Sector 42

before you demand the answer, seek out the question.

Compilation

Workstation

On a vanilla Linux machine, simply `./configure` and then `make` if successfully. You just need gfortran, mpi, fftw and lapack. Check make.sys to make sure BLAS_LIBS = -lblas, LAPACK_LIBS = -llapack and FFT_LIBS = -lfftw3.

If configuration is unsuccessful, bug someone for a make.sys. That’s the only thing the configuration script generates.

For 90% of the things I do, *make ld1 pw ph pp* is sufficient. ld1 is 1D DFT for atoms used to generate pseudopotentials. pw is the most popular package Pwscf, which runs DFT for solids in plane-wave basis. pp is the post-processing tool which can be used to plot band structure, charge density and many other exciting stuff. ph is from the Phonon package.

If you have intel compilers installed (especially parallel studio), follow Intel’s guide to compile with offload or native MIC. There will be some (fixable) problems with FFT. A more recent thread can be found in Intel Developer Zone here.

Just to quickly summarize the minimum one has to specify in order to configure espresso-5.3.0 with parallel studio 15.0.6 (remember to source psxevars). First:

```
./configure CC=mpiicc MPIF90=mpiifort F77=mpiifort --with-scalapack=intel
--with-hdf5
```

You will notice that the configure script fails to locate Intel’s fftw3 package. Next, open `make.sys` and add `-I$(MKLROOT)/include/fftw` to `IFLAGS`, and make sure `-D__FFTW3` is used instead of `-D__FFTW` (internal copy). Finally, Remove `-par-report0 and -vec-report0` to silence deprecation warnings. You should now have a working `make.sys`

If modded by QMCPACK, add: `--with-hdf5 HDF5_DIR=$HDF5_HOME` OpenMP support seems experimental for now.

update 2019-01-22: qe-6.3 converter may require a bit more tweaking to get HDF5 to work properly, because it assumes hdf5 is compiled from source and installed in a stand-alone folder. If you have hdf5 compiled in `/usr/local/hdf5` for example, the following command is supposed to work

```
./configure CC=mpiicc MPIF90=mpiifort F77=mpiifort --with-scalapack=intel
--with-hdf5=/usr/local/hdf5
```

If HDF5_LIBS not picked up, then manually edit the generated make.inc file.
DFLAGS = -D__DFTI -D__MPI -D__SCALAPACK -D__HDF5_C -DH5_USE_16_API
HDF5_LIB = -L/lib64 -lhdf5 -lhdf5_fortran -lhdf5_hl
Also check CFLAGS and FFLAGS to make sure -O3 exists for optimal performance.

hdf5-1.10.3 can compile correct after dos2unix.

Hint: To silence the compiler warnings, change -openmp to -qopenmp and remove -par-report0 and -vec-report0 in make.sys. To speed up the compilation process, use -j flag of make. This will get you 75% of the way then fail, but now `make pw` doesn't take nearly as long.

If you keep failing, download a clean espresso tarball for your sanity.

So far I haven't been able to get QE 5.3.0 to work with intel 16.0.2 compiler. Parallel execution gets stuck for no reason.

Supercomputer

TAUB/GOLUB

As of 2016/09/19, Parallel Studio 2015 is available on University of Illinois Urbana-Champaign campus cluster, thus the same instructions above still works.

Update 2018/05/21, `module load python` automatically loads `intel/17.0` and `gcc/6.2.0`. We should now have `icc` and `ifort` (check using `which ifort` for example). We want to use intel FFT and SCALAPACK for optimal performance. One can either export environment variables before `./configure`, or give definitions directly to `./configure`. I prefer the latter because it does not contaminate user environment after build. See Glenn K. Lockwood's blog for a good reference.

```
./configure \
CC=icc \
CXX=icpc \
FC=ifort \
F77=ifort \
BLAS_LIBS="-lmkl_intel_lp64 -lmkl_sequential -lmkl_core" \
SCALAPACK_LIBS="-lmkl_scalapack_lp64 -lmkl_blacs_openmpi_lp64" \
FFT_LIBS="-lmkl_intel_lp64 -lmkl_sequential -lmkl_core"
```

Make sure `-D__DFTI` and `-D__SCALAPACK` are in your `DFLAGS` in `make.sys`. Last but not least, to enable wavefunction conversion for QMCPACK, edit `make.sys`. Add `-D__HDF5_C -DH5_USE_16_API` to `DFLAGS`, and add `-L/lib -lhdf5_hl -lhdf5` to `HDF5_LIB`.

Finally, `make pw -j16`; `make pp -j16`.

I have gone through the above procedure and produced binaries in `/projects/physics/apps/qe-6.2.1/bin`. Please feel free to use them, but do double check their correctness for your application.

TITAN

To compile on Titan at Oak Ridge, follow instructions in install/Make.CRAY-XK7.[your choice].

```
module load cray-hdf5
./configure --enable-parallel --enable-openmp --with-scalapack --with-hdf5
ARCH=crayxt
```

I used PrgEnv-pgi and had to change make.sys:

1. Update TOPDIR
2. Update /opt/cray/libsci/[version] (module display cray-libsci)
3. Disable Scalapack (this is bad, but better than not compiled)