



DISCIPLINA	Computação Paralela e Distribuída		
CURSO	Engenharia Informática		
DISCENTE			
Nº MATRÍCULA		TURMA:	DATA:

- Não é permitido o uso de dispositivos electrónicos
- Responda apenas o que é perguntado
- Justifique todas as suas respostas

### Grupo I ( 2.0 valores)

#### Arquitectura de Memória Distribuída

1. Discuta sobre a melhor implementação possível da função broadcast nos seguintes sistemas:

- A. Um sistema de memória partilhada distribuída (DSM).
- B. Um sistema de memória distribuída (multicomputador).

### Grupo II (4 + 4 + 2 = 10.0 valores)

#### Programação concorrente com MPI

1. O MPI\_Scatter utiliza os seguintes parâmetros:

```
MPI_Scatter(void *sendbuf, int sendcnt, MPI_Datatype sendtype,  
void *recvbuf, int recvcnt, MPI_Datatype recvtype, int root,  
MPI_Comm comm)
```

- a) Escreva a implementação mais trivial da função MPI\_Scatter em C. Utilize apenas o MPI\_Send e MPI\_Recv como funções de comunicação.

Funções que podem ser úteis:

```
int MPI_Send(void *buf, int count, MPI_Datatype datatype, int  
dest, int tag, MPI_Comm comm)
```

```
int MPI_Recv(void *buf, int count, MPI_Datatype datatype, int  
source, int tag, MPI_Comm comm, MPI_Status *status)
```

```
int MPI_Comm_size( MPI_Comm comm, int *size)
```

```
int MPI_Type_size(MPI_Datatype datatype, int *size)
```

```
void *memcpy(void *dst, const void *restrict src, size_t n)
```

- b) Em uma boa implementação de MPI, discuta a complexidade da função para os dados de tamanho n e número de processadores p.

2. Considere que temos o seguinte programa MPI a executar em 2 processadores, N é igual a 6.

```

1. MPI_Comm_rank (MPI_COMM_WORLD, &id);

2. if (id == 0)
3.     for (i = 0; i < N; i++)
4.         a[i] = i;

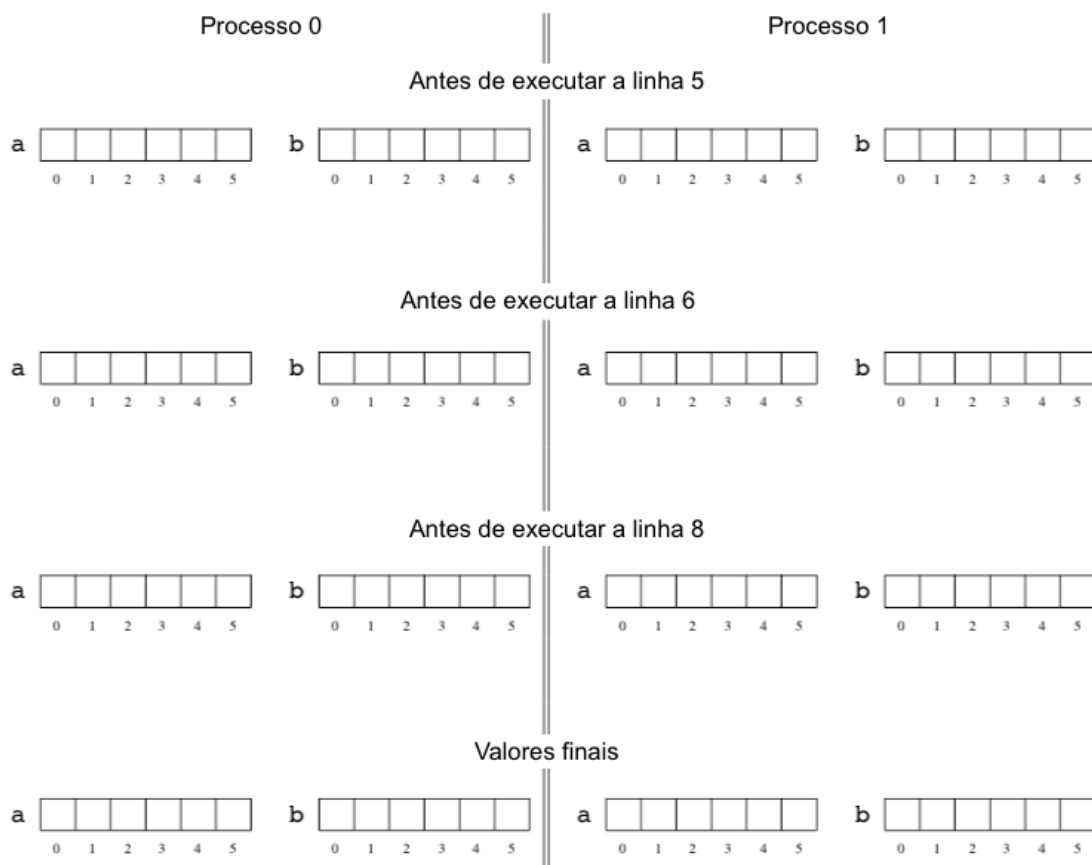
5. MPI_Bcast(a, N, MPI_INT, 0, MPI_COMM_WORLD);

6. for (i = 0; i < N; i++)
7.     b[i] = a[i] + id;

8. MPI_Alltoall(b, N, MPI_INT, a, N, MPI_INT, MPI_COMM_WORLD);

```

Indique os valores das matrizes a e b em cada processador nos seguintes pontos de execução. (utilize ? quando não estiver definido)



3. Considere o seguinte segmento de código MPI:



```
if (my_rank == 0) {  
    mydata = 2.5;  
    MPI_Bcast(&mydata, 1, MPI_FLOAT, 0, MPI_COMM_WORLD);  
}  
else  
    MPI_Bcast(&mydata, 1, MPI_FLOAT, 0, MPI_COMM_WORLD);
```

Este código está correcto? Em caso afirmativo, indique o valor de mydata no processo 1 antes e depois da execução. Se não, sugira uma implementação correcta.

### Grupo III (1.5 + 1.5 + 2.0 + 2.0 + 1 = 8.0 valores)

#### Análise de desempenho de programa paralelo

1. Qual é o significado de obter 0 para a métrica de fracção serial determinada experimentalmente ao avaliar um programa em 2, 4, 8 e 16 processadores?

2. Discuta a seguinte afirmação:

“Existem problemas para os quais é possível obter um algoritmo paralelo com eficiência acima de 100%.”

3. Considere um problema com um algoritmo sequencial que executa em  $\Theta(n)$  e com uma implementação paralela cujo overhead (comunicação + computação redundante) por processador é dado, respectivamente, por  $\Theta(\log p)$  e  $\Theta(n/p)$ . A sobrecarga paralela é  $\Theta(p \log p)$ . Calcule a função de escalabilidade para este algoritmo paralelo. Discuta o resultado obtido.

4. Considere que para  $p = 8$  processadores a “fracção serial determinada experimentalmente” avaliada em 0,2. É razoável esperar que para  $p = 16$  avalie 0,1? Se sim, o que isso significa? Se não, justifique.

5. A função de escalabilidade de uma aplicação paralela foi determinada como  $M(f(p))/p = p \log p$ . É um indicador bom ou ruim? Forneça uma justificativa cuidadosa em termos da definição da função de escalabilidade.

Bom trabalho!