Adiabatic Quantum Computing simulations using GPGPU

Sandra Díaz Pier
A00455576@itesm.mx

Salvador Venegas-Andraca
svenegas@itesm.mx

José Luis Gómez-Muñoz
jose.luis.gomez@itesm.mx

Quantum Information Processing Group
Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Estado de México

Quantum Computation aims at providing us with great advances in the solution of some problems for which we know no efficient algorithms using the classical computer models and systems are available [1].

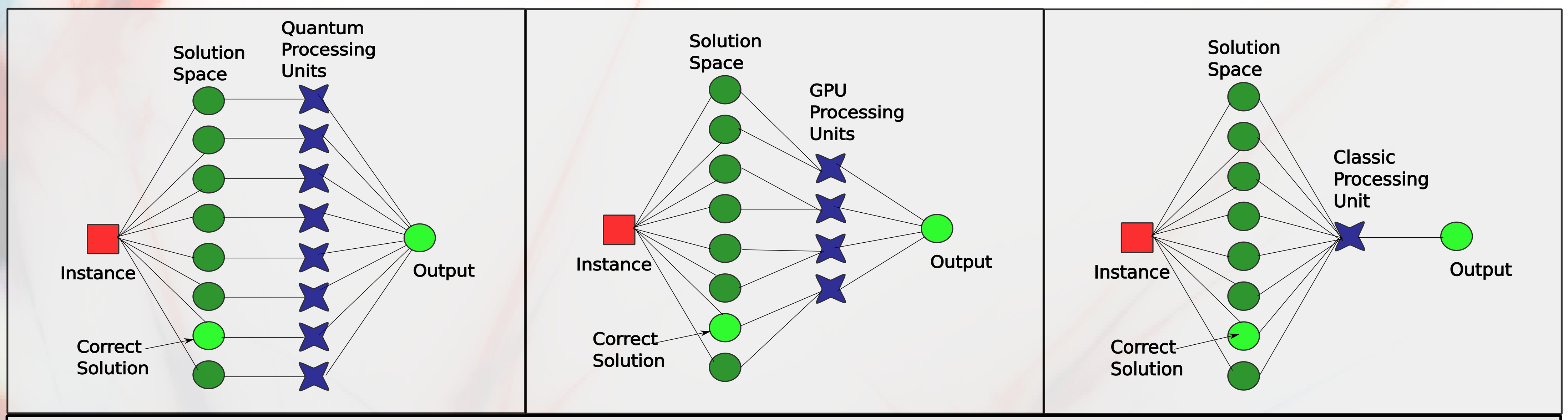

Figure 1: Quantum, multi-core and serial computational approaches

In this project we have created a simulation environment for testing quantum algorithms. Our implementation is based on a high level interface in Mathematica called Quantum, connected to C++ code capable of communicating with a GPU. Our project allows quantum scientists to enhance the performance of quantum computing simulations using a single PC equipped with an NVIDIA CUDA-compatible GPU.

We have tested our platform simulating hard instances of the 3SAT problem [2]. The problem is NP-complete, and becomes particularly difficult to solve when the ratio of number of clauses to number of variables is about 4.2 [3].

