

## CP2K 6.1.0 for LX (GNU)

### ウェブページ

<https://www.cp2k.org/>

### バージョン

6.1.0

### ビルド環境

- gcc 5.3.1 (devtoolset-4 Software Collections)
- Intel MPI 2017.0.3
- Intel MKL 2018.0.4
- CUDA Toolkit 9.1.85 (only for GPU version)
- spglib/1.11.1

### ビルドに必要なファイル

- cp2k-6.1.0.tar.gz
- plumed-2.4.3.tgz
- (以下のスクリプト内でもいくつかファイルを入手しています)
- config.in.gf53

```
Nopt=6
lmax=7
Nrun=3
FC_comp="gfortran -ffree-form"
FCFLAGS_OPT="-O3 -ffast-math -funroll-loops -ftree-vectorize -fno-omit-frame-pointer"
FCFLAGS_NATIVE="-march=native"
makefile_name="Makefile"
```

- install\_openmpi\_gf.patch

```
--- scripts/install_openmpi.sh.org 2018-11-20 10:08:10.000000000 +0900
+++ scripts/install_openmpi.sh 2018-11-20 10:07:47.000000000 +0900
@@ -56,13 +56,13 @@
;;
__SYSTEM__
echo "===== Finding OpenMPI from system paths ====="
- check_command mpirun "openmpi"
- check_command mpicc "openmpi"
- check_command mpif90 "openmpi"
- check_command mpic++ "openmpi"
- check_lib -lmpi "openmpi"
- add_include_from_paths OPENMPI_CFLAGS "mpi.h" $INCLUDE_PATHS
- add_lib_from_paths OPENMPI_LDFLAGS "libmpi.*" $LIB_PATHS
+ #check_command mpirun "openmpi"
+ #check_command mpicc "openmpi"
+ #check_command mpif90 "openmpi"
+ #check_command mpic++ "openmpi"
+ #check_lib -lmpi "openmpi"
+ #add_include_from_paths OPENMPI_CFLAGS "mpi.h" $INCLUDE_PATHS
+ #add_lib_from_paths OPENMPI_LDFLAGS "libmpi.*" $LIB_PATHS
;;
__DONTUSE__
@@ -90,24 +90,26 @@
else
mpi_bin=mpirun
fi
- # check openmpi version as reported by mpirun
- raw_version=$(($mpi_bin --version 2>&1 | \
```

```

-         grep "(Open MPI)" | awk '{print $4}'
- major_version=$(echo $raw_version | cut -d '.' -f 1)
- minor_version=$(echo $raw_version | cut -d '.' -f 2)
- # old versions required -lmpi_cxx to link cxx code, new version don't
- if [ $major_version -gt 1 ] ; then
+ ## check openmpi version as reported by mpirun
+ #raw_version=$(($mpi_bin --version 2>&1 | \
+ #         grep "(Open MPI)" | awk '{print $4}'
+ #major_version=$(echo $raw_version | cut -d '.' -f 1)
+ #minor_version=$(echo $raw_version | cut -d '.' -f 2)
+ ## old versions required -lmpi_cxx to link cxx code, new version don't
+ #if [ $major_version -gt 1 ] ; then
+     OPENMPI_LIBS="-lmpi"
- else
- else
-     OPENMPI_LIBS="-lmpi -lmpi_cxx"
- fi
+ #else
+ # OPENMPI_LIBS="-lmpi -lmpi_cxx"
+ #fi
+ # old versions didn't support MPI 3, so adjust __MPI_VERSION accordingly (needed e.g. for pexsi)
- if [ $major_version -lt 1 ] || \
- [ $major_version -eq 1 -a $minor_version -lt 7 ] ; then
-     mpi2_dflags="-D__MPI_VERSION=2"
- else
+ #if [ $major_version -lt 1 ] || \
+ # [ $major_version -eq 1 -a $minor_version -lt 7 ] ; then
+ #     mpi2_dflags="-D__MPI_VERSION=2"
+ #else
+     mpi2_dflags=""
- fi
+ #fi
+ OPENMPI_CFLAGS="-I${I_MPI_ROOT}/include64/gfortran/5.1.0 -I${I_MPI_ROOT}/include64"
+ OPENMPI_LDFLAGS="-L${I_MPI_ROOT}/lib64 -Wl,-rpath=${I_MPI_ROOT}/lib64"
+ cat <<EOF >> "${BUILDDIR}/setup_openmpi"
export OPENMPI_CFLAGS="${OPENMPI_CFLAGS}"
export OPENMPI_LDFLAGS="${OPENMPI_LDFLAGS}"

