



Universidade Tecnológica Federal do Paraná
Departamento Acadêmico de Computação
Bacharelado em Ciência da Computação

Sistemas Distribuídos

Comunicação Indireta

Prof. Rodrigo Campiolo

14/09/20

Tópicos

- Introdução
- Comunicação em grupo
- Sistemas *publish-subscribe*
- Filas de mensagem (*message queues*)
- Abordagens de memória compartilhada
- Resumo

Introdução

- Comunicação Indireta
 - Comunicação entre entidades em um SD por meio de um intermediário sem ligação direta entre o emissor e receptor.
- Duas propriedades
 - Desacoplamento de espaço
 - Desacoplamento de tempo
- Situações de uso
 - Mudanças são esperadas
 - Notificação de eventos

Introdução

	<i>Acoplamento temporal</i>	<i>Desacoplamento temporal</i>
<i>Acoplamento espacial</i>	<p><i>Propriedades:</i> comunicação direcionada para determinado destinatário (ou destinatários); o destinatário (ou destinatários) deve existir nesse momento no tempo.</p> <p><i>Exemplos:</i> passagem de mensagens, invocação remota</p>	<p><i>Propriedades:</i> comunicação direcionada para determinado destinatário (ou destinatários); o remetente (ou remetentes) e o destinatário (ou destinatários) podem ter tempos de vida independentes.</p>
<i>Desacoplamento espacial</i>	<p><i>Propriedades:</i> o remetente não precisa conhecer a identidade do destinatário (ou destinatários); o destinatário (ou destinatários) deve existir nesse momento no tempo.</p> <p><i>Exemplos:</i> multicast IP</p>	<p><i>Propriedades:</i> o remetente não precisa conhecer a identidade do destinatário (ou destinatários); o remetente (ou remetentes) e o destinatário (ou destinatários) podem ter tempos de vida independentes.</p> <p><i>Exemplos:</i> a maioria dos paradigmas de comunicação indireta abordados neste capítulo.</p>

Figura: Acoplamento versus desacoplamento de espaço e tempo.

Fonte: Coulouris

Comunicação em Grupo

- Comunicação em grupo
 - Mensagem é enviada para um grupo e a mensagem é entregue para todos os membros.
- Abstração sobre comunicação *multicast*.
- Situações de uso
 - Disseminação confiável
 - Auxílio para aplicações colaborativas
 - Auxílio para estratégias de tolerância a falhas
 - Auxílio para gerenciamento e monitoramento de sistemas.

Comunicação em Grupo

- Grupos de processos
 - Entidades que se comunicam são processos.
 - Orientada a fluxos de bytes simples.
- Grupos de objetos
 - Entidades que se comunicam são objetos.
 - Nível de abstração mais alto, uso de empacotamento de tipos de dados complexos.
- Exemplo: JGroup - <http://jgroups.org/>

Comunicação em Grupo

- Grupos
 - Abertos x grupos fechados
 - Sobrepostos x não sobrepostos
 - Sistemas síncronos x assíncronos

