

Relatório Aula 11 - Grupo 11

João Paulo Lins

Christian Braga Kedor

1 . Neste exercício a ideia era paralelizar o ex1-seq.c de modo a encontrar a primeira ocorrência em cada um dos vetores (A,B, C, D) de um valor escolhido pelo usuário. Para isso, utilizou-se o comando MPI_Bcast para enviar o valor desejado a cada um dos processos, os quais fazem uma busca do valor procurado e retornam o índice da primeira ocorrência.

```
if(rank == 0){
    printf("Numero :");
    fflush(stdout);
    scanf("%d",&numero);
}

MPI_Bcast(&numero,1,MPI_INT,0,MPI_COMM_WORLD);
```

Para finalizar, o rank principal recebe os valores através da função MPI_Gather, a qual junta os valores enviados em um vetor (vet_índices no problema em questão).

```
MPI_Gather(&indice,1,MPI_INT,vet_indices,1,MPI_INT,0,MPI_COMM_WORLD);
```

O resultado da execução pode ser visto a seguir, em que todos os vetores são inicializados como o vetor A abaixo, apenas para ilustração.

```
ckedor@ckedor-VirtualBox:~/Desktop/Aula 11$ mpirun -np 4 exercicio1
Numero :17
A: 3 - 6 - 17 - 15 - 13 - 15 - 6 - 12 - 9 - 1 -
Vetor[0]: 2
Vetor[1]: 2
Vetor[2]: 2
Vetor[3]: 2
```

2. Exercício 2 consiste em calcular o máximo de uma matriz dividindo em tasks. Essa divisão será feita utilizando a função MPI_Scatter. Cada task vai executar o processamento em um trecho da matriz (para np = 4, a matriz será dividida em 4). Cada task calcula o max_parcial e o resultado final é o máximo entre eles.

Execução:

```
ckedor@ckedor-VirtualBox:~/Desktop/Aula 11$ mpirun -np 4 exercicio2
INICIO
INICIO
INICIO
INICIO
rank=0 max_parcial=771.000000
rank=1 max_parcial=62.000000
rank=3 max_parcial=853.000000
rank=2 max_parcial=135.000000
RESULTADO=853.000000
```