Quantum ESPRESSO

CMAKE – A new build system for QE

Quantum ESPRESSO Developer Forum, Trieste, 16 January 2020

Dr. Daniele Cesarini (d.cesarini@cineca.it)

Federico Ficarelli (f.ficarelli@cineca.it)

Outline

- Cross-compiling with Autotools
- Issues with Autotools
- Why CMAKE
- Work done
- Work in progress & Future release

Cross-compiling with Autotools

Federico encountered a lot of issues when he tried to compile QE for ThunderX2 server (ARM-based architecture and ARM flavor of LLVM).

Cross-compiling with Autotools

Federico encountered a lot of issues when he tried to compile QE for ThunderX2 server (ARM-based architecture and ARM flavor of LLVM).

He was not able to fix the autotool to compile QE even for a simple LaxLib test.

Cross-compiling with Autotools

Federico encountered a lot of issues when he tried to compile QE for ThunderX2 server (ARM-based architecture and ARM flavor of LLVM).

He was not able to fix the autotool to compile QE even for a simple LaxLib test.

He gave up due the limited support of the cross-compiling capability of autotools. He move to CMAKE and realized a working prototype build system in just half a day.

Issues with Autotools

- Mess up cross-compilation ability! Very limited support and it is very simple to not honor the sysroot specification.
- Truly ARCANE m4 macro syntax combined with verbose, twisted shell scripting for tests for "compatibility", etc.
- Does a HUGE amount of testing for problems with ancient, broken compilers and configurations that NOBODY currently uses with pretty much anything production in this day and age!
- When it breaks, you're going to spend HOURS chasing your tail trying to sort out the things that whomever wrote the scripting got wrong to sort out your build.
- QE toolchain is full of ancient that nobody knows why they are still present in last releases of QE (DEC Alpha, etc.).

Why CMAKE

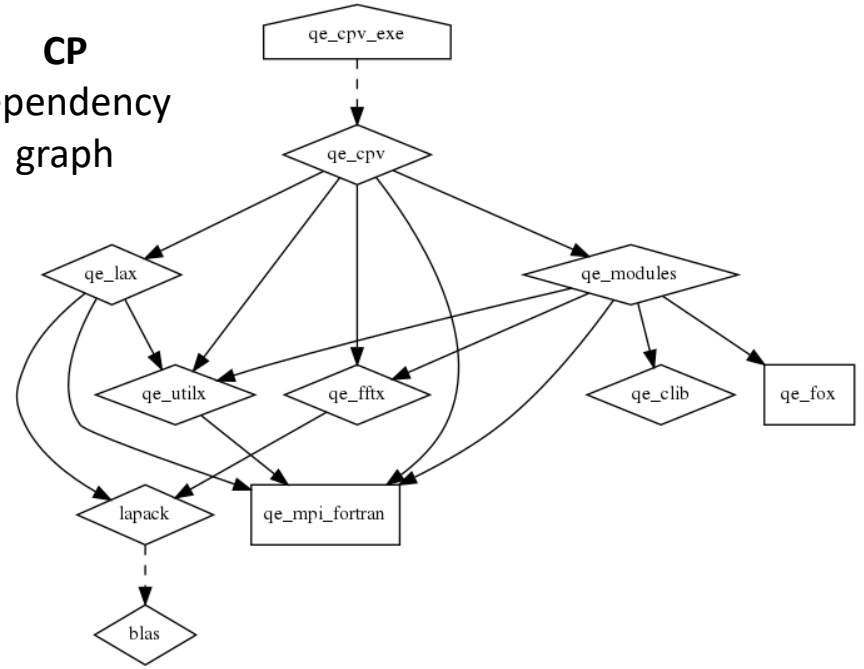
- CMake stands for Cross-platform Make, it is a cross-platform free and open-source software tool for managing the build process of software using a compiler-independent method.
- Complicated directory hierarchies and applications that rely on several libraries are well supported by Cmake (as instance QE).
- Broad support for build tools and IDEs.
- Good support for Fortran code bases and Fortran modules
- CMAKE automatically solve dependencies, it is able to compile in parallel QE.
- Very nice support of GIT and submodules.
- Mature, almost de-facto standard.

