

## SIMULAÇÃO QUÂNTICA DISTRIBUÍDA DE MPPS VIA GPUS

Anderson Braga de Avila<sup>1</sup>; Murilo Figueiredo Schumalfuss<sup>1</sup>;  
Adriano Maron<sup>2</sup>; Renata Reiser<sup>1</sup>; Mauricio Pilla<sup>1</sup>

<sup>1</sup> Universidade Federal de Pelotas – {abdavila, mfschumalfuss, reiser, pilla}@inf.ufpel.edu.br

<sup>2</sup> Pittsburgh – akk48@pitt.edu

### 1. INTRODUÇÃO

A Computação Quântica (CQ) introduz um novo paradigma na ciência da computação, afirmando que os algoritmos quânticos podem desenvolver tarefas mais complexas se comparadas com paradigma de programação atual, como no processamento da informação e criptografia (GROVER, 1996; SHOR 1997). Mas enquanto os computadores quânticos ainda não estão disponíveis, o estudo e desenvolvimento de algoritmos quânticos podem ser feito a partir da descrição matemática ou softwares de simulação. Uma vez que a simulação quântica realizado por computadores clássicos exige muito tempo e elevados recursos computacionais (processos e/ou memória) a pesquisa sobre arquiteturas paralelas pode fornecer possíveis melhorias de desempenho para novos algoritmos quânticos.

Neste contexto, este trabalho contribui com o desenvolvimento o Projeto D-GM, disponibilizando um ambiente de modelagem gráfica e um ambiente de execução e gerenciamento de aplicações, denominados VPE-qGM (MARON, 2013) e VirD-GM (AVILA, 2014).

O paradigma de programação GPGPU (*General Purpose Computing on Graphics Processing Units*) tornou-se recentemente uma das abordagens mais interessantes para HPC (*High Performance Computing*) devido ao seu bom equilíbrio entre custo e benefício. O trabalho em HENKEL 2010 explora GPUs para simulação quântica. Neste contexto, tem-se o espaço de memória disponível na GPU como principal restrição, considerando que o tempo de simulação se apresenta bem mais reduzido. A inovação na área de pesquisa de computação/simulação quântica é integrar num mesmo ambiente essas duas abordagens: simulação distribuída e o uso de GPU. Nosso projeto visa consolidar o ambiente D-GM como suporte para esse cálculo e processamento híbrido da simulação quântica.

A principal contribuição deste trabalho consiste no aumento das capacidades de simulação do ambiente VirD-GM pelo estudo e implementação de abstrações presentes no modelo qGM para interpretação de transformações quânticas a partir de Processos Mistos Parciais (MPPs) e a implementação distribuída/paralela dessas abstrações para correspondente execução sobre clusters de GPUs. Estas contribuições simbolizam os esforços iniciais para explorar a simulação de algoritmos quânticos usando arquiteturas híbridas.

### 2. MATERIAL E MÉTODOS

Com a modelagem e implementação dos MPPs no VirD-GM, o programador tem maior controle do uso da memória e do gerenciamento dos cálculos (granulosidade das computações) considerando os recursos disponíveis. A integração deste gerenciamento com as interfaces do VPE-qGM incrementam as simulações no ambiente D-GM.

Na Figura 1, ilustram-se exemplos de layouts de computação obtidos pela distribuição mista, considerando a Transformação Quântica (TQ) Hadamard  $H^{\otimes 2}$ . Neste contexto, pela aplicação da redução mista no processamento da matriz associada, tem-se a redução da granulosidade de 4,4 para 2,2.

