

OpenMolcas

This free version of Molcas is primarily used to perform excited state electronic structure computations with multiconfigurational CASSCF/RASSCF and CASPT2/RASPT2 methods, but it also features a number of DFT functionals and other correlation methods, continuum solvation approaches, geometry optimization and molecular dynamics. It contains a large part of the original [Molcas](https://molcas.gitlab.io/OpenMolcas/sphinx/) codebase that has been released as free and open-source software. Further documentation can be found at <https://molcas.gitlab.io/OpenMolcas/sphinx/>

The version installed on hilbert can be run in parallel mode. While this works efficient for SCF and RASSCF routines, RASPT2 is not heavily parallelized, so do not expect too much gain in performance here. Optimized versions are provided for ~~arch=hybridge~~ and ~~arch=skylake~~/icelake/zen2. The module is invoked with:

```
module load openmolcas/22.6
```

OpenMolcas uses a number of internal variables to define computational resources such as memory, disk space and location of input and output files. An example submit script for parallel runs is shown below.

```
#!/bin/bash
#PBS -S /bin/bash
#PBS -A "YOURPROJECT"
#PBS -l select=1:ncpus=4:mem=12500mb:arch=skylake
#PBS -l walltime=10:00:00

cd $PBS_O_WORKDIR

module load openmolcas/22.6

#Adapat to your needs
export INPUT=your_input_file
export MOLCAS_MEM=2000

#Molcas internal variables (leave unchanged)
export MOLCAS_NPROCS=$NCPUS
export PROG=pymolcas
export Project=$INPUT
export WORKDIR=/gpfs/scratch/$USER/$PBS_JOBID
export WorkDir=$WORKDIR
export MOLCAS_OUTPUT=$WORKDIR
export INPUTDIR=$PBS_O_WORKDIR

export OUTPUT=$INPUT.out
mkdir -p $WORKDIR
cp -r $INPUTDIR/* $WORKDIR
cd $WORKDIR

ln -s $MOLCAS/pymolcas pymolcas

# execute openmolcas
python3 $PROG -np $MOLCAS_NPROCS $WORKDIR/$INPUT 1> $WORKDIR/$OUTPUT 2>$WORKDIR/$INPUT.err

# copy back and clean up

cp -r $WORKDIR/* $INPUTDIR
rm -r $WORKDIR
```

MOLCAS_MEM is specified in megabytes and per cpu, but OpenMolcas does not always perfectly stick to this number. It is thus wise to add another ~30% of memory for allocation in PBS to prevent the job from being cancelled. For serial runs set MOLCAS_NPROCS=1.