Universidade Federal do Rio de Janeiro

Computação de Alto Desempenho - NACAD

Raphael Sathler - 111344078

Projeto 1

# Objetivo

O objetivo desse trabalho consiste em *codar* um algoritmo multiplicação de matrizes densas nas linguagens C e Fortran utilizando um *loop* de *fors*.

Para tal, faz-se necessário aumentar o tamanho de *n*, onde *n* é o tamanho da matriz quadrada, e anotando então o tempo de execução do algoritmo, desconsiderando o tempo de inicialização das matrizes.

Por fim, deve-se deliberar a complexidade desse algoritmo e o tempo de execução dos programas, além de comparar as diferenças entre as linguagens.

# Metodologia

Para a realização desse trabalho, utilizou-se uma máquina virtual com 8 cores, 12 Gb de memória RAM virtualizada no VMWare Workstation com o Ubuntu 17.10 64 bits como Sistema Operacional.

Em Fortran, utilizou-se dois programas diferentes, o primeiro, usando um *for* triplo, e a segunda, utilizando uma função interna do Fortran *matmul* [[1]](#footnote-1)que realiza a multiplicação de matrizes. Já em C, utilizou-se apenas o *for* triplo. Vale ressaltar que para todas as execuções, utilizou-se o grau máximo de otimização.

Por não haver paralelismo, a única diferença encontrada, **teoricamente**, entre rodar em uma máquina com mais ou menos *cores* é a capacidade de processamento dos núcleos, em geral, medida pela quantidade de instruções por segundo que eles são capazes de executar.

Para verificar e garantir isso, rodei os programas de Fortran em uma máquina com 24 cores e 32Gb de memória RAM, e a única diferença encontrada é que esta máquina suporta matrizes de tamanho maiores por possuir muito mais memória, entretanto o tempo de execução é aproximadamente o mesmo para as matrizes de tamanho em comum.

Para garantir resultados mais precisos, realizou-se a média entre 5 rodadas para cada tamanho de variável para cada programa. Os tamanhos de variáveis escolhidos foram 3, 10, 50, 100, 500, 1000, 1500, 2000, 5000, 10000, 15000, 20000.

# Códigos

Pela clareza dessa seção e para falarmos apenas sobre oque é realmente interessante, mostraremos somente a parte diferente do código que é relevante será mostrada aqui. O códigos em si são muito semelhantes, onde a diferença encontra-se somente na linguagem, praticamente.

## Fortran *For* Triplo

call cpu\_time(start)

m\_res=0.0D0

do j=1, n

do i=1, n

do k=1, n

m\_res(i,j)=m\_res(i,j)+m1(i,k)\*m2(k,j)

enddo

enddo

enddo

call cpu\_time(finish)

print '("Time = ",f15.3," seconds. Started:",f6.3," - Finished:",f15.3,".")',finish-start,start,finish

Código 1 - Fortran com For Triplo

Nesse código, utiliza-se 3 *fors* para realizar a multiplicação das matrizes.

## Fortran Matmul

Código 2 - Fortran com Matmul

call cpu\_time(start) ! Start timing

m\_res = matmul(m1, m2) ! Use the matmul function

call cpu\_time(finish) ! Finish timing

print '("Time = ",f15.3," seconds. Started:",f6.3," - Finished:",f15.3,".")',finish-start,start,finish

Já nesse código, utilizou-se a função *matmul*, integrante do Fotran 95, uma função que, por si só, realiza a multiplicação de matrizes, tendo como argumentos duas matrizes.

## C

Código 3 - C

m\_res = (float \*\*)calloc(N, sizeof(float \*)); // Allocate an initial array for matrix

for (int i = 0; i < N; i++)

m\_res[i] = (float \*)calloc(N, sizeof(float)); // Allocate the array for each array (now, it's N by N)

t = clock(); // Starts the time

float tmp;

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

tmp = 0.0;

for (k = 0; k < N; k++)

tmp += m1[i][k] \* m2[k][j]; // Computes the product

m\_res[i][j] = tmp; // Assign value to the matrix

}

}

double exec\_time = ((double)t)/CLOCKS\_PER\_SEC; // Finishes the time

O código em C realiza a alocação do espaço em memória e computa o produto da matriz.

Todo os códigos usados neste trabalho foram hospedados no [Bitbucket](https://bitbucket.org/sathler/cad/src/master/).

# Resultados

Para o Fortran, ao utilizar um *for* triplo, o tempo de execução foi maior que o tempo de execução utilizado pela função *matmul*. Possivelmente, essa função possui uma série de otimizações para favorecer a sua execução. Em comparação com o C, os tempos iniciais, *i.e.*, *n* menores, foi muito mais demorado que em Fortran, numa ordem de 1000 vezes mais lenta. Entranto, para valores maiores de *n*, o tempo de instanciação das matrizes em C foi muito mais lento, entretanto, o tempo de execução da multiplicação das matrizes foi melhor que em Fortran, ainda usando o *matmul*.

Como pode ser observado, o uso da função mat*mul* garantiu uma aproximação de uma exponencial melhor que os outros programas. Usando o *for* triplo no fortran, conseguimos calcular até *n* = 5000 ao custo de 779s, entretanto, no C, com o mesmo valor máximo computado para *n*, foi preciso de apenas 0,21s.

Ao fazer as comparações, esse algoritmos tem complexidade O(n²) pela sua essência. E a teoria condisse com a prática. Infelizmente, não foi possível computador valores maiores de *n*, mas se fosse, poderíamos observar o crescimento similar à n².

Gráfico - Gráfico de Comparação entre os Programas

# Conclusão

Conforme esperado, os tempos de execução dos algoritmos foram condizentes com o esperado teórico. Além disso, observou-se que a execução do fortran foi muito boa, em geral, melhor que a em C.

Por fim, questiono-me quanto à diferença no tempo de execução entre o processo de execução do Fortran e do C: como houve tanta discrepância no tempo encontrado com os resultados obtidos sendo coerentes entre os dois?

1. https://gcc.gnu.org/onlinedocs/gfortran/MATMUL.html [↑](#footnote-ref-1)