

QUALIFICANDO A SIMULAÇÃO QUÂNTICA NO VPE-qGM

**SCHMALFUSS, Murilo¹; RETZLAFF, Daniel¹;
MARON, Adriano²; REISER, Renata, PILLA, Maurício³**

¹UFPEL, Graduação em Ciência da Computação, Pelotas {mfschmalfuss,dkretzlaff}@inf.ufpel.edu.br;

²UFPEL, Mestrado em Ciência da Computação, Pelotas, akmaron@inf.ufpel.edu.br

³UFPEL, PPGC/CDTEC, Pelotas, Brasil reiser@inf.ufpel.edu.br

1 INTRODUÇÃO

Os avanços na Computação Quântica (CQ) estão viabilizando a construção de *hardwares* quânticos, os quais exploram fenômenos da Mecânica Quântica (KNILL, NIELSEN, 2002) para solução de problemas complexos, intratáveis para os supercomputadores atuais. Algoritmos quânticos estão sendo propostos, porém exigem plataformas para interpretações e testes, visto a simplicidade do *hardware* quântico atual (HANNEKE et al., 2009; MONZ et al., 2011). Nesse contexto, modelos e simuladores quânticos, sequenciais ou paralelos, introduzem novas abordagens para seu estudo e desenvolvimento (GUTIERREZ et al., 2010; WATANABE, 2002; RAEDT et al., 2006; QUANTIKI, 2011). Entretanto, algumas abordagens mostram-se ineficientes, não suportando sistemas quânticos complexos ou pouco intuitivos.

Conceitos fundamentais na simulação de algoritmos quânticos: (i) o *qubit* é a unidade básica de informação, definido por um vetor de estado, unitário e bidimensional, genericamente descrito, na notação de Dirac (QUANTIKI), pela expressão $|\psi\rangle = |0\rangle + |1\rangle$; (ii) os coeficientes α e β são números complexos, associados às amplitudes dos respectivos estados normalizados, com $|\alpha|^2 + |\beta|^2 = 1$.

O ambiente *VPE-qGM (Visual Programming Environment for the qGM Model)* (MARON, et al., 2010) está em desenvolvimento para suporte a modelagem e simulação distribuída de algoritmos da CQ, a partir das abstrações do modelo *qGM (Quantum Geometric Machine)* (REISER, AMARAL, 2011). O ambiente *VPE-qGM*, até então, viabiliza a simulação sequencial e distribuída de algoritmos quânticos com até 15 *qubits*. O elevado tempo de simulação e o alto consumo de memória justificam a necessidade por otimizações na biblioteca de execução.

Este trabalho deu suporte na implementação do projeto em desenvolvido junto ao Mestrado em Ciência da Computação, cujos principais resultados foram publicados em (MARON et al., 2011). A abordagem logarítmica, aqui reportada, aplica funções básicas, produtos e recursão na geração dinâmica dos valores da matriz resultante, visando a redução do custo de simulação.

2 METODOLOGIA

A biblioteca *qGM-Analyzer* substitui todas as matrizes associadas às transformações quânticas básicas por um conjunto de funções elementares. São utilizados, ainda, conjuntos de tuplas (valor, posição), denominados *VPP*, os quais armazenam produtos parciais de funções elementares, e um conjunto de posição base, identificados por *sizesList*, associadas a cada conjunto. Essas estruturas são utilizadas em uma função recursiva para geração dinâmica de cada elemento associado a um vetor componente.

A geração dinâmica dos *VPPs*, os quais são armazenados em uma lista (L_{vpp}), ocorre a partir da quantidade de transformações quânticas e da posição de escrita de um *PE*, descritas nos atributos *Acao* e *Posicao*, respectivamente. Exemplificando, busca-se gerar os elementos da linha 35 da matriz $H^{\otimes 4} \otimes Id^{\otimes 2}$. A parametrização do

correspondente PE é: (i) $Acao = H, H, H, H, Id, Id$; (ii) $Posicao = 35$; (iii) $Parametros = Null$. O correspondente L_{vpp} gerado é

$$L_{vpp} = \begin{bmatrix} [(\frac{1}{2}, 0), (\frac{1}{2}, 1), (-\frac{1}{2}, 2), (-\frac{1}{2}, 3)], \\ [(\frac{1}{2}, 0), (\frac{1}{2}, 1), (\frac{1}{2}, 2), (\frac{1}{2}, 3)], \\ [(1, 3)] \end{bmatrix}$$

De forma análoga, tem-se que $sizesList = [16, 4, 1]$. A partir de L_{vpp} e $sizesList$, tem-se a possibilidade de gerar os valores finais não nulos associados a cada elemento do vetor componente. E, para evitar a expansão do espaço de estados, utiliza-se uma função recursiva.

A recursão permite calcular os produtos entre elementos de cada VPP . Produtos e posições-base são gerados quando de uma chamada recursiva, sendo passados para os próximos níveis da recursão. No caso base, caracterizado pela chamada recursiva ao último VPP , são gerados os valores e posições finais.

