

## Extensões do VirD-GM para suporte a simulação quântica distribuída no Ambiente VPE-qGM

de AVILA, Anderson<sup>1</sup>; MARON, Adriano<sup>2</sup>; REISER, Renata<sup>3</sup>; PILLA, Mauricio<sup>3</sup>

<sup>1</sup>UFPEL, Graduação em Ciência da Computação, Pelotas abdavila@inf.ufpel.edu.br;

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

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

### 1 INTRODUÇÃO

Pela exploração dos fenômenos da Mecânica Quântica (MQ) (KNILL, 2002), o desenvolvimento de algoritmos quânticos tem-se mostrado, em vários cenários, exponencialmente mais rápidos que seus análogos clássicos (GROVER, 1996), (SHOR, 1997), (GISIN et. al., 2002). O *qubit* é a unidade básica de informação Computação Quântica (CQ), definido por um vetor de estado, unitário e bidimensional, genericamente descrito, na notação de Dirac (NIELSEN, 2003), pela expressão  $\alpha|0\rangle + \beta|1\rangle$ . Os coeficientes  $\alpha$  e  $\beta$ , são números complexos correspondendo às amplitudes dos respectivos estados básicos, respeitando a condição de normalização  $|\alpha|^2 + |\beta|^2 = 1$ . Tal construção garante o vetor unitário e ortonormalizado, representado por  $(\alpha, \beta)^t$ . As amplitudes permitem que o sistema represente, simultaneamente, estados básicos distintos, configurando um estado de superposição quântica, caracterizando o fenômeno do paralelismo quântico.

Entretanto, tais algoritmos podem ser eficientemente executados apenas sobre computadores quânticos, os quais estão em fase de desenvolvimento, nem sempre disponíveis, e ainda, sem suporte para sistemas mais complexos. Neste contexto, a simulação de algoritmos em computadores clássicos viabiliza o desenvolvimento e teste de algoritmos quânticos, antecipando o conhecimento acerca de seu comportamento quando da execução sobre um hardware quântico. A simulação de sistemas quânticos através de computadores clássicos ainda consiste em um desafio de pesquisa, justificando o estudo de soluções voltadas para a simplificação no processo de modelagem e interpretação de algoritmos quânticos. Mais significativa, a busca por otimizações voltadas para o ganho de desempenho da simulação contribui para o incrementar o suporte a sistemas quânticos complexos.

Fundamentado no modelo qGM, a interpretação de QPs (*Quantum Processes*) e QPPs (*Quantum Partial Processes*) é obtida pela aplicação de processos que contém informação parcial acerca de determinados conjuntos de vetores associados a sua matriz de representação. Tais estruturas de dados são implementadas no VPE-qGM (*Visual Programming Environment for the Quantum Geometric Machine Model*) (MARON et. Al., 2010) para modelagem e simulação, sequencial e distribuída de algoritmos quânticos, modelando a evolução dos estados dos sistemas quânticos a partir de um conjunto de interfaces gráficas.

Este trabalho está focado no ambiente VirD-qGM, objetivando otimizar o gerenciamento da execução distribuída no VPE-qGM. Considerando o alto custo computacional inerente à simulação de algoritmos quânticos em computadores clássicos, a motivação deste trabalho contempla a qualificação do ambiente VPE-qGM, para suporte a simulação de algoritmos quânticos com maior complexidade de forma distribuída. A principal contribuição deste trabalho consiste na extensão das capacidades de simulação distribuída do VirD-GM pela implementação dos conceitos de QPs e QPPs. Esses esforços objetivam reduzir o tempo de simulação de transformações quânticas pela distribuição das computações ao longo de clientes instanciados em um cluster de computadores.

### 2 METODOLOGIA (MATERIAL E MÉTODOS)

Para validação e análise de desempenho das extensões desenvolvidas, são considerados estudos de caso compostos por transformações quânticas *Hadamard* de até 17 *qubits*.

A metodologia de desenvolvimento dos testes contempla, para cada estudo de caso, a realização de 15 simulações sobre diferentes quantidades de clientes de execução. O número de QPPs utilizados são definidos pelo número de clientes utilizados na simulação. Foram utilizadas duas máquinas executando 4 (nodos) *VirD-Client* sem cada máquina. A seguinte configuração de *hardware* foi usada: processador Intel(R) Xeon(R) CPU E5620 @2.40GHz, 4GB RAM, SO Ubuntu 11.04 e rede Ethernet. Uma máquina adicional instancia o servidor do VirD-GM.

### 3 RESULTADOS E DISCUSSÃO

A execução distribuída de aplicações a partir do 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. O módulo VirD-Launcher contém os métodos para escalonamento dos processos e controle do fluxo de execução das tarefas. Assim, o módulo VirD-Exec controla o envio das tarefas para execução, criando os canais de comunicação com os clientes para execução nos VirD-Clients.

A computação de cada QPP, compreende: (i) geração dos valores não nulos associados a  $2^q/p$  vetores componentes da matriz de definição da transformação quântica modelada, sendo  $q$  a quantidade de *qubits* do sistema e  $p$  a quantidade de VirD-Clients utilizados; (ii) multiplicação desses valores pelas amplitudes obtidas da estrutura de memória que modela o espaço de estados do sistema quântico.

