Otimizações de simulações quânticas no ambiente VPE-qGM

Giovani de Quadros Rodrigues[*1], **Renata H. S. Reiser**[1], **Maurício L. Pilla**[1]

{gdqrodrigues,reiser,pilla}@inf.ufpel.edu.br

[1]Centro de Desenvolvimento Tecnológico – Universidade Federal de Pelotas (UFPel)
Caixa Postal 354 – 96.010-900 – Pelotas – RS – Brazil

Resumo. *A indisponibilidade de hardware quântico exige a simulação de circuitos quânticos em computadores clássicos. Este artigo propõe uma otimização no desempenho do ambiente VPE-qGM, simplificando a representação estrutural de dados, priorizando baixo uso de memória e operações de baixo nível.*

1. Introdução

A Computação Quântica é fundamentada nos princípios da mecânica quântica, estimando obter um desempenho superior ao da computação clássica em razão de seu paralelismo. Os estudos na área foram intensificados a partir do algoritmo de fatoração de números de alto valor proposto por Peter Shor [Shor 1994].

2. Conceitos básicos

Na computação clássica, como observado em [Nielsen and Chuang 2010], um bit pode assumir 0 ou 1 como valor, de forma determinística. Analogamente, os bits quânticos (qubits) são representados por um estado de sobreposição, definido na notação de Dirac [Dirac 1939] como $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, onde α e β são os coeficientes complexos do estado ortonormalizado ψ, onde $|\alpha|^2 + |\beta|^2 = 1$. A geração de estados quânticos com múltiplos qubits é dada através da aplicação do produto tensorial.

Assim, sejam $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$,$|\phi\rangle = \gamma|0\rangle + \delta|1\rangle$. Então o resultado do produto tensorial entre os estados $|\psi\rangle$ e $|\phi\rangle$ resulta no estado $|\psi\rangle \otimes |\phi\rangle = \alpha\gamma|00\rangle + \alpha\theta|01\rangle + \beta\gamma|10\rangle + \beta\theta|11\rangle$, representado abaixo matricialmente:

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \otimes \begin{pmatrix} \gamma \\ \theta \end{pmatrix} = \begin{pmatrix} \alpha\gamma & \alpha\theta & \beta\gamma & \beta\theta \end{pmatrix}^t$$

As transformações quânticas são correspondentes análogas às portas lógicas da computação clássica. Abaixo estão representadas as transformações básicas Pauli-X (NOT), Pauli-Y, Pauli-Z, Fase, $\pi/8$, Hadamard e Identidade, respectivamente:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Em cálculos envolvendo dois ou mais qubits, uma transformação quântica pode ser controlada, dependendo, para ser aplicada em um qubit, do valor de um segundo qubit.

*Bolsista SIGFAPERGS

Para alterar o valor de um qubit, é realizado um produto matricial entre a transformação e o qubit. A aplicação da transformação Hadamard nos estadoa clássicos $|0\rangle$ e $|1\rangle$ resulta em estados de sobreposição quântica, cuja correspondente representação matricial é dada por:

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}; \; e \; \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

3. VPE-qGM

O ambiente VPE-qGM [Maron et al. 2013], implementado na linguagem Python, oferece suporte para a modelagem e simulação de circuitos quânticos sincronizando processos elementares (PEs). Devido ao crescimento exponencial do tamanho dos circuitos em função ao número de qubits, o desempenho das simulações de circuitos quânticos obtêm uma complexidade assintótica $O\left(4^n\right)$, o que nos incentiva a buscar melhorias.

4. Otimizações no ambiente VPE-qGM

As otimizações se concentraram em ambas abordagens, redução espacial (dados e/ou manipulação da memória) e temporal (operações durantes as simulações).

4.1. Dados

Os dados são representados através de duas classes, sendo a primeira delas a definição de uma matriz para representar bits ou portas quânticas, cujos atributos são o número de linhas e um vetor de instâncias de uma segunda classe, que contém os seguintes valores:

(i) Números não nulos: Caso uma linha ou um estado apresente apenas um número diferente de zero, é armazenada apenas a posição onde encontra-se o número. Assim, o valor do estado $|010\rangle$ é 2 e o da linha $[\,0\;1\,]$ é 1.

No caso de existirem mais números, o valor armazenado é um número inteiro com um bit para cada coluna. Tal bit vale 1 quando houver um valor não nulo na posição correspondente. Logo, a linha $\frac{1}{\sqrt{2}}\,[1\;1\,]$ equivale ao binário 11.

O estado ou linha tem representação vetorial caso um de seus elementos não se forme através das portas básicas, isto é, possua ao menos um número complexo diferente de 0,1 ou potência de $\frac{1}{\sqrt{2}}$.

(ii) Sinais: Usado quando a linha ou estado possui ao menos um valor negativo. Se há um único número não nulo, assume um valor booleano de verdadeiro. Caso contrário, é gerado um inteiro semelhante ao do primeiro valor, onde os bits valem 1 quando seu correspondente no inteiro dos não nulos é negativo. Assim, a linha $[\,1\;-1\,]$ equivale ao valor 01.

(iii) Números imaginários: Possui as mesmas características do valor de sinais. Caso exista um valor i, o valor representa sua posição.

(iv) Raiz de i: Raramente usado. Representa valores múltiplos de $e^{i\pi/4}$.

