

Fundamentos de Processamento Paralelo e Distribuído

modelos de comunicação

Fernando Luís Dotti

Categorização

- **Envolvimento de processos:**
Ponto-a-ponto; Um para todos (difusão ou *broadcast*); Um para "vários" (*multicast*)
- **Confiabilidade:** Best Effort; Confiável Regular; Confiável uniforme; Epidêmico
- **Ordem:** sem ordem; FIFO; Causal; Total;
- **Modelo de interação:** send/receive; RPC; subscrição push (empurra) ou pull (puxa)

Categorização

- Envolvimento de Processos
 - Ponto-a-ponto
 - Difusão ou *broadcast*
 - *Multicast*

Confiabilidade Categorização

- Ponto-a-ponto
 - Fair Loss Links
 - Links perfeitos
- *broadcast*
 - Epidêmico
 - Best Effort,
 - Confiável Regular,
 - Confiável uniforme
- *multicast*
 - variantes análogas à difusão

Ordenação Categorização

- Ponto-a-ponto
 - sem ordem
 - FIFO
- *broadcast*
 - FIFO,
 - Causal,
 - Total
- *multicast*
 - Mesmas variantes da difusão
 - Ordem acíclica

Ordenação Categorização

- send/receive
 - plataformas de comunicação como: sockets (qualquer sistema operacional); MPI (Message Passing Interface); etc.
- broadcast/deliver
- RPC – Remote Procedure Call
 - Remote Method Invocation (RMI)
 - Plataformas de serviços – micro-serviços
- subscrição push (empurra) ou pull (puxa)
 - processos interessados em dados inscrevem-se em fluxos/fontes
 - fontes empurram e interessados recebem ou
 - interessados buscam (puxam) de tempos em tempos
- etc

O que utilizar ?

- O modelo de comunicação utilizado deve levar em consideração as necessidades da aplicação distribuída em termos de:
 - número (escala) de processos envolvidos;
 - nível de acuracidade necessária;
 - latência tolerada;
 - vazão desejada;
 - arquitetura da aplicação.

Exemplos

- Transmissão de mídia em tempo real
 - canais ponto a ponto ou ponto - multi-ponto (dependendo)
 - perdas toleradas
retransmissão de pacote perdido pode não fazer sentido
 - desordem tolerada
tipicamente a própria aplicação tem selo temporal aos pacotes
 - send-receive (ao invés de RPC por exemplo)

Exemplos

- Jogos
 - canais ponto a ponto ou ponto - multi-ponto (dependendo)
 - perdas podem ser toleradas – se no nível de aplicação existe tolerância suficiente para que o jogo fique consistente para os vários jogadores distribuídos
 - perdas podem não ser toleradas
 - minimização de atraso
 - send-receive ou RPC, dependendo do tipo de arquitetura do jogo

Exemplos

- Bases replicadas distribuídas
(como Google, Amazon, DropBox, Azure, outros)
 - difusão
 - confiável, uniforme
 - ordem total
 - minimização de atraso e maximização de vazão

Exemplos

- Bases replicadas distribuídas
(como Google, Amazon, DropBox, Azure, outros)
 - difusão confiável, uniforme,
com ordem total - chamada de difusão atômica
 - minimização de atraso e maximização de vazão
- Para aumento de escala – particionamento
 - multicast atômico
- Modelo de Replicação Máquina de Estados
 - State Machine Replication

.

Como se especifica e como é construída toda esta funcionalidade ?