

MC714 – Sistemas Distribuídos
Teste 2 (18/3/2013)

Nome:	RA:
--------------	------------

A duração do teste é de 15 minutos. Não é permitida a consulta a qualquer tipo de material.

1. Se um cliente e um servidor estão localizados longe um do outro, a latência da rede pode ser um fator predominante no desempenho geral. Cite e comente a eficácia de algumas técnicas que podem minimizar esse problema.

Caches no cliente ou em pontos intermediários da rede têm boa efetividade se os clientes tendem a solicitar as mesmas requisições, e as mudanças nos dados não são muito frequentes (ou imprecisões não importam). Pré-fetching automático é efetivo em casos em que é possível prever com boa acuidade o comportamento do cliente. Tornar o cliente multi-threaded e separar os serviços em mais de um servidor permite que o cliente execute fetching concorrente, melhorando o desempenho. Clientes multi-threaded também são eficientes com replicação de servidores, se a maioria das requisições é do tipo consulta (caso contrário a manutenção da consistência das réplicas afeta negativamente o desempenho).

2. O que é uma arquitetura cliente-servidor em três camadas (ou níveis — *tiers* em inglês)?

É uma arquitetura de software comum em aplicações com foco no uso de bancos de dados por usuários, em que o tratamento de uma requisição/serviço é realizada por três camadas bem definidas: interface com o usuário (contém tudo relativo à interface com o usuário, incluindo gerenciamento de tela), processamento (contém a lógica da aplicação) e dados (contém a parte relativa a manutenção e gerência dos dados utilizados pela aplicação — geralmente um banco de dados). A camada de interface com o usuário é geralmente implementada no cliente, a de processamento pode estar implementada parte no cliente (mas é raro) e parte no servidor (usualmente toda), e a de dados é implementada no servidor.

3. Considere uma rede overlay não estruturada com N nós, em que cada nó escolhe aleatoriamente c vizinhos. Como é feita a busca por um arquivo, a partir de um nó qualquer?

Para buscar um arquivo um nó inicia um algoritmo de inundação: nó envia uma mensagem de busca(id , arq) para todos os seus c nós vizinhos, onde id indica a instância de busca (por exemplo, cliente e hora da requisição) e arq indica o arquivo desejado. Ao receber uma mensagem de busca, nó inicialmente verifica se já tratou dessa busca específica; se já tratou (recebeu anteriormente), desconsidera. Se não tratou, verifica se tem o arquivo localmente; se tem, responde para o cliente. Se não tem, repassa a mensagem de busca para todos os seus vizinhos, que seguem o mesmo algoritmo.