Gromacs 2025.2 with CUDA support

ウェブページ

http://www.gromacs.org/

バージョン

2025.2

ビルド環境

- GCC 13.3.1 (gcc-toolset-13)
- CUDA 12.8 Update 1
- Open MPI 4.1.8 (CUDA-aware)
- cmake 3.31.6
- python 3.12.9 (2025/3/10 作成の conda 環境。libtorch ビルドに利用)
- openblas 0.3.29-lp64
- cuDNN 9.10.1
- cuDSS 0.5.0
- cuSPARSELt 0.7.1

必要なファイル

- gromacs-2025.2.tar.gz
- regressiontests-2025.2.tar.gz
- (以下手順中でも一部ファイルを取得)

ビルド手順

LibTorch 2.7.0 (簡易版)

CPU 版でダウンロードしたソースを利用。

```
[user@ccfep pytorch]$ mkdir build && cd build
[user@ccfep build]$ module purge
[user@ccfep build]$ . /apl/conda/20250310/conda_init.sh
(base) [user@ccfep build]$ export CUDSS_ROOT_DIR=/apl/cudss/0.5.0/libcudss-linux-x86_64-0.5.0.16_cuda12-archive
(base) [user@ccfep build]$ export CUSPARSELT_ROOT_DIR=/apl/cusparselt/0.7.1/libcusparse_lt-linux-x86_64-0.7.1.0-archive
(base) [user@ccfep build]$ export TORCH_CUDA_ARCH_LIST="8.0"
(base) [user@ccfep build]$ module load cuda/12.8u1 cudnn/9.10.1-cuda12 cudss/0.5.0.16-cuda12 cusparselt/0.7.1 openblas/0.3.29-lp64 gcc-
toolset/13
(base) [user@ccfep build]$ cmake3 .. \
    -GNinja \
    -DBLAS=OpenBLAS \
    -DBUILD_FUNCTORCH=OFF \
    -DBUILD_PYTHON=False \
    -DBUILD_TEST=True \
    -DCMAKE_BUILD_TYPE=Release \
    -DCMAKE_INSTALL_PREFIX=/apl/libtorch/2.7.0/cu128 \
    -DCMAKE_PREFIX_PATH="/apl/cuda/12.8u1;/apl/cudnn/9.10.1/cudnn-linux-x86_64-9.10.1.4_cuda12-archive" \
    -DPython_EXECUTABLE=/lustre/rccs/apl/ap/conda/20250310/envs/gpuenv/bin/python \
    -DTORCH_BUILD_VERSION=2.7.0a0+git1341794 \
    -DUSE_CUDA=ON \
    -DUSE_CUDNN=ON \
    -DUSE_CUDSS=ON \
    -DUSE_CUSPARSELT=ON \
    -DUSE_NUMPY=True
(base) [user@ccfep build]$ ninja -j2
(base) [user@ccfep build]$ ninja -j2 install
```

テストのみ ccgpu で実行。

```
(base) [user@ccgpu build]$ ninja -j2 test
```

- 以下のテストが失敗
  - 56 - vec_test_all_types_AVX512 (ILLEGAL) : CPU が AVX512 未対応のため仕方ないと思われる
  - 57 - vec_test_all_types_AVX2 (Failed) : ?
  - 144 - ProcessGroupNCCLErrorsTest (Failed) : ?

