



**Iniciado em** segunda-feira, 26 ago. 2024, 14:06

**Estado** Finalizada

**Concluída em** segunda-feira, 26 ago. 2024, 15:07

**Tempo empregado** 1 hora 1 minuto

**Avaliar** 7,70 de um máximo de 10,00(77%)

## Questão 1

Correto

Atingiu 1,00 de 1,00

Analise as afirmativas e, a seguir, marque a alternativa correta:

I - Apache Kafka é uma plataforma distribuída de tratamento de streaming de eventos em tempo real cujos tópicos podem ser divididos entre vários nós de um cluster

II - No Apache Kafka, os canais de acesso funcionam como uma fila de entrada/saída, no modelo FIFO - First In, First out, onde cada processo sempre recupera o último elemento do canal

III - O mecanismo publish-subscribe do Apache Kafka funciona como um protocolo de comunicação que equaliza os tempos de processamento dos vários consumidores do broker.

- ☐ a. Apenas as afirmativas II e III estão corretas
- ☐ b. Todas afirmativas estão corretas
- ☒ c. Apenas as afirmativas I e III estão corretas ✓
- ☐ d. Apenas as afirmativas I e II estão corretas
- ☐ e. Nenhuma das opções corresponde às afirmativas apresentadas

Sua resposta está correta.

I - Correto

II - Incorreto. Os processos podem acessar qualquer elemento da fita de streaming

III - Correto

A resposta correta é:

Apenas as afirmativas I e III estão corretas



## Questão 2

Correto

Atingiu 1,00 de 1,00



Analise as afirmações a seguir e marque a alternativa correta

I - O problema dos generais bizantinos é uma metáfora que descreve a dificuldade de se entrar em um acordo quando entidades centralizadas decidem em nome da maioria

II - A tecnologia blockchain é uma solução eficiente para o problema dos generais bizantinos

III - O algoritmo Paxos é uma solução de consenso distribuído cuja variante pode ser usada para coordenação e resolução de impasses em redes blockchain

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

Sua resposta está correta.

A resposta correta é:

Apenas II e III estão corretas

### Questão 3

Completo

Atingiu 0,70 de 2,00



O código MPI a seguir faz a impressão colaborativa de um vetor dinâmico, no qual (i) o MASTER faz a inicialização do vetor e (ii) cada processo faz a impressão de uma quantidade de elementos desse vetor, de acordo com o número de processos ativado. Percebe-se, no entanto, que esse programa possui um problema de alto consumo de memória, uma vez que todos os processos são obrigados a alocar o vetor completo (linha 15), embora façam a impressão de apenas um subconjunto desse vetor (para um vetor de 1 milhão de inteiros e 5 processos, serão alocados 5 milhões de inteiros, gerando alocação excessiva de memória). Promova as alterações nesse código, de modo a reduzir o consumo de memória e, ao mesmo tempo, garantir a impressão equitativa do vetor entre os processos.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <mpi.h>
4  #define MASTER 0
5  int main(int argc, char* argv[]) {
6      int rank, nprocs, *v, tamvet;
7      MPI_Init(&argc, &argv);
8      MPI_Comm_rank(MPI_COMM_WORLD, &rank);
9      MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
10     if (rank == MASTER) {
11         printf("Tamanho do vetor: ");
12         fflush(stdout); scanf("%d", &tamvet);
13     } /* fim-if */
14     MPI_Bcast(&tamvet, 1, MPI_INT, MASTER, MPI_COMM_WORLD);
15     v = (int *) malloc(tamvet * sizeof(int));
16     if (rank == MASTER) {
17         for (int i=0; i< tamvet; i++)
18             v[i]=(i+1)*10;
19     } /* fim-if */
20     MPI_Bcast(v, tamvet, MPI_INT, MASTER, MPI_COMM_WORLD);
21     int chunk = tamvet/nprocs; int ini = rank * chunk;
22     if (rank == (nprocs-1))
23         chunk=chunk+(tamvet%nprocs);
24     int fim = ini + chunk;
25     printf("%d/%d: ", rank, nprocs);
26     for (int i=ini; i<fim; i++)
27         printf("%d ", v[i]);
28     printf("\n");
29     MPI_Finalize();
30     return 0;
31 } /* fim-main */
```

 [questao3.c](#)

Comentário:

Vetor alocado duas vezes no MASTER, desnecessário.

Não funciona p todos os casos.

## Questão 4

Correto

Atingiu 1,00 de 1,00



Julgue as afirmações abaixo e marque a alternativa correta:

I - No Hadoop, o número de instâncias de funções `map()` é equivalente ao número de chunks que o HDFS promoveu no(s) arquivo(s) de entrada

II - No Spark, a função **`fold (1, lambda x, y: x+y)`** aplicada a um RDD contendo a lista [1, 2, 3, 4, 5] produzirá o resultado 24 se o número de partições do referido RDD for igual a 8

III - No Hadoop, o número de arquivos produzidos na pasta de saída é sempre igual ao número de funções `reduce()` instanciados na aplicação

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

Sua resposta está correta.

I - correto

II - correto

III - correto

A resposta correta é:

Nenhuma das opções satisfaz as afirmativas apresentadas

## Questão 5

Incorreto

Atingiu 0,00 de 1,00



Analise o programa a seguir:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <omp.h>
5  #define MAX 14
6
7  int main(int argc, char*argv[]){
8      int sum=0;
9
10     #pragma omp parallel for reduction(+:sum) schedule(runtime)
11     for (int i=1; i<=MAX; i++) {
12         printf("%4d @ %d\n", i, omp_get_thread_num());
13         sleep(i<4? i+1:1);
14         sum+=i;
15     } /* fim-for */
16     printf("Soma = %d\n", sum);
17     return 0;
18 } /*fim-main*/
```

