Recap of Summer:
Blind Delegated Quantum Computation

Sean Decker

August 21 2018

Introduction

Over the last 10 weeks at the AT&T Palo Alto Foundry, I have been researching into the field of Blind Delegated Quantum Computation (BDQC). The goal of BDQC is to allow a client with limited or no quantum abilities to be able to delegate some quantum computation to a quantum computer with a guarantee that the quantum computer is blind to their. In other words, in a BDQC protocol, some some client, Alice, who has some function $f(\cdot)$ that she is interested in applying to some input x, is able to use the quantum resources of some quantum server, Bob, to compute $f(x)$, all the while Bob learns nothing about x, $f(\cdot)$, or $f(x)$

Goals

In this paper, I hope to go over my work over the summer at the AT&T Palo Alto Foundry and describe some of my findings. This is in the hopes that it be possible for this work to possibly be used in future work.

Contents

1 Measurement Based Quantum Computing

Most every Blind Delegated Quantum Computation is based off of Measurement Based Quantum Computation. MBQC is universal for quantum computation, like the gate based model and unlike adiabatic quantum computation, but computations are carried out in a fundamentally different way, but it is this different way to compute that lends MBQC to be perfect for BDQC.

Resources

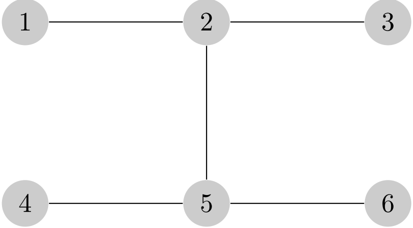
First, if need be for an overview of the standard for quantum computation (at least at the moment), check out Intro_to_QC.pdf [1] With a firm understanding of traditional QC, one can move on to read Intro_to_MBQC.pdf. From there, all other resources in the folder can be of use depending on what your interest is If you need to learn about computation on cluster states, cluster_states.pdf and MBQC_on_cluster_states.pdf both help you once you get used to the notation. If you need help understanding how to apply corrections at the end of your computation or are interested in proof that MBQC can be deterministic, then read Determinism_1_Way_Model.pdf. Practically it would be easier if we could confine our measurements to only one plane; all pdfs concerning the universality of the X-Y planes discuss this.

[1] all referenced pdfs in this section are located in the background_on_MBQC folder

Something Important that I learned

The thing that I most struggled with in this project was applying the corrective measurements at the end of the computation. The trick, that I don't believe anyone really went over in the pdfs is the following:

Say that we have the following graph state which we want to use to preform some computation



Firstly, we need to understand in what order we are going to measure qubits in order to do our computation. This requires some idea of flow (like an ordered from out input qubits to output qubits). From convention, we usually understand computation to propagate from left to right. Thus, we can see our input qubits as 1 and 4. Then our output qubits are 3 and 6. Thus, using the parallel of the gate based model, we can imagine 1, 2, and 3 as making up a wire and then 4, 5, 6 as another wire. This means that our measurement order is going to be 1 and 4 then 2 and 5 then we leave 3 and 6 as our output qubits. The wires can be measured in parallel. This is gone into in depth in One_Way_QC.pdf

Okay, now that we know measurement order, we need to understand corrections, and this is what had me stuck for quite some time. So first we measure 1 (as we just learned from the flow we parsed; equivalently we could measure 4) in the measurement basis required for the calculation that we are trying to do. After we do that, we will either measure a 0 or a 1. If 0, we don't need to do anything, everything is running smoothly. If we measure a 1, then we need to correct the resulting state so that it corresponds to what we would have if we had measured 0. This can be thought of as probabalistically going down one road or another road, and we need to force the roads to merge into one.

To correct for getting a 1, we need to apply an X gate to each neighbor of the measured qubit and apply an Z gate to all neighbors of neighbors. Thus in the case of qubit 1, if we measure 1, then we need to apply X to 2 and apply Z to 3 and to 5. If we measure 1 on 4, then we need to apply X to 5 and Z to 2 and 6. After this correction, we have merged our roads and we can continue on our way, disregarding 1 for the rest of the computation.

It is true that the Xs and the Zs can be absorbed into the measurement angle of the following qubits and furthermore, the Zs can be simply propelled along the flow so that they are only accounted for with the final output qubits.

Algorithms

With this very explicit description of how corrective measurements are meant to work, you can implement any computation that is described in the papers that are attached.

Blind Delegated Quantum Computation Protocols

With this background understanding, we can understand Blind Delegated Quantum Computation Protocols.

Resources

A great starting overview and resource is written by Joe Fitzsimmons, private_quantum_computation_Introduction.pdf. This resource is a little date, but it gives a great overview of the first few blind delegated quantum computation protocols that were designed before 2017. All of the protocols in the articles require a quantum channel, and some other quantum ability on the client's side. After this article there are a few classical client delegated protocols that were invented soon after. These are gone over in Classically_Driven_Blind_Quantum_Computation.pdf and Classical_Verification_and_BDQC.pdf.