

## DFTB+ 23.1 (MPI)

### ウェブページ

<https://dftbplus.org/>

### バージョン

23.1

### ビルド環境

- Intel oneAPI Compiler Classic 2023.1.0
- Intel MKL 2023.1.0
- HPC-X 2.11 (Open MPI 4.1.4)

### 必要なファイル

- elsi\_interface-v2.9.1.tar.gz
- plumed-src-2.9.0.tgz
- dftbplus-23.1.tar.xz

### ビルド手順

#### ELSI

```
#!/bin/sh
ELSI_VERSION=2.9.1
ELSI_DIRNAME=elsi_interface-v${ELSI_VERSION}
PARALLEL=64
COMPILER=intel2023.1.0
MPI=ompi
BASEDIR=${HOME}/Software/elsi/${ELSI_VERSION}
ELSI_TARBALL=${BASEDIR}/${ELSI_DIRNAME}.tar.gz
WORKDIR=/gwork/users/${USER}
INSTALLDIR=/apl/elsi/${ELSI_VERSION}/${COMPILER}/${MPI}
OPENMPI=openmpi/4.1.4-hpcx/intel2023.1.0
#-----
cd $WORKDIR
module -s purge
if [ ! -d $ELSI_DIRNAME ]; then
    tar xzf ${ELSI_TARBALL}
fi

cd $ELSI_DIRNAME
if [ ! -d build ]; then
    mkdir build
fi

sed -i -e 's/mpiifort/mpifort/g' -e 's/mpiicc/mpicc/g' -e 's/mpiicpc/mpicxx/g' -e 's/-std=c99/-std=c99 -diag-disable=10441/g' -e 's/-std=c++11/-std=c++11 -diag-disable=10441/g' -e 's/intelmpi/openmpi/g' toolchains/intel_gpu.cmake

if ! grep -q BUILD_SHARED_LIBS toolchains/intel_gpu.cmake ; then
    echo 'SET(BUILD_SHARED_LIBS ON CACHE BOOL "Build ELSI as a shared library")' >> toolchains/intel_gpu.cmake
fi

if [ -f build/CMakeCache.txt ]; then
    rm -rf build/*
fi

cd build
module load ${OPENMPI}
module load mkl
module load cuda

cmake -DCMAKE_TOOLCHAIN_FILE=../toolchains/intel_gpu.cmake -DCMAKE_INSTALL_PREFIX=${INSTALLDIR} ..
```

```
make -j ${PARALLEL}
make test
make install
```

## PLUMED(MPI版)

```
#!/bin/sh
PLUMED_VERSION=2.9.0
PLUMED_DIRNAME=plumed-${PLUMED_VERSION}
PARALLEL=64
COMPILER=intel2023.1.0
MPI=mpi
BASEDIR=${HOME}/Software/plumed/${PLUMED_VERSION}
PLUMED_TARBALL=${BASEDIR}/plumed-src-${PLUMED_VERSION}.tgz
WORKDIR=/gwork/users/${USER}
INSTALLDIR=/apl/plumed/${PLUMED_VERSION}/${COMPILER}/${MPI}
OPENMPI=openmpi/4.1.4-hpcx/intel2023.1.0
#-----
cd $WORKDIR
module -s purge
if [ ! -d $PLUMED_DIRNAME ]; then
    tar xzf ${PLUMED_TARBALL}
fi

cd $PLUMED_DIRNAME
module load ${OPENMPI}
module load mkl

FC=ifort CC=icc ./configure --prefix=${INSTALLDIR} --enable-modules=all --enable-libsearch=no LIBS="-lstdc++ -qmkl -lmpi -lmpi_cxx"
CXXFLAGS="-O3 -fopenmp -diag-disable=10441"

make -j ${PARALLEL}
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${WORKDIR}/${PLUMED_DIRNAME}/src/lib
make test
make install
```

## DFTB+

```
#!/bin/sh

DFTBPLUS_VERSION=23.1
DFTBPLUS_DIRNAME=dftbplus-${DFTBPLUS_VERSION}
PARALLEL=64
BASEDIR=${HOME}/Software/dftbplus/${DFTBPLUS_VERSION}
DFTBPLUS_TARBALL=${BASEDIR}/${DFTBPLUS_DIRNAME}.tar.xz
WORKDIR=/gwork/users/${USER}
INSTALLDIR=/apl/dftb+/${DFTBPLUS_VERSION}/mpi
ELSI_DIR=/apl/elsi/2.9.1/intel2023.1.0/mpi
PLUMED_DIR=/apl/plumed/2.9.0/intel2023.1.0/mpi
OPENMPI=openmpi/4.1.4-hpcx/intel2023.1.0
LIBMBD=${HOME}/libmbd/libmbd
#-----
cd $WORKDIR
module -s purge
if [ ! -d $DFTBPLUS_DIRNAME ]; then
    tar xjf ${DFTBPLUS_TARBALL}
fi

cd $DFTBPLUS_DIRNAME
if [ ! -d _build ]; then
    mkdir _build
fi

sed -i -e 's/intelmpi/openmpi/g' sys/intel.cmake
module -s purge
```

```
module load ${OPENMPI}
module load mkl
module load cuda

if [ -f _build/CMakeCache.txt ]; then
  rm -rf _build/*
fi

rm -rf external/mbd/origin
cp -a ${LIBMBD} external/mbd/origin

FC=ifort CC=icc CMAKE_PREFIX_PATH="${LD_LIBRARY_PATH}:${ELSI_DIR}:${PLUMED_DIR}" cmake -DCMAKE_INSTALL_PREFIX=${INSTALLDIR} -
DWITH_MPI=TRUE -DWITH_GPU=TRUE -DWITH_ELSI=TRUE -DWITH_TRANSPORT=TRUE -DWITH_TBLITE=TRUE -DWITH_SOCKETS=TRUE -
DWITH_SDFTD3=TRUE -DWITH_MBD=TRUE -DWITH_PLUMED=TRUE -DWITH_CHIMES=TRUE -B _build .

cmake --build _build -- -j ${PARALLEL}

# Test
ulimit -s unlimited
./utils/get_opt externals
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${PLUMED_DIR}/lib
pushd _build; ctest -j${PARALLEL}; popd

# Install
cmake --install _build
```

## メモ

- 特筆すべき事項無し