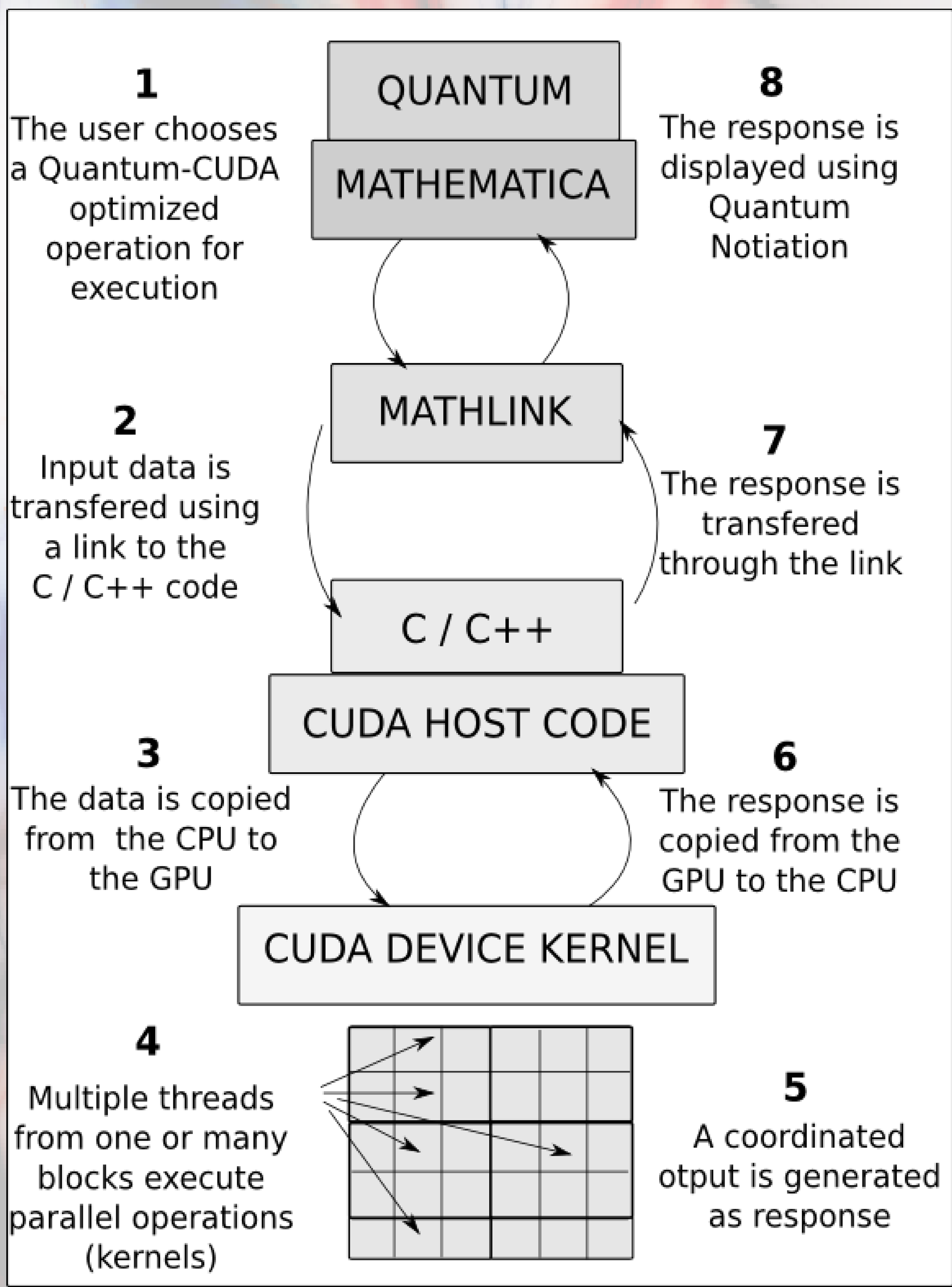

Figure 2: Information Flow among Quantum, Mathematica, Mathlink, C/C++, CUDA and GPUs.

The adiabatic quantum algorithm we have simulated [4] consists of the design of a time-dependent Hamiltonian which can be separated into three parts: an initial Hamiltonian which encodes the ground state of the system, a driving Hamiltonian, in charge of taking the system from the initial state to the final state and the final Hamiltonian, created from an energy function which will give every possible state an energy level proportional to the number of unsatisfied clauses.

An energy function for this algorithm depends on the instance and is constituted by a sum of smaller energy functions, one for each clause. The ground state of the final Hamiltonian encodes the solution to the problem.

