Gromacs 2025.4 with CUDA support

Weboage

http://www.gromacs.org/

Version

2025.4

Build Environment

- GCC 13.3.1 (gcc-toolset-13)
- CUDA 12.8 Update 1
- Open MPI 4.1.8
- cmake 3.31.6
- openblas 0.3.29-lp64
- cuDNN 9.10.1
- cuDSS 0.5.0
- cuSPARSELt 0.7.1

Files Required

- gromacs-2025.4.tar.gz
- regressiontests-2025.4.tar.gz
- (some of files will be downloaded during installation)

Build Procedure

Built and tested on ccgpu.

```
#!/bin/sh

VERSION=2025.4
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 # not CUDA-aware Open MPI!
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 ..
```

Tests

All the tests have passed successfully.

Notes

- The installation procedure is basically the same as that for version 2025.2.
    - CUDA-aware Open MPI is not used for this version, tentatively. This is because MPI version (gmx_mpi) does not work when CUDA-aware Open MPI is employed.
    - This issue is attributed to system side libraries/drivers (network of GPU). We have not yet been able to identify the cause. It is not believed to be caused by changes to the Gromacs code.
        - Previously installed 2025.2 CUDA version is also suffering from this issue. (There were, of course, absolutely no problems at the time of installation.)
    - For thread MPI version, the build procedure is the same as that for 2025.2. (MPI not used.)