

Validação do Protótipo VPE-qGM: Modelagem e Simulação do Algoritmo de Grover

Adriano K. Maron, Renata H. S. Reiser, Adenauer C. Yamin

¹Centro Politécnico – Universidade Católica de Pelotas (UCPEL)
96.030-000 – Pelotas – RS – Brazil

{maron, reiser, adenauer}@ucpel.tche.br

Abstract. *The advances in Quantum Computation (QC) have encouraged the study of its foundations, aimed at understanding and development of new algorithms. In this context, visual programming environments for modeling and simulation of quantum algorithms have contributed to these efforts, making the calculations required and showing the results through graphical interfaces. This paper presents the various models and simulations for the Grover Algorithm provided by the VPE-qGM environment, including different levels of detail.*

Resumo. *Os avanços na Computação Quântica têm incentivado o estudo de seus fundamentos, visando a compreensão e o desenvolvimento de novos algoritmos. Neste contexto, ambientes de programação visual para modelagem e simulação de algoritmos quânticos têm contribuído com esses esforços, realizando os cálculos exigidos e apresentando os resultados através de interfaces gráficas. Este trabalho apresenta as diferentes modelagens e simulações para o Algoritmo de Grover disponibilizadas pelo ambiente VPE-qGM, incluindo diferentes níveis de detalhamento.*

1. Introdução

A computação quântica (CQ) considera *qubits* ao invés de *bits*, os quais podem assumir, além dos estados básicos 0 e 1, uma superposição coerente que representa um novo estado. Um *qubit* é definido por um vetor pertencente ao espaço de Hilbert complexo e bi-dimensional, representado por um estado normalizado $|\psi\rangle = a|0\rangle + b|1\rangle$ ¹, ou seja, combinações lineares de vetores básicos $|0\rangle$ e $|1\rangle$, onde a e b são coeficientes complexos correspondentes às probabilidades associadas à superposição dos estados básicos.

A manipulação das probabilidades associadas a transformações dos estados dos *qubits* é feita pela execução de algoritmos quânticos [Nielsen and Chuang 2000], obtidos por composição de transformações unitárias, projeções e operações de medidas. Através das propriedades particulares da Mecânica Quântica, a qual fundamenta as teorias da CQ , esses algoritmos proporcionam um ganho de desempenho exponencial se comparados com os algoritmos clássicos. A complexidade desses algoritmos é diretamente proporcional a quantidade de operações aplicadas, justificando a utilização de programação paralela em simuladores quânticos. Mais significativo, o uso de macros na VPE-qGM viabiliza um detalhamento dos operadores de sincronização que definem o algoritmo.

¹Aplica-se a notação de Dirac [Nielsen and Chuang 2000] para processos e estados quânticos.

2. VPE-qGM

O protótipo VPE-qGM (*Visual Programming Environment for Quantum Geometric Machine Model*) [Maron et al. 2009] está sendo desenvolvido com o objetivo de disponibilizar uma ferramenta para modelagem de processos quânticos seguindo as abstrações definidas no modelo qGM (*Quantum Geometric Machine Model*) [Reiser et al. 2007]. Esse modelo introduz uma nova fundamentação para a construção e simulação de algoritmos quânticos, considerando a noção de processos associados a posições do espaço geométrico, modelando componentes da base computacional e provendo semântica para interpretação de portas quânticas pela sincronização de processos parciais clássicos.

A versão atual deste protótipo apresenta uma estrutura composta pelos seguintes componentes: (i) editor quântico de processos (*qPE*), o qual disponibiliza construtores para definição de operadores quânticos personalizados, macros, medidas quânticas, sincronização e composição de operadores quânticos; (ii) editor quântico de memória (*qME*), componente no qual são configuradas estruturas de memória com controle da condição de normalidade; (iii) simulador quântico (*qS*), apresentando uma interface para visualização de simulações (sequenciais) das computações modeladas.

3. Algoritmo de Grover: Descrição

O Algoritmo de Grover (*AG*) [Nielsen and Chuang 2000] caracteriza-se pela busca de um elemento específico em uma lista não-ordenada de N elementos. Classicamente, $N - 1$ consultas são executadas para identificação do elemento procurado. Usufruindo de propriedades como superposição de estados, paralelismo e interferência quântica, o algoritmo *AG* reduz a quantidade de testes para ordem \sqrt{N} .

