

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE CIÊNCIA DA COMPUTAÇÃO
INE 5425 – MODELAGEM E SIMULAÇÃO

**PROJETO, DESENVOLVIMENTO E PRODUÇÃO DE UM SIMULADOR EM
LINGUAGEM DE PROPÓSITO GERAL**

Introdução

Em um sistema cliente/servidor (SCS), um computador cliente envia pedidos de serviços específicos, enquanto um computador servidor atende aos pedidos de clientes, os processa e envia a resposta de volta para o cliente. O programa aplicativo associado e suas tarefas computacionais são divididos entre cliente e servidor, de modo a acelerar os tempos de resposta, com a redução da sobrecarga de comunicação ou com o alcance de certo grau de paralelismo. À máquina cliente, geralmente menos poderosa, pode ser atribuída parte do pré-processamento de consulta e o pós-processamento de tarefas. Já a máquina do servidor em geral, mais poderosa, executa as tarefas fundamentais de computação. Um cliente pode solicitar os serviços de vários servidores em seqüência ou em paralelo. Exemplos comuns de servidores incluem: servidores de arquivos e servidores de banco de dados. Como exemplos de sistemas cliente/servidor incluem: sistemas de processamento de transações (TP) como os sistemas bancários on-line, sistemas de compras on-line na Web, etc.

Uma característica de um SCS é sua arquitetura em camadas. O número de camadas (*tiers*) depende de como os clientes e servidores e interagem uns com os outros. Por exemplo, em uma arquitetura de duas camadas, o lado cliente utiliza uma interface de usuário que permite a comunicação direta com o servidor. Mesmo que possa haver mais de um servidor, os clientes podem manter ligações diretas com servidores remotos através de uma rede de comunicação subjacente. Arquiteturas de duas camadas parecem ser mais adequadas para sistemas com até 100 clientes. A arquitetura em três camadas (Figura 1) surgiu para superar as limitações da arquitetura em duas de camadas.

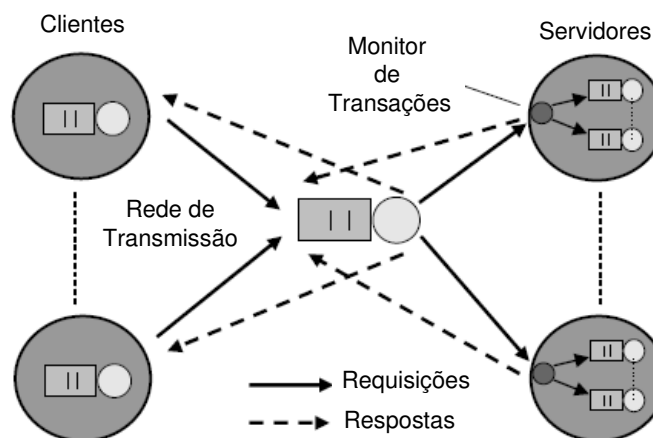


Figura 1: Representação esquemática de um SCS em três camadas

Nesta arquitetura, uma camada intermediária é adicionada entre o cliente e o servidor a qual, normalmente, envolve o processamento de transações ou monitoramento da funcionalidade. Esta camada intermediária pode, por exemplo, programar e controlar filas de mensagens, agendamento de tarefas, priorização dos processos da aplicação, ou alguma combinação

destes. Portanto, a camada intermediária realiza uma intermediação (interfaciamento) do lado cliente para o servidor, embora toda essa intermediação seja transparente para os clientes. O software de implementação de arquiteturas em três camadas é comumente conhecido como *middleware*. A Figura 1 mostra um SCS de três camadas na qual os servidores processam as transações que se encontram nas filas enquanto a camada intermediária monitora e realiza o roteamento das transações. O monitor de TP recebe as transações, coloca-as nas filas para o serviço, gerencia o seu caminho até a conclusão e, por fim, envia a resposta de volta para o cliente.

O problema a ser resolvido pelo simulador

Desenvolva um programa numa linguagem de propósito geral, que permita a modelagem e a simulação do sistema descrito a seguir. Considere um SCS com uma aplicação na área de recursos humanos (RH), configurado em duas camadas e composto por quatro nós cliente e um nó servidor. Os fluxos de tráfego e de processos são mostrados na representação esquemática da Figura 2.

