Uso de Computação Híbrida para Execução do Algoritmo Friends-of-Friends com OpenACC

Ana Luísa V. Solórzano[1], **Andrea S. Charão**[1]
Haroldo F. de Campos Velho[2], **Renata S. R. Ruiz**[2]

[1] Laboratório de Sistemas de Computação
Universidade Federal de Santa Maria

[2]Laboratório Associado de Computação e Matemática Aplicada
Instituto Nacional de Pesquisas Espaciais

{alsolorzano, andrea}@inf.ufsm.br, haroldo@lac.inpe.br, rennaruiz@gmail.com

Resumo. *A área de Cosmologia lida com grandes quantidades de dados observacionais do Universo. Algoritmos como o Friends-of-Friends são utilizados para processar esses dados. Anteriormente, o FoF foi paralelizado para execução em um ambiente composto por CPU e GPU utilizando OpenACC. Nesta nova abordagem, obteve-se ganho de desempenho superior às versões anteriores, explorando melhor o potencial do dispositivo acelerador.*

1. Introdução

Observatórios virtuais surgiram como um recurso para disponibilizar e analisar dados astronômicos provenientes de simulações sobre o Universo. Atualmente, essas simulações estão cada vez mais sofisticadas, capturando maiores quantidades de dados, utilizados para prever comportamentos similares na formação de estruturas no Universo, como galáxias e aglomerados de galáxias [Madsen 1996].

Visto isso, surge a necessidade de utilizar algoritmos eficientes e com alto desempenho para processar essa grande quantidade de dados. O uso de computação híbrida é uma opção vantajosa nesse sentido, podendo explorar o processamento em dispositivos aceleradores, como uma GPU ou uma FPGA. Um dos algoritmos utilizado para tratar dados de Observatórios Virtuais é o Friends-of-Friends (FoF).

Este trabalho é continuação de um trabalho anterior que utilizou um ambiente de computação híbrida composto por CPU e GPU para a execução do FoF com o padrão OpenACC. Nesta nova abordagem, paralelizou-se o programa que utiliza o algoritmo, resultando em um ganho de desempenho superior às versões do outro trabalho e explorando melhor o uso da GPU.

2. Friends-of-Friends

O Friends-of-Friends (FoF) é um algoritmo que classifica objetos astronômicos para identificar quais estão em interação gravitacional [Huchra and Geller 1982, Caretta et al. 2008]. Para isso, considera a proximidade física dos corpos (objetos ou partículas), dados pela sua posição em um espaço tridimensional.

A classificação considera um cenário de N-corpos em atividade gravitacional e um raio de interação gravitacional (raio de percolação), definido pelo usuário. Partículas

presentes à esfera definida pelo raio são consideradas "amigas" e classificadas em um mesmo grupo, de forma com que uma partícula possa ser o elo de ligação entre grupos distintos. Ao final, o algoritmo apresenta o número de grupos encontrados.

3. Trabalhos relacionados

Existem diversos classificadores de objetos astronômicos, como o AHF[1] e o Rockstar[2]. Ambos oferecem paralelização do código com MPI e com OpenMP, sendo que o AHF permite com que o usuário escolha o número de CPUs empregadas [Knollmann and Knebe 2009] e o Rockstar utiliza um algoritmo paralelo baseado no FoF utilizado neste trabalho [Behroozi et al. 2013].

Normalmente, esses programas são desenvolvidos por especialistas em Cosmologia com conhecimento de computação, e por isso podem não oferecer maiores análises de desempenho sobre a paralelização dos códigos, além de não explorarem o uso de dispositivos aceleradores [Knebe et al. 2011]. Supõe-se que isso ocorra pois os autores priorizam a funcionalidade dos programas, sem enfoque na área de computação.

