

# Consolidando uma Infraestrutura para Simulação Quântica Distribuída

Adriano Maron<sup>1</sup>, Anderson Pinheiro<sup>1</sup>, Renata Reiser<sup>2</sup>, Maurício Pilla<sup>2</sup>

<sup>1</sup>Centro Politécnico – Univ. Católica de Pelotas (UCPEL)

<sup>2</sup>Centro de Desenvolvimento Tecnológico – Univ. Federal de Pelotas (UFPEL)  
Pelotas – RS – Brasil

{maron.adriano,a.boettege}@gmail.com, {reiser, pilla}@inf.ufpel.edu.br

**Resumo.** *Este trabalho descreve uma infraestrutura para simulação distribuída de algoritmos quânticos, considerando a modelagem e simulação gráfica pelo ambiente VPE-qGM. A integração com o ambiente VirD-GM, o qual oferece suporte para o gerenciamento transparente da execução distribuída dos processos que representam o algoritmo, resulta em uma plataforma que mantém o foco no comportamento do sistema quântico, abstraindo detalhes de configuração inerentes à sistemas distribuídos. A validação da solução apresentada considera a simulação distribuída de sistemas quânticos compostos por até 12 qubits e transformações Hadamard.*

## 1. Introdução

A Computação Quântica (*CQ*) [Nielsen and Chuang 2003] é um paradigma computacional fundamentado nos postulados da Mecânica Quântica (*MQ*), a qual descreve matematicamente o comportamento de elementos em escala atômica/subatômica. A *CQ* prevê o desenvolvimento de computadores quânticos, os quais implementam algoritmos quânticos com melhor desempenho frente aos análogos clássicos.

Na *CQ*, o *qubit* é a unidade básica de informação, definido por um vetor de estado, unitário e bidimensional, genericamente descrito, na notação de Dirac [Nielsen and Chuang 2003], pela expressão  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ . Os coeficientes  $\alpha$  e  $\beta$  são números complexos correspondentes às amplitudes dos respectivos estados. O espaço de estados de um sistema quântico de múltiplos *qubits* é compreendido pelo produto tensorial do espaço de estados de seus sistemas componentes. Considerando um sistema quântico de dois *qubits*,  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$  e  $|\varphi\rangle = \gamma|0\rangle + \delta|1\rangle$ , o espaço de estados é composto pelo produto tensor  $|\psi\rangle \otimes |\varphi\rangle$ , ou seja  $\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$ .

A manipulação das probabilidades associadas aos estados dos *qubits* é feita por algoritmos quânticos [Nielsen and Chuang 2003], obtidos por composição de transformações unitárias, projeções e medidas, associadas a matrizes quadradas ortonormalizadas de ordem  $2^N$ , sendo  $N$  a quantidade de *qubits* da transformação.

## 2. Plataforma para Simulação Quântica Distribuída

A integração do *VPE-qGM* (*Visual Programming Environment for qGM Model*) [Maron et al. 2010] com o ambiente *VirD-GM* (*Virtual Distributed Geometric Machine Model*) [Fonseca et al. 2007] permite que uma aplicação quântica **simulada** no *VPE-qGM* usufrua das possibilidades de **execução** distribuída de *PEs*

disponibilizadas pelo *VirD-GM*, obtendo melhor desempenho frente à simulação sequencial. Essa integração considera o controle da simulação pela *interface qS*, gerenciando internamente a execução distribuída dos processos.

### 2.1. Descrição Arquitetural dos Ambientes

O ambiente *VPE-qGM* está em desenvolvimento visando o suporte a modelagem e simulação gráfica de algoritmos da *CQ*. Esse ambiente considera as abstrações do modelo *qGM* [Reiser and Amaral 2010] como semântica para interpretação de operações quânticas, diferenciando-o dos demais simuladores gráficos [Quantiki 2009], geralmente fundamentados no modelo de circuitos quânticos.

A arquitetura do ambiente é composta, basicamente, por três interfaces gráficas distintas: (i) Editor de Processos Quânticos (*qPE*), para modelagem das aplicações quânticas; (ii) Editor de Memórias Quânticas (*qME*), para configuração de estruturas de memória; (iii) Simulador Quântico (*qS*), no qual são realizadas as simulações a partir de arquivos descritores de processos e memória, gerados em suas respectivas *interfaces* de modelagem.

