11ª Jornada Científica e Tecnológica do IFSULDEMINAS & 8º Simpósio de Pós-Graduação

ANÁLISE E COMPARAÇÃO DA COMPLEXIDADE TEMPORAL ENTRE ALGORITMOS IMPLEMENTADOS EMPREGANDO OS PARADIGMAS CLÁSSICO E QUÂNTICO

Amanda G. VALÉRIO[1]; Taffarel BRANT-RIBEIRO[2]

RESUMO

Na busca por aumentar a capacidade de processamento dos computadores, foi desenvolvida a computação quântica, que se mostra mais eficiente que a computação clássica para a resolução de problemas considerados custosos. Com a manipulação constante de grandes volumes de dados armazenados, foi desenvolvido um algoritmo de busca para a computação quântica denominado Algoritmo de Grover, que promete ser mais eficiente que os algoritmos clássicos. Este trabalho almeja seu desempenho no simulador QCL e compará-lo através de análise matemática e empírica com os algoritmos de busca: Busca Binária e Busca Sequencial. O algoritmo Quick Sort será usado como apoio, para ordenar os dados antes da execução do algoritmo de Busca Binária. Espera-se com isso descobrir se o Algoritmo de Grover mantém sua eficiência ou, caso contrário, qual o obstáculo e a sua complexidade real nesta situação.

Palavras-chave:
Análise de complexidade; algoritmos de busca; algoritmo de Grover; computação quântica.

1. INTRODUÇÃO

A área de desenvolvimento de *hardwares* para computação tem como um de seus intuitos a busca pela redução no tamanho de seus processadores e aumento da capacidade de processamento (NIELSEN; CHUANG, 2010). Porém, em virtude de seu tamanho, os dispositivos estão começando a sofrer interferências causadas por efeitos quânticos (NIELSEN; CHUANG, 2010). Para resolver este problema, foi desenvolvida a computação quântica, onde estas interferências passam a ser vantajosas para o aumento da velocidade de processamento. Com isso, o paradigma quântico se transformou em uma ferramenta útil para a resolução de problemas considerados custosos ou impossíveis aos computadores atuais (clássicos), como por exemplo, problemas resolvidos apenas por força bruta e problemas da classe NP (NIELSEN; CHUANG, 2010).

O qubit, uma partícula com finalidade análoga ao bit da computação clássica, torna o processamento das informações mais rápido por conta de suas propriedades de sobreposição e de emaranhamento. Com a sobreposição de estados dos qubits, obtém-se o paralelismo, onde um único circuito calcula simultaneamente diferentes respostas para um problema, ao contrário da computação paralela clássica, onde cada um dos circuitos calcula uma resposta simultaneamente

[1] Iniciação Científica, IFSULDEMINAS – *Campus* Passos. E-mail: amanda.valério@alunos.ifsuldeminas.edu.br.
[2] Orientador, IFSULDEMINAS – *Campus* Passos. E-mail: brant.ribeiro@ifsuldeminas.edu.br.

(NIELSEN; CHUANG, 2010). O emaranhamento permite que os estados de diferentes qubits estejam correlacionados, permitindo a troca de informações rapidamente mesmo que estejam separados fisicamente (GAY, 2006).

Em meio às pesquisas sobre a computação quântica, foram desenvolvidos por Ömer (1998, 2000) a linguagem de programação QCL, nome derivado de Quantum Computation Language, e seu interpretador, escritos em C/C++, com sintaxe derivada do C e suporte a operações clássicas. Os programas em QCL podem ser executados em um computador clássico que controla um computador quântico (ÖMER, 2000). Seu interpretador também tem a capacidade de simular os algoritmos quânticos em um computador clássico, com um número de qubits que pode ser definido pelo programador (ÖMER, 1998).

Nos dias atuais, a computação trabalha constantemente com a manipulação de um grande volume de dados. Para buscar um elemento, os métodos mais eficientes requerem que estes estejam ordenados (CORMEN et al. 2009). Levando em consideração a eficiência do computador quântico para buscas exaustivas e sua velocidade de processamento, foi desenvolvido o Algoritmo de Grover (1998), cujo objetivo se baseia em buscar um elemento em um vetor desordenado. O algoritmo utiliza a propriedade de superposição para reduzir o tempo de processamento para $\Omega(\sqrt{n})$, enquanto um algoritmo clássico precisaria de, em média, $O(n/2)$. Apesar de ser o primeiro do ramo, é comprovadamente ótimo, ou seja, sua complexidade já está na forma mais otimizada possível (PORTUGAL, 2010).

Explicando o algoritmo proposto por Grover (1998), o sistema é iniciado com todos os estados com a mesma amplitude e em seguida é deixado em um estado aleatório. Repetem-se $\sqrt{n}$ vezes as operações unitárias de rotação de fases (realizada se o valor do estado for o procurado, ou seja, igual a 1), rotação de matrizes e aplicação da Transformada de Walsh-Hadamard (em que ambas podem ser vistas como uma operação de inversão sobre a média). Por fim, ao analisar o estado resultante, este tem uma probabilidade de pelo menos 50% de ser o elemento procurado.

Neste contexto, essa pesquisa tem como objetivo comparar a eficiência do algoritmo de Grover com os algoritmos clássicos de busca, onde os escolhidos foram Busca Binária e Busca Sequencial, tomando como parâmetro a complexidade de tempo entre eles. Além disso, pretende-se apresentar uma análise da eficiência do interpretador QCL, com base nos experimentos feitos usando o algoritmo de Grover.

