

LAMMPS 2Aug23

ウェブページ

<https://www.lammps.org>

バージョン

2Aug23

ビルド環境

- GCC 12.1.1 (gcc-toolset-12)
- Intel MPI 2021.10.0

ビルドに必要なファイル

- lammps-stable_2Aug2023.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

```
#!/bin/sh

VERSION=2Aug23
NAME=lammps-stable_2Aug2023
INSTALL_PREFIX=/apl/lammps/2023-Aug2

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

WORKDIR=/gwork/users/${USER}
LAMMPS_WORKDIR=${WORKDIR}/${NAME}

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
module -s load gcc-toolset/12
module -s load intelmpi/2021.10.0

PYTHONEXE=/usr/bin/python3.6m
PYTHONINC=/usr/include/python3.6m

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

tar xzf ${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
```

```
# ML-HDNNP: failed to build
# ML-IAP: failed to build
# KIM: CDDL is incompatible with GPL
# MPIIO: not maintained?
# INTEL: not necessary for gcc build
```

```
cmake ../cmake \
-DLAMMPS_MACHINE=rccs \
-DENABLE_TESTING=on \
-DCMAKE_INSTALL_PREFIX=${INSTALL_PREFIX} \
-DCMAKE_C_COMPILER=gcc \
-DCMAKE_CXX_COMPILER=g++ \
-DCMAKE_Fortran_COMPILER=gfortran \
-DCMAKE_MPI_C_COMPILER=mpicc \
-DCMAKE_MPI_CXX_COMPILER=mpicxx \
-DCMAKE_MPI_Fortran_COMPILER=mpif90 \
-DCMAKE_C_FLAGS_RELEASE="-O3 -DNDEBUG" \
-DCMAKE_CXX_FLAGS_RELEASE="-O3 -DNDEBUG" \
-DCMAKE_Fortran_FLAGS_RELEASE="-O3 -DNDEBUG" \
-DPython_EXECUTABLE=${PYTHONEXE} \
-DPython_INCLUDE_DIR=${PYTHONINC} \
-DLAMMPS_EXCEPTIONS=on \
-DBUILD_SHARED_LIBS=on \
-DBUILD_TOOLS=on \
-DBUILD_MPI=on \
-DBUILD_OMP=on \
-DBUILD_LAMMPS_GUI=off \
-DFFT=FFTW3 \
-DFFT_SINGLE=on \
-DFFT_FFTW_THREADS=on \
-DWITH_JPEG=on \
-DWITH_PNG=on \
-DWITH_GZIP=on \
-DPKG_ADIOS=off \
-DPKG_AMOEBA=on \
-DPKG_ASHERE=on \
-DPKG_ATC=on \
-DPKG_AWPMMD=on \
-DPKG_BOCS=on \
-DPKG_BODY=on \
-DPKG_BPM=on \
-DPKG_BROWNIAN=on \
-DPKG_CG-DNA=on \
-DPKG_CG-SPICA=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_ELECTRODE=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=off \
-DPKG_GRANULAR=on \
-DPKG_H5MD=on \
-DPKG_INTEL=off \
-DPKG_INTERLAYER=on \
-DPKG_KIM=off \
-DDOWNLOAD_KIM=off \
-DPKG_KOKKOS=on \
-DKokkos_ARCH_ZEN3=on \
-DKokkos_ENABLE_OPENMP=on \
-DPKG_KSPACE=on \
-DPKG_LATBOLTZ=on \
-DPKG_LEPTON=on \
-DPKG_MACHDYN=on \
-DDOWNLOAD_EIGEN3=on \
-DPKG_MANIFOLD=on \
-DPKG_MANYBODY=on \
-DPKG_MC=on \
-DPKG_MDI=on \
-DPKG_MEAM=on \
-DPKG_MESONT=on \
-DPKG_MGPT=on \
-DPKG_MISC=on \
-DPKG_ML-HDNNP=off \
-DDOWNLOAD_N2P2=off \
-DPKG_ML-IAP=off \
-DPKG_ML-PACE=on \
-DPKG_ML-QUIP=on \
-DDOWNLOAD_QUIP=on \
-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=on \
-DPKG_NETCDF=on \
-DPKG_OPENMP=on \
-DPKG_OPT=on \
-DPKG_ORIENT=on \
-DPKG_PERI=on \
-DPKG_PHONON=on \
-DPKG_PLUGIN=on \
-DPKG_PLUMED=on \
-DDOWNLOAD_PLUMED=on \
-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=on \
-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=on \
-DPKG_VTK=off \
-DPKG_YAFF=on \
-DBLA_VENDOR=OpenBLAS \
-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
  if [ -f $f ]; then
    ln -s $f .
  fi
done

cd lib64
if [ -f liblammps_rccs.so ]; then
  ln -s liblammps_rccs.so liblammps.so
fi
if [ -f liblammps_rccs.so.0 ]; then
  ln -s liblammps_rccs.so.0 liblammps.so.0
fi

