OTIMIZANDO AS SIMULAÇÕES DO AMBIENTE D-GM: REDUÇÃO E DECOMPOSIÇÃO BASEADAS NO OPERADOR IDENTIDADE

ANDERSON BRAGA DE AVILA[1];
RENATA HAX SANDER REISER[1]; MAURÍCIO LIMA PILLA[1]

[1]*Universidade Federal de Pelotas* –{abdavila, reiser, pilla}@inf.ufpel.edu.br

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 do ambiente D-GM, incrementando suas capacidades de simulação usando GPUs. Mais especificamente, a principal contribuição se dá pela otimização da representação de transformações quânticas pelo uso inteligente do operador Identidade e pela decomposição dos operadores, que apesar de aumentar o número de passos necessário para o cálculo de uma transformação quântica(TQ), reduz o tempo de execução.

2. METODOLOGIA

Em trabalhos anteriores (AVILA, 2014) processos relacionados a TQs *n*-dimensionais eram definidos por um conjunto de matrizes de menor ordem para reduzir a memória usada nas simulações devido ao seu crescimento exponencial (2^n x 2^n). Elementos da TQ necessários para sua computação eram gerados em tempo de execução pela iteração sobre estas matrizes, simulando o comportamento do produto tensor. No entanto, o tempo computacional gasto nestas iterações é alto, se tornando um gargalo para a execução de algoritmos quânticos.

A primeira otimização proposta neste trabalho consiste em explorar o comportamento do produto tensor entre o operador Identidade(*I*) e outros operadores. Nestes casos, o operador Identidade não só replica os dados dos outros operadores como também introduz esparcialidade na TQ.

Um exemplo é retratado na Eq. 1 para a transformação Hadamard. Portanto é possível armazenar somente a expansão do produto tensor entre operadores diferentes de *I*, e utilizar a posição destes operadores como referência para acessar os elementos quando se dá o cálculo da TQ, reduzindo então a complexidade espacial associada a representação de TQs com estas características.

Porém nem todas TQs apresentam operadores Identidade ou uma quantidade suficiente que torne possível representá-la por uma única matriz. Tendo isto em mente, a segunda otimização proposta consiste na decomposição dos operadores de uma TQ *n*-dimensional, aumentando o número de passos para o cálculo da mesma, o que permite controlar a quantidade de operadores *I* em cada passo, preservando o comportamento e propriedades da TQ.

$$I \otimes H = \begin{pmatrix} 1 & 0 \\ 0 & \mathbf{1} \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & \mathbf{-1} \end{pmatrix} \quad (1)$$

Na Figura 1 é mostrado que a transformação $H \otimes H$ pode ser descrita em dois passos, $H \otimes I$ e $I \otimes H$, mantendo o mesmo comportamento independente da ordem da composição destes passos. TQs controladas também podem ser decompostas desde que os operadores conservem os seus controles.

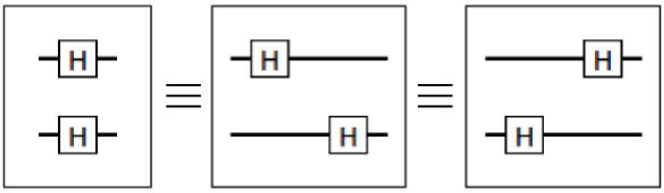

Figura 1. Decomposição do Operador Hadamard.

Usando estas duas otimizações, é possível criar um algoritmo eficiente que não precise simular o comportamento do produto tensor. Pois a complexidade espacial pode ser reduzida limitando o número de operadores diferentes de *I* presentes em cada passo da decomposição, tornando possível a representação deles por uma única matriz.

Apesar de ser possível modelar TQs *n*-dimnesionais com baixa complexidade espacial, usando as abordagens mostradas acima, o tamanho dos estados de leitura/escrita pode se tornar um fator limitante para simulação de TQs n-dimensionais, pois também crescem de forma exponencial (2^n).

Uma vez que a memória das GPUs normalmente são menores que a memória RAM principal, é necessário adotar uma abordagem que forneça escalabilidade para a simulação de TQs multi-qubits.

O conceito de Processos Mistos Parcias (MPP), apresentando em AVILA et al. (2014), provê controle sobre o tamanho das memórias de leitura/escrita no cálculo de uma TQ, contribuindo para o aumento da escalabilidade. Baseando-se neste conceito, TQs com mais qubits que o limite suportado pela memória da GPU, podem ter seus estados particionados em 2^p sub-estados, onde *p* indica o número de qubits acima do limite, tornando possível a simulação da TQ.

Para manter a consistência do resultado, cada passo da decomposição com operadores diferentes de *I* nos *p* primeiros qubits precisa calcular todas os sub-estados antes de prosseguir para o próximo passo, pois precisam acessar mais de um sub-estado de leitura. E para os outros passos, os sub-estados podem ser calculados de forma iterativa, pois só precisam acessar o correspondente sub-estado de leitura, ou seja, a saída de cada passo serve como entrada para o próximo.

No caso de controles serem afetados, somente as partes que satisfaçam estes controles são necessárias.

3. RESULTADOS E DISCUSSÃO

A principal contribuição deste trabalho consiste na implementação de uma biblioteca composta pelas otimizações apresentadas e pode ser validada pela simulação de transformações Hadamard de 21 até 28 qubits. Considerando parâmetros de limite de operadores por passo variando de 1 até 5 e limite de qubits para simulação em GPU de 26 até 28 qubits com o intuito de analisar o comportamento do novo algoritmo do projeto D-GM. A escolha da transformação Hadamard se dá pelo fato dela ser a operação com maior custo computacional na simulação de algoritmos quânticos.

