
Programação Paralela e Distribuída
Lista de Exercícios P2 – 2008/1

Sincronização de Relógio

Questão 1 – Explique por que existe a necessidade de sincronização de relógio em sistemas distribuídos.

Questão 2 – Entre dois relógios distribuídos, a diferença aceitável é de 2 minutos. Sabendo-se que ocorre um desvio de 7 minutos a cada 1000 minutos passados (ou seja, 0,007 minutos), de quanto em quanto tempo deve ocorrer uma sincronização neste caso?

Questão 3 – Descreva (brevemente) os algoritmos abaixo solicitados, explicando o seu funcionamento:

- a) Algoritmo centralizado: servidor passivo
- b) Algoritmo centralizado: servidor ativo
- c) Algoritmo centralizado de Berkeley
- d) Algoritmo global distribuído
- e) Algoritmo local distribuído

Questão 4 - Um sistema composto por 10 nós (de P1 até P10) atualiza o relógio utilizando o algoritmo de sincronização GLOBAL DISTRIBUÍDO. Sendo assim, observe os seguintes fatos:

- o sistema começou a operar em 12h21'12"
- o intervalo entre as sincronizações é de 11"
- o processo P1 vai realizar sua quarta sincronização
- as sincronizações anteriores não alteraram o relógio local de P1
- o intervalo de recebimento (janela de recebimento) é de 9"
- a tabela abaixo ilustra as mensagens dos demais processos e seus respectivos tempos:

Processo	Valor do relógio no processo que enviou	Tempo de chegada (no processo corrente – P1)
P2	12h21'59"	12h21'55"
P3	12h21'51"	12h21'59"
P4	12h22'05"	12h22'00"
P5	12h21'55"	12h22'02"
P6	12h21'49"	12h21'47"
P7	12h21'50"	12h21'51"
P8	12h21'52"	12h21'53"
P9	12h21'45"	12h21'46"
P10	12h21'50"	12h21'47"

Calcule qual será a hora com a qual deverá ser atualizado o relógio do processo corrente (P1) após o término da janela de sincronização. Mostre como chegou a sua resposta.

DICA: primeiro, encontre o valor do relógio local de P1.

Questão 5 – Faça o mesmo exercício anterior (exercício 4, utilizando a mesma tabela de tempos) para os seguintes casos:

a) Situação 1

- o sistema começou a operar em 12h21'20"
- o intervalo entre as sincronizações é de 11"
- o processo P1 vai realizar sua terceira sincronização
- as sincronizações anteriores não alteraram o relógio local de P1
- o intervalo de recebimento (janela de recebimento) é de 9"

b) Situação 2

- o sistema começou a operar em 12h21'00"
- o intervalo entre as sincronizações é de 10"
- o processo P1 vai realizar sua décima sincronização
- as sincronizações anteriores não alteraram o relógio local de P1
- o intervalo de recebimento (janela de recebimento) é de 7"

Programação Paralela

Questão 6 – Explique a importância do desenvolvimento de aplicações paralelas, e cite algumas aplicabilidades.

Questão 7 – Responda com **Verdadeiro (V)** ou **Falso (F)** as seguintes questões. Para as que julgar Falsas, justifique (**não corrija a sentença, mas sim JUSTIFIQUE o erro encontrado**).

- () Quando utiliza-se granularidade grossa em um programa paralelo, a comunicação entre os processos é mais rápida do que quando utiliza-se granularidade fina.
- () O balanceamento de carga com a utilização da granularidade fina torna-se mais difícil, pois existem diversas tarefas menores a serem distribuídas entre os processos.
- () Os modelos de máquina são normalmente divididos em: Workpool, Mestre/Escravo, Divisão e Conquista, Fases Paralelas e Pipeline.
- () Ao calcularmos o speedup, calculamos o tempo que a aplicação paralela leva para realizar suas tarefas.
- () Em uma arquitetura com memória compartilhada, os processos comunicam-se através de operações de leitura e escrita na memória, enquanto com memória distribuída eles comunicam-se através de troca de mensagens.
- () Tanto a comunicação síncrona quanto a comunicação assíncrona travam a execução até que algum processo receba sua mensagem ou lhe envie mensagens. A diferença entre estes dois tipos de comunicação se dá pela granularidade escolhida para particionar as tarefas (comunicação síncrona usa granularidade grossa e a assíncrona usa granularidade fina ou média).
- () MPI é uma linguagem de programação para criar programas paralelos.

Questão 8 – Quando utilizamos troca de mensagens para comunicação entre os processos?

Questão 9 – Explique o funcionamento e faça o diagrama explicativo (grafo) do modelo de aplicação Mestre/Escravo, e **descreva vantagens e desvantagens**.

Questão 10 – Explique o funcionamento e faça o diagrama explicativo (grafo) do modelo de aplicação Fases Paralelas, e **descreva vantagens e desvantagens**.

Questão 11 – Explique o funcionamento e faça o diagrama explicativo (grafo) do modelo de aplicação Divisão e Conquista, e **descreva vantagens e desvantagens**.

Questão 12 – Explique o funcionamento e faça o diagrama explicativo (grafo) do modelo de aplicação Pipeline, e **descreva vantagens e desvantagens**.

Questão 13 – Descreva de forma direta e simplificada as estratégias que você e seu grupo utilizaram para resolver o problema do trabalho 2 (RankSort Paralelo).

Questão 14 – Leia atentamente o que segue:

O programa abaixo resolve a multiplicação, adição e subtração dos elementos de um vetor de inteiros:

```
#include <stdio.h>
int main ()
{
    int x, soma=0, subtracao=0, mult=1, TAM = 1000;
    int vet[TAM];

    for (x=0;x<10;x++) {
        vet[x] = x + 1;
    }

    for (x=0;x<10;x++) {
        printf("vet[%d] = %d\n", x, vet[x]);
    }

    for (x=0;x<10;x++) {
        soma = soma + vet[x];
        subtracao = subtracao - vet[x];
        mult = mult * vet[x];
    }

    printf("Soma = %d\n", soma);
    printf("Subtracao = %d\n", subtracao);
    printf("Multiplicacao = %d\n", mult);
}
```

A variável “vet” trata-se do vetor sobre o qual as operações serão realizadas. Então, modifique o programa para executar em paralelo utilizando MPI. Apresente as primitivas utilizadas (comentando o código sempre que julgar necessário). Faça as seguintes versões MPI:

a) uma versão com apenas 4 processos executando. Nesta versão, cada processo faz uma função: 1 soma, 1 subtrai e 1 multiplica. O outro processo é responsável por avisar cada um dos outros 3 a sua função, e ao término imprimir os resultados.

b) uma versão com MPI com qualquer número de processos baseado em mestre/escravo. A divisão das tarefas deve ser a mais balanceada possível, e a forma de implementação é livre.

c) procure outras formas de implementar o problema através dos demais modelos (Divisão e Conquista, Pipeline e Fases Paralelas). Para cada um destes casos, se achar viável, apresente a implementação. Caso contrário, explique porque não é viável a implementação do modelo.