

T2 Aplicação prática em C/MPI para o BCC-T2

Dada uma matriz M quadrada de ordem N composta por números inteiros, construa um algoritmo concorrente em C/MPI, o qual deve determinar qual é a maior diferença de um dado número da matriz em relação aos seus (até 8) vizinhos mais próximos. Imprima as posições destes dois números na matriz (linha x coluna), e os valores destes números na matriz, como saída dessa aplicação. Os números de M deverão ser carregados a partir de um arquivo texto no início da execução. A aplicação concorrente executará com P processos.

Considere este exemplo com uma matriz M 4x4.

Arquivo de entrada:

4			
8	2	30	10
5	1	2	0
10	90	50	75
7	15	35	15

Arquivo de saída:

$M[2,1]=90$ $M[1,1]=1$

Faça o seu particionamento por dados e considere o uso de P processos C/MPI, onde P é o número de processadores disponíveis na plataforma.

Descreva de maneira sucinta o seu particionamento, comunicação, aglomeração e mapeamento como comentário no código fonte.

Coloque como comentário no código fonte o número da equipe e os nomes dos integrantes.

Exemplo do algoritmo sequencial a ser usado nesta aplicação: vide moodle.

Código sequencial turma B

// Based on Author: Ana Caroline Spengler

```
#include<stdio.h>
#include<stdlib.h>
#include <limits.h>
#include <math.h>
```

```
int calc_diferenca(int **matriz, int linha_maior,int coluna_maior,int *linha_menor,int
*coluna_menor,int *valor_maior,int *valor_menor){
```

```
    int dif, aux_dif = INT_MIN;
    int i, j;
```

```
    for (i = linha_maior - 1; i < linha_maior + 2; i++) {
        for (j = coluna_maior - 1; j < coluna_maior + 2; j++) {
            if (!(i < 0 || i > linha_maior || j < 0 || j > coluna_maior)){
                aux_dif = abs(matriz[linha_maior][coluna_maior] - matriz[i][j]);
                if (aux_dif > dif){
                    dif = aux_dif;
                    *linha_menor=i;
                    *coluna_menor=j;
                    *valor_maior=matriz[linha_maior][coluna_maior];
                    *valor_menor=matriz[i][j];
                }
            }
        }
    }
    return dif;
}
```

```
int main(void){
```

```
    FILE *arquivo_entrada,*arquivo_saida;
    arquivo_entrada=fopen("numeros.txt","r");
    arquivo_saida=fopen("saida.txt","w+");
```

int

```
    linha_menor_aux,coluna_menor_aux,linha_maior_aux,coluna_maior_aux,valor_maior_aux,v
    alor_menor_aux,tam,i,j=0;
```

```
    int linha_menor,coluna_menor,linha_maior,coluna_maior,valor_maior,valor_menor=0;
    fscanf(arquivo_entrada, "%d\n", &tam);
    int **matriz=((int**)malloc(tam*sizeof(int*)));
```

```
    for(i=0;i<tam;i++){
        matriz[i]=((int*)malloc(tam*sizeof(int)));
    }
```

```

for(i=0;i<tam;i++){
    for(j=0;j<tam;j++){
        fscanf(arquivo_entrada, "%d\n", &(matriz[i][j]));
    }

}
int aux_comp;
int diferenca = INT_MIN;
int pos_i, pos_j;

for (i = 0; i < tam; i++) {
    for (j = 0; j < tam; j++) {
        aux_comp = calc_diferenca(matriz, i,
j,&linha_menor_aux,&coluna_menor_aux,&valor_maior_aux,&valor_menor_aux);
        if (diferenca < aux_comp){
            diferenca = aux_comp;
            linha_maior=i;
            coluna_maior=j;
            linha_menor=linha_menor_aux;
            coluna_menor=coluna_menor_aux;
            valor_maior=valor_maior_aux;
            valor_menor=valor_menor_aux;
        }
    }
}
for(i=0;i<tam;i++){
    free(matriz[j]);
}

fprintf(arquivo_saida, "M[%d,%d]=%d
M[%d,%d]=%d",linha_maior,coluna_maior,valor_maior,linha_menor,coluna_menor,valor_me
nor);
free(matriz);
fclose(arquivo_entrada);
fclose(arquivo_saida);
}

```