A posição indica qual amplitude deve ser multiplicada pelo valor gerado no caso base. O resultado dessa multiplicação é somado com os resultados calculados nas iterações e chamadas recursivas anteriores. Após o término das recursões, o valor final é retornado à primeira chamada, sendo atualizado na posição de memória indicada pelo atributo $Posição$ do PE . Esse processo caracteriza a obtenção da amplitude associada a um dos estados do sistema quântico após a execução das transformações.

3 RESULTADOS E DISCUSSÃO

A validação da biblioteca de execução considera dois algoritmos quânticos: somador quântico de 5 bits (Sum) (TAKAHASHI et al., 2009) e Algoritmo de Grover (AG) (GROVER, 1969), em instâncias para listas de 128 ($AG1$), 256 ($AG2$) e 512 ($AG3$) elementos. A metodologia adotada considera os seguintes parâmetros: (i) tempo médio de simulação; (ii) desvio padrão; (iii) consumo de memória. Os dados foram gerados a partir da execução de 15 simulações para cada estudo de caso. Para efeito de comparação, considera-se o simulador universal $Zeno$ (BARBOSA), o qual possui interface gráfica intuitiva, permitindo a implementação de circuitos quânticos complexos. A máquina utilizada para os testes possui as seguintes configurações de $hardware$: Processador Intel Core i5 2,30 GHz, 4 GB Ram e sistema operacional Linux Ubuntu 11.04 64 bits. As principais informações da simulação são apresentadas nas Fig. 1 e 2.

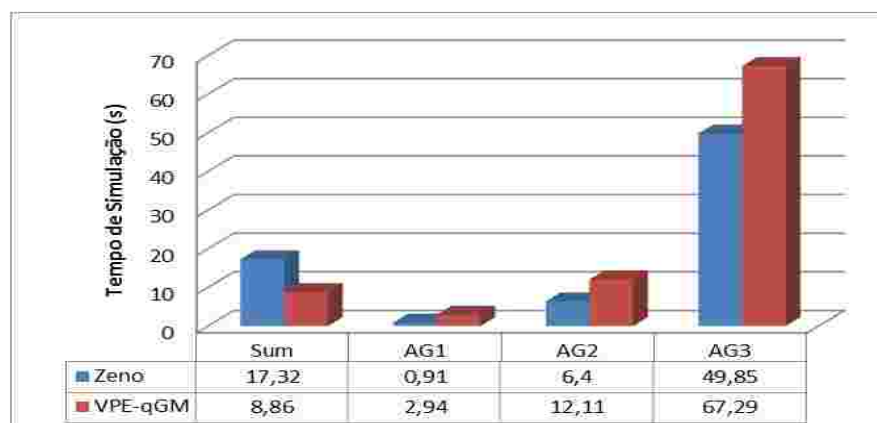


Figura 1 – Tempos obtidos para os estudos de caso

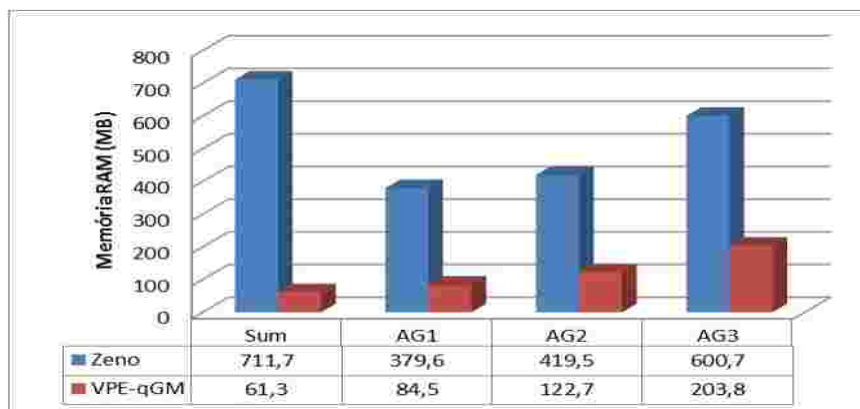


Figura 2 – Memória exigida nos experimentos.

O desvio padrão, no *VPE-qGM*, chegou a um máximo de 0,4(s), para o AG3. Para o *Zeno*, o desvio padrão máximo também foi obtido em AG3, com valor 2,11(s). Em uma primeira análise, percebe-se o melhor desempenho do simulador *Zeno* quando considerado o tempo de simulação. Essa diferença pode ser justificada pela melhor performance da linguagem Java (*Zeno*) frente a linguagem Python (*VPE-qGM*). O ambiente *VPE-qGM* mostra-se superior quando analisado o consumo de memória exigido para as simulações. Esse resultado provém do não armazenamento das estruturas de definição das transformações quânticas. Para as instâncias de AG, apesar da menor dimensão das transformações quânticas, o uso de memória mostrou-se maior do que a instância de *Sum*. Esse comportamento é justificado pela construção gráfica utilizada, a qual, para a modelagem adotada, requer uma quantidade elevada de componentes gráficos.

