

## LAMMPS 22Aug18 (stable release) for LX

### ウェブページ

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

### バージョン

22Aug18

### ビルド環境

- ▶ Intel Parallel Studio XE 2017 update 4
  - ▶ Intel Compiler 17.0.4.196
  - ▶ Intel MKL 2017 update 3
  - ▶ Intel MPI 2017.0.3
- ▶ CUDA 8.0.61/9.1.85
- ▶ libjpeg-turbo 1.2.90

### ビルドに必要なファイル

- ▶ lammps-stable\_22Aug18.tar.gz (16Mar18)
- ▶ lammps-testing-master.zip (2018/10/25 にダウンロード)
- ▶ (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

NAME=lammps-stable_22Aug2018
VERSION=22Aug18
INSTALL_PREFIX=/local/apl/lx/lammps22Aug18
BASEDIR=/home/users/${USER}
LAMMPS_TARBALL=${BASEDIR}/${NAME}.tar.gz
PARALLEL=12

#-- libs

LATTE_VER=1.2.1
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

#----- funcs -----
function buildgpu() {
  ## gpu
  # assume you are at lammps top directory
  ( cd lib/gpu && \
    sed -e "/^CUDA_ARCH/s/-arch=sm.*-arch=${LAMMPS_CUDA_ARCH}/" \
      -e "/^CUDA_LIB/s/ -L${CUDA_HOME}/lib64/stubs/" \
      -e "s/mpicxx/mpicpc/" \
      Makefile.linux > Makefile.rccs && \
    make -f Makefile.rccs clean && \
    make -f Makefile.rccs -j ${PARALLEL} && \
    sed -i -e "/^gpu_SYSPATH/s/ -L${CUDA_HOME}/lib64/stubs/" \
      Makefile.lammps && \
    cd ../ )
}

#-----
umask 0022

module purge
module load intel_parallelstudio/2017update4
module load cuda/9.1

cd ${INSTALL_PREFIX}
if [ -d ${NAME} ]; then
  mv ${NAME} lammps_erase
  rm -rf lammps_erase &
fi
tar xzf ${LAMMPS_TARBALL}
mv ${NAME}/.
rm -rf ${NAME}/.git*
rmdir ${NAME}

# setup makefiles, libraries, and external resources
## main
mkdir -p src/MAKE/MINE
```

```

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 ../ )
## gpu
export CUDA_HOME=/local/apl/lx/cuda-9.1
LAMMPS_CUDA_ARCH="sm_70" # V100
buildgpu
## h5md
( cd lib/h5md && \
  make -f Makefile.mpi -j ${PARALLEL} && \
  cd ../ )
### latte (make jobserver not available?)
( cd lib/latte && \
  curl -L -o "LATTE.tar.gz" https://github.com/lan/LATTE/archive/v\${LATTE\_VER}.tar.gz && \
  tar xzf LATTE.tar.gz && \
  cd LATTE-${LATTE_VER} && \
  sed -i -e "s/^FC = */FC = ifort/" -e "s/fopenmp/qopenmp -fPIC/" -e "s/^LIB = */LIB = -mkl=parallel/"
makefile.CHOICES && \
  make && \
  cd ../ && \
  ln -s LATTE-${LATTE_VER}/src includelink && \
  ln -s LATTE-${LATTE_VER} liblink && \
  ln -s LATTE-${LATTE_VER}/src/latte_c_bind.o filelink.o && \
  echo "latte_SYSINC =" > Makefile.lammps && \
  echo "latte_SYSLIB = -qopenmp ../lib/latte/filelink.o -latte -lifport" >> Makefile.lammps && \
  echo -n "latte_SYSPATH = -qopenmp" >> Makefile.lammps && \
  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 ${PARALLEL} && \
  cd ../ )
### molfile
( cd lib/molfile && \
  sed -i -e "s/molfile_SYSINC */molfile_SYSINC =I$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 ../ )
#----
# make volta version
cd src
make yes-all no-ext
make no-KOKKOS \
  yes-GPU \
  yes-LATTE \

```

