PARALELIZAÇÃO DO ALGORITMO K-MEDOIDS PARA AGRUPAMENTO DE SINAIS FRACOS

Vivian Renata Nunes Toito e André Leon Sampaio Gradvohl, Dr. (orientador)
vivire92@hotmail.com ; gradvohl@ft.unicamp.br
Faculdade de Tecnologia - FT/UNICAMP
Agência Financiadora: CNPq
Palavras-chave: algoritmo k-medoids, sinais fracos, paralelização.

Introdução

O k-medoids é um algoritmo de particionamento de dados baseado na busca por um objeto representativo, denominado medoid, cuja proximidade aos demais objetos do agrupamento é máxima, além de estar localizado mais ao centro, segundo Kaufman e Rousseeuw (1987). Se eficiente, é capaz de reconhecer padrões, tornando-se útil como uma ferramenta de tomada de decisão.

Quando se trata de uma grande base de dados, este algoritmo apresenta baixa eficiência em relação ao tempo de execução, o que motivou o tema deste projeto que tem como finalidade otimizar o k-medoids a partir de sua paralelização com a API OpenMP (OPENMP, 2013) e acoplá-lo a uma ferramenta de reconhecimento de padrões a partir do agrupamento de sinais fracos.

Metodologia

Para o desenvolvimento deste projeto, foi necessário a seleção de trechos passíveis de paralelização, tendo prioridade aqueles com iterações encadeadas.

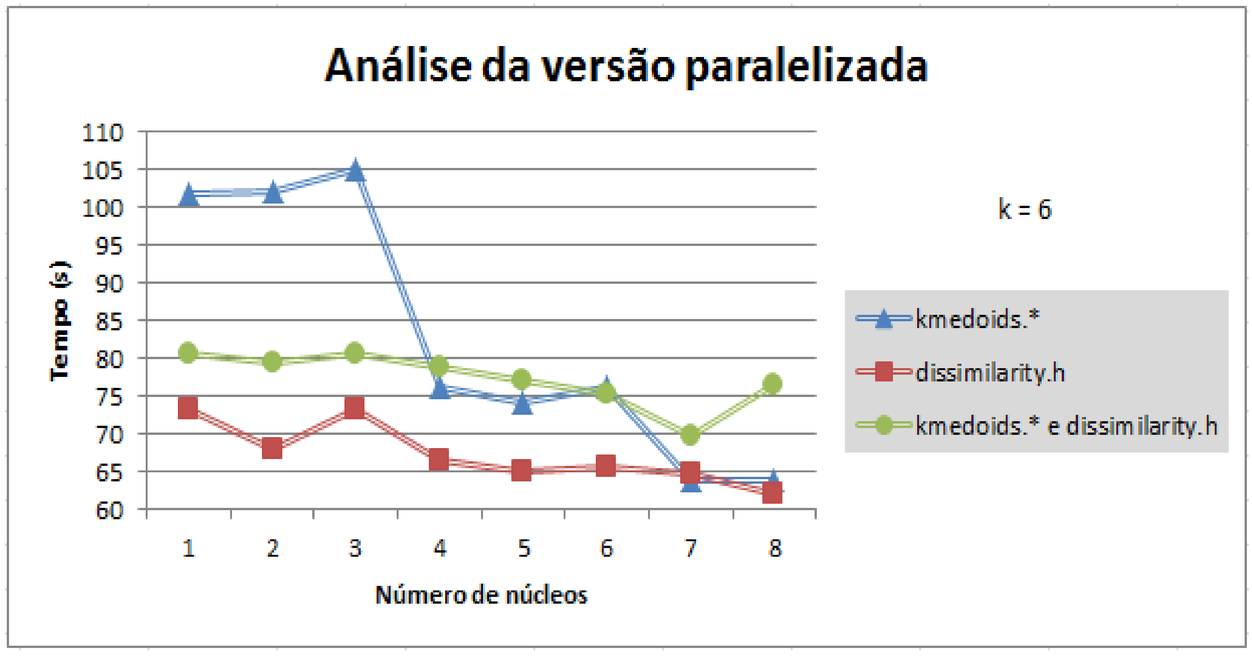
No primeiro semestre os estudos se concentraram na compreensão do algoritmo k-medoids e na implementação da versão serial desenvolvido por Todd Gamblin (GAMBLIN, 2010) na linguagem de programação C++. No segundo semestre, foram analisados os trechos para a paralelização do algoritmo e o desenvolvimento da versão paralelizada originando os testes de escabilidade para análise das soluções implementadas.

