

TRABALHO 2 - INTRODUÇÃO A COMPUTAÇÃO CIENTÍFICA

Otimização de Desempenho do Método de Newton Modificado

Fernando Gbur dos Santos GRR20211761

Guiusepe Oneda dal Pai GRR20210572

Introdução

O seguinte trabalho foi implementado com a intenção de melhorar o desempenho do método de Newton Modificado, utilizado no primeiro trabalho. Para fins de análise e comparação de desempenho, existem duas implementações do método. A primeira versão contém o código do primeiro trabalho sem alterações (apenas o método de Quasi-Newton foi retirado), enquanto a segunda versão contém o código com alterações em relação ao desempenho, mas mantendo o mesmo método.

Os testes foram realizados utilizando o computador h42 do Departamento de Informática da Universidade Federal do Paraná, utilizando métricas do programa LIKWID para a comparação dos dados.

A arquitetura da máquina utilizada para testes está a seguir:

CPU name: Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz

CPU type: Intel Coffeelake processor

CPU stepping: 9

O processador da máquina apresenta 3 níveis de cache.

Execução

Para a execução do trabalho, foi criado um script em bash que inicializa o programa com cada grupo de performance do LIKWID solicitado. Os resultados são escritos em arquivos, onde são utilizados o programa gnuplot, para uma análise gráfica.

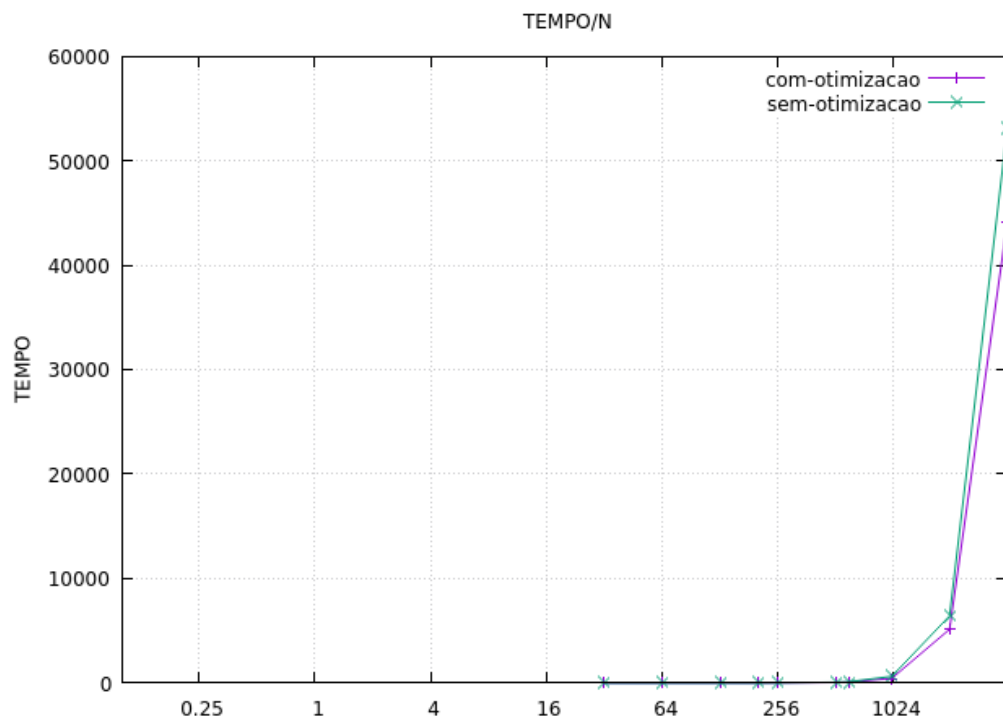
Otimizações

Foram realizadas 3 principais alterações, buscando a melhoria de desempenho.

1. A matriz hessiana de coeficientes foi substituída por uma estrutura que armazena apenas as diagonais não nulas, evitando ocupar espaço na memória com informações que não precisam ser apresentadas.
2. Todas as matrizes auxiliares utilizadas para cálculo ou armazenamento de funções foram alocadas de maneira menos custosa, diferente da primeira versão, onde eram alocadas como vetor de ponteiros de linhas separadas.
3. Algumas otimizações mais simples também foram aplicadas, como a fusão de laços que iteram sobre os mesmos limites.

