CP2K 7.1.0 (intel)

ウェブページ

https://www.cp2k.org/

バージョン

7.1.0

ビルド環境

- Intel Parallel Studio 2018 Update 4
- GCC 7.3.1 (devtoolset-7)
- cmake 3.16.3

ビルドに必要なファイル

- cp2k-7.1.0.tar.gz
- dbcsr-2.0.1.tar.gz
- tc_install_cp2k_toolchain.sh.diff

```
--- install_cp2k_toolchain.sh.org       2020-01-31 17:07:19.000000000 +0900
+++ install_cp2k_toolchain.sh   2020-01-31 17:07:30.000000000 +0900
@@ -942,8 +942,8 @@
     ./scripts/install_superlu.sh
     ./scripts/install_pexsi.sh
     ./scripts/install_quip.sh
-    ./scripts/install_plumed.sh
     ./scripts/install_gsl.sh
+    ./scripts/install_plumed.sh
     ./scripts/install_spglib.sh
     ./scripts/install_hdf5.sh
     ./scripts/install_libvdwxc.sh
```

- tc_install_mkl.sh.intel.diff

```
--- install_mkl.sh.org  2020-01-31 15:58:56.000000000 +0900
+++ install_mkl.sh      2020-02-04 17:20:01.000000000 +0900
@@ -66,37 +66,38 @@
         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="-Wl,--start-group ${mkl_lib_dir}/libmkl_gf_lp64.a ${mkl_lib_dir}/libmkl_core.a ${mkl_lib_dir}/libmkl_sequential.a"
+    MKL_LIBS="-L${mkl_lib_dir} -Wl,--no-as-needed -lmkl_scalapack_lp64 -lmkl_blacs_intelmpi_lp64 -lmkl_intel_lp64 -lmkl_sequential -lmkl_core -
lpthread -lm -ldl"
     # check optional libraries
-    if [ $MPI_MODE != no ] ; then
+    #if [ $MPI_MODE != no ] ; then
         enable_mkl_scalapack="__TRUE__"
-        mkl_optional_libs="libmkl_scalapack_lp64.a"
-        case $MPI_MODE in
-            mpich)
-                mkl_blacs_lib="libmkl_blacs_intelmpi_lp64.a"
-                ;;
-            openmpi)
-                mkl_blacs_lib="libmkl_blacs_openmpi_lp64.a"
-                ;;
-            *)
-                enable_mkl_scalapack="__FALSE__"
-                ;;
-        esac
-        mkl_optional_libs="$mkl_optional_libs $mkl_blacs_lib"
```

```
-       for ii in $mkl_optional_libs ; do
-           if ! [ -f "${mkl_lib_dir}/${ii}" ] ; then
-               enable_mkl_scalapack="__FALSE__"
-           fi
-       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}"
-       fi
-   else
-       echo "Not using MKL provided ScaLAPACK and BLACS"
-       enable_mkl_scalapack="__FALSE__"
-   fi
-   MKL_LIBS="${MKL_LIBS} -Wl,--end-group -lpthread -lm -ldl"
+   #    mkl_optional_libs="libmkl_scalapack_lp64.a"
+   #    case $MPI_MODE in
+   #        mpich)
+   #            mkl_blacs_lib="libmkl_blacs_intelmpi_lp64.a"
+   #            ;;
+   #        openmpi)
+   #            mkl_blacs_lib="libmkl_blacs_openmpi_lp64.a"
+   #            ;;
+   #        *)
+   #            enable_mkl_scalapack="__FALSE__"
+   #            ;;
+   #    esac
+       mkl_optional_libs=""
+   #    for ii in $mkl_optional_libs ; do
+   #        if ! [ -f "${mkl_lib_dir}/${ii}" ] ; then
+   #            enable_mkl_scalapack="__FALSE__"
+   #        fi
+   #    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}"
+   #    fi
+   #else
+   #    echo "Not using MKL provided ScaLAPACK and BLACS"
+   #    enable_mkl_scalapack="__FALSE__"
+   #fi
+   #MKL_LIBS="${MKL_LIBS} -Wl,--end-group -lpthread -lm -ldl"
    MKL_CFLAGS="${MKL_CFLAGS} -I${MKLROOT}/include -I${MKLROOT}/include/fftw"

    # write setup files
```

