

## GAMESS 2022R2 (2022Sep30)

(2023/3/6 に導入予定のもので)

### ウェブページ

<https://www.msg.chem.iastate.edu/games/index.html>

### バージョン

2022R2 (2022Sep30)

### ビルド環境

- Intel oneAPI Compiler Classic 2022.2.1
- Open MPI 3.1.6

### ビルドに必要なファイル

- gamess-current.tar.gz (2022R2)
- gmsnbo.i8.a (NBO7.0)
- rungms\_rccs\_openmpi3 (インストールされた rungms をご覧ください)
- 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=30 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=2022Sep30
DIRNAME=games${VERSION}
INSTDIR=/apl/games/2022R2

# files and patches
MYROOT="/home/users/${USER}/Software/GAMESS/games${VERSION}"
GAMESS_TARBALL="${MYROOT}/games-current.tar.gz"
GAMESS_NBOI8A="${MYROOT}/gmsnbo.i8.a"
PATCH_EXAM43="${MYROOT}/exam43.patch"
RUNGMS_RCCS="${MYROOT}/rungms_rccs_openmpi3"
PBS_REMSH="${MYROOT}/pbs_remsh"

PARALLEL=12

#-----
umask 0022
```

```

export LANG=C
export LC_ALL=C

module -s purge
. ~/intel/oneapi/compiler/2022.2.1/env/vars.sh
. ~/intel/oneapi/mkl/2022.2.1/env/vars.sh
module -s load openmpi/3.1.6/intel2022.2.1

OMPI_DIR=/apl/openmpi/3.1.6/intel2022.2.1

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

tar xzf ${GAMESS_TARBALL}
mv ${INSTDIR}/gamess/* .
rm -rf ${INSTDIR}/gamess # maybe already empty

for f in comp compall config lkcd gms-files.csh runall ddi/compddi \
  tools/remd.csh tools/localgms tools/libxc/download-libxc.csh \
  tools/mdi/download-mdi.csh; do
  sed -i -e "1s/.*/#!\bin\csh -f/" $f
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 "oneapi-ifort\r"
expect "hit <ENTER> 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 "${MKLROOT}\r"
expect "MKL version (or 'proceed')?"
send "proceed\r"
expect "Hit <ENTER> to continue to the GAMESS DDI communications setup."
send "\r"
expect "Hit <ENTER> to set up your network for Linux clusters."
send "\r"
expect "communication library ('serial','sockets' or 'mpi' or 'mixed')?"
send "mpi\r"
expect "Enter MPI library"
send "openmpi\r"
expect "Please enter your openmpi"
send "${OMPI_DIR}\r"
expect "Optional: Build LibXC interface?"
send "yes\r"
expect "Hit <ENTER>"
send "\r"

```

```

expect "Optional: Build MDI support?"
send "yes\r"
expect "Hit <ENTER>"
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 "Build GAMESS with OpenMP thread support?"
send "yes\r"
expect "Optional: Build GAMESS with VeraChem's VM2 library? (yes/no):"
send "no\r"
expect "Optional: Build GAMESS with TINKER plug-in? (yes/no):"
send "no\r"
expect "Optional: Build GAMESS with VB2000 plug-in? (yes/no):"
send "yes\r"
expect "Optional: Build GAMESS with XMVB plug-in? (yes/no):"
send "no\r"
expect "Optional: Build GAMESS with NEO plug-in? (yes/no):"
send "yes\r"
expect "Optional: Build GAMESS with NBO plug-in? (yes/no):"
send "yes\r"
expect "Please enter the full file name of your NBO library (being careful about your choice of i4 or i8 integers):"
send "${GAMESS_NBOI8A}\r"
expect eof
EXPECT

# ?
sed -i -e "s/MDI_INSTALL/GMS_3RD_PATH/" Makefile.in
sed -i -e "s/mdi.mod/mdi*.mod/" Makefile.in

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

/bin/cp -f 3rd-party/mdi/objdir/MDI_Library/*.mod 3rd-party/include/mdi
/bin/cp -f 3rd-party/mdi/objdir/MDI_Library/*.mod object/

make modules
make -j ${PARALLEL}

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

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

# localgms may need this env variable
export GMSPATH=${INSTDIR}

sed -i -e "s/tools/localgms/rungms/" \
-e "/RUNGMS/s/s/VERSION)/VERSION) \$(NCPUS)/" \
Makefile.in

export OMP_NUM_THREADS=8
make checktest
make clean_exams

export OMP_NUM_THREADS=1

TEST_LIST="eda qmefpea efp-ci standard"

```

```
# Excluded tests
# cim: does not run with parallel, and ksh not avail
# efmo: not enough memory? problem of input?
# excitatinos: no avail test for parallel?
# neb: too long
for tp in $TEST_LIST; do
  make checktest NCPUS=8 EXAM_PATH=$tp
done

ipcrm -a
```

## テスト

- `setenv OMPI_MCA_mpi_yield_when_idle 1` を行っても `mpirun` で `oversubscribe` すると明確に速度が落ちる
  - (`ncpus=32:mpiprocs=64` のようにしても速度は上がらないようです。 `ncpus=32:mpiprocs=32` のように実行してください。)
  - 確保したコアの半分しか計算に使わない場合でも `sockets` 版と遜色ない速度で動作。ノード並列時には `sockets` 版よりもやや有利
  - HPC-X 2.11 を使用時、リビルドしていない場合 UCX が `multi thread` に未対応のため利用できず。結果、ノード間並列が非常に遅い。
    - (エラーメッセージは `hcoll` のもの。そのため `hcoll` を `disable` することで UCX が動作する可能性もあるが未検証)エラーメッセージが消えるだけで速度や状況(`-x UCX_LOG_LEVEL=data` の出力で比較)に変化は見られない。
    - Open MPI 3.1.6 で使っている UCX はスレッド対応
- Intel MPI で並列数を上げると正常に動作せず。
  - (`libhcoll` 由来の問題と思われるので、そちらを回避すれば動作するかもしれない)