

Sistemas Distribuídos

Faculdade de Ciências e Tecnologia

2018/19

Teste 1 - Versão A

sem consulta - 1h 30 minutos + 15 minutos tolerância

Questão 1 (≥ 3 valores)

Considere que pretende implementar uma aplicação concorrente do Instagram, em que os produtores de conteúdo são anónimos. Assim, esta aplicação será centrada na noção de canal – um canal consiste numa lista de fotografias, identificadas por uma *hashtag*. Nesta sua aplicação, um utilizador pode submeter de forma anónima uma fotografia a associá-la a um ou mais canais. Um utilizador pode subscrever um ou mais canais existentes. Existem canais com uma abrangência global, em que existem utilizadores de todos o mundo interessados, e outros com abrangência local, em que apenas utilizadores duma dada região do mundo estão interessados.

Para tal, a sua aplicação fornece as seguintes operações: (1) submissão anónima de uma nova fotografia num ou mais canais; (2) listar as fotografias de um canal; (3) subscrição de um canal por um utilizador; (4) listar as fotografias que um utilizador tem subscritas; (5) de-subscrição de um canal por um utilizador.

As operações de subscrição e de-subscrição apenas têm efeitos para o futuro, i.e., a subscrição não adiciona à lista do utilizador fotografia já no canal e a de-subscrição não remove fotografias que estão na lista do utilizador.

- a) Considere que pretende implementar o seu sistema na Amazon AWS, tendo ao seu dispor um conjunto de três centros de dados – Europa, América do Norte e Ásia. Os utilizadores duma dada zona geográfica acedem ao centro de dados mais próximo, a menos que exista alguma falha. Discuta se faz sentido replicar as fotografias em múltiplos centros de dados, considerando os diferentes tipos de canais existentes?

- b) Suponha que implementa um serviço que mantém a informação das fotografias subscritas por um utilizador. Este serviço mantém, em cada centro de dados, a informação particionada em múltiplas máquinas. Para suportar os requisitos legais de proteção dos direitos de autor necessita de ter uma operação que permite remover uma fotografia. Pretende-se que quando uma fotografia é removida, sejam removidas todas as referências a essa fotografia nas listas de subscrição. Discuta se seria interessante usar um sistema de editor/assinante para implementar esta funcionalidade, explicando porque não ou como usaria esse sistema.

- c) Considere que tem ao seu dispor um serviço de CDN, que mantém um conjunto elevado de servidores distribuídos pelo mundo (nos diferentes ISPs). Discuta como poderia usar este serviço para melhorar o desempenho da sua aplicação.

PARTE 2 (≤ 12 valores)

As respostas erradas às perguntas V/F descontam o equivalente ao valor da resposta certa correspondente. A penalização aplica-se ao conjunto das perguntas, mas apenas se o número de respostas erradas for **superior ou igual a três**.

Questão 2

Para cada uma das seguintes afirmações, indique se é [V]erdadeira ou [F]alsa.

1. Um conjunto de componentes de *software* e *hardware* será um sistema distribuído se e só se todas as seguintes características se verificarem: (1) comuniquem pela troca de mensagens, sem recorrer a memória partilhada; (2) coordenem-se entre si com um mesmo objetivo; (3) não partilhem um relógio; (4) os componentes não falharem todos ao mesmo tempo.
2. A definição de sistema distribuído obriga (impõe) que os componentes de *software* estejam distribuídos por componentes de *hardware* separados fisicamente (distintos).
3. Os problemas da heterogeneidade no âmbito dos sistemas distribuídos resultam, exclusivamente, do recurso a componentes de *hardware* de arquiteturas distintas.
4. Um sistema de *middleware* pode ser usado no contexto de sistemas distribuídos para endereçar desafios no âmbito da transparência.
5. Os sistemas distribuídos abertos são aqueles cujos clientes acedem aos recursos de uma forma transparente da sua localização.
6. Anycast é um modo de comunicação que escolhe um receptor de entre um grupo de potenciais receptores. Mensagens *anycast* enviadas por dois processos, residentes na mesma máquina, irão ser encaminhadas para o mesmo receptor.
7. Um sistema distribuído pode ser replicado e não ser tolerante a falhas bizantinas.
8. Um sistema replicado será tão mais escalável quanto mais réplicas tiver.
9. A disponibilidade de um sistema distribuído pode ser alta mesmo que este seja pouco fiável.

Sim, se a recuperação for rápida e as falhas duraram muito pouco tempo.