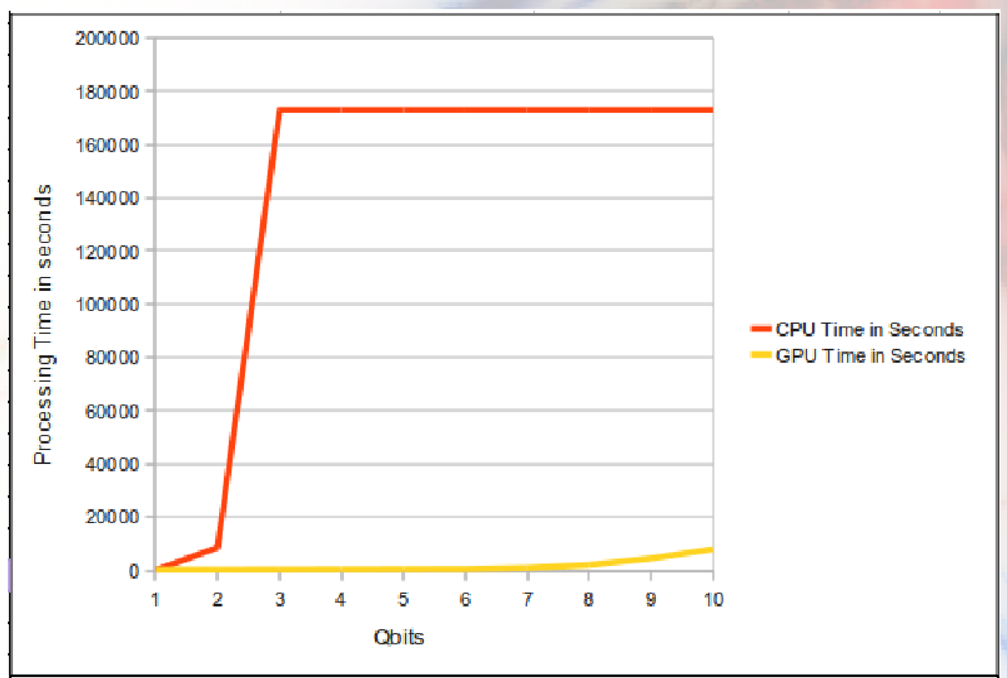

Figure 3: CPU and GPU simulation times for different number of qubits.

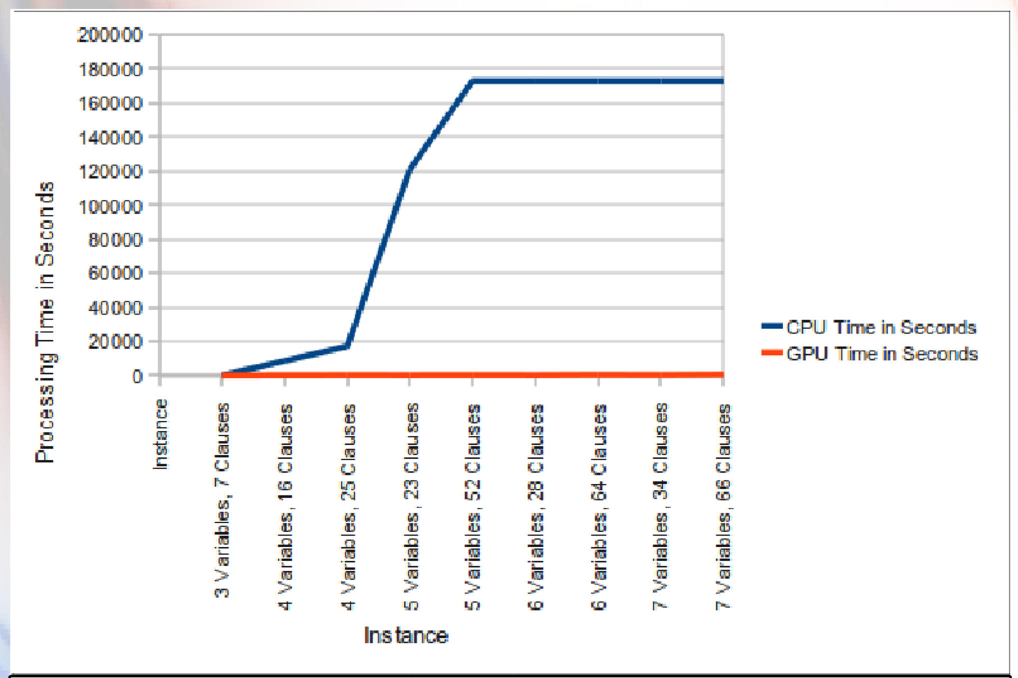

Figure 4: CPU and GPU execution times for different instances of the 3SAT problem.

