CP2K 8.2

ウェブページ

https://www.cp2k.org/

バージョン

8.2 (8.2.0)

ビルド環境

- Intel Parallel Studio 2020 Update 2 (MPI のみ)
- GCC 9.3.1 (devtoolset-9)
- cmake 3.16.3

ビルドに必要なファイル

- cp2k-8.2.tar.bz2
- tc_install_fftw3.sh.diff (libvdwxc のビルドエラー回避)

```
--- install_fftw.sh.org 2021-06-09 09:56:30.000000000 +0900
+++ install_fftw.sh     2021-06-09 09:56:56.000000000 +0900
@@ -96,10 +96,10 @@
   # we may also want to cover FFT_SG
   cat << EOF >> "${BUILDDIR}/setup_fftw"
 export FFTW3_INCLUDES="${FFTW_CFLAGS}"
-export FFTW3_LIBS="${FFTW_LIBS}"
+export FFTW3_LIBS="-L${pkg_install_dir}/lib ${FFTW_LIBS}"
 export FFTW_CFLAGS="${FFTW_CFLAGS}"
 export FFTW_LDFLAGS="${FFTW_LDFLAGS}"
-export FFTW_LIBS="${FFTW_LIBS}"
+export FFTW_LIBS="-L${pkg_install_dir}/lib ${FFTW_LIBS}"
 export CP_DFLAGS="\${CP_DFLAGS} -D__FFTW3 IF_COVERAGE(IF_MPI(|-U__FFTW3)|)"
 export CP_CFLAGS="\${CP_CFLAGS} ${FFTW_CFLAGS}"
 export CP_LDFLAGS="\${CP_LDFLAGS} ${FFTW_LDFLAGS}"
```

- tc_install_plumed.sh.gcc.diff (パス指定の簡略化のため)

```
--- install_plumed.sh.org       2021-06-09 11:57:44.000000000 +0900
+++ install_plumed.sh   2021-06-09 11:58:39.000000000 +0900
@@ -83,7 +84,7 @@
 esac

 if [ "$with_plumed" != "__DONTUSE__" ]; then
-  PLUMED_LIBS='-lplumed -ldl -lstdc++ -lz -ldl'
+  PLUMED_LIBS='-lplumedKernel -lplumed -ldl -lstdc++ -lz -ldl'
   if [ "$with_plumed" != "__SYSTEM__" ]; then
     cat << EOF > "${BUILDDIR}/setup_plumed"
 prepend_path LD_LIBRARY_PATH "$pkg_install_dir/lib"
```

ビルド手順

```
#!/bin/sh

INSTDIR=/local/apl/lx/cp2k820

GITHUB_VERSION=8.2.0
VERSION=8.2

SOURCE_ROOT=/home/users/${USER}/Software/CP2K/${GITHUB_VERSION}

TARBALL=${SOURCE_ROOT}/cp2k-${VERSION}.tar.bz2
#export RCCS_COSMA_TARBALL=${SOURCE_ROOT}/COSMA-vrccs.tar.gz
```

```
TC_PATCH_3_1=${SOURCE_ROOT}/tc_install_fftw3.sh.diff
#TC_PATCH_4_1=${SOURCE_ROOT}/tc_install_cosma.sh.diff
TC_PATCH_6_1=${SOURCE_ROOT}/tc_install_plumed.sh.gcc.diff

PARALLEL=12

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

module purge
module load mpi/intelmpi/2019.8.254
module load scl/devtoolset-9
module load cmake/3.16.3

cd $INSTDIR
if [ -d cp2k-${VERSION} ]; then
  mv cp2k-${VERSION} cp2k-erase
  rm -rf cp2k-erase &
fi
tar jxf ${TARBALL}
sleep 5
mv cp2k-${VERSION}/* .
sleep 5
rm -f cp2k-${VERSION}/.dockerignore
rmdir cp2k-${VERSION}

cd ${INSTDIR}/tools/toolchain

# apply patches
cd scripts
cd stage3 && patch < ${TC_PATCH_3_1} && cd -
#cd stage4 && patch < ${TC_PATCH_4_1} && cd -
cd stage6 && patch < ${TC_PATCH_6_1} && cd -
cd ../

export CC=gcc
export CXX=g++
export FC=gfortran
export MPICC=mpicc
export MPICXX=mpicxx
export MPIFC=mpif90

./install_cp2k_toolchain.sh --mpi-mode=intelmpi \
                            --math-mode=openblas \
                            --with-cmake=system \
                            --with-openmpi=no \
                            --with-mpich=no \
                            --with-intelmpi=system \
                            --with-libxc=install \
                            --with-libint=install \
                            --with-fftw=install \
                            --with-acml=no \
                            --with-mkl=no \
                            --with-openblas=install \
                            --with-scalapack=install \
                            --with-libsmm=no \
                            --with-libxsmm=install \
                            --with-elpa=no \
                            --with-ptscotch=install \
                            --with-superlu=install \
                            --with-pexsi=install \
                            --with-quip=install \
```

```
                --with-plumed=install \
                --with-sirius=no \
                --with-gsl=install \
                --with-libvdwxc=install \
                --with-spglib=install \
                --with-hdf5=install \
                --with-spfft=install \
                --with-cosma=no \
                --with-libvori=install \
                -j ${PARALLEL}

sed -e "s/-Werror / /g" install/arch/local.psmp > ../../arch/rccs.psmp

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

テスト