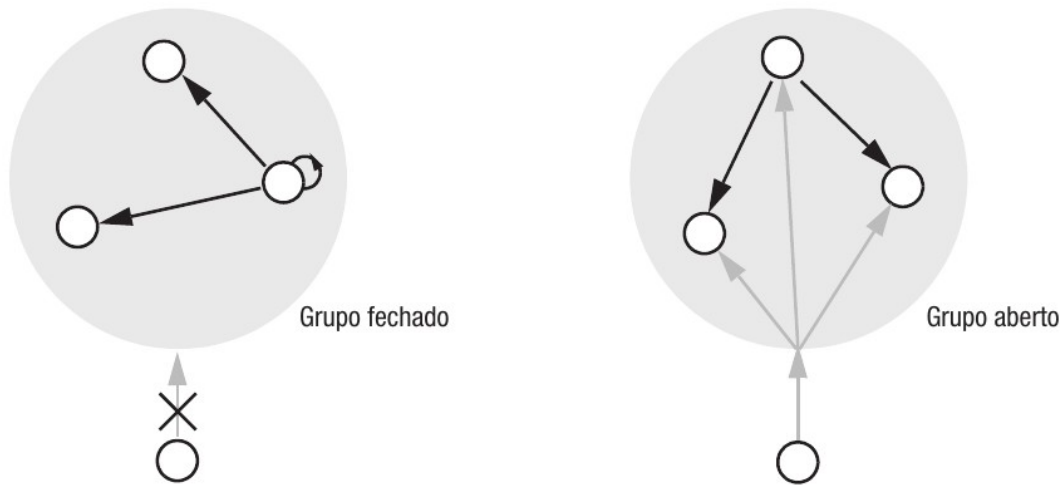


Figura: Grupos abertos e grupos fechados.

Fonte: Coulouris

Comunicação em Grupo

- Tipos de comunicação *multicast*
 - **multicast básico**: mensagem entregue se o processo não falhar, ordem arbitrária de recebimento.
 - **multicast confiável**: todos recebem a mensagem se um processo receber (integridade, validade e acordo).
 - **multicast ordenado**: obedece a ordem de entrega.
 - ordem FIFO (ordem de origem)
 - ordem causal (relação causal entre as mensagens)
 - ordem total (*atomic* – ordem de destino)

Sistemas *Publish-Subscribe*

- Sistemas Publicar/Assinar ou Sistemas distribuídos baseados em eventos.
- Conceitos:
 - **Publishers**: publicam eventos em um serviço de evento.
 - **Subscribers**: especificam interesses em eventos por meio de assinaturas (identificadores, padrões, ...).
 - **Event notifications**: as notificações de evento são enviadas aos assinantes.