10. As redes P2P não estruturadas são adequadas quando se espera que o número de participantes possa alterar-se substancialmente, frequentemente ou de forma abrupta.
11. A geo-replicação tem como objetivo principal melhorar o débito (throughput) do sistema.
12. A escalabilidade de uma DHT é o resultado de alguma estratégia de particionamento do domínio dos identificadores dos nós.
13. Uma DHT é tipicamente formada por um sistema P2P estruturado que inclui uma política de replicação (parcial) dos dados.
14. No sistema BitTorrent, no decurso da transferência dos blocos do ficheiro, os participantes formam uma rede P2P não-estruturada.

15. A descentralização completa do sistema BitTorrent passa por ter os participantes a formarem duas redes P2P, uma estruturada e outra não estruturada.
16. A rede P2P estruturada Kademia é utilizada no sistema BitTorrent para suportar as pesquisas dos nomes dos ficheiros disponíveis no sistema, em alternativa ao servidor central - o *tracker*.
Falso, porque serve para encontrar o .torrent com base no seu identificador
17. Um serviço REST, pode recorrer a um *proxy*, na mesma máquina do cliente, com o fim de mascarar ao cliente falhas de conectividade e que este trabalhe em modo desligado/*offline*.

Questão 3

No contexto da invocação remota, indique para cada um das seguintes afirmações se é [V]erdadeira ou [F]alsa.

18. Em SOAP, por omissão, a semântica de invocação é "pelo menos uma vez" (*at least once*).
19. No Java RMI é possível aceder diretamente aos atributos (variáveis) dos objetos remotos, se a operação for de leitura. As escritas têm de ser feitas explicitamente através de um método (*setX*).
20. Em .NET Remoting é possível fazer invocações remotas sem explicitar a interface do serviço através de uma IDL.
21. Em SOAP, a passagem de dados envolve a codificação dos mesmos exclusivamente no formato XML.
22. Em SOAP, os parâmetros e o resultado das invocações é passado por valor, sem exceções.
23. Tal como o Java RMI, ProtoBuf é um sistema de codificação de dados que usa um formato binário.
24. Em REST, o método PUT deve ser usado para alterar parcialmente um recurso já existente.
25. Nos web services SOAP, WSDL é um documento em formato XML que descreve integralmente a interface de um serviço. Por esse motivo, o WSDL tem a função de uma IDL.
26. No desenho de uma API REST é comum estipular-se que a tentativa de criar um recurso já existente deve falhar com 409 Conflict. Porém, se o identificador URI do recurso for gerado pelo servidor no momento da sua criação, a decisão de falhar a operação com 409 Conflict pode gerar situações em os clientes poderão ver-se incapazes de aceder a alguns dos recursos.

Questão 4

Considere o seguinte código Java (Java RMI) referente à definição de um objeto remoto (servidor) para gerir perfis de utilizadores.

```

public class Profile implements java.io.Serializable {           1
    String userId, fullName, photoUrl;                          2
    int posts, following, followers; // constructor, setter and getter methods omitted... 3
    transient String password;                                  4
}                                                                5
                                                                6
public interface Profiles extends java.rmi.Remote {             7
    Profile getProfile(String userId) throws java.rmi.RemoteException; 8
    void createProfile(Profile profile) throws java.rmi.RemoteException; 9
}                                                                10
                                                                11
public class RemoteProfilesServer extends java.rmi.server.UnicastRemoteObject implements Profiles { 12
    final Map<String, Profile> db = new ConcurrentHashMap<>(); 13
    RemoteProfilesServer() throws RemoteException { super(0); } 14
    public Profile getProfile(String userId) throws RemoteException { return db.get(userId); } 15
}                                                                16
                                                                17
public static void main(String[] args) throws Exception {      18
    java.rmi.registry.Registry registry = java.rmi.registry.LocateRegistry.getRegistry( args[0] ); 19
    Profiles a = new RemoteProfilesServer();                    20
    registry.bind("/profiles", a);                               21
    Profile p = a.getProfile("user123");                         22
    p.setPosts(p.getPosts() + 1);                               23
    Profiles b = (Profiles) registry.lookup("/profiles");      24
    Profile q = b.getProfile("user123");                        25
    q.setPosts(q.getPosts() + 1);                               26
    System.out.println(p.getPosts() == q.getPosts());          27
}                                                                28

```

27. Na invocação da linha 22, o objeto *p* (Profile), retornado como resultado da invocação, foi passado por referência.
28. Nas linhas 22 e 26, em ambos os casos, trata-se da invocação remota do método *getProfile()*.

