UMA ANÁLISE DO DESEMPENHO DO BENCHMARK AGGLOMERATIVE CLUSTERING EM DIFERENTES FERRAMENTAS DE PARALELIZAÇÃO

Juan da Silva Rios[1]*; Vitor Alano de Ataides[2]; Gerson Geraldo Homrich Cavalheiro[3]

[1]*Universidade Federal de Pelotas – jdsrios@inf.ufpel.edu.br*
[2]*Universidade Federal de Pelotas – vaataides@inf.ufpel.edu.br*
[3]*Universidade Federal de Pelotas – gerson.cavalheiro@inf.ufpel.edu.br*

1. INTRODUÇÃO

Benchmarking é uma prática amplamente utilizada em praticamente qualquer área da computação. Esta prática consiste na criação de programas, operações ou casos de testes para testar o desempenho de um objeto, seja ele software ou hardware (SAAVEDRA; SMITH, 1996). Neste artigo será apresentado o benchmark Agglomerative Clustering, algoritmo esse muito utilizado em diversas áreas da computação incluindo data-mining, bioinformática e compreensão onde essas áreas utilizam do processamento paralelo. Este mesmo consiste na divisão de um processo em outros menores, ou seja, processos mais leves que podem ser executados de maneira paralela, fazendo com que acelere a execução do mesmo. A proposta deste trabalho é fazer uma comparação da execução do benchmark Agglomerative Clustering em uma versão sequencial com diversas versões paralelas. Onde essas versões paralelas foram desenvolvidas em TBB (Thread Building Blocks) e OpenMP, bibliotecas desenvolvidas para C, C++ e Fortran. A linguagem de programação utilizada para o desenvolvimento dos algoritmos foi C++.

Agglomerative Clustering é um famoso algoritmo de data-mining. Sua entrada consiste em um data-set de pontos contidos em um espaço n-dimensional e uma função que mede a similaridade entre os itens do data-set. Esta função de similaridade é utilizada como uma métrica de distância, quanto mais similar dois elementos, mais próximos eles estão. A saída do algoritmo é uma árvore binária, chamada de dendograma, representando um agrupamento hierárquico de pares dos elementos do data-set.

Na sessão 2 é apresentado o benchmark Agglomerative Clustering e suas demais versões desenvolvidas neste trabalho. Na sessão 3 é apresentado os resultados bem como sua análise. Por fim, na sessão 4 é feita uma conclusão e sugestão de possíveis trabalhos futuros.

2. METODOLOGIA

Foi feito um estudo sobre a implementação do algoritmo Agglomerative Clustering e então decidiu – se implementar baseando – se na implementação descrita em Walter et al. (WALTER; PINGALI, 2008), onde sua implementação é feita de forma que se dois pontos concordarem que são vizinhos, seus pais próximos um do outro serão agrupados no dendograma final. Inicialmente, todos os pontos são inseridos em uma lista de nodos, são transformados em nodos de uma árvore e os nodos são inseridos em um Set. Então para cada nodo do set é feita a busca pelo seu vizinho mais próximo de seu vizinho mais próximo. Caso sim, os dois nodos são agrupados em um novo nodo, que terá como filho os dois nodos, que será adicionado a um Set auxiliar caso não, o nodo é adicionado ao Set auxiliar.

Para suas implementações em paralelo foi feito um estudo sobre as bibliotecas paralelas fornecidas pelo C++, e com isto foi escolhido TBB (Thread Building Blocks) e OpenMP, todas elas fornecidas e desenvolvidas pela Intel, para que se pudesse fazer uma comparação dos tempos de execução em cada versão e assim comparar o desempenho do algoritmo em cada biblioteca. Suas versões paralelas o laço interno foi removido e feito de forma paralela, e então em cada uma de suas execuções de seus laços externos foram lançadas a uma Thread para cada nodo do Set e então feita a sincronização das threads. Este processo é repetido até que o Set tenha tamanho igual a 1. O cuidado principal nas versões paralelas foi com as repetições, sendo então zonas críticas e necessitando de tratamento para isto.

Para a versão em TBB (Thread Building Blocks) foi utilizado seu template parallel_for para seu laço.

Em OpenMP foi utilizado um parallel for para seu laço e tratado suas zonas críticas com o pragma omp critical e para sincornização foi utilizado a diretiva pragma omp barrier.

Para a realização dos testes de desempenho do algoritmo em diferentes versões foi utilizado um computador com um processador Intel Core 2 Quad de 2.33 Ghz com 4 núcleos, 4 Gb de memória RAM e seu sistema operacional foi o Ubuntu 14.04 LTS com 64 bits.

