Quantum Espresso Recipe

for using Intel Software Development Tools.

This recipe summarizes some best known methods for using Intel Software Development Tools as well as running Quantum Espresso (QE) on Intel based systems. The steps have been tested and updated to cover QE 5.4, but have worked on QE 5.3 previously.

Building Quantum Espresso

The configure build system (Autotools) is further enhanced by a collection of configure-wrapper scripts, which are used to call the actual configure script.

1. Download the configure wrapper scripts for QE and ELPA from the "timings" branch of the QENESAP repository at GitHub. This repository and branch is temporarily used to provide the scripts.
$ git clone --branch timing https://github.com/NERSC/QENESAP.git
Please remember, that in contrast to Subversion one cannot checkout a subdirectory of a Git repository. Alternatively, one can download the content file by file using the following repository URL: https://github.com/NERSC/QENESAP/tree/timings/dev/intel.make.

2. Download ELPA's latest release from http://elpa.mpcdf.mpg.de/elpa-tar-archive, and unarchive the source code.
$ wget http://elpa.mpcdf.mpg.de/html/Releases/2016.05.002/elpa-2016.05.002.tar.gz
Copy the configure wrapper scripts into the ELPA root directory (where the actual configure script is located)
$ cp /path/to/QENESAP/dev/intel.make/configure-elpa* .
Make the Intel Compiler available for your command line:
$ source /opt/intel/compilers_and_libraries_2017.0.064/linux/bin/compilervars.sh intel64
Please note that 2017 series is the version which accompanies the release of the hardware, and that this (currently Beta) version is likely to reproduce the better performance. Otherwise, KNL is supported since the 2015 series of the Intel Compiler.

Run the wrapper for the desired configuration, for example:

$ ./configure-elpa-knl.sh

The configure wrapper already produced a directory layout which is suitable for building QE using this version of ELPA. Edit line #5 of the wrapper script to adjust the location. Build and install the ELPA library.

$ make -j ; make install

Please note, that QE's version of ELPA is pretty outdated and the current version of ELPA at least comes with Intel AVX2 support (with AVX-512 in the works).

3. Temporarily (and only when targeting KNL), an own build of NETLIB's ScaLAPACK is necessary for performance. The performance defect in Intel MKL is reported and being worked (2x slowdown for p?gemm on KNL). The recipe have been downloaded in step #1, but can also be viewed under:

https://github.com/NERSC/QENESAP/blob/timings/dev/intel.make/SLmake.txt

4. Download QE 5.4.0 from http://www.qe-forge.org/gf/project/qe/frs/, and unarchive the source code.

$ wget http://www.qe-forge.org/gf/download/frsrelease/211/968/espresso-5.4.0.tar.gz

Copy the configure wrapper scripts and patches into QE's root directory (where actual configure script is located)

$ cp /path/to/QENESAP/dev/intel.make/configure-qe* .

The Intel Compiler should be already available on your command line (see step #2 about building ELPA).

Run the wrapper for the desired configuration, for example:

$ ./configure-qe-knl.sh

Please note, that when you have edited the configure wrapper script for ELPA, you may also need to edit line #14 of the configure wrapper for QE. Build the QE binary:

$ make pw -j

Running Quantum Espresso

The given set of run scripts may help to get started, and can be used as a template, which need to be adapted for the own environment (cluster).

1. Download the run scripts for QE from the "timings" branch of the QENESAP repository at GitHub. This repository and branch is temporarily used to provide the scripts.
   $ git clone --branch timing https://github.com/NERSC/QENESAP.git
   Please remember, that in contrast to Subversion one cannot checkout a subdirectory of a Git repository. Alternatively, one can download the content file by file using the following repository URL:
   https://github.com/NERSC/QENESAP/tree/timings/dev/intel.run.

2. Copy the scripts into QE's root directory:
   $ cp /path/to/QENESAP/dev/intel.run/* .
   Read "run-dna.txt" (or perhaps "run-bsub.txt"), and adapt "run-dna.sh" ("run-bsub.sh"), "mynodes.sh", and "enable" according to your environment (cluster). For KNL, bring the system(s) into the desired cluster/memory mode. For the time being (no particular enabling of QE using the FASTMEM attribute, etc.), one may rely on the **Cache** memory mode with either **Quadrant or SNC4** cluster mode.

3. For example, run the following command line from inside of QE's root directory (assuming the AUSURF112 workload is in reach).
   $ ./run-dna.sh /path/to/ausurf.in 1 68 -npool 2 -ntg 32 > asrf-cq68x1.txt # cq stands for cache/quadrant
   The Intel Compiler runtime should be available on your command line (see step #2 about building ELPA, and sourcing the compiler).