Extensão do Ambiente VirD-GM para Suporte à Processos Parciais para Portas Controladas

Murilo F. Schmalfuss[1], Anderson B. Avila[1], Renata H. S. Reiser[1], Maurício L. Pilla[1]

[1]Centro de Desenvolvimento Teconológico – Universidade Federal de Pelotas (UFPel)
Caixa Postal 354 – 96.010-900 – Pelotas – RS – Brazil

{mfschmalfuss,abdavila,reiser,pilla}@inf.ufpel.edu.br

Resumo. *Este artigo descreve a proposta de extensão do ambiente VirD-GM para suporte à portas controladas. As portas controladas estão presentes na grande maioria dos algoritmos quânticos desenvolvidos e portanto de grande importância para as simulações quânticas. A abordagem de processos parciais e paralelismo massivo em conjunto com a computação distribuída proporciona uma redução no tráfego de dados entre clientes e servidor e consequentemente uma aceleração no desempenho dos tempos de simulação.*

1. Introdução

A Computação Quântica (*CQ*), explorando os conceitos da mecânica quântica, oferece um novo paradigma de computação, onde os algoritmos quânticos podem ser muito mais rápidos que seus análogos clássicos[Deutsch 1985][Grover 1996][Shor 1997]. Porém a indisponibilidade de *hardware* quântico para a modelagem e execução de algoritmos é atualmente um dos principais desafios da *CQ*. A solução mais viável é a simulação destes algoritmos em computadores clássicos. Entretanto, salienta-se que esta abordagem possui restrições, o *hardware* clássico não oferece suporte aos fenômenos da mecânica quântica e estes precisam ser implementados em *software*. Estas implementações normalmente ficam limitadas a poucos *qubits* devido a complexidade temporal e espacial associada aos algoritmos quânticos. Simuladores utilizando computação paralela e distribuída, placas gráficas (*GPUs*) e novas interpretações para transformações quânticas têm sido estudados para acelarar as simulações [Viamontes et al. 2003] [Gutiérrez et al. 2010]. Este trabalho propõe uma extensão do ambiente de simulação distribuído *Virtual Distributed Geometric Machine* (*VirD-GM*) para a abordagem de processos parciais em transformações quânticas controladas visando implementação distribuída/paralela e correspondente execução sobre clusters de *GPUs*.

Em sua organização, tem-se na seção 2 os conceitos básicos da *CQ*, o funcionamento do ambiente *VirD-GM* está reportado na seção 3. Na seção 4 é apresentada a metodologia. Por fim, na seção 5 são feitas as considerações finais do trabalho.

2. Conceitos Básicos

Nesta seção serão apresentados os conceitos básicos para o entendimento da mecânica quântica e da *CQ*.

2.1. Qubit: O Bit Quântico

A unidade básica da *CQ* é o *qubit*. Os estados possiveis do *qubit* são os estados básicos $|0\rangle$ e $|1\rangle$, com suas respectivas representações na *notação de Dirac* [Dirac 1939] e na notação

matricial representada na eq. (1), e também uma combinação linear desses estados, chamado de *superposição quântica*, representada na eq. (2).

$$|0\rangle \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad |1\rangle \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad (1) \qquad\qquad |\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad (2)$$

Os coeficientes α e β são números complexos, desta forma podemos descrever o *qubit* como um vetor em um espaço vetorial complexo bidimensional. Não podemos analisar o *qubit* para determinar seu estado quântico, ou seja, os valores α e β. A mecânica quântica nos diz que podemos apenas obter uma informação mais restrita, utilizando uma operação de medida podemos obter o estado clássico 0 com a probabilidade $|\alpha|^2$ e o estado clássico 1 com a probabilidade $|\beta|^2$ [Nielsen and Chuang 2011].

2.2. Portas Quânticas

Algumas das principais portas quânticas de um *qubit* serão detalhadas na eq. (3). A porta *Hadamard* (*H*) é o operador que coloca os *qubits* em estado de superposição quântica. A porta *Pauli X* (*X*), é um operador *not*, ela inverte o estado do *qubit* à qual ela é aplicada.

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad\qquad X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad (3)$$

As portas da eq. (4) são classificadas como portas quânticas controladas. Para que a porta atue sobre um *qubit* ela precisa satisfazer a condição do controle. No caso da porta $CNOT$, para que o *qubit* alvo do circuito seja invertido, o *qubit* de controle precisa estar em 1. A porta *Toffoli* é uma porta quântica controlada de três *qubits*, sendo dois controles e um alvo.

$$CNOT \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad Toffoli \equiv \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad (4)$$

As portas controladas são de extrema importância para a *CQ*, um exemplo é o algoritmo de Shor [Shor 1997], onde portas controladas são utilizadas para a construção de um algoritmo quântico para a fatoração de números primos, utilizado em criptografia. O algoritmo de Shor possui tempo polinomial em *n*, onde *n* é o número de digitos do inteiro a ser fatorado.

