

Benchmark

Comparação de Multiplicação de Matrizes

Fernando Schneider Leandro
Eduardo Zardin Rehfeld

¹Uniritter – Porto Alegre – RS – Brasil

²Ciências da Computação – Paradigmas de Programação

nando.leandro@hotmail.com

eduardo_zardin@hotmail.com

1. Referencial Teórico

Esse capítulo tem por objetivo apresentar o referencial teórico e fundamentar seus conceitos.

1.1. Benchmarking

Benchmarking é o processo que consiste em comparar o mesmo processo entre distintas soluções, onde geralmente são medidos, tempo, qualidade e custo. A performance é quantificada a partir da relação entre estas três grandezas.

1.2. Multiplicação de Matrizes

Na matemática a multiplicação de matrizes é uma operação binária que tem como entrada duas matrizes e resulta em uma terceira, dada pela soma das multiplicações entre linhas e colunas. *Figura 1*

2. Objetivo

O experimento tem como objetivo testar três linguagens de programação diferentes para realizar a multiplicação de matrizes quadradas e medir o tempo de execução de cada programa para diferentes tamanhos de matrizes.

3. Solução

Esse capítulo tem como objetivo apresentar a solução e algoritmos usados para a criação e multiplicação de matrizes.

3.1. Tecnologias

Foi criado a partir da IDE Visual Studio 2019 Community uma solução (solution, arquivo.sln) contendo três projetos diferentes, nas linguagens C++, C# e Python. Dentro de cada projeto se encontra um arquivo que contém o código para execução do algoritmo de multiplicação de matrizes.

3.2. Criação das Matrizes

Para a primeira medição foram usadas matrizes de tamanho 10 a 350 com dados do tipo inteiro de 0 a 9. Cada linguagem tem suas peculiaridades para criação das matrizes porém a execução desse processo não foi incluindo nas métricas.

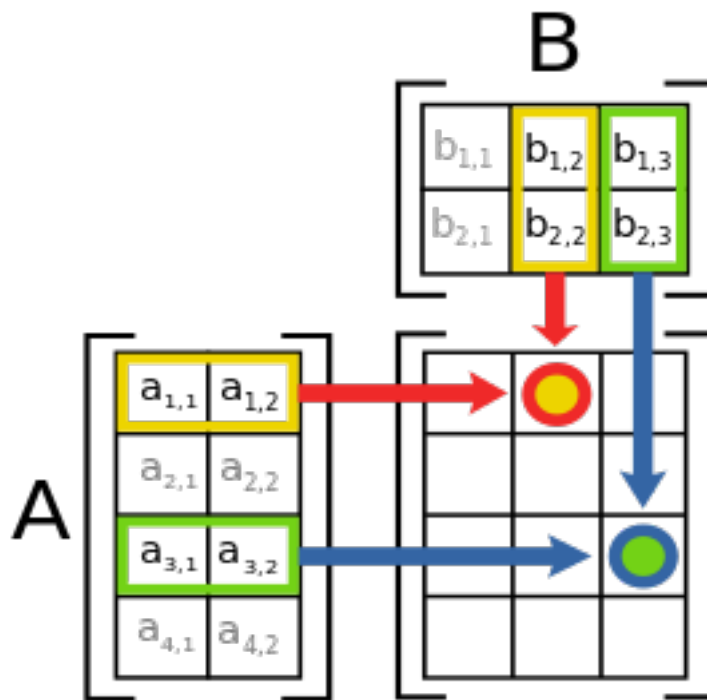


Figura 1. Multiplicação de Matrizes

3.3. Algoritmos

Para a multiplicação de matrizes foi usado o algoritmo básico de "nested loops", que é extremamente custoso pois consiste em três estruturas de repetições aninhadas. Também foi medido o tempo de execução na linguagem Python usando um algoritmo diferente vindo da biblioteca numpy que contém uma função específica para a operação.

