

NWChem-7.0.2

ウェブページ

<https://nwchemgit.github.io/>

バージョン

7.0.2

ビルド環境

- Intel Compiler 19.1.2 (Intel Parallel Studio 2020 update 2)
- Intel MPI 2018.0.4 (Intel Parallel Studio 2018 update 4)
- Intel MKL 2020.0.2 (Intel Parallel Studio 2020 update 2)

ビルドに必要なファイル

- nwchem-7.0.2-release.revision-b9985dfa-src.2020-10-12.tar.bz2
- runtest.md.mpi

```
#!/bin/sh
./runtests.mpi.unix procs 2 \
na_k/nak \
na_k/nak_md \
crown/crown_md \
ethanol/ethanol_md \
ethanol/ethanol_ti \
had/had_em \
had/had_md \
prep/a3n \
prep/aal \
prep/fsc \
water/water_md
```

ビルド手順

```
#!/bin/sh

VERSION=7.0.2
INSTALL_PREFIX=/local/apl/lx/nwchem702

BASEDIR=/home/users/${USER}/Software/NWChem/7.0.2
TARBALL=${BASEDIR}/nwchem-7.0.2-release.revision-b9985dfa-src.2020-10-12.tar.bz2

WORKDIR=/work/users/${USER}

RUNTESTMD=runtest.md.mpi
RUNTESTMD_PATH=${BASEDIR}/${RUNTESTMD}

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

module purge
module load intel/19.1.2
module load mpi/intelmpi/2018.4.274
module load mkl/2020.0.2

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

```

fi

tar jxf ${TARBALL}

export NWCHEM_TOP=${WORKDIR}/nwchem-7.0.2
export NWCHEM_MODULES="all python"
export NWCHEM_TARGET=LINUX64
export ARMCI_NETWORK=MPI-PR

export USE_OPENMP=y
export USE_MPI=y
export USE_MPIF=y
export USE_MPIF4=y

export USE_NOFSCHECK=TRUE
export USE_NOIO=TRUE
export MRCC_METHODS=TRUE
export CCSDTQ=TRUE
export LIB_DEFINES=-DDFLT_TOT_MEM=268435456 # 2GiB/process

export PYTHONVERSION=2.7

export BLAS_SIZE=8
# mkl_link_tool -libs -c intel_f -p yes -i ilp64 --quiet
export BLASOPT="-L${MKLRROOT}/lib/intel64 -lmkl_intel_ilp64 -lmkl_intel_thread -lmkl_core -liomp5 -lpthread -lm -ldl"
export LAPACK_SIZE=8
export LAPACK_LIB="${BLASOPT}"
export USE_SCALAPACK=y
export SCALAPACK_SIZE=8
# mkl_link_tool -libs -c intel_f -p yes -i ilp64 --cluster_library=scalapack -m intelmpi --quiet
export SCALAPACK="-L${MKLRROOT}/lib/intel64 -lmkl_scalapack_ilp64 -lmkl_blast_intelmpi_ilp64 -lmkl_intel_ilp64 -lmkl_intel_thread -lmkl_core -liomp5 -lpthread -lm -ldl"

export CC=icc
export FC=ifort

cd ${NWCHEM_TOP}/src
make nwchem_config
make

# involve also version info
cd ${NWCHEM_TOP}/src/util
make version
make
cd ${NWCHEM_TOP}/src
make link
cd ${NWCHEM_TOP}

# installation; assume INSTALL_PREFIX directory already exists
cp -fr LICENSE.TXT README.md release.notes.* examples ${INSTALL_PREFIX}

mkdir ${INSTALL_PREFIX}/bin
mkdir ${INSTALL_PREFIX}/data

cp -f ${NWCHEM_TOP}/bin/${NWCHEM_TARGET}/nwchem ${INSTALL_PREFIX}/bin
chmod 755 ${INSTALL_PREFIX}/bin/nwchem

cp -fr ${NWCHEM_TOP}/src/data      ${INSTALL_PREFIX}
cp -fr ${NWCHEM_TOP}/src/basis/libraries ${INSTALL_PREFIX}/data
cp -fr ${NWCHEM_TOP}/src/nwpw/libraryyps ${INSTALL_PREFIX}/data

# create default.nwchemrc
cat << EOS > ${INSTALL_PREFIX}/default.nwchemrc
nwchem_basis_library ${INSTALL_PREFIX}/data/libraries/
nwchem_nwpw_library ${INSTALL_PREFIX}/data/libraryyps/

```

