References

Quantum Rotations: A Case Study in Static and Dynamic Machine-Code Generation for Quantum Computers

Timothy M. Dee
Electrical & Computer Engineering
Iowa State University
Ames, IA, USA
deetimothy33@gmail.com

November 18, 2016
\[ |s\rangle = \alpha |0\rangle + \beta |1\rangle \] (1)

where \( |\alpha|^2 \) is the probability of measuring the qubit in state \( |0\rangle \), the classical low state, and \( |\beta|^2 \) is the probability of measuring the qubit in state \( |1\rangle \).
evidenced by the Bloch Sphere, phase does not directly effect the outcome of a measurement however it is very important in determining how qubits interact with one another. This
Shor's factorization algorithm. For the first time, we study a larger range of benchmarks and find that another critical issue is the generation of code sequences for quantum rotation operations. Specifically, quantum algorithms require arbitrary rotation angles, while quantum technologies and error correction codes provide only for discrete angles and operators. A sequence of quantum machine instructions must be generated to approximate the arbitrary rotation to the required precision.
Solovay-Kitaev decomposes the input gate by recursively factoring it to a predetermined depth. In the leaves of the recursion tree, each factor is approximated by a short sub-sequence (generally about 20 gates long) selected from a database. These subsequences are then concatenated to form the final approximation of the rotation.
The Solovay-Kitaev Theorem\cite{18} is a result in quantum computing which states that any single qubit gate can be approximated to a precision $\epsilon$ by a sequence of $\Theta(log^c(\frac{1}{\epsilon}))$ gates from a universal set. The Solovay-Kitaev Algorithm,
We introduce a new method for compiling arbitrary rotations dynamically, designed to minimize compilation time.
The new method reduces compilation time by up to five orders of magnitude while increasing code size by one order of magnitude.
\[ T_{rotation} = t_{compile} + l \times t_{gate}, \]  \hspace{1cm} (7)

where \( t_{compile} \) is the compilation time of the gate, \( l \) is the length of the generated sequence and \( t_{gate} \) is the technology-specific gate time. When evaluating compilation strategies
From this simple equation, we see that the choice of technology and the precision to which we are compiling have a sizeable impact on our choice of compilation technique. The
cuses on minimizing compilation time. In this technique, a set of rotations is pre-compiled (using a static compilation technique such as Solovay-Kitaev) and stored as a library. Arbitrary rotations are then rapidly assembled by concatenating rotations from this library. We are essentially trading storage (and a less optimized result) for execution time. The
By adding the corresponding rotations in our library set,

$$\theta \approx \frac{\pi}{2^1} + \frac{\pi}{2^2} + \frac{\pi}{2^6} + \frac{\pi}{2^7} + \frac{\pi}{2^8} + \ldots$$  \hspace{1cm} (6)

we are guaranteed an approximation to within $\frac{\pi}{2^k}$ of the original angle.
END