```

パッケージ

AMOeba ASPHERE ATC AWPMD BOCS BODY BPM BROWNIAN CG-DNA;CG-SPICA CLASS2 COLLOID COLVARS COMPRESS CORESHELL DIELECTRIC DIFFRACTION DIPOLE DPD-BASIC DPD-MESO DPD-REACT DPD-SMOOTH DRUDE EFF ELECTRODE EXTRA-COMPUTE EXTRA-DUMP EXTRA-FIX EXTRA-MOLECULE EXTRA-PAIR FEP GRANULAR H5MD INTEL INTERLAYER KSPACE LATBOLTZ LEPTON MACHDYN MANIFOLD MANYBODY MC MDI MEAM MESONT MGPT MISC ML-PACE ML-RANN ML-SNAP MOFFF MOLECULE MOLFILE MSCG NETCDF 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

テストログのコピーは [/apl/lammps/2023-Aug2/Testing/](#) に置いています。

詳細についてはそちらをご確認ください。

(軽微な数値エラーが多いですが、いくつかは注意すべきものがあるかもしれません。)

```

37 - SimpleCommands (SEGFAULT)
39 - Groups (Failed)
127 - MolPairStyle:coul_slater_long (Failed)
249 - AtomicPairStyle:buck_coul_cut_qeq_point (Failed)
250 - AtomicPairStyle:buck_coul_cut_qeq_shielded (Failed)
267 - AtomicPairStyle:edip (Failed)
273 - AtomicPairStyle:lepton_sphere (Failed)
274 - AtomicPairStyle:lj_cut_sphere (Failed)
275 - AtomicPairStyle:lj_expand_sphere (Failed)
280 - AtomicPairStyle:meam_spline (Failed)
281 - AtomicPairStyle:meam_sw_spline (Failed)
284 - AtomicPairStyle:reaxff-acks2 (Failed)
285 - AtomicPairStyle:reaxff-acks2_efield (Failed)
286 - AtomicPairStyle:reaxff (Failed)
287 - AtomicPairStyle:reaxff_lgvdw (Failed)
288 - AtomicPairStyle:reaxff_noqeq (Failed)
289 - AtomicPairStyle:reaxff_tabulate (Failed)
290 - AtomicPairStyle:reaxff_tabulate_flag (Failed)
315 - ManybodyPairStyle:ilp-graphene-hbn (Failed)
316 - ManybodyPairStyle:ilp-graphene-hbn_notaper (Failed)

```

363 - BondStyle:harmonic_restrain (Failed)
409 - KSpaceStyle:pppm_ad (Failed)
410 - KSpaceStyle:pppm_cg (Failed)
411 - KSpaceStyle:pppm_cg_ad (Failed)
412 - KSpaceStyle:pppm_cg_tiled (Failed)
414 - KSpaceStyle:pppm_dielectric (Failed)
429 - KSpaceStyle:pppm_tip4p (Failed)
434 - KSpaceStyle:scafacos_direct (Failed)
435 - KSpaceStyle:scafacos_ewald (Failed)
436 - KSpaceStyle:scafacos_fmm (Failed)
437 - KSpaceStyle:scafacos_fmm_tuned (Failed)
438 - KSpaceStyle:scafacos_p2nfft (Failed)
444 - FixTimestep:addtorque_const (Failed)
506 - FixTimestep:rigid_nvt (Failed)
519 - FixTimestep:spring_rg (Failed)
526 - FixTimestep:wall_harmonic_const (Failed)
527 - FixTimestep:wall_lepton_const (Failed)
531 - FixTimestep:wall_morse_const (Failed)
533 - FixTimestep:wall_table_linear (Failed)
534 - FixTimestep:wall_table_spline (Failed)
547 - DihedralStyle:quadratic (Failed)
549 - DihedralStyle:table_cut_linear (Failed)
551 - DihedralStyle:table_linear (Failed)
552 - DihedralStyle:table_spline (Failed)

メモ

- Intel MPI 2021.10.0 には gcc12 用の mpi.mod 等がないため、Fortran 並列向けのものはビルドできず。
- Open MPI でもビルドは可能。テスト結果についても SimpleCommands の segfault が消えるなど、特に問題は無し。しかし、計算速度(rhodo を使用)が若干安定しないため、今回は見送り。
- Intel コンパイラを使った場合の状況については[こちらの記事を参照](#)
- GCC11 に比べて GCC12 は数 % ほど速度が向上している。intel パッケージを使わない条件では、GCC12 は試した範囲では一番速度が出ていた。(rhodo 系, intel パッケージは使わない条件, gcc, aocc, intel classic, intel oneapi のコンパイラで比較)
- AOCC 4.1 を使った場合は GCC11 よりは速度が出ていた(rhodo のみで検証)。ただ、GCC12 以外には若干負けるため、今回は GCC12 を採用。
- INTEL パッケージについては gcc では本来の性能が出ないため、外してあります。
- MKL を使った場合(cmake で -DBLA_VENDOR=Intel10_64lp 指定、FFTW も MKL を指定)、omp パッケージを使ってスレッド並列した場合に結果がおかしくなる例を確認。libgomp と libiomp の混在が原因のエラー？ひとまず mkl は回避して openblas を使用したところ問題無く動作。
 - 公式テスト中にはこの問題を検証できるテストが存在しない？
 - MKL を使った場合でも OpenMP スレッドを使わない場合は問題無し。
 - MKL で libgomp を使うことは可能ではあるものの、lammmps のビルドシステム上でそれを行うにはかなり手がかかるので今回は実施せず。
 - 非標準な場所にある openblas の指定も標準的な方法では難しい？
 - 実行前に openblas/0.3.21-lp64 の module を読み込めば、openmp 版の openblas を利用することも可能。