```

yes-VORONOI \
yes-USER-H5MD \
yes-USER-MOLFILE \
yes-USER-NETCDF
make -j ${PARALLEL} rccs
make -j ${PARALLEL} rccs mode=shlib
mv lmp_rccs lmp_rccs_volta
mv liblammps_rccs.so liblammps_rccs_volta.so
cd ../
#---
# make pascal version
module switch cuda/8.0
export CUDA_HOME=/local/apl/lx/cuda-8.0
LAMMPS_CUDA_ARCH="sm_60" # P100
buildgpu
cd src
make no-GPU
make yes-GPU
make -j ${PARALLEL} rccs
make -j ${PARALLEL} rccs mode=shlib
mv lmp_rccs lmp_rccs_gpu
mv liblammps_rccs.so liblammps_rccs_gpu.so
cd ../
#---
# make default non-gpu version (just for easier use)
module unload cuda
cd src
make no-GPU
make -j ${PARALLEL} rccs
make -j ${PARALLEL} rccs mode=shlib
cd ../

```

## パッケージリスト

ASPHERE, BODY, CLASS2, COLLOID, COMPRESS, CORESHELL, DIPOLE, (GPU; GPU版のみ), GRANULAR, KSPACE, LATTE, MANYBODY, MC, MEAM, MISC, MOLECULE, MPIIO, OPT, PERI, POEMS, PYTHON, QEQ, REAX, REPLICA, RIGID, SHOCK, SNAP, SPIN, SRD, VORONOI  
 USER-ATC, USER-AWPMD, USER-BOCS, 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-MOFFF, USER-MOLFILE, USER-NETCDF, USER-OMP, USER-PHONON, USER-QT, USER-REAXC, USER-SMTBQ, USER-SPH, USER-TALLY, USER-UEF

## テスト

(環境設定部分については別途対応; Volta版は現状未テスト)

```

#!/bin/sh
VERSION=22Aug18
WORKDIR=/home/users/${USER}/ramdisk
LAMMPSDIR=/local/apl/lx/lammps${VERSION}
LAMMPSTESTS=/home/users/${USER}/lammps-testing-master.20181025.zip
LAMMPSDIR_EXC=`echo $LAMMPSDIR | sed -e 's/V/\W/g'`
export LD_LIBRARY_PATH="${LD_LIBRARY_PATH}:${LAMMPSDIR}/src"
export PYTHONPATH="${PYTHONPATH}:${LAMMPSDIR}/python"
JOBNAME=basic
# classification
NOAVAIL="msgc kim USER/quip USER/misc/i-pi"
NODE1="gcmc"
NODES3="tad neb USER/dpd/dpdx-shardlow"
STALL="USER/eff/ECP/Si2H6 USER/eff/CH4"
STALL_INTEL="balance"
UNKNOWN="USER/misc/pimd USER/misc/grem USER/atc COUPLE ASPHERE/tri ASPHERE/line"
TOOLONG="USER/misc/imd USER/lb"
EXCLUDES="${NOAVAIL} ${NODE1} ${NODES3} ${STALL} ${STALL_INTEL} ${UNKNOWN} ${TOOLONG}"
cd $WORKDIR
if [ -d lammps-testing-master ]; then
  mv lammps-testing-master lammps-testing-master_erase
  rm -rf lammps-testing-master_erase &
fi
unzip ${LAMMPSTESTS}
cd lammps-testing-master
# mod and prep some files
sed -i -e "s/^read_data.*read_data    ${LAMMPSDIR_EXC}vbenchv/data.rhodo"
tests/examples/accelerate/in.rhodo
sed -i -e "s/^ pair_coeff.* pair_coeff    ** ${LAMMPSDIR_EXC}vpotentialsV/Cu_mishin1.eam.alloy Cu/"

```

```

tests/examples/USER/misc/ti/in.ti_spring
cp ${LAMMPSDIR}/potentials/(CC.KC,CH.airebo) tests/examples/USER/misc/kolmogorov-crespi

# basic tests
NPROCS=4
python \
  lammmps_testing/regression.py \
  ${JOBNAME} \
  "mpirun -np ${NPROCS} ${LAMMPSBIN}" \
  ${WORKDIR}/lammmps-testing-master/tests/examples/ \
  -exclude ${EXCLUDES}

# gcmc
NPROCS=1
python \
  lammmps_testing/regression.py \
  ${JOBNAME} \
  "mpirun -np ${NPROCS} ${LAMMPSBIN}" \
  ${WORKDIR}/lammmps-testing-master/tests/examples/ \
  -only ${NODE1}

