GENESIS 2.0.3

Webpage

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

Version

2.0.3

Build Environment

- gcc 11.2.1 (gcc-toolset/11)
- MKL 2022.2.1
- HPC-X 2.11 (Open MPI 4.1.4)
    - (HPC-X 2.13.1 was used in the actual build. However, that version caused a problem when large number of MPI processes were employed. Switch runtime library to HPC-X 2.11 solved the problem.)
    - (In the following, assuming HPC-X 2.11 is used instead of HPC-X 2.13.1.)

Files Required

- genesis-2.0.3.tar.bz2
- tests-2.0.3.tar.bz2

Build Procedure

```
#!/bin/sh

VERSION=2.0.3
BASEDIR=/home/users/${USER}/Software/GENESIS/${VERSION}
SRC_TARBALL=${BASEDIR}/genesis-${VERSION}.tar.bz2
TESTS_TARBALL=${BASEDIR}/tests-${VERSION}.tar.bz2

INSTALLDIR=/apl/genesis/2.0.3

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/11
module -s load mkl/2022.2.1
module -s load openmpi/4.1.4-hpcx/gcc11

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 -Wl,--no-as-needed -lmkl_gf_lp64 -lmkl_gnu_thread -lmkl_core -lgomp -lpthread -lm -ldl" \
  ./configure --prefix=${INSTALLDIR}

# parallel make does not work
make && make install

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

cd ${TESTSDIR}/regression_test

for f in test.py test_remd.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
sed -i -e "s/env python/env python2/" test_nonstrict.py

# 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_nonstrict.py "mpirun -np ${PARALLEL_TESTS} $ATDYN" # ?

# 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_nonstrict.py "mpirun -np ${PARALLEL_TESTS} $SPDYN" # ?

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

cd test_spana
./test_spana.py ${INSTALLDIR}/bin
cd ..
```

Notes

- When Intel oneAPI Compiler Classic was used with "-xHost" option (default setting), compilation aborted with the messages below.
    - This happens for oneAPI Compiler Classic 2022.2.1 and 2022.0.2. This error is resolved by removing "-xHost" option.
    - When "ulimit -s unlimited" is performed before the compilation, the node became not responding. Probably the error depleted the system memory.

```
ifort: error #10105: /lustre/home/users/xxx/intel/oneapi/compiler/2022.2.1/linux/bin/intel64/../../bin/intel64/fortcom: core dumped
ifort: warning #10102: unknown signal(-448501680)
ifort: error #10106: Fatal error in /lustre/home/users/xxx/intel/oneapi/compiler/2022.2.1/linux/bin/intel64/../../bin/intel64/fortcom, terminated by unknown
compilation aborted for at_energy_table_cubic.f90 (code 1)
```

```
ifort: error #10105: /lustre/home/users/xxx/intel/oneapi/compiler/2022.0.2/linux/bin/intel64/../../bin/intel64/fortcom: core dumped
ifort: warning #10102: unknown signal(-902248368)
ifort: error #10106: Fatal error in /lustre/home/users/xxx/intel/oneapi/compiler/2022.0.2/linux/bin/intel64/../../bin/intel64/fortcom, terminated by
```

unknown
compilation aborted for sp_energy_pme_noopt_1dalltoall.f90 (code 1)

- Even when the error was avoided (by switching -xHost to -march=core-avx2), ifort version met errors in tests. Also, there is no clear advantage for Intel version. Therefore, we didn't employ Intel compiler this time.
- MKL is used to meet LAPACK requirement. libgomp is used instead of libiomp; performance of libiomp version seems to be slightly worse. (This can be a wrong guess, though.)
    - (The library setting is from output of "mkl_link_tool -libs -c gnu_f -p yes -o gomp".)