

Siesta 4.1.5 MPI (HPE)

ウェブページ

<https://gitlab.com/siesta-project/siesta>

バージョン

4.1.5 (+ELPA 2021.05.002)

ビルド環境

- Intel oneAPI Compiler Classic 2022.2.1
- Intel MKL 2022.2.1
- HPC-X 2.13.1 (Open MPI 4.1.5)

ビルドに必要なファイル

- siesta-v4.1.5.tar.gz
- arch.make

```
.SUFFIXES:
.SUFFIXES: .f .F .o .c .a .f90 .F90

SIESTA_ARCH = rccs-intel2022-mkl

CC = mpicc
FPP = $(FC) -E -P -x c
FC = mpif90
FC_SERIAL = ifort
FFLAGS = -O2 -fPIC -march=core-avx2 -fp-model source

AR = ar
ARFLAGS_EXTRA =
RANLIB = ranlib

SYS = nag

SP_KIND = 4
DP_KIND = 8
KINDS = $(SP_KIND) $(DP_KIND)

DEFS_PREFIX =

LDFLAGS =
FCFLAGS_fixed_f = -fixed
FCFLAGS_free_f90 = -free
FPPFLAGS_fixed_F = -fixed
FPPFLAGS_free_F90 = -free

BLAS_LIBS = -mkl=sequential
LAPACK_LIBS = -mkl=sequential
SCALAPACK_LIBS = -L$(MKLROOT)/lib/intel64 -lmkl_scalapack_lp64 -lmkl_blacs_openmpi_lp64 -lmkl_intel_lp64 -lmkl_sequential -lmkl_core -lpthread -lm
-ldl

COMP_LIBS =

NETCDF_ROOT = /apl/siesta/4.1.5-mpi/exts
NETCDF_INCLFLAGS = -I$(NETCDF_ROOT)/include
NETCDF_LIBS = -L$(NETCDF_ROOT)/lib -lnetcdff -lnetcdf

MPI_INTERFACE = libmpi_f90.a
MPI_INCLUDE = .
```

```

FPPFLAGS = $(DEFS_PREFIX)-DFC_HAVE_ABORT -DSIESTA__ELPA -DMPI -DCDF -DFC_HAVE_ABORT -DFC_HAVE_FLUSH

LIBS = $(NETCDF_LIBS) $(SCALAPACK_LIBS) $(LAPACK_LIBS) $(MPI_LIBS) $(COMP_LIBS)

FFLAGS_DEBUG = -g -O1 -fp-model source

# ELPA
ELPA_ROOT = /apl/siesta/4.1.5-mpi/exts
ELPA_INCFLAGS = -DSIESTA__ELPA -I$(ELPA_ROOT)/include/elpa-2021.05.002/modules
ELPA_LIBS = -L$(ELPA_ROOT)/lib -lelpa

LIBS += $(ELPA_LIBS)
FPPFLAGS += $(ELPA_INCFLAGS)

atom.o: atom.F
    $(FC) -c $(FFLAGS_DEBUG) $(INCFLAGS) $(FPPFLAGS) $(FPPFLAGS_fixed_F) $<
state_analysis.o: state_analysis.F
    $(FC) -c $(FFLAGS_DEBUG) $(INCFLAGS) $(FPPFLAGS) $(FPPFLAGS_fixed_F) $<

.c.o:
    $(CC) -c $(CFLAGS) $(INCFLAGS) $(CPPFLAGS) $<

.F.o:
    $(FC) -c $(FFLAGS) $(INCFLAGS) $(FPPFLAGS) $(FPPFLAGS_fixed_F) $<

.F90.o:
    $(FC) -c $(FFLAGS) $(INCFLAGS) $(FPPFLAGS) $(FPPFLAGS_free_F90) $<

.f.o:
    $(FC) -c $(FFLAGS) $(INCFLAGS) $(FCFLAGS_fixed_f) $<

.f90.o:
    $(FC) -c $(FFLAGS) $(INCFLAGS) $(FCFLAGS_free_f90) $<

```

- elpa-2021.05.002.tar.gz
- netcdf-c-4.8.1.tar.gz
- netcdf-fortran-4.5.3.tar.gz

ビルド手順

ELPA 2021.05.002

```

#!/bin/sh

ELPA_VERSION=2021.05.002
INSTDIR=/apl/siesta/4.1.5-mpi/exts
WORKDIR=/gwork/users/${USER}

BASEDIR=/home/users/${USER}/Software/ELPA/${ELPA_VERSION}
TARBALL=${BASEDIR}/elpa-${ELPA_VERSION}.tar.gz

PARALLEL=12

# -----
umask 0022
ulimit -s unlimited

module purge
. ~/intel/oneapi/compiler/2022.2.1/env/vars.sh
module load compiler-rt/2022.2.1
module load mkl/2022.2.1
module load openmpi/4.1.5-hpcx/intel2022.2.1

export LANG=C
export LC_ALL=C

export FC=mpif90
export CC=mpicc
export CXX=mpicxx