Sistemas *Publish-Subscribe*

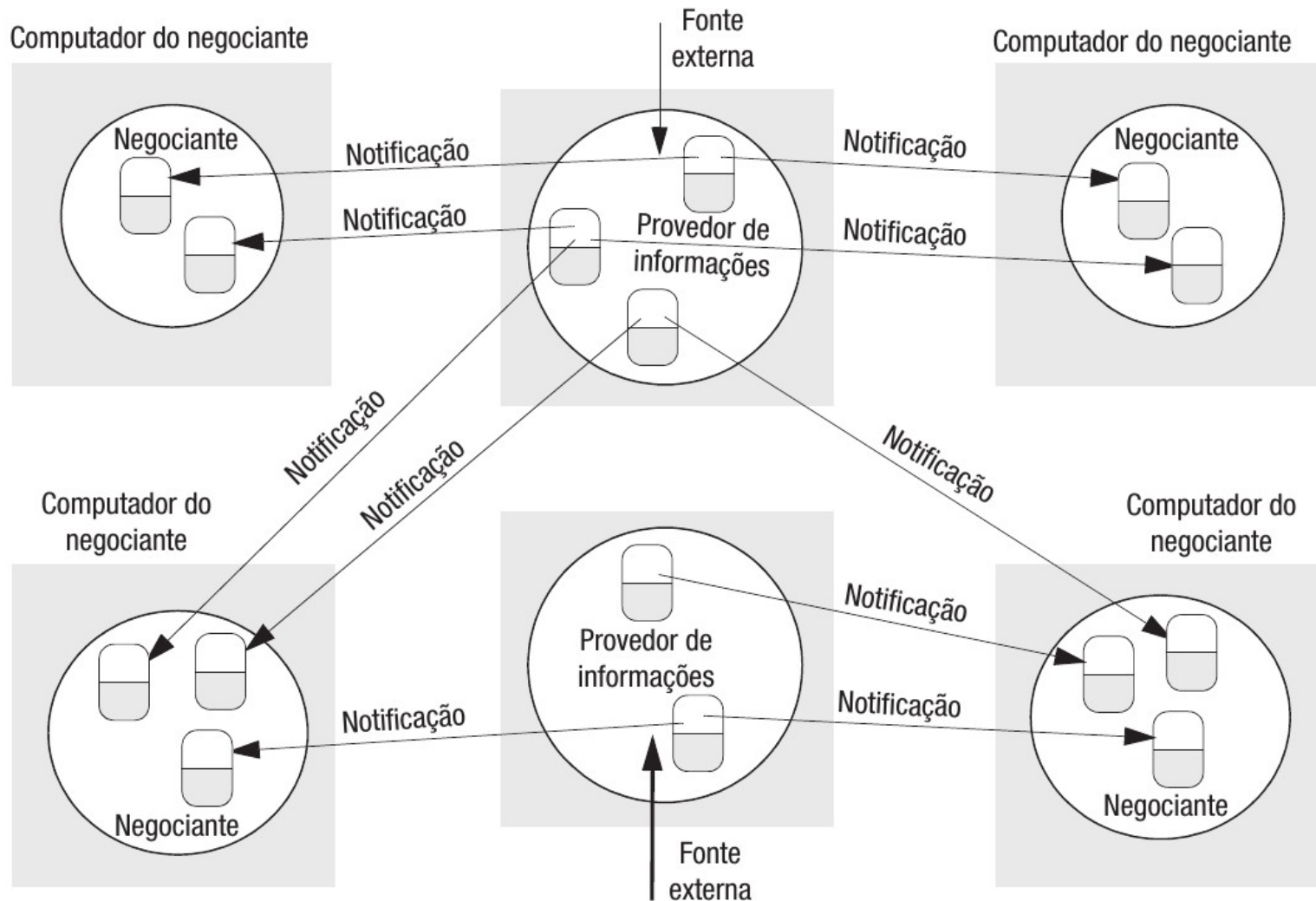


Figura: Exemplo de sistemas de notificação de eventos.

Fonte: Coulouris

Sistemas *Publish-Subscribe*

- Características
 - Heterogeneidade
 - Assincronismo
- Os eventos devem ser descritos de forma estruturada.

