

## LAMMPS 23Jun22 with GPU support

### ウェブページ

<https://www.lammps.org>

### バージョン

23Jun22

### ビルド環境

- Intel oneAPI Compiler Classic 2022.2.1
- Intel MKL 2022.2.1
- HPC-X 2.11 (Open MPI 4.1.4)
  - 実際のビルド時には HPC-X 2.13.1 (Open MPI 4.1.5)を利用。エラー回避のため runtime を HPC-X 2.11 に切り替え
  - 以下の手順では HPC-X 2.11 として記述
- CUDA 11.6

### ビルドに必要なファイル

- lammps-stable.tar.gz
- (一部ファイルは以下スクリプト中で取得)

### ビルド手順

#### conda 環境

lammps 29Sep21 CPU 版で作成したものを流用。

#### lammps 本体

```
#!/bin/sh

VERSION=23Jun22
NAME=lammps-23Jun2022
INSTALL_PREFIX=/apl/lammps/2022-Jun23-CUDA

BASEDIR=/home/users/${USER}/Software/LAMMPS/${VERSION}
LAMMPS_TARBALL=${BASEDIR}/lammps-stable.tar.gz

WORKDIR=/gwork/users/${USER}/lammps-23Sep2022-cuda
LAMMPS_WORKDIR=${WORKDIR}/${NAME}

GPU_ARCH=sm_80
VMD_MOLFILE_INC=/home/users/${USER}/Software/VMD/1.9.4/vmd-1.9.4a57/plugins/include

PARALLEL=12

#-----
umask 0022
export LANG=C
ulimit -s unlimited

module -s purge

# load intel compiler
. ~/intel/oneapi/compiler/2022.2.1/env/vars.sh

. /apl/lammps/2022-Jun23-CUDA/conda_init.sh

module -s load mkl/2022.2.1
module -s load cuda/11.6
module -s load openmpi/4.1.4-hpcx/intel2022.2.1
```

```

export CC=mpicc
export CXX=mpicxx
export FC=mpif90
export MPICC=mpicc
export MPICXX=mpicxx
export MPIFC=mpif90

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

tar zxf ${LAMMPS_TARBALL}

cd ${NAME}
sed -i -e "s/xHost/march=core-avx2/" cmake/CMakeLists.txt
mkdir build && cd build

# Disabled PKGs:
# FFMPEG, ADIOS, MDI, VTK: noavail
# MSCG: gsl too old
# MESSAGE: ZeroMQ support not enabled
# QUIP: failed to build
# ML-HDNNP: failed to build
# KIM: CDDL is incompatible with GPL
# LATTE: technical problem of cmake? (LAPACK and BLAS)
# NETCDF: to avoid EVP_KDF_ctrl error
# MPIIO: not maintained?

cmake ../cmake \
-DLAMMPS_MACHINE=rccs-cuda \
-DENABLE_TESTING=on \
-DCMAKE_INSTALL_PREFIX=${INSTALL_PREFIX} \
-DCMAKE_C_COMPILER=mpicc \
-DCMAKE_CXX_COMPILER=mpicxx \
-DCMAKE_Fortran_COMPILER=mpif90 \
-DCMAKE_MPI_C_COMPILER=mpicc \
-DCMAKE_MPI_CXX_COMPILER=mpicxx \
-DCMAKE_MPI_Fortran_COMPILER=mpif90 \
-DCMAKE_CXX_FLAGS_DEBUG="-Wall -Wextra -g" \
-DCMAKE_CXX_FLAGS_RELWITHDEBINFO="-Wall -Wextra -g -O2 -DNDEBUG" \
-DCMAKE_CXX_FLAGS_RELEASE="-O3 -DNDEBUG" \
-DCMAKE_Fortran_FLAGS_DEBUG="-Wall -Wextra -g" \
-DCMAKE_Fortran_FLAGS_RELWITHDEBINFO="-Wall -Wextra -g -O2 -DNDEBUG" \
-DCMAKE_Fortran_FLAGS_RELEASE="-O3 -DNDEBUG" \
-DCMAKE_C_FLAGS_DEBUG="-Wall -Wextra -g" \
-DCMAKE_C_FLAGS_RELWITHDEBINFO="-Wall -Wextra -g -O2 -DNDEBUG" \
-DCMAKE_C_FLAGS_RELEASE="-O3 -DNDEBUG" \
-DLAMMPS_EXCEPTIONS=on \
-DBUILD_SHARED_LIBS=on \
-DBUILD_TOOLS=on \
-DBUILD_MPI=on \
-DBUILD_OMP=on \
-DFFT=MKL \
-DFFT_SINGLE=on \
-DFFT_MKL_THREADS=on \
-DWITH_JPEG=yes \
-DWITH_PNG=yes \
-DWITH_GZIP=yes \
-DPKG_ASPHERE=on \
-DPKG_ATC=on \
-DPKG_AWPMO=on \
-DPKG_BOCS=on \