```

- install\_mkl\_gf\_intelmpi.patch

```

--- scripts/install_mkl.sh.org 2018-11-12 14:44:45.000000000 +0900
+++ scripts/install_mkl.sh 2018-11-12 14:50:40.000000000 +0900
@@ -66,7 +66,7 @@
     fi
done
# set the correct lib flags from MLK link adviser
- MKL_LIBS="-Wl,--start-group ${mkl_lib_dir}/libmkl_gf_lp64.a ${mkl_lib_dir}/libmkl_core.a ${mkl_lib_dir}/libmkl_sequential.a"
+ MKL_LIBS="-lmkl_gf_lp64 -lmkl_sequential -lmkl_core"
# check optional libraries
if [ $MPI_MODE != no ] ; then
    enable_mkl_scalapack="__TRUE__"
@@ -74,11 +74,11 @@
    case $MPI_MODE in
        mpich)
            mkl_optional_libs="$mkl_optional_libs libmkl_blacs_lp64.a"
-            mkl_blacs_lib="libmkl_blacs_lp64.a"
+            mkl_blacs_lib="-lmkl_blacs_lp64"
            ;;
        openmpi)
            mkl_optional_libs="$mkl_optional_libs libmkl_blacs_openmpi_lp64.a"
-            mkl_blacs_lib="libmkl_blacs_openmpi_lp64.a"
+            mkl_blacs_lib="-lmkl_blacs_intelmpi_lp64"
            ;;
    *)
        enable_mkl_scalapack="__FALSE__"

```

```

@@ -91,13 +91,13 @@
done
if [ $enable_mkl_scalapack = "__TRUE_" ]; then
    echo "Using MKL provided ScaLAPACK and BLACS"
-   MKL_LIBS="${mkl_lib_dir}/libmkl_scalapack_lp64.a ${MKL_LIBS} ${mkl_lib_dir}/${mkl_blacs_lib}"
+   MKL_LIBS="-lmkl_scalapack_lp64 ${MKL_LIBS} ${mkl_blacs_lib}"
    fi
else
    enable_mkl_scalapack="__FALSE_"
fi
- MKL_LIBS="${MKL_LIBS} -Wl,--end-group -lpthread -lm -ldl"
- MKL_CFLAGS="${MKL_CFLAGS} -I${MKLROOT}/include"
+ MKL_LIBS="-L${mkl_lib_dir} ${MKL_LIBS} -lpthread -lm -Wl,-rpath,${mkl_lib_dir}"
+ MKL_CFLAGS="${MKL_CFLAGS} -I${MKLROOT}/include"

# write setup files
cat <<EOF > "${BUILDDIR}/setup_mkl"

```

## ビルド手順

### libgrid.a のビルド

```

#!/bin/sh
#PBS -l select=1:ncpus=40:mpiprocs=40:ompthreads=1:jobtype=small
#PBS -l walltime=03:00:00

CP2KVER=6.1.0
CP2KTARBALL=/home/users/qf7/Software/CP2K/${CP2KVER}/cp2k-6.1.0.tar.gz
WORKDIR=/home/users/qf7/Software/CP2K/${CP2KVER}/libgrid

CONFIGIN=/home/users/qf7/Software/CP2K/${CP2KVER}/libgrid/config.in.gf53
PARALLEL=36

#-----
umask 0022
export LC_ALL=C

module purge
module load scl/devtoolset-4

cd $WORKDIR
if [ -d cp2k-${CP2KVER} ]; then
    mv cp2k-${CP2KVER} cp2k-erase
    rm -rf cp2k-erase &
fi
if [ -d cp2k-${CP2KVER}-gf ]; then
    mv cp2k-${CP2KVER}-gf cp2k-gf-erase
    rm -rf cp2k-gf-erase &
fi

tar zxvf ${CP2KTARBALL}
mv cp2k-${CP2KVER} cp2k-${CP2KVER}-gf53
cd cp2k-${CP2KVER}-gf53/tools/autotune_grid
mv config.in config.in.org
cp $CONFIGIN ./config.in
tar zxf data.tgz

./generate_makefile.sh

make -j ${PARALLEL} all_gen
make -j ${PARALLEL} all_run
make gen_best
make libgrid.a

