

Estendendo o Ambiente VirD-GM para Execução Distribuída de Processos Quânticos

Anderson B. de Avila, Adriano K. Maron,
Maurício L. Pilla, Renata S. Reiser

¹Centro de Desenvolvimento Tecnológico – Universidade Federal de Pelotas (UFPEL)
Caixa Postal 15.064 – 91.501-970 – Pelotas – RS – Brasil

{abdvila, akmaron, pilla, reiser}@inf.ufpel.edu.br

Resumo. A principal contribuição deste trabalho consiste na extensão do ambiente VirD-GM, para que suporte a simulação distribuída de algoritmos quânticos descritos a partir de Processos Quânticos e Processos Quânticos Parciais, os quais requerem menor tempo de execução. A validação e análise de desempenho se dá pela simulação de Hadamard de até 17 qubits, obtendo resultados que comprovam o ganho de desempenho frente a simulação sequencial.

1. Introdução

A simulação de algoritmos quânticos 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 se mostra um desafio de pesquisa em aberto, justificando o estudo de soluções voltadas para a simplificação no processo de modelagem e interpretação de algoritmos quânticos [Nielsen and Chuang 2003]. Mais significante, as otimizações no ganho de desempenho da simulação contribuem para o suporte a sistemas quânticos mais complexos.

Encontram-se em desenvolvimento os ambientes (i) *VPE-qGM* (*Visual Programming Environment for the Quantum Geometric Machine Model*) [Maron et al. 2010] - que visa auxiliar na modelagem e simulação, sequencial e distribuída de algoritmos quânticos, apresentando as construções e a evolução dos estados dos sistemas quânticos a partir de um conjunto de *interfaces* gráficas; e (ii) *VirD-GM* [Fonseca et al. 2007] (*Virtual Distributed Geometric Machine Environment*)- cujo principal objetivo é realizar um gerenciamento transparente 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.

Neste sentido, a principal contribuição deste trabalho consiste na **extensão das capacidades de simulação distribuída do VirD-GM pela implementação dos conceitos de Processos Quânticos (QPs) e Processos Quânticos Parciais (QPPs)**. Os quais contribuem com a possibilidade de estabelecer interpretações parciais de uma transformação quântica, permitindo que alguns parâmetros/processos sejam desconhecidos, uma vez que, em um **contexto local** da computação de cada *QPP*, é possível gerar apenas conjuntos restritos de elementos associados a matriz de definição da transformação quântica aplicada. *QPPs* complementares (que interpretam conjuntos disjuntos de linhas) podem

ser sincronizados e executados de forma independente (localmente ou em diferentes nodos de processamento). Esses esforços tem por objetivo reduzir o tempo de simulação de transformações quânticas pela distribuição das computações ao longo de clientes instanciados em um *cluster* de computadores.

Este artigo está estruturado da seguinte forma: na Seção 2 esta descrito o funcionamento do ambiente *VirD-GM*, bem como as implementações realizadas. Os resultados obtidos na simulação distribuída dos processos quânticos é apresentada na Seção 3. Por fim, a Seção 4 resume as principais contribuições do trabalho e os resultados obtidos, indicando a perspectiva de continuidade do trabalho.

2. Extensão do ambiente *VirD-GM*

O ambiente *VirD-GM* encontra-se em desenvolvimento com o objetivo de realizar um gerenciamento transparente da execução distribuída no ambiente *VPE-qGM*.

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 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*.

A possibilidade de modelar transformações quânticas a partir da sincronização de *QPPs* exige a extensão dos três módulos principais do ambiente *VirD-GM*, visando o suporte à simulação distribuída das novas construções. Dessa forma, algumas funcionalidades são adicionadas/estendidas.

- (i) As alterações no módulo *VirD-Loader* viabilizaram o reconhecimento de um *QP* ou *QPP* quando da interpretação de um arquivo de processos.
- (ii) A biblioteca *QGMAnalyzer*, responsável pela execução dos processos, foi alterada para que pudesse receber e computar processos quânticos.
- (iii) E a forma de comunicação foi alterado para suportar a utilização de *QPPs*.

