An Algorithm for the T-count

David Gosset[2,3], Vadym Kliuchnikov[1,3], Michele Mosca[1,2,3], Vincent Russo[1,3] (arXiv:1308.4134)

[1]David R. Cheriton School of Computer Science, [2]Department of Combinatorics & Optimization, and [3]Institute for Quantum Computing
University of Waterloo, Waterloo, Ontario N2L 3G1, Canada

Introduction

Motivation: T gates are expensive to implement fault tolerantly, so unitaries which minimize the number of T gates also minimize the cost of implementation.

In this work, we are interested in:

- Unitaries *without* ancillae
- *Exact* implementation over Clifford+T gate set

Notation

Pauli matrices:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \; Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \; Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Clifford+T gate set:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \; T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}, \; CNOT = \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{pmatrix}$$

Pauli operators: The set of n-qubit Pauli operators

$$\mathcal{P}_n = \{Q_1 \otimes Q_2 \otimes \cdots \otimes Q_n : Q_i \in \{\mathbb{I}, X, Y, Z\}\},$$

T-count: The minimum $m \in \mathbb{N}$ for unitary U for which

$$\mathcal{T}(U) = e^{i\phi}U = C_m T_{(q_m)} C_{m-1} T_{(q_{m-1})} \cdots T_{(q_1)} C_0$$

where $\phi \in [0, 2\pi)$, C_i are in the n-qubit Clifford group, $q_j \in \{1, \cdots, n\}$, and $T_{(r)}$ indicates the T gate acting on the rth qubit.

A Decomposition for Clifford + T Unitaries $U \in \mathcal{J}_n$

Define

$$R(P) = \frac{1}{2}\left(1 + e^{i\frac{\pi}{4}}\right)\mathbb{I} + \frac{1}{2}\left(1 - e^{i\frac{\pi}{4}}\right)P, \quad P \in \mathcal{P}_n.$$

Proposition 1. For any $U \in \mathcal{J}_n$ there exists a phase $\phi \in [0, 2\pi)$, a Clifford $C_0 \in \mathcal{C}_n$ and Paulis $P_i \in \mathcal{P}_n \setminus \{\mathbb{I}\}$ for $i \in [\mathcal{T}(U)]$ such that

$$U = e^{i\phi}\left(\prod_{i=\mathcal{T}(U)}^{1} R(P_i)\right) C_0.$$

Acknowledgments

This research was supported by Canada's NSERC, MITACS, CIFAR, and CFI.

UNIVERSITY OF WATERLOO | IQC Institute for Quantum Computing

Main Problem

COUNT-T: Given $U \in \mathcal{J}_n$ and $m \in \mathbb{N}$, decide if $\mathcal{T}(U) \le m$.

The Group $\mathcal{J}_n$ Generated by Clifford and T gates

Clifford+T group: The group generated by the n-qubit Clifford+T gate set

$$\mathcal{J}_n = \langle H_{(i)}, T_{(i)}, CNOT_{(i,j)} : i, j \in [n] \rangle$$

Giles and Selinger's characterization of the Clifford+T group [4]: An n-qubit unitary U is an element of the Clifford+T group if and only if its matrix elements are in the ring

$$\mathbb{Z}\left[i, \frac{1}{\sqrt{2}}\right] = \left\{ \frac{a + bi + c\sqrt{2} + di\sqrt{2}}{\sqrt{2}^k} : a, b, c, d \in \mathbb{Z}, k \in \mathbb{N} \right\}$$

and $\det U = e^{i\frac{\pi}{8}Nr}$ for some $r \in [8]$ and $N = 2^n$.

An Algorithm for the T-count (Special Case)

We focus on a special case of COUNT-T where m is even and where $U \in \mathcal{J}_n$ can be written as in Proposition 1 with $\phi = 0$ and $C_0 = \mathbb{I}$. This special case only illustrates the main idea of our algorithm (see our paper [1] for more details).

Simplified version of our algorithm for COUNT-T (tailored to a special case)

❶ **Precompute sorted databases of unitaries with T-count at most $\frac{m}{2}$.** Generate a database $\mathcal{D}$ of all unitaries of the form

$$U = \prod_{i=\frac{m}{2}}^{1} R(P_i)$$

and then sort the database according to the lexicographic order on matrices. If the unitary U is ever inserted in the database then stop and output $\mathcal{T}(U) \le m$ (in this case we have $\mathcal{T}(U) \le \frac{m}{2}$), otherwise proceed to step 2. Note that $\mathcal{D}$ contains at most $2^{nm} = N^m$ unitaries

❷ **Meet-in-the-middle search.** For each $W \in \mathcal{D}$, let $V_W = W^\dagger U$ and use binary search (using the fact that the database is sorted) to determine if $V_W \in \mathcal{D}$. If $V_W \in \mathcal{D}$ for some $W \in \mathcal{D}$ then output $\mathcal{T}(U) \le m$; otherwise output $\mathcal{T}(U) > m$.

