Aula prática 5 – Estrutura de Dados
Aluna: Raissa Gonçalves Diniz
Matrícula: 2022055823

Análise de desempenho de um programa, com foco na caracterização da complexidade de acesso à memória

O objetivo dessa prática é caracterizar um programa em termos de padrão de acesso à memória e localidade de referência. Para isso, foram utilizadas as seguintes funções para instrumentação:

- iniciaMemLog: registra o início do monitoramento;
- ativaMemLog e desativaMemLog: controlam o estado de ativação do monitoramento.;
- defineFaseMemLog: define a fase do monitoramento, caso seja necessário distinguir diferentes fases do algoritmo;
- leMemLog e escreveMemLog: registram leituras e escritas nas estruturas de dados monitorizadas;
- finalizaMemLog: registra o fim do monitoramento.

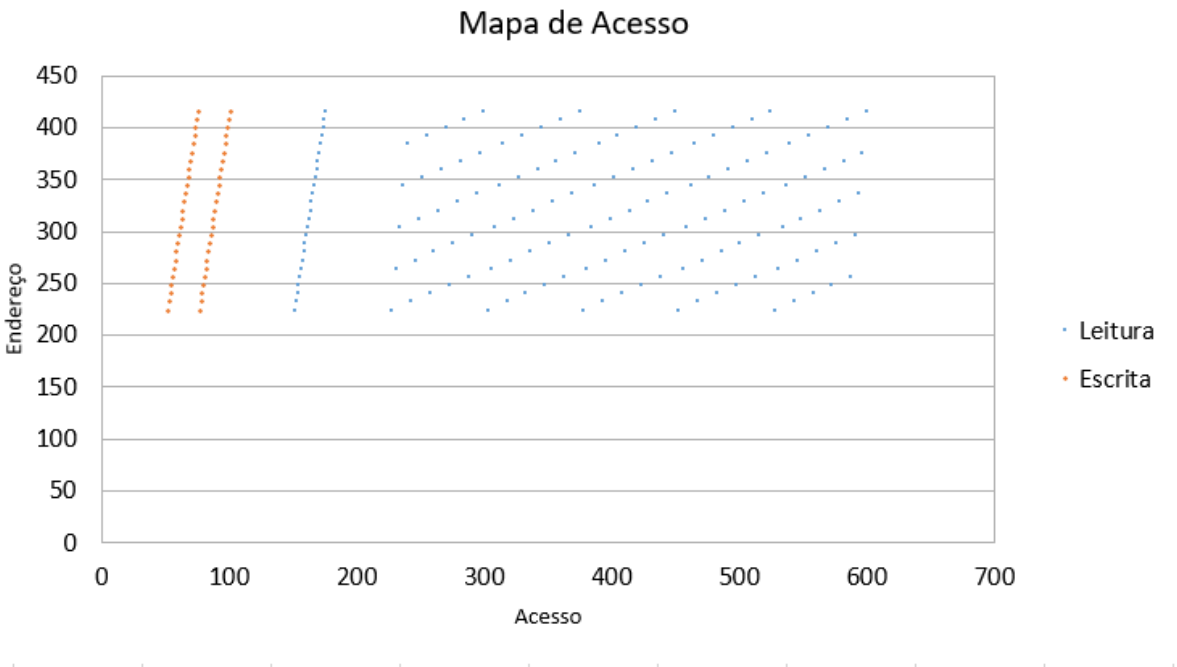
O código analisado foi provido no Moodle da disciplina, sob o nome “analisamem”. O teste de aplicação utiliza os arquivos exemplo gerados pela biblioteca memlog. São eles:

- multlog.out (multiplicação de matrizes)
- somalog.out (soma de matrizes)
- transplot.out (transposição de matrizes)

Para visualização, foram gerados gráficos para o mapa de acesso, histograma de distância da pilha e evolução da distância em relação a memória. Os gráficos se referem a operação de mutiplicação de matrizes.