O algoritmo é composto por dois registradores, sendo o primeiro (*R1*) composto por N *qubits* e inicializado no estado clássico $|0\dots 0\rangle$. Esse registrador representa os elementos que compõem a lista, sendo cada elemento denotado por um estado $|i\rangle$. Assim, um registrador de N *qubits* pode representar uma lista contendo 2^N elementos. O segundo registrador (*R2*), contém o qubit de controle, inicializado no estado $|1\rangle$.

Para este estudo de caso, considera-se uma lista com $N = 4$, na qual busca-se o terceiro elemento, identificado pelo estado básico $|10\rangle$. Neste caso, *R1* tem dois *qubits*. Nas seções seguintes, são apresentadas modelagens por processos elementares e macros, exibindo o algoritmo em diferentes níveis de detalhamento. De acordo com o circuito para *AG*, descrito em [Prokopenya 2009], aplicam-se as seguintes portas quânticas: *Hadamard* (*H*), *Not* (*X*), *Identity* (*Id*), *CompositeGate* ($C^x(Y)$) e a *Rotation* (*Z*).

3.1. Modelagem no Ambiente VPE-qGM

No protótipo VPE-qGM, a modelagem do algoritmo através de processos elementares apresenta detalhadamente a construção dos operadores quânticos. Cada processo quântico é definido pelas sincronizações dos processos parciais indexados por distintas posições de memória. Tem-se, portanto, uma representação diretamente proporcional à quantidade de estados da base computacional requerida pelo operador, condição garantida pela geração dinâmica de processos implementada no componente *qPE*.

A aplicação dos construtores produto paralelo (*PP*) e produto sequencial (*PS*) a dois ou mais processos quânticos permite a escolha da representação por macros, possibilitando a determinação de diferentes níveis de detalhamento para a aplicação modelada,

conforme apresentado na Figura 1. As aplicações modeladas no componente qPE podem ser salvas em um arquivo descritor de processos no formato XML, sendo este utilizado para reconstrução dos operadores e simulação através do componente qS .

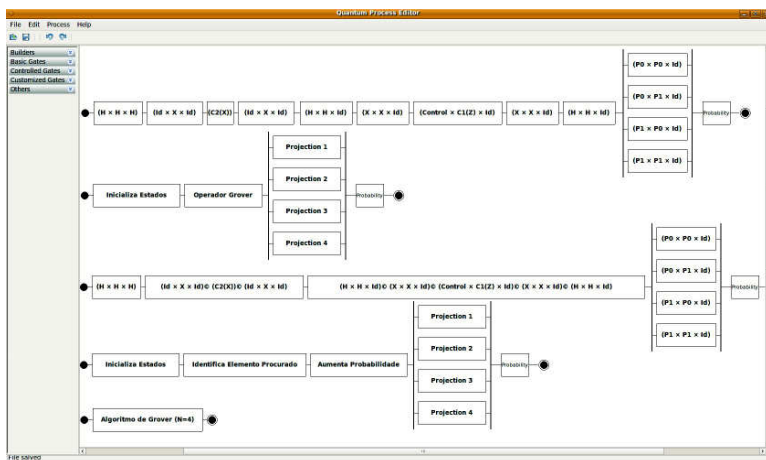


Figura 1. Possíveis Modelagens para o Algoritmo de Grover

3.2. Simulações no Ambiente VPE-qGM

A simulação de algoritmos quânticos através do componente qS exige a seleção do arquivo descritor de processos, gerando a lista de execução de processos utilizada durante a simulação. Segue-se uma análise dos parâmetros dos processos, identificando as portas quânticas utilizadas em cada unidade de tempo computacional, representada no VPE-qGM por uma sincronização de processos elementares ou pela correspondente macro.

