

# GPU-Aware: Simulação Quântica Distribuída via VirD-GM

**Anderson Avila — Murilo Schmalfuss**  
**Renata Reiser — Maurício Pilla**

Universidade Federal de Pelotas  
Mestrado em Computação  
Centro de Desenvolvimento Tecnológico

ERAD - 2015 - GRAMADO/RS

# Sumário

- 1 Introdução
- 2 Contribuição
- 3 Principais Resultados
- 4 Considerações Finais

# Introdução

## Computação Quântica

- Impõe novas fronteiras com relação à velocidade das computações utilizando-se dos fenômenos da mecânica quântica.
- Pode ser utilizada em algoritmos quânticos de busca, fatoração, criptografia, etc.

## Dificuldades

- Simplicidade do hardware quântico atual.
- Criação de algoritmos quânticos.

## Soluções Viáveis

- Simulação (hardware / **software**) de algoritmos quânticos.

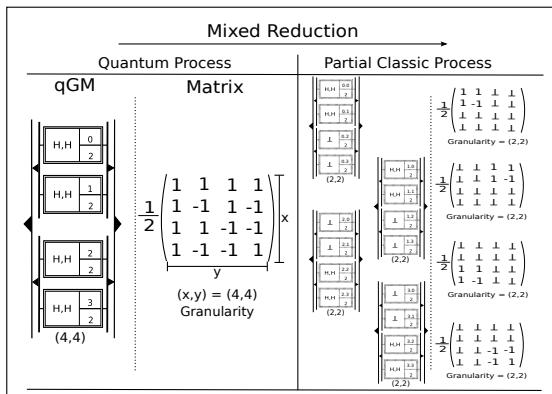
## VPE-qGM: Visual Programming Environment for the qGM Model

- Concebido segundo os fundamentos do modelo qGM: Quantum Geometric Machine Model.
- Visa a modelagem e a simulação distribuída de algoritmos da CQ.
- Abordagem de processos computacionais.

## VirD-GM: Virtual Distributed Geometric Machine Environment

- Ambiente de execução distribuída de algoritmos da CQ.
- Integrado ao ambiente de execução VPE-qGM.
- Realiza um gerenciamento transparente da execução distribuída quando solicitada pelo ambiente *VPE-qGM*.

- *MPPs* (Mixed Partial Processes) contribuem com a interpretação parcial de uma transformação quântica.
- Assim o programador tem maior controle do uso da memória e do gerenciamento dos cálculos (granulidade das computações) considerando os recursos disponíveis.



# Objetivos

- Extensão das capacidades de simulação distribuída do *VirD-GM* para suportar a execução de **Processos Mistos Parciais (MPPs)** via GPUs.



## Mudanças Realizadas

- Reestruturação do arquivo que descreve os processos, modificado para interpretação e reconhecimento de um MPP.
- Reestruturação do módulo Vird-Launcher, responsável pela comunicação, enviando para um VirD-Client, cliente de execução, o MPP e somente a porção de memória que ele necessita para a simulação.
- Extensão da biblioteca QGM-Analyzer, responsável pela maior parte da computação, viabilizando invocar o kernel CUDA quando solicitado pelo gerenciamento a execução da computação na GPU.
- Alteração do kernel CUDA para suportar execuções de MPPs.

# Envio/Recebimento de Dados

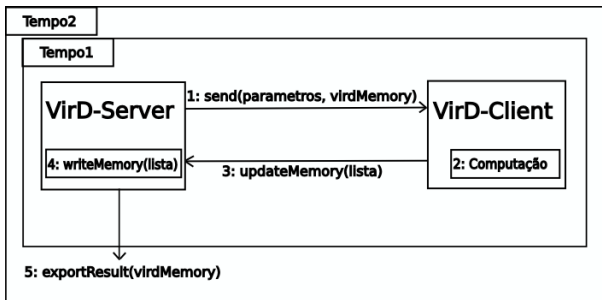


Figura: Estrutura do gerenciamento de dados da VirD-GM

# Metodologia

## Metodologia:

- Simulação de transformações quânticas *Hadamard* de até 21 *qubits* ( $H^{\otimes 18}$ ,  $H^{\otimes 19}$ ,  $H^{\otimes 20}$  e  $H^{\otimes 21}$ );
- 10 simulações de cada instância das tranformações considerando cada uma das configurações de distribuição dentro do cluster de *GPUs*;
- Medido tempo de simulação;
- Comparação entre a simulação com configuração 1 – 1 e as demais.

