Contents

1
Introduction

1.1 Motivation

Quantum circuit compilation is one of the oldest problems in quantum computing. The problem is to find a sequence of elementary gates that implements a given unitary transformation. Often stated as a constrained optimisation problem: find a "quantum circuit" implementing the desired computation that minimises a cost function such as the number of gates of type X, or the *depth* of the computation.

As quantum computers become bigger and computational demands from users increase, new challenges are emerging in quantum compilation

- Circuits get bigger, making the compilation more computationally expensive.
- To scale and improve performance, quantum hardware architecture becomes more complex. The result is more special-purpose gate sets, additional execution constraints and more edge cases in the compilation problem.
- Given the cost and scarcity of quantum computational resources, quantum compilation will increasingly be expected to leverage large classical compute clusters to lighten as much as possible the quantum resource requirements.

The goal of this thesis is to explore new research questions that arise from these developments. All of these changing demands also come in hand in hand with substantial engineering challenges.

1.2 Contributions

The work we present here contributes towards the scaling up of quantum compilation in three ways.

Quantum Supercompilation. We explore the new challenges for compilation that will arise from larger quantum computations, and in particular from the inevitable merging of quantum and classical large compute clusters. We explore current tools, their limitations and the requirements for future solutions. See chapter 3.

Novel solutions to compilation. Building on a proven *supercompilation* approach, we present two novel results that significantly improve the viability and performance of rewrite rule-based quantum optimisation. The first is a fast pattern matching algorithm (chapter 4). This is semantics-agnostic, meaning that it can be applied not only on quantum circuits but also on hybrid quantum-classical computations and even pure-classical Machine Learning model training. The second is a contribution to circuit optimisation using the ZX-calculus, where we show how the pattern matching above can be used in causal-flow preserving ZX optimisations (chapter 5).

Tooling for large scale compilation. We make performance-focused open source infrastructure available that can be used to scale compilation to large quantum computations. Significant enginnering efforts have gone into the design and implementation of this tooling, which we describe in chapter 6.

Current Research State and Timetable

Main contributions

Chapters 4 to 6 form the core of my research (and thesis). The contents of chapter 4 are in peer-review, with a view of being published in the proceedings of the International Conference on Formal Structures for Computation and Deduction (FSCD 24). Chapter 5 will be written as part of the next paper that I aim to submit in the next few months. I am currently targetting a submission to QPL24 but the implied timeframe is somewhat optimistic. It may be that an alternative venue with a later deadline is more appropriate.

Chapter 6 will present a new software architecture for quantum compilation, for which a significant code base already exist. The vision is that this work can be extended and packaged in a way that is a significant contribution to the tooling available to the quantum compilation community.

Introductory chapters and litterature review

The quantum side of the background material for chapter 2 is content that I have covered and summarised many times before, so I do not anticipate any difficulties. I expect the larger effort will be in the classical side of the background material, where I would like to do some more reading into compiler optimisation and infrastructure – I think for this thesis it will be important to put quantum compilation within a larger compilation context. On the "hands-on" side, I have some experience working with LLVM and QIR as part of my contributions to the `pytket` compiler.

Finally, I am suggesting chapter 3 as an in-depth analysis of the current state and direction of quantum compilation, with a focus on compute intensive (but scalable) optimisation techniques. I intend this chapter to be composed of one half literature review and one half a critical study and discussion of the current tools and solutions. The hope is that this will highlight some of the key challenges in compilation, and frame the contributions of the rest of the thesis. This is also the first chapter I intend to write and use to direct the focus of the remaining research.

Timeline

Until March/April, the writing of the QPL24 paper and addressing reviewers' comments for the FSCD24 paper will be the main focus. This will form the basis of chapter 5 and chapter 4 respectively. April to May will be spent doing background reading and writing of chapter 3. I then aim to get the bulk of the thesis in good shape as we go into the summer, with the exception of chapter 6. I look forward to taking a break from writing at that point to focus on the implementation of the new compiler infrastructure. A secondary objective of that work will be to publish benchmarking results for quantum circuit optimisation. We already have some promising early results in this direction that I believe are worth expanding on. The remaining time will be spent writing chapter 6, writing conclusion and introduction as well as addressing comments.

Aimed submission time: 1st November 2024.