- tc_install_mpich.sh.intel.diff

```
--- install_mpich.sh.org        2020-01-29 15:58:59.000000000 +0900
+++ install_mpich.sh    2020-01-29 16:03:53.000000000 +0900
@@ -50,14 +50,14 @@
        ;;
    __SYSTEM__)
        echo "==================== Finding MPICH from system paths ===================="
-       check_command mpirun "mpich"
-       check_command mpicc "mpich"
-       check_command mpif90 "mpich"
-       check_command mpic++ "mpich"
-       check_lib -lmpi "mpich"
-       check_lib -lmpicxx "mpich"
-       add_include_from_paths MPICH_CFLAGS "mpi.h" $INCLUDE_PATHS
-       add_lib_from_paths MPICH_LDFLAGS "libmpi.*" $LIB_PATHS
+       #check_command mpirun "mpich"
+       #check_command mpicc "mpich"
+       #check_command mpif90 "mpich"
+       #check_command mpic++ "mpich"
```

```
+       #check_lib -lmpi "mpich"
+       #check_lib -lmpicxx "mpich"
+       #add_include_from_paths MPICH_CFLAGS "mpi.h" $INCLUDE_PATHS
+       #add_lib_from_paths MPICH_LDFLAGS "libmpi.*" $LIB_PATHS
        ;;
    __DONTUSE__)
        ;;
@@ -87,15 +87,17 @@
        mpi_bin=mpirun
    fi
    # check MPICH version, versions less than 3.0 will get -D__MPI_VERSION=2 flag
-   raw_version=$($mpi_bin --version | \
-                 grep "Version:" | awk '{print $2}')
-   major_version=$(echo $raw_version | cut -d '.' -f 1)
-   minor_version=$(echo $raw_version | cut -d '.' -f 2)
-   if [ $major_version -lt 3 ] ; then
-       mpi2_dflags="-D__MPI_VERSION=2"
-   else
+   #raw_version=$($mpi_bin --version | \
+   #              grep "Version:" | awk '{print $2}')
+   #major_version=$(echo $raw_version | cut -d '.' -f 1)
+   #minor_version=$(echo $raw_version | cut -d '.' -f 2)
+   #if [ $major_version -lt 3 ] ; then
+   #    mpi2_dflags="-D__MPI_VERSION=2"
+   #else
        mpi2_dflags=''
-   fi
+   #fi
+   MPICH_CFLAGS="-I${I_MPI_ROOT}/include64"
+   MPICH_LDFLAGS="-L${I_MPI_ROOT}/lib64 -Wl,-rpath=${I_MPI_ROOT}/lib64"
    cat <<EOF >> "${BUILDDIR}/setup_mpich"
export MPI_MODE="${MPI_MODE}"
export MPICH_CFLAGS="${MPICH_CFLAGS}"
```

- tc_install_libint.sh.intel.diff (不必要でかつビルドに失敗するものを無理矢理外しています)

```
--- install_libint.sh.org       2020-02-03 18:13:20.000000000 +0900
+++ install_libint.sh   2020-02-03 19:16:36.000000000 +0900
@@ -72,6 +72,8 @@
          #cmake --build . > cmake.log 2>&1
          #cmake --build . --target install > install.log 2>&1

+          # extremely ad hoc workaround
+          sed -i -e "s/fortran_example check_test/libint_f.o check_test/" fortran/Makefile.in
          ./configure --prefix=${pkg_install_dir} \
                      --with-cxx="$CXX $LIBINT_CXXFLAGS" \
                      --with-cxx-optflags="$LIBINT_CXXFLAGS" \
```

