

Ciência de Computadores

Sistemas Distribuídos e Móveis

Lista de Exercícios

Data: 4 de Novembro de 2013

Questões sobre o capítulo 1, Tanenbaum & van Steen: Fundamentos

- 1) Explique o significado de “transparência”, no contexto de sistemas distribuídos, e dê exemplos de tipos de transparência.
- 2) Por que é difícil esconder falhas e recuperar destas falhas num sistema distribuído?
- 3) Por que, em alguns casos, não é desejável implementar um sistema que não é totalmente transparente?
- 4) O que é um sistema distribuído aberto (open distributed system) e quais são as suas vantagens?
- 5) O que é um sistema escalável?

Questões sobre o capítulo 2, Tanenbaum & van Steen: Arquiteturas

- 6) Como é que se pode resolver o problema de perda de desempenho quando um cliente e um servidor estão distantes e a latência de rede torna-se dominante?
- 7) O que é uma arquitetura cliente-servidor de três “tiers”?
- 8) Qual é a diferença entre distribuição horizontal e distribuição vertical?
- 9) Considere um conjunto de processos P_1, P_2, \dots, P_n numa arquitetura cliente-servidor “multi-tiered”. O processo P_i é um cliente do processo P_{i+1} e P_i somente retorna algum resultado ao processo P_{i-1} depois de receber um resultado de P_{i+1} . Qual é o problema com esta arquitetura em relação ao desempenho do processo P_1 ?
- 10) Em uma rede “overlay” estruturada as mensagens são roteadas de acordo com a topologia da rede “overlay”. Qual é a desvantagem desta abordagem?
- 11) Considere a rede CAN da Figura 2.8 do livro. Como é que encaminharia uma mensagem do nó com coordenadas (0.2,0.3) para o nó com coordenadas (0.9,0.6)?
- 12) Considere uma rede “overlay” estruturada em que cada nó escolhe aleatoriamente c vizinhos. Para buscar um ficheiro, um nó envia requisições a todos os vizinhos e pede a estes vizinhos que façam requisições a seus vizinhos. Quantos nós seriam alcançados?

13) Nem todos os nós numa rede peer-to-peer devem tornar-se superpeers. Quais são os requisitos que um nó superpeer deve ter?

Questões sobre o capítulo 3, Tanenbaum & van Steen: Processos e capítulo 7, Sukumar Ghosh: Exclusão Mútua Distribuída

14) Critique as seguintes soluções em software para resolver o problema de exclusão mútua em memória compartilhada.

SOLUTION 1

```
int turn;
proc(i)
int i;
{
    while (TRUE)
    {
        compute;
        while (turn <> i);
        CS;
        turn <- i+1 mod 2;
    }
}
turn <- 1;
proc(0) AND proc(1);
```

SOLUTION 2

```
boolean flag[2];
proc(i)
int i;
{
    while (TRUE)
    {
        compute;
        while (flag[i+1 mod 2]);
        flag[i] <- TRUE;
        CS;
        flag[i] <- FALSE
    }
}
flagp[0] <- flag[1] <- FALSE;
proc(0) AND proc(1);
```

SOLUTION 3

```
boolean flag[2];
proc(i)
int i;
{
    while (TRUE)
    {
        compute;
        flag[i] <- TRUE;
        while (flag[i+1 mod 2]);
        CS;
        flag[i] <- FALSE;
    }
}
flagp[0] <- flag[1] <- FALSE;
proc(0) AND proc(1);
```

15) Neste problema, você deve comparar a leitura de um ficheiro utilizando um sistema de ficheiros

“single-threaded” com um sistema de ficheiros “multi-threaded”. Assuma que as operações de requisitar trabalho, despachar e fazer o restante do processamento demora 15 milissegundos e que os dados necessários estão em cache. Se uma operação em disco for necessária (o que acontece aproximadamente em um terço das vezes) a operação demora 75 milissegundos, e a thread dorme durante este período. Neste cenário, quantas requisições por segundo o servidor consegue tratar se for “single-threaded”? E se for multi-threaded?

16) Faz sentido limitar o número de threads por processo em um servidor? Por que?

