Implementation of Quantum Verification of Matrix Products

Elton Pinto [1] Jeffrey Young [1] Thomas Conte [1]

[1]Georgia Institute of Technology

Abstract

This study implements the Quantum Verification of Matrix Products (QVMP) [1] using the Qiskit framework. We evaluate this implementation using gate count, qubit count, circuit depth, and transpilation time metrics. Through this study, we hope to expand the existing suite of experimental quantum hardware evaluation, provide feedback to quantum framework authors, and suggest improvements to existing hardware.

Motivation

Quantum computing has experienced a recent surge in popularity given the advancements in NISQ machines. Several studies have experimentally evaluated quantum algorithms on quantum hardware. Similarly, extensive work has been carried out on developing quantum programming frameworks. However, no substantial work has been done to evaluate the efficacy of these programming frameworks in developing quantum-based systems.

Our study tries to fill these gaps by evaluating the Qiskit framework. We selected the QVMP algorithm because it uses a combination of the Grover search and amplitude amplification algorithms as subroutines, thereby enabling us to analyze the oracle generation capabilities of Qiskit.

Background

Given three square matrices A, B, and C of size n, the verification of matrix products (VMP) decides if $AB = C$.

In our study we implemented the following recursive Grover search based QVMP algorithm.

Input: $n \times n$ matrices A, B, C
Output: 1 if $AB = C$ and 0 otherwise

1. Partition B and C into sub-matrices of size $n \times \sqrt{n}$
2. Perform amplitude amplification for $n^{\frac{1}{4}}$ iterations using this subroutine:
   1. Pick a random vector x of size $\sqrt{n}$
   2. Classically compute $y = B_i x$ and $z = C_i x$
   3. Using Grover search with $\sqrt{n}$ iterations, find a row of index j such that $(Ay \neq z)_j$

Grover's algorithm is a popular quantum search algorithm. Given an input space of N elements and an oracle U_f, Grover search can find M solution indices in $O(\sqrt{\frac{N}{M}})$ time. For simplicity, we assume that N is a power of 2.

The algorithm works by repeatedly applying a Grover operator G to the initial state $H^{\otimes n} |0\rangle^{\otimes n}$. Amplitude amplification is a generalization of the Grover operator G.

Key Results

- **Too many qubits**: Using a naive encoding of matrices results in large qubit counts. It rises to more than 1000 for even small matrices of order 32 (table 2).
- **Transpilation times rise exponentially**: Quantum circuits need to be transpiled using the gate set of the targetted backend. For the Aer backend, we observed that it reaches over a minute for circuits involving a couple hundred qubits (table 1).
- **Lack of oracle authoring utilities**: Grover oracles are typically classical functions that need to be encoded into a quantum circuit. Qiskit offers no automation in this regard. Programmers have to hand-write the circuits, which is difficult to get right.
- **Difficult to debug and verify outputs**: Qiskit does not have facilities to mechanically prove properties of quantum circuits. We believe that formal verification is an important feature given the non-deterministic behavior of quantum programs.

Indexer circuit

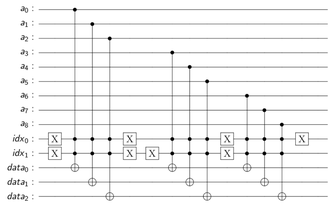

Figure 1. Indexer Circuit for a 3×3 matrix

The role of the indexer is to output $A[i]$, where A is an $m \times n$ matrix and i is an integer index.

We encode the matrix A using $size(A)$ qubits, where $size(A)$ is the total number of elements in A. The integer index is encoded using $log_2(m)$ qubits in the binary format.

Inner Product circuit

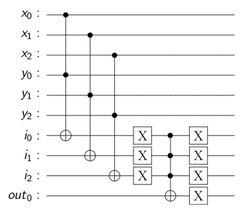

Figure 2. Inner Product Circuit for 3×1 vectors

The inner product circuit 2, as the name suggests, computes the inner product between two vectors encoded as qubits.

Our implementation only handles binary numbers (where multiplication is defined as logical AND and addition as logical OR), but can be easily extended to support other data types.

QVMP Marking Oracle

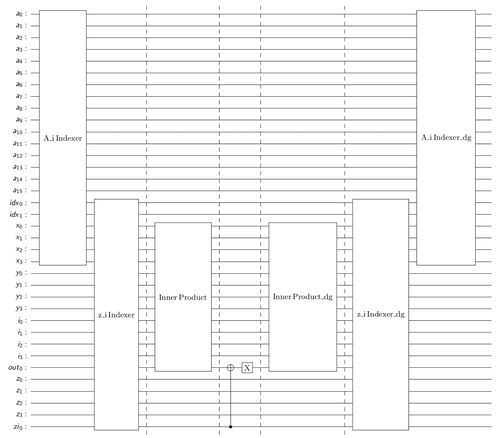

Figure 3. QVMP Marking Oracle for a 4×4 matrix

Metrics

Order of matrix	Time (in seconds)
2	0.24
4	0.81
8	2.38
16	27.48
24	66.20

Table 1. VMP Circuit Transpilation Stats

Order of matrix	Total Gates	Circuit Depth	Qubit Count
2	269	94	15
4	1131	415	36
8	3164	1255	101
16	30740	13445	326
24	63193	28581	679
32	130960	60564	1159

Table 2. VMP Circuit Stats

Future Work

- **Optimize for space complexity**: Our current implementation uses a naive encoding of matrices which can be quite wasteful since most of the qubits are not used for the rest of the computation within the oracle. We can use quantum multiplexers like QRAM [3] to achieve a quadratic improvement in space complexity.
- **Investigate automatic synthesis of Grover oracles**: Implementing quantum oracles by hand is challenging. Matters become more complicated when entanglement gets involved. Further, hand-tuning for space-complexity while maintaining correct can get tricky. We can approach this problem by developing compilers that automate the synthesis of reversible oracles from high-level classical descriptions. To encorporate verification, we either develop our compiler within existing formal verification frameworks like VOQC [5] or target standard formats like OpenQASM which can then be consumed by formally-verified optimizers. Quipper [4] and REVS [2] have made strides in this direction

References

[1] Andris Ambainis, Harry Buhrman, Peter Høyer, Marek Karpinski, and P Kurur. Quantum matrix verification.

[2] Matthew Amy, Martin Roetteler, and Krysta M. Svore. Verified compilation of space-efficient reversible circuits. In *Proceedings of the 28th International Conference on Computer Aided Verification (CAV 2017)*, pages 3–12. Springer, July 2017.

[3] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Physical Review Letters*, 100(16), Apr 2008.

[4] Alexander S. Green, Peter LeFanu Lumsdaine, Neil J. Ross, Peter Selinger, and Benoît Valiron. Quipper. *ACM SIGPLAN Notices*, 48(6):333–342, Jun 2013.

[5] Kesha Hietala, Robert Rand, Shih-Han Hung, Xiaodi Wu, and Michael Hicks. A verified optimizer for quantum circuits. *Proceedings of the ACM on Programming Languages*, 5(POPL):1–29, Jan 2021.