AMBER 11 with Intel Cluster Ready

HPC ADVISORY COUNCIL NETWORK OF EXPERTISE BEST PRACTICES

1. Introduction

The following best practices document is provided as courtesy of the HPC Advisory Council.

2. Application Description:

AMBER is a popular software application for analyzing large-scale molecular dynamics (MD) simulation trajectory data. AMBER Reads either CHARMM or AMBER style topology/trajectory files as input, and its analysis routines can scale up to thousands of compute cores or hundreds of GPU nodes with either parallel or UNIX file I/O. AMBER has dynamic memory management, and each code execution can perform a variety of different structural, energetic, and file manipulation operations on a single MD trajectory at once. The code is written in a combination of Fortan90 and C, and its GPU kernels are written with NVIDIA's CUDA API to achieve maximum GPU performance.

3. Version Information:

Download AMBER 11 and AmberTools 1.5 at:

http://ambermd.org

http://ambermd.org/AmberTools-get.html

Download the list of benchmarks for Amber from:

http://ambermd.org/amber11_bench_files/Amber11_Benchmark_Suite.tar.gz

4. Prerequisites:

The instructions from this best practice have been tested with the following configuration:

4.1 Hardware:

- Dell PowerEdge M610 38-node cluster
- Intel Xeon X5670 CPUs @ 2.93 MHz
- Memory: 24GB per node @ 1333MHz
- Mellanox ConnectX-2 QDR InfiniBand Adapters
- Mellanox QDR InfiniBand Switch

4.2 Software:

- Intel® Cluster Ready running RHEL 5.5
- Mellanox OFED 1.5.2 InfiniBand Software Stack
- Application: AMBER
- Compilers: Intel compilers
- MPI: Intel MPI 4, Open MPI 1.5.3 with KNEM 0.9.6, Platform MPI 8.1.1
- Benchmark workload: primates.nex

5. Building AMBER

Extract Amber and Ambertools:

```
$ mkdir ~/amber
$ cd ~/amber
$ tar xvfj ~/Amber11.tar.bz2
$ tar xvfj ~/AmberTools-1.4.tar.bz2
$ cd amber11
$ wget http://ambermd.org/bugfixes/11.0/bugfix.all
$ patch -p0 -N < bugfix.all
$ cd ~/amber/amber11/AmberTools/src
#
# use one of the following three:
#
$ ./configure -mpi gnu    # for GNU Compilers
$ ./configure -mpi intel  # for Intel Compiler
$ ./configure -mpi pgi    # for PGI Compilers


#
# If you are using Intel Compilers with Intel MPI, please make sure the
# following variables are set in config.h, after the configure has been run
#
# CC=mpiicc
# FC=mpiifort
# PMEMD_F90=mpiifort
# PMEMD_CC=mpiicc
# PMEMD_LD=mpiifort
#


$ cd ../../src/
# Modify the config.h as mentioned above if you are
```

```
using Intel MPI
# with Intel Compilers
# vim config.h
$ make clean
$ make parallel
```

6. Building AMBER

Change to the directory with the dataset to run the job:

```
$ cd ~/amber/Amber11_Benchmark_Suite/PME/Facto-
rIX_production_NVE_128_64_64
```

a. Running with Intel MPI

```
$ mpdboot –parallel-startup -r ssh -f <PATH_TO_
HOSTFILE> -n 16

$ mpiexec -ppn 12 -np 192 -IB ~/amber/amber11/bin/
pmemd.cuda.MPI -O -i mdin

$ mpdallexit
```

b. Running with Platform MPI

```
$ mpirun -np 192 -IBV -prot -hostfile <PATH_TO_
HOSTFILE> ~/amber/amber11/bin/pmemd.cuda.MPI
-O -i mdin
```

c. Running with Open MPI

```
$ mpirun -np 16 -mca btl self,sm,openib -hostfile
<PATH_TO_HOSTFILE> < <AMBER_HOME>/mb
<AMBER_HOME>/input
```

d. Running with MVAPICH2

```
$ mpiexec.hydra -hosts <HOSTLIST> -np 192 /amber/
amber11/bin/pmemd.cuda.MPI -O -i mdin
```