FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Departamento de Informática

Licenciatura / Mestrado Integrado em Engenharia Informática
Sistemas Distribuídos – exame, 30 de junho de 2015
1º Semestre, 2014/2015

NOTAS: Leia com atenção cada questão antes de responder. **O exame é sem consulta e tem a duração de 2h30min**. O exame contém **6** páginas.

NOME: _____ **NÚMERO**: _____

1) Considere o contexto do trabalho prático, em que um conjunto de servidores fornecem acesso a ficheiros através de RMI e Web Services (SOAP) e em que um servidor de contacto mantém informação sobre os nomes que cada servidor tem (como no trabalho prático, pode haver mais do que um servidor com um dado nome). Complete o código do anexo A, um excerto do código do sistema com o código adequado face aos comentários no código.

2) Para cada pergunta, assinale como **V[erdadeira]** ou **F[alsa]** cada uma das afirmações. **As respostas erradas descontam**.

___ A autenticação dos parceiros de comunicação é um dos aspetos importantes na segurança em sistemas distribuídos.

___ Num sistema distribuído não é possível saber se um componente remoto falhou.

___ Num sistema cliente/servidor particionado, cada servidor deve tratar, em média, (r + w) / n pedidos, com r o número de pedidos de leitura, w o número de pedidos de escrita e n o número de servidores.

___ Numa rede peer-to-peer DHT, semelhante à estudada nas aulas, é possível diminuir o número de passos no encaminhamento duma mensagem mantendo mais informação sobre os nós do sistema.

___ Um edge-server é muito útil para reduzir a carga nos servidores no acesso a dados multimédia.

___ O IP multicast implementa um modelo de comunicação assíncrono.

___ O TCP implementa um modelo de comunicação persistente.

___ No contexto do Java RMI, ao serializar um grafo de objetos em que exista mais do que uma referência para o mesmo objeto (e.g. hashtable em que duas chaves apontam para o mesmo objeto), o estado desse objeto só é serializado uma vez.

___ No .NET remoting é possível configurar diretamente o endereço de um servidor remoto no código do cliente.

___ Um serviço que apenas tenha operações idempotentes não necessita de guardar estado adicional nos servidores para implementar uma semântica "exatamente uma vez".

___ Os Web Sockets são úteis para a implementação de sistemas produtor-consumidor na Web (publish-subscribe).

___ O protocolo SOAP define, entre outros, a possibilidade de invocações unidirecionais (oneway).

___ Um método Java RMI que não devolve resultados (void) é equivalente a uma invocação unidirecional.

___ O sistema de verificação da validade da cache do NFS obriga a que se envie um pedido por cada bloco em cache, mesmo quando se faz cache de múltiplos blocos do mesmo ficheiro.

___ No protoloco primário-secundário, se o cliente mantiver a versão da última versão lida/escrita e a transmitir ao fazer um pedido de leitura, o secundário pode detetar que lhe faltam operações.

___ No sistema Coda, quando o vetor versão que o cliente tem relativo a um ficheiro é diferente do vetor versão do servidor isso indica que o ficheiro foi alterado por outro cliente.

___ Um sistema de comunicação em grupo com *uniform agreement* deve ser fiável.

___ Um sistema de comunicação em grupo pode respeitar a ordem causal sem respeitar a ordem FIFO.

3) Considere que pretende implementar um sistema para disponibilizar atualizações de software, semelhante ao existentes nos sistemas Windows, Mac OS. Nestes sistemas, periodicamente, são disponibilizadas atualizações que ao serem descarregadas e executadas numa máquina permitem atualizar a versão do sistemas de operação (ou outro componente). As atualizações são tipicamente instaladas na esmagadora maioria das máquinas num período de 1 semana após a sua disponibilização, embora num reduzido número de computadores sejam instaladas até 1 ano após a sua disponibilização.

   a) Considere que tem à sua disposição três centros de dados, um em cada um dos seguintes continentes: Europa, Ásia, América. Proponha uma arquitetura para implementar este serviço. Justifique a sua opção, indicando como distribuiria as atualizações pelos três centros de dados em função de T, a antiguidade da atualização.

b) Suponha que, devido aos custo, é possível ter uma solução que englobe ou (1) três centros de dados (DC), um em cada um dos seguintes continentes: Europa, Ásia, América; ou (2) um centro de dados e servidores presentes nos ISP de todos os países. Que solução escolheria? Responda indicando as vantagens e desvantagens da solução escolhida.

Três DCs / Um DC + servidores nos ISP , porque...

4) No Java RMI é possível registar no RMI registry a executar na máquina X um servidor a executar na máquina Y? Discuta as vantagens e desvantagens da solução adoptada.

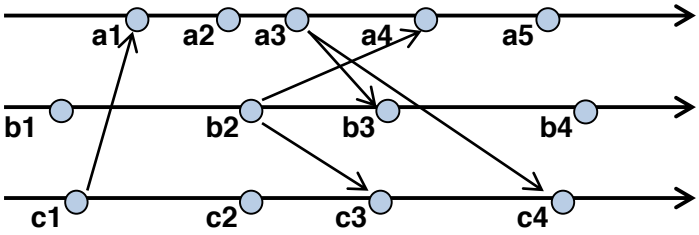
Sim / Não

Vantagens:

Desvantagens:

5) Considere um sistema distribuído com três processos, em que ocorrem os eventos assinalados a1, a2, ... As setas indicam o envio de uma mensagem.

a) Neste contexto, assinale como **V[erdadeira]** ou **F[alsa]** cada uma das afirmações. **As respostas erradas descontam**.



