

Computação de Alto Desempenho - COC472

Trabalho 1



UNIVERSIDADE FEDERAL
DO RIO DE JANEIRO

- Danyel Clinário dos Santos - DRE 119045123
– `danyel.clinario@poli.ufrj.br`

Rio de Janeiro - RJ
21 de Maio de 2022 (2022.1)

1 Introdução

Nesse trabalho, foram desenvolvidos 2 programas com o objetivo de calcular a multiplicação de uma matriz quadrada por um vetor. Cada programa possui 2 formas de iterar sobre os elementos da matriz e do vetor. A quantidade de tempo usado para realizar essa ação é cronometrada em cada método, sendo desenvolvido um programa na linguagem C e outro na linguagem Fortran90.

No programa em C, foram desenvolvidas duas funções extras, a primeira para preencher a matriz e vetor com valores aleatórios e outra para checar se os valores dos 2 métodos são iguais. Já no programa em Fortran90, o preenchimento foi feito no módulo principal, sem checagem dos valores após a multiplicação.

Os resultados foram guardados em um arquivo CSV, para posterior criação dos gráficos utilizando um programa em Python. Todo o código se encontra no **Github** ou **Replit**.

2 Questões

2.1 Estime o tamanho máximo dos *arrays* **A**, **x** e **b** que podem ser alocados no seu sistema para realização da tarefa.

Utilizando variáveis de ponto flutuante com dupla precisão, usaríamos $(n^2 + 2n) * 64$ bits para alocar as variáveis com dimensão n . Considerando meu computador com 12GB RAM ($12 * 10^9 * 8$ bits), teríamos que o valor máximo de n quando:

$$(n^2 + 2n) * 64 = 12 * 10^9 * 8, \quad (1)$$

ou seja, encontrar a raiz positiva da equação do segundo grau:

$$n^2 + 2n - \frac{12 * 10^9 * 8}{64} = 0 \quad (2)$$

No meu caso, temos o máximo de n quando $n = 38.729$. Vale notar que esse valor não será alcançado, já que o sistema operacional, além de outras tarefas ativas consomem parte dessa mesma memória.

2.2 Os *arrays* a serem utilizados durante as operações devem ser inicializados com números aleatórios (**A** e **x** no caso acima).

No programa em C, foi desenvolvida uma função para preencher a Matriz e os vetores, enquanto que no programa desenvolvido em Fortran90 o preenchimento foi realizado no módulo principal. O tempo dessas ações, bem como da função criada para checar se os valores calculados nos 2 métodos são iguais, não é levado em consideração durante a cronometragem.

2.3 Comece com um problema de tamanho pequeno e tente chegar ao tamanho máximo estimado no **item 1**. Contabilize o tempo para realização das operações para todos os tamanhos de sistema e para ambas as ordens de execução dos *loops*.

Os programas recebem como parâmetro um lista de quantidade de RAM em GB que será usada, por exemplo: `./c_code 1 1.5 2`

O comando acima calculará qual o valor máximo de n quando é disponibilizado 1GB de RAM, e depois o mesmo para 1.5GB, etc. Isso permite um maior controle da quantidade de RAM usada.

2.4 Apresente curvas mostrando o tempo de execução para cada dimensão do problema e relacione estas curvas à complexidade computacional do produto matriz - vetor ($O(n^2)$).

Seguindo a ideia apresentada no **item 2.3**, foram obtidos os seguintes resultados utilizando como entrada a lista no intervalo $[0.1, 8)$, acrescentando 0.3GB por rodada:

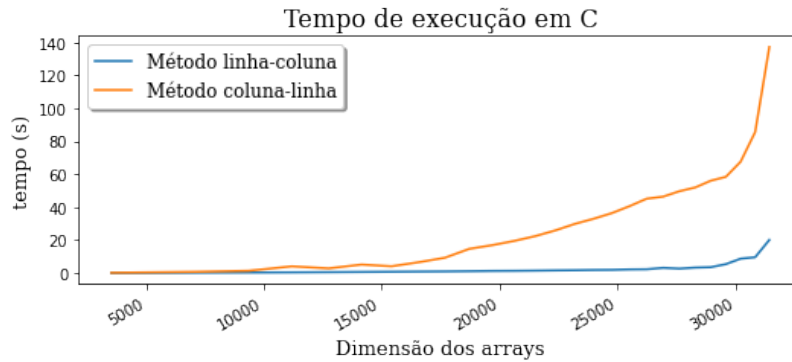


Figura 1: Tempo de execução do programa na linguagem C.

