Qiskit Advocate Mentorship Program Spring 2022

Transpiler Hackathon

Team (alphabetically by last name)

- **Vishnu Ajith**
- **Harshit Gupta**
- **Pulkit Sinha**
- **Oskar Słowik**

Mentor: Jack Woehr

IBM Quantum / © 2022 IBM Corporation

What is a Transpiler

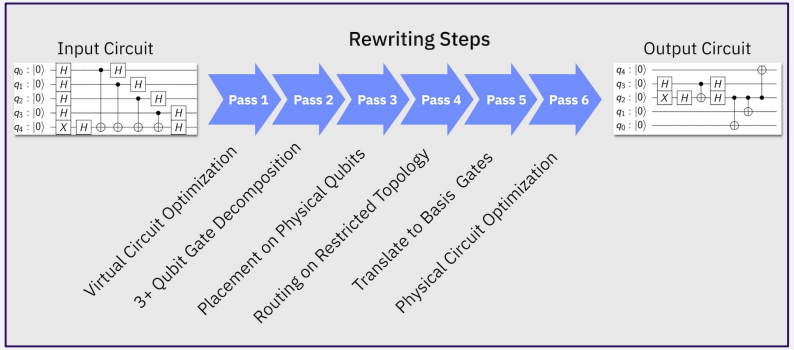
– Quantum algorithms and applications are usually written as **abstract, device agnostic** quantum circuits that may contain any unitary operations.

– However, real quantum devices can only execute a limited set of **hardware-specific, physically calibrated** quantum gates within a given routing layout.

– We need to rewrite an abstract quantum circuit into a functionally equivalent one that matches the constraints and characteristics of a specific target quantum device.

– This essential process is known as transpilation. And the tool responsible for it is called the **transpiler**.

– See [Qiskit/qiskit-terra/qiskit/transpiler]



Functions of a Transpiler

The transpiler is built as a collection of single purpose passes and a PassManager which is responsible for collecting those passes, and coordinating their execution in order to achieve two main goals:

– **Compatibility**: transform a given quantum circuit into one which is executable on a specific device, preserving measurement outcomes.

– **Optimization**: find an implementation which takes maximum advantage of device resources, while minimizing influence of decoherence and errors.

Compatibility

- Expand high level instructions.
- Device's native gate set.
- Layout virtual to physical qubits.
- Device architecture (superconducting – ion trap – ...)
- Device-specific constraints (no mid-circuit measurements – resets – ...)

Optimization

- Remove gate-inverse pairs
- Compact chains of single-qubit gates.
- Commutation analysis and adjacent gate cancellation
- Noise-aware layout selection
- Optimal synthesis of two-qubit blocks

Goals of the Transpiler Hackathon

Qiskit

- As of 2022-01-29 there were 127 open Transpiler issues in the Qiskit Terra repository.
- Participants will close Transpiler issues for the duration of the QAMP Spring 2022 event.
- If there is time and energy left over, we will look at future directions for the Transpiler.

Issues we are looking at

Qiskit

Issues

7296 .. PR 7875 ready for review

7386 .. PR 7542 was merged

7387 .. stalled

7705 .. Oskar to continue on this

7181 .. Oskar and Harshit take this one up after presentation

7113 .. Deal with this after 7181

Issue 7296

double swap is not being optimized out when followed by a gate

- Main idea is to do logical optimizations of the swap gate in the transpilation process, agnostic of a backend.
- Current approach uses a TransformationPass of the qiskit transpiler to identify and cancel out such gates.
- This pass identifies back to back, self inverting gates symmetric in their qubit arguments and cancels them out. (eg. Swap, CZ, etc.)
- More details - PR #7875