1. Mapa de Acesso:

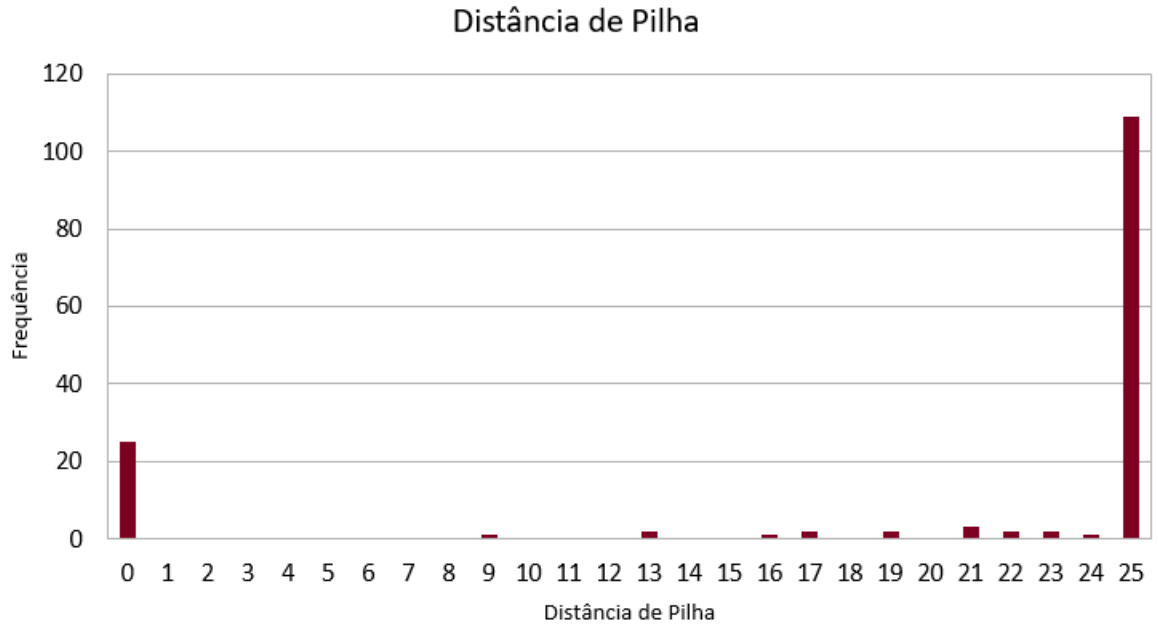
O Mapa de Acesso é uma representação gráfica dos padrões de acesso à memória durante a execução de um programa. Ele mostra onde e como os dados são lidos ou gravados na memória. Cada ponto no mapa representa um endereço de memória, e a intensidade da cor ou outro indicador visual pode representar a frequência ou a quantidade de acessos a esse endereço.



Nesse caso, o código realiza a escrita da matriz C, resultante da multiplicação, preenchendo os elementos dessa matriz. Isso é tipicamente feito em um loop que percorre as linhas e colunas, preenchendo a matriz sequencialmente, e resultando em um padrão linear, onde os elementos são escritos um após o outro. A mesma coisa acontece para a primeira leitura da primeira matriz. Contudo, nota-se um padrão diferente na leitura da segunda matriz. Isso ocorre porque, ao multiplicar as linhas da matriz A pelas colunas da matriz B, você acessa elementos da matriz B em um padrão específico. Cada elemento da matriz B é usado várias vezes em combinações diferentes com as linhas da matriz A, criando um padrão de acesso em blocos .Esses padrões refletem a estrutura da multiplicação de matrizes, onde as operações de multiplicação e soma são realizadas repetidamente entre elementos da matriz A e elementos da matriz B.