```

### cp2k のビルド

```

#!/bin/sh

INSTDIR=/local/apl/lx/cp2k610-gnu
WORKDIR=/work/users/${USER}

VERSION=6.1.0

TARBALL=/home/users/${USER}/Software/CP2K/${VERSION}/cp2k-${VERSION}.tar.gz
LIBGRID=/home/users/${USER}/Software/CP2K/${VERSION}/libgrid/cp2k-${VERSION}-gf53/tools/autotune_grid
PATCH_INST_OPENMPI=/home/users/${USER}/Software/CP2K/${VERSION}/install_openmpi_gf.patch
PATCH_INST_MKL=/home/users/${USER}/Software/CP2K/${VERSION}/install_mkl_gf_intelmpi.patch

PLUMED_VERSION=2.4.3
PLUMED_TARBALL=/home/users/${USER}/Software/PLUMED/${PLUMED_VERSION}/plumed-${PLUMED_VERSION}.tgz

PARALLEL=12

# NOTE: Some of header and library paths are defined in CPATH and
#       LIBRARY_PATH environment variables from modulefiles.

#-----
umask 0022
export LANG=""
export LC_ALL=C

module purge
module load scl/devtoolset-4
module load mpi/intelmpi/2017.4.262
module load mkl/2018.0.4
module load spglib/1.11.1

cd $INSTDIR
if [ -d cp2k-${VERSION} ]; then
  mv cp2k-${VERSION} cp2k-erase
  rm -rf cp2k-erase &
fi

tar xzf ${TARBALL}
mv cp2k-${VERSION}/* .
rm -rf cp2k-${VERSION}/.{dockerignore,.gitignore}
rmdir cp2k-${VERSION}
#rm -rf tools/autotune_grid
#cp -rf ${LIBGRID} tools/autotune_grid
cp ${LIBGRID}/libgrid.a tools/autotune_grid

# plumed
mkdir ${INSTDIR}/plumed
cd $WORKDIR
if [ -d plumed-${PLUMED_VERSION} ]; then
  mv plumed-${PLUMED_VERSION} plumed-erase
  rm -rf plumed-erase &
fi

tar xzf ${PLUMED_TARBALL}
cd plumed-${PLUMED_VERSION}
./configure --prefix=${INSTDIR}/plumed
make -j ${PARALLEL}
make check
make install

# cp2k toolchain
cd ${INSTDIR}/tools/toolchain
sed -e "s/blacs_openmpi/blacs_intelmpi/" scripts/install_mkl.sh > scripts/install_mkl.sh.impi
chmod 755 scripts/install_mkl.sh.impi