# tad
NPROCS=3
python \
  lammmps_testing/regression.py \
  ${JOBNAME} \
  "mpirun -np ${NPROCS} ${LAMMPSBIN} -partition 3x1" \
  ${WORKDIR}/lammmps-testing-master/tests/examples/ \
  -only tad

# neb(sivac)
NPROCS=3
TESTDIR=tests/examples/neb
mv ${TESTDIR}/in.neb.hop1 ${TESTDIR}/ln.neb.hop1
mv ${TESTDIR}/in.neb.hop2 ${TESTDIR}/ln.neb.hop2
python \
  lammmps_testing/regression.py \
  ${JOBNAME} \
  "mpirun -np ${NPROCS} ${LAMMPSBIN} -partition 3x1" \
  ${WORKDIR}/lammmps-testing-master/tests/examples/ \
  -only neb
mv ${TESTDIR}/ln.neb.hop1 ${TESTDIR}/in.neb.hop1
mv ${TESTDIR}/ln.neb.hop2 ${TESTDIR}/in.neb.hop2
# neb(hop1,hop2)
NPROCS=4
TESTDIR=tests/examples/neb
mv ${TESTDIR}/in.neb.sivac ${TESTDIR}/ln.neb.sivac
python \
  lammmps_testing/regression.py \
  ${JOBNAME} \
  "mpirun -np ${NPROCS} ${LAMMPSBIN} -partition 4x1" \
  ${WORKDIR}/lammmps-testing-master/tests/examples/ \
  -only neb
mv ${TESTDIR}/ln.neb.sivac ${TESTDIR}/in.neb.sivac

# dpdx-shardlow
NPROCS=3
python \
  lammmps_testing/regression.py \
  ${JOBNAME} \
  "mpirun -np ${NPROCS} ${LAMMPSBIN} -partition 3x1" \
  ${WORKDIR}/lammmps-testing-master/tests/examples/ \
  -only USER/dpd/dpdx-shardlow

# ECP(Si2H6)
NPROCS=4
TESTDIR=tests/examples/USER/eff/ECP/Si2H6
mv ${TESTDIR}/in.Si2H6.ang ${TESTDIR}/ln.Si2H6.ang
python \
  lammmps_testing/regression.py \
  ${JOBNAME} \
  "mpirun -np ${NPROCS} ${LAMMPSBIN}" \
  ${WORKDIR}/lammmps-testing-master/tests/examples/ \
  -only USER/eff/ECP/Si2H6
mv ${TESTDIR}/ln.Si2H6.ang ${TESTDIR}/in.Si2H6.ang

# ECP(CH4)
NPROCS=4
TESTDIR=tests/examples/USER/eff/CH4
mv ${TESTDIR}/in.ch4_ionized.dynamics ${TESTDIR}/ln.ch4_ionized.dynamics
python \
  lammmps_testing/regression.py \
  ${JOBNAME} \
  "mpirun -np ${NPROCS} ${LAMMPSBIN}" \

```