## Hardware:

- 2 Desktops: Intel Core i7, 8GB de RAM, GPU NVIDIA GT 640.
- 2 Desktops: Intel Core i7, 8GB de RAM, GPU NVIDIA GTX 560.

## Software:

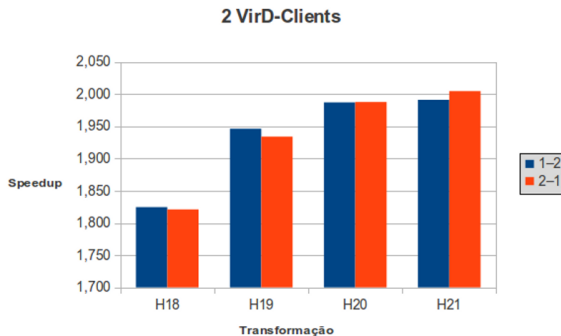
- Ubuntu 12.04 64 *bits*, Jcuda 0.5.0a, NVIDIA CUDA TOOLKIT 5.0.

# Tempos de Simulação

**Tabela:** Tempos médios de simulação obtidos, medidos em segundos.

|          | <i>1 VirD-Client</i> | <i>2 VirD-Clients</i> |        | <i>4 VirD-Clients</i> |        |        |
|----------|----------------------|-----------------------|--------|-----------------------|--------|--------|
|          | 1 – 1                | 1 – 2                 | 2 – 1  | 2 – 2                 | 1 – 4  | 4 – 1  |
| $H^{18}$ | 12,88                | 7,06                  | 7,07   | 4,20                  | 4,16   | 4,25   |
| $H^{19}$ | 47,99                | 24,66                 | 24,81  | 13,22                 | 13,14  | 13,39  |
| $H^{20}$ | 187,80               | 94,51                 | 94,48  | 48,49                 | 48,30  | 48,87  |
| $H^{21}$ | 744,06               | 373,69                | 371,17 | 188,79                | 187,93 | 188,41 |

# Speedup para 2 VirD-Clients



# Speedup para 4 VirD-Clients



# Atividades em andamento

- Criação de um novo algoritmo de execução mais eficiente.
- Suporte para portas controladas, projeções e operações de medida na abordagem distribuída;
- Concepção e implementação do ambiente de execução híbrido, em que o cálculo será executado por CPUs e GPUs de forma distribuída.

# Conclusão

- A principal contribuição deste trabalho é prover uma solução para o problema da alta complexidade espacial e temporal da simulação de algoritmos quânticos através da simulação distribuída/paralela usando clusters de GPUs e modelando as transformações por MPPs.
- Os resultados mostraram que é possível obter ganho de desempenho com o aumento do número de clientes, e para transformações mais pesadas este ganho é maior.
- Como os tempos de simulação com diferentes configurações para uma mesma quantidade de VirD-Clients são semelhantes, significa que é possível dividir as computações de acordo com os recursos disponíveis sem que haja com perda de desempenho.

# Agradecimento e Perguntas

## Agradecimentos

- FAPERGS (PqG 002/2014, processo 309533/2013-9,)
- MCTI/CNPQ (Universal processo 448766/2014-0)

## Dúvidas

Contato:

- [abdavila@inf.ufpel.edu.br](mailto:abdavila@inf.ufpel.edu.br)

# REFERÊNCIAS

- AVILA, A.; MARON, A.; REISER, R.; PILLA, M. GPU-aware distributed quantum simulation. In Proceedings of 29th Symposium On Applied Computing, pages 1-6, 2014.
- AVILA, A.; Schmalfuss, M.; Maron, A.; Reiser, R; Pilla, M. Simulação distribuída de algoritmos quânticos via GPUs. In Proceedings of the XV WSCAD, pages 1-12, 2014.
- Nielsen, M. A. e Chuang, I. L. (2000). Quantum Computation and Quantum Information. Cambridge University Press.

# GPU-Aware: Simulação Quântica Distribuída via VirD-GM

**Anderson Avila — Murilo Schmalfuss**  
**Renata Reiser — Maurício Pilla**

Universidade Federal de Pelotas  
Mestrado em Computação  
Centro de Desenvolvimento Tecnológico

ERAD - 2015 - GRAMADO/RS