Uma proposta de análise comparativa de desempenho de implementações Barnes-Hut em clusters de computadores

Rodrigo Morante Blanco[1], **Wagner M. Nunan Zola**[1]

[1]Departamento de Informática – Universidade Federal do Paraná (UFPR)
Caixa Postal 19.081 – Curitiba – PR – Brazil

rodrigomorante@gmail.com, wagner@inf.ufpr.br

Resumo. *Neste trabalho apresentamos resultados parciais de desempenho do método Barnes-Hut em clusters de computadores e uma proposta para futuras melhorias. Os resultados mostram a escalabilidade deste método distribuído.*

1. Introdução

O problema dos N corpos se refere classicamente ao cálculo das forças entre corpos celestes que interagem entre si gravitacionalmente, com complexidade de $\mathcal{O}(N^2)$. Barnes e Hut propuseram um algoritmo hierárquico aproximado, baseado em *octrees*, com custo de $\mathcal{O}(N \log N)$. Novas técnicas foram desenvolvidas recentemente para otimização do cálculo de forças no método BH, tanto em processadores *multicore* [Delgado et al. 2019] quanto em *manycore* [Meyer et al. 2021]. O trabalho anterior é um exemplo recente da aplicação do algoritmo BH no contexto de aplicações que necessitam analisar grandes volumes de dados de alta dimensionalidade. No presente trabalho apresentamos um estudo inicial de possíveis métodos a serem empregados em versões distribuídas do algoritmo BH. Duas variantes são consideradas, ambas utilizando uma *octree* esparsa (baseada em ponteiros) construída a partir de todos os corpos presentes na simulação. A distribuição de trabalho pode ser *estática*, isto é, cada nodo processa uma quantidade fixa de corpos, ou *dinâmica*, na qual um nodo envia intervalos de corpos para os restantes nodos participantes à medida que o trabalho alocado a estes é finalizado.

2. Proposta

Será feita a análise da influência do processamento MPI no desempenho de várias implementações do método de Barnes-Hut, com o objetivo de determinar o peso da troca de mensagens no desempenho dos algoritmos. Mais adiante será possível comparar a execução em múltiplos núcleos com memória compartilhada usando *threads* (pthreads ou OpenMP) em relaçao às implentações com MPI. Deve ser verificada a escalabilidade utilizando MPI em um cluster computacional. Os experimentos descritos a continuação serão a base de futuros trabalhos. Para executar os experimentos descritos a continuação dispomos de um *cluster* de computadores com processadores Intel i7-4770 @ 3.40GHz de quatro núcleos físicos (ou oito núcleos virtuais utilizando a tecnologia *hyperthreading*) e 8GB de RAM, com sistema operacional Linux Mint LMDE 5, conectados através de uma rede Gigabit Ethernet. Nas simulações preliminares foram utilizados 10^6 corpos seguindo uma distribuição de Plummer. Os seguintes experimentos preliminares foram realizados:

Experimento 1: São utilizados 1, 2, 4 e 8 processos MPI realizando os cálculos em um único nodo físico, utilizando *hyperthreads*. Os resultados estão apresentados na Figura 1. Pode-se ver que ambos métodos apresentam *speedup* sublinear, mas a tendência é a aumentar se mais nodos são utilizados.

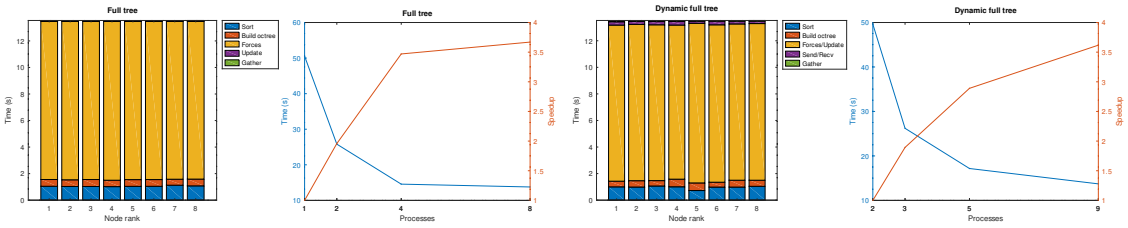

Figura 1. Tempo para 8 processos MPI em um único nodo físico e *speedup* à medida que aumenta o número de processos MPI. À esquerda, método com carga estática. À direita, com carga dinâmica.

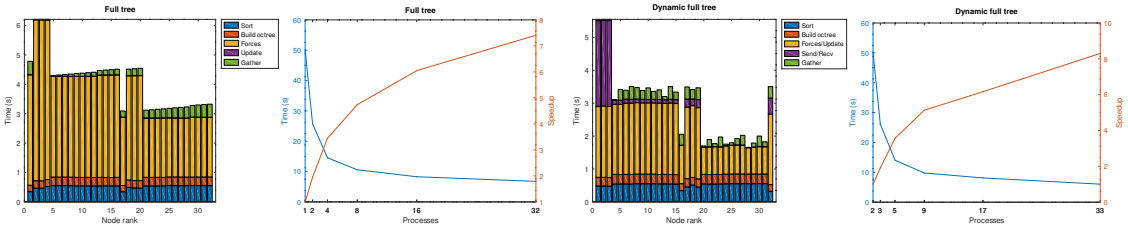

Figura 2. Tempo para 32 processos MPI, métodos com carga estática e dinâmica, em um *cluster* computacional com 4 processos MPI por nodo físico e *speedup* à medida que aumenta o número de nodos físicos.

Experimento 2: Utilizando quatro processos MPI por nodo físico, sem utilizar *hyperthreads*, em 1, 2, 4 e 8 nodos físicos. Com resultados mostrados na Figura 2, verificamos que, efetivamente os métodos continuam escalando com o aumento do número de nodos.

3. Conclusões e trabalhos futuros

O método de Barnes-Hut foi adaptado para ser distribuído utilizando MPI. Os resultados foram obtidos em um único computador e em um *cluster* computacional e mostram que as estratégias empregadas são promissoras. O custo da transmissão de dados e impacto das latências deve ser melhor estudado para determinar se seu impacto pode ser reduzido. Uma estratégia que deve ser explorada é o uso concomitante de paralelismo distribuído e de memória compartilhada. Para isto no futuro adaptaremos nossos algoritmos para utilizar OpenMP ou pthreads em cada nodo físico. Os resultados poderão ser comparados com a implementação de BH no *benchmark* Splash-3 [Sakalis et al. 2016] que utiliza memória compartilhada.

Este trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-Brasil (CAPES)–Código 001.

Referências

Delgado, A., Blanco, R. M., and Nunan Zola, W. (2019). Caminhamento paralelo Barnes-Hut com vetorização AVX2. In *Anais do XX Simpósio em Sistemas Computacionais de Alto Desempenho*.

Meyer, B. H., Pozo, A. T. R., and Zola, W. M. N. (2021). Improving Barnes-Hut t-SNE algorithm in modern GPU architectures with random forest KNN and simulated wide-warp. *J. Emerg. Technol. Comput. Syst.*, 17(4).

Sakalis, C., Leonardsson, C., Kaxiras, S., and Ros, A. (2016). Splash-3: A properly synchronized benchmark suite for contemporary research. In *2016 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*.