```

```

export FCFLAGS=-I${MKLRROOT}/include/intel64/lp64/
# mkl_link_tool -libs -c intel_f -p no --cluster_library=scalapack
export LDFlags="-L${MKLRROOT}/lib/intel64 -lmkl_scalapack_lp64 -lmkl_blacs_openmpi_lp64 -lmkl_intel_lp64 -lmkl_sequential -lmkl_core -lpthread -lm
-ldl"

cd ${WORKDIR}
if [ -d elpa-${ELPA_VERSION} ]; then
  mv elpa-${ELPA_VERSION} elpa-erase
  rm -rf elpa-erase &
fi
tar xzf ${TARBALL}
cd elpa-${ELPA_VERSION}

./configure --prefix=${INSTDIR} --disable-avx512
make -j ${PARALLEL}
make check
make install

```

Siesta

```

#!/bin/sh

SIESTA_VERSION=4.1.5
INSTDIR=/apl/siesta/4.1.5-mpi
WORKDIR=/gwork/users/${USER}
BASEDIR=/home/users/${USER}/Software/Siesta/${SIESTA_VERSION}
TARBALL=${BASEDIR}/siesta-v${SIESTA_VERSION}.tar.gz
ARCHMAKE=${BASEDIR}-hpe/arch.make

NETCDF_C_VERSION=4.8.1
NETCDF_F_VERSION=4.5.3
BASEDIR_NETCDF=/home/users/${USER}/Software/NETCDF
TARBALL_NETCDF_C=${BASEDIR_NETCDF}/c${NETCDF_C_VERSION}/netcdf-c-${NETCDF_C_VERSION}.tar.gz
TARBALL_NETCDF_F=${BASEDIR_NETCDF}/f${NETCDF_F_VERSION}/netcdf-fortran-${NETCDF_F_VERSION}.tar.gz

PARALLEL=12 # NOTE: parallel make cannot be used for siesta

# -----
umask 0022
ulimit -s unlimited

module -s purge
. ~/intel/oneapi/compiler/2022.2.1/env/vars.sh
module -s load compiler-rt/2022.2.1
module -s load mkl/2022.2.1
module -s load openmpi/4.1.5-hpcx/intel2022.2.1

export LANG=C
export LC_ALL=C
export FC=ifort
export CC=icc

cd ${WORKDIR}
if [ -d netcdf-c-${NETCDF_C_VERSION} ]; then
  mv netcdf-c-${NETCDF_C_VERSION} netcdf-c-erase
  rm -rf netcdf-c-erase &
fi
tar xzf ${TARBALL_NETCDF_C}
cd netcdf-c-${NETCDF_C_VERSION}

./configure --prefix=${INSTDIR}/exts
make -j${PARALLEL}
# i20u2: failed on tst_charvlenbug.c
make -j${PARALLEL} check # can pass correctly
make install

```

```

export PATH="${PATH}:${INSTDIR}/exts/bin"
export CPATH="${CPATH}:${INSTDIR}/exts/include"
export LD_LIBRARY_PATH="${LD_LIBRARY_PATH}:${INSTDIR}/exts/lib"
export LIBRARY_PATH="${LIBRARY_PATH}:${INSTDIR}/exts/lib"

cd ${WORKDIR}
if [ -d netcdf-fortran-${NETCDF_F_VERSION} ]; then
  mv netcdf-fortran-${NETCDF_F_VERSION} netcdf-fortran-erase
  rm -rf netcdf-fortran-erase &
fi
tar zxf ${TARBALL_NETCDF_F}
cd netcdf-fortran-${NETCDF_F_VERSION}

./configure --prefix=${INSTDIR}/exts
make -j${PARALLEL}
make -j${PARALLEL} check
make install

cd ${INSTDIR}
if [ -d siesta-v${SIESTA_VERSION} ]; then
  mv siesta-v${SIESTA_VERSION} siesta-erase
  rm -rf siesta-erase
fi
tar zxf ${TARBALL}
mv siesta-v${SIESTA_VERSION}/* .
rmdir siesta-v${SIESTA_VERSION}

# hidoiyo...
echo >> Tests/OMM_h2o/OMM_h2o.fdf
echo >> Tests/OMM_si/OMM_si.fdf

mkdir bin # install dir

cd Obj
./Src/obj_setup.sh
cp ${ARCHMAKE} ./arch.make

# build transiesta
cd ${INSTDIR}/Obj && make transiesta
# build siesta
make clean-transiesta && make

# utils
cd ${INSTDIR}/Util
echo "m_cite.o: version.o" >> Gen-basis/Makefile
sh build_all.sh

# test siesta & transiesta
cd ${INSTDIR}/Obj/Tests
make MPI="mpirun -np 2" SIESTA="${INSTDIR}/Obj/siesta" check >& make_check.log
make MPI="mpirun -np 2" TS="${INSTDIR}/Obj/transiesta" tests-ts >& make_check_ts.log
cd ../
mv siesta ${INSTDIR}/bin
mv transiesta ${INSTDIR}/bin

```

メモ

- 時々(ある程度の時間?)固まる場合がある。原因不明。シングルノードの計算でも発生。
 - Intel MPI を使った場合はそもそも動かない。
 - 同じバージョンを使っていた前システムではこのような症状は確認されていない。
 - MKL を使うと若干固まりやすい? MKL を使わない場合の方が若干安定しているようにも見える。しかし、固まる時は固まる。
 - HPC-X 2.11 の切り替えたり、HPC-X 2.11 を使ってビルドしたりしても状況改善せず。