Sobre este código, julgue as afirmações feitas a seguir:

- I - O escalonamento com **static,1** produz um desempenho pior do que escalonamento com **dynamic,1**
- II - O escalonamento com **static,2** produz um desempenho melhor do que o escalonamento com **static,1**
- III - O escalonamento com **static,3** tem desempenho pior do que o escalonamento com **dynamic,2**

Agora marque a alternativa CORRETA:

- ☐ a. Apenas a afirmativa II é verdadeira
- ☐ b. Apenas a afirmativa III é verdadeira
- ☐ c. As afirmativas I e III são verdadeiras
- ☒ d. Nenhuma das opções corresponde às afirmativas apresentadas ✖
- ☐ e. Apenas a afirmativa I é verdadeira

Sua resposta está incorreta.

A resposta correta é:

As afirmativas I e III são verdadeiras

## Questão 6

Correto

Atingiu 1,00 de 1,00



Analise as afirmativas a seguir e marque a alternativa correta.

- I - A replicação de dados em diferentes nós melhora a confiabilidade e a performance do sistema como um todo, mas gera problemas de consistência entre as réplicas.
- II - O algoritmo Paxos é uma solução proposta por Lamport a fim de garantir um acordo sobre o estado global do sistema (máquina de estados distribuída), incluindo valores de variáveis.
- III - Uma das soluções de consistência de dados entre réplicas mais eficiente é permitir operações de leitura em apenas um dos nós do cluster (denominado de master), desde que esse nó faça a atualização dessa operação sofrida para os demais nós do cluster de replicação.

- ☐ a. Apenas II e III estão corretas
- ☐ b. Apenas III está correta
- ☐ c. Apenas I está correta
- ☒ d. Apenas I e II estão corretas ✓
- ☐ e. Nenhuma das respostas

Sua resposta está correta.

A resposta correta é:

Apenas I e II estão corretas

## Questão 7

Completo

Atingiu 2,00 de 2,00



Sabe-se que **#pragma omp parallel for** faz a divisão de iterações do laço de execução entre as threads instanciadas no momento da execução. Elabore um programa que seja capaz de manipular um vetor de entrada com tamanho aleatório e que possa identificar/imprimir o número de índices do vetor de entrada que cada thread do programa ficou responsável (número de iterações por thread) com o pragma citado.

 [Questao7.c](#)

```
#include <stdio.h>
#include <omp.h>
#include <string.h>
#define MAX 10
int main(int argc, char *argv[]) {
    int vetor[omp_get_max_threads()];
    memset(&vetor, 0, 4*omp_get_max_threads());
    #pragma omp parallel
    {
        #pragma omp for
        for (int i=0;i<MAX;i++) {
            vetor[omp_get_thread_num()]++;
        } /* fim-for */
    } /* fim omp parallel */
    for (int i=0;i<omp_get_max_threads(); i++)
        printf("Thread[%d] iterou %d vezes \n", i, vetor[i]);
    return 0;
}
```

Comentário:

## Questão 8

Correto

Atingiu 1,00 de 1,00



Julgue as afirmações abaixo e marque a alternativa correta:

I - RDDs (**Resilient Distributed Datasets**) são estruturas tipadas do Spark que podem ser alteradas por comandos python ou R

II - As transformações (**Transformations**) são implementadas no Spark em modo **lazy**, ou seja, são executadas posteriormente, apenas quando é instanciado uma ação (**Action**), visando melhoria de performance

III - O Spark é mais rápido do que o Hadoop/Map-Reduce, porque os estágios de execução são implementados com uso intensivo de memória ao invés de uso de disco (memória secundária).

- ☐ a. Todas as alternativas estão corretas
- ☒ b. Apenas as alternativas II e III estão corretas ✓
- ☐ c. Apenas as alternativas I e III estão corretas
- ☐ d. Apenas a alternativa II está correta
- ☐ e. Apenas as alternativas I e II estão corretas

Sua resposta está correta.

A resposta correta é:

Apenas as alternativas II e III estão corretas