Sistemas *Publish-Subscribe*

- Modelo de programação

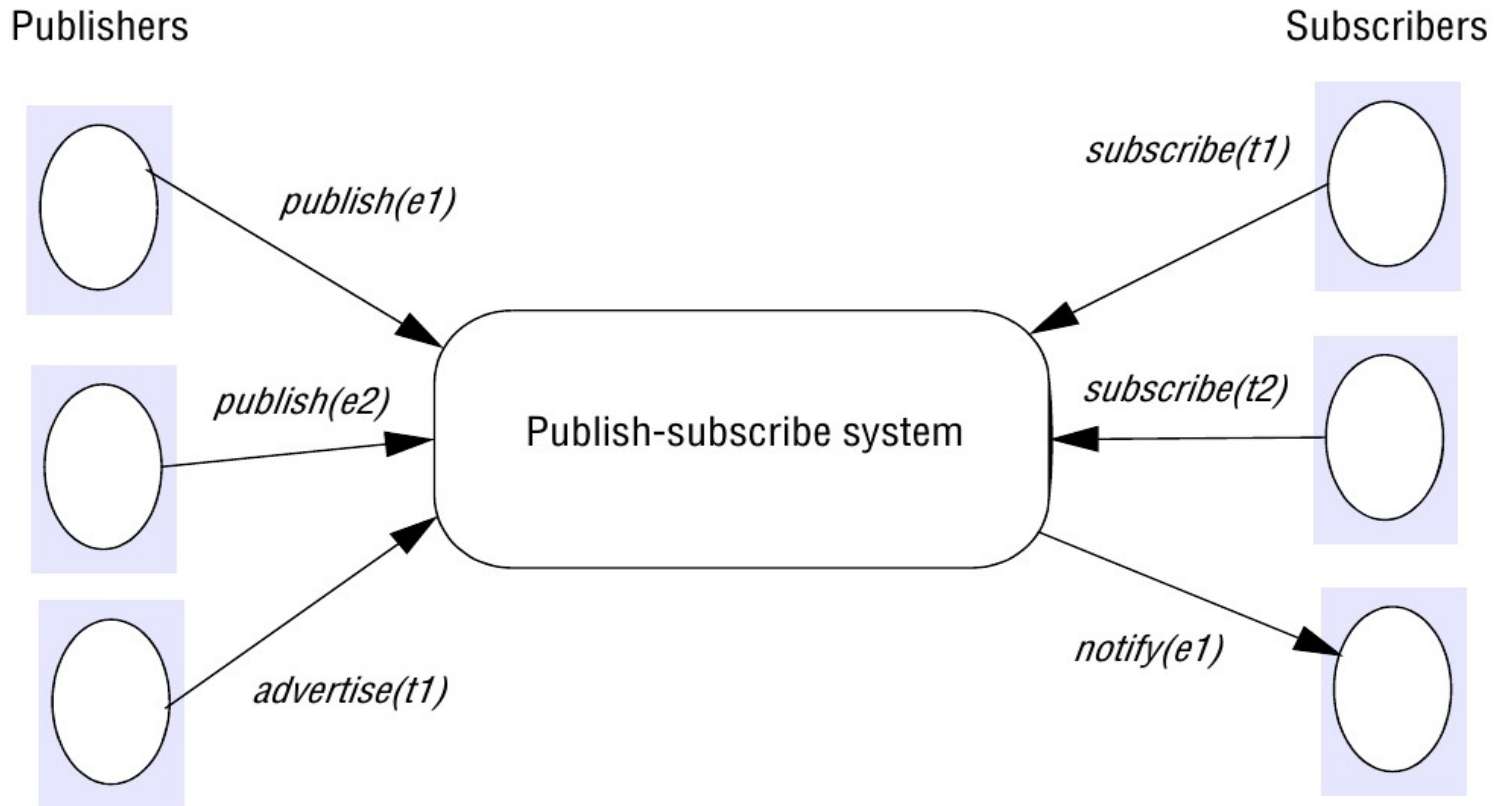


Figura: Operações para sistemas publish-subscribe.

Fonte: Coulouris

Sistemas *Publish-Subscribe*

- Modelos de assinaturas (Filtros)
 - **Channel-based**: publicadores publicam em um canal e os assinantes do canal recebem todos os eventos.
 - **Topic-based**: publicadores publicam um campo tópico e os assinantes do tópico recebem os eventos associados ao tópico.
 - **Content-based**: publicadores publicam campos e os assinantes recebem eventos segundo critérios considerando os campos.
 - **Type-based**: publicadores publicam objetos de um tipo e assinantes assinam tipos de objeto de interesse.

