

## OTIMIZAÇÕES NA MODELAGEM GRÁFICA DO AMBIENTE VPE-QGM

**RETZLAFF, DANIEL<sup>1</sup>; MARON, ADRIANO<sup>1</sup>; REISER, RENATA<sup>1</sup>; PILLA, MAURÍCIO<sup>1</sup>**

<sup>1</sup> Centro de Desenvolvimento Tecnológico CDTEC,  
Universidade Federal de Pelotas.  
Email: {dkretzlaff, akmaron, reiser, pilla}@inf.ufpel.edu.br.

### 1. INTRODUÇÃO

Os avanços na Computação Quântica (CQ) estão viabilizando a construção de hardwares quânticos que exploram fenômenos da Mecânica Quântica (MQ) para solução de problemas complexos, intratáveis para os supercomputadores atuais, KNILL et al. (2011). Algoritmos quânticos estão sendo estudados e novas propostas de aplicações vem sendo desenvolvidas, porém poucas plataformas estão disponíveis para interpretações e testes, visto a simplicidade do hardware quântico atual, HANNEKE et al. (2009) e MONZ et al. (2011).

Nesse contexto, modelos e simuladores quânticos, sequenciais ou paralelos, introduzem novas abordagens para estudo e desenvolvimento de aplicações baseadas neste paradigma, veja QUANTIKI et al. (2011). Entretanto, tais abordagens são ainda ineficientes no sentido de prover suporte a sistemas quânticos complexos, ou são pouco intuitivas para interação com usuários não familiarizados com a MQ.

Fundamentado nas abstrações do modelo qGM (*Quantum Geometric Machine*) REISER et al. (2011), o ambiente VPE-qGM (*Visual Programming Environment for the qGM Model*) está em desenvolvimento integrando esforços junto aos grupos LUPS/UFPEL e MFFM-CC/UFPEL, para suporte a modelagem e simulação distribuída de algoritmos da CQ. O ambiente VPE-qGM viabiliza a simulação sequencial e/ou distribuída de algoritmos quânticos com até 24 *qubits*, detalhados em MARON et al. (2012).

O objetivo principal deste trabalho é otimizar a disposição gráfica do vetor de estados nas interfaces do ambiente VPE-qGM, tendo como principal motivação a possibilidade de representação multidimensional para a melhor visualização dos resultados gerados na simulação de uma aplicação. Neste contexto, otimiza-se também a estrutura dos dados utilizados pelo sistema de controle de memória do VPE-qGM. Outra proposta visa é prover suporte a sistemas multidimensionais, disponibilizando opções de redistribuição dos dados gerados pela simulação de algoritmos quânticos nas interfaces dos editores de memória e processos.

### 2. PRELIMINARES

Conceitualmente, o *qubit* consiste na unidade básica de informação quântica, sendo matematicamente definido por um vetor de estado, unitário e bidimensional, genericamente descrito na notação de Dirac pela expressão  $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ , onde os coeficientes  $\alpha$  e  $\beta$  são números complexos, associados às amplitudes dos respectivos estados normalizados.

O espaço de estados de um sistema quântico de 2 qubits é obtido pelo produto tensorial de seus sistemas componentes. Sejam os sistemas quânticos  $|\Psi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$  e  $|\Psi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$ , tem-se que o espaço de estados composto pelo produto tensor será dado pela expressão:

$$|\Psi_1\rangle \otimes |\Psi_2\rangle = \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \alpha_2\beta_1|10\rangle + \alpha_2\beta_2|11\rangle$$

## 2. METODOLOGIA

Os níveis de organização arquitetural estão graficamente descritos na Figura 1. Cada nível no D-GM é concebido com uma finalidade cuja integração resulta na unificação do processo de modelagem e/ou simulação.