A simulação distribuída de um algoritmo quântico a partir de QPs consiste no envio de cada QPP para um VirD-Client através do método `send`, passando os parâmetros do QPP e a memória atual. Então, o VirD-Client realiza a computação e gera uma lista com as posições alteradas e seus novos valores, a qual é enviada para o VirDServer através do método `updateMemory`, e o VirDServer atualiza a sua memória com os dados contidos nesta lista. Este processo é feito para todos os arquivos temporários gerados na simulação, sendo necessário um arquivo para cada passo da simulação. Após todos os passos terem sido executados, o método `exportResult` é chamado, o qual é responsável por salvar o resultado obtido no arquivo `tempMemory.xml` que será utilizado posteriormente pelo VPE-qGM para carregar o resultado da simulação. Nesse contexto, o módulo `VirD-GMConnection` gera os arquivos descritores necessários, configurando e/ou executando os comandos de disparo de aplicações. A estrutura que representa a parte de execução do ambiente VirD-GM pode ser vista na figura 1.

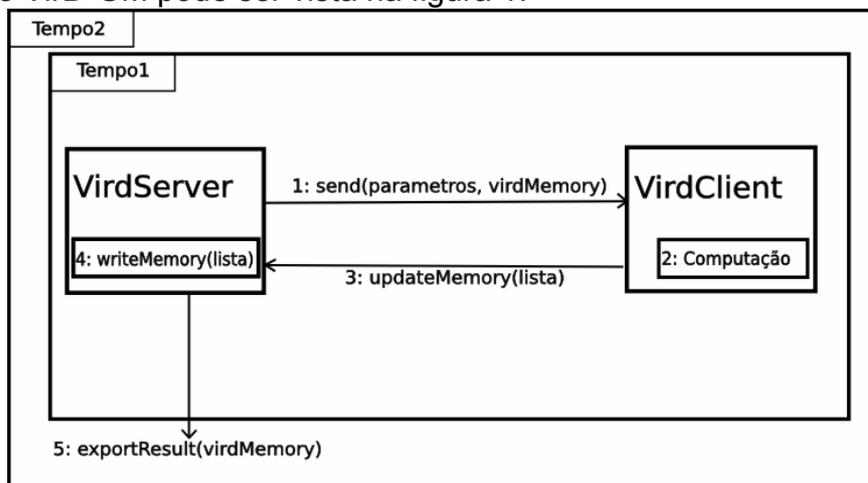


Figura 1: Estrutura de execução do ambiente VirD-GM.

Os tempos de simulação, incluindo a quantidade de vetores componentes associados a cada QPP para cada estudo de caso e a correspondente quantidade de valores gerados, estão descritos na Tabela 1. Salienta-se que tempo referente ao método `exportResult` não foi considerado. Na figura 2, podem ser vistos os *speedups* obtidos nas simulações.

Tabela 1: Tempos de simulação.

Operação	1 VirD-Client		2 VirD-Client		4 VirD-Client		8 VirD-Client	
	Nº de valores	Tempo (s)						
H <sup>14</sup>	268435456	36061	134217728	19293	67108864	11436	33554432	7179
H <sup>15</sup>	1073741824	138338	536870912	72824	268435456	41069	134217728	24384
H <sup>16</sup>	4294967296	1126529	2147483648	573611	1073741824	304008	536870912	160049
H <sup>17</sup>	17179869184	3075729	8589934592	1632369	4294967296	934069	2147483648	509275

Logo a seguir, tem-se os resultados da simulação.



Figura 2: Speedup das simulações dos algoritmos quânticos

## 4 CONCLUSÃO

A possibilidade de modelagem de transformações quânticas a partir de sincronizações de QPPs viabiliza o particionamento (granularidade) da computação que modela a transição de estado em um sistema quântico. O suporte a execução paralela e distribuída dos QPPs foi viabilizada a partir das extensões dos módulos do ambiente VirD-GM, de acordo com o apresentado neste trabalho.

Os resultados mostram ganho de desempenho conforme o aumento do número de clientes, mas para isto ocorrer, é necessário que o número de clientes no cluster não ultrapasse seu número de processadores.

Na continuidade do trabalho, tem-se a implementação e simulação distribuída de algoritmos quânticos compostos de transformações controladas e operações de medidas, analisando seu correspondente.

## 5 REFERÊNCIAS

MARON, Adriano; PINHEIRO, Anderson; REISER, Renata; PILLA, Maurício. Distributed Quantum Simulation on the VPE-qGM. In: **WSCAD-SSC**, 2010. p.128-135.

KNILL, Emanuel; NIELSEN, Michael. **Theory of quantum computation**. 2002.

NIELSEN, Michael; CHUANG, Isaac. **Computação Quântica e Informação Quântica**. Local de Edição: Bookman, 2003.

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, 1998.

SHOR, P. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In: **SIAM Journal on Computing**, 1997.

GISIN, N.; RIBORDY, G. and TITTEL, W.; ZBINDEN, H.. Quantum cryptography. In: **Reviews of Modern Physics**, p. 145-195, 2002.