A execução distribuída de *PEs* pelo ambiente *VirD-GM* exige, como parâmetros de entrada, um arquivo descritor de processos, um arquivo descritor de memória e os endereços dos nodos de execução. O módulo *VirD-Loader* é responsável por interpretar os arquivos descritores de processos e memória exportados pelo *VPE-qGM*. O módulo *VirD-Launcher* contém métodos para escalonamento dos processos e controle do fluxo de execução das tarefas. Por fim, o módulo *VirD-Exec* controla o envio das tarefas para execução, criando os canais de comunicação com os clientes para execução nos *nodos*, denominados *VirD-Clients*.

### 2.2. Integração dos Ambientes

No ambiente *D-GM*, a simulação distribuída de um algoritmo quântico consiste na correspondente divisão em *M* aplicações independentes, executadas concorrentemente no *VirD-GM*. Nesse contexto, o módulo *VirD-GMConnection* gera os arquivos descritores necessários e configura/executa os comandos de disparo de aplicações.

No *VPE-qGM*, quando da simulação distribuída, a execução de uma macro implica na criação de *N* arquivos descritores de processos temporários (*tempProcessN.xml*), onde cada arquivo corresponde a uma aplicação composta por uma das construções iterativas paralelas associadas à macro. A seguir, o *qS* realiza *N* iterações para configuração e execução de um novo comando de disparo de aplicação no *VirD-GM*, considerando um arquivo descritor de processos (*tempProcess1.xml*, por exemplo), um arquivo descritor de memória (*tempMemory.xml*), contendo o estado de memória corrente da aplicação e a lista de *IPs* dos *VirD-Clients*. Ao final da execução de cada aplicação no *VirD-GM*, o arquivo *tempMemory.xml* é atualizado com dados de memória obtidos na simulação distribuída. Após executar todos os arquivos descritores de processos temporários associados à macro, o arquivo *tempMemory.xml* é lido pelo *qS*, e os dados são atualizados na *interface* gráfica do *qS*.

## 3. Estudo de Caso

A validação das implementações e a análise de desempenho da integração entre os ambientes *VPE-qGM* e *VirD-GM* se dá pela simulação distribuída de 3 sistemas

quânticos  $S1$ ,  $S2$  e  $S3$ , compostos por 10, 11 e 12 *qubits*, respectivamente, e transformações *Hadamard* aplicadas 3 vezes à todos os *qubits*. A metodologia dos testes considera 15 simulações de cada algoritmo em um *cluster* de PCs com as seguintes configurações principais: processador Intel Dual Core 1.6GHz, 1GB Ram, rede *Fast Ethernet* e sistema operacional Ubuntu 9.10. Em cada nodo estão instanciados 2 *VirD-Clients*, resultando da execução simultânea de 2 *PEs* em cada nodo.

Os tempos médios de simulação estão descritos na Tabela 1, na qual são considerados apenas os processadores dos nodos de execução – um nodo adicional é utilizado para controle e gerenciamento da simulação. Inicialmente, percebe-se o ganho de desempenho com o aumento da quantidade de nodos, porém a redução no tempo não é proporcional a quantidade de processadores adicionados.

**Tabela 1. Tempos para simulação distribuída da transformação Hadamard.**

-	1 Proc.	2 Procs.	4 Procs.	8 Procs.	16 Procs.
$S1$	122,32(s)	71,25(s)	47,37(s)	37,45(s)	34,98(s)
$S2$	390,92(s)	227,25(s)	147,25(s)	123,56(s)	121,32(s)
$S3$	1415,85(s)	711,62(s)	496,00(s)	440,61(s)	428,19(s)

A Tabela 2 relaciona algumas informações adicionais obtidas durante a simulação de  $S3$ , permitindo algumas interpretações acerca do desempenho obtido.

**Tabela 2. Informações adicionais da simulação de  $S3$ .**

-	Dados Env. (Base)	Dados Receb. (Nodos)	Ocupação dos Nodos
1 Proc.	1,5 - 1,7 MB/s	1,5 - 1,7 MB/s	50%
2 Procs.	3,0 - 3,2 MB/s	3,0 - 3,2 MB/s	85 - 90%
4 Procs.	4,0 - 4,5 MB/s	2,1 MB/s	70 - 75%
8 Procs.	5,0 - 5,5 MB/s	1,2 - 1,5 MB/s	50 - 55%
16 Procs.	5,0 - 5,5 MB/s	600 - 700 KB/s	25 - 30%

Primeiramente, constata-se que a inserção de novos nodos acarreta em uma redução significativa na taxa de ocupação do *cluster*, degradando o desempenho. Esse comportamento pode ser justificado por duas características principais do *cluster*: rede de interconexão e configuração da máquina-base.