Work Done

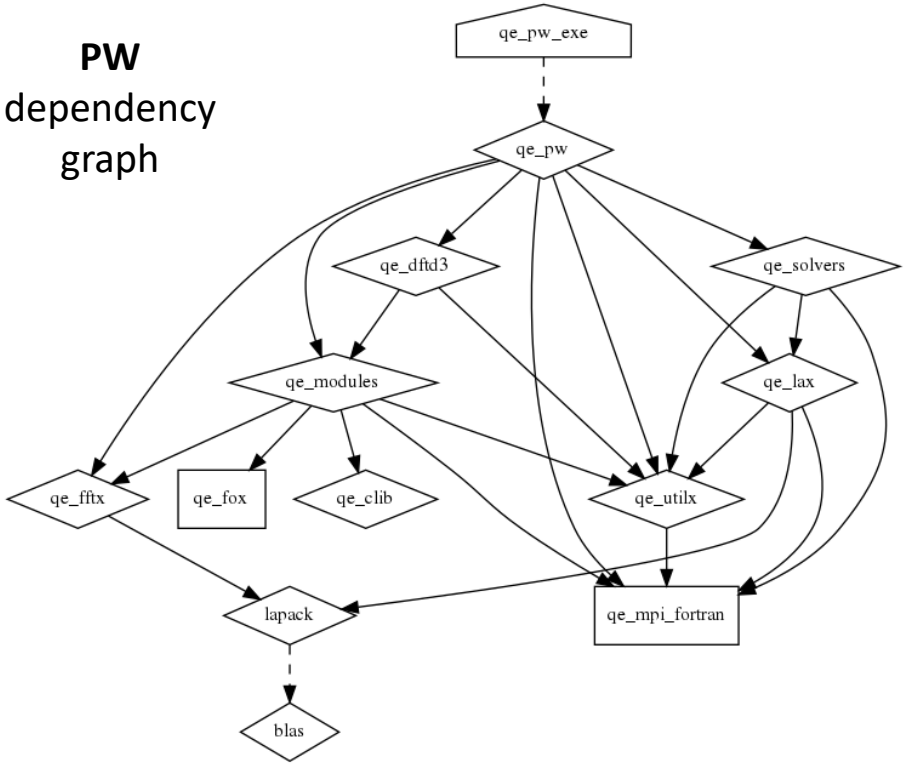
- The CMAKE build system is currently ready for the following executables: **cp.x**, **manycp.x**, **pw.x**

- The CMAKE build system is ready for the following libraries: **LAX**, **UTILX**, **Solvers**, **Modules**, **FFTX**, **DFTD3**, **CLIB**, **FOX** (vendored via a git submodule pointing at upstream)

CP dependency graph



PW dependency graph



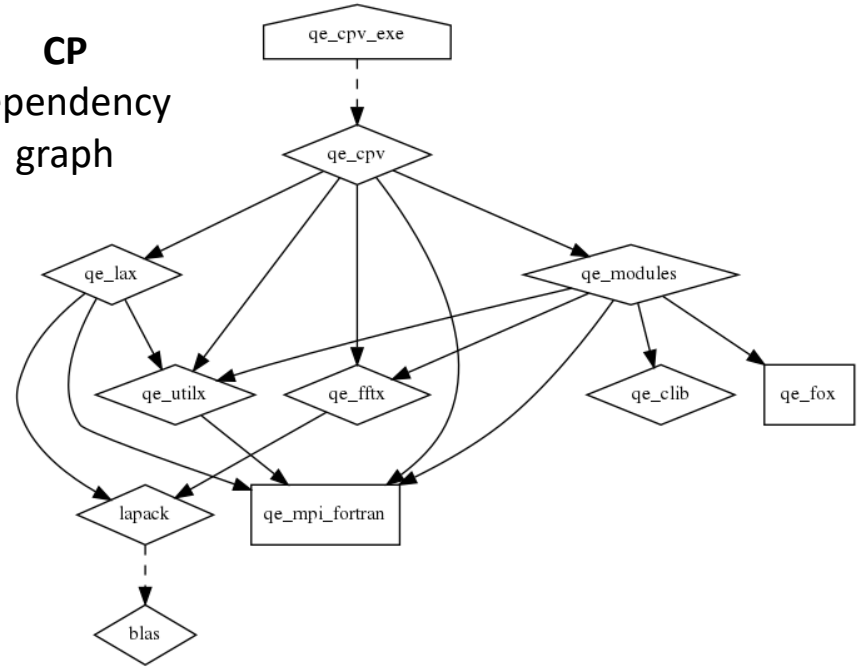
Work Done

- The CMAKE build system is currently ready for the following executables: **cp.x**, **manycp.x**, **pw.x**

