With MVAPICH

Category: Porting to Pleiades

DRAFT

This article is being reviewed for completeness and technical accuracy.

On Pleiades, there are multiple modules of MVAPICH2 built with either gcc or Intel compilers.

```
mpi-mvapich2/1.2p1/gcc
mpi-mvapich2/1.2p1/intel
mpi-mvapich2/1.2p1/intel-PIC
mpi-mvapich2/1.4.1/gcc
mpi-mvapich2/1.4.1/intel
```

You can get more information of what options were used to build each module as follows:

- Load the desired MVAPICH2 module

  %module load mpi-mvapich2/1.4.1/intel

- Use the *mpiname* utility provided with the module

```
%mpiname -a
MVAPICH2 1.4.1 2010-03-12 ch3:mrail

Compilation
CC: icc -fpic -DNDEBUG -O2
CXX: icpc  -DNDEBUG -O2
F77: ifort -fpic -DNDEBUG -O2
F90: ifort  -DNDEBUG -O2

Configuration
--prefix=/nasa/mvapich2/1.4.1/intel.sles11 --enable-f77 --enable-f90 --enable-cxx
```

Building Applications

Here is an example of how to build an MPI application with MVAPICH2:

```
%module load mpi-mvapich2/1.4.1/intel
%module load comp-intel/11.1.072
%mpif90 program.f90
```

Running Applications

To run your job, submit your job through PBS. Within the PBS script, there are two ways to run MPI applications built with MVAPICH2.

1.
```
#PBS ..
...
module load mpi-mvapich2/1.4.1/intel
module load comp-intel/11.1.072

mpiexec -np TOTAL_CPUS your_executable
```

2.
```
#PBS ..
...
module load mpi-mvapich2/1.4.1/intel
module load comp-intel/11.1.072

mpirun_rsh -np TOTAL_CPUS -hostfile $PBS_NODEFILE your_executable
```

Performance Issues

To pin processes, the MVAPICH library uses the environment variable VIADEV_USE_AFFINITY, which does something similar to SGI's MPI_DSM_DISTRIBUTE. By default, VIADEV_USE_AFFINITY is set to 1.

If you wish to pin processes explicitly, beware that with MVAPICH, 1 shepherd process is created for each MPI process. You can use the command

```
/u/scicon/tools/bin/qsh.pl jobid \
  'ps -C executable -L -opsr,pid,ppid,lwp,time,comm'
```

to see these processes of your running job. To properly pin MPI processes using *dplace*, one cannot skip the shepherd processes. In addition, knowledge of the processor labeling in each processor type is essential when you use *dplace*. Below are the recommended ways of pinning an 8 MPI process job with every 4 processes on 4 processors of a node:

- Harpertown

```
mpiexec -np 8 dplace -c2,3,6,7 ./your_executable
```

- Nehalem-EP

```
mpiexec -np 8 dplace -c2,3,4,5 ./your_executable
```

- Westmere-EP

```
mpiexec -np 8 dplace -c4,5,6,7 ./your_executable
```

Further information about pinning can be found here.

For more descriptions including the MVAPICH User Guide and other MVAPICH publications, see http://mvapich.cse.ohio-state.edu.

Article ID: 101
Last updated: 28 Oct, 2011
Computing at NAS -> Porting & Developing Applications -> Porting to Pleiades -> With MVAPICH
http://www.nas.nasa.gov/hecc/support/kb/entry/101/?ajax=1