- tc_install_sirius.sh.diff

```
--- install_sirius.sh.org       2020-02-25 14:50:50.000000000 +0900
+++ install_sirius.sh   2020-02-25 14:48:01.000000000 +0900
@@ -131,8 +131,8 @@
                -DSpFFT_DIR="${SPFFT_ROOT}/lib/cmake/SpFFT" \
                -DCMAKE_CXXFLAGS_RELEASE="${SIRIUS_OPT}" \
                -DCMAKE_CXX_FLAGS_RELWITHDEBINFO="${SIRIUS_DBG}" \
-               -DCMAKE_CXX_COMPILER=mpic++ \
-               -DCMAKE_C_COMPILER=mpicc \
+               -DCMAKE_CXX_COMPILER=${MPICXX} \
+               -DCMAKE_C_COMPILER=${MPICC} \
               ${COMPILATION_OPTIONS} .. > compile.log 2>&1
           make -j $NPROCS -C src >> compile.log 2>&1

@@ -155,8 +155,8 @@
                  -DCMAKE_CXX_FLAGS_RELWITHDEBINFO="${SIRIUS_DBG}" \
                  -DUSE_CUDA=ON \
```

```
                  -DGPU_MODEL=P100 \
-                 -DCMAKE_CXX_COMPILER=mpic++ \
-                 -DCMAKE_C_COMPILER=mpicc ${COMPILATION_OPTIONS} .. >> compile.log 2>&1
+                  -DCMAKE_CXX_COMPILER=${MPICXX} \
+                  -DCMAKE_C_COMPILER=${MPICC} ${COMPILATION_OPTIONS} .. >> compile.log 2>&1
              make -j $NPROCS -C src >> compile.log 2>&1
              install -d ${pkg_install_dir}/lib/cuda
              install -d ${pkg_install_dir}/include/cuda
```

- tc_install_superlu.sh.intel.diff

```
--- install_superlu.sh.org      2020-02-04 11:46:01.000000000 +0900
+++ install_superlu.sh  2020-02-04 11:46:22.000000000 +0900
@@ -46,12 +46,12 @@
          cat <<EOF >> make.inc
 PLAT=_${OPENBLAS_ARCH}
 DSUPERLULIB= ${PWD}/lib/libsuperlu_dist.a
-LIBS=\$(DSUPERLULIB) ${PARMETIS_LDFLAGS} ${METIS_LDFLAGS} ${MATH_LDFLAGS} ${PARMETIS_LIBS} ${METIS_LIBS} $(resolve_string
"${MATH_LIBS}" OMP) -lgfortran
+LIBS=\$(DSUPERLULIB) ${PARMETIS_LDFLAGS} ${METIS_LDFLAGS} ${MATH_LDFLAGS} ${PARMETIS_LIBS} ${METIS_LIBS} $(resolve_string
"${MATH_LIBS}" OMP)
 ARCH=ar
 ARCHFLAGS=cr
 RANLIB=ranlib
 CC=${MPICC}
-CFLAGS=${CFLAGS} ${PARMETIS_CFLAGS} ${METIS_CFLAGS} ${MATH_CFLAGS}
+CFLAGS=${CFLAGS} -std=c99 -fPIC ${PARMETIS_CFLAGS} ${METIS_CFLAGS} ${MATH_CFLAGS}
 NOOPTS=-O0
 FORTRAN=${MPIFC}
 F90FLAGS=${FFLAGS}
```

- tc_install_libvdwxc.sh.intel.diff

```
--- install_libvdwxc.sh.org     2020-02-04 16:23:50.000000000 +0900
+++ install_libvdwxc.sh 2020-02-04 16:48:35.000000000 +0900
@@ -65,7 +65,7 @@
          unset MPICC MPICXX MPIF90 MPIFC MPIF77
          if [ "$MPI_MODE" = "no" ]; then
              # compile libvdwxc without mpi support since fftw (or mkl) do not have mpi support activated
-             ./configure \
+              CC=${CC} FC=${FC} ./configure \
                --prefix="${pkg_install_dir}" \
                --libdir="${pkg_install_dir}/lib" \
                --with-fftw3=${FFTW_ROOT} \
@@ -73,12 +73,11 @@
                --without-mpi \
                >> configure.log 2>&1
          else
-             CC=mpicc FC=mpifort ./configure \
+              MPICC=mpiicc MPIFC=mpiifort ./configure \
                --prefix="${pkg_install_dir}" \
                --libdir="${pkg_install_dir}/lib" \
                --with-fftw3=${FFTW_ROOT} \
                --disable-shared \
-               --with-mpi \
                >> configure.log 2>&1
          fi
          make -j $NPROCS > compile.log 2>&1
```

