ENGENHARIA DE TELECOMUNICAÇÕES E INFORMÁTICA
UNIVERSIDADE DO MINHO

Sistemas Distribuídos

Exame

15 de julho de 2023 Duração: 2h00m

I

1 Compare as estratégias *1-thread-por-ligação* e *1-thread-por-pedido* do ponto de vista da eficiência.

2 Identifique a dificuldade principal na serialização de estruturas de dados ligadas e resuma a técnica genérica normalmente usada para a ultrapassar.

3 Explique como é que o protocolo 2PC (*two phase commit*) resolve uma falha de um participante durante a segunda fase.

4 Considere um serviço de gestão de uma fila de espera semelhante ao descrito no grupo II mas assegurado por, em vez de um único, um grupo de servidores. Identifique aquele que lhe parece ser o problema de sistemas distribuídos mais importante neste caso e proponha, justificando, uma solução típica.

II

Considere um sistema cliente/servidor para controlar o acesso a um sistema de computação paralela com N *cores*, que deve limitar as tarefas em execução simultânea. Os clientes usam uma aplicação para solicitar o acesso e para indicar que a tarefa no computador paralelo está concluída.

1 Apresente uma classe Java (para ser usada no servidor) que implemente a interface abaixo, tendo em conta que os seus métodos serão invocados num ambiente *multi-threaded*.

```
interface HPC {
  int inicio(int ncores);
  void fim(int tarefa, long tempo);
  Map<Integer, Long> historia();
}
```

O método `inicio` deve bloquear até haver pelo menos `ncores` livres, podendo depois o cliente executar a sua tarefa. O método `fim` avisa que a tarefa está terminada e qual a sua duração, ficando o computador paralelo livre de novo. O método `historia` devolve, para cada tarefa que foi executada, a sua duração.

Valorização: Considere a possibilidade de dar prioridade às tarefas maiores. Minimize os *threads* que são acordados.

2 Considere o serviço ao qual clientes e funcionários se ligam por TCP, que gere o acesso ao computador paralelo. Ao chegar, um cliente envia o número de *cores* que necessita, devendo o servidor responder com `RESERVADO` quando o computador estiver disponível, indicando um identificador da tarefa. Quando terminada a tarefa, o cliente envia `FIM`, indicando a tarefa e o tempo gasto. Implemente só o programa servidor usando *threads*, *sockets* TCP, e a classe desenvolvida na pergunta anterior. Use um protocolo o mais simples possível, por exemplo, baseado em linhas de texto.

Valorização: Considere que o cliente poderá ainda obter, para cada pedido já executado, qual o tempo que demorou, enviando `HISTORIA` para o servidor.