Transformações quânticas aplicadas simultaneamente à diferentes *qubits* são geradas pelo produto tensor entre as portas quânticas correspondentes, mostrado na eq. (5).

$$H^{\otimes^2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \qquad (5)$$

3. VirD-GM

O ambiente *VirD-GM* [Avila et al. 2014] é responsável pela distribuição da computação entre os clientes. Ele gerencia a execução distribuída lançada pelo ambiente *VPE-qGM* de forma transparente. A simulação distribuída dos algoritmos quânticos na *VirD-GM* é controlada por três módulos:

- (i) *VirD-Loader*: é o módulo responsável pela interpretação dos arquivos descritores que contém o algoritmo quântico a ser simulado e também o seu vetor de estado inicial;
- (ii) *VirD-Launcher*: gerencia o escalonamento e o controle de fluxo;
- (iii) *VirD-Exec*: controla a comunicação e a transferência de dados entre os clientes.

Melhorias para o suporte ao paralelismo massivo utilizando *GPUs* [Maron et al. 2013] e utilizações de processos parciais [Maron et al. 2012] com portas não controladas foram feitas para otimizar as simulações quânticas.

4. Metodologia

O conceito de processos parciais nos permite transferir apenas os dados necessários para uma parte da computação. O resultado gerado será apenas um resultado parcial. A eq. (6) exemplifica como é dividida a matriz utilizando dois alvos em X no primeiro e terceiro *qubit* e o controle no segundo *qubit*.

$$X^{\otimes 2} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad Contr. = \begin{array}{c} 000 \\ 001 \\ \mathbf{010} \\ \mathbf{011} \\ 100 \\ 101 \\ \mathbf{110} \\ \mathbf{111} \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{0} & \mathbf{0} & 0 & 0 & \mathbf{0} & \mathbf{1} \\ 0 & 0 & \mathbf{0} & \mathbf{0} & 0 & 0 & \mathbf{1} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \mathbf{0} & \mathbf{1} & 0 & 0 & \mathbf{0} & \mathbf{0} \\ 0 & 0 & \mathbf{1} & \mathbf{0} & 0 & 0 & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{6}$$

Para a construção da matriz é passada apenas o operador e as posições a serem calculadas. Para que possa ser possível realizar os cálculos é necessário saber quais as posições dos controles. Como destacada na matriz $Contr.$, a porta $X^{\otimes 2}$ é inserida na transformação levando em conta os índices binários, linha e coluna (omitido na matriz), relativos a posição do controle, no caso o segundo em 1.

A semântica da transformação é representada na eq. (7), como apenas a parte relevante para os cálculos são transmitidas, é preciso um cálculo de *offset* para a atualização da memória.

$$\begin{array}{c} \mathbf{010} \\ \mathbf{011} \\ \mathbf{110} \\ \mathbf{111} \end{array} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \delta \\ \gamma \end{bmatrix} = \begin{array}{c} \mathbf{111} \\ \mathbf{110} \\ \mathbf{011} \\ \mathbf{010} \end{array} \begin{bmatrix} \gamma \\ \delta \\ \beta \\ \alpha \end{bmatrix} \tag{7}$$

Extendendo para múltiplos *qubits*, a matriz correspondente à transformação terá um crescimento exponencial 2^n, onde n é o número de *qubits*. Utilizando os processos parciais haverá uma redução no volume de dados transferidos.

5. Considerações Finais

A extensão do ambiente *VirD-GM* para suporte à portas controladas proporcionará a utilização do mesmo para simulação de algoritmos quânticos mais complexos. Em trabalhos relacionados, o suporte à paralelização massiva e processos parciais nas demais portas obteve um desempenho significativo em relação à versão distribuída inicial. Estes resultados motivaram o estudo e a futura implementação do suporte para as portas controladas.

Como trabalhos futuros busca-se a implementação da proposta e geração de resultados utilizando algoritmos quânticos para a mensuração do desempenho.

Referências

Avila, A., Maron, A., Reiser, R., Pilla, M., and Yamin, A. (2014). Gpu-aware distributed quantum simulation. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, SAC '14, pages 860–865.

Deutsch, D. (1985). Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117.

Dirac, P. A. M. (1939). A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 35:416–418.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219.

Gutiérrez, E., Romero, S., Trenas, M. A., and Zapata, E. L. (2010). Quantum computer simulation using the {CUDA} programming model. *Computer Physics Communications*, 181(2):283 – 300.

Maron, A., Reiser, R., and Pilla, M. (2013). High-performance quantum computing simulation for the quantum geometric machine model. In *Proceedings of the 13th International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pages 474–481.

Maron, A., Reiser, R., Pilla, M., and Yamin, A. (2012). Quantum processes: A new interpretation for quantum transformations in the vpe-qgm environment. In *Proceedings of the XXXVIII Conferencia Latinoamericana En Informatica*, CLEI '12, pages 1–10.

Nielsen, M. A. and Chuang, I. L. (2011). *Quantum Computation and Quantum Information*. Cambridge University Press, New York, NY, USA, 10th edition.

Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509.

Viamontes, G. F., Rajagopalan, M., Markov, I. L., and Hayes, J. P. (2003). Gate-level simulation of quantum circuits. In *Proceedings of the 2003 Asia and South Pacific Design Automation Conference*, ASP-DAC '03, pages 295–301.