- tc_install_plumed.sh.diff

```
--- install_plumed.sh.org       2020-02-04 15:13:41.986747619 +0900
+++ install_plumed.sh   2020-02-04 16:00:15.980396838 +0900
@@ -40,7 +40,7 @@
```

```
            echo "Installing from scratch into ${pkg_install_dir}"
            cd plumed-${plumed_ver}
-           ./configure CXX="${MPICXX}" --prefix=${pkg_install_dir} --libdir="${pkg_install_dir}/lib" > configure.log 2>&1
+            ./configure CXX="${MPICXX}" --prefix=${pkg_install_dir} --libdir="${pkg_install_dir}/lib" CXXFLAGS="-I${GSLROOT}/include" LIBS="-
L${GSLROOT}/lib -L${MKLROOT}/lib/intel64 -Wl,--no-as-needed -lmkl_intel_lp64 -lmkl_sequential -lmkl_core -lpthread -lm -ldl" > configure.log 2>&1
            make -j $NPROCS > make.log 2>&1
            make install > install.log 2>&1
            write_checksums "${install_lock_file}" "${SCRIPT_DIR}/$(basename ${SCRIPT_NAME})"
@@ -63,7 +63,7 @@
 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/cp2k710

VERSION=7.1.0
DBCSR_VERSION=2.0.1

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

TARBALL=${SOURCE_ROOT}/cp2k-${VERSION}.tar.gz
TARBALL_DBCSR=${SOURCE_ROOT}/dbcsr-${DBCSR_VERSION}.tar.gz

TC_PATCH0=${SOURCE_ROOT}/tc_install_cp2k_toolchain.sh.diff
TC_PATCH1=${SOURCE_ROOT}/tc_install_mkl.sh.intel.diff
TC_PATCH2=${SOURCE_ROOT}/tc_install_mpich.sh.intel.diff
TC_PATCH3=${SOURCE_ROOT}/tc_install_libint.sh.intel.diff
#TC_PATCH4=${SOURCE_ROOT}/tc_install_quip.sh.intel.diff
TC_PATCH5=${SOURCE_ROOT}/tc_install_sirius.sh.diff
TC_PATCH6=${SOURCE_ROOT}/tc_install_superlu.sh.intel.diff
TC_PATCH7=${SOURCE_ROOT}/tc_install_libvdwxc.sh.intel.diff
TC_PATCH8=${SOURCE_ROOT}/tc_install_plumed.sh.diff

PARALLEL=12

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

module purge
module load scl/devtoolset-7
module load intel_parallelstudio/2018update4
module load cmake/3.16.3

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

```
cd ${INSTDIR}/tools/toolchain
patch < ${TC_PATCH0}

cd scripts
patch < ${TC_PATCH1}
patch < ${TC_PATCH2}
patch < ${TC_PATCH3}
#patch < ${TC_PATCH4}
patch < ${TC_PATCH5}
patch < ${TC_PATCH6}
patch < ${TC_PATCH7}
patch < ${TC_PATCH8}
cd ../

export CC=icc
export CXX=icpc
export FC=ifort
export MPICC=mpiicc
export MPICXX=mpiicpc
export MPIFC=mpiifort

./install_cp2k_toolchain.sh --math-mode=mkl \
                            --mpi-mode=mpich \
                            --with-cmake=system \
                            --with-mpich=system \
                            --with-openmpi=no \
                            --with-libxc=install \
                            --with-libint=install \
                            --with-fftw=install \
                            --with-openblas=no \
                            --with-scalapack=no \
                            --with-reflapack=no \
                            --with-libxsmm=install \
                            --with-elpa=install \
                            --with-ptscotch=install \
                            --with-pexsi=install \
                            --with-parmetis=install \
                            --with-superlu=install \
                            --with-quip=no \
                            --with-plumed=install \
                            --with-gsl=install \
                            --with-libvdwxc=install \
                            --with-spglib=install \
                            --with-hdf5=install \
                            --with-spfft=install \
                            -j ${PARALLEL}

sed -e "/^LIBS /s/\$/ -nofor_main/" \
    install/arch/local.psmp > ../../arch/rccs.psmp

cd ${INSTDIR}/exts
rmdir dbcsr
tar zxf ${TARBALL_DBCSR}
mv dbcsr-${DBCSR_VERSION} dbcsr
cd ../
make -j ${PARALLEL} ARCH=rccs VERSION=psmp
```

テスト

以下のスクリプトを ccfep 上で実行しています。

```
#!/bin/sh

export LC_ALL=C
export LANG=""
```

```
# intel
module purge
module load scl/devtoolset-7
module load intel_parallelstudio/2018update4
module load cmake/3.16.3
CP2K=/local/apl/lx/cp2k710

CP2K_ARCH=rccs
CP2K_VER=psmp
TIMEOUT=120
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 ${TIMEOUT} \
       -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 ${TIMEOUT} \
       -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 ${TIMEOUT} \
       -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 ${TIMEOUT} \
```

