

## LAMMPS 16Mar18 (stable release) for LX

### Webpage

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

### Version

16Mar18

### Build Environment

- Intel Compiler 2018.2.199
- Intel MKL 2018 Update 2
- Intel MPI 2018.2.199
- libjpeg-turbo 1.2.90

### Files Required

- lammps-stable.tar.gz (16Mar18)
- (some files are obtained in the following script)

### Build Procedure

```
#!/bin/sh

VERSION=16Mar18
INSTALL_PREFIX=/local/apl/lx/lammps16Mar18

BASEDIR=/home/users/${USER}
LAMMPS_TARBALL=${BASEDIR}/lammps-stable.tar.gz
WORKDIR=/work/users/${USER}

PARALLEL=12

#-- libs

VMD_PLUGIN_INC=`echo /local/apl/lx/vmd193/lib/plugins/include | sed -e 's/\///g'` # molfile
VORO_VER=0.4.6 # voronoi
VORO=http://math.lbl.gov/voro++/download/dir/voro+-+-${VORO_VER}.tar.gz

#-----
umask 0022

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

tar zxf ${LAMMPS_TARBALL}
cd lammps-${VERSION}

# setup makefiles, libraries, and external resources
## main
sed -e "/intel_cpu_intelmpi/s./# rccs = USER-INTEL package, Intel MPI, MKL FFT/" src/MAKE/OPTIONS/Makefile.intel_cpu >
src/MAKE/MINE/Makefile.rccs
## atc
( cd lib/atc && \
  sed -e s/icc/mpiicc/ -e s/lammps.installed/lammps.empty/ Makefile.icc > Makefile.rccs && \
  make -f Makefile.rccs -j ${PARALLEL} && \
  cd ../ )
## awpmd
( cd lib/awpmd && \
  sed -e s/linalg/empty/ -e s/mpicxx/mpiicc/ Makefile.mpi > Makefile.rccs && \
```

```

make -f Makefile.rccs -j ${PARALLEL} && \
cd ../..)
## colvars
( cd lib/colvars && \
sed -e s/mpicxx/mpiicc/ -e s/-funroll-loops/-unroll/ Makefile.mpi > Makefile.rccs && \
make -f Makefile.rccs -j ${PARALLEL} && \
cd ../..)
## h5md
( cd lib/h5md && \
make -f Makefile.mpi -j ${PARALLEL} && \
cd ../..)
## meam
( cd lib/meam && \
sed -e s/mpifort/mpiifort/ -e s/mpicc/mpiicc/ -e s/mpicxx/mpiicpc/ Makefile.mpi > Makefile.rccs && \
make -f Makefile.rccs -j ${PARALLEL} && \
cd ../..)
### molfile
( cd lib/molfile && \
sed -i -e "s/molfile_SYSINC.*/molfile_SYSINC =-!$VMD_PLUGIN_INC/" Makefile.lammps && \
cd ../..)
## poems
( cd lib/poems && \
make -f Makefile.icc -j ${PARALLEL} && \
cd ../..)
## reax
( cd lib/reax && \
make -f Makefile.ifort -j ${PARALLEL} && \
cd ../..)
## voronoi
( cd lib/voronoi && \
wget ${VORO} && \
tar xzf voro+-${VORO_VER}.tar.gz && \
cd voro+-${VORO_VER} && \
sed -i -e "s/^CXX=.*CXX=icpc/" -e "s/^CFLAGS=.*CFLAGS=-Wall -O3 -fPIC/" config.mk && \
make -j ${PARALLEL} && \
cd ../ && \
ln -s voro+-${VORO_VER}/src includelink && \
ln -s voro+-${VORO_VER}/src liblink && \
cd ../..)

#----

# now make lammps
cd src
make yes-all no-ext
make no-KOKKOS \
no-GPU \
no-LATTE \
yes-VORONOI \
yes-USER-H5MD \
yes-USER-MOLFILE \
yes-USER-NETCDF
make -j ${PARALLEL} rccs
make -j ${PARALLEL} rccs mode=shlib
cd ../

# mkdir and install files
mkdir -p ${INSTALL_PREFIX}/src
cp src/lmp_rccs src/liblammps_rccs.so src/*.h ${INSTALL_PREFIX}/src
ln -s ${INSTALL_PREFIX}/src/liblammps_rccs.so ${INSTALL_PREFIX}/src/liblammps.so
cp -r LICENSE \
README \
bench/ \
doc/ \
examples/ \

```

```
potentials/\  
python/\  
tools/\  
${INSTALL_PREFIX}
```

## List of Packages

ASPHERE, BODY, CLASS2, COLLOID, COMPRESS, CORESHELL, DIPOLE, GRANULAR  
KSPACE, MANYBODY, MC, MEAM, MISC, MOLECULE, MPIIO, OPT, PERI, POEMS  
PYTHON, QEQ, REAX, REPLICA, RIGID, SHOCK, SNAP, SRD, VORONOI