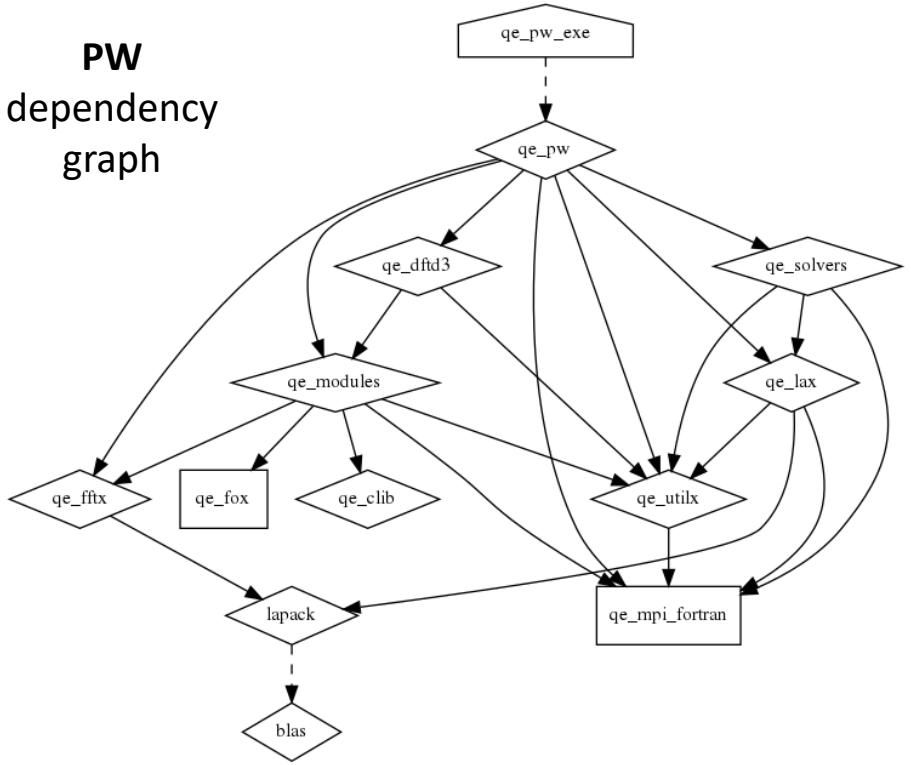
- The CMAKE build system is ready for the following libraries: **LAX**, **UTILX**, **Solvers**, **Modules**, **FFTX**, **DFTD3**, **CLIB**, **FOX** (vendored via a git submodule pointing at upstream)

- We compiled QE using CMAKE for the following architectures:
  - **x86-64**: Ubuntu 18 LTS (GNU, Intel), Red Hat 7 (GNU, Intel), Centos 7 (GNU, Intel), Mac Osx (GNU)
  - **IBM Power8**: Red Hat 7 (GNU)
  - **ARMv8**: Centos 7 (GNU, ARM-Clang/Flang)

CP dependency graph



PW dependency graph



Work Done

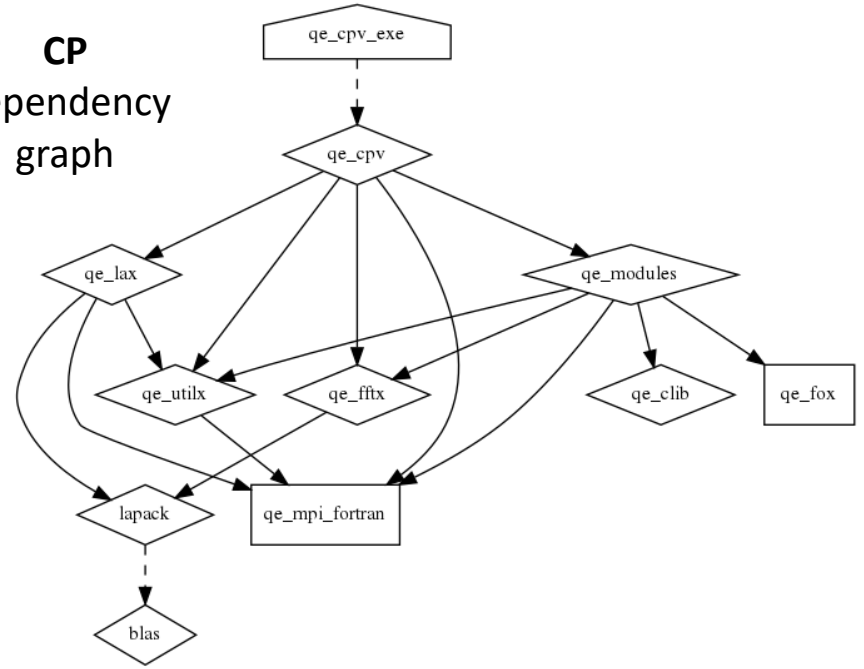
- The CMAKE build system is currently ready for the following executables: **cp.x**, **manycp.x**, **pw.x**