A estrutura deste ambiente pode ser vista na Fig. 1.

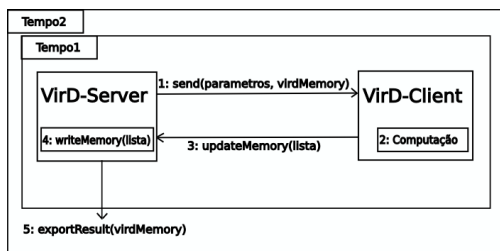


Figure 1. Estrutura do ambiente *VirD-GM*

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 *VirD-Server* através do método *updateMemory*. E assim, o *VirD-Server* 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 e configura/executa os comandos de disparo de aplicações.

3. Resultados

Para validação e análise de desempenho das extensões desenvolvidas, são considerados estudos de caso compostos por transformações quânticas *Hadamard* de até 17 *qubits*, pois é a transformação que apresenta maior carga de execução no ambiente de simulação.

A metodologia dos testes utilizada contempla, para cada estudo de caso, a realização de 15 simulações sobre diferentes quantidades de clientes de execução. O número de *QPPs* utilizados são equivalentes ao número de clientes utilizados na simulação. Foram utilizadas duas máquinas, executando 4 *VirD-Clients* cada, com a seguinte configuração: processador Intel(R) Xeon(R) CPU E5620 @2.40GHz, 4GB RAM e rede Ethernet. Uma máquina adicional instancia o servidor do *VirD-GM*. Essas máquinas são conectadas através de um *switch* em uma rede Fast Ethernet.

Os tempos de simulação, a quantidade de vetores componentes associados a cada *QPP* para cada estudo de caso e a correspondente quantidade de valores gerados estão descritos nas tabelas abaixo, e pode ser observado que o aumento do número de clientes acarretou diminuição do tempo de execução para cada estudo de caso, já que os valores a serem calculados por cada cliente diminui. O *speedup* obtido nas simulações pode ser visto na Figura 2.

Table 1. Tempos de simulação

Op	1 <i>VirD-Client</i>		2 <i>VirD-Clients</i>	
	Valores	Tempo (s)	Valores	Tempo (s)
H^{14}	$0,2 \times 10^9$	36,061	$0,1 \times 10^9$	19,293
H^{15}	$0,1 \times 10^{10}$	138,338	$0,5 \times 10^9$	72,824
H^{16}	$0,4 \times 10^{10}$	1126,529	$0,2 \times 10^{10}$	573,611
H^{17}	$0,1 \times 10^{11}$	3075,729	$0,8 \times 10^{10}$	1632,369

Op	4 <i>VirD-Clients</i>		8 <i>VirD-Clients</i>	
	Valores	Tempo (s)	Valores	Tempo (s)
H^{14}	$0,6 \times 10^8$	11,436	$0,3 \times 10^8$	7,179
H^{15}	$0,2 \times 10^9$	41,069	$0,1 \times 10^9$	24,384
H^{16}	$0,1 \times 10^9$	304,008	$0,5 \times 10^8$	160,049
H^{17}	$0,4 \times 10^{10}$	934,069	$0,2 \times 10^{10}$	509,275



Figure 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 da computação, modelando uma transição de estado em um sistema quântico. O suporte a execução paralela e distribuída dos $QPPs$ foi viabilizado a partir das extensões dos módulos do ambiente *VirD-GM*, previamente apresentadas neste trabalho.

Os resultados obtidos 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 o desempenho das implementações sob diversos cenários.

References

- 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.
- Gisin, N., Ribordy, G., Tittel, W., and Zbinden, H. (2002). Quantum cryptography. *Reviews of Modern Physics*, pages 145–195.
- Grover, L. (1996). A fast quantum mechanical algorithm for database search. *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pages 212–219. Disponível por WWW em <http://doi.acm.org/10.1145/237814.237866> (dec.2011).
- 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.
- Shor, P. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*.