Sistemas *Publish-Subscribe*

- Arquiteturas

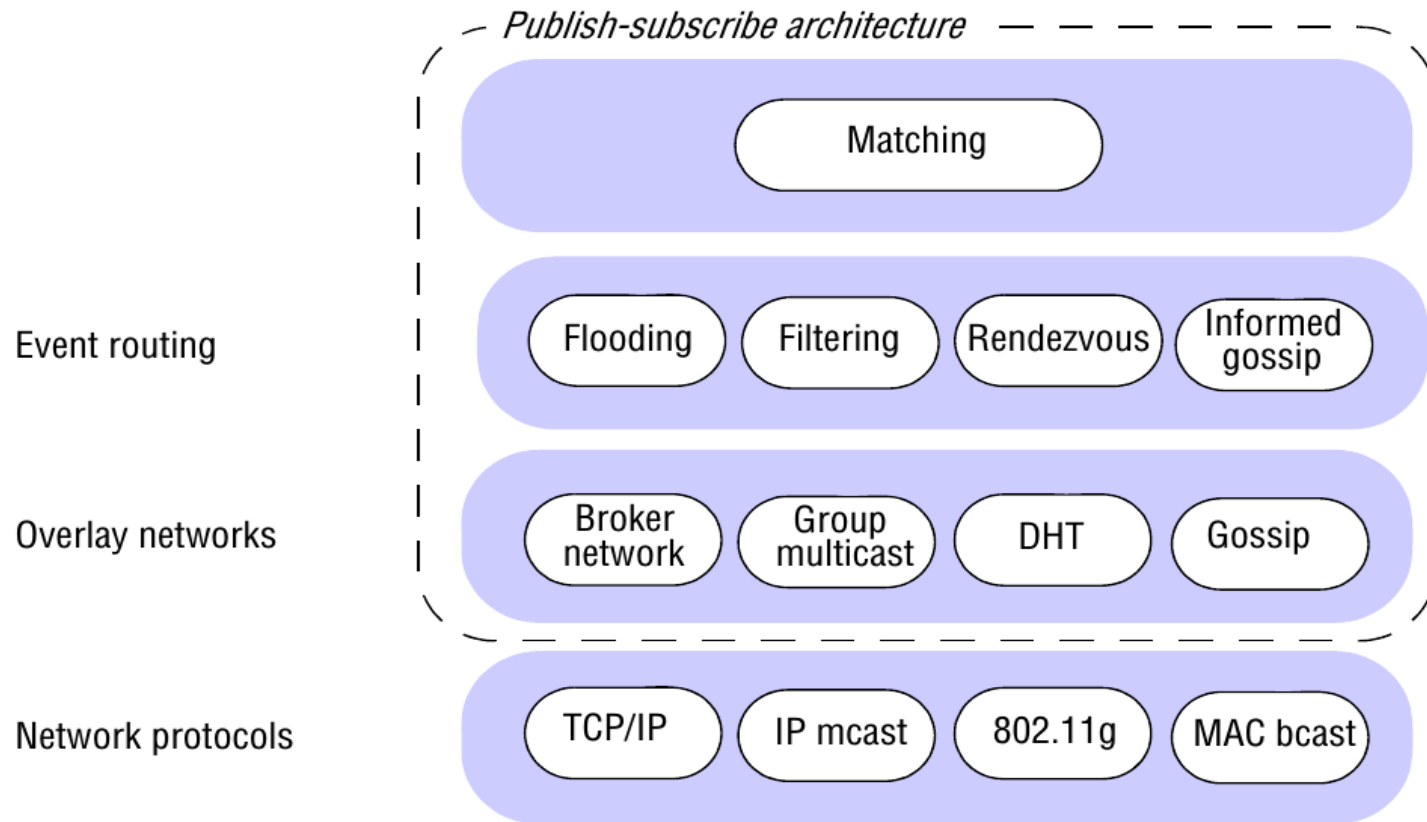


Figura: Arquitetura de sistemas *publish-subscribe*.

Fonte: Coulouris

Sistemas *Publish-Subscribe*

<i>System (and further reading)</i>	<i>Subscription model</i>	<i>Distribution model</i>	<i>Event routing</i>
CORBA Event Service (Chapter 8)	Channel-based	Centralized	-
TIB Rendezvous [Oki <i>et al.</i> 1993]	Topic-based	Distributed	Filtering
Scribe [Castro <i>et al.</i> 2002b]	Topic-based	Peer-to-peer (DHT)	Rendezvous
TERA [Baldoni <i>et al.</i> 2007]	Topic-based	Peer-to-peer	Informed gossip
Siena [Carzaniga <i>et al.</i> 2001]	Content-based	Distributed	Filtering
Gryphon [www.research.ibm.com]	Content-based	Distributed	Filtering
Hermes [Pietzuch and Bacon 2002]	Topic- and content-based	Distributed	Rendezvous and filtering
MEDYM [Cao and Singh 2005]	Content-based	Distributed	Flooding
Meghdoot [Gupta <i>et al.</i> 2004]	Content-based	Peer-to-peer	Rendezvous
Structure-less CBR [Baldoni <i>et al.</i> 2005]	Content-based	Peer-to-peer	Informed gossip

Figura: Exemplos de sistemas publish-subscribe.