2. MATERIAL E MÉTODOS

Para a codificação dos algoritmos, foram escolhidas as linguagens QCL, para o paradigma quântico, e C, para o paradigma clássico. A escolha das linguagens se deu por motivos de

familiaridade entre elas e alto nível para programação (ÖMER, 2000; GAY, 2006). Para a instalação do interpretador QCL[3], são necessários os pacotes *bison*, *libplot*, *ncurses*, *readline5, flex* e um compilador C/C++. No caso, foi utilizado o GNU Compiler Collection (GCC).

No trabalho de Ömer (2000), além de detalhes sobre a linguagem, encontram-se também as instruções para a codificação do algoritmo de Grover, que foram usadas nesta pesquisa. Descobriu-se que o QCL não inclui uma função nativa para o cálculo do tempo gasto. Para suprir essa necessidade, tem sido desenvolvido um Shell Script que salvará os horários iniciais e finais das execuções em um arquivo externo.

Os algoritmos clássicos Busca Binária, Busca Sequencial e Quick Sort estão em desenvolvimento de acordo com a literatura e podem ser encontrados em Cormen (2014) e Cormen et al. (2009). Optou-se pela utilização do Quick Sort porque este é considerado um dos melhores algoritmos de ordenação (CORMEN et al. 2009) e o algoritmo de Busca Binária necessita de que seus dados estejam ordenados antes de sua execução. Cada algoritmo terá seu tempo de execução calculado separadamente.

As entradas com o escopo para a busca serão iguais para todos algoritmos e geradas aleatoriamente por uma função externa. Para uma análise mais completa, serão realizadas execuções com bancos de dados de tamanhos variando de 100.000 posições até 10.000.000. Caso a linguagem ou o paradigma não suporte essa quantidade, será usado o maior tamanho possível e este número será tomado como o teto também para o tamanho utilizado nos outros algoritmos.

A partir disso será possível realizar as análises empírica e matemática. Para a análise empírica serão realizadas execuções sucessivas testando o melhor caso, pior caso e o caso médio para cada situação. Com os gastos de tempo de cada algoritmo, serão gerados gráficos para análise e comparação com a função obtida através da análise matemática de complexidade. Com todos os resultados, será possível dizer qual dos algoritmos se mostrou mais eficiente para cada situação.

3. RESULTADOS ESPERADOS E CONTRIBUIÇÕES

Esta pesquisa atualmente se encontra em fase de execução. Os algoritmos de Grover, Busca Binária, Busca Sequencial e Quick Sort foram desenvolvidos e se encontram estáveis, porém sem acesso aos dados que serão utilizados nos testes. Um grande tempo foi gasto realizando uma busca na documentação da linguagem por uma função nativa que pudesse acessar a hora atual do sistema ou contar o tempo gasto. No entanto, como uma função com essas características não foi encontrada, optou-se por desenvolver um Shell Script exclusivo para esta finalidade. O mesmo já se encontra em fase final de codificação, restando apenas a criação de uma função para salvar os

[3] Disponível em: http://tph.tuwien.ac.at/~oemer/qcl.html.

valores de tempo em um arquivo externo.

A partir disso, como resultados esperados almeja-se observar se a simulação do algoritmo de Grover se mostra competitiva com os algoritmos clássicos comparados e, caso contrário, qual o obstáculo para isso. Como contribuições, caso o QCL se mostre eficiente, sua utilização também poderá trazer otimização de tempo em outras áreas como análise de dados, problemas NP, simulações e afins. Além disso, a plataforma para a simulação de programas quânticos utilizada nesta pesquisa poderá ajudar na familiarização dos programadores com a programação quântica.

4. CONSIDERAÇÕES FINAIS

Com o desenvolvimento da mecânica quântica e a busca por aumento na capacidade de processamento, surge a computação quântica. Neste contexto, esta pesquisa se propõe a verificar até que ponto o uso de um simulador de computador quântico se mostraria viável para a implementação de algoritmos quânticos, usando como base para esta verificação o algoritmo de Grover e sua comparação com os algoritmos clássicos análogos. Após definido um valor teto para entrada, será possível perceber se o QCL se mostra eficiente para auxiliar em tarefas que necessitam de computação de alto desempenho, ou até mesmo suprir a ausência de um computador quântico real, como, por exemplo, para fins didáticos.

REFERÊNCIAS

CORMEN, Tomas H. **Desmistificando Algoritmos.** 1. ed. Rio de Janeiro: Elsevier, 2014.

CORMEN, Thomas H. et al. **Introduction to Algorithmics**. 3. ed. Cambridge, Massachusetts: The MIT Press, 2009.

GAY, Simon J. **Quantum programming languages: Survey and bibliography**. Mathematical Structures in Computer Science, v. 16, n. 4, p. 581-600, 2006.

GROVER, Lov K. **A Fast Quantum Mechanical Algorithm for Database Search.** Proceedings of the 30th Annual ACM Symposium on Theory of Computing, ACM Press, New York, 1998.

NIELSEN, Michael A.; CHUANG, Isaac L. **Quantum computation and quantum information**. Edição de 10° aniversário. United Kingdom: Cambridge University Press, 2010.

ÖMER, Bernhard. **A procedural formalism for quantum computing.** Dissertação de Mestrado. Technical University of Vienna, 1998.

ÖMER, Bernhard. **Quantum programming in QCL**. Dissertação de Mestrado, Institute of Information Systems Technical University of Vienna, 2000.

PORTUGAL, Renato. **Algoritmos Quânticos de Busca**. Notas em Matemática Aplicada, Sociedade Brasileira de Matemática Aplicada e Computacional, Vol. 47, 2010