Número: Nome:

Sistemas Distribuídos

FCT NOVA
2020/21 - Teste 2 (6 páginas)
sem consulta ; duração total: 1h30
NOTA IMPORTANTE: o tamanho das caixas é indicativo da dimensão esperada da resposta e não um desafio para descobrir quem consegue fazer a letra mais pequena.

As respostas erradas às perguntas V/F descontam até o equivalente ao valor da resposta certa correspondente. Para as perguntas de escolha de múltipla, o desconto é de $1/(n-1)$, com n o número de opções. A penalização acumula apenas no contexto da mesma pergunta. Em cada pergunta, a primeira resposta errada não desconta.

Questão 1
Para cada uma das seguintes afirmações, indique se é [V]erdadeira ou [F]alsa ou escolha a resposta apropriada, no caso das perguntas de escolha múltipla.

1. _ O objetivo dos testes funcionais é verificar se um sistema funciona de acordo com a especificação.
2. _ Uma diferença importante do teste de sistemas distribuídos face ao teste de software que executa num só processo é que é necessário considerar as falhas do sistema (e o modelo de falhas definido).
3. _ Se o evento e_1 aconteceu antes do evento e_2, então é porque necessariamente o evento e_1 foi a causa de ocorrência do evento e_2.
4. _ Num sistema distribuído, dois eventos, e_1 e e_2 estão sempre relacionados pela relação aconteceu antes, i.e., ou se tem que e_1 aconteceu antes de e_2 ou então e_2 aconteceu antes de e_1.
5. _ Um relógio vetorial pode ser visto como uma representação compacta duma história causal.
6. _ Um vetor versão pode ser visto como uma representação compacta duma história causal (em que se registam apenas alguns eventos).
7. _ Num sistema com dois serviços (e.g., o trabalho realizado nas aulas práticas em qualquer dos últimos anos), não é possível manter a relação de aconteceu antes entre eventos que ocorreram em serviços diferentes.
8. _ No algoritmo primário/secundário, apenas o primário necessita de manter um log com as operações já executadas.
9. _ O algoritmo de replicação com consistência eventual estudado nas aulas implementa a replicação de máquina de estados.
10. _ No algoritmo de replicação com consistência eventual estudado nas aulas para garantir que todas as réplicas convergem para o mesmo estado é necessário que cada réplica sincronize com todas as outras.
11. _ Usando o algoritmo de replicação com consistência eventual estudado nas aulas, se o cliente não propagar informação de causalidade entre os servidores (e.g., relógio de Lamport) quando executa duas escritas sucessivas em diferentes servidores, é possível que o estado do servidor fique com o resultado da execução da primeira operação (e não da segunda).
12. _ No sistema Coda, um cliente pode detetar que houve uma escrita concorrente com a sua, se o vetor versão no servidor for maior do que o seu (com maior definido como tendo todas as entradas do vetor maiores ou iguais, e pelo menos uma estritamente maior).

Questão 2
Para cada uma das seguintes afirmações, indique se é [V]erdadeira ou [F]alsa ou escolha a resposta apropriada, no caso das perguntas de escolha múltipla.

_ Num sistema de comunicação indireta, um emissor não indica explicitamente os recetores das mensagens que envia.

No sistema Kafka é possível paralelizar a receção de eventos dum tópico, usando múltiplos threads/processos para receber mensagens concorrentemente.

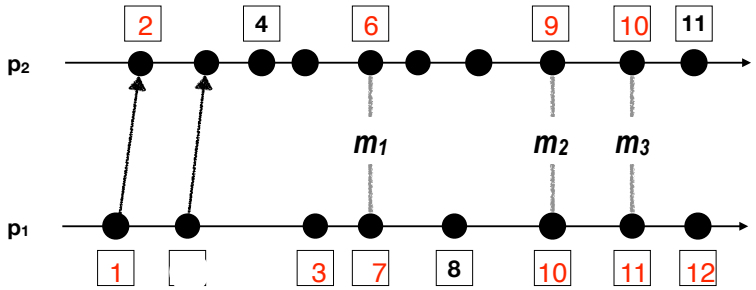
_ O sistema Kafka pode ser usado para implementar um sistema de comunicação em grupo fiável com ordenação total.