17) Para o algoritmo de exclusão mútua distribuída de Ricart e Agrawala mostre que:

- os processos entram na sua seção crítica de acordo com a ordem de seus “timestamps”
- a correção é garantida mesmo que os canais de comunicação não sejam FIFO

18) Uma versão generalizada do problema de exclusão mútua, em que no máximo L processos ($L \geq 1$) podem estar nas sua regiões críticas simultaneamente, é conhecida como o problema da exclusão- L (L -exclusion). Modifique o algoritmo de Ricart e Agrawala de forma a resolver este problema.

19) Considere o seguinte algoritmo de exclusão mútua entre dois processos em memória compartilhada:

```

programa resolve {for process i: member(i,{1,2})}
define    x: integer, y: boolean
initially y = false

do true -->
start: x := i;
    if y -->
        do y --> skip od;
        goto start;
    fi;
    y := true
    if x <> i -->
        y := false;
        do x <> 0 --> skip od;
        goto start;
    fi;
    CS
    y := false; x := 0;
    non-CS;
od

```

- Este código está correto?
- Se N processos $1,2,\dots,N$ executarem este código, qual é o número máximo de processos que executarão simultaneamente a seção crítica CS?

20) A Figura 1 mostra a rota de carros para passarem numa ponte que conecta os pontos A e B. Dois carros vermelhos (r1 e r2) e dois carros azuis (b1 e b2) movem-se ao longo da rota indicada na figura num número indefinido de viagens.



Figura 1: para a questão 20

- Usando um modelo de troca de mensagens, escreva um protocolo descentralizado que permita que no máximo um carro passe pela ponte de cada vez e que todos os carros tenham uma chance de passar pela ponte. Cada carro corresponde a um processo. Assuma que seus relógios não estão sincronizados (não há relógio global).
- Escreva um outro protocolo que permita que mais de um carro passe pela ponte ao mesmo tempo, mas na mesma direção e que todos os carros tenham uma chance de passar pela ponte.

21) Um servidor que mantenha uma conexão TCP/IP com um cliente é considerado com estado (stateful) ou sem estado (stateless)?

22) Considere um processo P que precisa aceder a um ficheiro F que está localizado na máquina onde P está executando. Se P migrar para outra máquina e o tipo de ligação do ficheiro à máquina for fixo (fixed file-to-machine binding), como é que acha que P poderia continuar a sua execução?

Questões sobre o capítulo 4, Tanenbaum & van Steen: Communication

23) Por que os serviços de comunicação da camada de transporte (OSI transport layer) são inapropriados para construir aplicações distribuídas?

24) Suponha que precisa implementar uma aplicação que utiliza comunicação síncrona transiente, mas somente dispõe de primitivas de comunicação assíncronas. Como é que implementaria a sua aplicação?

25) Tabelas de roteamento utilizadas pelo sistema IBM Websphere são construídas manualmente. Descreva um procedimento simples para a construção automática destas tabelas.

26) Quando utilizamos comunicação persistente, o processo “receiver” mantém um buffer local onde mensagens são armazenadas. Para criar este buffer, normalmente precisamos especificar um tamanho. Dar um tamanho ao buffer é preferível do que deixar o tamanho do buffer em aberto?

27) Explique porque comunicação síncrona transiente pode ter problemas de escalabilidade. Como estes problemas podem ser resolvidos?

28) Ao procurar por ficheiros num sistema peer-to-peer não estruturado pode ser útil buscar os ficheiros em nós que tenham ficheiros semelhantes ao que está a ser procurado. Explique como “gossiping” pode ajudar nesta tarefa.

Questões sobre a seção 5.2.3 do capítulo 5, Tanenbaum & van Steen: Distributed Hash Tables

29) Considere o sistema Chord mostrado na Figura 5.4 do livro e assuma que o nó 7 juntou-se à rede. Qual seria o conteúdo de sua “finger table”? Outras tabelas precisariam ser modificadas?

30) Considere uma DHT baseada no modelo Chord em que k bits de um identificador de m bits são utilizados para endereçar superpeers. Se os identificadores forem atribuídos de forma aleatória quantos superpeers espera-se ter num sistema com N nós?

31) Se inserirmos um novo nó num sistema Chord, há a necessidade de fazer a atualização instantânea de todas as “finger tables”?