- The CMAKE build system is ready for the following libraries: **LAX**, **UTILX**, **Solvers**, **Modules**, **FFTX**, **DFTD3**, **CLIB**, **FOX** (vendored via a git submodule pointing at upstream)

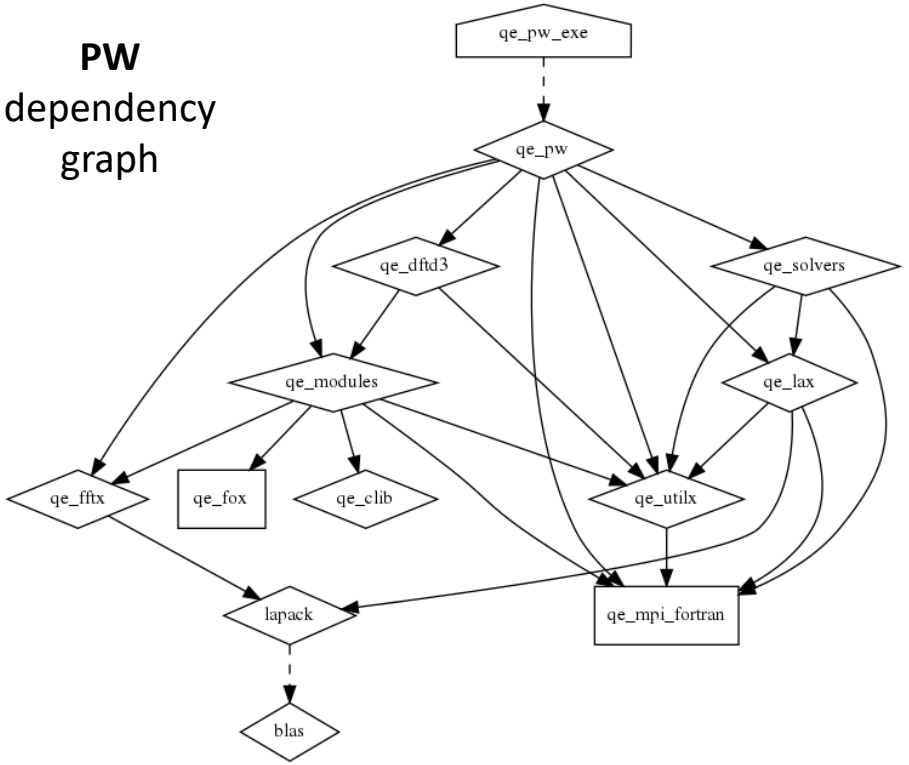
- We compiled QE using CMAKE for the following architectures:
  - **x86-64**: Ubuntu 18 LTS (GNU, Intel), Red Hat 7 (GNU, Intel), Centos 7 (GNU, Intel), Mac Osx (GNU)
  - **IBM Power8**: Red Hat 7 (GNU)
  - **ARMv8**: Centos 7 (GNU, ARM-Clang/Flang)