```
#!/bin/sh
#PBS -l select=1:ncpus=16:mpiprocs=16:ompthreads=1:jobtype=core
#PBS -l walltime=12:00:00

export LC_ALL=C
export LANG=""
export OMP_STACKSIZE=64M

# gcc9
module purge
module load mpi/intelmpi/2019.8.254
module load scl/devtoolset-9
module load cmake/3.16.3
CP2K=/local/apl/lx/cp2k820

CP2K_ARCH=rccs
CP2K_VER=psmp
TIMEOUT=600
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 \
      -arch ${CP2K_ARCH} \
      -version ${CP2K_VER} \
      -mpiranks 1 \
      -ompthreads 1 \
      -jobmaxtime ${TIMEOUT} \
      -cp2kdir ../../../ \
      -maxtasks ${PARALLEL} >& regtest_mpi1_omp1.log
rm -rf LAST-${CP2K_ARCH}-${CP2K_VER}

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

```
# mpi test
../../../tools/regtesting/do_regtest \
       -nobuild \
       -arch ${CP2K_ARCH} \
       -version ${CP2K_VER} \
       -mpiranks 2 \
       -ompthreads 1 \
       -jobmaxtime ${TIMEOUT} \
       -cp2kdir ../../../ \
       -maxtasks ${PARALLEL} >& regtest_mpi2_omp1.log
rm -rf LAST-${CP2K_ARCH}-${CP2K_VER}

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

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

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

テスト結果