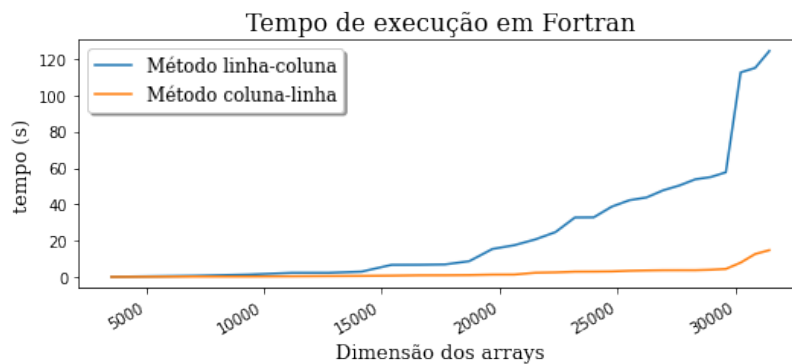
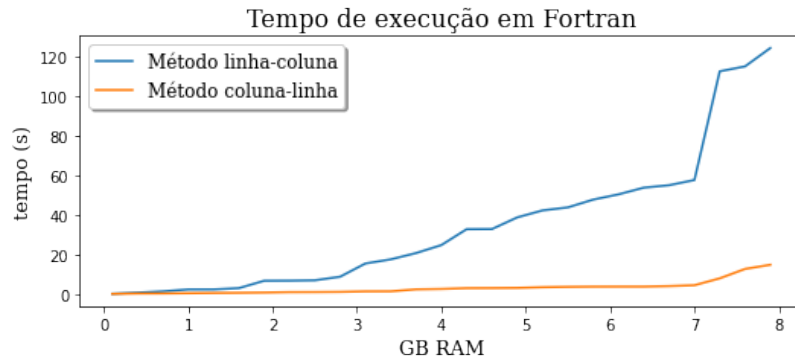


Figura 2: Tempo de execução do programa na linguagem Fortran.

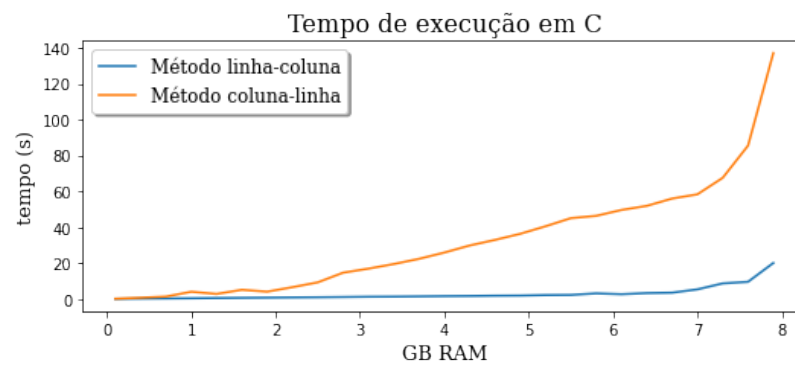
Vale ressaltar que o limite superior foi definido por tentativa e erro, ou seja, o maior valor possível antes do computador travar ou o código ser finalizado pelo sistema operacional.

2.5 Explique como o modo em que os *arrays* são armazenados nas duas linguagens afetam os resultados.

O Fortran armazena os dados de forma que os valores em colunas são armazenados sequencialmente em memória. Com isso, tem-se um acesso mais rápido e eficaz quando percorremos em loops do tipo “coluna-linha”. Já na linguagem C, os dados são armazenados de um modo que os valores das linhas são armazenados sequencialmente em memória. Logo, tem-se um acesso mais rápido e eficaz quando percorremos em loops do tipo “linha-coluna”



(a) Tempo de execução do programa na linguagem Fortran.



(b) Tempo de execução do programa na linguagem C.

Figura 3: Gráficos levando em consideração a quantidade de memória usada.

3 Referências

Quick Start sobre Fortran
Dezenas de páginas do Stackoverflow
Quick Start sobre C
GNU Fortran Compiler
Fortran 77 Language Reference Oracle
Funções em Fortran - Stanford