```
        -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 ${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 \
        -nosvn \
        -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: MPI1 - OMP1

```
-------------------------------- Summary --------------------------------
Number of FAILED  tests 1
Number of WRONG   tests 2
Number of CORRECT tests 3214
Number of NEW     tests 3
Total number of   tests 3220
```

- QS/regtest-ri-mp2/opt_basis_O_auto_gen.inp: WRONG
- QS/regtest-almo-2/ion-pair.inp: RUNTIME FAIL
- Fist/regtest-3/2d_pot.inp: WRONG

■ テスト結果: MPI1 - OMP2

```
-------------------------------- Summary --------------------------------
Number of FAILED  tests 2
Number of WRONG   tests 2
Number of CORRECT tests 3213
Number of NEW     tests 3
Total number of   tests 3220
```

- SIRIUS/regtest-1/He-full-potential.inp: RUNTIME FAIL
- QS/regtest-ri-mp2/opt_basis_O_auto_gen.inp: WRONG
- QS/regtest-almo-2/ion-pair.inp: RUNTIME FAIL
- Fist/regtest-3/2d_pot.inp: WRONG

■ テスト結果: MPI2 - OMP1

```
-------------------------------- Summary --------------------------------
Number of FAILED  tests 0
Number of WRONG   tests 2
Number of CORRECT tests 3275
Number of NEW     tests 8
Total number of   tests 3285
```

- QS/regtest-ri-mp2/opt_basis_O_auto_gen.inp: WRONG

- Fist/regtest-3/2d_pot.inp: WRONG

■ テスト結果: MPI2 - OMP2

```
---------------------------------- Summary ----------------------------------
Number of FAILED  tests 1
Number of WRONG   tests 2
Number of CORRECT tests 3274
Number of NEW     tests 8
Total number of   tests 3285
```

- SIRIUS/regtest-1/He-full-potential.inp: RUNTIME FAIL
- QS/regtest-ri-mp2/opt_basis_O_auto_gen.inp: WRONG
- Fist/regtest-3/2d_pot.inp: WRONG

■ テスト結果: MPI8 - OMP1

```
---------------------------------- Summary ----------------------------------
Number of FAILED  tests 7
Number of WRONG   tests 10
Number of CORRECT tests 3219
Number of NEW     tests 6
Total number of   tests 3242
```

- QS/regtest-mp2-lr/H2O-mp2-gpw-lr.inp: WRONG
- QS/regtest-mp2-grad/H2O_grad_mme.inp: WRONG
- QS/regtest-gpw-4/H2O-debug-5.inp: WRONG
- QS/regtest-gpw-4/H2O-debug-6.inp: WRONG
- QS/regtest-ri-mp2/opt_basis_O_auto_gen.inp: WRONG
- Fist/regtest-3/2d_pot.inp: WRONG
- QS/regtest-mp2-4/H2O_NO_HFX.inp: WRONG
- QS/regtest-rma-3D/H2O-32-dftb-ls-2_mult.inp: RUNTIME FAIL
- QS/regtest-rma-3D/H2O-32-dftb-ls-2.inp: RUNTIME FAIL
- QS/regtest-rma-3D/H2O-OT-ASPC-1.inp: RUNTIME FAIL
- QS/regtest-rma-3D/H2-big-nimages.inp: RUNTIME FAIL
- QS/regtest-rma-3D/H2O_grad_gpw.inp: RUNTIME FAIL
- QS/regtest-rma-3D/OH-H2O-bsse.inp: RUNTIME FAIL
- QS/regtest-rma-3D/H2O-6.inp: RUNTIME FAIL
- TMC/regtest_ana_on_the_fly/TMC_ana_start_with_exist_traj.inp: WRONG
- TMC/regtest_ana_on_the_fly/TMC_ana_restart.inp: WRONG
- QS/regtest-mp2-2/H2O-02.inp: WRONG

■ テスト結果: MPI8 - OMP2

```
---------------------------------- Summary ----------------------------------
Number of FAILED  tests 8
Number of WRONG   tests 10
Number of CORRECT tests 3218
Number of NEW     tests 6
Total number of   tests 3242
```

