

LAMMPS 29Oct20

Webpage

<http://lammps.sandia.gov/>

Version

29Oct20

Build Environment

- Intel Parallel Studio XE 2018 update 4
- cmake 3.16.3

Files Required

- lammps-29Oct20.tar.gz (renamed downloaded lammps-stable.tar.gz)
- (some of files are downloaded during the installation)

Build Procedure

```
#!/bin/sh

VERSION=29Oct20
NAME=lammps-${VERSION}
INSTALL_PREFIX=/local/apl/lx/lammps${VERSION}

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

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

VMD_MOLFILE_INC=/local/apl/lx/vmd193/lib/plugins/include

PARALLEL=12

#-----
umask 0022
export LANG=C

module purge
module load intel_parallelstudio/2018update4
module load cmake/3.16.3

export CC=mpiicc
export CXX=mpiicpc
export FC=mpiifort
export MPICC=mpiicc
export MPICXX=mpiicpc
export MPIFC=mpiifort

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

tar xzf ${LAMMPS_TARBALL}

cd ${NAME}
mkdir build && cd build
```

```
# Disabled PKGs:
# ADIOS, QUIP, QMMM, VTK: noavail
# MSCG: gsl too old
# KOKKOS: not sure what this is
# MESSAGE: ZeroMQ support not enabled

cmake ../cmake \
-DLAMMPS_MACHINE=rccs \
-DENABLE_TESTING=on \
-DCMAKE_INSTALL_PREFIX=${INSTALL_PREFIX} \
-DCMAKE_C_COMPILER=mpiicc \
-DCMAKE_CXX_COMPILER=mpiicpc \
-DCMAKE_Fortran_COMPILER=mpiifort \
-DCMAKE_MPI_C_COMPILER=mpiicc \
-DCMAKE_MPI_CXX_COMPILER=mpiicpc \
-DCMAKE_MPI_Fortran_COMPILER=mpiifort \
-DBUILD_SHARED_LIBS=on \
-DBUILD_TOOLS=on \
-DBUILD_MPI=on \
-DBUILD_OMP=on \
-DFFT=MKL \
-DFFT_SINGLE=on \
-DDOWNLOAD_LATTE=on \
-DDOWNLOAD_SCAFACOS=on \
-DDOWNLOAD_VORO=on \
-DDOWNLOAD_PLUMED=on \
-DDOWNLOAD_EIGEN3=on \
-DMOLFILE_INCLUDE_DIRS=${VMD_MOLFILE_INC} \
-DWITH_JPEG=yes \
-DWITH_PNG=yes \
-DWITH_GZIP=yes \
-DPKG_OPT=on \
-DPKG_USER-OMP=on \
-DPKG_USER-INTEL=on \
-DPKG_GPU=off \
-DPKG_KOKKOS=off \
-DPKG_ASHERE=on \
-DPKG_BODY=on \
-DPKG_CLASS2=on \
-DPKG_COLLOID=on \
-DPKG_COMPRESS=on \
-DPKG_CORESHELL=on \
-DPKG_DIPOLE=on \
-DPKG_GRANULAR=on \
-DPKG_KSPACE=on \
-DPKG_LATTE=on \
-DPKG_MANYBODY=on \
-DPKG_MC=on \
-DPKG_MESSAGE=on \
-DPKG_MISC=on \
-DPKG_MLIAP=on \
-DPKG_MOLECULE=on \
-DPKG_MPIIO=on \
-DPKG_PERI=on \
-DPKG_POEMS=on \
-DPKG_PYTHON=on \
-DPKG_QEQ=on \
-DPKG_REPLICA=on \
-DPKG_RIGID=on \
-DPKG_SHOCK=on \
-DPKG_SNAP=on \
-DPKG_SPIN=on \
-DPKG_SRD=on \
-DPKG_USER-ATC=on \
-DPKG_USER-AWPMO=on \
```

```

-DPKG_USER-BOCS=on \
-DPKG_USER-CGDNA=on \
-DPKG_USER-CGSDK=on \
-DPKG_USER-COLVARS=on \
-DPKG_USER-DIFFRACTION=on \
-DPKG_USER-DPD=on \
-DPKG_USER-DRUDE=on \
-DPKG_USER-EFF=on \
-DPKG_USER-FEP=on \
-DPKG_USER-H5MD=on \
-DPKG_USER-LB=on \
-DPKG_USER-MANIFOLD=on \
-DPKG_USER-MEAMC=on \
-DPKG_USER-MESODPD=on \
-DPKG_USER-MESONT=on \
-DPKG_USER-MGPT=on \
-DPKG_USER-MISC=on \
-DPKG_USER-MOFFF=on \
-DPKG_USER-MOLFILE=on \
-DPKG_USER-NETCDF=on \
-DPKG_USER-PHONON=on \
-DPKG_USER-PLUMED=on \
-DPKG_USER-PTM=on \
-DPKG_USER-QTB=on \
-DPKG_USER-REACTION=on \
-DPKG_USER-REAXC=on \
-DPKG_USER-SCAFACOS=on \
-DPKG_USER-SDPD=on \
-DPKG_USER-SMD=on \
-DPKG_USER-SMTBQ=on \
-DPKG_USER-SPH=on \
-DPKG_USER-TALLY=on \
-DPKG_USER-UEF=on \
-DPKG_USER-YAFF=on \
-DPKG_VORONOI=on \
-DBLAS_LIBRARIES="-mk" \
-DCMAKE_BUILD_TYPE=Release

make -j ${PARALLEL}

# to avoid error on COMPILER_SUPPORTSfast=2:INTERNAL= line
sed -i -e "s/line\.split.*\/line\.rsplit('=','1)'" \
  ../unittest/python/python-capabilities.py \
  ../unittest/python/python-pylammps.py

make test # will put error...
make install

cp -a ../examples ${INSTALL_PREFIX}

# no extra tests... current test suite is designed for developers, not for us
cd ${INSTALL_PREFIX}
for f in etc/profile.d/*; do
  ln -s $f .
done