4. _ Quando temos um sistema de comunicação em grupo com apenas dois processos (os únicos a enviar e receber as mensagens), um sistema que entregue as mensagens por ordem FIFO também entrega as mensagens por ordem total.
5. _ A utilidade dum sistema de controlo de acessos num serviço distribuído (e.g. servidor de chat) que não tenha um mecanismo de autenticação é muito limitado (ou mesmo nulo).
6. _ Os ataques de negação de serviço podem ser evitados com recurso a técnicas criptográficas.

7. . _ A cifra por blocos encadeados permite evitar os ataques por análise de frequência.
8. . _ O protocolo de Needham-Schroeder com chaves secretas garante a segurança futura perfeita.
9. . _ O protocolo TLS apenas permite autenticar o servidor - o cliente tem sempre de ser autenticado usando outro mecanismo, e.g. par nome do utilizador/password.
10. . _ A resolução recursiva de nomes impõe maior carga nos servidores do que a resolução iterativa.
11. . _ O DNS apenas permite registar o endereço de máquinas, i.e., fazer o mapeamento entre o nome duma máquina e o seu endereço IP.
12. . _ A funcionalidade de nós efémeros do Zookeeper é útil na criação de sistemas replicados.

Questao 3

Considere o seguinte diagrama temporal correspondente a um sistema composto por dois processos: p_1 e p_2. São representados vários eventos, incluindo alguns relativos à comunicação entre os dois processos, a qual é ponto-a-ponto e uni-direcional. Para as mensagens m_1, m_2, m_3 a direção de envio e receção não está representada. Os valores contidos, nas caixas junto aos eventos, correspondem a estampilhas de um relógio de Lamport. Assumindo que um processo incrementa o seu relógio de Lamport pelo valor 1 quando necessita de atribuir uma nova estampilha, preencha os valores em falta e determine a direção das mensagens m_1, m_2, m_3, utilizando para cada entrada o valor máximo admissível, tendo em conta que as estampilhas devem ser valores inteiros.



Questão 4

Suponha que pretende implementar o sistema discord com base num sistema de comunicação em grupo para propagar as operações para todos os utilizadores do sistema. Por simplicidade, assuma que existe apenas um canal, já criado e apenas duas operações: envio de mensagem (e.g., cada linha de texto que o utilizador enviar para o canal) e envio de resposta. Indique **justificadamente** qual o tipo de ordem mais fraco (sem ordem/FIFO/causal/total) e fiabilidade (fiável/não fiável) que usaria na propagação de cada uma das operações.

Envio de mensagem: Multicast fiável/não fiável e Ordem: sem ordem / FIFO / causal / total,

Envio de resposta: Multicast fiável/não fiável e Ordem: sem ordem / FIFO / causal / total,

Número: Nome:

Questão 5

Considere o algoritmo primário/secundário estudado nas aulas. Neste algoritmo, num sistema com N servidores/réplicas para tolerar F falhas, o primário deve esperar pela receção da confirmação de F secundários antes de devolver a resposta ao cliente. Em caso de falha do primário, o novo primário (e.g., eleito usando o Zookeeper) deve contactar quantos secundários antes de poder começar a processar novas operações? Justifique.

N-1 / N-2 / N-F / N-F-1 / F secundários, porque... (risque o que não interessar)

Questão 6

Considere que pretende implementar um algoritmo de replicação recorrendo ao sistema Kafka para disseminar as operações entre os servidores do sistema. Para tal, decide implementar o seu sistema da seguinte forma, considerando S o estado do serviço (na sua resposta, considere que a operação op executa de forma atómica e que a alteração do estado do serviço, no thread em background, também é atómica):

```
S : estado do serviço

function op( params): result
    res = exec_op( S)        // altera o estado do serviço S na réplica local
    kafka.publish( S)        // publica o novo estado do serviço
    return res

Background thread
   forever
      newS = kafka.receive() // recebe uma evento com um novo estado
      S = newS               // substitui o estado local pelo estado recebido
```

a) A implementação indicada garante que todas as réplicas convergem para o mesmo estado caso cessem de ser executadas novas operações? Justifique.

