MC714

Sistemas Distribuídos

2° semestre, 2014

Chamada de Procedimento Remoto

RPC – operação básica

- Ao cliente: chamada comum, não send e receive.
- Resumo:

1. Procedimento cliente chama apêndice cliente.
2. Apêndice cliente constrói uma mensagem e chama SO local.
3. SO cliente envia mensagem ao SO remoto.
4. SO remoto dá a mensagem ao apêndice servidor.
5. Apêndice servidor descompacta parâmetros e chama o servidor.

RPC – operação básica

6. Servidor faz serviço e retorna resultado para apêndice.
7. Apêndice de servidor empacota resultado em uma mensagem e chama SO local.
8. SO do servidor envia mensagem ao SO cliente.
9. SO cliente dá a mensagem ao apêndice de cliente
10. Apêndice desempacota resultado e retorna ao cliente.

- Efeito líquido: nem cliente nem servidor ficam cientes das etapas intermediárias ou da existência da rede.

Comunicação orientada a mensagem

Comunicação orientada a mensagem

- RPC contribui para transparência de acesso.
- Entretanto, nem sempre é adequado
  - Se receptor não está executando
  - Sincronismo de procedimento pode precisar ser substituído

- Outro mecanismo: troca de mensagens
  - Partes executando ou não (enfileiramento de mensagens)

Comunicação transiente orientada a mensagem - Sockets

- 1. Através de portas da camada de transporte
- Socket: terminal de comunicação
  - Aplicação pode escrever dados para a rede e ler dados da rede
- Primitivas para interface TCP
  - Servidores:
    - Socket: cria terminal de comunicação para o protocolo de transporte
    - Bind: associa endereço local com socket criado (IP+porta)
    - Listen: comunicação orientada a conexão; chamada não bloqueante que permite reservar buffers para um número de conexões.
    - Accept: bloqueia chamador até receber requisição; SO cria novo socket e retorna ao chamador; permite bifurcar processo.

Comunicação transiente orientada a mensagem - MPI

- 2. Interface de troca de mensagens (MPI)
- Nível de abstração diferente de sockets
- Manipulação de diferentes formas de buffer e sincronização
- Redes e multicomputadores de alto desempenho: bibliotecas de comunicação proprietárias
  - Primitivas de alto nível eficientes, mas incompatíveis com outras bibliotecas
  - Problemas de interoperabilidade
- Definição de padrão para troca de mensagens: MPI – Message passing interface

Comunicação transiente orientada a mensagem - MPI

- Premissa: comunicação ocorre dentro de um grupo conhecido de processos.
- Cada grupo recebe identificador
- Cada processo recebe identificador (local no grupo)
- Par (groupID, processID) identifica fonte ou destinatário.
  - Usado no lugar do endereço de nível de transporte

Comunicação transiente orientada a mensagem - MPI

- Primitivas MPI:
  - MPI_bsend: assíncrona (copia para buffer local MPI e retorna).
  - MPI_send: pode bloquear até que receptor tenha iniciado operação de recebimento (sistema de execução MPI).
  - MPI_ssend: comunicação síncrona – bloqueia até que receptor receba mensagem.
  - MPI_sendrcv: bloqueia até receber resposta do receptor. Corresponde a uma RPC.
  - MPI_send e MPI_ssend: possuem variantes que evitam cópia de mensagens para buffers de sistema.
  - MPI_isend: remetente passa ponteiro para mensagem e continua.
  - MPI oferece primitivas para evitar sobrescrever buffer (verificar se terminou ou bloquear).

Comunicação transiente orientada a mensagem - MPI

- Primitivas MPI:
  - MPI_issend: remetente também passa somente ponteiro para sistema de execução MPI. Sistema indica que processou a mensagem e remetente continua.
  - MPI_recv: receber mensagem; bloqueia até chegar uma mensagem.
  - MPI_irecv: variante assíncrona (não bloqueante);

- Algumas vezes primitivas diferentes podem ser trocadas sem afetar correção de programa.
  - Variantes oferecem possibilidade de otimizar desempenho

Comunicação persistente orientada a mensagem

- Sistema de enfileiramento de mensagem ou middleware orientado a mensagem (MOM)
- Suporte para comunicação assíncrona persistente
- Capacidade de armazenamento de médio prazo para mensagens
  - Não exigem que remetente ou receptor estejam ativos.
- Suporte a transferências de mensagens que podem durara minutos ao invés de *ms*.

Comunicação persistente orientada a mensagem

- Idéia básica: aplicações se comunicam inserindo mensagens em filas específicas.
- Remetente → servidores → destinatário (mesmo se offline quando remetente enviou)

- Cada aplicação tem sua fila onde outras aplicações enviam mensagens
  - É possível aplicações compartilharem uma fila

Comunicação persistente orientada a mensagem

- Em geral, remetente sabe apenas que mensagem foi inserida na fila: entrega depende do receptor.
- Permite comunicação fracamente acoplada
- 4 combinações
- Fig. 74

- Mensagens podem conter qualquer dado
- Importante para middleware é que sejam adequadamente endereçadas.