```

```

patch -p0 < $PATCH_INST_OPENMPI
patch -p0 < $PATCH_INST_MKL

MPICXX=mpicxx \
./install_cp2k_toolchain.sh --math-mode=mkl \
--mpi-mode=openmpi \
--with-elpa=install \
--with-cmake=system \
--with-mpich=no \
--with-openmpi=system \
--with-reflapack=no \
-j ${PARALLEL}

# cheat the script
mv -f scripts/install_mkl.sh.impi scripts/install_mkl.sh

MPICXX=mpicxx \
./install_cp2k_toolchain.sh --math-mode=mkl \
--mpi-mode=openmpi \
--with-elpa=install \
--with-ptscotch=install \
--with-parmetis=install \
--with-superlu=install \
--with-pexsi=install \
--with-quip=install \
--with-cmake=system \
--with-mpich=no \
--with-openmpi=system \
--with-reflapack=no \
-j ${PARALLEL}

LIBGRID_ESC=`echo ${INSTDIR}/tools/autotune_grid | sed -e 's/V/\\\\V/g'`

# finish building toolchain, copy arch file to the proper place
## modify arch for cpu-only version
sed -e "/^CFLAGS    =/s/CFLAGS    =/CFLAGS    = -O2 /" \
-e "/^FCFLAGS    =/s/FCFLAGS    =/FCFLAGS    = -O2 /" \
-e "/^DFLAGS /s/\$/ -D_HAS_LIBGRID -D_SPLIB -D_PLUMED2/" \
-e "/^LDFLAGS /s/\$/ -L${LIBGRID_ESC}"/" \
-e "/^LIBS /s/\$/ -lgrid -lsymspg -lz -lgsl/" \
install/arch/local.psm > ${INSTDIR}/arch/rccs.psm
echo "include ${INSTDIR}/plumed/lib/plumed/src/lib/Plumed.inc" >> ${INSTDIR}/arch/rccs.psm
echo "EXTERNAL_OBJECTS=\$(PLUMED_STATIC_DEPENDENCIES)" >> ${INSTDIR}/arch/rccs.psm

## modify arch for cpu-gpu version
sed -e "/^CFLAGS    =/s/CFLAGS    =/CFLAGS    = -O2 /" \
-e "/^FCFLAGS    =/s/FCFLAGS    =/FCFLAGS    = -O2 /" \
-e "/^DFLAGS /s/\$/ -D_HAS_LIBGRID -D_SPLIB -D_PLUMED2 -D_ACC -D_DBCSR_ACC -D_PW_CUDA/" \
-e "/^LDFLAGS /s/\$/ -L${LIBGRID_ESC}"/" \
-e "/^LIBS /s/\$/ -lgrid -lsymspg -lz -lgsl -lcudart -lcufft -lcublas/" \
install/arch/local.psm > ${INSTDIR}/arch/rccs_cuda.psm
echo "include ${INSTDIR}/plumed/lib/plumed/src/lib/Plumed.inc" >> ${INSTDIR}/arch/rccs_cuda.psm
echo "EXTERNAL_OBJECTS=\$(PLUMED_STATIC_DEPENDENCIES)" >> ${INSTDIR}/arch/rccs_cuda.psm
echo "NVCC      = nvcc -D_GNUC__=5 -D_GNUC_MINOR__=3 -Xcompiler=--std=gnu++98" >> ${INSTDIR}/arch/rccs_cuda.psm
echo "NVFLAGS   = -gencode=arch=compute_60,code=sm_60 -gencode=arch=compute_70,code=sm_70 \$(DFLAGS)" >>
${INSTDIR}/arch/rccs_cuda.psm

#--

cd ${INSTDIR}/makefiles
make -j ${PARALLEL} ARCH=rccs VERSION=psmp

module load cuda/9.1
make -j ${PARALLEL} ARCH=rccs_cuda VERSION=psmp

```

## テスト

### plumed

```
+++++
+ Final report:
+ 248 tests performed, 161 tests not applicable
+ 0 errors found
+ Well done!!
+++++
```

### cp2k

テスト結果は /local/apl/lx/cp2k610-gnu/regtesting/rccs/psmp もしくは /local/apl/lx/cp2k610-gnu/regtesting/rccs\_cuda/psmp に残っています。

GPU版以外は ccfep 上でテストを実行しています。

```
#!/bin/sh

export LC_ALL=C
module purge
module load scl/devtoolset-4
module load mpi/intelmpi/2017.4.262
module load mkl/2018.0.4
module load spglib/1.11.1

CP2K=/local/apl/lx/cp2k610-gnu/
CP2K_ARCH=rccs
CP2K_VER=psmp
PARALLEL=16

ulimit -s unlimited

cd ${CP2K}/regtesting/${CP2K_ARCH}/${CP2K_VER}
rm -rf LAST-${CP2K_ARCH}-${CP2K_VER}

# serial test
../../tools/regtesting/do_regtest \
  -nobuild \
  -nosvn \
  -arch ${CP2K_ARCH} \
  -version ${CP2K_VER} \
  -mpiranks 1 \
  -ompthreads 1 \
  -jobmaxtime 120 \
  -cp2kdir ../../ \
  -maxtasks ${PARALLEL} >& regtest_mpi1_omp1.log
rm -rf LAST-${CP2K_ARCH}-${CP2K_VER}

# omp test
../../tools/regtesting/do_regtest \
  -nobuild \
  -nosvn \
  -arch ${CP2K_ARCH} \
  -version ${CP2K_VER} \
  -mpiranks 1 \
  -ompthreads 2 \
  -jobmaxtime 120 \
  -cp2kdir ../../ \
  -maxtasks ${PARALLEL} >& regtest_mpi1_omp2.log
rm -rf LAST-${CP2K_ARCH}-${CP2K_VER}

# mpi test
../../tools/regtesting/do_regtest \
  -nobuild \
  -nosvn \
```

```

-arch ${CP2K_ARCH} \
-version ${CP2K_VER} \
-mpiranks 2 \
-ompthreads 1 \
-jobmaxtime 120 \
-cp2kdir .././.\
-maxtasks ${PARALLEL} >& regtest_mpi2_omp1.log
rm -rf LAST-${CP2K_ARCH}-${CP2K_VER}

# mpi/openmp test
../././tools/regtesting/do_regtest \
-nobuild \
-nosvn \
-arch ${CP2K_ARCH} \
-version ${CP2K_VER} \
-mpiranks 2 \
-ompthreads 2 \
-jobmaxtime 120 \
-cp2kdir .././.\
-maxtasks ${PARALLEL} >& regtest_mpi2_omp2.log
rm -rf LAST-${CP2K_ARCH}-${CP2K_VER}

# yet another mpi test
../././tools/regtesting/do_regtest \
-nobuild \
-nosvn \
-arch ${CP2K_ARCH} \
-version ${CP2K_VER} \
-mpiranks 8 \
-ompthreads 1 \
-jobmaxtime 120 \
-cp2kdir .././.\
-maxtasks ${PARALLEL} >& regtest_mpi8_omp1.log
rm -rf LAST-${CP2K_ARCH}-${CP2K_VER}

# yet another mpi/openmp test
../././tools/regtesting/do_regtest \
-nobuild \
-nosvn \
-arch ${CP2K_ARCH} \
-version ${CP2K_VER} \
-mpiranks 8 \
-ompthreads 2 \
-jobmaxtime 120 \
-cp2kdir .././.\
-maxtasks ${PARALLEL} >& regtest_mpi8_omp2.log
rm -rf LAST-${CP2K_ARCH}-${CP2K_VER}