```

Notes

- This time the library is built as a shared library. You need to set LD_LIBRARY_PATH correctly (see samples).
- LAMMPS_POTENTIALS environment variable will be set in samples and module (lammps/29Oct20/intel).
- Build of QMMM was skipped this time (might be possible with QE 6.5).

Tests

- We performed only unittests for this version. Current test suite (lammps-testing) is designed to run in the

singularity container; this is not what we want.

- In case you found misbehavior, official samples under "examples" directory may be helpful to verify the issue.
- Result of unittests (make test) were copied to Testing directory. 87 errors were found.
 - Most of those errors seem to be minor numerical errors (error criterion is often too strict). Changing Intel compiler version (17, 19, 19.1) does not help.
 - Result of "lattice" command (which is also used in unittest of reax/c) seems to be different between GCC and Intel versions. (see below)
 - Following unittests shows large deviations from the reference.
 - 6 AtomStyle:AtomStyleTest.body_nparticle: quaternion value is wrong for only one case.
 - 64 MolPairStyle:coul_diel: forces(?) after "displace_atoms all random" are wrong. Random caused the issue?
 - 211 ManybodyPairStyle:lcbop PairStyle.plain: not sure what happens
 - Following tests also showed large deviations. This may be caused by "lattice" command. At least in the case of reax_c, no significant deviations are observed when the coordinate is loaded with "read_data".
 - 180 AtomicPairStyle:edip PairStyle.omp
 - 188 AtomicPairStyle:meam_sw_spline: PairStyle.Plain
 - 191 AtomicPairStyle:reax_c PairStyle.plain, PairStyle.omp
 - 192 AtomicPairStyle:reax_c_lgvdw PairStyle.plain, PairStyle.omp
 - 193 AtomicPairStyle:reax_c_noeq PairStyle.plain, PairStyle.omp

```
lattice diamond 3.77
region box block 0 2 0 2 0 2
create_box 2 box
create_atoms 1 box
displace_atoms all random 0.1 0.1 0.1 623426
mass 1 12.0
mass 2 13.0
set type 1 type/fraction 2 0.5 998877
set type 1 charge 0.01
set type 2 charge -0.01
velocity all create 100 4534624 loop geom
```

In unittest (such as reax/c), test coordinate is generated with "lattice" command.

```
--- dump.reaxc-gnu.rdx 2021-03-04 14:57:55.000000000 +0900
+++ dump.reaxc-intel.rdx 2021-03-04 14:57:29.000000000 +0900
@@ -19,7 +19,7 @@
10 1 0.538716 0.255633 0.224506
11 1 0.794884 0.959892 0.252133
12 1 0.734352 0.247648 0.976632
-13 2 0.628096 0.160143 0.0790692
+13 1 0.628096 0.160143 0.0790692
14 2 0.60761 0.39555 0.363303
15 1 0.850337 0.118908 0.384987
16 1 0.870172 0.326989 0.11101
@@ -30,7 +30,7 @@
21 2 0.157099 0.614141 0.118317
22 1 0.0980522 0.863381 0.390315
23 2 0.333814 0.606643 0.34671
-24 1 0.351575 0.866212 0.132087
+24 2 0.351575 0.866212 0.132087
25 1 0.505105 0.495226 0.958289
26 2 0.506033 0.746602 0.241533
27 1 0.777748 0.457362 0.238711
```

However, the result depends on the compiler type (GCC or Intel)...