Após análise, os dados referentes aos operadores quânticos são enviados para a biblioteca $qGM-Analyzer$, a qual contém métodos de suporte à aplicações quânticas. Essa biblioteca identifica, caso existam definições, os operadores matriciais associados aos parâmetros de entrada. Caso contrário, consultam-se os dados no campo relativo às definições de operadores personalizados dos processos elementares e macros.

Segue-se a aplicação do produto tensorial, definido na biblioteca $qGM-Analyzer$, entre os operadores matriciais obtidos na etapa anterior, gerando a matriz referente à sincronização dos operadores quânticos. Caso seja utilizada uma modelagem na forma de processos elementares, os vetores componentes da matriz resultante são atribuídos aos processos elementares. Na modelagem por macros, a matriz resultante é armazenada como parâmetro dos respectivos componentes. Para situações onde uma macro representa n sincronizações, aplica-se o produto tensorial entre as matrizes das respectivas sincronizações, gerando n matrizes que são armazenadas como parâmetros da macro.

Durante a realização da simulação, cada solicitação de execução de uma macro ou de uma sincronização de processos elementares implica na transferência dos valores presentes nas posições de memória para um vetor, o qual representa o estado global corrente

da simulação. Se o componente a ser executado é uma macro, o(s) operador(es) associado(s) a esta macro durante o processo de inicialização é(são) aplicado(s) sequencialmente ao vetor de memória através do operador produto tensorial. Assim, atualizam-se as posições de memória do simulador com o resultado obtido. Se for solicitada a execução de um processo elementar, o componente associado a este será aplicado ao vetor de memória, alterando somente a posição de memória especificada em sua definição.

O resultado da simulação (Figura 2) é apresentado na área de memória do *qS*, onde cada posição contém a probabilidade associada ao seu correspondente estado da base. No estudo de caso, a aplicação de medida sobre os dois primeiros *qubits* do sistema *AG* apresenta os estados $|100\rangle$ ou $|101\rangle$ com 100% de probabilidade de ocorrência.

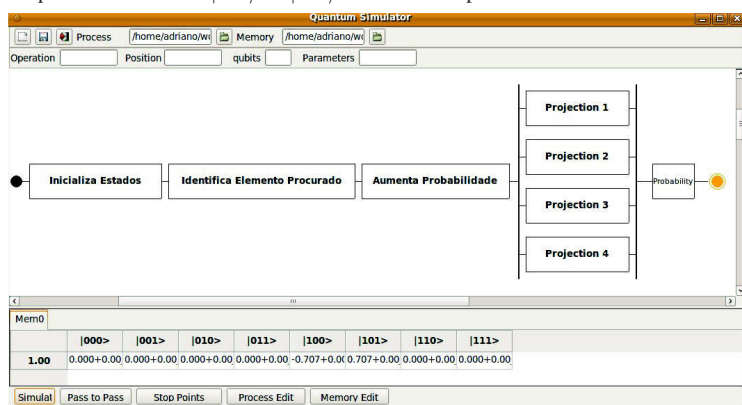


Figura 2. Simulação do Algoritmo de Grover

4. Considerações Finais

Na busca de novos resultados, as atividades futuras consideram a integração do protótipo VPE-qGM ao ambiente de execução VirD-GM [Fonseca et al. 2007]. Esta integração viabiliza a implementação paralela e/ou distribuída de algoritmos (básicos) da CQ, enfatizando a simulação do paralelismo quântico em arquiteturas multi-processadas.

Referências

- Fonseca, V., Reiser, R., Yamin, A., and Pilla, M. (2007). VirD-gm: Towards a grid computing environment. In *Proceedings of CCGRID 2007*, pages 1–6.
- Maron, A., Reiser, R., Yamin, A., and Pilla, M. (2009). Aplicações do protótipo vpe-qgm: Simulação via software do algoritmo de grover. In *X WSCAD/WIC*, pages 1–4.
- Nielsen, M. and Chuang, I. (2000). *Quantum Computation and Quantum Information*. Cambridge University.
- Prokopenya, A. (2009). Wolfram demonstrations project - quantum circuit implementing grover's search algorithm.
- Reiser, R., Amaral, R., and Costa, A. (2007). Quantum computing: Computation in coherence spaces. In *Proceedings of WECIQ 2007*, pages 1–10. UFCG.