3. RESULTADOS E DISCUSSÃO

Com a execução dos testes foram construído um gráfico para uma análise do algoritmo. Foram realizados diferentes testes variando o número de threads. O gráfico da figura 1 de tempo de execução do sequencial por tempo de execução em paralelo.

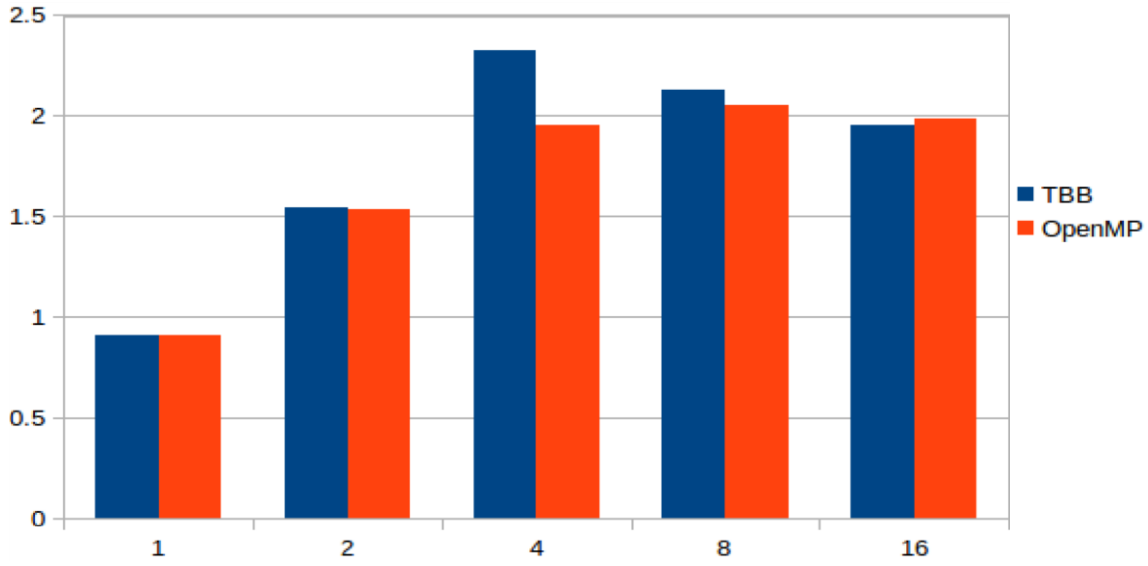

Figura 1. Speedup das versões em TBB e OpenMP

Foram utilizados 1000 pontos para os testes, e com uma thread as bibliotecas apresentaram um resultado praticamente igual, já que o algoritmo não está se benificiando do paralelismo. Com duas threads o TBB obteve um pequeno melhor resultado com relação ao OpenMP. Quatro threads TBB obteve um resultado ótimo, bastante supeior a outra biblioteca, OpenMP fazendo então uso da capacidade máxima do computador. Oito threads as duas ferramentas obtiveram um bom resultado, mas o TBB obteve uma pequena diferença em relação ao OpenMP obtendo assim um melhor desempenho. E por final dezesseis threads, ambas as ferramentas obtiveram um bom resultado, porém ainda inferior ao de quatro threads, o motivo que se dá essa diferença é que com a utilização de quatro threads o algoritmo utiliza a capacidade máxima do computador utilizado para os testes.

4. CONCLUSÕES

Neste trabalho foi implementado três versões do benchmark Agglomerative Clustering, uma versão sequencial e duas versões paralelas. Após a execução dos testes para a versão sequencial e suas versões paralelas pode – se concluir que, quanto maior o número de threads maior o desempenho da versão paralela e com isso maior speedup, porém, este ganho limita – se ao número de núcleos utilizado e disponível.

Um possível trabalho futuros seria a implementação do algoritmo em outras ferramentas paralela tais como, CilkPlus, C++11 e CUDA, para que possa explorar mais o paralelismo.

5. REFERÊNCIAS BIBLIOGRÁFICAS

PANG-NING Tan, STEINBACH Michael e KUMAR Vipin. **Introduction to Data Mining**. Boston: Addison-Wesley Longman, 2005. 1v.

SAAVEDRA, R. H. e SMITH, A. J. Analysis of benchmark characteristics and benchmark performance prediction. **Computer Science Technical Report UCB/CSD 92/715,** UC Berkeley. P.344-384.

B. Walter, K. Bala, M. K. and Pingali, K. In: **IEEE Symposium on Interactive Ray Tracing,** Los Angels, 2008.