Ion Trap Compiler Project

Quantum Computing Project M1 for NCSU's ECE 592. *Compiler for Duke Ion Trap*. "from a high-level language (Qiskit, Cirq, Q#, etc) to ion trap gate-level direct access"

Authors

Alexander Allen <arallen4>

Keith Mellendorf <kmellen>

Website Link: https://github.ncsu.edu/kmellen/ion_trap_compiler

Problem Statement

- We propose developing a new backend for Qiskit that uses the ARTIQ framework to control the ion trap quantum computer.

Ion Trap Computers

- Ion trap computers uses trapped ionized atoms as qubits.
- Sets, changes, and measures state via lasers.
- Each qubit can be coupled with the others in a chain
    - Hardware limitations allow up to 11 coupled pairs (IonQ)
    - 79 total single-qubits
    - "Our native entangling gate is created by simultaneously 'plucking' two ions in on our chain with precise pulses of laser light. This creates vibrations in the ion chain which are dependent on the qubit state. Since the amount that qubits interact depends on their distance from each other, these different vibrations result in an interaction between the two qubits that depends on the qubit state. At the end of the operation, the vibrations are gone, and the ions are left entangled." (IonQ)

ARTIQ

- ARTIQ is a framework for controling quantum devices.
- This is the interface that we will call from Qiskit to set up the physical ion-trap quantum computing device
- ARTIQ is a Python package, which should allow for easy interfacing with other Python based languages or frameworks

Qiskit

- Qiskit is a quantum programing language developed by IBM
- It is written in the high-level format of Python before being compiled to to qasm
- The qasm is then sent to the backend, traditionally either Aer, a simulator, or IBM's Quantum Computer IBMQ.
- Here is an example provider that allows use of AQT ion-trap quantum devices
    - https://github.com/qiskit-community/qiskit-aqt-provider

    - This system uses an AQT's API system, which is not something we have for Duke's, but it does provide good example of a minimum implementation

Development Process

- We would implement a provider in Qiskit that creates or calls the artiq framework in order to program the physical device
- Different gates would have to be implemented in ARTIQ
- Data must be collected from the physical machine and displayed in a clear mannor
- Ideally we would have access to some example ARTIQ code from Duke to allow us to implement gates
- We may not be able to implement all of the gates that Qiskit, or Duke's ion-trap quantum computer supports

Timeline

- 10/1 Receive the original project
- 10/8 Define first understanding of the problem statement
- 10/9 Meet with Duke
- 10/15 Establish detailed scope of project
- 10/29 Proof of concept
- 11/12 Miminum Viable Product
- 11/19 All requirements met
- 11/27 Submit compiler and documentation

References

- https://ionq.com/technology
- http://dukespace.lib.duke.edu/dspace/bitstream/handle/10161/10461/Ahsan_duke_0066D_13028.pdf?sequence=1
- https://m-labs.hk/
- https://github.com/qiskit-community/qiskit-aqt-provider
- https://qiskit.org