QCOR Documentation

Release 1.0.0

Alex McCaskey

Mar 24, 2022

CONTENTS:

QCOR is a single-source C++, retargetable quantum-classical compiler enabling low and high level quantum programming, compilation, and execution. QCOR represents the integration of the XACC quantum framework with the ubiquitous Clang/LLVM classical compiler frameworks. The QCOR compiler extends both of these infrastructures via simple plugin extensions, and enables programmers to express quantum kernel expressions (functors containing quantum code) alongside standard C++.

CHAPTER

ONE

DESCRIPTION OF ARCHITECTURE

For class documentation, check out this site.

CHAPTER

TWO

QCOR DEVELOPMENT TEAM

QCOR is developed and maintained by:

- Alex McCaskey
- Travis Humble
- Eugene Dumitrescu
- Dmitry Liakh
- Thien Nguyen
- Daniel Chaves-Claudino
- Tyler Kharazi
- Anthony Santana

CHAPTER

THREE

QUESTIONS, BUG REPORTING, AND ISSUE TRACKING

Questions, bug reporting and issue tracking are provided by GitHub. Please report all bugs by creating a new issue. You can ask questions by creating a new issue with the question tag.

3.1 Installation

3.1.1 Quick-Start with Docker

To get up and running quickly and avoid installing the prerequisites you can pull the `qcor/qcor` Docker image. This image provides an Ubuntu 18.04 container that serves up an Eclipse Theia IDE. QCOR is already built and ready to go.

3.1.2 Dependencies

Note that you must have a C++17 compliant compiler and a recent version of CMake (version 3.12+). You must have XACC installed (see Bulding XACC)

Easiest way to install CMake - do not use the package manager, instead use *pip*, and ensure that */usr/local/bin* is in your PATH:

```
$ python3 -m pip install --upgrade cmake
$ export PATH=$PATH:/usr/local/bin
```

For now we require our users build a specific fork of LLVM/Clang that provides Syntax Handler plugin support. We expect this fork to be upstreamed in a future release of LLVM and Clang, and at that point users will only need to download the appropriate LLVM/Clang binaries (via *apt-get* for instance).

To build this fork of LLVM/Clang (be aware this step takes up a good amount of RAM):

```
$ apt-get install ninja-build [if you dont have ninja]
$ git clone https://github.com/hfinkel/llvm-project-csp llvm
$ cd llvm && mkdir build && cd build
$ cmake -G Ninja ../llvm -DCMAKE_INSTALL_PREFIX=$HOME/.llvm -DBUILD_SHARED_LIBS=TRUE -
↪DCMAKE_BUILD_TYPE=Release -DLLVM_TARGETS_TO_BUILD="X86" -DLLVM_ENABLE_DUMP=ON -DLLVM_
↪ENABLE_PROJECTS=clang
$ cmake --build . --target install
$ sudo ln -s $HOME/.llvm/bin/llvm-config /usr/bin
```

3.1.3 Building from Scratch

Note that, for now, developers must clone QCOR manually:

```
$ git clone https://github.com/qir-alliance/qcor
$ cd qcor && mkdir build && cd build
$ cmake ..
$ [with tests] cmake .. -DQCOR_BUILD_TESTS=TRUE
$ make -j$(nproc) install
```

Update your PATH to ensure that the `qcor` compiler is available.

```
$ export PATH=$PATH:$HOME/.xacc/bin (or wherever you installed XACC)
```

3.2 Basics

3.3 Indices and tables

- genindex
- modindex
- search