

LAMMPS 16Mar18 (stable release) for LX with GPU support

ウェブページ

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

バージョン

16Mar18

ビルド環境

- Intel Compiler 2015.1.133
- Intel MPI 5.0.2
- Intel MKL 11.2.1
- CUDA 8.0.61
- libjpeg-turbo 1.2.90

ビルドに必要なファイル

- lammps-stable.tar.gz (16Mar18)
- (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=16Mar18
INSTALL_PREFIX=/local/apl/lx/lammps16Mar18-CUDA8

BASEDIR=/home/users/${USER}/Software/LAMMPS/${VERSION}
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/V/\\V/g'` # molfile
VORO_VER=0.4.6 # voronoi
VORO=http://math.lbl.gov/voro++/download/dir/voro++-${VORO_VER}.tar.gz

#-----
umask 0022

./local/apl/lx/intel2015update1/bin/compilervars.sh intel64

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

tar xzf ${LAMMPS_TARBALL}
cd lammps-${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 ${PARALLEL} && \
cd ../ )
## gpu
( cd lib/gpu && \
sed -e "/^CUDA_ARCH/s/arch=sm.*/arch=sm_60/" \
-e "/^CUDA_LIB/s/$/ -L$(CUDA_HOME)/lib64/stubs/" \
-e "s/mpicxx/mpiicc/" \
Makefile.linux > Makefile.rccs && \
make -f Makefile.rccs -j ${PARALLEL} && \
sed -i -e "/^gpu_SYSPATH/s/$/ -L$(CUDA_HOME)/lib64/stubs/" \
Makefile.lammps && \
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/mpiicc/ 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 ../ )

#----

# now make lammps
cd src
make yes-all no-ext
make no-KOKKOS \
yes-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/Imp_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}
```

パッケージリスト

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

USER-ATC, USER-AWPMD, 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

テスト

- シリアルのテスト(via run_tests.py) については通過(legacyテストは除く)
- 並列テストについては、[非GPU版](#)と同じ方法で実行
- コンパイラのバージョンが違うものの、非GPU版でエラーになったテストはこちらでも同じようにエラーになりました(エラー時の数値には差があり)
 - 詳細についてはそちらのページをご覧ください。

注意点

- ファイルは /local/apl/lx/lammps16Mar18-CUDA8/ 以下にあります(/local/apl/lx/lammps16Mar18-CUDA でもアクセスできます)
- 実行バイナリ(Imp_rccs)とライブラリは src/ 以下にあります。(bin/ というシンボリックリンクからもアクセスできます)
- サンプルは samples/ ディレクトリに置いてあります。
- pythonのファイルについてもpython/以下にコピーしました
- lammpsの src/ 内にあったヘッダファイルはまとめて src/ 以下にコピーしてあります。
- latteはlammpsと組み合わせたものが正常に動作しなかったので導入せず
- 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