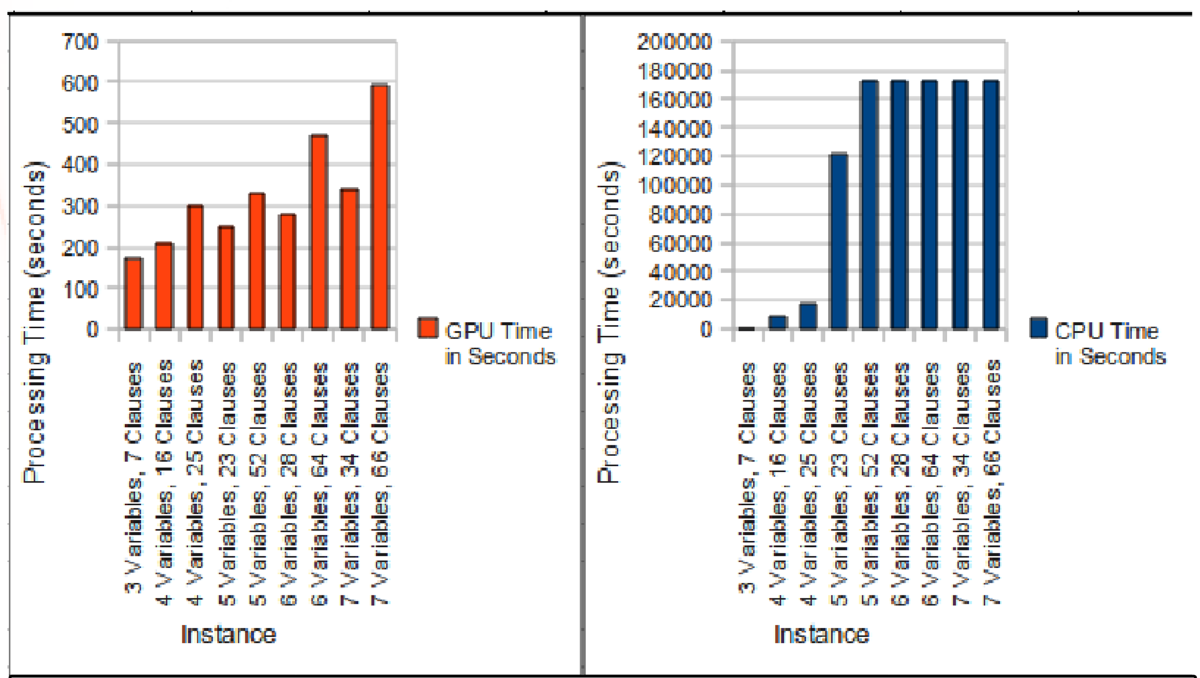

Figure 5: CPU and GPU execution times for different instances of the 3SAT problem.

Our tests were run using a PC with Intel Core 2 Duo processor @ 2.66GHz, 8GB of RAM memory running with Windows Vista and an NVIDIA Geforce GTX 8800 video card of 512MB of video memory and 128 parallel cores. The simulation environment currently runs on Mathematica 7.

Based on our results, we observe that the number of qubits simulated using our GPU tools easily double the ones simulated on a CPU using our setup. These results are possible due to the combination of two characteristics in our simulation: firstly, we aid the simulation tasks with the power of multi-core GPU processing with kernels designed to take advantage of the special memory, thread management and synchronization capabilities of NVIDIA cards. Secondly, we simulate quantum parallelism directly with classical multicore parallelism, which allows us to exploit the GPU occupancy factor to the maximum on every run.

References:
[1] M. Nielsen and L.-I.Chuang. Quantum Computation and Quantum Information. Cambridge University Press, UK, 2000.
[2] A. Perdomo, S. Venegas-Andraca and A. Aspuru-Guzik. A study of heuristic guesses for adiabatic quantum computation. Quantum Information Processing. 10(1):33-52, February 2011.
[3] D. Achlioptas, A. Naor and Y. Peres. Rigorous location at phase transition in hard optimization problems. Nature, 435:759-764, 2005.
[4] E. Farhi, Et Al. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. Science, 292(5516):472-475, 2001