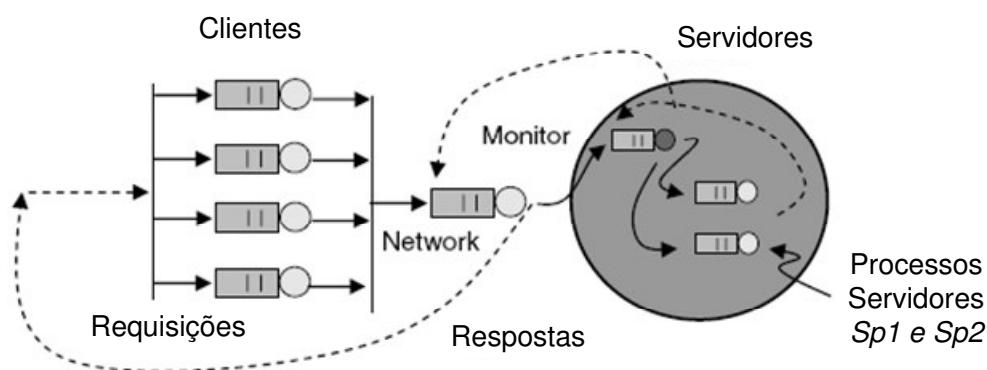


Figura 2: Representação esquemática dos processos e fluxos de tráfego no SCS.

Esta aplicação suporta solicitações de diversos tipos de serviços (pedidos) relacionados aos funcionários da empresa. Usa um servidor de banco de dados de RH (RHDB) contendo informações relacionadas aos empregados. Os tipos de operações que são realizadas no RHDB e seus atributos podem ser vistos na Tabela 1. As requisições de serviços são as seguintes:

1. Requisição do tipo *Add* adiciona um novo empregado com todas as suas informações (nome, endereço, telefone, perícia, etc) ao RHDB em um tamanho da mensagem de 1024 bytes. O sistema retorna uma mensagem de confirmação do tamanho de UNIF DISC (256, 512) bytes.
2. Requisição do tipo *Delete* exclui uma entrada do funcionário (com todas as informações relacionadas) da RHDB. A mensagem de requisição possui 1024 bytes. O sistema retorna uma mensagem de confirmação do tamanho de UNIF DISC (512, 1024) bytes.
3. Uma transação do tipo *Find* busca as informações completas do empregado no RHDB, com base em dados parciais (por exemplo, o nome). A mensagem de requisição da transação possui UNIF DISC (512, 1024) bytes. O sistema retorna uma resposta numa mensagem de tamanho de TRIA DISC (256, 512, 1024) bytes.
4. Uma transação do tipo *Search* no RHDB para todos os funcionários com características comuns (por exemplo, a mesma competência, mesmo departamento, etc.). A requisição da transação possui UNIF DISC (256, 512) bytes. O sistema retorna uma resposta de tamanho aleatório; sua distribuição é dada na Tabela 1.

Tabela 1: Transações e atributos da aplicação de RH no SCS

Requisição Tipo	Requisição Nome	Serviço Requisitado	Tamanho Requisição (bytes)	Tamanho Resposta (bytes)
1	Add	Adicionar empregado	1024	UNIF DISC (256, 512)
2	Delete	Excluir empregado	1024	UNIF DISC (512, 1024)
3	Find	Encontrar empregado	256	TRIA DISC (256, 512, 1024)
4	Search	Pesquisar por empregados	UNIF DISC (256, 512)	Disc (0.2, 64, 0.50, 512, 0.80, 1024, 1.00, 2048)

Para caracterizar os padrões de tráfego no sistema, é preciso especificar as chegadas no servidor, dos processos de todos os tipos enviados por cada nó do cliente. A Tabela 2 fornece os perfis de solicitações de serviço.

Tabela 2: Perfil das requisições solicitadas à aplicação de RH no SCS

Cliente	TEC Requisições	
	Mensagens/mseg.	Distribuição por tipo de requisição
1	Expo ($1/\lambda_1$)	Disc (0.2, 1, 0.35, 2, 0.75, 3, 1.00, 4)
2	Expo ($1/\lambda_2$)	Disc (0.4, 1, 1.00, 2)
3	Expo ($1/\lambda_3$)	Disc (0.6, 3, 1.00, 4)
4	Expo ($1/\lambda_4$)	Disc (0.6, 3, 1.00, 4)

Inicialmente, considere os seguintes valores: $\lambda_1 = 30$, $\lambda_2 = 60$, $\lambda_3 = 120$ e $\lambda_4 = 120$. Quando o programa for utilizado o usuário poderá alterar os valores de λ permitindo-se assim testar o comportamento do sistema modelado diante diferentes demandas de serviços.

Note que a segunda coluna especifica o pedido de fluxos multiplexados de cada nó cliente como processos de Poisson, com os tempos entre chegadas associados a distribuições exponenciais. O tipo de qualquer pedido é especificado por uma distribuição discreta (terceira coluna). O nó do servidor tem um monitor TP que recebe os pedidos e os encaminha para o processo no servidor para a prestação do serviço solicitado. Existem dois processos de servidor, chamados SP1 e SP2.

O primeiro fornece serviços de tipos de *Add* e *Delete*, e o último provê os serviços do tipo *Find* e *Search*. Os tempos associados a cada tipo de serviço para cada tipo de solicitação (incluindo todos os tempos de CPU e de gerenciamento do banco de dados) são apresentados na Tabela 3.