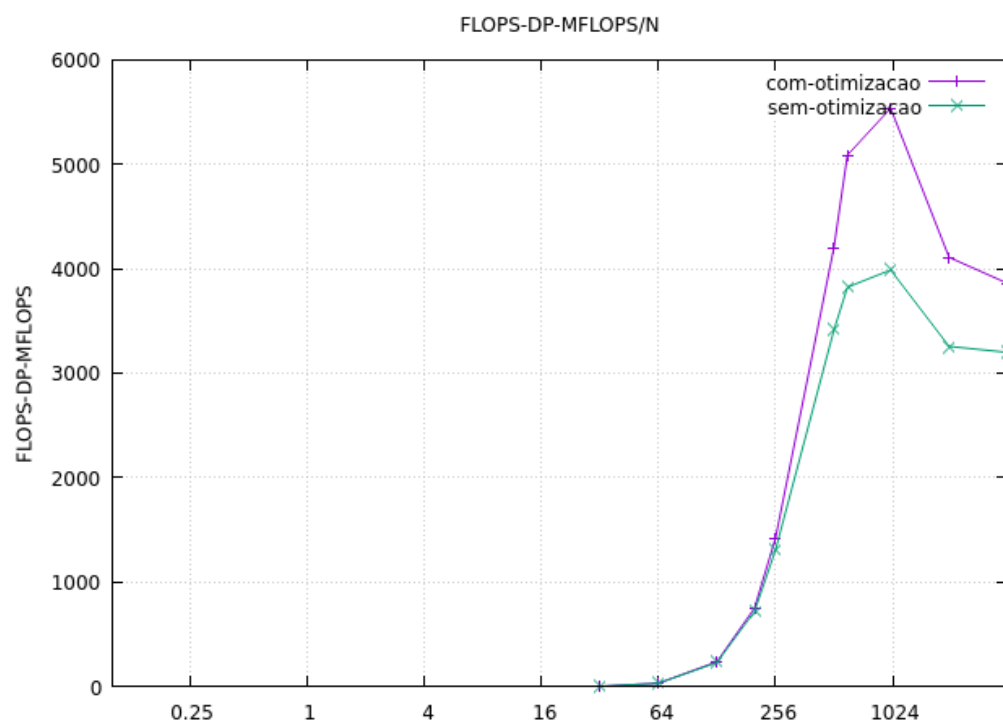
Resultados

1.Tempo



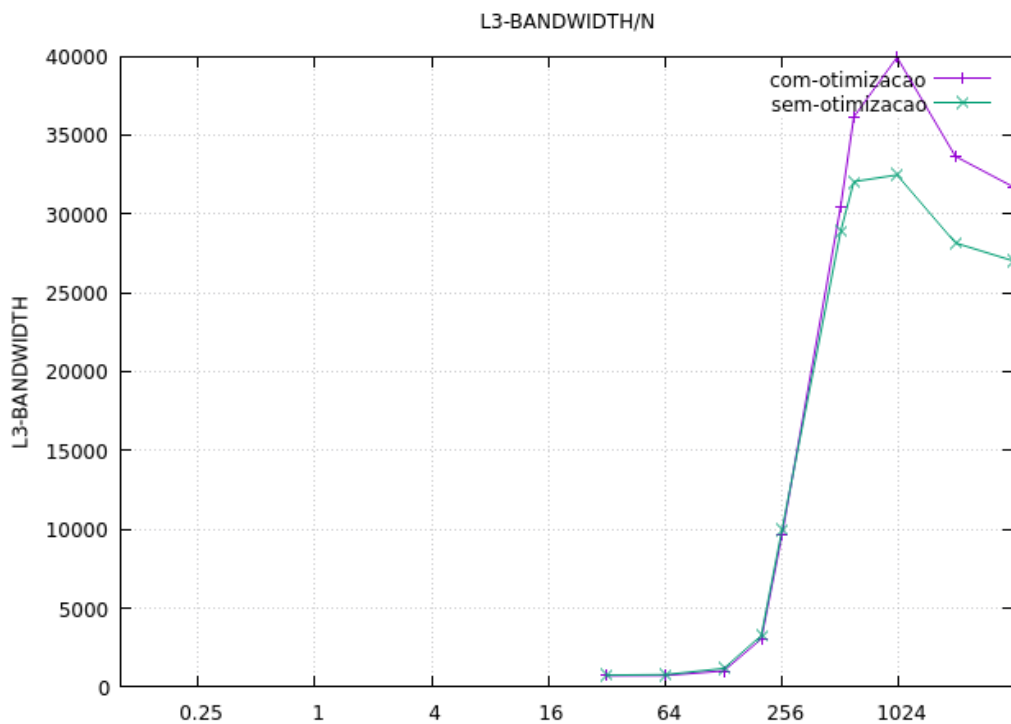
Mesmo não sendo muito grande, existe uma diferença de tempo de execução entre as duas versões, principalmente para valores de N muito grandes, a execução da versão 2 é menor (obs: o tempo está em milissegundos).

