Quantum Espresso 6.5

Webpage

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

Version

6.5

Build Environment

- Intel Parallel Studio 2018update4

Files Required

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

Build Procedure

```
#!/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 ..
```

Tests

The built binaries have passed basic tests (run-tests-serial, run-tests-parallel; 4 MPI) without errors.
(You might see test failure due to the out of threshold number of iterations even when you completely follow the procedure above; due to the randomness of the initial condition.)

Notes

- Python script in EPW directory assumes /usr/bin/python is python3. We thus modified the code slightly ("sed" line in the procedure above).
  - (Related error message "Install future. e.g. "pip install --user future" is not removed.)
- We avoided OpenMP since that seems to be only effective for massive parallelization of huge system.
  - We tested up to 8-nodes parallel runs for OpenMP version performance. However, we couldn't get any meaningful speedup by OpenMP. There may be some advantages for further massive runs, though.
  - https://www.quantum-espresso.org/Doc/user_guide/node18.html
  - Specifying --enable-openmp has negative effect on performance for single thread runs.
  - Option such as -nk maybe the first target for performance optimization.
- GCC 8 version met errors on some of tests.
  - Regarding performance, performance of this is comparable to intel18 version. (Intel version may be slightly faster, though.)
- Intel 19 MPI version strangely stop running upon launching. This maybe caused by Intel MPI 19. (Intel Compiler19 + Intel MPI 18 may work. Not tested, though.)
  - Intel 17 version works fine. But that performance is slightly worse than Intel 18 version performance.
- GPU version would be discussed in different page, since the source code repository itself is different from this one.