4. Paralelização do FoF com OpenACC

Em trabalho anterior, investigou-se a paralelização do FoF com OpenACC para um ambiente composto por CPU e GPU, com mínima modificação possível do código original [Solórzano et al. 2018]. Como resultado, foram apresentadas duas versões, que apesar de obterem desempenho superior à versão serial, não exploraram todo o potencial da GPU. Nesta nova abordagem, buscou-se implementar um código que executasse o FoF de maneira otimizada em GPU. Os processos realizados são descritos a seguir:

4.1. Execução do FoF

A nova proposta consistiu em usar diretivas OpenACC para executar o algoritmo FoF sequencialmente em múltiplos *kernels* da GPU, cada um atuando sobre uma partição do total de partículas de entrada. Entretanto, após essa execução, poderiam existir partículas em *zonas de fronteira*, que são responsáveis por associar dois grupos classificados por *kernels* distintos, o que exigiu a implementação de um pós-processamento.

4.2. Pós-processamento

Após a leitura do arquivo de entrada, ordenou-se as partículas por um dos seus eixos de posicionamento. Ao finalizar a execução do FoF em GPU, obteve-se a classificação inicial dos grupos e realizou-se a verificação de partículas nas *zonas de fronteira*. Para isso, utilizou-se a fórmula para o cálculo da distância entre dois pontos para duplas de partículas, considerando o raio de percolação como limite de distância entre elas.

Caso uma partícula fosse identificada como pertencente a um grupo distinto, ela e todas as outras associadas ao grupo atual deveriam ser modificadas para pertencer ao novo grupo. Entretanto, como a junção de mais de um grupo poderia estar envolvida, foi preciso realizar uma varredura, paralelizada com o padrão OpenMP, em todas as partículas após cada reclassificação. Esta situação foi relatada em [Madalosso et al. 2015] chamada de *relabel*, sendo implementada de modo distinto no trabalho citado, que utilizou outra estrutura de dados para implementação do FoF.

[1] ttp://popia.ft.uam.es/AHF/Download.html

[2] https://bitbucket.org/gfcstanford/rockstar

5. Experimentos

Os experimentos foram realizados em uma máquina com dois processadores Intel® Xeon® E5-2650 v3 de dez cores físicos e vinte virtuais e com duas GPUs NVIDIA Tesla K80. O sistema operacional é Ubuntu 18.04.1 LTS e compilador da Portland Group edição Community[3] versão 18.4-0.

O número de *kernels* foi definido pelo usuário pela entrada padrão e o número de *threads* definido pela variável de ambiente OMP_NUM_THREADS. Foram realizadas 30 execuções com três amostras de dados observacionais do Consórcio Virgo[4]. As amostras tinham 65.536 partículas (Arquivo 1), 174.761 (Arquivo 2) e 249.420 (Arquivo 3), utilizando-se 200 *kernels* para o Arquivo 1, 300 para o Arquivo 2 e 600 para o Arquivo 3. Utilizou-se raio de percolação 0.1 conforme utilizado em Caretta et al. para processamento dos mesmos dados de simulação do Consórcio Virgo sobre galáxias.

6. Resultados

A Tabela 1 apresenta as médias dos tempos de execução e o *speedup* para a versão apresentada neste trabalho (ERAD/RS 2019) e para a melhor versão apresentada no trabalho anterior (ERAD/RS 2018), em relação ao tempo médio de execução do FoF serial. Desconsiderou-se o tempo de leitura da entrada, e se considerou a soma do tempo de execução do FoF em GPU e do pós-processamento em CPU para a nova abordagem.

Arquivos	Tempos de Execução (s)		Speedup	
de Entrada	ERAD/RS 2019	ERAD/RS 2018	ERAD/RS 2019	ERAD/RS 2018
1	1,43	13,79	31,96	3,31
2	6,27	76,61	48,97	4,00
3	290,17	1.335,67	18,52	4,02

Tabela 1. Tempos de execução e *speedup* para as três amostras de dados

A partir dos resultados, notou-se ganho de desempenho da nova versão em relação à serial e em relação à versão com OpenACC apresentada anteriormente. Destaca-se o *speedup* de quase 49 para o Arquivo 2 na nova versão, onde se encontrou um caso que explorou de maneira adequada os *kernels* da GPU definidos para realizar o processamento.