- **Git update**: pull request of Federico Ficarelli is still pending

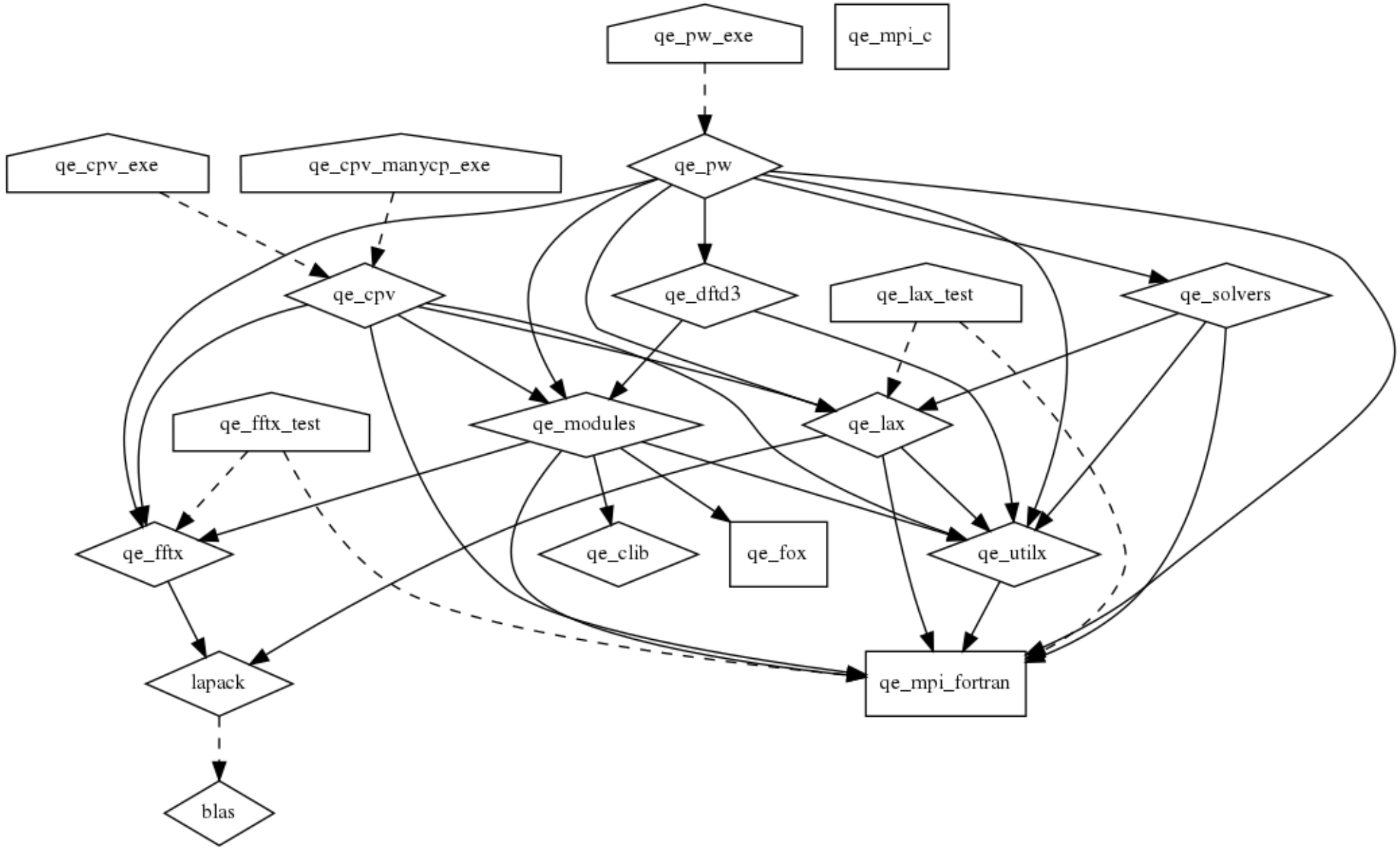
CP dependency graph



PW dependency graph



Global Dependency Graph



Work in Progress & Future release

We are working to conclude the build system for the libraries and the executables of QE. Today are missing the following components: ***COUPLE, Doc generation, EPW, GWW, HP, LR_Modules, NEB, Phonon, PP, PWCOND, PlotPhon, QHA, TDDFPT, XSpectra, archive, atomic, pseudo, upftools.***

Work in Progress & Future release

We are working to conclude the build system for the libraries and the executables of QE. Today are missing the following components: ***COUPLE, Doc generation, EPW, GWW, HP, LR_Modules, NEB, Phonon, PP, PWCOND, PlotPhon, QHA, TDDFPT, XSpectra, archive, atomic, pseudo, upftools.***

Next steps are:

- Community acceptance and discussion about the real usefulness of the changes.
- Integrate CMAKE toolchain in the continuous integration to maintain the build system up-to-date.
- Create common compiler flags configuration files to enclose architecture-specific optimization flags.
- Develop CMAKE finders to automatically discover Scalapack and ELPA.

Work in Progress & Future release

We are working to conclude the build system for the libraries and the executables of QE. Today are missing the following components: ***COUPLE, Doc generation, EPW, GWW, HP, LR_Modules, NEB, Phonon, PP, PWCOND, PlotPhon, QHA, TDDFPT, XSpectra, archive, atomic, pseudo, upftools.***

Next steps are:

- Community acceptance and discussion about the real usefulness of the changes.
- Integrate CMAKE toolchain in the continuous integration to maintain the build system up-to-date.
- Create common compiler flags configuration files to enclose architecture-specific optimization flags.
- Develop CMAKE finders to automatically discover Scalapack and ELPA.

In future we hope to see the stable release of CMAKE build system merged in QE-GPU repository -> we plan to extend CMAKE toolchain to QE-GPU (also targeting ARM+GPU)!

N.B. WE NEED COMMENTS AND SUGGESTIONS FROM THE COMMUNITY!!!

Thank you, questions?