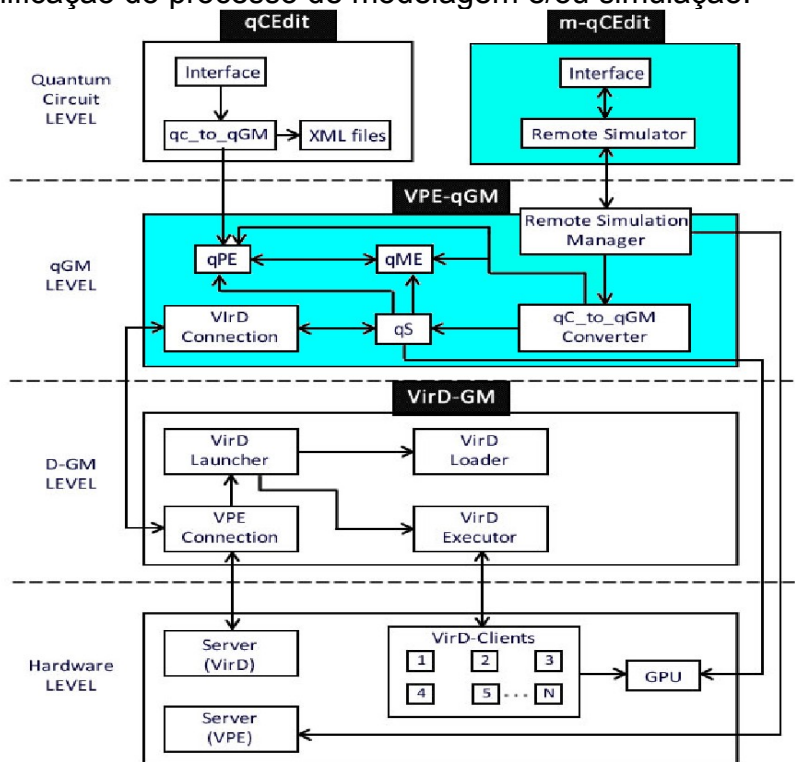


Figura 1: Arquitetura do ambiente D-GM.

No nível de circuitos pelo ambiente q-CEdit (m-qCEdit), pode-se descrever a aplicação (móvel) seguindo o modelo de circuitos quânticos e, em seguida, exportá-la, automaticamente, para uma representação correspondente no modelo qGM, com visualização nas interfaces gráficas do VPE-qGM. No VPEq-GM(Figura 2),

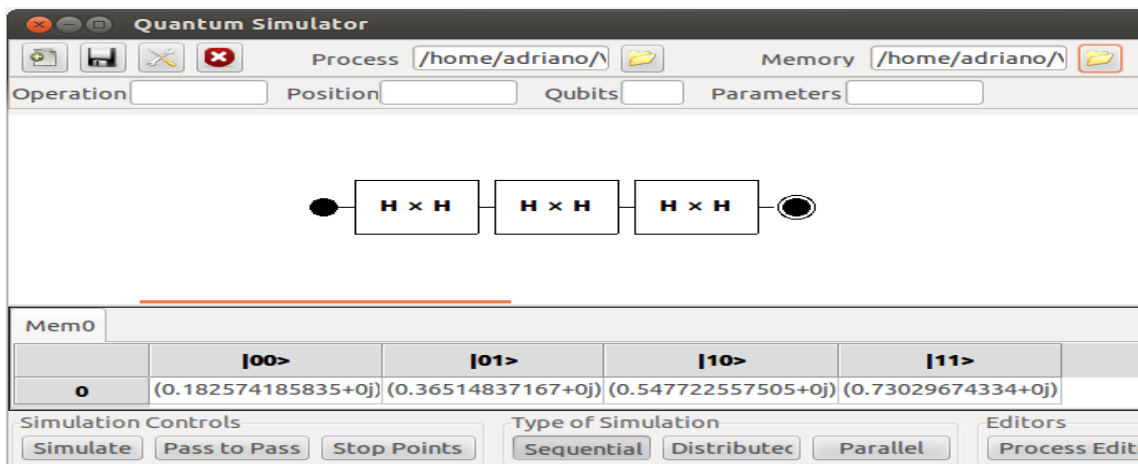


Figura 2: Interface gráfica do ambiente VPE-qGM.

o usuário é capaz de descrever e simular as computações referentes a aplicação em desenvolvimento. No nível do D-GM, o gerente de simulação distribuída, nomeado VirD-GM (*Virtual Distributed Geometric Machine*) e responsável por tarefas como comunicação, agendamento e sincronização, quando da simulação distribuída. Finalmente, em nível de hardware, são listados todos os dispositivos que podem ser explorados por este framework, de computadores desktops normais até simulação sequencial em clusters com múltiplas GPUs. Este trabalho concentrou-se em otimizações nas interfaces do Editor de Memórias Quântico m-qCEdit e do qME do VPE-qGM, detalhadas na sequência.

### 3. RESULTADOS E DISCUSSÃO

As modificações efetuadas sobre o ambiente VPE-qGM visam um melhor aproveitamento do espaço reservado ao vetor de estados. Tanto a modelagem quanto a implementação e otimizam a apresentação e formatação de estados de memória multidimensionais.

A otimização na interface gráfica teve como principal característica a disposição e/ou reorganizada do sistema de memória. O objetivo desta otimização é prover ao usuário a possibilidade de limitar a visualização de estados de memória n-dimensionais (representado por um vetor de  $2^n$  posições), tendo como foco principal disponibilizar na interface posições específicas, as quais contém resultados significativos do algoritmo.

Através da opção *Limit Memory* (Figura 3), já adicionada ao Editor de Memória do VPE-qGM, pode-se limitar/expandir o *grid* de disposição dos estados de memória nas interfaces de processos e memória, com o auxílio de uma janela de interação (Figura 3). Neste caso, satisfazendo os critérios da aplicação e ainda, reduzindo a expressão do estado final. O usuário informa o intervalo desejado (posições selecionadas) para melhor visualização dos correspondentes valores de memória de um estado. As modificações sobre os parâmetros, que eram armazenados na forma de uma estrutura matricial, utilizados para leitura e escrita na memória do ambiente, passam a ser implementados em forma de um vetor de memória.