Conforme relatado em Madalosso et al., o FoF sequencial não trata casos de *relabel*, o que dificultou comparações iniciais com o novo algoritmo com OpenACC. Para garantir a corretude desta nova implementação, realizou-se execuções com entradas controladas, confirmando que a versão serial não trata o *relabel*. Assim, os resultados foram coerentes com o esperado, sendo encontrados para o Arquivo 1 46.382 grupos na versão serial e 45.715 na nova versão com OpenACC que realizou a junção de grupos nas *zonas de fronteira*, 113.660 e 111.313 para o Arquivo 2, e 157.682 e 154.743 para o Arquivo 3.

A paralelização do pós-processamento com OpenMP não foi utilizada nessas execuções, pois utilizando um raio pequeno de 0.1, menor será o número de partículas associadas aos grupos e assim menor o número de partículas a serem reclassificadas, tornando essa paralelização custosa. Porém, com testes hipotéticos para raio de 1.7, a

[3]https://developer.nvidia.com/openacc-toolkit

[4]http://www.mpa-garching.mpg.de/Virgo/data_download.html

execução do Arquivo 2 com 200 *kernels* trouxe um ganho de desempenho de 1.3 com 4 *threads* no pós-processamento em relação à execução serial desta etapa.

7. Considerações finais

Este trabalho apresentou uma nova proposta para a execução do FoF utilizando OpenACC. Diferente do trabalho anterior que utilizou diretivas OpenACC para paralelizar o algoritmo sem modificação no código, aqui, reimplementou-se o programa utilizando as diretivas para controlar as execuções do FoF de maneira sequencial em diversos *kernels* da GPU, que atuaram paralelamente sobre partes do conjunto total de partículas.

Os resultados mostraram que é possível fazer bom uso da GPU para o processamento do FoF, obtendo-se ganho de desempenho em comparação à versão serial e à versão anterior com OpenACC. Porém, devido ao FoF serial não tratar de casos de *relabel*, a consistência dos grupos definidos em cada versão deve ser melhor analisada realizando-se mais testes com entradas controladas inclusive em comparação com outras versões do algoritmo.

8. Agradecimentos

Os autores agradecem ao PIBIC-CNPq-INPE pelo suporte na forma de bolsa de Iniciação Científica para o primeiro autor participante do projeto de nº 136023/2018-5.

Referências

Behroozi, P. S., Wechsler, R. H., and Wu, H.-Y. (2013). The rockstar phase-space temporal halo finder and the velocity offsets of cluster cores. *The Astrophysical Journal*, 762(2):109.

Caretta, C. A., Rosa, R. R., de Campos Velho, H. F., Ramos, F. M., and Makler, M. (2008). Evidence of turbulence-like universality in the formation of galaxy-sized dark matter haloes. *Astronomy & Astrophysics*, 487(2):445–451.

Huchra, J. P. and Geller, M. J. (1982). Groups of galaxies. I - Nearby groups. *Astrophysical Journal*, 257:423–437.

Knebe, A. et al. (2011). Haloes gone mad14: The halo-finder comparison project. *Monthly Notices of the Royal Astronomical Society*, 415(3):2293–2318.

Knollmann, S. R. and Knebe, A. (2009). Ahf: Amiga's halo finder. *The Astrophysical Journal Supplement Series*, 182(2):608.

Madalosso, O. M., Charão, A. S., de Campos Velho, H. F., and da Rocha Ruiz, R. S. (2015). Implementação do algoritmo friends of friends de complexidade n*logn(n) para classificação de objetos astronômicos. In *Anais da XV Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul*, pages 245 – 248.

Madsen, M. S. (1996). In *The Dynamic Cosmos - Exploring the Physical Evolution of the Universe*, New York, NY, USA. Chapman & Hall.

Solórzano, A. L. V., Charão, A. S., da Rocha Ruiz, R. S., and de Campos Velho, H. F. (2018). Exploração de computação híbrida com openacc em um algoritmo friends-of-friends para classificação de objetos astronômicos. In *Anais da XVIII Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul*, pages 129 – 132.