

## Quantum Espresso 6.5

### ウェブページ

<https://www.quantum-espresso.org/>

### バージョン

6.5

### ビルド環境

- Intel Parallel Studio 2018update4

### ビルドに必要なファイル

- q-e-qe-6.5.tar.gz

### ビルド手順

```
#!/bin/sh

VERSION=6.5
BASEDIR=/home/users/${USER}/Software/QE/${VERSION}
TARBALL=${BASEDIR}/q-e-qe-${VERSION}.tar.gz

INSTDIR=/local/apl/lx/espresso65
PARALLEL=12

# -----
umask 0022

module purge
module load intel_parallelstudio/2018update4

export LANG=C
export LC_ALL=C
ulimit -s unlimited

cd ${INSTDIR}
if [ -d q-e-qe-${VERSION} ]; then
  mv q-e-qe-${VERSION} q-e-qe-erase
  rm -rf q-e-qe-erase &
fi

tar zxf ${TARBALL}
mv q-e-qe-${VERSION}/* .
mv q-e-qe-${VERSION}/.[a-z]* .
rmdir q-e-qe-${VERSION}

export MPIF90=mpiifort
export MPIFC=mpiifort
export MPIF77=mpiifort
export MPICC=mpiicc
export MPICXX=mpiicpc

# ad hoc correction for python2
sed -i -e s/"user_input = input"/"user_input = raw_input"/ EPW/bin/pp.py

FC=ifort F90=ifort F77=ifort CC=icc CXX=icpc \
FFLAGS="-O3 -assume byterecl -ip -g" \
./configure --enable-parallel --with-scalapack=intel

echo "==== all ==== # pwall(pw ph pp pwcond neb) cp ld1 upf tddfpt hp
```

```
make -j${PARALLEL} all

for i in epw xspectra gwl plumed w90 couple; do
  echo "==== $i ====="
  make -j${PARALLEL} $i
done

for i in gipaw yambo; do
  echo "==== $i ====="
  make $i
done

cd test-suite
make run-tests-serial
make clean
make run-tests-parallel
cd ..
```

## テスト

run-tests-serial, run-tests-parallel (4 MPI)の両テストは問題無くパスしています。  
(初期値に乱数があるため、上記と同じ操作をしても iteration 回数のチェックに引っかかるケースがあるかもしれません。)

## メモ

- EPW で呼ばれる python のスクリプトはどれも python3 を想定しているようだったので、修正を加えています。
  - (エラーメッセージ "Install future. e.g. "pip install --user future" は残したままにしています)
- OpenMP については、かなりの規模にならないとメリットが得られないようだったので、今回も回避。
  - 8 ノード並列までは試しましたが、明確にメリットを得られるケースはありませんでした。それ以上ならば有利になる条件もあるかもしれません。
  - [https://www.quantum-espresso.org/Doc/user\\_guide/node18.html](https://www.quantum-espresso.org/Doc/user_guide/node18.html)
  - configure 時に --enable-openmp を指定してしまうと、1 スレッドでの実行も少し遅くなったので完全に排除。
  - 問題規模にもよりますが、やはり -nk 等を先に考えるべきですかね。
- gcc8 ではテストの一部でエラーが発生
  - 速度的には intel18 と比較して遜色はない気配(最低限のチェック時点での結果)
- intel19 MPI を使うと計算が不意に stop するため、回避。(intel compiler19 + intel mpi 18 のような状況では動きそうではあるが、未検証)
  - intel17 では intel18 版より少し速度が遅い。テストについては問題無く通過。
- 別リポジトリの GPU 版については別途検討予定