4.2. Simulações de Processos Quânticos

A principal alteração realizada é aplicar para cada qubit a sua transformação correspondente ao longo dos estados. Uma vez atingido o estado final, realiza-se um produto

tensorial entre os qubits deste estado. Assim, para simular circuitos quânticos como o apresentado na figura 1, são realizados os seguintes procedimentos:

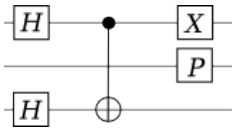

Figura 1. Circuito quântico

Primeiramente, aplicam-se diretamente as portas Hadamard nos qubits 1 e 3, a partir do estado inicial $|100\rangle$.

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix} \qquad \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

A seguir, ocorre a aplicação da CNOT. O primeiro passo é analisar os controladores. Uma vez que o bit controlador assume um valor diferente de $|1\rangle$, a transformação NOT não acontece no terceiro qubit. O estado quântico, assim, permanece inalterado.

Para finalizar as transformações, o primeiro qubit recebe a transformação Pauli-X e o segundo passa pela transformação não-básica P.

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} -1 \\ 1 \end{pmatrix} \qquad \begin{pmatrix} 0,6 & 0,8 \\ 0,8 & 0,6 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0,6 \\ 0,8 \end{pmatrix}$$

Ao final, é realizado o produto tensorial: $\frac{1}{\sqrt{2}}\begin{pmatrix} -1 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0,6 \\ 0,8 \end{pmatrix} \otimes \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ resultando no vetor transposto:

$$\frac{1}{2}\left(-0.6, -0.6, -0.8, -0.8, -0.6, -0.6, -0.8, -0.8\right)^t.$$

4.2.1. Transformações Quânticas

Para aplicar uma transformação sobre um qubit adaptada à representação de dados proposta em 4.1, três casos são possíveis.

No primeiro caso, o qubit possui apenas um número não nulo representado em um valor V, como ocorre em $|0\rangle$ e $|1\rangle$. os valores do qubit após a transformação recebem os valores da linha V da porta nele aplicada, independente da representação de dados da linha, uma vez que a linguagem Python possui o recurso de conversão automática de tipo.

No segundo caso, os dois números do qubit são múltiplos de $\frac{1}{\sqrt{2}}$. Se uma linha da transformação possuir um único número não nulo, seu valor de não-nulos V é convertido para 2^V, bem como seus outros valores, caso existam. O valor dos não-nulos resultante é o resultado de OR entre as duas linhas. Os outros valores são obtidos entre o operador XOR entre as linhas e o qubit. Também é realizada uma operação AND entre esses três valores, fazendo com que o valor final referente a i seja o resultado de OR entre esse valor

e a AND do valor referente a $\sqrt{i}$ e o valor final de sinais seja o resultado de OR entre esse valor e a AND do valor referente a i, uma vez que $\left(e^{i\pi/4}\right)^2 = i$ e $i^2 = -1$.

No terceiro caso, o qubit ou um dos vetores da transformação não deriva de estados ou transformações básicos. Assim, todos os valores otimizados sãoconvertidos para matrizes, executando-se a multiplicação matricial entre a transformação e o qubit.

4.2.2. Produto Tensorial Final

Se os n qubits possuírem apenas um valor não nulo, então o numero de oper ações é dado pela expressão$\sum_{i=0}^{n} (V_i \times 2^i)$.

Caso um dos qubits tenha dois números não nulos, o valor V dos que possuem um valor apenas é convertido para 2^V e os valores finais são acumulados pela operação AND para valor dos não-nulos e XOR para os demais valores, seguindo os mesmos cálculos vistos no segundo caso de na Sessao 4.2.1, entre o resultado de $V_b \times (2^n - 1)/(10^{2^{n-b-1}} + 1)$ de cada qubit b em um total de n qubits.

Se pelo menos um qubit for formado por um número complexo, os outros qubits passam a ser representados por vetores de números complexos e cada célula do vetor resultante recebe o valor de $\prod_{q=0}^{n} (L_q)$, onde L_q representa a linha L de um qubit q, em estados quânticos n-dimensionais. Assim, a linha 5 (101 em binário) de um vetor final em um estado com 3 qubits corresponde ao produto entre a linha 1 do primeiro qubit, a linha 0 do segundo e a linha 1 do terceiro.

5. Considerações Finais

As mudanças estruturais propostas visam alcançar uma significativa redução no tempo de execução das simulações realizadas no ambiente VPE-qGM, uma vez que conquistam ambos objetivos: (i) redução de memória através da alteração na representação de dados, e (ii) redução na complexidade temporal do algoritmo, atingindo, a partir da redução de memória, a notação assintótica $O(2^n)$ em análise de pior caso, para estados com n qubits. Como trabalho futuro, pretende-se implementar tais otimizações no código do ambiente VPE-qGM.

Referências

Dirac, P. A. M. (1939). A new notation for quantum mechanics. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 35, pages 416–418. Cambridge Univ Press.

Maron, A., Reiser, R., and Pilla, M. (2013). High-performance quantum computing simulation for the quantum geometric machine model. In *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, pages 474–481.

Nielsen, M. A. and Chuang, I. L. (2010). *Quantum computation and quantum information*. Cambridge university press.

Shor, P. W. (1994). Algorithms for quantum computation: discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 124–134. IEEE.