GPU-Aware: Simulação Distribuída de Algorítmos Quânticos via VirD-GM

Anderson B. de Avila[1], **Murilo F. Schmalfuss**[1], **Renata H. S. Reiser**[1], **Mauricio L. Pilla**[1]

[1]Centro de Desenvolvimento Tecnológico
Universidade Federal de Pelotas (UFPEL)
Caixa Postal 15.064 - 91.501-970 - Pelotas - RS - Brasil

{abdavila,mfschumalfuss,reiser,pilla}@inf.ufpel.edu.br

Resumo. *A principal contribuição deste trabalho é a otimização do ambiente VirD-GM, concebida em duas etapas:* (i) *o estudo teórico e implementação das abstrações de Processos Mistos Parcias definidos no modelo qGM, visando a redução no consumo de memória associado à transformações quânticas multidimensionais;* (ii) *e a implementação distribuída/paralela dessas abstrações para correspondente execução sobre clusters de GPUs.*

1. Introdução

A simulação de algoritmos quânticos em computadores clássicos viabiliza o desenvolvimento e teste de algoritmos quânticos, antecipando o conhecimento acerca de seu comportamento quando da execução sobre um *hardware* quântico. A simulação de sistemas quânticos através de computadores clássicos ainda se mostra um desafio de pesquisa em aberto, justificando o estudo de soluções voltadas para a simplificação no processo de modelagem e interpretação de algoritmos quânticos [Nielsen and Chuang 2000]. Mais significante, as otimizações no ganho de desempenho da simulação contribuem para o suporte a sistemas quânticos mais complexos.

A principal contribuição deste trabalho consiste no **aumento das capacidades de simulação do ambiente *VirD-GM* [Avila et al. 2014a] 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 execução em clusters de *GPUs***.

2. Resultados

Com as computações baseadas em *MPPs* [Avila et al. 2014b], tem-se um gerenciamento do limite nas amplitudes acessadas e nos resultados destas computações. Dando controle sobre a granulosidade da computação de cada *MPP*. Para validação e análise de desempenho foram considerados estudos de casos com tranformações quânticas (TQs) *Hadamard* de até 21 *qubits* ($H^{\otimes 18}$, $H^{\otimes 19}$, $H^{\otimes 20}$ e $H^{\otimes 21}$).

Foram realizadas 10 simulações de cada instância do operador *Hadamard*, considerando cada uma das configurações de *MPPs* possíveis dentro do *cluster* com quatro *GPUs*, sendo que o número de *MPPs* nunca ultrapassa o número de *GPUs*.

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. 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.

Na Figura 1, tem-se os speedups obtidos para os diferente tipos de configurações, com relação a configuração $1 - 1$, que usa um *VirD-client*. A análise dos resultados apresentados mostraram ganho de desempenho com o aumento do número de clientes. Tranformações com diferentes configurações, mas que utilizam o mesmo número de clientes, possuem um ganho semelhante, implicando maior flexibilidade ao programador na configuração dos *MPPs* de acordo com os recursos disponíveis, sem gerar grandes perdas de desempenho. 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.

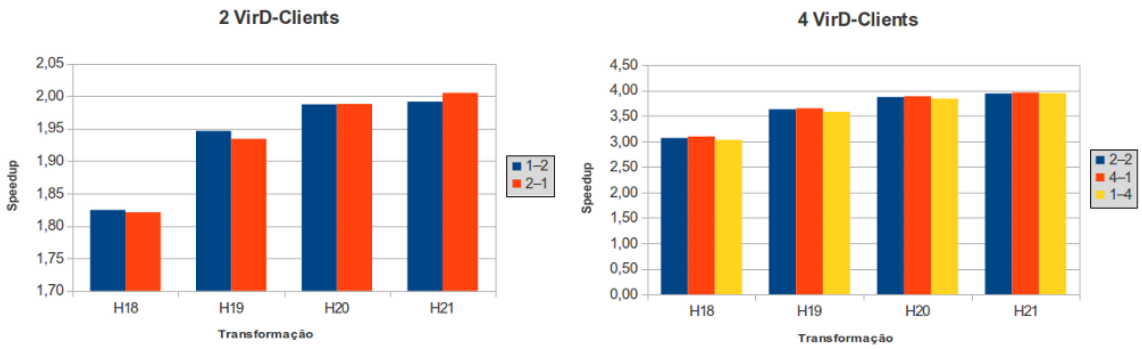

Figura 1. Speedup relativo à simulação com um cliente e configuração $1 - 1$

3. Conclusão

A modelagem de tranformações quânticas através de *MPPs* possibilitou agrupar computações parciais que compartilham um mesmo subconjunto de amplitudes, assim ao atribuir um *MPP* à um nó de processamento evita-se o acesso a todo o vetor de estado de leitura e/ou escrita, reduzindo a complexidade espacial associada às aplicações do VirD-GM. O desenvolvimento da GPU-Aware, como extensão da VirD-GM, provê suporte a simulação quântica distribuída a partir de *GPUs* constituindo uma solução para redução do tempo de simulação decrementando a complexidade temporal nas aplicações.

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.

Referências

Avila, A., Maron, A., Reiser, R. e Pilla, M. (2014a). GPU-aware distributed quantum simulation. In *Proceedings of 29th Symposium On Applied Computing*, pages 1–6.

Avila, A., Schmalfuss, M., Maron, A., Reiser, R. e Pilla, M. (2014b). Simulação distribuída de algoritmos quânticos via GPUs. In *Proceedings of the XV WSCAD*, pages 1–12.

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