

LAMMPS 2Aug23

Webpage

<https://www.lammps.org>

Version

2Aug23

Build Environment

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

Files Required

- lammps-stable_2Aug2023.tar.gz
- (many packages will be downloaded in the procedure below.)

Build Procedure

```
#!/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_ASPHERE=on \

-DPKG_ATC=on \

-DPKG_AWPMO=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

```

List of Available Packages

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

Tests

Copy of test results is available at </apl/lammps/2023-Aug2/Testing/>. Please see that directory for details.
 (Most of the errors are minor numerical ones. However, some of them might not be negligible.)

```

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)

Notes

- Packages depending on MPI Fortran are not installed, since Intel MPI 2021.10.0 doesn't have module files (such as mpi.mod) for GCC12.
- Open MPI also works for this version. Segmentation fault of SimpleCommands test disappears when Open MPI is employed. However, computational performance is bit unstable in comparison with Intel MPI version. So we employed Intel MPI this time.
- For Intel Compilers, [please see this page](#).
- Performance of GCC12 version is better than GCC11 one by ~5%. GCC12 was the best among when "intel" package is not used. (test condition: rhodo test, without "intel" package, gcc, aocc, intel classic, intel oneapi compilers were tested.)
- AOCC 4.1 also shows good performance. It is better than GCC11 but slightly worse than GCC12 in rhodo test.
- INTEL package is disabled, since it is targeting only Intel compilers.
- If MKL is used by specifying `-DBLA_VENDOR=Intel10_64lp` and `-DFFTW=MKL` (FFT_FFTW_THREADS is also modified to FFT_MKL_THREADS), omp package does not work correctly with multiple threads. Mixing of libgomp and libiomp might be the cause of the problem?
 - This error can't be detected in the official test suite?
 - Single thread run is OK when MKL is employed.
 - It is possible to use libgomp for MKL. However, it is tedious in this case...
 - It is difficult to use OpenBLAS installed in non-standard path. There seems to be no easy way to do this?
 - You can use openmp version of openblas by loading "openblas/0.3.21-lp64" module before executing lammmps.