

GENESIS 2.1.2

ウェブページ

<https://www.r-ccs.riken.jp/labs/cbrt/>

バージョン

2.1.2

ビルド環境

- gcc 12.1.1 (gcc-toolset/12)
- MKL 2024.0
- Open MPI 4.1.6

ビルドに必要なファイル

- genesis-2.1.2.tar.bz2
- tests-2.1.2.tar.bz2

ビルド手順

```
#!/bin/sh
VERSION=2.1.2
BASEDIR=/home/users/${USER}/Software/GENESIS/${VERSION}
SRC_TARBALL=${BASEDIR}/genesis-${VERSION}.tar.bz2
TESTS_TARBALL=${BASEDIR}/tests-${VERSION}.tar.bz2

INSTALLDIR=/apl/genesis/2.1.2

WORKDIR=/gwork/users/${USER}
BUILDDIR=${WORKDIR}/genesis-${VERSION}
TESTSDIR=${WORKDIR}/tests-${VERSION}

PARALLEL_TESTS=8

# -----
umask 0022

module -s purge
module -s load gcc-toolset/12
module -s load mkl/2024.0
module -s load openmpi/4.1.6/gcc12

export LANG=C
export LC_ALL=C
export OMP_NUM_THREADS=1
ulimit -s unlimited

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

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

tar jxf ${SRC_TARBALL}
tar jxf ${TESTS_TARBALL}
```

```

cd ${BUILDDIR}
FC=mpif90 CC=mpicc \
LAPACK_LIBS="-L${MKLROOT}/lib/intel64 -WI,--no-as-needed -lmkl_gf_lp64 -lmkl_gnu_thread -lmkl_core -lgomp -lpthread -lm -ldl" \
./configure --prefix=${INSTALLDIR}

make && make install

ATDYN=${INSTALLDIR}/bin/atdyn
SPDYN=${INSTALLDIR}/bin/spdyn

cd ${TESTSDIR}/regression_test

for f in test.py test_remd.py test_fep.py test_rpath.py \
    test_vib.py test_gamd.py \
    test_analysis/test_analysis.py \
    test_spana/test_spana.py; do
    sed -i -e "s/env python/env python3/" $f
done

# atdyn tests
./test.py      "mpirun -np ${PARALLEL_TESTS} $ATDYN"
./test_remd.py "mpirun -np ${PARALLEL_TESTS} $ATDYN"
./test_rpath.py "mpirun -np ${PARALLEL_TESTS} $ATDYN"
./test_vib.py  "mpirun -np ${PARALLEL_TESTS} $ATDYN"
./test_gamd.py "mpirun -np ${PARALLEL_TESTS} $ATDYN"
#./test_fep.py "mpirun -np ${PARALLEL_TESTS} $ATDYN" # spdyn only

# spdyn tests
./test.py      "mpirun -np ${PARALLEL_TESTS} $SPDYN"
./test_remd.py "mpirun -np ${PARALLEL_TESTS} $SPDYN"
./test_rpath.py "mpirun -np ${PARALLEL_TESTS} $SPDYN"
./test_gamd.py "mpirun -np ${PARALLEL_TESTS} $SPDYN"
./test_fep.py  "mpirun -np ${PARALLEL_TESTS} $SPDYN"

cd test_analysis
./test_analysis.py ${INSTALLDIR}/bin
cd ../

cd test_spana
# ad hoc
sed -i -e 'ref_fobj =/s/"r"/"r", encoding="utf-8"/' test_spana.py
sed -i -e 'out_fobj =/s/"r"/"r", encoding="utf-8"/' test_spana.py
./test_spana.py ${INSTALLDIR}/bin
cd ..

```

メモ

- gcc12 の方が gcc11 よりもベンチマーク結果が良い(uun ベンチマーク系)
 - mkl を使う場合は iomp を使うと速度が落ちてしまうため、gomp を選択
- intel compiler よりも gcc の方が速度が出ている
 - intel compiler の場合は、-axCORE-AVX2 よりも -march=core-avx2 で最適化した方が少し速度が出る
 - intel compiler の場合は fseeko64, ftello64 を fseeko, ftello で置き換えないとビルドできない場合あり
 - oneAPI 2024 ではテストに一つエラー(wham_analysis)
- aocc ではビルドできず