Avaliação de Desempenho da Paralelização do Sequenciamento Genético em GPU*

Cristiano A. Künas[1], Vinicios Dutra Schulze[1], Edson L. Padoin[1]

[1]Universidade Reg. do Noroeste do Estado do Rio G. do Sul (UNIJUI)
DCEEng - Santa Rosa - RS - Brasil

cristiano.kunas@sou.unijui.edu.br, vine.vinedutra@hotmail.com

padoin@unijui.edu.br

Resumo. *Este trabalho apresenta uma avaliação de desempenho da paralelização do sequenciamento de DNA em ambientes com aceleradores. O seu objetivo é analisar os ganhos de desempenho alcançados nos dois modelos de aceleradores. Uma versão paralela foi implementada utilizando o ambiente CUDA da NVIDIA para explorar os dois modelos de arquiteturas. Os resultados iniciais demonstram reduções no tempo total de até* $7,4$ *e de até* $12,4$ *vezes quando executado nas placas GPGPU.*

1. Introdução

Aplicações de simulação, previsão do tempo, exploração de petróleo, modelagem climática e sequenciamento de genoma requerem equipamentos com grande poder de processamento e modelos eficientes. Algumas dessas aplicações científicas tem feito uso de unidade de processamento gráfico de uso geral (GPGPU) para conseguir simular problemas reais e atingir resultados precisos [Pavan et al. 2018, Martínez et al. 2018, Padoin et al. 2013]. Mais especificamente, as pesquisas que envolvem a comparação e sequenciamento de DNAs usam algoritmos que executam procedimentos de grande complexidade o que resulta em elevados tempos computacionais. Nesse contexto, diferentes abordagens têm sido implementadas para execução paralela almejando explorar o poder computacional das atuais gerações de GPUs.

Esta área de pesquisa, hoje denominada de Bioinformática, trabalha com grande complexidade de sequenciamento de genomas, resultado este do volume de dados que é analisado, em especial na comparação de longas sequências. O alinhamento de sequências é cada vez mais realizado em análise genética de organismos e possui grande importância em estudos de evolução molecular, bem como na detecção de novos vírus ou ainda na predições de doenças.

Neste contexto, o objetivo deste trabalho é avaliar os ganhos de desempenho da paralelização do sequenciamento de DNA em ambientes com aceleradores. Para tanto foi utilizado uma implementação paralela que realiza o sequenciamento de DNA com os modelos de programação CUDA permitindo que a mesma seja executada nas diferentes arquiteturas. A solução, utilizada baseia-se em uma versão modificada do algoritmo Smith-Waterman sendo capaz de realizar buscas com sequenciamento de 4 caracteres.

O restante do trabalho está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados. A Seção 3 apresenta a paralelização do algoritmo de alinhamento de DNA e descreve a metodologia utilizada na implementação bem como o ambiente de execução utilizado

*Trabalho desenvolvido com recursos do edital MCTIC/CNPq - Universal 28/2018 sob número 436339/2018-8 e do edital da VRPGPE bolsa PIBIC/UNIJUI.

na realização dos testes. Resultados são discutidos na Seção 4, seguidos das conclusões e trabalhos futuros.

2. Trabalhos Relacionados

As pesquisas realizadas buscam a comparação de sequências biológicas almejando definir a similaridade denominada de alinhamento entre duas sequências de nucleotídeos ou aminoácidos. A similaridade é calculada por *match* onde se encontram as coincidência e *mismatch* para as posições de divergência entre as sequências. Alguns algoritmos utilizam também penalização ou *gap* para os sequenciamentos onde é necessário adicionar um espaço em uma das sequências para obter o alinhamento [Mount 2004].

Dentre os tipo alinhamento, quatro são os mais reconhecidos: global, local e semi-global e *Multiple Sequence Alignment* (MSA). No primeiro tipo de alinhamento, global, as duas sequências são comparadas em toda a sua extensão de sequências. Assim, a implementação avalia se todos os caracteres das sequências estão em alinhamento. No segundo tipo de alinhamento, local, uma parte de cada uma das sequências é alinhada. Esta pesquisa busca definir uma região de maior similaridade entre as duas sequencias analisadas. Já o alinhamento semi-global, busca definir o prefixo ou o sufixo das sequências a estão sendo alinhadas. Por fim, o alinhamento múltiplo de sequências mais conhecido por MSA permite que várias sequências sejam alinhadas ao mesmo tempo [Batzoglou 2005, Edgar and Batzoglou 2006]