```

ffield amber
amber_1 ${INSTALL_PREFIX}/data/amber_s/
amber_2 ${INSTALL_PREFIX}/data/amber_q/
amber_3 ${INSTALL_PREFIX}/data/amber_x/
amber_4 ${INSTALL_PREFIX}/data/amber_u/
spce  ${INSTALL_PREFIX}/data/solvents/spce.rst
charmm_s ${INSTALL_PREFIX}/data/charmm_s/
charmm_x ${INSTALL_PREFIX}/data/charmm_x/
EOS

# run test
export PSM2_MEMORY=large
export NWCHEM_EXECUTABLE=${INSTALL_PREFIX}/bin/nwchem
export OMP_NUM_THREADS=1

# some tests need this...
cp -f ${INSTALL_PREFIX}/default.nwchemrc ~/.nwchemrc

cd ${NWCHEM_TOP}/QA
./doqmtests.mpi 2 >& doqmtests.mpi.log

cp -f ${RUNTESTMD_PATH} .
sh ${RUNTESTMD} >& runtest.md.mpi.log

mkdir ${INSTALL_PREFIX}/testlog
cp -fr doqmtests.mpi.log runtest.md.mpi.log testoutputs/ ${INSTALL_PREFIX}/testlog

```

テスト

- ログのコピーが /local/apl/lx/nwchem702/testlog 以下にあります。
- runtest.md については実行だけはしたものの、詳細なチェックについては省略
 - きちんとメンテされているのだろうか？
- doqmtests.mpi
 - grpwrite: insuff eafsize でクラッシュしたテスト。grid:eaf_size_in_dbl を適切に設定すれば動作するかもしれない(未検証)
 - tests/dft_siosi3/dft_siosi3
 - tests/dft_xdm1/dft_xdm1
 - tests/ch3f_zora_shielding/ch3f_zora_shielding
 - tests/ch2_props4_bp/ch2_props4_bp
 - tests/cho_bp_props/cho_bp_props
 - tests/pkzb/pkzb
 - tests/tpss/tpss
 - tests/tpssh/tpssh
 - tests/h2o2_fde/h2o2_fde
 - tests/oh2/oh2: 失敗すること前提のテスト; 問題無し
 - tests/pspw_SiC/pspw_SiC: ?
 - tests/paw/paw: Total PAW energy : nan
 - 将来的に廃止されるということのようなので見なかったことにする
 - <https://nwchemgit.github.io/Plane-Wave-Density-Functional-Theory.html#pseudopotential-plane-wave-density-functional-theory-nwpw>
 - tests/pspw_stress/pspw_stress: ?
 - tests/tce_cr_eom_t_ozone/tce_cr_eom_t_ozone: createfile: failed ga_create size/nproc bytes (GA メモリ不足)
 - tests/tce_mrcc_bwcc/tce_mrcc_bwcc: 微小な数値エラー
 - tests/tce_mrcc_mkcc/tce_mrcc_mkcc: 微小な数値エラー
 - tests/tce_mrcc_bwcc_subgroups/tce_mrcc_bwcc_subgroups: 実行エラー(integer divide by zero)
 - 詳細は未検証
 - tests/h2o_vscf/h2o_vscf: 微小な数値エラー
 - tests/hi_zora_so/hi_zora_so: ptsalloc: increase memory in input line
 - tests/qmmm_opt0/qmmm_opt0: 微小な数値エラー
 - tests/qmmm_freq/qmmm_freq: 微小な数値エラー
 - tests/h2o_hcons/h2o_hcons: 微小な数値エラー
 - 以下、"fast" テストで実行されない部分については省略

メモ

- MPI 時はノードあたり 1 つのプロセスがデータサーバとして働く点に注意
 - ノードあたり最低でも 2 プロセス必要です。
 - ARMCI_NETWORK=MPI-PR を指定したため。なお、MPI-PR は Omni-Path 利用時の推奨設定。
- /local/apl/ix/nwchem702/default.nwchemrc に準備したファイルを ~/.nwchemrc にコピーすると色々楽かもしれません
- intel2020u2 の impi を使うと挙動がおかしいため回避
- intel2019 以降の cck を読み込んでいると libutil のリンク時に問題が出る。(usr/lib64/libutil.so とのバッティング)
- gcc10+openmpi+openblas(nwchem 任せでビルド)の場合は intel+impi+mkl よりもエラーが少ない。しかしごくわずかにインテルよりも遅いように見える。
 - ビルドの速度については gcc (gfortran) の方が圧倒的に速い。
- print debug version で詳細情報が表示できるようにしています。(make version あたりの操作に対応)