

NAMD 2.7 – Execution Guidelines for running applications in aggregated environment using ScaleMP’s vSMP Foundation

Overview

NAMD is a parallel, object-oriented molecular dynamics code designed for high-performance simulation of large bio-molecular systems [1]. This document describes the best practice for running NAMD 2.7 on a vSMPowered system.

NAMD uses MPI to communicate between its processes. The MPI library affects the way the application scales to large number of CPUs. The default MPI library NAMD ships with is MPICH2.

ScaleMP recommends running NAMD with MPICH2 tuned for vSMP Foundation, and to compile NAMD with the Intel Compiler.

Building NAMD using Intel Compilers

Following is the build script required to build NAMD executable using Intel compilers:

```
#!/bin/sh
. /opt/intel/Compiler/latest/bin/iccvars.sh intel64
. /opt/intel/Compiler/latest/bin/ifortvars.sh intel64

export BASE=`pwd`/NAMD_2.7_Source
export CC=icc
export CXX=icpc
export FC=ifort
export MPICH_CC=icc
export MPICH_CXX=icpc
export MPICH_F77=ifort
export MPICH_F90=ifort

export MPIHOME=/opt/ScaleMP/mpich2/1.3.2
export PATH=$MPIHOME/bin:$PATH
export LD_LIBRARY_PATH=$MPIHOME/lib:$LD_LIBRARY_PATH

rm -rf NAMD_2.7_Source
tar xfz NAMD_2.7_Source.tar.gz || exit 1
cd $BASE
rm -rf charm-6.2.2
tar xf charm-6.2.2.tar || exit 1
cd charm-6.2.2
sed -e "s/CpvCExtern(int, _traceCoreOn);/\/\/\/CpvCExtern(int,
_traceCoreOn);/g" -i ${BASE}/charm-6.2.2/src/conv-perf/traceCore.h
./build charm++ mpi-linux-amd64 --no-shared -O -DCMK_OPTIMIZE=1 || exit 1

cd $BASE
./config tcl fftw Linux-x86_64-icc --charm-base ./charm-6.2.2 --charm-arch
mpi-linux-amd64 --tcl-prefix ${BASE}/tcl/linux-amd64 --fftw-prefix
${BASE}/fftw/linux-amd64 --cxx icpc --cxx-opts "-fast" --cc icc --cc-opts
"-fast" || exit 1
cd Linux-x86_64-icc

make || exit 1
```

Running NAMD 2.7 with MPICH2 tuned for vSMP

Although NAMD uses MPICH2, it is not invoked via mpirun. NAMD uses a dedicated job-launcher called "charmrun" instead. Charmrun takes an option to specify the number of MPI processes to be invoked, the name of the application (namd2) and the name of the input file (stmv.namd). Below is an example showing execution of NAMD with multiple MPI processes. Also listed are the environment variables required by MPICH2 tuned for vSMP Foundation.

Environment variables – MPICH2

```
export VSMP_PLACEMENT=PACKED
export VSMP_MEM_PIN=YES
export VSMP_VERBOSE=YES
```

Sample Run-Script

```
export BASE=`pwd`
export MPIHOME=/opt/ScaleMP/mpich2/1.3.2
export PATH=$MPIHOME/bin:${BASE}/NAMD_2.7_Source/Linux-x86_64-icc:$PATH

export KMP_ALIGN_ALLOC=4K
export MALLOC_TOP_PAD_=4294967296

export LD_PRELOAD=${BASE}/libvsmpclib.so:$LD_PRELOAD
sudo su -c "echo 0 > /proc/sys/kernel/vsyscall164"
sudo su -c "echo 0 > /proc/sys/kernel/randomize_va_space"

# Run NAMD with 16 MPI processes
charmrun +p16 namd2 stmv/stmv.namd >& ./stmv-16.txt
```

References

[1] <http://www.ks.uiuc.edu/Research/namd>