Our algorithm solves COUNT-T using $\mathcal{O}(N^m \mathrm{poly}(m, N))$ time and space requirements.

From the Special Case to the General Algorithm

There are *three* main differences between the special case of COUNT-T and the general case. The most non-trivial difference is that C_0 can be any Clifford $\mathcal{C}_n$. In this case $U = WVC_0$ where W and V are

$$V = \prod_{i=\frac{m}{2}}^{1} R(P_i), \quad W = \prod_{i=\frac{m}{2}}^{1} R(S_i)$$

for some $S_i, P_i \in \mathcal{P}_n$.

Naive solution: In Step 2, search for $V_W = W^\dagger U C_0^\dagger$ instead of $V_W = W^\dagger U$ for each $C_0 \in \mathcal{C}_n$.

Disadvantage: This introduces an overhead of $\Omega(2^{n^2})$.

Our Solution: Avoid overhead by using a labeling scheme of unitaries. Different unitaries that differ by an overall phase and/or right-multiplication by a Clifford have the same label.

T-count of Toffoli and Fredkin is 7

An application of our result is that the following circuits, which do not make use of ancilla qubits, are T-optimal.

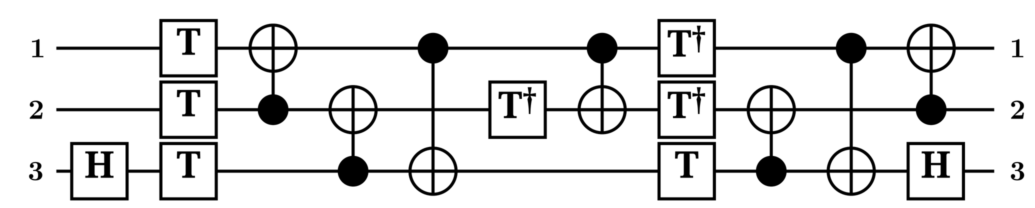

Figure 1: Toffoli cannot be implemented using 3 qubits with less than 7 T gates.

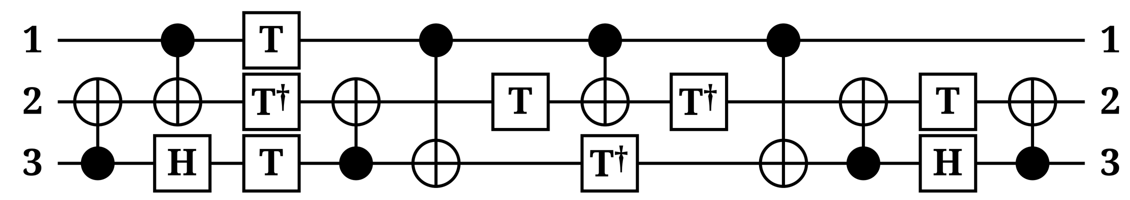

Figure 2: Fredkin cannot be implemented using 3 qubits with less than 7 T gates.

Open Problems

- Does there exist a polynomial time (as a function of N and m) algorithm for calculating the T-count of a given unitary?

Software

We implemented our algorithm in C++.

Two qubits: Generated coset databases $\mathcal{D}_0^2, \cdots, \mathcal{D}_6^2$ (taking 3.96 GB of space).

Three qubits: Generated coset databases $\mathcal{D}_0^3, \cdots, \mathcal{D}_3^3$ (taking 4.60 GB of space).

This enables us to run the two-qubit algorithm with $m = 12$ or the three qubit algorithm with $m = 6$.

References

[1] D. Gosset, V. Kliuchnikov, M. Mosca, V. Russo. arXiv:1308.4134, (2013).

[2] M. Amy, D. Maslov, M. Mosca. arXiv:1303.2042, (2013).

[3] M. Amy, D. Maslov, M. Mosca, M. Roetteler. Computer-Aided Design of Integrated Circuits and Systems, 32:818-830, (2013).

[4] B. Giles, P. Selinger. Physical Review A, 87(3):032332, (2013).

[5] V. Kliuchnikov. arXiv:1306.3200, (2013).

[6] V. Kliuchnikov, D. Maslov, M. Mosca. arXiv:1206.5236, (2012).