

# NAMD Best Practices for Intel® Cluster Ready



BEST PRACTICES

## 1. Introduction:

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

## 2. Application Description:

NAMD is a parallel molecular dynamics code designed for high-performance simulation of large biomolecular systems. Based on Charm++ parallel objects, NAMD scales to hundreds of processors on high-end parallel platforms and tens of processors on commodity clusters using gigabit ethernet. NAMD uses the popular molecular graphics program VMD for simulation setup and trajectory analysis, but is also file-compatible with AMBER, CHARMM, and X-PLOR. NAMD is distributed free of charge with source code. For further information, see <http://www.ks.uiuc.edu/Research/namd>

## 3. Version Information:

NAMD Version 2.7 (b1)

Download from:

<http://www.ks.uiuc.edu/Development/Download/download.cgi?PackageName=NAMD>

## 4. Prerequisites:

### 4.1 Hardware:

The instructions from this best practice have been tested on the HPC Advisory Council, Dell PowerEdge M610 blade server based cluster.

- Intel® Xeon 5570 processors
- Mellanox QDR InfiniBand HCA (Driver: MLNX\_OFED 1.4)
- Mellanox QDR InfiniBand switch

### 4.2 Software:

1. OS

Intel® Cluster Ready platform, using Red Hat Enterprise Linux 5.3

2. The Intel® C and C++ Compiler for Linux; We used Intel® version 11.0. The full versions of compilers are required.

3. Libraries in addition to the Intel® Cluster Ready configuration

TCL

Open MPI 1.3.3 (compiled by Intel® compiler)

FFTW 2.1.5 (compiled by Intel® compiler)

charm++ 6.1 (compiled by Intel® compiler)

## 5. Building FFTW and Charm++

Since NAMD will be compiled by Intel® compiler in this test, FFTW and Charm++ also need to be built by Intel® compiler.

1. Steps to build FFTW 2.1.5:

```
./configure --prefix=/application/fftw-2.1.5-intel  
F77=ifort CC=icc CFLAGS=-O3 FFLAGS="-O3  
--enable-shared --enable-threads" --enable-type-  
prefix --enable-float  
make  
make install
```

This will create FFTW single-precision library which is needed by NAMD.

2. Steps to build Charm++ 6.1:

```
./build charm++ mpi-linux-x86_64 mpicxx ifort  
--no-shared -O3 -DCMK_OPTIMIZE=1
```

## 6. Building NAMD:

1. Modify Make.charm

Point Charm++ to the right directory

2. Run NAMD configure:

```
./config Linux-x86_64-MPI-icc --charm-arch mpi-  
linux-x86_64-ifort-mpicxx
```

3. Modify architecture file:

arch/Linux-x86\_64-MPI-icc.arch to have to the same architecture name you created - mpi-linux-x86\_64-ifort-mpicxx.

arch/Linux-x86\_64.fftw to reflect current fftw directory. Similarly file arch/Linux-x86\_64.tcl

4. Compile NAMD

```
cd Linux-x86_64-MPI-icc
```

```
make -j 4
```

Executable namd2 will be generated

## 7. Running NAMD:

1. Download benchmark file from:

<http://www.ks.uiuc.edu/Research/namd/utilities/apoa1.tar.gz>

2. Untar the data files

```
tar -vzxf apoa1.tar.gz
```

3. Run benchmark

```
mpirun -np <number of procs> -hostfile  
<hostfile> Linux-x86_64-MPI-icc/namd2 apoa1/  
apoa1.namd
```

Benchmark time will be printed near the end of the output file.