```

## ■ テスト結果: MPI1 - OMP1

```

----- Summary -----
Number of FAILED tests 0
Number of WRONG tests 0
Number of CORRECT tests 3014
Number of NEW tests 14
Total number of tests 3028
GREPME 0 0 3014 14 3028 X
-----

```

## ■ テスト結果: MPI1 - OMP2

```

----- Summary -----
Number of FAILED tests 0
Number of WRONG tests 0
Number of CORRECT tests 3014
Number of NEW tests 14

```

```
Total number of tests 3028
GREPME 0 0 3014 14 3028 X
-----
```

#### ■ テスト結果: MPI2 - OMP1

```
----- Summary -----
Number of FAILED tests 0
Number of WRONG tests 0
Number of CORRECT tests 3059
Number of NEW tests 19
Total number of tests 3078
GREPME 0 0 3059 19 3078 X
-----
```

#### ■ テスト結果: MPI2 - OMP2

```
----- Summary -----
Number of FAILED tests 0
Number of WRONG tests 0
Number of CORRECT tests 3059
Number of NEW tests 19
Total number of tests 3078
GREPME 0 0 3059 19 3078 X
-----
```

#### ■ テスト結果: MPI8 - OMP1

```
----- Summary -----
Number of FAILED tests 0
Number of WRONG tests 0
Number of CORRECT tests 3027
Number of NEW tests 16
Total number of tests 3043
GREPME 0 0 3027 16 3043 X
-----
```

#### ■ テスト結果: MPI8 - OMP2

```
----- Summary -----
Number of FAILED tests 0
Number of WRONG tests 1
Number of CORRECT tests 3026
Number of NEW tests 16
Total number of tests 3043
GREPME 0 1 3026 16 3043 X
-----
```

- QMMM/SE/regtest-force-mixing/Lysozyme\_small\_NVT.inp : WRONG RESULT TEST 2

#### ■ テスト結果: GPU版 MPI2 - OMP2 (@ccca)

```
----- Summary -----
Number of FAILED tests 0
Number of WRONG tests 0
Number of CORRECT tests 3060
Number of NEW tests 19
Total number of tests 3079
GREPME 0 0 3060 19 3079 X
-----
```

### ベンチマーク

tests/QS/benchmark/H2O-64.inp を利用。(時間は grep "CP2K " \*.log で表示される値から)

```
jobtype 総コア数 MPI OMP GPU elapse(sec)
        (ノード数)
```



core	18 (1)	18	1	-	129.823
small	40 (1)	40	1	-	78.827
small	160 (4)	160	1	-	36.681
small	160 (4)	80	2	-	38.323
small	160 (4)	32	5	-	36.854
gpu	12 (1)	12	1	1	207.190

## 雑多な情報

- gcc6(+openmpi)でエラーがあったため、gcc5を利用
  - (intelmpi ならば大丈夫だったかもしれない)
  - openmpi を使うとテストのエラー数が明確に増加する
- (intel 2017 の mkl でも大丈夫だったかもしれないが未検証)
- (openmpi-mkl2017の組み合わせはやはりダメに見える)