

**Universidade de São Paulo  
Instituto de Ciências Matemáticas e de Computação  
Departamento de Sistemas de Computação  
Laboratório de Sistemas Distribuídos e de Programação Concorrente**

***Caderno de Desafio para Programação Paralela***

***Caderno Desafio C/CUDA  
Soma Produtos Escalares, Mínimo e Máximo Duas Matrizes Quadradas***

por

*Paulo Sérgio Lopes de Souza*

Baseado em desafio anterior feito por Guilherme Martins e Paulo S. L. de Souza

Este caderno de desafio representa um Recurso Educacional Aberto para ser usado por alunos e professores, como uma introdução aos estudos de programação paralela com C e *CUDA*. Este material pode ser utilizado e modificado desde que os direitos autorais sejam explicitamente mencionados e referenciados. Utilizar considerando a licença *GPL (GNU General Public License)*.

São Carlos/BR, dezembro de 2020

1. Desafio	2
1.1. Calcular em C/CUDA a soma dos produtos escalares, o mínimo e o máximo de duas matrizes quadradas.	2
3. Um ponto de partida para a solução do desafio	3
3.1 Implementação sequencial do desafio	3

# 1. Desafio

## 1.1. Calcular em C/CUDA a soma dos produtos escalares, o mínimo e o máximo de duas matrizes quadradas.

Considere as matrizes A[Dim\*Dim] e B[Dim\*Dim].

Matriz A [3, 3]

11	2	3
4	5	6
7	8	9

Matriz B [3,3]

19	8	7
6	5	4
3	2	1

Os produtos escalares entre as linhas de A e B são:

$$11*19+2*8+3*7=246$$

$$4*6+5*5+6*4=73$$

$$7*3+8*2+9*1=46$$

A soma destes produtos é **365**.

O valor mínimo das duas matrizes é **1** (um valor mínimo para as duas matrizes).

O valor máximo das duas matrizes é **19** (um valor máximo para as duas matrizes).

Faça um programa em C/CUDA considerando os conceitos vistos até agora nas aulas, para calcular a soma dos produtos escalares das linhas das duas matrizes quadradas do tipo **int**, conforme o exemplo acima. Encontre também o valor máximo e o valor mínimo existentes nas duas matrizes

Considere como entrada um arquivo de texto contendo, na primeira linha, a dimensão (*dim*) das matrizes. A partir da segunda linha até a linha *dim+1*, estão os elementos da matriz A, do tipo **int**, onde as linhas são separadas por uma quebra de linha simples e as colunas por um único espaço. A partir da linha *dim+2* até o fim do arquivo, estão os elementos da matriz B, também do tipo **int**.

Conteúdo do arquivo de entrada, por exemplo, **entrada.txt**:

```
3
11 2 3
4 5 6
7 8 9
19 8 7
6 5 4
3 2 1
```

Para executar no **bash**, por exemplo, utilize este padrão:

```
.\desafio entrada.txt <enter>
```

Obs: na linha de comando acima, considera-se que o programa foi inserido em **desafio.cu** e o executável chama-se **desafio** e está no diretório atual.

A saída deve ser impressa, utilizando o *output* (**stdout**) padrão, contendo a soma dos produtos escalares, seguido do valor mínimo e do valor máximo (todos separados por um espaço). Não há espaço após o valor máximo. Há uma quebra de linha após o valor máximo.

```
365 1 19
```

### 3. Um ponto de partida para a solução do desafio

#### 3.1 Implementação sequencial do desafio

OBS: este código está disponível no arquivo fonte em anexo.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
```

```
int main(int argc, char **argv)
{
    //Declara as matrizes
    int *A, *B;

    //Declara as variáveis de índice
    int i,j,k,dim;
```

```

//Declara o acumulador para o produto escalar global
int somape, minimo, maximo;

//Declara um vetor para os produtos escalares locais
int *prod_escalar;

FILE *inputfile; // handler para o arquivo de entrada
char *inputfilename; // nome do arquivo de entrada

if (argc < 2)
{
    printf("Please run with input file name, i.e., num_perf_mpi inputfile.ext\n");
    exit(-1);
}

inputfilename = (char*) malloc (256*sizeof(char));
strcpy(inputfilename,argv[1]);

// printf("inputfilename=%s\n", inputfilename);
// fflush(0);

if ((inputfile=fopen(inputfilename,"r")) == 0)
{
    printf("Error openning input file.\n");
    exit(-1);
}

fscanf(inputfile, "%d\n", &dim); //Lê a dimensão das matrizes

//Aloca as matrizes
A=(int *)malloc(dim *dim * sizeof(int));
B=(int *)malloc(dim *dim * sizeof(int));

//Aloca um vetor para armazenar os produtos escalares de cada linha
prod_escalar=(int *)malloc(dim * sizeof(int));

//Lê a matriz A
for(i=0;i<dim;i++)
{
    for(j=0;j<dim;j++)
    {
        fscanf(inputfile, "%d ",&(A[i*dim+j]));
    }
}

```

```

    }
}

//Lê a matriz B
for(i=0;i<dim;i++)
{
    for(j=0;j<dim;j++)
    {
        fscanf(inputfile, "%d ", &(B[i*dim+j]));
    }
}

// fecha o arquivo de entrada
fclose(inputfile);

//Calcula o produto escalar de cada linha
for(i=0;i<dim;i++)
{
    for(j=0;j<dim;j++)
    {
        prod_escalar[j] += A[i*dim+j]*B[i*dim+j];
    }
}

//Acumula os produtos das linhas (faz o produto escalar global)
somape=0;
for(i=0;i<dim;i++)
{
    somape += prod_escalar[i];
}

//encontra o mínimo da matriz A
minimo = A[0];
for(i=0;i<dim;i++)
{
    for(j=0;j<dim;j++)
    {
        if (A[i*dim+j] < minimo)
        {
            minimo = A[i*dim+j];
        }
    }
}
}

```

```

//encontra o mínimo da matriz B
for(i=0;i<dim;i++)
{
    for(j=0;j<dim;j++)
    {
        if (B[i*dim+j]< minimo)
        {
            minimo = B[i*dim+j];
        }
    }
}
//encontra o máximo da matriz A
maximo = A[0];
for(i=0;i<dim;i++)
{
    for(j=0;j<dim;j++)
    {
        if (A[i*dim+j] > maximo)
        {
            maximo = A[i*dim+j];
        }
    }
}
//encontra o máximo da matriz B
for(i=0;i<dim;i++)
{
    for(j=0;j<dim;j++)
    {
        if (B[i*dim+j] > maximo)
        {
            maximo = B[i*dim+j];
        }
    }
}
//Imprime o resultado
printf("%d %d %d\n",somape, minimo, maximo);
//Libera as matrizes
free(A);
free(B);
//Libera o vetor
free(prod_escalar);
}

```