29. Em Java RMI, um objeto Serializable, como indicado na linha 1, é codificado e enviado num formato textual, semelhante ao JSON, mas mais compacto.
30. A codificação dos objetos da classe Profile apenas inclui os valores dos campos, não sendo incluída informação sobre os tipos dos mesmos.
31. A operação *bind* na linha 21 é uma invocação remota sobre a *registry*, logo o valor de *a* é passado por referência.
32. O programa termina imprimindo *true*, como resultado da operação contida na linha 28.
33. O valor de *args[0]* só poderá ser "localhost", caso contrário o programa irá falhar.

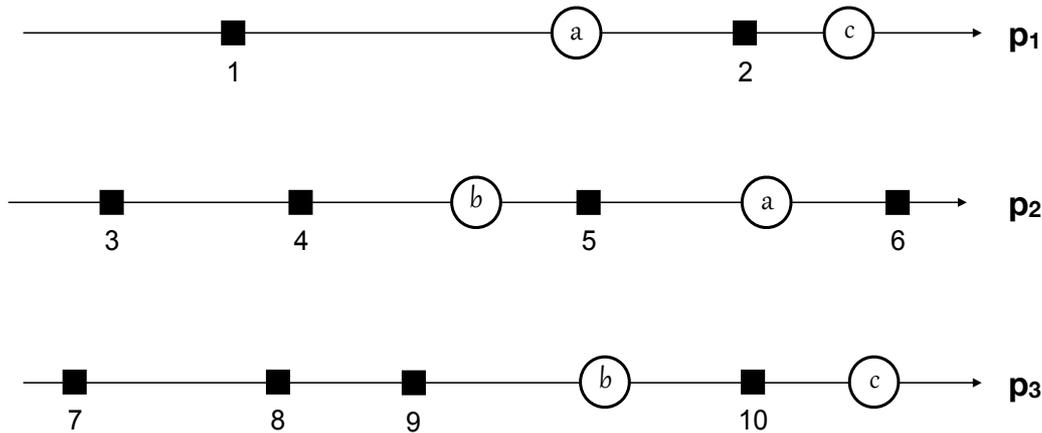
Só é possível registar objetos implementados localmente por questões de segurança.

34. Se o valor de *args[0]* não for "localhost", o programa irá falhar na linha 21 por uma questão de segurança.

Questão 5

Considere o seguinte diagrama que ilustra um padrão de **comunicação em grupo**, envolvendo 3 processos: p_1 , p_2 e p_3 . No total, são enviadas 3 mensagens: *a*, *b*, *c*, por ordem arbitrária. O diagrama está incompleto. A origem destas mensagens e o momento de envio foram omitidos, e nem todos os eventos de entrega estão representados. Os eventos de entrega que são conhecidos estão representados por um círculo com uma letra referente à mensagem em questão. Os quadrados numerados, de 1 a 10, representam eventos que tanto podem ser o envio (multi-ponto) de uma mensagem, ou a entrega de uma mensagem. A comunicação é fiável e todos os processos deverão receber cada uma das mensagens.

Nota: Não rasure o diagrama, vai precisar dele para responder à última pergunta.



Para cada uma das seguintes questões indica se a afirmação é verdadeira (V) ou falsa (F).

35. Se o envio das mensagens *a*, *b*, *c* tiver origem num mesmo processo, externo ao grupo p_1 , p_2 e p_3 , a primitiva de comunicação é garantidamente incompatível com a semântica de ordenação FIFO;
36. Para a semântica da primitiva de comunicação respeitar a ordem total, então o evento 1 corresponde, necessariamente, à entrega da mensagem *b*;
37. Se a mensagem *b* foi emitida por p_2 e o evento 1 for a entrega de *b* em p_1 , então o evento 4 não pode ser o envio de *b*.
38. Para a semântica da primitiva de comunicação respeitar a ordem total, então a mensagem *c* não pode ter sido enviada antes da mensagem *b*.

quando se fala de ordem total, a questão prende-se com a ordem de entrega ser a mesma em todos os processos, não com a ordem de envio.

39. Se p_3 for o emissor das 3 mensagens *a*, *b* e *c*, a primitiva de comunicação respeita as ordens total e FIFO.
40. Os eventos 1, 2, 5, 9 são compatíveis com uma semântica de ordenação compatível com uma ordem total causal.

Se respondeu V, complete o diagrama de modo a mostrar um padrão de comunicação compatível com uma ordem total causal, que envolva os eventos indicados.

1 - entrega de b, 9 - envio de b, 2 - envio de c, 5 envio de a, 6 - entrega de c, 10 entrega de a

Se respondeu F, complete o diagrama de modo a representar um padrão de comunicação compatível com uma ordem total causal, utilizando o maior número de eventos indicados.

Número:

Nome:

RASCUNHO