___ a1 aconteceu antes de b4.

___ c1 aconteceu antes de b4.

___ a1 aconteceu antes de c3.

b) Suponha que pretende identificar os eventos com relógios lógicos de Lamport. Indique o valor para cada um dos seguintes eventos, sabendo que o primeiro evento de cada processo será identificado com o relógio 1 e que cada relógio será incrementado sempre pelo menor valor possível.

a2: _____ a4: _____ b3: _____ c3:_____c4: _____

c) Se para dois eventos e1 e e2, possivelmente executados em réplicas diferentes, tivermos o valor do seu relógio lógico, é possível verificar se e1 aconteceu antes de e2? Explique como ou porque não.

Sim, porque... / Não, porque...

6) Considere que pretende criar um serviço a ser usado por aplicações de jogos para manter uma lista ordenada com a pontuação dos jogadores nesse jogo. Assuma que pode existir mais do que um utilizador a jogar usando a mesma conta - todos os resultados destes utilizadores ficam registados sob o mesmo nome, sendo guardado apenas o mais elevado. O serviço disponibiliza três operações:
   void **novaPontuação**(id_jogo, id_utilizador, pontuação),
   array<id_utilizador,pontuação> **topN**(id_jogo, startPos, endPos),
   int **posição**(id_jogo, id_utilizador)
   que permitem, respetivamente, indicar uma nova pontuação, obter as entradas na lista entre duas posições da lista e obter a posição de um utilizador na lista de pontuações.

   a) Discuta como poderia implementar este serviço usando comunicação em grupo para replicar a informação em mais do que um servidor. Nota: indique justificadamente que tipo de multicast usaria para cada uma das operações.

novaPontuação:

topN:

posição:

   b) Assuma que em cada jogo se pretende mostrar sempre o top-10 da pontuação. Discuta se faria sentido utilizar uma solução produtor/consumidor (publish/subscribe) para obter a informação necessária.

c) Indique como poderia implementar cada operação usando REST (indique a operação, e o URL).

novaPontuação:

topN:

posição:

7) Suponha que pretende usar um sistema de ficheiros distribuído para gerir um conjunto de ficheiros com as seguintes características:

- Para cada máquina, existem vários ficheiros que são acedidos pelos programas que correm na máquina para adicionar pequenas sequências de bytes ao fim do ficheiro -- os ficheiros são diferentes para diferentes máquinas;
- Existem programas que correm periodicamente - e.g. uma vez a cada hora - e executam as seguintes operações: leem o conteúdo completo dos ficheiros anteriores e escrevem outros ficheiros.

Qual o sistema de gestão de cache que considera preferível para gerir estes ficheiros - NFS ou CIFS (baseado em oportunistic locks) ?

6) Considere a seguinte alteração ao protocolo primário/secundário para processar uma operação de escrita: (1) cliente envia operação de escrita para o primário; (2) primário envia resposta ao cliente; (3) primário envia operação para os secundários; (4) ao receber uma operação do primário, caso detecte que não recebeu alguma operação anterior, envia pedido da operação em falta ao primário; caso contrário, executa a operação e envia resposta diretamente ao cliente; (5) cliente considera a operação concluída com sucesso quando recebe n / 2 + 1 respostas, com n o número de secundários.
Discuta se esta modificação permite manter a correção do protocolo. Em caso afirmativo, discuta as vantagens/desvantagens face à solução original; em caso negativo, indique em que situação não se mantém a correção do sistema.

ANNEX A

```
/** Exceção indicando que ficheiro/diretório não existe. */
public class WriteException extends Exception { ... }

/** Interface do servidor de contacto RMI. */
public interface IContactServer extends                    {
/* Devolve lista com URLs dos servidores com o nome name ou null caso não exista nenhum servidor. */
   public String[] listServers( String name) throws
}

/** Interface do servidor de ficheiros RMI. */
public interface IFileServer extends ...                    {
/* Escreve o conteúdo do ficheiro path. Em caso de erro lança WriteException. */
   public void putFile ( String path, byte[] arr) throws WriteException, ... ;
}

/** Classe do servidor de ficheiros RMI. */
public class FileServer extends                    implements                    { ...
   public void main(String[] args) {
      try { ...
         FileServer server = new FileServer();
```

```
      } catch(Exception th) { ... }
} }

/** Classe do servidor de ficheiros web service SOAP. */
```

```
public class FileServerWS {
```

```
   public void putFile ( String path, byte[] arr) throws                    {...}
   ...
   }

/** Classe principal do cliente */
public class FileClient {
   private IContactServer cServer;    /* referência para o servidor de contacto devidamente iniciada */

   /** Devolve referência para servidor RMI. Em caso de erro devolve null. */
   public IFileServer getRMIRef ( String url) { ... }

   /** Devolve referência para servidor web services - considere que existe uma classe "service" com o
   nome FileServerWSService e com um construtor com apenas um parâmetro que é o URL. */
   public ws.FileServerWS getWSRef ( String url) {
```

```
   }
```

/** Escreve conteúdo do ficheiro em todos os servidores possíveis que replicam um dado ficheiro.
* Devolve a fração dos servidores em que a escrita ocorreu com sucesso, devendo tentar escrever
* o conteúdo em cada servidor 3 vezes. */
protected float putFile (String path, byte[] arr) {

} }