Sim, porque... / Não, porque.... (risque o que não interessar)

b) Excluindo eventuais problemas de convergência e considerando que um cliente se mantém ligado sempre ao mesmo servidor, indique comportamentos anómalos que um cliente pode observar na evolução do estado da réplica. NOTA: considere como comportamento anómalo qualquer sequência de resultados (de operações) que não pudessem ter sido gerados caso o sistema não fosse replicado.

Número: Nome:

Questão 7

Considere um sistema distribuído concebido para editar folhas de cálculo online, composto por um serviço de utilizadores e vários servidores dedicados a alojar e manipular as folhas de cálculo em benefício dos utilizadores. (Note que este sistema difere do sistema desenvolvido no trabalho prático deste ano por ter apenas um domínio, mas nesse domínio além dum servidor de utilizadores existiram múltiplos servidores de folhas de cálculo).

Neste sistema, os utilizadores para poderem realizar operações nos servidores das folhas de cálculo, precisam de fornecer uma prova da sua identidade. Esta prova, com validade limitada, é obtida no serviço de utilizadores. Para tal, cada utilizador do sistema tem uma chave simétrica secreta partilhada com o serviço de utilizadores. Adicionalmente, existe ainda uma chave simétrica secreta K_S, partilhada entre todos os servidores do sistema (i.e, servidores de utilizadores e servidores de folhas de cálculo).

a) Supondo que a Alice é um utilizador registado no sistema que pretende aceder às suas folhas de cálculo, complete o protocolo que a Alice executa para executar de forma segura (com autenticação dos parceiros, confidencialidade e evitando ataques de *replaying*) uma operação *op* com resultado *res* no servidor S_1, partindo do princípio que quando a Alice se registou no sistema da primeira vez forneceu a sua chave simétrica secreta K_A e recebeu em troca uma chave pública K_{pubSU} associada ao serviço de utilizadores. Assuma que todos os servidores conhecem a chave pública K_{pubSU}.

(1) A $\rightarrow$ SU :
(2) SU $\rightarrow$ A :
(3) A $\rightarrow S_1$:
(4) $S_1 \rightarrow$ A :

Indique a melhor resposta, tendo em conta o objetivo do protocolo e os pressupostos enunciados (só deve escolher a opção "nenhuma das anteriores" se nenhuma das soluções anteriores solucionar o problema).

(a) B O passo 1 deve ser preenchido com:
(A) $A, \{S_1, op\}K_A$
(B) $A, S_1, Na,$
(C) $\{A, S_1, Na, op\}K_{pubSU}$
(D) $\{A, S_1, K\}K_{pubSU}$
(E) nenhuma das anteriores

(b) B O passo 2 deve ser preenchido com:
(A) $\{\{A, S_1, t, op\}K_S\}K_A$
(B) $\{Na, K, \{A, S_1, t, K\}K_S\}K_A$
(C) $\{Na, \{A, S_1, t, K\}K_S\}K$
(D) $\{Na, K, \{A, S_1, t, K\}K_S\}K_{privSU}$
(E) nenhuma das anteriores

(c) C O passo 3 deve ser preenchido com:
(A) $\{A, S_1, t, op\}K_S$
(B) $\{t, op\}K_S$
(C) $\{A, S_1, t, K\}K_S\{Nb, op\}K$
(D) $\{A, S_1, t, K\}K_S\{Nb, op\}K_{pubSU}$
(E) nenhuma das anteriores

(d) D O passo 4 deve ser preenchido com:
(A) $\{res\}K_S$
(B) $\{Nb, res\}K_S$
(C) $\{res\}K$
(D) $\{Nb, res\}K$
(E) nenhuma das anteriores

m) Através de uma fórmula ***importrange(uri, range, token)***, este sistema permite criar folhas cujas células podem conter valores obtidos de outras folhas de cálculo. Essa fórmula requer três parâmetros, um *uri* que identifica a folha de onde se importam os valores, um *range* que delimita os valores importados. O terceiro parâmetro, o *token*, tem como objetivo permitir aos servidores do sistema comprovarem que o utilizador (o dono da folha que contém a fórmula *importrange*) foi autorizado a importar os valores da outra folha.

Proponha um formato para o *token*, descrevendo-o usando a notação criptográfica habitual e explique como o mesmo poderia ser gerado.

Número: Nome:

Rascunho

NOTA: esta página não será corrigida a menos que haja um referência explícita numa pergunta para uma resposta aqui.