- QS/regtest-mp2-lr/H2O-mp2-gpw-lr.inp: WRONG
- SIRIUS/regtest-1/He-full-potential.inp: RUNTIME FAIL
- QS/regtest-mp2-grad/H2O_grad_mme.inp: WRONG
- QS/regtest-gpw-4/H2O-debug-5.inp: WRONG
- QS/regtest-gpw-4/H2O-debug-6.inp: WRONG
- QS/regtest-ri-mp2/opt_basis_O_auto_gen.inp: WRONG
- Fist/regtest-3/2d_pot.inp: WRONG
- QS/regtest-mp2-4/H2O_NO_HFX.inp: WRONG
- QS/regtest-rma-3D/H2O-32-dftb-ls-2_mult.inp: RUNTIME FAIL
- QS/regtest-rma-3D/H2O-32-dftb-ls-2.inp: RUNTIME FAIL
- QS/regtest-rma-3D/H2O-OT-ASPC-1.inp: RUNTIME FAIL
- QS/regtest-rma-3D/H2-big-nimages.inp: RUNTIME FAIL
- QS/regtest-rma-3D/H2O_grad_gpw.inp: RUNTIME FAIL
- QS/regtest-rma-3D/OH-H2O-bsse.inp: RUNTIME FAIL
- QS/regtest-rma-3D/H2O-6.inp: RUNTIME FAIL

- TMC/regtest_ana_on_the_fly/TMC_ana_start_with_exist_traj.inp: WRONG
- TMC/regtest_ana_on_the_fly/TMC_ana_restart.inp: WRONG
- QS/regtest-mp2-2/H2O-02.inp: WRONG

ベンチマーク

6.1.0 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)
21 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	-	68.154
small	40 (1)	40	1	-	45.426
small	80 (2)	80	1	-	31.327
small	160 (4)	32	5	-	23.398

雑多な情報

- GPU 版はあまり意味が無いようなので今回ははじめから除外。
  - SIRIUS の GPU 版については MAGMA 等まで対応させれば速度が出るのかもしれませんが、今回は回避しています。
- libgrid を使うとわずかに遅くなったため、使用せず。
  - pyratemp 0.3.2 で試行。そのままビルドすると xyz_to_vab 内の *.template ファイルを正しく処理できておらず、失敗する。
  - sed -i -e "s/\\$/\!/g" -e "s/\@</\@\!/g" -e "s/>\@/\!\@/g" であらかじめ *.template ファイルを処理することで一応動作は確認。速度面ではメリット無し。
    - (スクリプト中で実行する場合は sed -i -e "s/\\\$/\!/g" -e "s/\@</\@\!/g" -e "s/>\@/\!\@/g")
- libsmm については今回も未検証
- cp2k 本体は -O2 でビルド。-O2 -xHost や -O3 -xHost も試したが、速度は上がらない。
- dbcsr を指示通りに git で持ってくると master ブランチのものを使うことになる。タイミングによってはダメなことが起こるので、リリース版を使っている。
- QUIP はテストをパスしないため、最終的に外している。GCC 版では正常に動いている。
  - QUIP をビルドする際に必要だったパッチファイル tc_install_quip.sh.intel.diff (今回は最終的に利用していません。あくまで参考情報です)

```
--- install_quip.sh.org 2020-02-04 13:38:25.000000000 +0900
+++ install_quip.sh     2020-02-04 12:53:16.000000000 +0900
@@ -68,34 +68,24 @@
                -e "s|\(cd build/.*\)|\1 >&- 2>&-|g" \
                bin/find_sizeof_fortran_t
        fi
-       sed -i \
-          -e "s|\(F77 *=\).*|\1 ${FC}|g" \
-          -e "s|\(F90 *=\).*|\1 ${FC}|g" \
-          -e "s|\(F95 *=\).*|\1 ${FC}|g" \
-          -e "s|\(CC *=\).*|\1 ${CC}|g" \
-          -e "s|\(CPLUSPLUS *=\).*|\1 ${CXX}|g" \
-          -e "s|\(LINKER *=\).*|\1 ${FC}|g" \
-          -e "s|\(FPP *=\).*|\1 ${FC} -E -x f95-cpp-input|g" \
-          -e "s|\(QUIPPY_FCOMPILER *=\).*|\1 ${FC}|g" \
-          -e "s|\(QUIPPY_CPP *=\).*|\1 ${FC} -E -x f95-cpp-input|g" \
-          arch/Makefile.linux_${quip_arch}_gfortran
        # enable debug symbols
-       echo "F95FLAGS       += -g" >> arch/Makefile.linux_${quip_arch}_gfortran
-       echo "F77FLAGS       += -g" >> arch/Makefile.linux_${quip_arch}_gfortran
-       echo "CFLAGS         += -g" >> arch/Makefile.linux_${quip_arch}_gfortran
-       echo "CPLUSPLUSFLAGS += -g" >> arch/Makefile.linux_${quip_arch}_gfortran
-       export QUIP_ARCH=linux_${quip_arch}_gfortran
+        echo "F95FLAGS       += -g" >> arch/Makefile.linux_${quip_arch}_ifort_icc
+        echo "F77FLAGS       += -g" >> arch/Makefile.linux_${quip_arch}_ifort_icc
+        echo "CFLAGS         += -g" >> arch/Makefile.linux_${quip_arch}_ifort_icc
+        echo "CPLUSPLUSFLAGS += -g" >> arch/Makefile.linux_${quip_arch}_ifort_icc
+        export QUIP_ARCH=linux_${quip_arch}_ifort_icc
        # hit enter a few times to accept defaults
-       echo -e "${MATH_LDFLAGS} $(resolve_string "${MATH_LIBS}") \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n" | make config > configure.log
```

```
+           QUIP_MATH_LIBS="-lmkl_intel_lp64 -lmkl_sequential -lmkl_core -lpthread -lm -ldl"
+           echo -e "${MATH_LDFLAGS} ${QUIP_MATH_LIBS} \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n" | make config > configure.log
            # make -j does not work :-(
            make > make.log 2>&1
            ! [ -d "${pkg_install_dir}/include" ] && mkdir -p "${pkg_install_dir}/include"
            ! [ -d "${pkg_install_dir}/lib" ] && mkdir -p "${pkg_install_dir}/lib"
-           cp build/linux_x86_64_gfortran/quip_unified_wrapper_module.mod \
+           cp build/linux_x86_64_ifort_icc/quip_unified_wrapper_module.mod \
              "${pkg_install_dir}/include/"
-           cp build/linux_x86_64_gfortran/*.a \
+           cp build/linux_x86_64_ifort_icc/*.a \
              "${pkg_install_dir}/lib/"
-           cp src/FoX-4.0.3/objs.linux_${quip_arch}_gfortran/lib/*.a \
+           cp src/FoX-4.0.3/objs.linux_${quip_arch}_ifort_icc/lib/*.a \
              "${pkg_install_dir}/lib/"
            cd ..
            write_checksums "${install_lock_file}" "${SCRIPT_DIR}/$(basename ${SCRIPT_NAME})"
```