```

${WORKDIR}/lammps-testing-master/tests/examples/ \
-only USER/eff/CH4
mv ${TESTDIR}/ln.ch4_ionized.dynamics ${TESTDIR}/ln.ch4_ionized.dynamics
# USER/lb
NPROCS=4
TESTDIR1=tests/examples/USER/lb/polymer
mv ${TESTDIR1}/ln.polymer_setgamma ${TESTDIR1}/ln.polymer_setgamma
mv ${TESTDIR1}/ln.polymer_default_gamma ${TESTDIR1}/ln.polymer_default_gamma
TESTDIR2=tests/examples/USER/lb/fourspheres
mv ${TESTDIR2}/ln.fourspheres_set_gamma ${TESTDIR2}/ln.fourspheres_set_gamma
mv ${TESTDIR2}/ln.fourspheres_default_gamma ${TESTDIR2}/ln.fourspheres_default_gamma
python \
  lammps_testing/regression.py \
  ${JOBNAME} \
  "mpirun -np ${NPROCS} ${LAMMPSBIN}" \
  ${WORKDIR}/lammps-testing-master/tests/examples/ \
  -only USER/lb
mv ${TESTDIR1}/ln.polymer_setgamma ${TESTDIR1}/ln.polymer_setgamma
mv ${TESTDIR1}/ln.polymer_default_gamma ${TESTDIR1}/ln.polymer_default_gamma
mv ${TESTDIR2}/ln.fourspheres_set_gamma ${TESTDIR2}/ln.fourspheres_set_gamma
mv ${TESTDIR2}/ln.fourspheres_default_gamma ${TESTDIR2}/ln.fourspheres_default_gamma
# balance
NPROCS=4
TESTDIR=tests/examples/balance
mv ${TESTDIR}/ln.balance.bond.fast ${TESTDIR}/ln.balance.bond.fast
mv ${TESTDIR}/ln.balance.bond.slow ${TESTDIR}/ln.balance.bond.slow
mv ${TESTDIR}/ln.balance.neigh.rcb ${TESTDIR}/ln.balance.neigh.rcb
mv ${TESTDIR}/ln.balance ${TESTDIR}/ln.balance
python \
  lammps_testing/regression.py \
  ${JOBNAME} \
  "mpirun -np ${NPROCS} ${LAMMPSBIN}" \
  ${WORKDIR}/lammps-testing-master/tests/examples/ \
  -only balance
mv ${TESTDIR}/ln.balance.bond.fast ${TESTDIR}/ln.balance.bond.fast
mv ${TESTDIR}/ln.balance.bond.slow ${TESTDIR}/ln.balance.bond.slow
mv ${TESTDIR}/ln.balance.neigh.rcb ${TESTDIR}/ln.balance.neigh.rcb
mv ${TESTDIR}/ln.balance ${TESTDIR}/ln.balance

```

## テスト結果

(GPU版、CPU版で共通です。)

- ▶ USER/eff/Li-dendritic: ファイル足りず? 動作させられず
- ▶ tad/tad: データ送受信でエラー
- ▶ neb/neb.sivac: 比較するべきファイルを間違えている? 多少の数値誤差も出ている
- ▶ neb/neb.hop1: 比較するべきファイルを間違えている? 多少の数値誤差も出ている
- ▶ neb/neb.hop2: 比較するべきファイルを間違えている? 軽微な数値誤差有り
- ▶ dpd/dpdx-shardlow: 比較するべきファイルを間違えている? 0ステップは良さそうだけど1ステップ時点で数値がずれている
- ▶ balance/balance.var.dynamics: 軽微な数値エラー
- ▶ balance/balance.kspace: 数値エラー
- ▶ balance/balance.clock.dynamic: 軽微な数値エラー

実行しなかったテスト関連

- ▶ balance/balance.neigh.rcb: ストール(GCCでビルドすればおそらく動作する)
- ▶ balance/balance.bond.slow: ストール(GCCでビルドすればおそらく動作する)
- ▶ balance/balance.bond.fast: ストール(GCCでビルドすればおそらく動作する)
- ▶ balance/balance: ストール(GCCでビルドすればおそらく動作する)
- ▶ mscg kim USER/quip USER/misc/i-pi: パッケージ未導入
- ▶ USER/eff/ECP/Si2H6/Si2H6.ang: ストール
- ▶ USER/eff/CH4/ch4\_ionized.dynamics: ストール
- ▶ USER/misc/pimd USER/misc/grem USER/atc COUPLE ASPHERE/tri ASPHERE/line: ファイル足りない等の問題
- ▶ USER/misc/imd USER/lb: 時間がかかるためスキップ

## 注意

- ▶ モジュール名(CUDA環境以外についてはほぼ共通です)
  - ▶ CPU版: lammps/22Aug18/intel
  - ▶ GPU版: lammps/22Aug18/intel-CUDA
- ▶ ファイルはソースも含めて /local/apl/lx/lammps22Aug18 以下にあります
- ▶ CPU版の実行バイナリは /local/apl/lx/lammps22Aug18/src/lmp\_rccs です
- ▶ GPU版の実行バイナリは /local/apl/lx/lammps22Aug18/src/lmp\_rccs\_gpu です
  - ▶ GPUを使わなければCPU版と同等です。(CUDAの環境設定だけは必要ですが)
- ▶ vmd molfile plugin の実体は /local/apl/lx/vmd193/lib/plugins/LINUXAMD64/molfile ディレクトリにあります
- ▶ サンプルは /local/apl/lx/lammps22Aug18/samples ディレクトリにあります。