Simulating the Quantum Fourier Transform with Distributed Computing

F.L. Marquezino[1] **, R.R. Mello Junior** [1] **, R. Portugal** [2] **, G.N. Cunha** [1]

[1]Universidade Católica de Petrópolis
Faculdade de Informática
Rua Barão do Amazonas, 124 - Centro
25.685-070 Petrópolis (RJ)

`franklin.marquezino@gmail.com , rui.rmj@gmail.com , gerson.nunes@inf.ucp.br`

[2]Laboratório Nacional de Computação Científica
Rua Getúlio Vargas, 333 - Quitandinha
25.651-075 Petrópolis (RJ)

`portugal@lncc.br`

Abstract. *In this paper a new approach for quantum computer simulations is presented. The proposal is creating a simulator where the main concern is not simply the results of the algorithm for a given input. Instead, this simulator will imitate, as close as possible, the internal behavior of a real quantum computer. In order to do that, Distributed Computing is necessary.*

1. Introduction

Quantum Computation is a recent area of investigation, where computers are studied under a different perspective, which is the Quantum Mechanics. These studies were originated specially by [Feynman, 1982, Manin, 1980]. When a computer is built according to the principles of Quantum Mechanics, some interesting properties appear, allowing the development of faster algorithms and safer cryptosystems [Marquezino, 2003]. However, the building of this kind of computer is still a technological challenge.

In order to test the algorithms for this new model of computation, some researchers throughout the last years have developed many quantum computer simulators [Raedt and Michielsen, 2004]. They are never efficient [Feynman, 1982], but necessary to the understanding of Quantum Computation. The proposal of this work is to create a quantum computer simulator emphasizing the internal behavior of a real quantum computer, and not the results *per si* of the algorithm.

When creating a simulator of quantum physical systems, a natural choice certainly is Distributed Computing [Rosé et al., 2004]. It is important to stress that in the present work these techniques will not be used to obtain performance. Instead, the proposed simulator will be distributed only with the goal of imitating, as close as possible, the behavior of a quantum computer. It seems that a quantum computer simulator for grids would be efficient only if an exponential number of computers were provided [Feynman, 1982, Rosé et al., 2004].

The algorithm chosen for simulation is the Quantum Fourier Transform (QFT). It is an important algorithm, developed by Peter Shor [Shor, 1994], with many interesting applications, such as the factorization of large integers and the solution of the discrete logarithm problem. While the best classical[1] algorithm to perform the Discrete Fourier Transform, known as Fast Fourier Transform, requires $\theta(n2^n)$ operations, the quantum algorithm by Peter Shor needs only $\theta(n^2)$ operations [Nielsen and Chuang, 2000].

All the time expended by the simulator in calculations without physical meaning shall be eliminated from the total simulation time. It is expected that the remaining time reveals the polynomial complexity of the algorithm, – against the exponential complexity of the whole simulation – when it is run in a real quantum computer. The statistics expected to be generated by the simulator may reveal important aspects of Quantum Computation, contributing for the researches in this area.

2. Basic Concepts of Quantum Computation

The fundamental concept in Quantum Computation is the quantum bit (also known as *qubit*). While a classical bit represents only one of two different states (0 or 1), a quantum bit may represent both, at the same time ($|0\rangle$ and $|1\rangle$, where $|\cdot\rangle$ is the so-called Dirac's notation, very useful in Quantum Mechanics). The qubit is represented as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1}$$

with $\alpha, \beta \in \mathbb{C}$, and $\|\alpha\|^2 + \|\beta\|^2 = 1$.

Although one qubit may store a huge amount of information, it is not completely accessible. When someone measures it, the state of the qubit collapses into a classical bit $|0\rangle$ or $|1\rangle$, with probabilities $\|\alpha\|^2$ and $\|\beta\|^2$, respectively.

When a quantum computer is composed by several *qubits*, the complete state is represented by the tensorial product of each qubit. Hence, the state of a quantum computer composed by two qubits, $|\psi_A\rangle$ and $|\psi_B\rangle$ is $|\psi_A\rangle \otimes |\psi_B\rangle \equiv |\psi_A\rangle|\psi_B\rangle \equiv |\psi_A\psi_B\rangle$. It is important to notice that $|\psi_A\psi_B\rangle$ may occur as any superposition of the kind

$$|\psi_A\psi_B\rangle = \alpha|00\rangle + \beta|01\rangle + \delta|10\rangle + \gamma|11\rangle, \tag{2}$$

with $\alpha, \beta, \delta, \gamma \in \mathbb{C}$, and $\|\alpha\|^2 + \|\beta\|^2 + \|\delta\|^2 + \|\gamma\|^2 = 1$.