4. Conclusão

Os algoritmos tempos medidos foram bem representativos, tanto em relação com a linguagem quanto ao algoritmo usado, o código escrito em C++ foi o mais performático usando o algoritmo de repetições aninhadas, porém foi observado que o algoritmo é extremamente importante na operação, a biblioteca numpy para a linguagem Python tem uma operação específica que utiliza um algoritmo muito mais performático.

Comparando as linguagens C++ e C# temos uma diferença muito maior, devido as camadas de abstração da linguagem C# torna a operação mais custosa, enquanto C++ com acesso de baixo nível a memória torna operações em listas e matrizes muito mais rápidas. Já em Python puro utilizando o algoritmo de repetições aninhadas a operação fica quase inviável devido a ser uma linguagem interpretada. AS medidas viáveis estão representadas nas figura 2 e figura 3

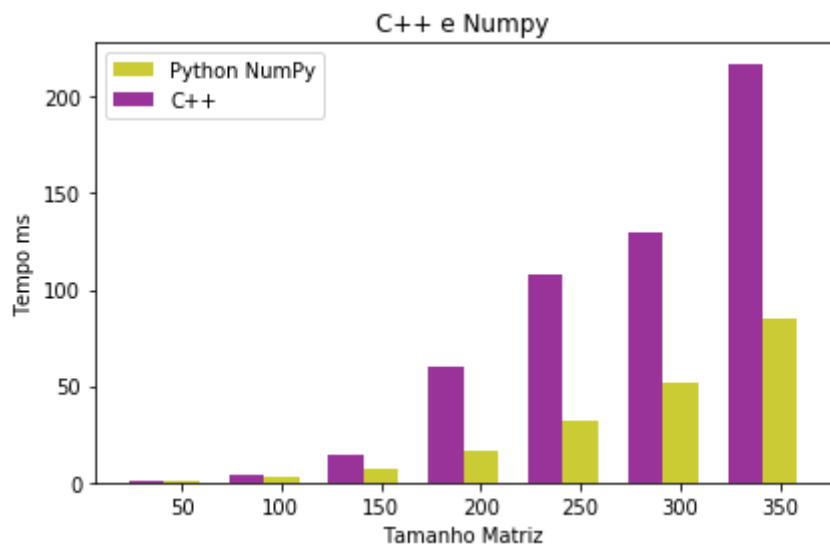


Figura 2. Python Numpy vs C++

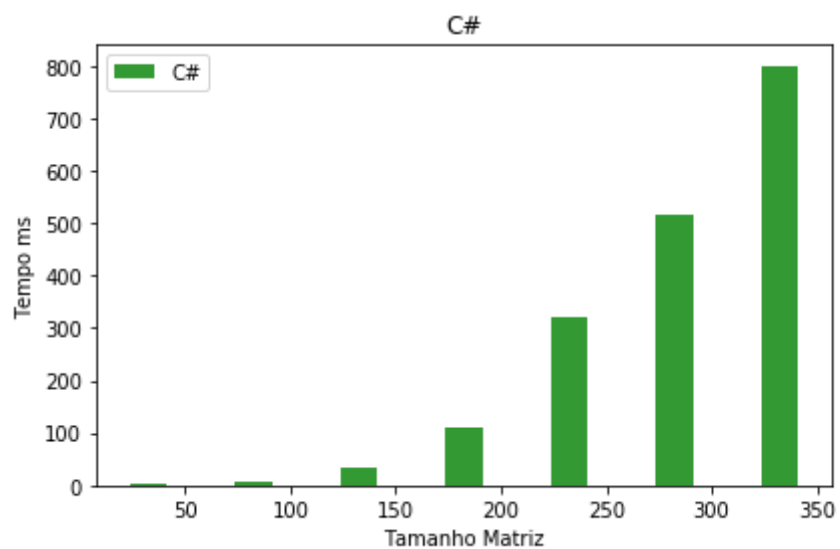


Figura 3. C#