Gromacs 2025.2

ccgpu にてビルド

```
#!/bin/sh

VERSION=2025.2
INSTALL_PREFIX=/apl/gromacs/${VERSION}-CUDA

BASEDIR=/home/users/${USER}/Software/Gromacs/${VERSION}/
GROMACS_TARBALL=${BASEDIR}/gromacs-${VERSION}.tar.gz
REGRESSION_TARBALL=${BASEDIR}/regressiontests-${VERSION}.tar.gz
WORKDIR=/gwork/users/${USER}
REGRESSION_PATH=${WORKDIR}/regressiontests-${VERSION}

FFTW_VER=3.3.10
FFTW_PATH=${BASEDIR}/fftw-${FFTW_VER}.tar.gz

PARALLEL=12
export LANG=C

#------------------------------------------------------------------------
umask 0022

module -s purge
module -s load gcc-toolset/13
module -s load openmpi/4.1.8/gcc13-cuda12.8u1
module -s load cuda/12.8u1
module -s load cmake/3.31.6
module -s load openblas/0.3.29-lp64
module -s load cudnn/9.10.1-cuda12
module -s load cudss/0.5.0.16-cuda12
module -s load cusparselt/0.7.1

TORCH_DIR=/apl/libtorch/2.7.0/cu128
OPENBLAS_DIR=/apl/openblas/0.3.29/lp64
export CUDNN_ROOT_DIR=/apl/cudnn/9.10.1/cudnn-linux-x86_64-9.10.1.4_cuda12-archive
export CUDSS_ROOT_DIR=/apl/cudss/0.5.0/libcudss-linux-x86_64-0.5.0.16_cuda12-archive
export CUSPARSELT_ROOT_DIR=/apl/cusparselt/0.7.1/libcusparse_lt-linux-x86_64-0.7.1.0-archive

#export CUDA_VISIBLE_DEVICES=0,1
unset OMP_NUM_THREADS

cd ${WORKDIR}
if [ -d gromacs-${VERSION} ]; then
 mv gromacs-${VERSION} gromacs_erase
 rm -rf gromacs_erase &
fi

if [ -d regressiontests-${VERSION} ]; then
 mv regressiontests-${VERSION} regressiontests_erase
 rm -rf regressiontests_erase &
fi

tar xzf ${GROMACS_TARBALL}
tar xzf ${REGRESSION_TARBALL}
cd gromacs-${VERSION}

# single precision, no MPI
mkdir rccs-s
cd rccs-s
```

```
cmake .. \
  -DCMAKE_PREFIX_PATH="${TORCH_DIR};${OPENBLAS_DIR}" \
  -DCMAKE_INSTALL_PREFIX=${INSTALL_PREFIX} \
  -DCMAKE_VERBOSE_MAKEFILE=ON \
  -DCMAKE_C_COMPILER=gcc \
  -DCMAKE_CXX_COMPILER=g++ \
  -DGMX_MPI=OFF \
  -DGMX_GPU=CUDA \
  -DGMX_DOUBLE=OFF \
  -DGMX_THREAD_MPI=ON \
  -DGMX_USE_CUFFTMP=OFF \
  -DGMX_NNPOT=TORCH \
  -DCAFFE2_USE_CUDNN=ON \
  -DCAFFE2_USE_CUSPARSELT=ON \
  -DUSE_CUDSS=ON \
  -DPython_EXECUTABLE=/usr/bin/python3 \
  -DGMX_BUILD_OWN_FFTW=ON \
  -DGMX_BUILD_OWN_FFTW_URL=${FFTW_PATH} \
  -DREGRESSIONTEST_DOWNLOAD=OFF \
  -DREGRESSIONTEST_PATH=${REGRESSION_PATH}
make -j${PARALLEL} && make check && make install
cd ..

# single precision, with MPI
mkdir rccs-mpi-s
cd rccs-mpi-s
cmake .. \
  -DCMAKE_PREFIX_PATH="${TORCH_DIR};${OPENBLAS_DIR}" \
  -DCMAKE_INSTALL_PREFIX=${INSTALL_PREFIX} \
  -DCMAKE_VERBOSE_MAKEFILE=ON \
  -DCMAKE_C_COMPILER=mpicc \
  -DCMAKE_CXX_COMPILER=mpicxx \
  -DGMX_MPI=ON \
  -DGMX_GPU=CUDA \
  -DGMX_DOUBLE=OFF \
  -DGMX_THREAD_MPI=OFF \
  -DGMX_USE_CUFFTMP=OFF \
  -DGMX_NNPOT=TORCH \
  -DCAFFE2_USE_CUDNN=ON \
  -DCAFFE2_USE_CUSPARSELT=ON \
  -DUSE_CUDSS=ON \
  -DPython_EXECUTABLE=/usr/bin/python3 \
  -DGMX_USE_PLUMED=ON \
  -DGMX_BUILD_OWN_FFTW=ON \
  -DGMX_BUILD_OWN_FFTW_URL=${FFTW_PATH} \
  -DREGRESSIONTEST_DOWNLOAD=OFF \
  -DREGRESSIONTEST_PATH=${REGRESSION_PATH}
make -j${PARALLEL} && make check && make install
cd ..
```

テスト(Gromacs)

全てパス。

メモ

- libtorch 2.7.0 と合わせてビルドしているため、Neural Network Potential が利用可能です。
  - ただし、公式ドキュメントによると、利用するモデルも pytorch 2.7.0 で作成する必要があるとのことです。pytorch の別バージョンを使う場合には上記の手順を参考にご自身でビルドする必要がありそうです
  - gromacs 実行時の NNP 計算で GPU を使うためには環境変数 GMX_NN_DEVICE に cuda を設定する必要があります。(デフォルトは cpu です。)
    - RCCS の gromacs/2025.2-CUDA モジュールでは明示的に cuda に設定しています。
    - 設定はログファイル中の "GMX_NN_DEVICE environment variable found. Using device: cuda." のようなメッセージから確認できます。
  - libtorch の公式配布のバイナリがそのまま使える場合は libtorch のビルドは省略できます。今回は glibc のバージョンの問題でバイナリ版を利用できませんでした。

- plumed も有効になっています。
    - RCCS 提供の gromacs/2025.2-CUDA の module では PLUMED_KERNEL 環境変数に plumed 2.9.3 の libplumedKernel.so が指定されています。