

PCD: Quiz 9

O e-mail do participante (**igor.ribeiro@unifesp.br**) foi registrado durante o envio deste formulário.

Indique entre as alternativas abaixo uma possível desvantagem de usar a comunicação coletiva. *

- ☒ A comunicação coletiva pode dificultar a coordenação de mensagens entre um grande número de processos, podendo causar deadlocks.
- ☐ A comunicação coletiva é incompatível com a comunicação ponto a ponto.
- ☐ Chamadas de comunicação coletiva podem resultar em sobrecarga de sincronização.

Resposta correta

- ☒ Chamadas de comunicação coletiva podem resultar em sobrecarga de sincronização.



Indique entre as alternativas abaixo qual chamada deve ser usada quando uma sincronização simples através de um comunicador for necessária. *

- ☐ MPI_REDUCE
- ☒ MPI_BARRIER
- ☐ MPI_BROADCAST
- ☐ MPI_SCAN

Algumas rotinas de comunicação coletiva, como o SCATTER, possuem duas formas de implementação. Pesquise o funcionamento da versão com 'V' e responda: em uma chamada para MPI_SCATTERV, é necessário que todos os dados a serem enviados sejam contíguos no buffer de envio? *

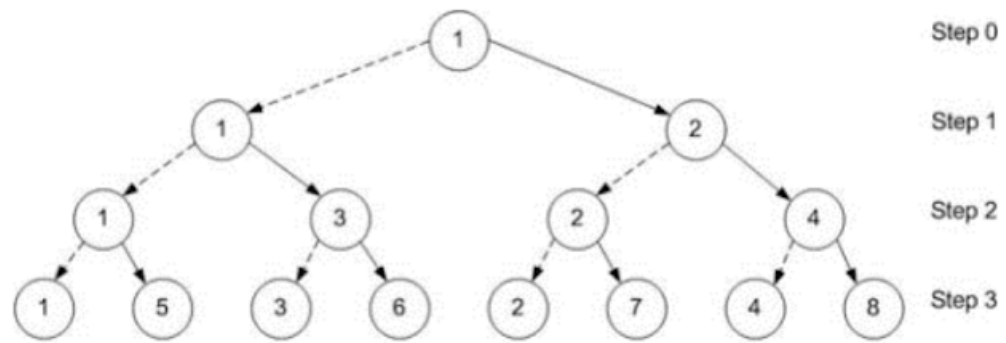
- ☐ Não, várias seções de dados podem ser enviadas em qualquer mensagem
- ☒ Não, mas as mensagens individuais devem ser compostas de seções contíguas do buffer de envio
- ☐ Sim, o buffer de envio deve ser organizado de forma que todos os dados enviados sejam contíguos antes da chamada para MPI_SCATTERV

Qual das operações abaixo pode ser considerada o inverso da operação SCATTER? *

- ☒ MPI_GATHER
- ☐ MPI_RECV
- ☐ MPI_BROADCAST
- ☐ MPI_REDUCE



As rotinas de comunicação coletiva costumam ser mais eficientes porque elas próprias implementam algoritmos eficientes para a distribuição das mensagens. O esquema abaixo apresenta implementação da rotina BROADCAST usando um algoritmo em "árvore". Responda: considerando um comunicador com 512 processos com uma mensagem relativamente pequena, quantos passos serão necessárias nesse algoritmo (conforme o exemplo abaixo) para transferir a mensagem para todos os processos?



Solid line: data transfer
Dotted line: carry-over from previous transfer

Amount of data transferred: $(N-1)*p$
N = number of processes
p = size of message

- ☐ 1
- ☐ 2
- ☐ 4
- ☐ 8
- ☒ 9
- ☐ 10
- ☐ 512



Qual dos seguintes tipos de comunicação do MPI suspende a execução do programa até que a comunicação atual seja concluída?

- ☐ Não bloqueante
- ☒ Bloqueante
- ☐ Assíncrono
- ☐ Pronto

A fim de oferecer opções ao programador de ajuste entre confiabilidade e desempenho, o MPI oferece diferentes modos de envios de mensagens, considerando ou não o uso de buffer e de protocolos confiáveis. Pesquise estes modos e responda qual dos métodos a seguir seria utilizado para enviar mensagens com a menor sobrecarga possível? *

- ☒ Modo Síncrono
- ☐ Modo Buffer (B - buffered)
- ☐ Modo Pronto (R - Ready)
- ☐ Não há diferenças entre os mencionados.

Resposta correta

- ☒ Modo Pronto (R - Ready)



Assinale as alternativas corretas. *

- ☒ A rotina MPI_Sendrecv, utiliza um buffer de sistema para enviar e outro para receber.
- ☐ Em um grupo de processos MPI de tamanho N, os processos de rank mais alto recebem prioridade de recursos mais alta.
- ☒ Uma comunicação não-bloqueante com MPI_Isend pode ser acessada imediatamente após o comando MPI_Irecv.
- ☒ Um deadlock em MPI ocorre quando vários processos são logicamente impedidos de aceitar as mensagens uns dos outros.

Resposta correta

- ☒ A rotina MPI_Sendrecv, utiliza um buffer de sistema para enviar e outro para receber.
- ☒ Um deadlock em MPI ocorre quando vários processos são logicamente impedidos de aceitar as mensagens uns dos outros.



Suponha que MPI_COMM_WORLD consiste de três processos (0, 1 e 2) e que o seguinte código é executado. Assinale a alternativa que contém os valores corretos de x, y e z em cada processo após a execução do código. *

```
MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);
int x, y, z;
switch(my_rank){
case 0: x=0; y=1; z=2;
    MPI_Bcast(&x, 1, MPI_INT, 0, MPI_COMM_WORLD);
    MPI_Send(&y, 1, MPI_INT, 2, 43, MPI_COMM_WORLD);
    MPI_Bcast(&z, 1, MPI_INT, 1, MPI_COMM_WORLD);
    printf("P0 - x = %d, y = %d, z = %d\n", x, y, z);
    break;
case 1: x=3; y=4; z=5;
    MPI_Bcast(&x, 1, MPI_INT, 0, MPI_COMM_WORLD);
    MPI_Bcast(&y, 1, MPI_INT, 1, MPI_COMM_WORLD);
    printf("P1 - x = %d, y = %d, z = %d\n", x, y, z);
    break;
case 2: x=6; y=7; z=8;
    MPI_Bcast(&z, 1, MPI_INT, 0, MPI_COMM_WORLD);
    MPI_Recv(&x, 1, MPI_INT, 0, 43, MPI_COMM_WORLD, &status);
    MPI_Bcast(&y, 1, MPI_INT, 1, MPI_COMM_WORLD);
    printf("P2 - x = %d, y = %d, z = %d\n", x, y, z);
    break;
}
```

- ☐ P0: x=0; y=1; z=4; P1: x=0; y=2; z=5; P2: x=1; y=2; z=0;
- ☒ P0: x=0; y=1; z=4; P1: x=0; y=4; z=5; P2: x=1; y=4; z=0;
- ☐ P0: x=0; y=1; z=2; P1: x=3; y=4; z=5; P2: x=6; y=7; z=8;
- ☐ P0: x=0; y=4; z=2; P1: x=0; y=4; z=5; P2: x=1; y=4; z=4;

Este formulário foi criado em Universidade Federal de Sao Paulo.

Google Formulários



