

## DIRAC 19.0

### ウェブページ

<http://www.diracprogram.org/doku.php>

### バージョン

19.0

### ビルド環境

- Intel Compiler 19.1.2
  - ifort 19.1.2
  - icc 19.1.2
  - icpc 19.1.2
- Intel MKL 2020.0.2
- cmake 3.16.3

### ビルドに必要なファイル

- DIRAC-19.0-Source.tar.gz
- openmpi-3.1.0.tar.bz2
  - (デフォルトを 8 バイト整数にするために既存のものとは別にビルド)
- diff\_memcon (see /local/apl/lx/dirac190/patches/diff\_memcon)
  - 利用可能なメモリ量を増やすためのパッチです

### ビルド手順

```
#!/bin/sh

VERSION=19.0
INSTALL_PREFIX=/local/apl/lx/dirac190

# openmpi (8-byte integer)
OMPI_VERSION=3.1.0
OMPI_INSTALL_PREFIX=${INSTALL_PREFIX}/openmpi310_i8
OMPI_TARBALL=/home/users/${USER}/Software/OpenMPI/${OMPI_VERSION}/openmpi-${OMPI_VERSION}.tar.bz2
PBSROOT=/local/apl/lx/pbs14

# dirac
BASEDIR=/home/users/${USER}/Software/DIRAC/${VERSION}
TARBALL=${BASEDIR}/DIRAC-${VERSION}-Source.tar.gz

PATCH_README=${BASEDIR}/README.patch
PATCH_MEMCONTROL=${BASEDIR}/diff_memcon

WORKDIR=/work/users/${USER}

PARALLEL=8
#-----
umask 0022

export LC_ALL=C
export LANG=C
export OMP_NUM_THREADS=1

ulimit -s unlimited

module purge
module load intel/19.1.2
module load mkl/2020.0.2
module load cmake/3.16.3
```

```

# openmpi (8-byte integer default)
cd ${WORKDIR}
if [ -d openmpi-${OMPI_VERSION} ]; then
  mv openmpi-${OMPI_VERSION} openmpi_erase
  rm -rf openmpi_erase &
fi

tar jxf ${OMPI_TARBALL}
cd openmpi-${OMPI_VERSION}
mkdir rccs-i8 && cd rccs-i8
CC=icc CXX=icpc FC=ifort FCFLAGS=-i8 CFLAGS=-m64 CXXFLAGS=-m64 \
  ./configure --prefix=${OMPI_INSTALL_PREFIX} \
    --with-tm=${PBSROOT} \
    --enable-mpi-cxx \
    --with-psm2
make -j ${PARALLEL} && make install && make check

# dirac
cd ${WORKDIR}
if [ -d DIRAC-${VERSION}-Source ]; then
  mv DIRAC-${VERSION}-Source DIRAC_erase
  rm -rf DIRAC_erase &
fi

export PATH="${OMPI_INSTALL_PREFIX}/bin:$PATH"
export LIBRARY_PATH="${OMPI_INSTALL_PREFIX}/lib:$LIBRARY_PATH"
export LD_LIBRARY_PATH="${OMPI_INSTALL_PREFIX}/lib:$LD_LIBRARY_PATH"

export DIRAC_TMPDIR=${WORKDIR}

tar zxf ${TARBALL}
cd DIRAC-${VERSION}-Source
patch -p0 < ${PATCH_MEMCONTROL}

./setup --mpi \
  --fc=mpif90 \
  --cc=mpicc \
  --cxx=mpicxx \
  --mkl=parallel \
  --int64 \
  --extra-fc-flags="-xHost -I${OMPI_INSTALL_PREFIX}/lib" \
  --extra-cc-flags="-xHost" \
  --extra-cxx-flags="-xHost" \
  --prefix=${INSTALL_PREFIX} \
  build.rccs
cd build.rccs
make -j ${PARALLEL} && make install

# copy license and patch files
cp -f ../LICENSE ${INSTALL_PREFIX}
cp -f ${PATCH_README} ${INSTALL_PREFIX}
mkdir ${INSTALL_PREFIX}/patches
cp -f ${PATCH_MEMCONTROL} ${INSTALL_PREFIX}/patches

# store test results
mkdir ${INSTALL_PREFIX}/test_results
mkdir ${INSTALL_PREFIX}/test_results/serial
mkdir ${INSTALL_PREFIX}/test_results/parallel

# serial test
export DIRAC_MPI_COMMAND="mpirun -np 1"
make test
cp Testing/Temporary/LastTest.log ${INSTALL_PREFIX}/test_results/serial
if [ -f Testing/Temporary/LastTestsFailed.log ]; then
  cp Testing/Temporary/LastTestsFailed.log ${INSTALL_PREFIX}/test_results/serial

```

```
fi
# parallel test
export DIRAC_MPI_COMMAND="mpirun -np ${PARALLEL}"
make test
cp Testing/Temporary/LastTest.log ${INSTALL_PREFIX}/test_results/parallel
if [ -f Testing/Temporary/LastTestsFailed.log ]; then
  cp Testing/Temporary/LastTestsFailed.log ${INSTALL_PREFIX}/test_results/parallel
fi
exit 0
```

## テスト結果

### 失敗したテスト: シリアル版(mpirun -np 1)

- 20 - eomcc
- 65 - eedm\_mhyp\_ensps\_krci
- 73 - bss\_energy
- 74 - pam\_test
- 129 - operators\_mo\_mtx\_elements
- 131 - spinrot

### 失敗したテスト: 並列版(mpirun -np 8)

- 18 - polprp\_ph
- 46 - fscc\_restart
- 65 - eedm\_mhyp\_ensps\_krci
- 73 - bss\_energy
- 74 - pam\_test
- 95 - mp2\_srdft\_energies (Timeout)
- 111 - lucita\_short
- 129 - operators\_mo\_mtx\_elements
- 131 - spinrot

## メモ

- テスト結果については /local/apl/lx/dirac190/test\_results 以下にコピーしてあります。(個々のアウトプットは保存していません)
- インテルコンパイラの方が速度的に優位なため、こちらを採用。
- OpenMPI 4.0.2 を使うと 71 - basis\_input\_scripted のテストが Timeout するため回避
- テスト18 (polprp\_ph) 並列時は常にエラー
- テスト20 (eomcc) int64 指定時にシリアル版のみで発生。一部数値が異なる。インテルコンパイラでも gcc でも発生
- テスト46 (fscc\_restart) 並列時は常に数値エラー
- テスト65 (eedm\_mhyp\_ensps\_krci) は軽微な数値エラーに見える。インテルコンパイラでも gcc でも発生
- テスト73 (bss\_energy) int64 の指定が原因と思われるため回避できず
- テスト74 (pam\_test) 良くわからないエラー。インテルコンパイラでも gcc でも発生
- テスト95 (mp2\_srdft\_energies) 並列時には常に Timeout。
- テスト111 (lucita\_short) int64 指定時のみエラーとなる気配(73 と同様のエラー)。シリアル時には通る。
- テスト129 (operators\_mo\_mtx\_elements) 良くわからないエラー。インテルコンパイラでも gcc でも発生
- テスト131 (spinrot) インテルコンパイラ時のみ、一部のデータが B と C で入れ替わっているように見える。