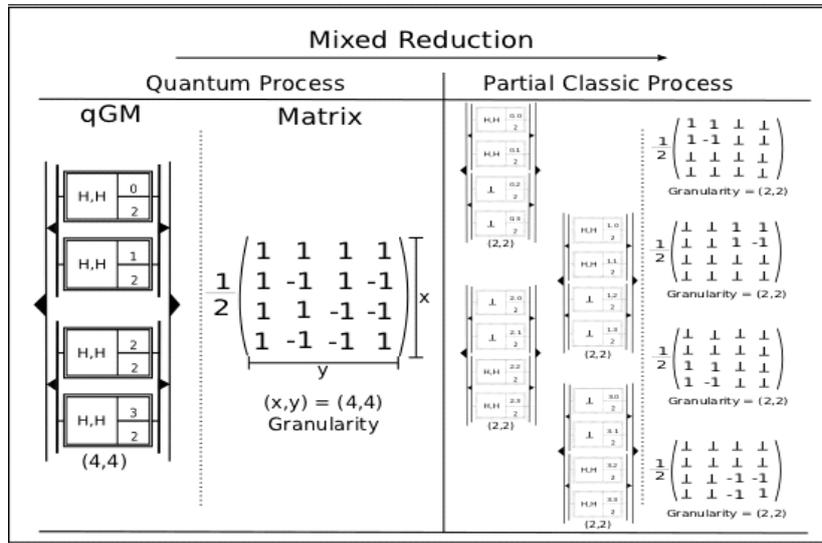


Figura 1. Distribuição mista da computação do operador  $H^{\otimes 2}$ .

As principais alterações que viabilizaram a extensão do ambiente VirD-GM para suporte ao MPPs estão brevemente descritos logo a seguir.

- (i) Reestruturação do arquivo que descreve os processos, modificado para interpretação e reconhecimento de um MPP.
- (ii) Reestruturação do módulo VirD-Exec, responsável pela comunicação, enviando para um VirD-Client, cliente de execução, o MPP e somente a porção da memória que ele necessita.
- (iii) Extensão da biblioteca QGM-Analyzer, responsável pela maior parte da computação, viabilizando invocar o kernel CUDA quando solicitado pelo gerenciamento a execução da computação na GPU.
- (iv) Alteração do kernel CUDA para suportar execuções de MPPs.

### 3. RESULTADOS E DISCUSSÃO

A preocupação nesta parte do resumo deve ser a de expor o que já foi feito até o momento, quais os resultados encontrados e o estado em que se encontra o trabalho. Esta parte serve também para que o autor evidencie o desenvolvimento do trabalho, ou seja, a análise do trabalho de campo e do objeto de estudo propriamente dito.

Se forem usadas tabelas e figuras, seus títulos deverão ser centralizados, com as letras iniciais maiúsculas e fonte Arial, corpo 12.

### 4. CONCLUSÕES

Para validação e análise de desempenho da simulação de algoritmos quânticos a partir de MPPs no VirD-GM, foram considerados estudos de casos

com TQs Hadamard de 18 até 21 qubits. Justifica-se a escolha do operador Hadamard pois é a TQ com densa matriz associada, apresentando maior carga de execução no ambiente de simulação.

A metodologia adotada para a simulação distribuída via GPU considera a execução de 10 simulações de cada instância do operador Hadamard, considerando cada uma das configurações dentro do cluster de GPUs, sendo que o número de MPPs nunca ultrapassa o número de GPUs.

Os testes foram realizados em quatro desktops, sendo dois desktops com processador Intel Core i7, 8 GB de RAM e uma GPU NVIDIA GT640 e dois desktops com processador Intel Core i7, 8 GB de RAM e uma GPU NVIDIA GTX560. Uma máquina adicional, conectada ao cluster por uma rede Fast Ethernet, foi necessário para executar o servidor do VirD-GM. Como componentes de software, tem-se: JCUDA 0.5.0a, NVIDIA CUDA TOOLKIT 5.0, e Ubuntu 12.04 64 bits. Para as simulações com 1 e 2 clientes, os dois desktops utilizados foram os com GPU NVIDIA GT640.

As configurações de MPPs usadas são descritas na forma R-W, onde R é o número de partes em que a memória de leitura foi particionada e W é o número de partes em que a memória de escrita foi particionada. E o número de clientes usados para cada configuração é equivalente a  $R \times W$ , que é o número de MPPs necessários para a simulação completa da TQ naquela configuração, possuindo todas as combinações entre as memórias de leitura e escrita particionadas.

Os tempos de simulação obtidos podem ser vistos na tabela 1. O desvio padrão máximo de 0,81% foi medido para a TQ  $H^{\otimes 19}$  com configuração 1-4.