Considerando uma relação entre a quantidade de dados enviados pela base ( $DE$ ) e a quantidade de dados recebidos pelos nodos ( $DR$ ) nas diferentes configurações de *cluster*, podem ser obtidos os valores esperados para o tráfego de dados na rede. Sendo,  $DE$  e  $DR$  com valor aproximado de 3,2MB/s, quando utilizado um nodo com 2 *VirD-Clients* (2 Procs.), espera-se que, para a execução simultânea de 16 *VirD-Clients* (8 nodos), o valor de  $DE$  aproxime-se de 25,6MB/s, enquanto que a taxa de  $DR$  permanece constante em 3,2MB/s. Analisando a Tabela 2, nota-se que os valores de  $DE$  não ultrapassam 5,5MB/s, e a taxa de  $DR$  diminui com o aumento da quantidade de processadores. Assim, os nodos necessitam de uma parcela maior de tempo para receber a quantidade total de dados necessários para executar a computação, aumentando o tempo de comunicação com os nodos e reduzindo o *speedup* das aplicações executadas sobre muitos processadores.

#### 4. Conclusão

A integração entre *VPE-qGM* e *VirD-GM* está concluída, sendo realizados testes para validação do módulo *VirD-GMConnection*. Simulações sequenciais de *S1*, *S2* e *S3*, considerando apenas o ambiente *VPE-qGM* (sem o *VirD-GM* como middleware), resultaram nos seguintes tempos de simulação, respectivamente: 48,790s, 194,929s e 791,299s. A simulação distribuída dessas aplicações apresentou ganho em comparação à simulação sequencial, obtendo bom *speedup* para até 4 processadores, porém a escalabilidade é limitada, a princípio, pela rede de interconexão.

Em comparação com outros simuladores quânticos paralelos [Niwa et al. 2002, Raedt et al. 2006, Gutierrez et al. 2010], nota-se significativa diferença na dimensão dos sistemas quânticos suportados, visto que sistemas com até 31 *qubits* já foram simulados. Porém, nesses casos, usualmente são considerados supercomputadores ou arquiteturas diferenciadas para execução dos cálculos. O diferencial deste trabalho é a simulação distribuída a partir de um conjunto de máquinas usualmente disponíveis em laboratórios de pesquisa e ensino, exigindo pouca configuração do sistema e viabilizando a modelagem gráfica das aplicações.

Para obtenção de um *speedup* adequado a quantidade de processadores disponíveis no *cluster*, duas soluções iniciais podem ser adotadas: (1) utilização de uma rede *gigabit Ethernet*, proporcionando maior vazão de dados; (2) reduzir a quantidade de dados enviados aos *VirD-Clients*, o que pode ser obtido pela integração do *VPE-qGM* com o ambiente *VirD-GM* que implementa o módulo *ShareD-GM*, o qual compartilha o objeto de memória através do sistema DSM Terracotta.

#### Referências

- Fonseca, V., Reiser, R., Yamin, A., and Pilla, P. (2007). *VirD-gm: Towards a grid computing environment*. In *Proceedings of CCGRID 2007*, pages 1–6.
- Gutierrez, E., Romero, S., Trenas, M., and Zapata, E. (2010). Quantum computer simulation using the cuda programming model. *Computer Physics Communications*, pages 283–300.
- Maron, A., Pinheiro, A., Reiser, R., Yamin, A., and Pilla, M. (2010). Ambiente *vpe-qgm*: Em direção a uma nova abordagem para simulações quânticas. *Revista do CCEI*, 14:29–46.
- Nielsen, M. A. and Chuang, I. L. (2003). *Computação Quântica e Informação Quântica*. Bookman.
- Niwa, J., Matsumoto, K., and Imai, H. (2002). General-purpose parallel simulator for quantum computing. *Lecture Notes in Computer Science*, 2509:230–249.
- Quantiki (2009). Quantum information wiki and portal. Disponível por WWW em <<http://www.quantiki.org>> (jun.2010).
- Raedt, K. D., Michielsen, K., Raedt, H. D., Trieu, B., Arnold, G., Richter, M., Lippert, T., Watanabe, H., and Ito, N. (2006). Massive parallel quantum computer simulator. <http://arxiv.org/abs/quant-ph/0608239>.
- Reiser, R. and Amaral, R. (2010). The quantum states space in the qgm model. In *Anais do III WECCIQ*, pages 92–101, Petrópolis, RJ.