CP2K 2023.1

ウェブページ

https://www.cp2k.org/

バージョン

2023.1

ビルド環境

- GCC 11.2.1 (gcc-toolset-11)
- HPC-X 2.11 (Open MPI 4.1.4)

ビルドに必要なファイル

- cp2k-2023.1.tar.gz
- patch-openblas.diff

```
--- scripts/stage2/install_openblas.sh.org      2023-01-17 10:19:31.000000000 +0900
+++ scripts/stage2/install_openblas.sh  2023-01-17 10:20:21.000000000 +0900
@@ -47,6 +47,7 @@
       [ -d OpenBLAS-${openblas_ver} ] && rm -rf OpenBLAS-${openblas_ver}
       tar -zxf ${openblas_pkg}
       cd OpenBLAS-${openblas_ver}
+      sed -i -e "/NOTPARALLEL/s/$/ shared/" Makefile

       # First attempt to make openblas using auto detected
       # TARGET, if this fails, then make with forced
```

- patch-plumed.diff

```
--- scripts/stage6/install_plumed.sh.org        2023-04-04 15:56:54.000000000 +0900
+++ scripts/stage6/install_plumed.sh    2023-04-04 15:57:28.000000000 +0900
@@ -90,7 +90,7 @@
 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"
```

- patch-spla-nogpu.diff

```
--- scripts/stage8/install_spla.sh.org  2023-04-04 15:31:42.000000000 +0900
+++ scripts/stage8/install_spla.sh      2023-04-04 15:37:09.000000000 +0900
@@ -204,6 +204,11 @@
 export CP_LDFLAGS="\${CP_LDFLAGS} IF_CUDA(${SPLA_CUDA_LDFLAGS}|${SPLA_LDFLAGS})"
 EOF
   fi
+  if [ "$ENABLE_CUDA" != "__TRUE__" -a "$ENABLE_HIP" != "__TRUE__" ]; then
+    cat << EOF >> "${BUILDDIR}/setup_spla"
+export CP_LDFLAGS="\${CP_LDFLAGS} ${SPLA_LDFLAGS}"
+EOF
+  fi
   cat "${BUILDDIR}/setup_spla" >> $SETUPFILE
 fi
```

ビルド手順

```
#!/bin/sh
```

```
VERSION=2023.1
DBCSR_VERSION=v2.5.0

INSTDIR=/apl/cp2k/${VERSION}

SOURCE_ROOT=/home/users/${USER}/Software/CP2K/${VERSION}
TARBALL=${SOURCE_ROOT}/cp2k-${VERSION}.tar.bz2

TC_PATCH_2_1=${SOURCE_ROOT}/patch-openblas.diff
TC_PATCH_6_1=${SOURCE_ROOT}/patch-plumed.diff
TC_PATCH_8_1=${SOURCE_ROOT}/patch-spla-nogpu.diff

PARALLEL=32

# ------------------------------------------------------------------------
umask 0022
export LANG=C
export LC_ALL=C
ulimit -s unlimited

module -s purge
module -s load gcc-toolset/11
module -s load openmpi/4.1.4-hpcx/gcc11

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 -rf cp2k-${VERSION}/.dockerignore
rmdir cp2k-${VERSION}

cd ${INSTDIR}/tools/toolchain

# apply patches
patch -p0 < ${TC_PATCH_2_1}
patch -p0 < ${TC_PATCH_6_1}
patch -p0 < ${TC_PATCH_8_1}
sed -i -e  "/PARMETISLIB=FALSE/"a'-DCMAKE_C_COMPILER=${MPICC} -DCMAKE_CXX_COMPILER=${MPICXX} \\' scripts/stage5/install_superlu.sh

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

./install_cp2k_toolchain.sh --mpi-mode=openmpi \
                            --math-mode=openblas \
                            --with-gcc=system \
                            --with-cmake=system \
                            --with-openmpi=system \
                            --with-mpich=no \
                            --with-intelmpi=no \
                            --with-libxc=install \
                            --with-libint=install \
                            --with-fftw=install \
                            --with-acml=no \
                            --with-mkl=no \
                            --with-openblas=install \
```

```
                    --with-scalapack=install \
                    --with-libxsmm=install \
                    --with-elpa=install \
                    --with-ptscotch=install \
                    --with-superlu=install \
                    --with-pexsi=install \
                    --with-quip=install \
                    --with-plumed=install \
                    --with-sirius=install \
                    --with-gsl=install \
                    --with-libvdwxc=install \
                    --with-spglib=install \
                    --with-hdf5=install \
                    --with-spfft=install \
                    --with-spla=install \
                    --with-cosma=install \
                    --with-libvori=install \
                    --with-libtorch=install

cp install/arch/local.psmp ../../arch/rccs.psmp
cd ${INSTDIR}

# dbscr source code is already available

make -j ${PARALLEL} ARCH=rccs VERSION=psmp
make -j ${PARALLEL} ARCH=rccs VERSION=psmp libcp2k
```

テスト

以下のジョブスクリプトで実行

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

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

module -s purge
module -s load gcc-toolset/11
module -s load openmpi/4.1.4-hpcx/gcc11

CP2K=/apl/cp2k/2023.1

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}
```

結果

```
regtest_mpi1_omp1.log:GREPME 0 0 3868 0 3868 X
```

```
regtest_mpi1_omp2.log:GREPME 0 0 3868 0 3868 X
regtest_mpi2_omp1.log:GREPME 0 0 3926 0 3926 X
regtest_mpi2_omp2.log:GREPME 0 0 3926 0 3926 X
regtest_mpi8_omp1.log:GREPME 10 15 3908 0 3933 X
regtest_mpi8_omp2.log:GREPME 10 15 3908 0 3933 X
```

メモ

- テストの詳細については /apl/cp2k/2023.1/regtesting/rccs/psmp 以下のディレクトリ内にある情報をご確認ください。
  - summary.txt, error_summary, timings.txt あたりが参考になるかと思います。
- 前回断念した superlu についてはコンパイラをきちんと渡すことで通過できている
- 今回は cosma を有効化。確認した範囲ではビルドに問題は無く、速度的におかしなことも起こっていない。
- GPU 版は速度が出るケースもあるものの、挙動が安定しないため今回は回避。
  - 速度が向上する条件を探るのにも一手間かかる。
  - メモリ消費的にもかなり厳しい。mpirun に -mca coll_hcoll_enable 0 をつけないと動かないケースも多い。
  - toolchain 部分だけ GPU 対応した状況はそこまで不安定では無い。しかし、速度向上が微妙。速度が落ちるケースも多い。
- 以下のメッセージについては -mtune=native でビルドしているため問題は無いと思われる。
  - -march での指定のみを見ていると思われる

```
*** HINT in environment.F:884 :: The compiler target flags (generic) used ***
*** to build this binary cannot exploit all extensions of this CPU model  ***
*** (x86_avx2). Consider compiler target flags as part of FCFLAGS and     ***
```

- (2024/7/26) HPC-X 2.11, 2.13.1 の実行時ライブラリを使った場合 H20-256 系の 128 並列計算で速度が大幅に落ちる現象を確認。Open MPI 4.1.5, 4.1.6, HPC-X 2.16 を使えば問題は発生しない。