SISTEMAS OPERATIVOS DISTRIBUÍDOS

1ª Chamada (96/97) - 15 de Julho de 1997

Nota prévia

- O exame tem duração de duas horas e trinta minutos.
- Não se esqueça de identificar todas as folhas com o seu nome e número.

IPC e RPC

Considere uma aplicação distribuída em que dois processos interagem através de disciplina do tipo pedido-resposta. Os dois processos executam-se em máquinas de arquitecturas diferentes e utilizam protocolos de comunicação da pilha TCP/IP. Considere também que os processos utilizam o seguinte algoritmo, para garantir a fiabilidade da comunicação:

```
// Cliente
pedido-resposta (char *pedido, char *resp) {
  Uid      id_s, id_r;
  Timeout  tim = 60; // em segundos

  id_s = CriaUid ();
  do {
    cliente_envia  (id_s, pedido);
    cliente_recebe (&id_r, resp, tim);
  } while (id_r != id_p);
}

// Servidor
extern Historico log;

serve () {
  char  *pedido = recbuffer ();
  char  *resp;
  Uid   id;

  while (1) {
    servidor_recebe (&id, pedido, FOREVER);
    resp = procura (log, id);
    if ( resp == NULL ) {
      resp = processa (pedido);
      acrescenta (&log, id, resp);
    }
    servidor_envia (id, resp);
  }
}
```

Questão 1 (1 valor) *Este esquema garante que o pedido é processado exactamente uma vez? Justifique.*

Questão 2 (1 valor) *Por quanto tempo deve ser mantido cada registo no histórico do servidor?*

Questão 3 (1.5 valores) *Altere o protocolo para, no caso mais favorável, diminuir o tamanho do histórico no servidor (sem enfraquecer a semântica).*

Questão 4 (2 valores) *A localização do cliente e do servidor está embutida no código das funções "envia" e "recebe" do cliente e do servidor. O que é necessário acrescentar para o código do cliente e do servidor ficar independente da localização?*

Serviços de nomes

Questão 5 (1 valor) *Em sistemas baseados em servidores cooperativos, diga o que entende por resolução de nomes recursiva e não-recursiva.*

Questão 6 (0.5 valor) *Considere o sistema DNS. Dado que para aumentar a eficiência, os servidores mantém uma "cache" de algumas traduções realizadas no passado, como é possível saber se uma dada resolução está a devolver o endereço actual?*

Sistemas de ficheiros distribuídos (SFD)

Questão 7 (1.5 valores) *O NFS é um SFD que utiliza "servidores sem estado". Indique o que entende por servidores sem estado neste contexto, e indique algumas vantagens e desvantagens desta solução.*

Questão 8 (1.5 valores) *Descreva de modo sumário, como é feita a gestão da memória escondida (cache) no cliente NFS.*

O sistema AFS mantém no disco dos clientes cópias integrais dos ficheiros a que estes acedem. Deste modo, o cliente pode realizar uma operação de leitura sem qualquer sobrecarga do servidor.

Questão 9 (1 valor) *Diga como é que no AFS o cliente pode saber se a sua cópia é válida sem contactar o servidor.*

Replicação

O SFD Coda permite que várias cópias do mesmo ficheiro sejam mantidas em diferentes servidores. Cada cliente, ao fazer uma alteração ao ficheiro, actualiza todos os servidores a que consegue aceder. Caso existam partições na rede, é possível que dois clientes realizem alterações concorrentes em diferentes réplicas. De modo a detectar estas situações, cada servidor mantém, para cada ficheiro, um "vector de versões".

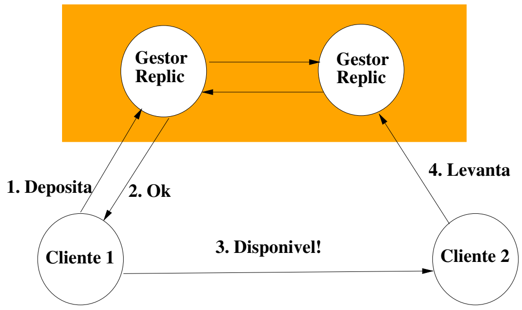
Sejam V_1, V_2 e V_3 os vectores de versões de três rélicas de uma dado ficheiro que após uma partição na rede, voltam a conseguir comunicar entre sí.

Questão 10 (1.5 valores) *Para cada um dos seguintes cenários, indique se existe uma situação de conflito e, caso não exista, que servidores possuem a cópia mais recente do ficheiro.*
a) $V_1 = [2, 2, 2]$, $V_2 = [2, 4, 4]$ *e* $V_3 = [2, 4, 4]$*;*
b) $V_1 = [1, 1, 1]$, $V_2 = [1, 5, 2]$ *e* $V_3 = [1, 2, 2]$
c) $V_1 = [6, 2, 2]$, $V_2 = [2, 4, 4]$ *e* $V_3 = [2, 4, 4]$

Questão 11 (2 valores) *Imagine que pretendia evitar conflitos impedindo que a réplica fosse alterada simultaneamente em duas partições distintas. Que restrições colocaria às actualizações? Indique que tipo de protocolo ou primitiva de comunicação utilizaria para disseminar as alterações.*

Uma técnica de gerir a replicação é deixar cada cliente comunicar com a réplica que se encontra mais próxima e garantir que as diferentes réplicas se sincronizam para oferecer a ilusão de uma única cópia aos clientes. Um método de minimizar a carga consiste em agrupar diversas actualizações que são posteriormente propagadas em grupo (periodicamente ou por necessidade). Este esquema designa-se por actualização diferida ("lazy replication" ou "gossip architecture").

Considere a sequência ilustrada na figura seguinte, em que um cliente faz um depósito. Posteriormente, avisa outro cliente que o depósito está disponível, despoletando um levantamento (cada cliente, contacta uma réplica diferente).



Questão 12 (1.5 valores) *Segundo esta aproximação, como é possível garantir que a operação de levantamento se executa após o depósito na segunda réplica (note-se que existe uma relação causa-efeito entre as diversas mensagens)?*

Questão 13 (1 valor) *Caso não existisse relação causa-efeito entre a operação de depósito e a operação de levantamento, seria necessária alguma sincronização entre ambas a réplicas?*

Transacções

Questão 14 (1 valor) *Considere um sistema transaccional cujo controlo da concorrência é baseado em ferrolhos ("locks"). Diga qual a vantagem de distinguir dois tipos de ferrolhos, nomeadamente de leitura e de escrita. Diga também porque é que geralmente se escolhe uma concretização em que os ferrolhos, depois de adquiridos, só são libertados quando a transacção termina.*

Questão 15 (2 valores) *Diga o que entende por um protocolo de confirmação atómica (atomic commit) e descreva o protocolo conhecido por "two-phase commit".*