- Intel 2019 では少し遅い + MPI の挙動に何か問題があるのか、エラーが多い。
- Intel 2017, 2018, 2019 で試した範囲では 2018 が一番計算速度が速い。エラーの数も 2018 がもっとも少ない (ただし、2017 と 2018 はほぼ同じ水準)。
- Intel 2017 ではコードを修正しないとビルドが通らない。iso_fortran_env で未対応のものが使われているため。
    - Intel 2017 でビルドする際に必要だった src/start/cp2k.F のパッチ

```
--- src/start/cp2k.F.org        2020-02-17 15:26:54.474087502 +0900
+++ src/start/cp2k.F    2020-02-17 15:40:07.489475278 +0900
@@ -55,8 +55,6 @@
    USE input_cp2k,                      ONLY: create_cp2k_root_section
    USE input_section_types,             ONLY: section_release,&
                                               section_type
-   USE iso_fortran_env,                 ONLY: compiler_options,&
-                                              compiler_version
    USE kinds,                           ONLY: default_path_length,&
                                               default_string_length
    USE machine,                         ONLY: default_output_unit
@@ -245,14 +243,6 @@
                cp2k_version//TRIM(dev_flag), &
                "Source code revision "//TRIM(compile_revision), &
                TRIM(cp2k_flags())
-             compiler_options_string = compiler_options()
-             WRITE (output_unit, "(T2,A,A)") "compiler: ", compiler_version()
-             WRITE (output_unit, "(T2,A)") "compiler options:"
-             DO i = 0, (LEN(compiler_options_string) - 1)/68
-               WRITE (output_unit, "(T4,A)") &
-                 compiler_options_string(i*68 + 1:MIN(LEN(compiler_options_string), (i + 1)*68))
-             END DO
-             DEALLOCATE (compiler_options_string)
            END IF
          END IF
```