```

-DPKG\_BODY=on \  
-DPKG\_BROWNIAN=on \  
-DPKG\_CG-DNA=on \  
-DPKG\_CG-SDK=on \  
-DPKG\_CLASS2=on \  
-DPKG\_COLLOID=on \  
-DPKG\_COLVARS=on \  
-DPKG\_COMPRESS=on \  
-DPKG\_CORESHELL=on \  
-DPKG\_DIELECTRIC=on \  
-DPKG\_DIFFRACTION=on \  
-DPKG\_DIPOLE=on \  
-DPKG\_DPD-BASIC=on \  
-DPKG\_DPD-MESO=on \  
-DPKG\_DPD-REACT=on \  
-DPKG\_DPD-SMOOTH=on \  
-DPKG\_DRUDE=on \  
-DPKG\_EFF=on \  
-DPKG\_EXTRA-COMPUTE=on \  
-DPKG\_EXTRA-DUMP=on \  
-DPKG\_EXTRA-FIX=on \  
-DPKG\_EXTRA-MOLECULE=on \  
-DPKG\_EXTRA-PAIR=on \  
-DPKG\_FEP=on \  
-DPKG\_GPU=on \  
-DGPU\_API=cuda \  
-DGPU\_ARCH=\${GPU\_ARCH} \  
-DPKG\_GRANULAR=on \  
-DPKG\_H5MD=on \  
-DPKG\_INTEL=on \  
-DPKG\_INTERLAYER=on \  
-DPKG\_KIM=off \  
-DDOWNLOAD\_KIM=no \  
-DPKG\_KOKKOS=off \  
-DPKG\_KSPACE=on \  
-DPKG\_LATBOLTZ=on \  
-DPKG\_MACHDYN=on \  
-DDOWNLOAD\_EIGEN3=on \  
-DPKG\_MANIFOLD=on \  
-DPKG\_MANYBODY=on \  
-DPKG\_MC=on \  
-DPKG\_MDI=off \  
-DPKG\_MEAM=on \  
-DPKG\_MESONT=on \  
-DPKG\_MESSAGE=on \  
-DPKG\_MGPT=on \  
-DPKG\_MISC=on \  
-DPKG\_ML-HDNNP=off \  
-DDOWNLOAD\_N2P2=no \  
-DPKG\_ML-IAP=on \  
-DPKG\_ML-PACE=on \  
-DPKG\_ML-QUIP=off \  
-DDOWNLOAD\_QUIP=no \  
-DPKG\_ML-RANN=on \  
-DPKG\_ML-SNAP=on \  
-DPKG\_MOFFF=on \  
-DPKG\_MOLECULE=on \  
-DPKG\_MOLFILE=on \  
-DMOLFILE\_INCLUDE\_DIR=\${VMD\_MOLFILE\_INC} \  
-DPKG\_MPIIO=off \  
-DPKG\_MSCG=off \  
-DPKG\_NETCDF=off \  
-DPKG\_OPENMP=on \  
-DPKG\_OPT=on \  
-DPKG\_ORIENT=on \

```
-DPKG_PERI=on \  
-DPKG_PHONON=on \  
-DPKG_PLUGIN=on \  
-DPKG_PLUMED=on \  
-DDOWNLOAD_PLUMED=yes \  
-DPKG_POEMS=on \  
-DPKG_PTM=on \  
-DPKG_PYTHON=on \  
-DPKG_QEQ=on \  
-DPKG_QMMM=on \  
-DPKG_QTB=on \  
-DPKG_REACTION=on \  
-DPKG_REAXFF=on \  
-DPKG_REPLICA=on \  
-DPKG_RIGID=on \  
-DPKG_SCAFACOS=on \  
-DDOWNLOAD_SCAFACOS=yes \  
-DPKG_SHOCK=on \  
-DPKG_SMTBQ=on \  
-DPKG_SPH=on \  
-DPKG_SPIN=on \  
-DPKG_SRD=on \  
-DPKG_TALLY=on \  
-DPKG_UEF=on \  
-DPKG_VORONOI=on \  
-DDOWNLOAD_VORO=yes \  
-DPKG_VTK=off \  
-DPKG_YAFF=on \  
-DBLAS_LIBRARIES="-qmk" \  
-DCMAKE_BUILD_TYPE=Release
```

```
make VERBOSE=1 -j ${PARALLEL}
```

```
export OMP_NUM_THREADS=2
```

```
#make test # will put error...
```

```
make install
```

```
cp -a ../examples ${INSTALL_PREFIX}
```

```
cd ${INSTALL_PREFIX}
```

```
for f in etc/profile.d/*; do
```

```
ln -s $f .
```

```
done
```

```
cd lib64
```

```
if [ -f liblammps_rccs-cuda.so ]; then
```

```
ln -s liblammps_rccs-cuda.so liblammps.so
```

```
fi
```

```
if [ -f liblammps_rccs-cuda.so.0 ]; then
```

```
ln -s liblammps_rccs-cuda.so.0 liblammps.so.0
```

```
fi
```