2. Histograma de Distância de Pilha:

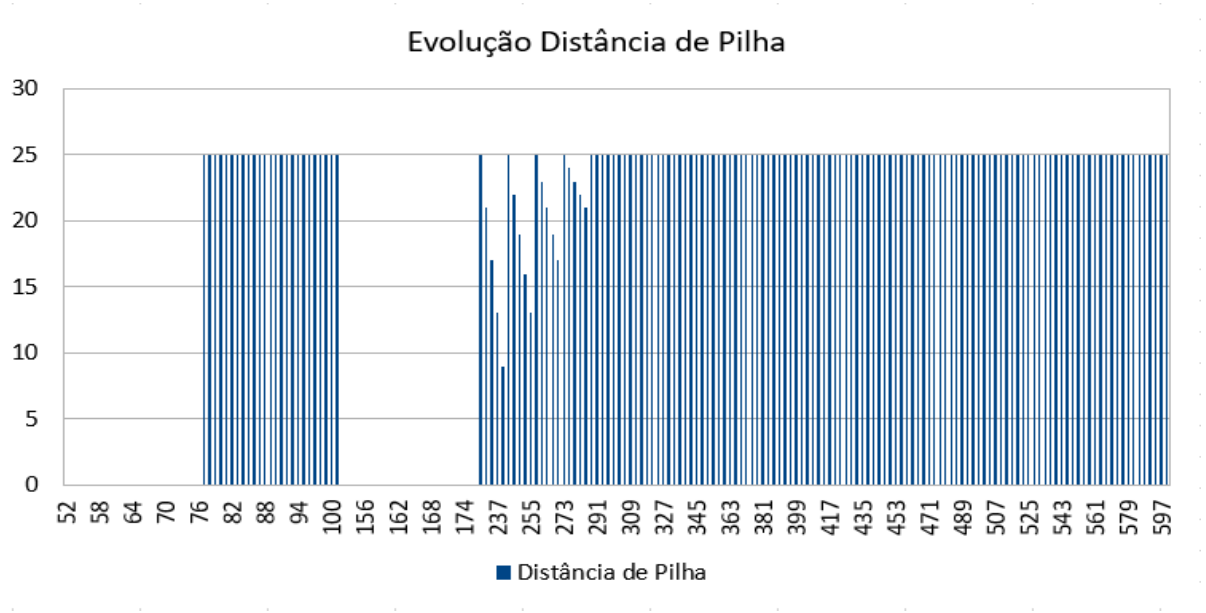
O Histograma de Distância de Pilha mostra a distribuição das distâncias entre os acessos à memória em relação a uma pilha de referências. Em outras palavras, ele analisa a sequência de acessos à memória e mede a distância entre cada acesso e o acesso anterior.



Quando a distância é igual a zero, significa que a função “multiplicaMatrizes” está acessando as matrizes a, b e c de forma sequencial, em um loop. Isso resulta em uma alta frequência porque os acessos à memória são próximos e frequentes.
Quando a distância atinge 25, observa-se um aumento significativo na frequência. Na multiplicação de matrizes, quando a distância é igual ao número de colunas (ou seja, 25 neste caso), você está acessando elementos da próxima linha da matriz. Isso pode resultar em uma mudança de bloco de memória, e é possível que a função esteja realizando esses acessos em blocos consecutivos de memória que não estão na cache. Como resultado, é possível observar um aumento na frequência de acesso à memória, já que a função está acessando regularmente elementos de matrizes distantes em termos de distância em relação aos blocos de memória.

3. Evolução da Distância em Relação à Memória:

A Evolução da Distância em Relação à Memória se refere ao acompanhamento da mudança nas distâncias entre acessos à memória ao longo do tempo durante a execução de um programa. Graficamente, isso é mostrado como as distâncias entre acessos à memória variam à medida que o programa é executado.



Nesse caso, percebe-se a presença de retas verticais paralelas e em alguns espaços pequenos traços entre essas retas. Creio que isso possa indicar um comportamento de alocação e desalocação de memória consistente e controlado, onde não há mudanças significativas na alocação e desalocação de memória na pilha relacionadas a esses identificadores. Os pequenos traços entre as retas verticais podem apresentar pequenas variações na distância de pilha em pontos específicos.

Otimização:
A transposição de matrizes poderia ajudar a otimizar esse código de multiplicação de matrizes, melhorando a eficiência do acesso à memória e, consequentemente, o desempenho da operação. Essa operação poderia melhorar a localidade de referência, pois a transposição reorganiza os elementos da matriz de forma que as operações subsequentes de multiplicação acessem elementos adjacentes com mais frequência. Na atual multiplicação de matrizes, os elementos da matriz são acessados linha por linha e coluna por coluna. Se as matrizes forem transpostas, a multiplicação pode ser realizada em um padrão mais natural de linha por linha, o que pode ser mais eficiente.