Dalton 2020.1

ウェブページ

https://daltonprogram.org/

バージョン

2020.1

ビルド環境

- GCC 13.1.1 (gcc-toolset-13)
- Intel MPI 2021.14
- MKL 2025.0

必要なファイル

- (ファイルは以下手順中で取得)

ビルド手順

```
#!/bin/sh

VERSION=2020.1
INSTALL_PREFIX=/apl/dalton/2020.1

BASEDIR=/home/users/${USER}/Software/Dalton/${VERSION}
DALTON_URL=https://gitlab.com/dalton/dalton.git
BRANCH=Dalton${VERSION}
WORKDIR=/gwork/users/${USER}

PARALLEL=24
export LANG=C
export OMP_NUM_THREADS=1

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

module -s purge
module -s load gcc-toolset/13
module -s load intelmpi/2021.14
module -s load mkl/2025.0

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

git clone --recursive -b ${BRANCH} ${DALTON_URL}
cd dalton
for f in $(find . -name "*.py*")
 do sed -i -e "s/env python$/env python3/" $f
done
sed -i -e "s/env python$/env python3/" setup
sed -i -e "s/COMMAND python/COMMAND python3/" cmake/ConfigExternal.cmake

export MATH_ROOT=/apl/oneapi/mkl/2025.0
./setup \
 --prefix=${INSTALL_PREFIX} \
 --mpi

cd build
make -j ${PARALLEL}
```

```
ctest -j ${PARALLEL} -L dalton
make install
```

テスト

すべてパスしています。

メモ

- Open MPI 4.1.6 を使うと異常に遅くなる。原因については不明。詳細な検証は行っていない。
- OpenBLAS を使うと cc2_r12_aux_ch4_sym_a2 のテストで失敗するため回避。速度的には mkl の場合と遜色ないように見える。(あくまで手順中のテストの実行時間でのざっくりとした比較)
- gcc8 よりは gcc13 の方が速い傾向。
- インテルコンパイラ(2022 or 2023)を使うと失敗するテストが増えるため回避。インテルコンパイラ+ Open MPI の組み合わせでも速度が落ちるように見える