

Iniciado em	segunda-feira, 15 jul. 2024, 14:07
Estado	Finalizada
Concluída em	segunda-feira, 15 jul. 2024, 15:19
Tempo empregado	1 hora 11 minutos
Avaliar	8,1 de um máximo de 10,0(81 %)



Questão 1

Completo

Atingiu 3,6 de 4,0

Elaborar um programa MPI para calcular a média dos valores de uma matriz de inteiros e quadrada cujo tamanho n (correspondente ao número de linhas e colunas) é informado pelo usuário. Considerar as seguintes restrições:

- a) A inicialização da matriz com valores inteiros aleatórios deve ser feita por apenas um dos m processos instanciados
- b) O trabalho de cálculo da média deve ser colaborativo, de modo que cada um dos m processos instanciados seja responsável por calcular a média de uma quantidade de linhas (proporcional ao número m de processos instanciados)
- c) O programa deve eleger um processo para receber as médias calculadas para consolidação e impressão da média geral dos elementos da matriz
- d) Considerar o uso de apenas primitivas MPI de comunicação ponto a ponto (MPI_Send, MPI_Recv)

caso o arquivo corrompa está aqui o código:

```
#include <stdio.h>
#include <stdlib.h>
#include <mpi.h>
#include <time.h>

void inicia_matriz(int *matrix, int n) {
    srand(time(NULL));
    for (int i = 0; i < n * n; i++) {
        matrix[i] = rand() % 100;
    }
}

double calcula_media(int *matrix, int n, int row) {
    int sum = 0;
    for (int i = 0; i < n; i++) {
        sum += matrix[row * n + i];
    }
    return (double)sum / n;
}

int main(int argc, char *argv[]) {
    int rank, size, n;
    int *matrix = NULL;
    double total_sum = 0.0;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    if (rank == 0) {
        printf("Informe o tamanho n da matriz (n x n): ");
        fflush(stdout);
        scanf("%d", &n);
        matrix = (int*) malloc(n * n * sizeof(int));
        inicia_matriz(matrix, n);
        printf("Matriz inicializada:\n");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                printf("%d ", matrix[i * n + j]);
            }
            printf("\n");
        }
    }

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



```

if (rank != 0) {
    matrix = (int*) malloc(n * n * sizeof(int));
}

MPI_Bcast(matrix, n * n, MPI_INT, 0, MPI_COMM_WORLD);

int rows_per_process = n / size;
int start_row = rank * rows_per_process;
int end_row = (rank == size - 1) ? n : start_row + rows_per_process;
double local_sum = 0.0;

for (int i = start_row; i < end_row; i++) {
    local_sum += calcula_media(matrix, n, i);
}

if (rank == 0) {
    total_sum += local_sum;
    for (int i = 1; i < size; i++) {
        double recv_sum;
        MPI_Recv(&recv_sum, 1, MPI_DOUBLE, i, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
        total_sum += recv_sum;
    }
    double global_average = total_sum / n;
    printf("A média geral dos elementos da matriz é: %f\n", global_average);
    free(matrix);
} else {
    MPI_Send(&local_sum, 1, MPI_DOUBLE, 0, 0, MPI_COMM_WORLD);
    free(matrix);
}

MPI_Finalize();
return 0;
}

```

 [_questão_1_Danilo_Domingo_180015311.rar](#)

Comentário:

Vc não distribuiu as linhas por diretivas ponto-a-ponto (MPI_Send/MPI_Recv)



Questão 2

Completo

Atingiu 1,0 de 1,0

Observe o seguinte código MPI, cujo objetivo é conseguir gravar 100 elementos do vetor data em arquivo:

```
1 #include <mpi.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4 #define FILE_NAME "file.bin"
5 #define MAX 100
6
7 int main(int argc, char** argv) {
8     int rank, size;
9     MPI_Init(NULL, NULL);
10    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
11    MPI_Comm_size(MPI_COMM_WORLD, &size);
12    int data[MAX];
13    MPI_File fh;
14
15    int chunk = MAX / size;
16    int start = rank * chunk * sizeof(int);
17    if (rank == (size-1)) chunk+=(MAX%size);
18    for (int i = 0; i < chunk; i++)
19        data[i] = rank * chunk + i + 1;
20    MPI_File_open(MPI_COMM_WORLD, FILE_NAME, MPI_MODE_CREATE | MPI_MODE_WRONLY, MPI_INFO_NULL, &fh);
21    MPI_File_write_at(fh, start, data, chunk, MPI_INT, MPI_STATUS_IGNORE);
22    MPI_File_close(&fh);
23    MPI_Finalize();
24 } /* fim-main */
```

Considerando o propósito definido para o código, avalie as afirmativas e, a seguir, marque a opção correta:

- I - O código não funciona adequadamente porque a função da linha 21 necessita um laço para garantir que cada processo faça a escrita dos elementos sob sua responsabilidade na posição correta do arquivo
- II - Este código funciona adequadamente e a instrução da linha 17 garante que os valores sequenciais, de 1 a 100, no vetor data, independente do número de processos
- III - O código apresentado não funciona adequadamente porque a função de escrita (linha 21) exige que o vetor a ser gravado seja dividido em partes iguais entre os processos MPI

- ☐ a. Apenas a afirmativa II está correta
- ☐ b. Apenas as afirmativas I e III estão corretas
- ☒ c. Nenhuma das alternativas está correta
- ☐ d. Apenas a afirmativa III está correta
- ☐ e. Apenas a afirmativa I está correta

Sua resposta está correta.

O código funciona bem da forma como está (afirmativa I é falsa). A afirmativa II é falsa porque não há gravação de todos os valores de 1 a 100 (no entanto, o comando equilibra a gravação entre o número de processos). Afirmativa III é falsa (a linha 17 garante que eventuais sobras do vetor sejam gravadas pelo último processo com a função MPI_File_write_at)

A resposta correta é:

Nenhuma das alternativas está correta



Questão 3

Completo

Atingiu 2,5 de 4,0

Elaborar um microserviço RPC (linguagem C) que contabilize palavras em um dicionário da seguinte forma:

- se palavra_recebida = IMPRIMIR
 - listar o conteúdo do dicionário (cada palavra e o número de ocorrências)
- senão
 - se palavra_recebida existe no dicionário:
 - incrementar o contador de palavras relativo à palavra_recebida
 - senão
 - incluir a palavra_recebida no dicionário
 - contador de palavras de palavra_recebida = 1

Por sua vez, a função consumidora do microserviço (função main) deve ter os seguintes modos:

- Modo inclusão:
 - obter as palavras a serem contabilizadas a partir de um arquivo de entrada
 - enviar cada palavra identificada para o microserviço remoto
- Modo consulta:
 - enviar a string IMPRIMIR para o microserviço remoto

Na resposta, entregar arquivo compactado contendo: (i) arquivo de definição de interface, (ii) os códigos .c do cliente e do servidor da aplicação, e (iii) um README com identificação do aluno (matrícula/nome) e instruções de execução

Identificação

****Aluno:**** Danilo Domingo Vitoriano Silva

****Matricula:**** 18/0015311

****Prova 1 - Questão 3****

Arquivos

- `contador_palavras.x`: Arquivo de definição de interface RPC.
- `contador_palavras_server.c`: Código do servidor RPC.
- `contador_palavras_client.c`: Código do cliente RPC.

Compilação e Execução

Passo 1: Gerar arquivos a partir do `.x`

```
```bash
rpcgen contador_palavras.x
```
```

Passo 2: Compilar

Servidor:

```
```bash
gcc -o contador_palavras_server contador_palavras_server.c contador_palavras_svc.c contador_palavras_xdr.c -lnsl
```
```

Cliente:

```
```bash
gcc -o contador_palavras_client contador_palavras_client.c contador_palavras_clnt.c contador_palavras_xdr.c -lnsl
```
```

Passo 3: Executando Servidor

Servidor:

```
```bash
./contador_palavras_server
```



```
...
Passo 4: Executar o cliente
Modo Inclusão:
```bash  
./contador_palavras_client <hostname> inclusao <arquivo_de_entrada>  
...  
  
**Modo Consulta:**  
```bash  
./contador_palavras_client <hostname> consulta
...`
```

 [questão 3 Danilo Domingo 180015311.rar](#)

Comentário:

## Questão 4

Completo

Atingiu 1,0 de 1,0

Julgue as afirmações abaixo:

I - Sockets UDP, por serem não orientados à conexão, permitem a comunicação persistente entre processos cliente e servidor

II - Sincronicidade é uma das funcionalidades atendidas pela biblioteca MPI, uma vez que esta garante a entrega da mensagem no receptor, mesmo que o processo destinatário não esteja executando

III - Brokers como Kafka e RabbitMQ são interessantes para viabilizar comunicação persistente entre processos

- ☒ a. Apenas a afirmação III está correta
- ☐ b. Apenas as afirmações II e III estão corretas
- ☐ c. Apenas a afirmação I está correta
- ☐ d. Todas as afirmações estão corretas
- ☐ e. Apenas as afirmações I e III estão corretas

Sua resposta está correta.

A resposta correta é:

Apenas a afirmação III está correta