Comunicação persistente orientada a mensagem

- Endereçamento: nome exclusivo no âmbito do sistema da fila destinatária.
- Tamanho de mensagem pode ser limitado ou pode ser fragmentada pelo sistema subjacente.
- Interface pode ser simples:
  - Put: anexe mensagem na fila especificada
  - Get: bloqueie até que a fila especificada esteja não vazia e retire a primeira mensagem
  - Poll: verifique uma fila especificada em busca de mensagens e retire a primeira. Nunca bloqueie
  - Notify: instale um manipulador a ser chamado quando uma mensagem for colocada em uma fila específica

Comunicação persistente orientada a mensagem

- Arquitetura geral de sistema de enfileiramento de mensagens
- Fig. 75
- Sistema de enfileiramento fornece:
  - Filas de fonte
  - Filas de destino
  - Providencia transferência entre filas
- Deve manter mapeamento de filas para localizações de rede: banco de dados de nomes de filas (análogo ao DNS).

Comunicação persistente orientada a mensagem

- Filas: gerenciadores de fila
- Interage com aplicação...
- ... ou como repassadores/roteadores.
  - Similar a roteamento em redes.
  - Fig. 76
  - Repassadores podem ser usados para multicasting.

Comunicação persistente orientada a mensagem - Brokers

- Aplicações diversas em sistemas distribuídos: formatos de mensagem variados.
- Em sistemas de enfileiramento, conversões são manipuladas por nós chamados brokers de mensagens.
- Converter mensagens que chegam para que sejam entendidas pela aplicação destino.
- Fig. 77
- Para sistema de enfileiramento, broker é uma aplicação

Comunicação persistente orientada a mensagem - Brokers

- Broker: de reformatador de mensagens a gateway de nível de aplicação, p.ex. conversor entre aplicações diferentes de bancos de dados.
  - Nem sempre pode-se realizar conversão
- Comum broker para EAI (Enterprise Application Integration) – integração de aplicações empresariais.
- Converte mensagens e combina aplicações com base nas mensagens que são trocadas.
  - Publish/subscribe

Comunicação orientada a fluxo

Comunicação orientada a fluxo

- Antes: troca de unidades de informação completas/ independentes.
  - Tempo não importa (pode ficar lento, mas continua correto)

- Comunicação orientada a fluxo:
  - Temporização tem papel crucial
  - Ex.: áudio: amostras de fluxo devem tocar em ordem e a intervalos pré-definidos pela frequência de amostragem.

Comunicação orientada a fluxo

- Mídia contínua versus mídia discreta
  - Contínua: relações temporais fundamentais para significado dos dados (ex. áudio, movimento)
  - Discreta: relações temporais não são fundamentais (texto, imagens estáticas)
- Fluxos de dados:
  - Aplicados a mídia discreta e contínua
  - Ex.: Unix pipe, conexões TCP/IP.
  - Reprodução de arquivo de áudio normalmente requer estabelecimento de um fluxo contínuo de dados.

Comunicação orientada a fluxo

- Modo de transmissão assíncrono
  - Sem restrição de temporização
  - Fluxos discretos
- Modo de transmissão síncrono
  - Atraso fim-a-fim máximo para cada unidade do fluxo.
  - Ex. sensores
- Modo de transmissão isócrono
  - Atraso fim-a-fim com valor máximo e mínimo

Comunicação orientada a fluxo

- Fluxo simples: uma única sequência de dados
- Fluxo complexo: vários fluxos simples relacionados (subfluxos). Ex.: áudio em mais de 1 canal, filme + áudio + legendas em canais separados.

Comunicação orientada a fluxo - QoS

- Requisitos de temporização: expressados como requisitos de qualidade de serviço (Quality of Service – QoS).
  - Descrevem o que SD deve fornecer à aplicação.

- Ex. de propriedades importantes:
  - Taxa de transmissão requerida
  - Atraso máximo para estabelecer sessão
  - Atraso máximo fim-a-fim
  - Variância máxima de atraso (jitter)
  - Atraso máximo de ida-e-volta.

Comunicação orientada a fluxo - QoS

- Pilha de protocolos da Internet: melhor esforço - não oferece garantias de QoS.

- Há mecanismos para ocultar falta de QoS
  - Serviços diferenciados

Comunicação orientada a fluxo - QoS

- Serviços diferenciados
  - Repasse acelerado: pacote deve ser repassado com absoluta prioridade
  - Repasse garantido: 4 subclasses + 3 modos de descartar pacotes
- Buffers
  - Permitem repasse a taxas regulares
    Fig. 78

Sincronização de fluxos

- Fluxo discreto + fluxo contínuo
  - Ex.: slides com áudio pela web.

