

## NAMD 3.0b2 (GPU)

### Webpage

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

### Version

3.0b2

### Build Environment

- GCC 11.2.1 (gcc-toolset/11)
- Intel MKL 2022.2.1
- CUDA 12.0

### Files Required

- NAMD\_3.0b2\_Source.tar.gz
- charm-7.0.tar.gz
- tcl8.5.9-linux-x86\_64.tar.gz
- tcl8.5.9-linux-x86\_64-threaded.tar.gz
  - obtained from <http://www.ks.uiuc.edu/Research/namd/libraries>
  - for fftw, MKL was used.

### Build Procedure

```
#!/bin/sh

VERSION=3.0b2
CHARM_VERSION=7.0.0
WORKDIR=/gwork/users/${USER}/namd-gnu-cuda
SOURCEDIR=/home/users/${USER}/Software/NAMD/${VERSION}
NAME=NAMD_${VERSION}_Source

TARBALL=${SOURCEDIR}/${NAME}.tar.gz
TARBALL_CHARM=${SOURCEDIR}/charm-${CHARM_VERSION}.tar.gz

LIBURL=http://www.ks.uiuc.edu/Research/namd/libraries
#FFTW=fftw-linux-x86_64
#FFTW_URL=${LIBURL}/${FFTW}.tar.gz
TCL=tcl8.5.9-linux-x86_64
TCL_URL=${LIBURL}/${TCL}.tar.gz
TCL_THREADED=tcl8.5.9-linux-x86_64-threaded
TCL_THREADED_URL=${LIBURL}/${TCL_THREADED}.tar.gz

#TARBALL_FFTW=${SOURCEDIR}/${FFTW}.tar.gz
TARBALL_TCL=${SOURCEDIR}/${TCL}.tar.gz
TARBALL_TCL_THREADED=${SOURCEDIR}/${TCL_THREADED}.tar.gz

PARALLEL=12

#-----
umask 0022

export LANG=""
export LC_ALL=C

module -s purge
module -s load gcc-toolset/11
module -s load mkl/2022.2.1
module -s load cuda/12.0

cd ${WORKDIR}
```

```

if [ -d ${NAME} ]; then
  mv ${NAME} namd_erase
  rm -rf namd_erase &
fi

tar zxf ${TARBALL}
cd ${NAME}
tar zxf ${TARBALL_CHARM}
ln -s charm-${CHARM_VERSION} charm

cd charm-${CHARM_VERSION}

export CC=gcc
export CXX=g++
export F90=gfortran
export F77=gfortran

./build charm++ verbs-linux-x86_64 smp gcc \
  --no-build-shared --with-production -j${PARALLEL}
cd ../

tar zxf ${TARBALL_TCL}
mv ${TCL} tcl
tar zxf ${TARBALL_TCL_THREADED}
mv ${TCL_THREADED} tcl-threaded

./config Linux-x86_64-g++ \
  --charm-arch verbs-linux-x86_64-smp-gcc \
  --with-mkl \
  --with-python \
  --with-single-node-cuda
cd Linux-x86_64-g++

make -j${PARALLEL}
make release

```

(Release tarball was then unpacked in /apl/namd/3.0b2.)

## Notes

- Please check <https://www.ks.uiuc.edu/Research/namd/alpha/3.0alpha/>
  - In the beta versions, +pmePEs should be replaced with +pmepes.
- This build maybe equivalent to official binary version. (Our nvcc version might be newer then official build.)
- To offload things to GPU, "CUDASOAintegrate on" should be specified. ~~Please note that this is not available for minimization.~~
  - ~~According to official notes, minimize and dynamics should be performed in separate runs.~~
  - ~~You can't specify "CUDASOAintegrate on" after the minimization in an input file.~~
  - (fix: 3.0b1 or later version (at least) can perform minimization with GPU.)
- # of cpu cores is specified with +p(number). In the case of this number is 2, namd3 puts following error at the end of run and freezes.
  - +p1 and +p4 are free from this issue. (I only tried +p1, +p2, and +p4.)

```
FATAL ERROR: CUDA error cudaMemcpyAsync(h_array, d_array, sizeofT*array_len, cudaMemcpyDeviceToHost, stream) in file src/CudaUtils.C, function copy_DtoH_async_T, line 235 on Pe 0 (ccg001 device 0 pci 0:7:0): invalid argument
```

- If you don't output coordinates (both of restartFreq and dcdFreq are not specified), energy of the final step won't be shown.
- UCX version does not work correctly.
  - ("openpmix" and "ompmix" options were not yet tried.)