Qiskit Backend Specifications for OpenQASM and OpenPulse Experiments
David C. McKay, Jay M. Gambetta, et al
arXiv:quant-ph
16 May 2019
Dylan Miracle
ICS 698-02
Spring 2021
Mar 2, 2021
Dr. Jigang Liu

Qiskit Backend Specifications for OpenQASM and OpenPulse Experiments

Dylan Miracle
Department of Computer Science
Metropolitan State University
St. Paul, Minnesota, USA
dylan.miracle@my.metrostate.edu

March 2, 2021

1 Background

Quantum computing is available to users now. These quantum computers are known as noisy intermediate scalable quantum (NISQ) devices. Qiskit is the most widely used framework for programming quantum computers. The article specifically calls out three types of users: algorithm developers, circuit designers, and quantum physicists. This paper explores the technical detail of how to provide all types of users with the information they need from the qiskit framework.

2 Main idea/conclusion

This paper demonstrates how a user can use QASM to define a quantum circuit and how to use Qiskit to submit the circuit to simulators and quantum hardware at IBM. The quantities of interest are different for different types of users. While some users want a simulator that gives access to idealized quantum states, others are interested in the impacts of noise on the performance of certain gates. OpenQASM and OpenPulse are two low level languages that Qiskit gives users access to for defining quantum data.

3 Support facts/algorithms/methods

A general overview of the qiskit API consists of a Provider (hardware if available), a Backend (simulator or hardware handler), Job (execution) and Result Data Structure. This method will outline all qiskit experiments and gives users a framework to design and implement runs on quantum computing hardware or

simulators. This is also important because it provides a consistent data structure for result data that means we can write tools for interpreting results that are extensible.

Further breaking down the backend we can find the complete configuration of a backend via the Backend.configuration() method. This method will return the backend configuration data structure. This includes name, version, available qubits, available gates and other information.

4 Arguments/disagreements/concerns

It is not clear how qiskit fits into the ecosystem of other quantum frameworks. Qiskit is used at IBM but other providers of quantum software are using other frameworks. In digital computing businesses often weigh the impact of "vendor lock-in". Will using qiskit create vendor lock in requiring users to stay in the IBM quantum space?

All the programs used in this paper are gate level, meaning they are designed as a series of unitary transforms on qubits. This does not allow for abstracted models of computing that will be necessary for quantum computing to reach a broader audience.

5 Interesting findings

Code for creating a bell state using OpenQASM is developed. This code shows the basics of creating a simple gate circuit, but taking two qubits, putting one into a superposition of the computational basis stated, then entangling the two qubits into what is known as a Bell state. This is interesting because the fundamental differences between quantum computing and digital computing is that qubits can experience superposition and multiple qubits can be entangled.

OpenPulse is a language for defining a different kind of computation than gate model computation. This uses a pulse model that takes into account the time dependence of a quantum calculation, where the gate model ignores time dependence and must be done well withing the coherence time of whatever device is being used. The pulse model allows researchers and algorithm developers to consider the time dependance of quantum hardware, ultimately resulting in more realistic simulations and hardware runs that are less susceptible to coherence time problems.

6 Quotations

Qiskit is designed to accommodate three user levels – the algorithm designer, the circuit design and the quantum physicist. The algorithm designer wants to research and develop quantum algorithms and applications. The circuit designer wants to optimize quantum circuits for a given device and explore topics such as error correction, quantum verification and validation, and circuit optimizations.

The quantum physicist wants to optimize and design quantum gates to perform the best circuit on a given device.

The motivation for this specification is to enable experiments that are beyond the scope of OpenQASM. For example, the OpenPulse specification enables user experimentation of improved decoupling schemes, calibrations, pulse-shaping, and optimal control.