```
[root@ccfep4 psmp]# grep "GREPME" regtest_mpi*
regtest_mpi1_omp1.log:GREPME 0 0 3396 0 3396 X
regtest_mpi1_omp2.log:GREPME 0 0 3396 0 3396 X
regtest_mpi2_omp1.log:GREPME 0 0 3447 0 3447 X
regtest_mpi2_omp2.log:GREPME 0 0 3447 0 3447 X
regtest_mpi8_omp1.log:GREPME 0 10 3407 0 3417 X
regtest_mpi8_omp2.log:GREPME 0 10 3407 0 3417 X
```

- 8 MPI 時のみいくつかのテストで数値エラー
  - MPI*8, OMP*1
    - QS/regtest-mp2-lr/H2O-mp2-gpw-lr.inp: ENERGY| Total FORCE_EVAL : ref = -16.964068900743456 new = -16.964157811107025
    - QS/regtest-mp2-grad/H2O_grad_mme.inp: ENERGY| Total FORCE_EVAL : ref = -16.766973106034889 new = -16.766973179928165
    - QS/regtest-gpw-4/H2O-debug-5.inp: DIPOLE : CheckSum = : ref = -0.535129866059 new = -

0.535129747930E+00
        - QS/regtest-gpw-4/H2O-debug-6.inp: DIPOLE : CheckSum = : ref = -0.535125994114 new = -0.535125875984E+00
        - QS/regtest-mp2-4/H2O_NO_HFX.inp: ENERGY| Total FORCE_EVAL : ref = -17.253519557463612 new = -17.291360866609697
        - QS/regtest-rma-3D/H2O-32-dftb-ls-2_mult.inp: ENERGY| Total FORCE_EVAL : ref = -32.574187310759356 new = -32.563908850166179
        - QS/regtest-rma-3D/H2O-32-dftb-ls-2.inp: ENERGY| Total FORCE_EVAL : ref = -32.574187310759356 new = -32.563908850166179
        - QS/regtest-rma-3D/H2O-OT-ASPC-1.inp: Total energy: : ref = -17.13993294716182 new = -17.13993294752104
        - QS/regtest-rma-3D/H2O-6.inp: Total energy: : ref = -17.14603641576940 new = -17.14603641519600
        - QS/regtest-mp2-2/H2O-02.inp: ENERGY| Total FORCE_EVAL : ref = -17.157097357548857 new = -17.181101307832947
    - MPI*8, OMP*2
        - QS/regtest-mp2-lr/H2O-mp2-gpw-lr.inp: ENERGY| Total FORCE_EVAL : ref = -16.964068900743456 new = -16.964157811107032
        - QS/regtest-mp2-grad/H2O_grad_mme.inp: ENERGY| Total FORCE_EVAL : ref = -16.766973106034889 new = -16.766973179928165
        - QS/regtest-gpw-4/H2O-debug-5.inp.out: DIPOLE : CheckSum = : ref = -0.535129866059 new = -0.535129747930E+00
        - QS/regtest-gpw-4/H2O-debug-6.inp: DIPOLE : CheckSum = : ref = -0.535125994114 new = -0.535125875984E+00
        - QS/regtest-mp2-4/H2O_NO_HFX.inp: ENERGY| Total FORCE_EVAL : ref = -17.253519557463612 new = -17.291360866609637
        - QS/regtest-rma-3D/H2O-32-dftb-ls-2_mult.inp: ENERGY| Total FORCE_EVAL : ref = -32.574187310759356 new = -32.563908850166179
        - QS/regtest-rma-3D/H2O-32-dftb-ls-2.inp: ENERGY| Total FORCE_EVAL : ref = -32.574187310759356 new = -32.563908850166179
        - QS/regtest-rma-3D/H2O-OT-ASPC-1.inp: Total energy: : ref = -17.13993294716182 new = -17.13993294752103
        - QS/regtest-rma-3D/H2O-6.inp: Total energy: : ref = -17.14603641576940 new = -17.14603641519601
        - QS/regtest-mp2-2/H2O-02.inp: ENERGY| Total FORCE_EVAL : ref = -17.157097357548857 new = -17.181101307832943

ベンチマーク

7.1.0 と同じように H2O-64.inp を利用。(時間は grep "CP2K " *.log で表示される値から)
20 回実行し、最初の 1 回を除いた平均値。(初回は速度が安定しないため)

jobtype	総コア数 (ノード数)	MPI	OMP	GPU	elapse(sec)
core	18 (1)	18	1	-	59.96
small	40 (1)	40	1	-	46.00
small	80 (2)	80	1	-	29.05
small	160 (4)	160	1	-	21.61

- 前バージョンほど OpenMP 並列が有効では無い。
- (jobtype=small 1 ノードの時は 7.1.0 intel 版の方がわずかに速い可能性がありそうです。)
- core についてはノードを全て確保した上で 18 コアで実行。(実行ノードに他のジョブがある場合は速度がこの数字よりも落ちます。)

メモ

- インテルコンパイラよりも gcc の方が速度が出る。7.1 の時とは違う傾向。
- 今回は OpenMP では性能が出にくいように見える。
- GPU については未検証。
- ELPA を有効にすると BFGS 計算に問題が出る場合があるため今回は回避。
    - regtest の場合 1-2 MPI では問題無いものの、8 MPI 時に大量の失敗が発生する。
    - 参考: https://groups.google.com/g/cp2k/c/BAvvW_qGG2I
- cosma については 8.1 のような速度低下問題は無いが顕著な速度向上もなさそうなため、今回は回避。
    - toolchain デフォルトの COSMA-v2.5.0 では今回使ったビルド環境では正常にビルドできず。インテルコンパイラを使った場合も同様。COSMA 本体の unittest も通らない。
    - COSMA-v2.5.1 に置き換えても問題は変わらず。

- github 上の最新(最新コミット 2021/6/18)では問題無く動作。が、大きな速度上昇は認められない上、バージョン未定義のスナップショットとなるためあまり好ましくないと判断。そのため今回は cosma 自体を回避。
- tc_install_cosma.sh.diff: COSMA 最新を使う場合のパッチ。git clone した上で名前を変えてアーカイブ化した COSMA-vrccs.tar.gz を使う。(上記インストールスクリプトに記述あり)

```
--- install_cosma.sh.org        2021-06-21 13:52:52.000000000 +0900
+++ install_cosma.sh    2021-06-21 13:58:10.000000000 +0900
@@ -9,7 +9,7 @@
 [ "${BASH_SOURCE[0]}" ] && SCRIPT_NAME="${BASH_SOURCE[0]}" || SCRIPT_NAME=$0
 SCRIPT_DIR="$(cd "$(dirname "$SCRIPT_NAME")/.." && pwd -P)"

-cosma_ver="2.5.0"
+cosma_ver="rccs"
 cosma_sha256="7f68bb0ee5c80f9b8df858afcbd017ad4ed87ac09439d13d7d890844dbdd3d54"
 source "${SCRIPT_DIR}"/common_vars.sh
 source "${SCRIPT_DIR}"/tool_kit.sh
@@ -37,9 +37,7 @@
       if [ -f COSMA-v${cosma_ver}.tar.gz ]; then
         echo "COSMA-v${cosma_ver}.tar.gz is found"
       else
-        download_pkg ${DOWNLOADER_FLAGS} ${cosma_sha256} \
-          "https://github.com/eth-cscs/COSMA/releases/download/v${cosma_ver}/COSMA-v${cosma_ver}.tar.gz" \
-          -o COSMA-v${cosma_ver}.tar.gz
+        cp ${RCCS_COSMA_TARBALL} .
       fi
       echo "Installing from scratch into ${pkg_install_dir}"
       [ -d COSMA-${cosma_ver} ] && rm -rf COSMA-${cosma_ver}
```