NAMD 3.0b2 (GPU)

ウェブページ

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

バージョン

3.0b2

ビルド環境

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

ビルドに必要なファイル

- 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
    - http://www.ks.uiuc.edu/Research/namd/libraries より取得
    - fftw については MKL を利用

ビルド手順

```
#!/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
```

(作成されたリリース用の tarball を /apl/namd/3.0b2 以下に展開)

メモ

- https://www.ks.uiuc.edu/Research/namd/alpha/3.0alpha/ を参照
  - ベータ版では +pmePEs のかわりに +pmepes と書かなければならないように見える
- 配布されているバイナリ版とほぼ同じ構成だと思われる。(nvcc のバージョンはこちらの方が新しいかもしれない)
- GPU に全てを投げる場合には CUDASOAintegrate on の指定が必要。~~minimize 時には使えないので注意。~~
  - ~~minimize と dynamics は別のインプットで行うよう推奨されている。~~
  - ~~一つのインプット中で minimize の後に CUDASOAintegrate on を指定するようなこともできない~~
  - (訂正: 少なくとも 3.0b1 以降では minimize でも GPU を使えるようです)
- 実行時の CPU コア数は +p(数字) で指定するが、+p2 の時にはなぜか終了時に以下のようなエラーが出て、正常に終了しない
  - +p1 や +p4 では問題は起きていない(+p1, +p2, +p4 以外では未検証)

```
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
```

- 座標を出力しない場合(restartFreq と dcdFreq を指定していない場合)、最終ステップのエネルギーが表示されない
- UCX 版は正常に動作しなかったため回避。
  - openpmix や ompipmix は指定していない。シングルノード版ではこれらは未検証。