Os tempos médios de execução, para todas as combinações de transformações e parâmetros, foram obtidos depois de 40 execuções, descartando os 5 menores e maiores tempos obtidos. Os teste foram realizados em um desktop com processador Intel Core i7-3770, 8 GB RAM, GPU NV TeslaK20 e sistema operacional Ubuntu Linux 12.04, 64 bits, com CUDA 5.0.

A Figura 2 mostra os tempos de simulações obtidos, sem considerar limitação de qubits, divididos em dois gráficos com escalas diferentes para melhor visualização. Observa-se que o tempo para qualquer Hadamard diminui conforme o limite de operadores varia de 1 até 3, e aumenta quando varia de 3 até 5. A tendência é continuar aumentando conforme aumenta-se o limite, mostrando que simulações com limite de operadores igual a 3 obtêm o melhor desempenho neste hardware.

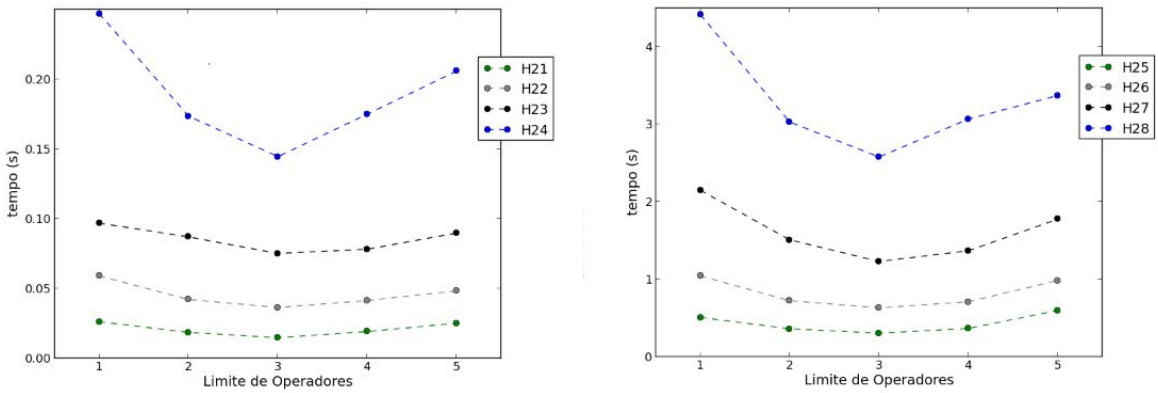

Figura 2. Tempos médios de simulação sem limite de qubits.

A Figura 3 mostra os tempos de simulação para Hadamares de 27 e 28 qubits, considerando limites de qubits para execução de 26 até 28 qubits. Como esperado as implementações usando MPPs permitem a simulação mesmo quando o limite de qubits para execução é menor que o número de qubits da transformação sendo calculada. No entanto, o tempo de simulação aumenta conforme a quantidade de qubits que passam do limite, pois haverá mais operadores afetados pela partição do estado.

Tempos médios de simulação usando nosso método anterior foram medidos para transformações Hadamard de 21 e 22 qubits com tempos de 110,407s e 395,951s respectivamente. O speedup relativo obtido comparando nosso melhor resultado neste trabalho com o método anterior foi de ≈ 10.829 vezes. Este speedup tende a escalar com o número de qubits, pois a taxa de crescimento do

tempo conforme o aumento do número de qubits é de ≈ 2 vezes, enquanto no método anterior é de ≈ 3,6 vezes.

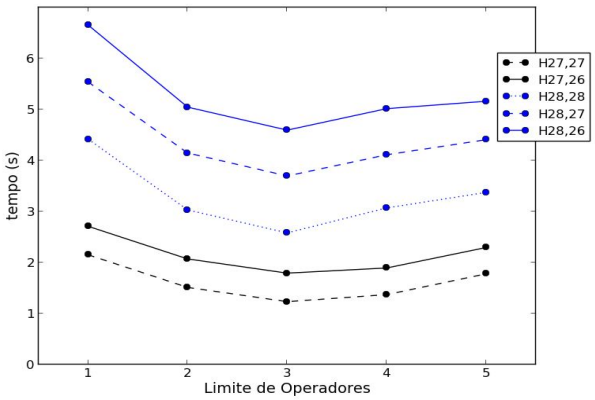

Figura 3. Tempos médios de simulação com limite de qubits.

4. CONCLUSÕES

Neste artigo, foi apresentada uma nova abordagem para reduzir a complexidade espacial e temporal na simulação de computação quântica. Pelo uso da otimização dos operadores Identidade, decomposição de transformações e de processos mistos parciais, foi possível simular um grande número de qubits em uma única GPU.

Experimentos com transformações Hadamard mostraram que é possível realizar simulações de até 28 qubits em uma única GPU. Quando comparado com nosso trabalho anterior, o melhor speedup relativo foi obtido para a Hadamard de 22 qubits, 10.829 vezes mais rápido que nossa abordagem anterior usando o mesmo hardware.

Em trabalhos futuros, pretende-se estender a abordagem desenvolvida neste trabalho para simulações em ambientes heterogêneos distribuídos compostos por vários computadores com GPUs.

5. REFERÊNCIAS BIBLIOGRÁFICAS

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.

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

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

AVILA, A.; SCHMALFUSS, M.; MARON, A.; REISER, R.; PILLA, M. Optimizing Quantum Simulation for Heterogeneous Computing: a Hadamard Transformation Study. In: Journal of Physics Conference Series, pages 1–17, 2015. (Submetido)