2.Flops/dp



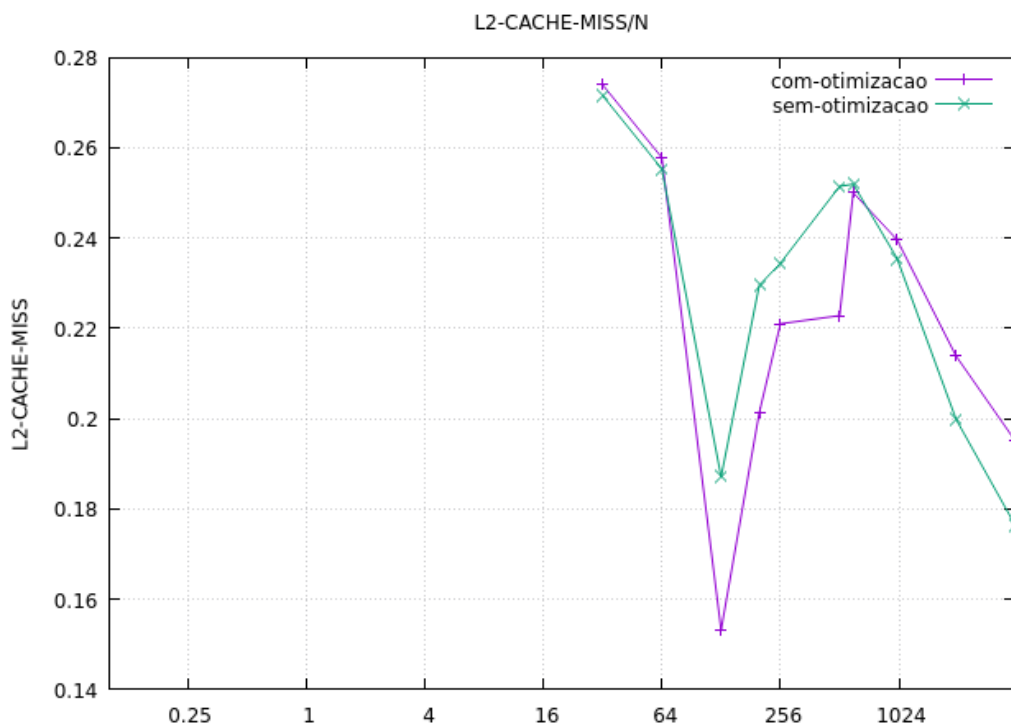
Esta métrica nos permite analisar a quantidade de operações de ponto flutuante por segundo. Com o código otimizado, é possível perceber uma melhoria considerável em relação ao não otimizado.

3.L3 Bandwidth



Utilizada para análise de cobertura da banda da cache L3, também é possível observar melhores resultados do código otimizado em relação ao não otimizado, principalmente para N's grandes.

4.L2 Cache-miss



Para a métrica de cache-miss na cache L2 não foi possível observar um comportamento conclusivo. Como melhoria para esta métrica, o código otimizado

poderia incluir na sua lista de técnicas de otimização o loop unroll, onde os dados em cache seriam melhor aproveitados e os “misses” seriam menos frequentes.

Conclusão

As modificações realizadas mostraram ter impacto considerável no desempenho do código, principalmente o armazenamento da matriz k-diagonal em uma estrutura própria ao invés de uma matriz.

Técnicas como o loop unroll e blocking, se implementadas, podem aumentar ainda mais a eficiência do código, em métricas que não sofreram tantas alterações, como o cache miss.