Relatório do Trabalho: Perfilagem de Código

Lucas da S. Inocencio

[1]Escola Politécnica da Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro – RJ – Brasil

lucas.inocencio@Poli.ufrj.br

1. Enunciado

Usando o programa no link:

`https://github.com/UoB-HPC/intro-hpc-jacobi/blob/master/jacobi.c`

para a solução de um sistema de equações lineares pelo método iterativo de Jacobi, faça:

- Compile e execute o programa para pelo menos 3 tamanhos de matrizes (objetivo: obter tempos significativos, pelo menos da ordem de minutos, para facilitar a observação dos efeitos das otimizações). Em seguida, escolha um tamanho para prosseguir.
- Recompile e execute o programa com otimização automática.
- Execute a perfilagem do programa usando o *gprof* ou o *Intel VTune*.
- Analise o resultado da perfilagem.
- Com base nos resultados da análise, tente realizar uma ação de otimização.
- Compare os tempos das versões (inicial, com otimização automática e após perfilagem).
- Faça um relatório descrevendo a sua experiência.

2. Desenvolvimento

Para cumprir o primeiro item, foram escolhidos os tamanhos de matrizes 1500, 1750 e 2000. Os seguintes resultados foram obtidos com a compilação usando *gcc jacobi.c*:

- Tamanho 1500: média 63.84s, variância 2.16s.
- Tamanho 1750: média 112.62s, variância 1.34.
- Tamanho 2000: média 154.22s, variância 6.16s.

Para prosseguir, foi escolhido o tamanho de matriz 2000. A compilação com otimização automática usando *gcc -Ofast jacobi.c* resultou no seguinte:

- Tamanho 2000 (Otimização automática): média 98.45s, variância 0.15s.

Em seguida, foi realizada a perfilagem do código usando o *gprof* com a compilação *gcc -pg -Ofast jacobi.c*. Os resultados foram os seguintes:

Com base na perfilagem, o código foi modificado para incluir uma otimização manual aprendida no primeiro trabalho, que consiste em multiplicar a matriz de forma contígua na memória, em vez de utilizar saltos. Portanto, trechos de código que antes eram escritos como `A[row + col*N]` foram alterados para `A[col + row*N]`. Isso resultou no seguinte resultado:

- Tamanho 2000 (Otimização manual): média 17.40s, variância 0.04s.

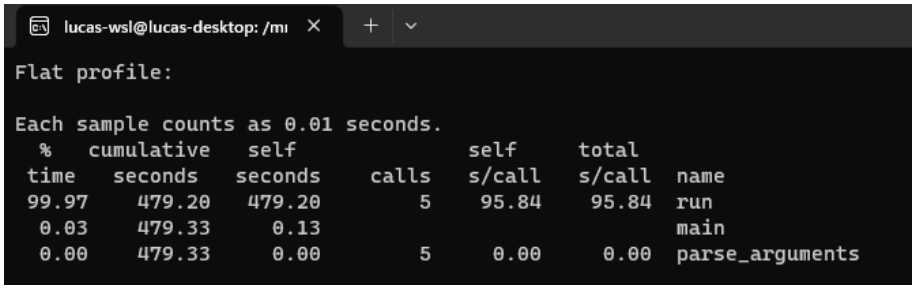

Figure 1. Perfilagem do programa com gprof

3. Resultados

Os resultados foram obtidos em um processador Ryzen 3400G, utilizando apenas 1 núcleo de CPU, 8 GB de RAM e o sistema operacional Windows 11. O gráfico a seguir apresenta os diagramas de caixa dos tempos obtidos:

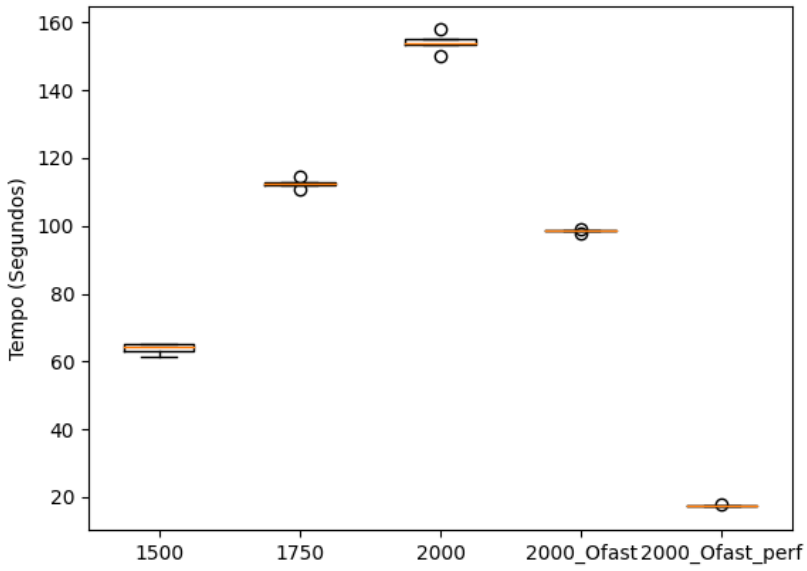

Figure 2. Boxplot dos tempos

4. Conclusão

De forma anedótica, é possível concluir que houve ganhos significativos ao alterar a estrutura do código, apesar de ser utilizado as opções de otimizações automáticas do compilador.