

NWChem-7.0.2

Webpage

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

Version

7.0.2

Build Environment

- 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)

Files Required

- 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
```

Build Procedure

```
#!/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${MKLROOT}/lib/intel64 -lmkl_intel_ilp64 -lmkl_intel_thread -lmkl_core -liomp5 -lpthread -lm -ldl"
export LAPACK_SIZE=8
export LAPACK_LIB="${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${MKLROOT}/lib/intel64 -lmkl_scalapack_ilp64 -lmkl_blacs_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/libraryps ${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/libraryps/
```

```

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

```

Test

- Test log files were copied to /local/apl/lx/nwchem702/testlog.
- Log files from "runtest.md" were not analyzed in detail...
- doqmtests.mpi
 - grpwrite: list of tests crashed with "insuff eafsize". Setting appropriate value using grid:eaf_size_in_dbl might fix the problem (not investigated).
 - 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: this test must be failed; no problem.
 - tests/pspw_SiC/pspw_SiC: ?
 - tests/paw/paw: Total PAW energy : nan
 - Probably, we can ignore this error. See following link:
 - <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 insufficient memory error)
 - tests/tce_mrcc_bwcc/tce_mrcc_bwcc: minor numerical error
 - tests/tce_mrcc_mkcc/tce_mrcc_mkcc: minor numerical error
 - tests/tce_mrcc_bwcc_subgroups/tce_mrcc_bwcc_subgroups: failed to run (integer divide by zero)
 - no idea
 - tests/h2o_vscf/h2o_vscf: minor numerical error
 - tests/hi_zora_so/hi_zora_so: ptsalloc: increase memory in input line
 - tests/qmmm_opt0/qmmm_opt0: minor numerical error
 - tests/qmmm_freq/qmmm_freq: minor numerical error
 - tests/h2o_hcons/h2o_hcons: minor numerical error
 - No detailed analysis were performed for larger tests (skipped when "fast" option is enabled).

Notes

- Note: one process per node will be used as a data server when running calculation.
 - At least two process per node are necessary.
 - This is the restriction from ARMCI_NETWORK=MPI-PR, which is recommended setting for Omni-Path.
- You can use /local/apl/lx/nwchem702/default.nwchemrc for your ~/.nwchemrc; path to basis set directory etc. are defined in this file.
- Binary won't run correctly if Intel MPI from PSXE 2020u2 was employed.
- libutil.so in clk (Intel PSXE 2019 or later) can be a problem upon link stage. (libutil.so from clk might be used instead of /usr/lib64/libutil.so)
- gcc10+openmpi+openblas (BUILD_OPENBLAS=y) version yields less errors than intel+impi+mkl one. However, in terms of performance, intel version seems to be slightly better than the gcc10 one.
 - For build speed, gcc (gfortran) is very much faster than intel.
- detailed information can be shown by "print debug version". (related to "make version" in the procedure above.)