## パッケージ

```
ASPHERE ATC AWPMD BOCS BODY BROWNIAN CG-DNA CG-SDK CLASS2 COLLOID COLVARS  
COMPRESS CORESHELL DIELECTRIC DIFFRACTION DIPOLE DPD-BASIC DPD-MESO  
DPD-REACT DPD-SMOOTH DRUDE EFF EXTRA-COMPUTE EXTRA-DUMP EXTRA-FIX EXTRA-MOLECULE  
EXTRA-PAIR FEP GPU GRANULAR H5MD INTEL INTERLAYER KSPACE LATBOLTZ MACHDYN  
MANIFOLD MANYBODY MC MEAM MESONT MGPT MISC ML-IAP ML-PACE ML-RANN ML-SNAP  
MOFFF MOLECULE MOLFILE OPENMP OPT ORIENT PERI PHONON PLUGIN PLUMED POEMS  
PTM PYTHON QEQ QMMM QTB REACTION REAXFF REPLICA RIGID SCAFACOS SHOCK  
SMTBQ SPH SPIN SRD TALLY UEF VORONOI YAFF
```

## テスト

(make test の部分のみ GPU 演算ノードで実行)

テストログのコピーを /apl/lammps/2022-Jun23-CUDA/Testing/ に置いています。

The following tests FAILED:

- 11 - AtomStyle:coul (Failed)
- 42 - ComputeGlobal (Failed)
- 94 - MolPairStyle:coul\_diel (Failed)
- 100 - MolPairStyle:coul\_shield (Failed)
- 102 - MolPairStyle:coul\_slater\_long (Failed)
- 137 - MolPairStyle:lj\_class2\_soft (Failed)
- 152 - MolPairStyle:lj\_cut\_soft (Failed)
- 158 - MolPairStyle:lj\_expand\_coul\_long (Failed)
- 171 - MolPairStyle:lj\_sdk\_coul\_long (Failed)
- 172 - MolPairStyle:lj\_sdk\_coul\_table (Failed)
- 176 - MolPairStyle:lj\_switch3\_coulgauss\_long (Failed)
- 200 - MolPairStyle:tip4p\_long\_soft (Failed)
- 203 - MolPairStyle:wf\_cut (Failed)
- 211 - AtomicPairStyle:buck\_coul\_cut\_qeq\_point (Failed)
- 212 - AtomicPairStyle:buck\_coul\_cut\_qeq\_shielded (Failed)
- 229 - AtomicPairStyle:edip (Failed)
- 236 - AtomicPairStyle:meam (Failed)
- 237 - AtomicPairStyle:meam\_spline (Failed)
- 238 - AtomicPairStyle:meam\_sw\_spline (Failed)
- 241 - AtomicPairStyle:reaxff-acks2 (Failed)
- 242 - AtomicPairStyle:reaxff-acks2\_efield (Failed)
- 243 - AtomicPairStyle:reaxff (Failed)
- 244 - AtomicPairStyle:reaxff\_lgvdw (Failed)
- 245 - AtomicPairStyle:reaxff\_noqeq (Failed)
- 246 - AtomicPairStyle:reaxff\_tabulate (Failed)
- 247 - AtomicPairStyle:reaxff\_tabulate\_flag (Failed)
- 264 - ManybodyPairStyle:comb (Failed)
- 272 - ManybodyPairStyle:ilp-graphene-hbn (Failed)
- 273 - ManybodyPairStyle:ilp-graphene-hbn\_notaper (Failed)
- 277 - ManybodyPairStyle:lcbop (Failed)
- 286 - ManybodyPairStyle:pace\_product (Failed)
- 287 - ManybodyPairStyle:pace\_recursive (Failed)
- 299 - ManybodyPairStyle:tersoff (Failed)
- 304 - ManybodyPairStyle:tersoff\_shift (Failed)
- 314 - BondStyle:gaussian (Failed)
- 357 - KSpaceStyle:ewald\_tri (Failed)
- 359 - KSpaceStyle:pppm\_ad (Failed)
- 360 - KSpaceStyle:pppm\_cg (Failed)
- 362 - KSpaceStyle:pppm\_cg\_tiled (Failed)
- 369 - KSpaceStyle:pppm\_disp\_tip4p (Failed)
- 377 - KSpaceStyle:pppm\_tip4p (Failed)
- 382 - KSpaceStyle:scafacos\_direct (Failed)
- 383 - KSpaceStyle:scafacos\_ewald (Failed)
- 384 - KSpaceStyle:scafacos\_fmm (Failed)
- 385 - KSpaceStyle:scafacos\_fmm\_tuned (Failed)
- 386 - KSpaceStyle:scafacos\_p2nfft (Failed)
- 387 - FixTimestep:adapt\_coul (Failed)
- 390 - FixTimestep:addforce\_const (Failed)
- 392 - FixTimestep:addtorque\_const (Failed)
- 411 - FixTimestep:nph (Failed)
- 412 - FixTimestep:nph\_sphere (Failed)
- 414 - FixTimestep:npt\_iso (Failed)
- 415 - FixTimestep:npt\_sphere\_aniso (Failed)
- 416 - FixTimestep:npt\_sphere\_iso (Failed)
- 440 - FixTimestep:rigid\_npt\_small (Failed)
- 454 - FixTimestep:smd\_couple (Failed)
- 462 - FixTimestep:temp\_csld (Failed)
- 483 - DihedralStyle:table\_cut\_linear (Failed)
- 485 - DihedralStyle:table\_linear (Failed)
- 486 - DihedralStyle:table\_spline (Failed)
- 496 - ImproperStyle:inversion\_harmonic (Failed)

- ほとんどは軽微な数値エラーと、インテルコンパイラ利用時の lattice 関連。大筋では問題無いと判断。

## メモ

- ccfep でビルド。GPU 演算ノードにてテスト。
- HPC-X 2.13.1 の実行環境利用時は並列計算で通信エラー。HPC-X 2.11 の実行環境に切り替えることでエラーは解消。
- NETCDF を on にするとビルドに失敗するため、今回は回避。システム側のライブラリの問題の可能性が高い。
- システムの python 3.6 を使っていれば、conda 環境は不要であった可能性が高い。また、NETCDF も on にできたかもしれない。