- Dois fluxos contínuos
  - Ex.:1. vídeo + áudio; 2. audio estéreo – ideal < 20 microsegundos
  - Video NTSC 29,97Hz + áudio em 44Khz: unidade de áudio para sincronismo = 1470 amostras.

Sincronização de fluxos - Mecanismos

- 1. Processo executa operações de leitura/escrita em vários fluxos simples.
  - Ex. Processo alterna leitura de uma imagem e de um bloco de amostras de áudio.
  - Aplicação responsável pela sincronização.
- 2. Interface de middleware para aplicação controlar fluxos e dispositivos.
  - Ex.: Facilidade para registrar manipulador definido pelo usuário, chamado quando k novas imagens chegarem.
  - Middleware multimídia.

Comunicação multicast

Multicast

- Multicast: envio de dados a vários receptores
- Inicialmente nível de rede de transporte
  - Estabelecer caminhos
  - Esforço de gerência

- P2P → multicast em nível de aplicação.
  - Roteadores não estão envolvidos na associação dos grupos
  - Roteamento pode não ser o melhor em comparação ao que poderia ser conseguido em nível inferior

Multicast - aplicação

- Rede overlay em árvore ou malha
- Exemplo: árvore multicast sobre Chord (Scribe é sobre Pastry)
- 1. Nó quer iniciar sessão multicast
  - Gera identificador multicast, chave de 160 bits aleatória: *mid*
  - Consulta succ(mid) na rede P2P (responsável pela chave). Succ(mid) vira raiz da árvore multicast.
- 2. Nó *P* quer se juntar à árvore
  - *P* faz lookup(mid)
  - Mensagem com requisição para entrar no grupo é roteada de *P* até succ(mid).

Multicast - aplicação

- Requisição de associação passa por vários nós.
- Se um nó *Q* vê pela primeira vez requisição a *mid*, torna-se repassador; P torna-se filho de Q, que repassa requisição à raiz.
- Se nó seguinte, R, ainda não é repassador: Q vira filho de R.
- Se Q (ou R) já for repassador para *mid:*
  - Registra nó anterior como seu filho (P ou Q, respectiv.)
  - Não precisa repassar requisição para raiz: já é repassador
- Multicasting: nó envia mensagem em direção à raiz

Multicast - aplicação

- Construir árvore é relativamente fácil
  - Árvore eficiente é diferente
- Fig. 79
- Qualidade da árvore:
  - Estresse de enlace: quantas vezes pacote cruza mesmo enlace (físico)
  - Alongamento (penalidade de atraso relativo): razão entre atraso de dois nós na sobreposição em relação ao atraso na rede subjacente. Ex. B → C em fig. 79.
  - Custo da árvore: custo agregado dos enlaces → minimização: spanning tree mínima

Multicast - aplicação

- Exemplo: nó conhecido que monitora outros nós
- Nó emite requisição de associação a esse nó conhecido e obtém lista (potencialmente parcial) de membros.
- Seleciona “melhor” membro para ser seu pai na árvore.
  - Diversas formas de selecionar.
    - Limitar número de vizinhos (grau de cada nó) para não formar estrela na raiz
    - Árvores de troca

Protocolos epidêmicos

- Protocolos epidêmicos
  - Propagar informações rapidamente entre grande conjunto de nós usando informações locais
- Tentam “infectar” nós com informações novas o mais rápido possível
- Tipos de nó:
  - Infectado: tem dados que está disposto a espalhar.
  - Suscetível: nó que ainda não viu tais dados novos de outros nós.
  - Removido: nó que já está atualizado; não disposto ou capacitado a propagar informação

Protocolos epidêmicos

- Modelo de propagação popular: antientropia
- Nó P escolhe nó Q aleatoriamente e troca atualizações
- Três abordagens:
  - 1. P só envia suas atualizações a Q
  - 2. P só recebe novas atualizações de Q
  - 3. P e Q enviam atualizações um ao outro.

Gossiping

- Variação: propagação de boato (gossiping).
- 1. Nó recebe item de dado *x*
- 2. Contata nó arbitrário Q e tenta enviar *x*
- 3. Se Q já foi atualizado por outro nó, P pode perder interesse em propagar *x* com certa probabilidade

- Se conta uma fofoca a alguém, e esse alguém já sabe, perde interesse em contar a outros.

Gossiping

- Não garante que todos são atualizados.
- Combinar antientropia com gossiping.
- Levar em conta topologia de rede pode ajudar.
- Remover é mais difícil: manter registro de remoção para não ser atualizado novamente com o que foi removido
  - Propagar certificados de óbito.
  - Adicionar marca de tempo; remover certificado depois de tempo máximo de propagação.
- Ex. uso: propagar informações sobre a rede.