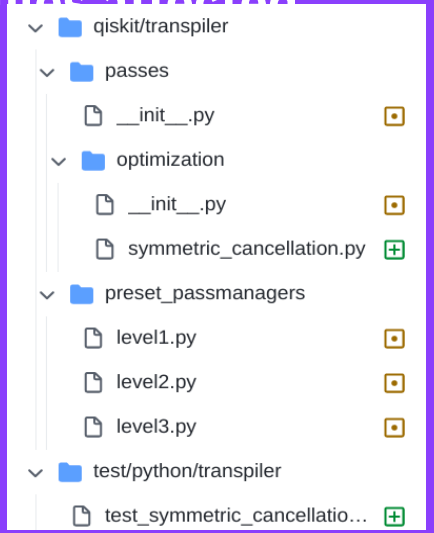
- Status

PR *in review*

- Mentees working

Harshit Gupta

- Files affected



Issue 7386

better convergence criteria in preset passmanagers

- This issue required a change in how the transpiler decided to stop its optimization loop
- Previously if the depth of a quantum circuit was constant after an optimization pass, then the transpiler stopped its operations
- If not, the optimization loop continued
- After this PR the check was expanded to also include the size of the quantum circuit, allowing a greater level optimization.
- More details - PR #7542

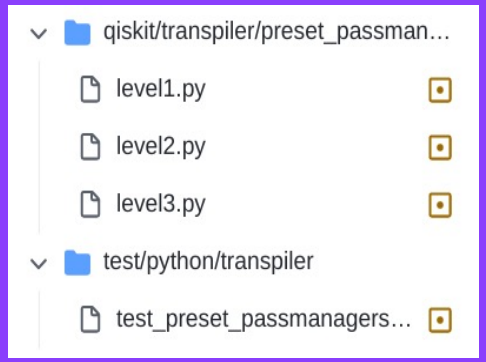
- Status

PR *merged*

- Mentees working

Harshit Gupta

- Files affected



Issue 7705

Add adaptive limits for VF2Layout in preset passmanagers

- The vf2layout pass searches for layout by reducing it to the subgraph isomorphism problem.
- Currently, the allowed runtime limits are hardcoded for each optimization level.
- The goal is to obtain adaptive limits that would scale with the size of the problem.
- We are experimenting with benchmarks using heavy square and heavy hex graphs as targets and various subgraphs, including random ones.
- Based on that we intend to identify the scaling properties that would allow to appropriately set the limits.

- Status

In progress

- Mentees working

Harshit Gupta

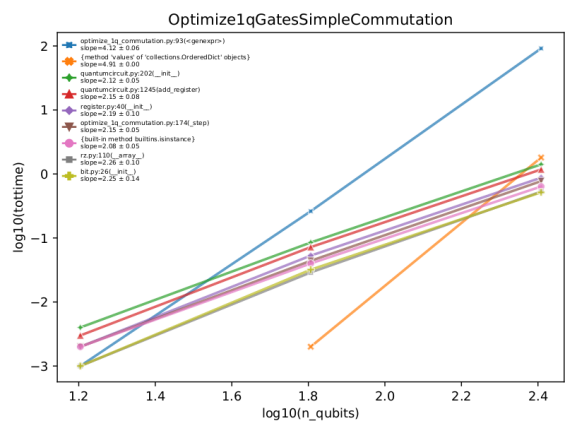
Oskar Słowik

- Files affected

Issue 7181

Optimize1qGatesSimpleCommutation scales superlinearly

- The Opimize1qGatesSimpleCommutation pass scales as the ~4th power in the number of qubits for a random circuit with equal width and depth.
- The goal is to improve the poor scaling of the pass' runtime.



- Status

Scheduled

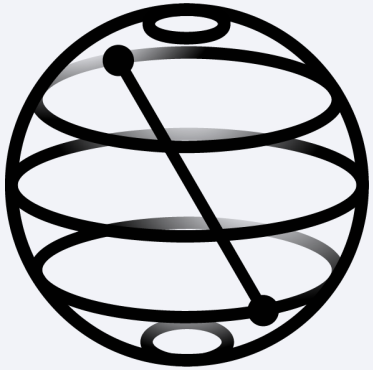
- Mentees working

Harshit Gupta (?)

Oskar Słowik (?)

- Files affected

Thank You!



Our Team's Web Page: https://qamp-spring-2022-transpiler-hackathon.github.io