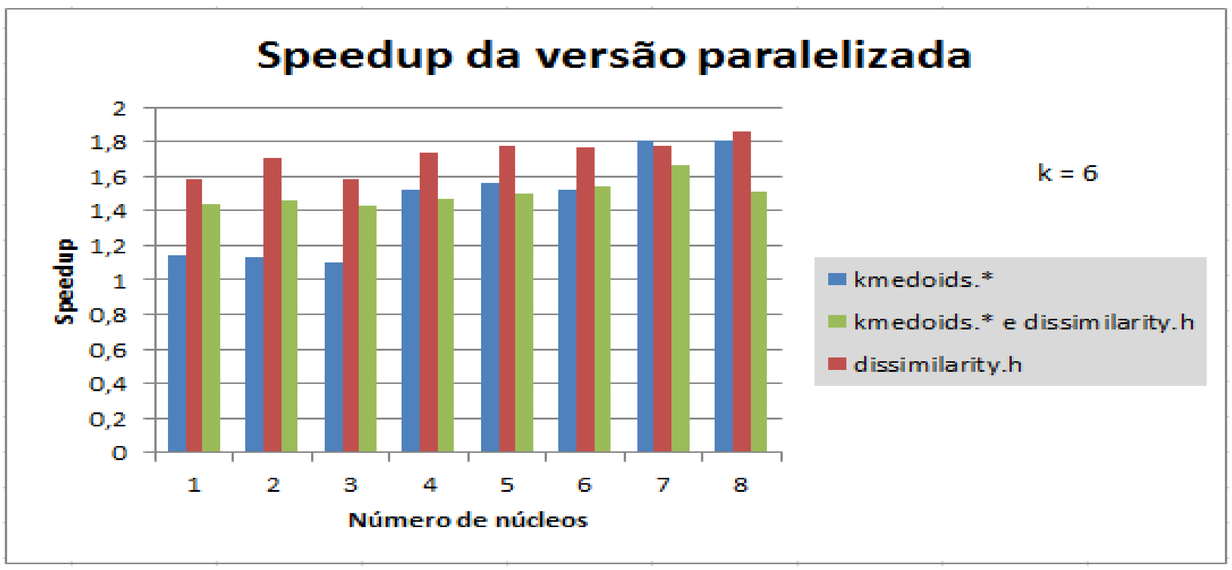
Para o desenvolvimento deste trabalho, os testes foram feitos em ambiente Linux através de acesso remoto utilizando uma máquina de 32 Gigabytes de memória RAM com processador AMD FX, com 8 núcleos lógicos (4 físicos), 4,2 GHz.

Resultados e Discussão

Os testes desenvolvidos para análise de escabilidade contém três valores para k (número de agrupamentos): 3, 6 e 9, sendo avaliados três módulos principais do algoritmo - `kmedoids.h`, `kmedoids.cpp` e `dissimilarity.h` – que apresentam os cálculos de custo e trocas dos medoids e cálculos de similaridade, respectivamente. Os resultados mais significativos foram os agrupamentos com k = 6, principalmente, no módulo `dissimilarity.h`. Veja os gráficos. É importante destacar que a versão serial do algoritmo com esse mesmo valor de k, apresentou tempo médio de 115,585 segundos.

Tal comportamento se dá pelo fato do módulo ser responsável por gerar as matrizes de similaridade do algoritmo que darão origem a k agrupamentos, sendo esta uma tarefa com alto custo de processamento. Dessa forma, quando paralelizada, o trabalho dos processadores é amenizado e, consequentemente, o tempo de execução é reduzido gradativamente conforme o número de núcleos aumenta





Conclusões

Com os dados apresentados, é possível observar que a paralelização dos módulos principais do algoritmo: `kmedoids.h`, `kmedoids.cpp` e `dissimilarity.h`, apresentaram resultados significativos quando comparado com o tempo obtido a partir da versão serial, em particular o módulo `dissimilarity.h`, na maioria dos casos.

Sendo assim, concluímos que a partir dos testes realizados a paralelização do algoritmo se mostra eficaz com a utilização da API OpenMP e apresenta resultados positivos quando o conjunto é formado por números inteiros. Em trabalhos futuros, faremos uma análise mais concentrada em informações textuais que geralmente compõem os sinais fracos.

Referências bibliográficas

GAMBLIN, Todd (2010) “Muster Documentation”. Disponível em: <http://tgamblin.github.com/muster/main.html>.

KAUFMAN, L. and Rousseeuw, P. J., “Clustering by means of medoids,” in Statistical Data Analysis Based on the L1Norm, Y. Dodge, Ed., pp. 405–416. North Holland/Elsevier, Amsterdam, 1987

OPENMP, The OpenMP API specification for parallel programming. About the OpenMP ARB and OpenMP.org. 2013. Disponível em: <http://openmp.org/wp/>.