

## Molpro 2024.1.0

### ウェブページ

<https://www.molpro.net/>

### バージョン

2024.1.0

### ビルド環境

- GCC 12.1.1 (gcc-toolset-12)
- Intel MPI 2021.11 / Open MPI 4.1.6
- Eigen 3.4.0
- MKL 2024.0

### ビルドに必要なファイル

- molpro-2024.1.0.tar.gz
- ga-5.8.2.tar.gz
- work.patch
- patch-argos-bininput.F
- patch-cic-ItfFortranInt.h
- patch-common\_modules-common\_cconf1
  - 大きな CI 計算のためのパラメータの変更と、一時ディレクトリのデフォルトの場所を変更しています。
  - パッチファイルは /apl/molpro/2024.1.0/patches 以下に置いています。
- token

### ビルド手順

```
#!/bin/sh

GA_VERSION=5.8.2
GA_ARCHIVE=/home/users/${USER}/Software/GlobalArrays/${GA_VERSION}/ga-${GA_VERSION}.tar.gz

MOLPRO_VERSION=2024.1.0
MOLPRO_DIRNAME=molpro-${MOLPRO_VERSION}
PARALLEL=12
BASEDIR=/home/users/${USER}/Software/Molpro/${MOLPRO_VERSION}
MOLPRO_TARBALL=${BASEDIR}/${MOLPRO_DIRNAME}.tar.gz

PATCH0=${BASEDIR}/work.patch
PATCH1=${BASEDIR}/patch-argos-bininput.F
PATCH2=${BASEDIR}/patch-cic-ItfFortranInt.h
PATCH3=${BASEDIR}/patch-common_modules-common_cconf1

TOKEN=${BASEDIR}/token

WORKDIR=/gwork/users/${USER}
GA_INSTALLDIR=${WORKDIR}/ga-temporary
INSTALLDIR=/apl/molpro/${MOLPRO_VERSION}

#-----
umask 0022
ulimit -s unlimited

export LANG=
export LC_ALL=C
export OMP_NUM_THREADS=1

cd $WORKDIR
if [ -d ga-${GA_VERSION} ]; then
  mv ga-${GA_VERSION} ga_tmp
```

```

rm -rf ga_tmp &
fi
if [ -d ga-temporary ]; then
  mv ga-temporary ga_tmp_tmp
  rm -rf ga_tmp_tmp &
fi
if [ -d ${MOLPRO_DIRNAME} ]; then
  mv ${MOLPRO_DIRNAME} molpro_tmp
  rm -rf molpro_tmp &
fi

module -s purge
module -s load gcc-toolset/12
module -s load intelmpi/2021.11
#module -s load openmpi/4.1.6/gcc12
module -s load eigen/3.4.0

tar zxf ${GA_ARCHIVE}
cd ga-${GA_VERSION}

export CFLAGS="-mpc80"
export FFLAGS="-mpc80"
export FCFLAGS="-mpc80"
export CXXFLAGS="-mpc80"

export F77=mpif90
export F90=mpif90
export FC=mpif90
export CC=mpicc
export CXX=mpicxx
export MPIF77=mpif90
export MPICC=mpicc
export MPICXX=mpicxx
export GA_FOPT="-O3"
export GA_COPT="-O3"
export GA_CXXOPT="-O3"

./autogen.sh
./configure --enable-i8 \
  --with-mpi-pr \
  --prefix=${GA_INSTALLDIR}

make -j ${PARALLEL}
make check
make install

# mkl for molpro
module -s load mkl/2024.0

cd ${WORKDIR}
tar zxf ${MOLPRO_TARBALL}
cd ${MOLPRO_DIRNAME}

patch -p0 < ${PATCH0}
patch -p0 < ${PATCH1}
patch -p0 < ${PATCH2}
patch -p0 < ${PATCH3}

export PATH="${GA_INSTALLDIR}/bin:$PATH" # where ga-config exists

CPPFLAGS="-I${GA_INSTALLDIR}/include" \
LDLFLAGS="-L${GA_INSTALLDIR}/lib64" \
./configure --prefix=${INSTALLDIR} \
  --enable-slater

```

```
make -j ${PARALLEL}
cp $TOKEN lib/.token

make tuning

MOLPRO_OPTIONS="" make quicktest
MOLPRO_OPTIONS="-n2" make test

make install
cp -a testjobs ${INSTALLDIR}/molpro*
cp -a bench ${INSTALLDIR}/molpro*
```

## テスト結果

h2o\_rvci\_dip のテストで失敗。RVCI\_DIP(4:6)の値が問題と思われる。

```
SETTING RVCI_TEST(1) = 140.24404946 CM-1
SETTING RVCI_TEST(2) = 2040.72148410 CM-1
SETTING RVCI_TEST(3) = 4309.43539874 CM-1
SETTING RVCI_DIP(1) = 6.6023777D-20
SETTING RVCI_DIP(2) = 6.9439373D-20
SETTING RVCI_DIP(3) = 5.6819299D-19
SETTING RVCI_DIP(4) = 9.6702224D-20
SETTING RVCI_DIP(5) = 1.2687290D-18
SETTING RVCI_DIP(6) = 4.3975422D-19
SETTING RVCI_DIP(7) = 4.5351150D-20
SETTING RVCI_DIP(8) = 3.5013711D-19
SETTING RVCI_DIP(9) = 1.3565481D-20
RVCI_TEST(1:3) = [ 140.24404946 2040.72148410 4309.43539874] CM-1
SETTING RVCIFREQ_REF(1)= 140.24000000
SETTING RVCIFREQ_REF(2)= 2040.72000000
SETTING RVCIFREQ_REF(3)= 4309.44000000
SETTING RVCIDIP_REF(1) = 6.6023777D-20
SETTING RVCIDIP_REF(2) = 6.9439373D-20
SETTING RVCIDIP_REF(3) = 5.6819299D-19
SETTING RVCIDIP_REF(4) = 3.2234075D-20
SETTING RVCIDIP_REF(5) = 4.2290966D-19
SETTING RVCIDIP_REF(6) = 1.3192627D-18
SETTING RVCIDIP_REF(7) = 4.5351150D-20
SETTING RVCIDIP_REF(8) = 3.5013711D-19
SETTING RVCIDIP_REF(9) = 1.3565481D-20
SETTING DRVCI = 0.00460126 CM-1
SETTING DRVCI_DIP = 2.00000008
```

Open MPI, Intel MPI のどちらでも同じエラーが発生。

非並列にした場合や MKL のかわりに OpenBLAS を使った場合でもエラーを回避できず。gcc11 に変えても同様。

## メモ

- (上記手順で intelmpi のかわりに openmpi の行を有効にすればそのまま利用できます。)
- Intel MPI を使うとシングルノードのデフォルト設定である disk option で今のところ問題が発生していない
  - Open MPI の場合には disk option を使った場合にフリーズする場合がある。
  - (disk option を使った場合計算時間が短くなることもあるが、逆に遅くなる場合もある。)
- Open MPI を使った時の方が平均的に速度が出るように見える。一方で disk option 時にフリーズの危険がある。そのため、インテル MPI 版を標準のバージョンとしている。
- Eigen 3.3 ではビルドできなかったため、Eigen 3.4.0 を利用
  - mkl+eigen 3.3 の場合は configure でエラー、openblas+eigen 3.3 の場合はビルド中にエラー