As pesquisas atuais usam algoritmos para realizar a comparação de sequências e definir o alinhamento. São representados por, A (adenina), C (citosina), G (guanina) e T (timina) para pesquisas de nucleotídeos em DNA e A, T, G e U (Uracila) para pesquisas de nucleotídeos em RNA. Assim, como resultado, as implementações definem uma matriz similaridade a qual apresenta um escore final obtido. Muita delas, são baseadas em algoritmos exatos, que buscam um alinhamento ótimo, ou algoritmos heurísticos, que adotam técnicas para encontrar soluções sub-ótimas. Dentre os algoritmos exatos os mais conhecidos são Needleman-Wunsch, Smith-Waterman, Gotoh, Myers-Miller e Fickett.

A utilização de Unidade de processamento gráfico de uso geral (GPGPU) tem-se mostrado como uma solução para incrementos de paralelismo. Simulações científicas que utilizam GPU tem alcançado ganhos de desempenho em pesquisas de DNA.

No trabalho de Baskett foram alcançados ganhos de uma ordem de magnitude com a adoção GPU na pesquisa de DNA em comparação com o método de pesquisa sequencial [Baskett et al. 2017]. Rastogi apresenta uma implementação usando CUDA para a estratégia DFS (*Depth First Search*). Os resultados do emprego da metodologia proposta alcançaram um fator de aceleração médio de sete, em comparação com o da execução da CPU [Rastogi and Guddeti 2014]. Samsi apresenta uma implementação para GPU de comparação de sequência de DNA que adota uma abordagem de multiplicação de matriz sobrecarregada para comparações de DNA [Samsi et al. 2017].

Diferente de outras abordagens que alocam carga de trabalho nas arquiteturas de CPU e GPU, ou trabalhos que usam GPUs para obter ganhos de desempenho consideráveis quando comparados à arquitetura tradicional de CPU, nosso objetivo visa aumentar o desempenho considerando o uso de diferentes estratégicas para acessar diferentes níveis de memória das GPUs.

3. Implementação Paralela e Metodologia de Avaliação

O algoritmo de alinhamento utilizado para analisar os ganhos de performance utiliza um arquivo com as 24 combinações de A, C, G e T de um DNA. Primeiramente o algoritmo busca a quantidade que cada um dos 24 nucleotídeos está presente no primeiro DNA. Na sequência ele verifica se os 5 nucleotídeos mais encontrados no primeiro DNA encontram-se na mesma posição no segundo DNA.

A implementação paralela para GPGPU foi desenvolvida utilizando CUDA e almeja utilizar todos os cores disponíveis em cada placa acelaradora da fabricante NVIDIA. A programação com CUDA pemite o acesso aos diferentes níveis de memória das GPUs, as quais são: *global, local, texture, constant, shared* e *register memory*. Os dados armazenados em cada nível são visíveis e acessados de diferentes modos. Quando armazendos em memória global podem ser acessados por todos os blocos e threads. Já em memória local e registradores, os dados possuem acesso restrito às threads que escreveram os dados. Por fim, em memória compartilhada é somente acessada pelas threads de um mesmo bloco.

A implementação foi realizada com a versão 6 do CUDA que permite o uso da *Memória Unificada*. Deste modo, a programação é facilitada e os aplicativos possuem acesso as duas memórias, da CPU e da GPU, sem a necessidade de copiar dados manualmente entre elas.

Para validar a estratégia de proposta foram utilizados dois equipamentos. O primeiro com processador Intel Core i7-8700 3.20GHz, 16 GB de Memória RAM, acelerador NVIDIA GeForce GTX 1050 Ti com 4GB de GDDR5 e 768 CUDA cores. O segundo com processador Intel Core i7-9750 2.60 GHz, 16 GB de Memória RAM, acelerador NVIDIA GeForce GTX RTx 2060 com 6GB de GDDR6 e 1920 CUDA cores. O Sistema Operacional instalado é Linux Mint com o kernel 4.15.0-70-generic. A implementação do algoritmo foi realizada na linguagem de programação C, com o compilador g++ a versão 7.4.0 e o nvcc 10.1.243. Os resultados apresentados representam uma média de 10 repetições.

4. Resultados

Na Figura 1 são apresentados os tempos de execução mensurados nas execuções realizadas com os dois modelos de GPGPU.