O ganho proveniente de nova implementação da biblioteca *qGM-Analyzer* é evidenciado quando analisado o uso de memória para o algoritmo *Sum*. Nesse cenário, para simuladores que fazem uso da operação de produto tensor, espera-se um elevado uso de memória, como observado no simulador *Zeno*. Por utilizar uma abordagem diferenciada, o *VPE-qGM* não apresentou esse comportamento, exigindo apenas 61,3 MB de memória RAM.

Os estados finais dos sistemas quânticos obtidos para ambos os ambientes mostraram-se iguais, validando os resultados obtidos pelo *VPE-qGM*.

A otimização da execução de *PEs* no *VPE-qGM* garante o suporte a sistemas quânticos com muitos *qubits*, uma vez que o ambiente não necessita de recursos de armazenamento exponencialmente maiores que o tamanho do sistema simulado. A nova abordagem da biblioteca *qGM-Analyzer* otimiza o acesso aos valores parciais que, até então, eram explicitamente armazenados em vetores, caracterizando um elevado custo de acesso.

4 CONCLUSÃO

Os resultados obtidos neste trabalho compreendem um avanço importante para suporte a algoritmos quânticos com muitos *qubits* no *VPE-qGM*. Além de contribuir com uma nova interpretação para definição das transformações quânticas, obteve-se uma redução significativa na complexidade de espaço do algoritmo implementado na biblioteca de execução do ambiente.

A continuidade do trabalho consiste na extensão da biblioteca de execução do ambiente. *VirD-GM* considerando as otimizações apresentadas neste trabalho, visando a otimização da simulação quântica distribuída. Busca-se, ainda, a otimização da representação da estrutura de memória associada ao espaço de

estados dos sistemas quânticos, evitando o crescimento exponencial e otimizando o acesso aos dados armazenados.

5 REFERÊNCIAS

- BARBOSA, Alexandre. **Um Simulador Simbólico de Circuitos Quânticos**. Universidade Federal de Campina Grande, 2007.
- MARON, Adriano; PINHEIRO, Anderson; REISER, Renata; PILLA, Maurício. Distributed Quantum Simulation on the VPE-qGM. In: **WSCAD-SSC**, 2010. p.128-135.
- MARON, Adriano; REISER, Renata; PILLA, Maurício. Distributed Quantum Simulation on the VPE-qGM. In: Anais **DINCON 2011**, 2011. p.1-6.
- MARON, Adriano; PINHEIRO, Anderson; REISER, Renata. Modelagem e Implementação da Medida Projetiva Quântica no Ambiente VPE-qGM. In: **CNMAC**, 2010. p. 1-2.
- HANNEKE, D; HOME, J. P.; JOST, J. D.; AMINI, J. M.; LEIBFRIED, D.; WINELAND, D. J. Realizations of a Programmable Two-Qubit Quantum Processor. **Nature Physics**, v. 6, p. 13-16, 2009.
- GUTIERREZ, Eladio; ROMERO, Sergio; TRENAS, Maria; ZAPATA, Emilio. Quantum Computer Simulation Using the CUDA Programming Model. **Computer Physics Communications**, p. 283-300, 2010.
- KNILL, Emanuel; NIELSEN, Michael. **Theory of quantum computation**. 2002.
- WATANABE, Hiroshi. **QCAD: GUI environment for Quantum Computer Simulator**. Super Computing Division, Information Technology Center, University of Tokyo, 2002.
- GIRARD, Jean-Yves. Linear Logic. **Theoretical Computer Science**, v. 50, p. 1-102, 1987.
- RAEDT, K. De; MICHIELSEN, K.; RAEDT, H. De; TRIEU, B.; ARNOLD, G.; RICHTER, M.; LIPPERT, T.; WATANABE, H.; ITO, N. Massive Parallel Quantum Computer Simulator. **Quantum Physics**, 2006.
- GROVER, Lov. A Fast Quantum Mechanical Algorithm for Database Search. In: **Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing**, p. 212-219, 1968.
- NIELSEN, Michael; CHUANG, Isaac. **Computação Quântica e Informação Quântica**, Bookman, 2003.
- QUANTIKI. **Quantum Information Wiki and Portal**, Singapore: Centre for Quantum Technology of the National University of Singapore, 2011.
- REISER, Renata; AMARAL, Rafael. The quantum states space in the qGM Model. In: **Anais Workshop Escola de Comp. e Inf. Quântica**, Petrópolis/RJ, 2011, p.92-101.
- MONZ, Thomas; SCHINDLER, Philipp; BARREIRO, Julio; CHWALLA, Michael; NIGG, Daniel; COISH, William; HARLANDER, Maximilian; HÄNSEL, Wolfgang; HENRICH, Markus; BLATT, Rainer. 14-Qubit Entanglement: Creation and Coherence. **Physical Review Letters**, v. 106, 2011.
- TAKAHASHI, Yasuhiro; TANI, Seiichiro; KUNIHURO, Noburo. Quantum Addition Circuits and Unbounded Fan-Out. **Quantum Physics**, 2009.