OMP_NUM_THREADS 4

cd $PBS_O_WORKDIR

mpirun -np 2 ./your_executable
```

Article ID: 109

Last updated: 27 Feb, 2013

Computing at NAS -> Porting & Developing Applications -> Porting to Columbia -> Porting to Columbia: With MPI and OpenMP

<http://www.nas.nasa.gov/hecc/support/kb/entry/109/?ajax=1>