USER-ATC, USER-AWPMO, USER-CGDNA, USER-CGSDK, USER-COLVARS,  
USER-DIFFRACTION, USER-DPD, USER-DRUDE, USER-EFF, USER-FEP,  
USER-H5MD, USER-INTEL, USER-LB, USER-MANIFOLD, USER-MEAMC,  
USER-MESO, USER-MGPT, USER-MISC, USER-MOLFILE, USER-NETCDF,  
USER-OMP, USER-PHONON, USER-QTB, USER-REAXC, USER-SMTBQ,  
USER-SPH, USER-TALLY, USER-UEF

## Tests

- Serial tests via "run\_tests.py" passed (excluding legacy tests).
- Parallel test is manually performed in the following procedure. Some tests are excluded or performed separately.

```
VERSION=16Mar18  
WORKDIR=/work/users/${USER}  
LAMMPSDIR=${WORKDIR}/lammps-${VERSION}  
LAMMPSTESTS=${WORKDIR}/lammps-testing-master  
LAMMPSDIR_EXC=`echo $LAMMPSDIR | sed -e 's/\/\\\\/g'`  
  
export LD_LIBRARY_PATH="${LD_LIBRARY_PATH}:${LAMMPSDIR}/src"  
export PYTHONPATH="${PYTHONPATH}:${LAMMPSDIR}/python"  
  
NPROCS=4  
JOBNAME=mpi_4  
LOGFILE=tests_mpi4_all  
EXCLUDES="tad neb mscg USER/quip kim gcmc USER/misc/imd USER/lb USER/atc USER/misc/pimd USER/misc/i-pi USER/misc/grem USER/eff/Li-  
dendritic COUPLE ASPHERE/tri ASPHERE/poly ASPHERE/line USER/dpd/dpdx-shardlow"  
  
sed -i -e "s/^read_data.*/read_data    ${LAMMPSDIR_EXC}\\\\bench\\\\data.rhodo/" tests/examples/accelerate/in.rhodo  
sed -i -e "s/^ pair_coeff.*/ pair_coeff    ** ${LAMMPSDIR_EXC}\\\\potentials\\\\Cu_mishin1.eam.alloy Cu/" tests/examples/USER/misc/ti/in.ti_spring  
cp /home/users/qf7/ramdisk/lammps-16Mar18/potentials/{CC.KC,CH.airebo} tests/examples/USER/misc/kolmogorov-crespi  
  
python \  
  lammps_testing/regression.py \  
  ${JOBNAME} \  
  "mpirun -np ${NPROCS} ${LAMMPSDIR}/src/lmp_rccs" \  
  ${LAMMPSTESTS}/tests/examples/ \  
  -exclude ${EXCLUDES} >& ${LOGFILE}
```

## Notices

- Files are installed in /local/apl/lx/lammps16Mar18/
- Executable (lmp\_rccs) and library are installed under src/. (Can be accessed via symbolic link bin/)
- Sample job script is available in samples/ directory.
- Python module files are copied in python/ directory.
- LAMMPS header files in src/ directory are also copied in src/ directory.

Notices about parallel tests (Items written in bold are common errors to gcc version)

- **balance/balance.var.dynamics: minor(?) numerical errors (TotEng(4.9e-6), Press(1.93e-5) etc.)**
- balance/balance.neigh.rcb: get stuck
- **balance/balance.kspace: numerical errors (TotEng(0.21), Press(0.83)他)**
- **balance/balance.clock.dynamic: minor(?) numerical errors (TotEng(1.16e-5), Press(4.56e-5))**
- balance/balance.bond.slow: get stuck
- balance/balance.bond.fast: get stuck
- balance/balance: get stuck
- **VICOSITY/nemd.2d: numerical errors (TotEng(0.0; no error), Press(0.36), v\_visc(3.82) etc.)**

- **USER/eff/ECP/Si2H6/Si2H6.ang: get stuck**
- **USER/eff/CH4/ch4\_ionized.dynamics: get stuck**

#### Notices about excluded or separately performed tests

- mscg: package not installed
- USER/quip: package not installed
- kim: package not installed
- gcrc: package not installed
- tad/tad: requires additional argument. Even when the argument added, it does not work. (error in handling 4-bytes and 8-bytes variables)
- neb: requires additional argument. It works if the argument added. Numerical accuracy is not evaluated.
- dpd/dpdx-shardlow: requires additional argument. Works if the argument added. Numerical accuracy is not checked.
- imd, USER/lb: too time consuming.
- COUPLE, USER/misc/i-pi, USER/misc/grem: skipped (only COUPLE/multiple is tested and works fine).
- ASPHERE/tri: ERROR: Illegal neigh\_modify command
- ASPHERE/line: ERROR: Illegal neigh\_modify command
- ASPHERE/poly: ERROR on proc 0: Neighbor list overflow, boost neigh\_modify one
- USER/eff/Li-dendritic: missing file?
- USER/atc: file missing.
- USER/misc/pimd: needs to mkdir before tun tests. I don't know how to verify this.

#### Others

- latte doesn't work in this build environment.
- vmd molfile plugin files are available in /local/apl/lx/vmd193/lib/plugins/LINUXAMD64/molfile directory.