Tabela 3: Tempos de processo por tipo de requisição solicitada à aplicação de RH no SCS

Requisição Tipo	Requisição Nome	Tempo de Processamento por tipo de requisição (mseg)
1	Add	10
2	Delete	8
3	Find	15
4	Search	30

Os tempos apresentados na terceira coluna da tabela 3 são apenas para fins de referencia inicial para a construção do programa. Da mesma forma que para o parâmetro λ , quando o programa estiver em produção e for utilizado pelo usuário, você deve permitir que este possa

alterar os tempos de processamento do BD e assim testar o comportamento do sistema modelado diante diferentes cargas de processamento.

Após a conclusão do serviço, as respostas aos pedidos são enviadas para o monitor de TP que, por sua vez, as envia de volta para seus respectivos clientes através da rede de transmissão. O tempo de serviço para o monitor TP é 1 milissegundo por visita, enquanto capacidade da rede de transmissão é de 200 bytes/mseg. Note que a eficiência da rede é de 70% (+/- 10%).

Estatísticas

Para medir o desempenho deste sistema, algumas variáveis de respostas devem ser acompanhadas. Para este programa, as seguintes variáveis abaixo enumeradas devem ser tratadas:

- a) *Número de Entidades nas Filas, mínimo, máximo e médio*: Ao longo do período simulado, o número de entidades presentes nas filas (uma variável de estado aleatória) dos recursos, se altera, podendo assumir diversos valores discretos. Os valores mínimos e máximos são respectivamente o menor e maior valor observado ao longo da simulação. Para se obter uma estatística do valor esperado (médio) destas variáveis, é necessário um acompanhamento (ao longo do período simulado) dos diversos valores assumidos e dos períodos de tempo ao longo dos quais estes permaneceram constantes. Em outras palavras, estas são variáveis dependentes do tempo. Sua obtenção requer o cálculo de uma média ponderada, cujos pesos serão parcelas (percentuais) do tempo total de observação (tempo simulado) nos quais a variável n.º de elementos na fila, permaneceu em determinado estado.
- b) *Taxa Média de Ocupação dos recursos*: Assim como o Número Médio de Entidades na Fila, esta também é uma estatística dependente do tempo. Porém, como se sabe antecipadamente os possíveis estados dos servidores (neste caso apenas dois: livre ou ocupado), a média ponderada necessária é mais facilmente calculada.
- c) *Tempo de uma Entidade na Fila, mínimo, máximo e médio*: Cada uma das entidades que aderem a uma fila dos recursos despende ali um determinado período de tempo. Este período é também uma variável aleatória, uma vez que é dependente dos tempos de serviços das entidades que estão sendo servidas. Os valores mínimos e máximos são respectivamente o menor e maior valor observado ao longo da simulação. O cálculo da média requer apenas que se calcule uma média aritmética simples, considerando os tempos de todas as entidades que por ali circularam.
- d) *Tempo de Resposta, mínimo, máximo e médio*: Estas estatísticas deverão ser coletadas de forma semelhante à anterior. O tempo de resposta de uma entidade é contado entre o momento do envio de uma requisição ao BD e o momento em que o cliente recebe a correspondente resposta.
- e) *Contador de Entidades*: Este é um elemento típico de qualquer programa de simulação. Trata-se apenas de um simples acumulador. Neste caso, como o nome pressupõe, deverá incrementar uma variável designada, sempre que for ativado. Se dois contadores forem colocados no sistema, um na entrada (cliente emitindo requisição) e outro na saída (cliente recebendo resposta), a diferença entre os dois indicará o número de entidades que se encontram entre os dois pontos naquele momento.

O programa construído deverá permitir que as estatísticas acima descritas sejam coletadas para que se possa fazer uma análise do desempenho deste sistema sob diferentes condições de funcionamento.

Itens que devem ser contemplados no seu programa:

- a. Funcione de acordo com o enunciado (lógica);
- b. Os tempos das atividades possam ser determinísticos ou aleatórios;
- c. Permita que o usuário possa (de alguma forma) acompanhar a evolução da simulação (use variáveis na tela, gráficos, etc.);
- d. Permita que o usuário possa (de alguma forma) acompanhar que o avanço do tempo com passo variável, de acordo com calendário de eventos;
- e. Permita a parada/continuação da simulação e a observação das estatísticas até aquele momento;
- f. Emita um relatório final contendo todas as estatísticas desejadas;

Forma de apresentação (entrega) do programa

1. Via Internet (mail/anexado) ou uma informação de um link para baixá-lo, contendo o programa executável e todos os arquivos necessários à sua execução. Considere que no computador, onde o teste será realizado, exista apenas o SO disponível. Considere que o usuário não conheça nenhuma linguagem de programação.
2. Um relatório contendo toda a documentação do programa (variáveis, parâmetros, rotinas, classes, objetos, fluxogramas das rotinas, etc.);
3. Um pequeno manual sobre como definir ou modificar parâmetros e executar o programa.