Check Point 2: QAMP Fall 2022

Adding a "no-inline" option to Qiskit transpiler #16

Mentor: Adrien Suau
Mentees: Juon Kim, Pranshi Saxena

Current quantum compilers systematically inline the transpiled quantum circuit. A no-inline transpiler would be able to reuse previously compiled subroutines so that we can transpile the quantum circuit more efficiently (compile once, re-use multiple times) and adapt the transpiler to classical feedback.
To do so, we analyzed whether the transpiler passes are compatible with a no-inline transpiler or not.

Transpiler pass	Compatible	Why?
DenseLayout	X	With a non-flattened circuit, subroutines will be mixed so that the qubit connection will be changed
VF2Layout	X	With a non-flattened circuit, the solution for subgraph isomorphism problem needs swap mapping
CheckMap	O	Checking qubit interactions is not related with the flattened circuit
Unroll3qOrMore	X	With a non-flattened circuit, recursive operations will be changed with subroutines

Then we researched how to divide the entire code into subroutines and mapped it to a coupling map.

Another step was to implement classical control flow for optimization_level = 1.
It turns out that Qiskit developers already implemented classical control flow recently (see https://github.com/Qiskit/qiskit-terra/issues/8630), so we based our work on this implementation as the requirements to have a control flow-aware transpiler are very similar to the requirements for a "no-inline" transpiler.
We have forked the branch in Github and are currently working to implement the Qiskit code for setting up transpiler passes to enable control flow. We then analyze the algorithm and approach, and compare the results.

Table below is an example of compatible transpiler passes and why it adds control flow handling and how it is changed.

Transpiler pass	Definition	Why?	How?
DenseLayout	Choose a layout by finding the most connected subset of qubits	DenseLayout pass is heuristic, so control flow is not exactly correct	Count the number of measurement and cx gates per qubit
VF2Layout	Choose a layout of a circuit onto a coupling graph	To solve the subgraph isomorphism problem	Score the control flow operation which is transparent in the pass
CheckMap	Check if a DAG circuit is mapped to a coupling map	More complicated to check if there is no control flow blocks	Control flow blocks are not in the outer circuits but inner bits that reference the root register
Unroll3qOrMore	Expands 3q+ gates repeatedly until the circuit contains 2q or 1q qubits	To make repeated operations recursive	Modify control flow operations in a depth-first pass before running the circuit

Based on this, we have tried to apply no-inline with compatible passes and complement more codes with incompatible passes.

Along with setting up the development environment and coding, we have also analyzed the research paper “Tackling the Qubit Mapping For NISQ - Era Quantum Devices”.
The technique presented in this paper, called SABRE, can be used with NISQ devices that have any number of connections between qubits. SABRE outperforms the most well-known algorithm with an exponential speedup and results that are comparable to or better on a variety of benchmarks by optimizing each search attempt, globally optimizing the initial mapping using a novel reverse traversal technique, and introducing the decay effect to enable the trade-off between the depth and the number of gates of the entire algorithm.
As provided in the research paper, the need for SABRE technique is required to adapt the provided quantum circuit to the targeted hardware topology. The two-qubit gates in the algorithm must be enabled by dynamically remapping logical qubits to physical qubits in the compiler, which adds extra processes and consequently lowers algorithm fidelity. High complexity, poor initial mapping quality, and limited flexibility and controllability plague the prior attempts to find such remappings.
The study of such mapping algorithm is important as we expect these algorithms to be a major issue in making the transpiler “no-inline”.