Tabela 1: Tempos médios de simulação obtidos, medidos em segundos

Op	1 VirD-Client		2 VirD-Client		4 VirD-Client	
	1-1	1-2	2-1	2-2	1-4	4-1
$H^{\otimes 18}$	12,88	7,06	7,07	4,20	4,16	4,25
$H^{\otimes 19}$	47,99	24,66	24,81	13,22	13,14	13,89
$H^{\otimes 20}$	187,80	94,51	94,48	48,49	48,30	48,87
$H^{\otimes 21}$	744,06	373,69	371,17	188,79	187,93	188,41

Na Figura 2, tem-se os speedups obtidos para os diferentes tipos de configurações, com relação a configuração 1-1, que usa um cliente. Salienta-se o ganho de desempenho com o aumento do número de clientes, independente da configuração utilizada, e o speedup se aproxima do ideal com o aumento do número de qubits das TQs. Isto ocorre porque o tempo gasto pela comunicação entre servidor e clientes se torna menos significativo com relação ao tempo gasto em execução no cliente.

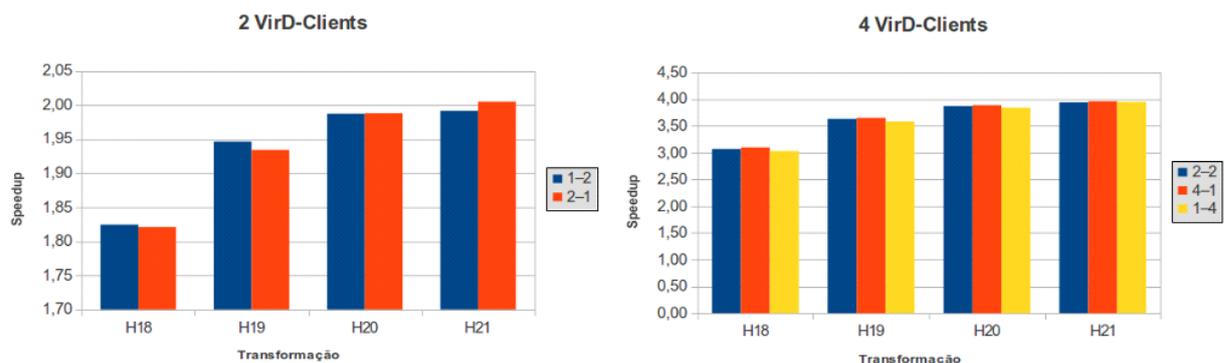


Figura 2. Speedup relativo à configuração 1-1.

O desenvolvimento da extensão para suporte a simulação quântica distribuída a partir de GPUs, utilizando o ambiente VirD-GM, constitui uma solução viável para redução do tempo de simulação.

Os resultados obtidos mostraram que é possível obter ganho de desempenho com o aumento do número de clientes e que transformações com diferentes configurações, mas que utilizam o mesmo número de clientes, possuem um tempo de simulação semelhante, isto implica que o programador possui liberdade para definir a configuração dos MPPs que lhe parece melhor de acordo com os recursos disponíveis, sem se preocupar em ter uma grande perda de desempenho.

Trabalhos futuros em nosso projeto estão descritos nos seguintes tópicos: (i) suporte para portas controladas, projeções e operações de medida na abordagem distribuída; (ii) concepção e implementação do cliente de execução híbrido, em que o cálculo será executado por CPUs e GPUs de forma distribuída.

## 5. REFERÊNCIAS BIBLIOGRÁFICAS

AVILA, A.; MARON, A.; REISER, R.; PILLA, M. GPU-aware distributed quantum simulation. **Proc. of 29th Symposium On Applied Computing**, pages 1–6, 2014.

GROVER, L. A fast quantum mechanical algorithm for database search. In: **Proc. of the Twenty-Eighth Annual ACM Symp. on Theory of Computing**, p. 212–219, 1996.

HENKEL, M. **Quantum computer simulation: New world record on JUGENE**, 2010. Available at <http://www.hpcwire.com/hpcwire/2010-06-28/quantum-computer-simulation-new-world-record-on-jugene.html> (feb. 2013).

MARON, A.; REISER, R.; PILLA, M. High-performance quantum computing simulation for the quantum geometric machine model. In **Cluster, Cloud and Grid Computing (CCGrid) 2013**, pages 474–481, 2013.

SHOR, P. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. **SIAM Journal on Computing**, 1997.