GAMESS-2020Jun30

ウェブページ

https://www.msg.chem.iastate.edu/gamess/index.html

バージョン

June 30, 2020 (2020 R1)

ビルド環境

- Intel Parallel Studio XE 2019 update 5
- cmake 3.16.3

ビルドに必要なファイル

- gamess-current.tar.gz (version Jun 30, 2020)
- gmsnbo.i8.a (NBO7.0)
- rungms_rccs (インストールされた rungms をご覧下さい)
- exam43.patch

```
--- tests/standard/exam43.inp.orig      2018-03-13 11:58:15.322187865 +0900
+++ tests/standard/exam43.inp   2018-03-13 11:58:32.049289234 +0900
@@ -48,7 +48,7 @@
 ! geometry in $DATA, although this is not necessary.
 !
  $contrl scftyp=rhf runtyp=g3mp2 $end
- $system timlim=2 mwords=2 memddi=5 $end
+ $system timlim=2 mwords=10 memddi=5 $end
  $scf    dirscf=.true. $end
  $data
 Methane...G3(MP2,CCSD(T))
```

- pbs_remsh

```
#!/bin/sh
host="$1"
shift
/usr/bin/ssh -n "$host" env PBS_JOBID="$PBS_JOBID" pbs_attach $*
```

ビルド手順

```
#!/bin/sh

VERSION=2020Jun30
DIRNAME=gamess${VERSION}
INSTDIR=/local/apl/lx/${DIRNAME}

# files and patches
MYROOT="/home/users/${USER}/Software/GAMESS/gamess${VERSION}"
GAMESS_TARBALL="${MYROOT}/gamess-current.tar.gz"
GAMESS_NBOI8A="${MYROOT}/gmsnbo.i8.a"
PATCH_EXAM43="${MYROOT}/exam43.patch"
RUNGMS_RCCS="${MYROOT}/rungms_rccs"
PBS_REMSH="${MYROOT}/pbs_remsh"

PARALLEL=12

#-------------------------------------------------------------------------------
umask 0022

export LANG=C
export LC_ALL=C
```

```
module purge
module load cmake/3.16.3
module load intel_parallelstudio/2019update5
IFORTVER=19

cd ${INSTDIR}
if [ -d gamess ]; then
  mv gamess gamess-erase
  rm -rf gamess-erase &
fi

tar zxf ${GAMESS_TARBALL}
mv ${INSTDIR}/gamess/* .
rm -rf ${INSTDIR}/gamess # remove a dot file and a dot directory

sed -i -e 's/MAXCPUS=32/MAXCPUS=80/' ddi/compddi
sed -i -e "s/GMS_OPENMP='false'/GMS_OPENMP='true'/" config
sed -i -e "s/ext=log/ext=gamess/" tests/standard/checktst
for f in comp compall config lked gms-files.csh runall ddi/compddi; do
  sed -i -e "1s/.*/#\!\/bin\/csh -f/" comp
done

patch -p0 < ${PATCH_EXAM43}
cp ${PBS_REMSH} .

expect << EXPECT
spawn csh -f ./config
expect "After the new window is open"
send "\r"
expect "please enter your target machine name:"
send "linux64\r"
expect "GAMESS directory?"
send "${INSTDIR}\r"
expect "GAMESS build directory?"
send "${INSTDIR}\r"
expect "Version?"
send "\r"
expect "Please enter your choice of FORTRAN:"
send "ifort\r"
expect "Version?"
send "${IFORTVER}\r"
expect "hit <return> to continue to the math library setup."
send "\r"
expect "Enter your choice of 'mkl' or 'atlas' or 'acml' or 'libflame' or 'openblas' or 'pgiblas' or 'armpl' or 'none':"
send "mkl\r"
expect "MKL pathname?"
send "${MKLROOT}\r"
expect "MKL version (or 'proceed')?"
send "proceed\r"
expect "please hit <return> to compile the GAMESS source code activator "
send "\r"
expect "please hit <return> to set up your network for Linux clusters."
send "\r"
expect "communication library ('serial','sockets' or 'mpi' or 'mixed')?"
send "sockets\r"
expect "Optional: Build LibXC interface?  (yes/no): "
send "yes\r"
expect "And do not forget to run"
send "\r"
expect "Optional: Build Michigan State University CCT3 & CCSD3A methods?"
send "yes\r"
expect "Do you want to try LIBCCHEM"
send "no\r"
expect eof
```

```
EXPECT

## remove fpe0
#sed -i -e "s/-fpe0//" install.info

cd ddi && csh -f compddi && mv -f ddikick.x ../ && cd -

# do libxc first (according to the installation guide)
csh -f ./tools/libxc/download-libxc.csh
make -j ${PARALLEL} libxc

make modules
make -j ${PARALLEL}

GAMESS_NBOI8A_ESC=`echo ${GAMESS_NBOI8A} | sed -e 's/\//\\\\\//g'`
sed -i -e "s/NBO=false/NBO=true/" lked
sed -i -e "s/NBOLIB=.*/NBOLIB=${GAMESS_NBOI8A_ESC}/" lked
sed -i -e "s/LDOPTS='-i8'/LDOPTS='-static-intel -i8'/" lked

# retry linking; first try above would fail
csh -f ./lked

mv rungms rungms.orig
cp ${RUNGMS_RCCS} ./rungms

chmod -R o-rwx source object libcchem
chmod -R o-rwx ddi/src ddi/server ddi/olddi ddi/kickoff
find . -name "src" | xargs chmod -R o-rwx

# ---- test
cd tests/standard
export OMP_NUM_THREADS=8
../runtest ${INSTDIR}/rungms 00 1
./checktst
export OMP_NUM_THREADS=2
../runtest ${INSTDIR}/rungms 00 8
./checktst
cd ../../

ipcrm -a
```

メモ

- libxc 対応は有効にしています。最低限の動作確認まではしています。
- intel18, intel19 でテストをパスするようになっていたため、intel19 を利用しています。
    - 簡単な計算で試した限りではありますが、計算速度的には intel18, intel19 の方が少し良いように見えます。
- Omni-Path を使う RCCS のシステムでは DDI running over MPI は非効率になるため、sockets を利用しています。(Omni-Path の仕様の問題です)
- libcchem についてはビルドに関するまとまったドキュメントが存在するようには見えなかったので、今回も回避しています。