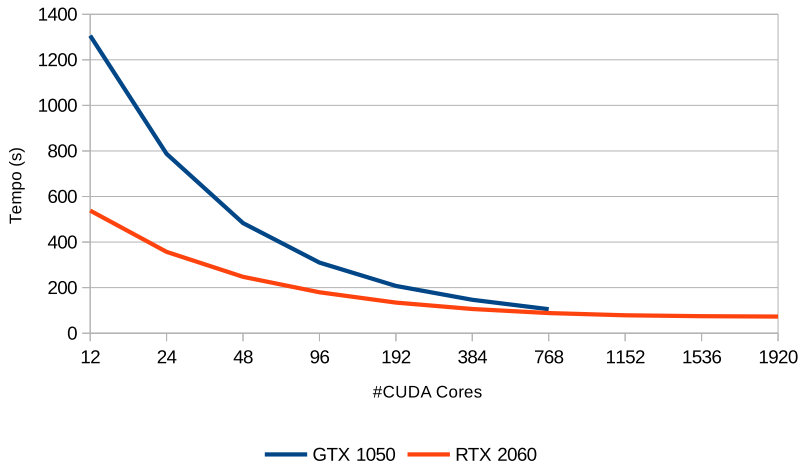

Figura 1. Tempos de execução (s) nos diferentes modelos de GPGPU

No primeiro modelo de GPU foram variados o número de cores entre 12 a 768 CUDA cores e no segundo de de 12 a 1920. Nos testes, os tempos de execução foram reduzidos de $1306,1$ segundos para $105,4$ segundos no primeiro modelo e de $538,31$ segundos para $73,1$ segundos no segundo modelo. Tais reduções representa ganhos de $12,4$ e de $7,4$ vezes respectivamente. Destaca-se que maiores ganhos não foram alcançados devido a parte sequencial do código que realiza sincronismo entre as duas etapas da implementação.

5. Conclusões e trabalhos futuros

Este trabalho apresentou uma avaliação de desempenho da paralelização do sequenciamento de DNA em ambientes com diferentes aceleradores. Os resultados preliminares da versão paralela implementada apresentaram reduções no tempo total de execução.

Como futuros trabalhos, pretende-se realizar melhorias no algoritmo analisado os dois tipos de gerencimaneto de memória com $cudaMalloc()$ e $cudaMallocManaged()$. Também pretende-se analisar os ganhos com a paralelização de outros algoritmos que buscam alinhamento ótimo ou algoritmos heurísticos.

Referências

Baskett, W., Spencer, M., and Shyu, C. (2017). Efficient gpu-accelerated extraction of imperfect inverted repeats from dna sequences. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 516–520.

Batzoglou, S. (2005). The many faces of sequence alignment. *Briefings in bioinformatics*, 6(1):6–22.

Edgar, R. C. and Batzoglou, S. (2006). Multiple sequence alignment. *Current opinion in structural biology*, 16(3):368–373.

Martínez, V., Serpa, M., Navaux, P. O. A., Padoin, E. L., and Panetta, J. (2018). Performance prediction of geophysics numerical kernels on accelerator architectures. In *The Eighth International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies (ENERGY 2018)*, pages 1–6, Nice - França.

Mount, D. W. (2004). *Bioinformatics: sequence and genome analysis. 2nd*, volume 692. Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press. xii.

Padoin, E. L., Pilla, L. L., Boito, F. Z., Kassick, R. V., Velho, P., and Navaux, P. O. A. (2013). Evaluating application performance and energy consumption on hybrid CPU+GPU architecture. *Cluster Computing*, 16(3):511–525. 10.1007/s10586-012-0219-6.

Pavan, P. J., Serpa, M., Carreno, E. D., Martínez, V., Padoin, E. L., Navaux, P., Panetta, J., and Méhaut, J.-F. (2018). Improving performance and energy efficiency of geophysics applications on gpu architectures. *High Performance Computing: 5th Latin American Conference, CARLA 2018, Bucaramanga*, pages 1–11.

Rastogi, P. and Guddeti, R. (2014). Gpu accelerated inexact matching for multiple patterns in dna sequences. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 163–167.

Samsi, S., Helfer, B., Kepner, J., Reuther, A., and Ricke, D. O. (2017). A linear algebra approach to fast dna mixture analysis using gpus. In *2017 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–6.