Durante a execução dos testes das aplicações, todos os cálculos se estruturam na forma de vetor na representação da memória, sendo passados parâmetros importantes para a execução da aplicação. Entretanto, as computações realizadas baseada neste conceito limitavam o desempenho do ambiente, pela organização da informação na memória do simulador. A nova abordagem gerencia um sistema totalmente vetorial para o acesso à memória, viabilizando ainda futuras melhorias dentro do ambiente VPE-qGM. Caracteriza-se, uma otimização do espaço de memória utilizado pelo simulador incrementando o acesso aos dados e a apresentação dos mesmos nas interfaces.

Um estado de memória em aplicações n-dimensionais é apresentado na interface como um vetor unidimensional da ordem de  $2^n$ . A otimização desenvolvida permite a representação por uma estrutura matricial bidimensional.

Durante a execução e edição de processos quânticos, a apresentação dos dados estava limitada à visualização de sistemas exponenciais de ordens elevadas, que agora podem ser comprimidos em forma de uma matriz, obtendo-se uma visualização otimizada dos valores de memória para cada processo selecionado. Esta matriz deve respeitar a condição de multiplicidade por 2,  $2^n \times 2^m$  é igual ao tamanho máximo do vetor inicial, onde n está relacionado as linhas e m as colunas. Assim, para um vetor de 8 posições, tem-se algumas das

opções de seleção:  $2^2 \times 2^1$  ou  $2^1 \times 2^2$  ou  $2^3 \times 2^0$  ou  $2^0 \times 2^3$ , a ser definida pelo usuário.

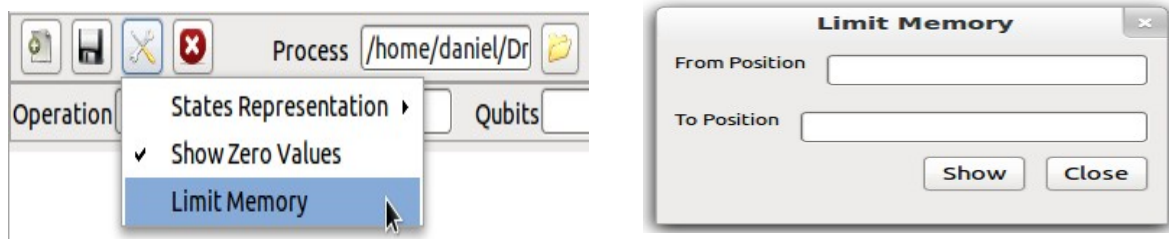


Figura 3: Interfaces gráficas da função *Limit Memory* (Opção e Janela).

#### 4. CONCLUSÕES

A constante evolução das interfaces do VPE-qGM visam tratar computações que exigem múltiplos qubits em sua execução. Neste trabalho, foram realizadas otimizações no ambiente VPE-qGM para modelagem e visualização e evolução dos estados de memória de aplicações multidimensionais.

Na continuidade, busca-se a integração da biblioteca JINC, JINC (2012) que utiliza dos fundamentos de *Binary Decision Diagrams*, LEE (1959), provendo uma forma otimizada de representar os dados para sistemas com múltiplos qubits, diminuindo a complexidade espacial durante a simulação. Na sequência, considera-se a paralelização dos BDDs para melhor desempenho e gerenciamento pelo VirD-GM das simulações no ambiente D-GM.

#### 5. REFERÊNCIAS BIBLIOGRÁFICAS

- KNILL, E.; NIELSEN, M. **Theory of quantum computation**. 2002.
- MONZ, T.; SCHINDLER, P.; BARREIRO, J.; CHWALLA, M.; NIGG, D.; COISH, W.; HARLENDER, M.; HÄNSEL, W.; HENRICH, M.; BLATT, R. 14-Qubit Entanglement: Creation and Coherence. **Physical Review Letters**. v. 106, 2011.
- HANNEKE, D; HOME, J.; JOST, J.; AMIMI, J. ; LEIBFRIED, D.; WINELAND, D. Realizations of a Programmable Two-Qubit Quantum Processor. **Nature Physics** v. 6, p. 13 - 16, 2009.
- QUANTIKI. **Quantum Information Wiki and Portal**, Singapore: Centre for Quantum Technology of the National University of Singapore, 2011.
- REISER, R.; AMARAL, R. The quantum states space in the qGM Model. In: Anais do WECIQ 2011, Petrópolis/RJ, p. 92-101, 2011.
- MARON, A.; REISER, R.; PILLA, M. Processos Quânticos: Uma Nova Interpretação para Transformações Quânticas no Ambiente VPE-qGM, In: Anais CLEI 2012, p. 1-8, 2012.
- LEE, C. Representation of switching circuits by binary decision diagrams. *Bell System Tech.*, v. 38, p. 985-999, 1959.
- FUJITA, M.; MCGEER, P.C., YANG, J.C.-Y.: Multi-terminal binary decision diagrams: An efficient data structure for matrix representation. **Form. Methods Syst. Des.** 10(2-3), p. 149-169, 1997.
- JINC, <http://www.jossowski.de/projects/jinc/jinc.html>, 2012. Acessado em 15 de julho de 2012.