There are some states $|\psi\rangle$, for instance, $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, for which no states $|\psi_A\rangle$ and $|\psi_B\rangle$ can be found such that $|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$. These states are called *entangled states*, and they play an important role in Quantum Computation.

A logical operation in Quantum Computation is represented by a unitary operator. If the input of a quantum computer is $|\psi\rangle$, and the logical gate to be executed is represent bu U, then the output is obtained by

$$|\psi_{\text{output}}\rangle = U|\psi\rangle. \tag{3}$$

[1]By *classical* we mean any physical system, or algorithm, or model of computation, which is related only to the *Newtonian* Mechanics, and not with the *Quantum* Mechanics.

It is possible to build a quantum circuit to perform the following operation

$$|a\rangle|b\rangle \xrightarrow{U} |a\rangle|b \oplus f(a)\rangle, \tag{4}$$

using a unitary operator U [Nielsen and Chuang, 2000].

If the input qubit $|a\rangle$, in eq. (4) is in a superposition state, for instance,

$$\frac{1}{\sqrt{2}}\Big(|0\rangle + |1\rangle\Big),$$

we have the quite interesting output

$$\frac{1}{\sqrt{2}}\Big(|0\rangle|b \oplus f(0)\rangle + |1\rangle|b \oplus f(1)\rangle\Big).$$

The important fact is that the quantum computer could find the values of $f(0)$ and $f(1)$ with only one execution of the program (i.e., the operator U), while any classical computer should run it twice. This property is called *quantum parallelism*.

3. The Simulator

In the proposed distributed simulator, each process must have a physical interpretation. This is the main concern before modeling the system. The natural choice is to make each process take care of one particle. In this version of the simulator, for simplicity, only quantum computers built on two-level quantum particles will be considered. In this case, each process will take care of exactly one qubit.

An additional process will be necessary, so that the master node will be able to represent the circuit to be simulated, and to send the appropriate logical gates to the correct qubits. If the processes running on the slave nodes are equivalent to the quantum particles in a quantum computer, then the process running on the master node is equivalent to the physical apparatus used to manipulate the qubits.

The first question that may arise in this model is how to simulate entangled particles. The proposed simulator should isolate the representation of each quantum particle in one process. However, if two or more of these particles are entangled, by definition, their states cannot be represented isolatedly. The solution is to represent each state with redundancy. When a logical gate is sent to a process representing an entangled state, the message must be retransmitted to all others it is entangled with.

All the inefficiency caused by this redundant representation must be counted and eliminated from the total simulation time. By doing this, the physical interpretations of the simulation are not lost.

4. Conclusion

This work has presented a different approach for quantum computer simulations, where Distributed Computing is used in order to obtain a better comprehension of the quantum-mechanical concepts. Here, the concern was not performance but the physical interpretation of the quantum processes involved.

There is a software under development, using the ideas exposed here. The choice was the C++ language in a message passing environment using PC clusters. The platforms used for most part of the development are Red Hat Linux 9 and Sun OS.

The future directions of this work are full of possibilities. It will be important to simulate different quantum algorithms. It will be also interesting to simulate quantum computers in the presence of noise, using quantum error correction. Finally, the simulation of large inputs using LNCC's grid environment and UCP's experimental cluster should reveal important aspects about quantum computation.

Acknowledgments

The authors thank the National Laboratory for Scientific Computing (LNCC, Brazil), and the Catholic University of Petrópolis (UCP, Brazil). Some of us (FLM and RRMJ) also thank the Brazilian Scientific Council (CNPq), for financial support. The Brazilian Centre for Researches in Physics (CBPF), in particular Prof. J.A. Helayël-Neto, should also be acknowledged for providing helpful discussions.

References

Feynman, R. (1982). Simulating Physics with Computers. *Int. Journ. Theor. Phys.*, (21):467.

Manin, Y. (1980). *Computable and Uncomputable (in Russian)*. Sovetskoye Radio, Moscow.

Marquezino, F. (2003). Estudo Introdutório do Protocolo Quântico BB84 para Troca Segura de Chaves. *Centro Brasileiro de Pesquisas Físicas, Série Monografias*, (CBPF-MO-001/03).

Nielsen, M. and Chuang, I. (2000). *Quantum Computation and Quantum Information*. Cambridge University Press, UK.

Raedt, H. D. and Michielsen, K. (2004). Computational Methods for Simulating Quantum Computers. `<http://www.arxiv.org/quant-ph/0406210>`.

Rosé, H., Aßelmeyer-Maluga, T., Kolbe, M., Niehörster, F., and Schramm, A. (2004). The Fraunhofer Quantum Computing Portal: a web-based simulator of quantum computing processes. `<http://www.arxiv.org/quant-ph/0406089>`.

Shor, P. (1994). Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In *Proceedings, 35th Annual Symposium on Foundations of Computer Science*, Los Alamitos, CA. IEEE Press.