Fonte: Coulouris

Filas de Mensagens

- Filas de mensagens (*Message queues*)
 - Comunicação em sistemas distribuídos por meio de filas.
 - Possibilita comunicação ponto-a-ponto por meio do uso de filas de mensagens.
 - Exemplos: IBM WebSphere MQ e Java Messaging Service (JMS).
- Mensagens são persistentes
- Modelo: Produtor/Consumidor

Filas de Mensagens

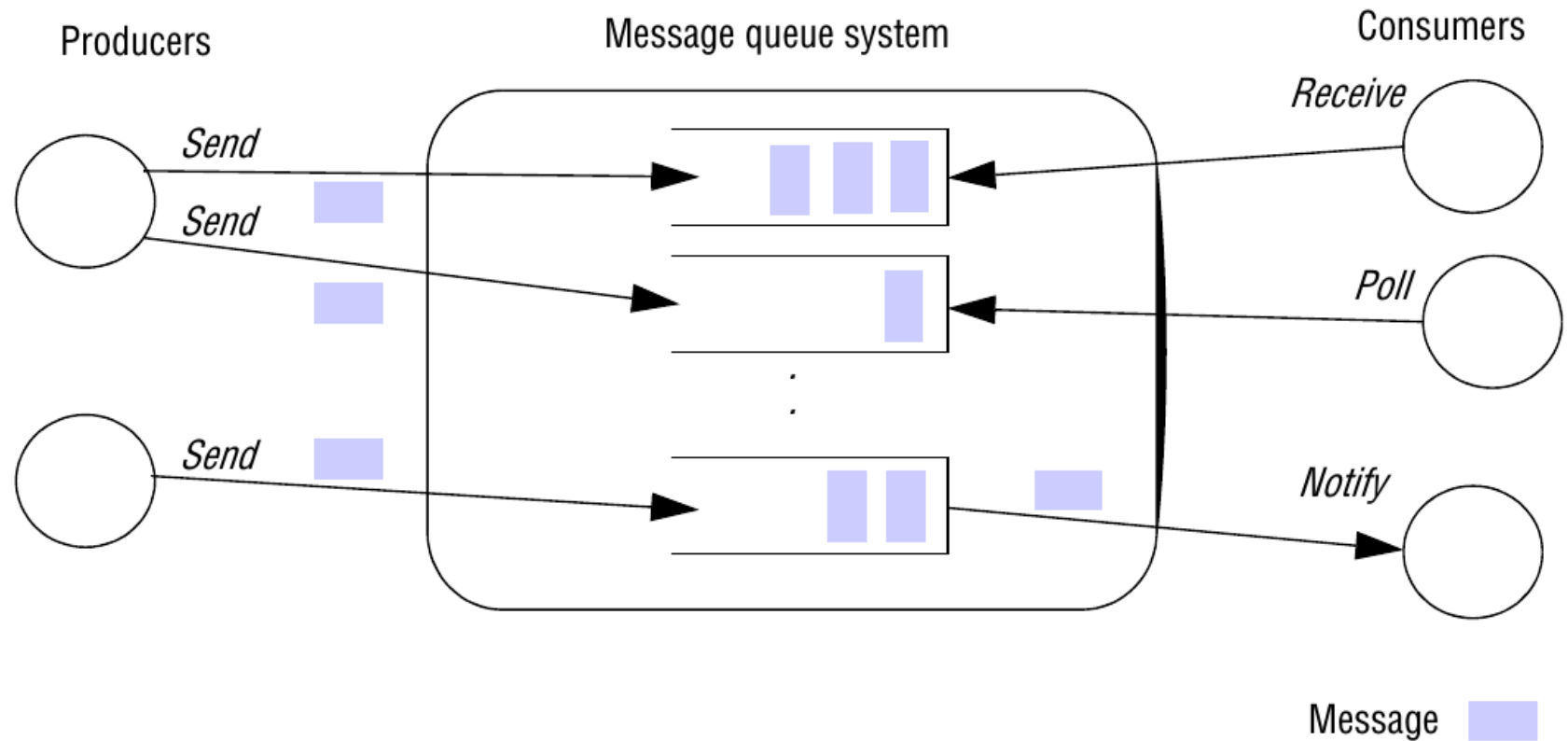


Figura: Modelo de programação usando Filas de Mensagens.

Fonte: Coulouris

Memória Compartilhada

- Memória compartilhada distribuída (*Distributed Shared Memory – DSM*)
 - Compartilhar dados entre computadores que não compartilham memória física.
 - Processos acessam e atualizam memórias definidas para seus espaços.
 - Comumente usada em programação paralela.

Memória Compartilhada

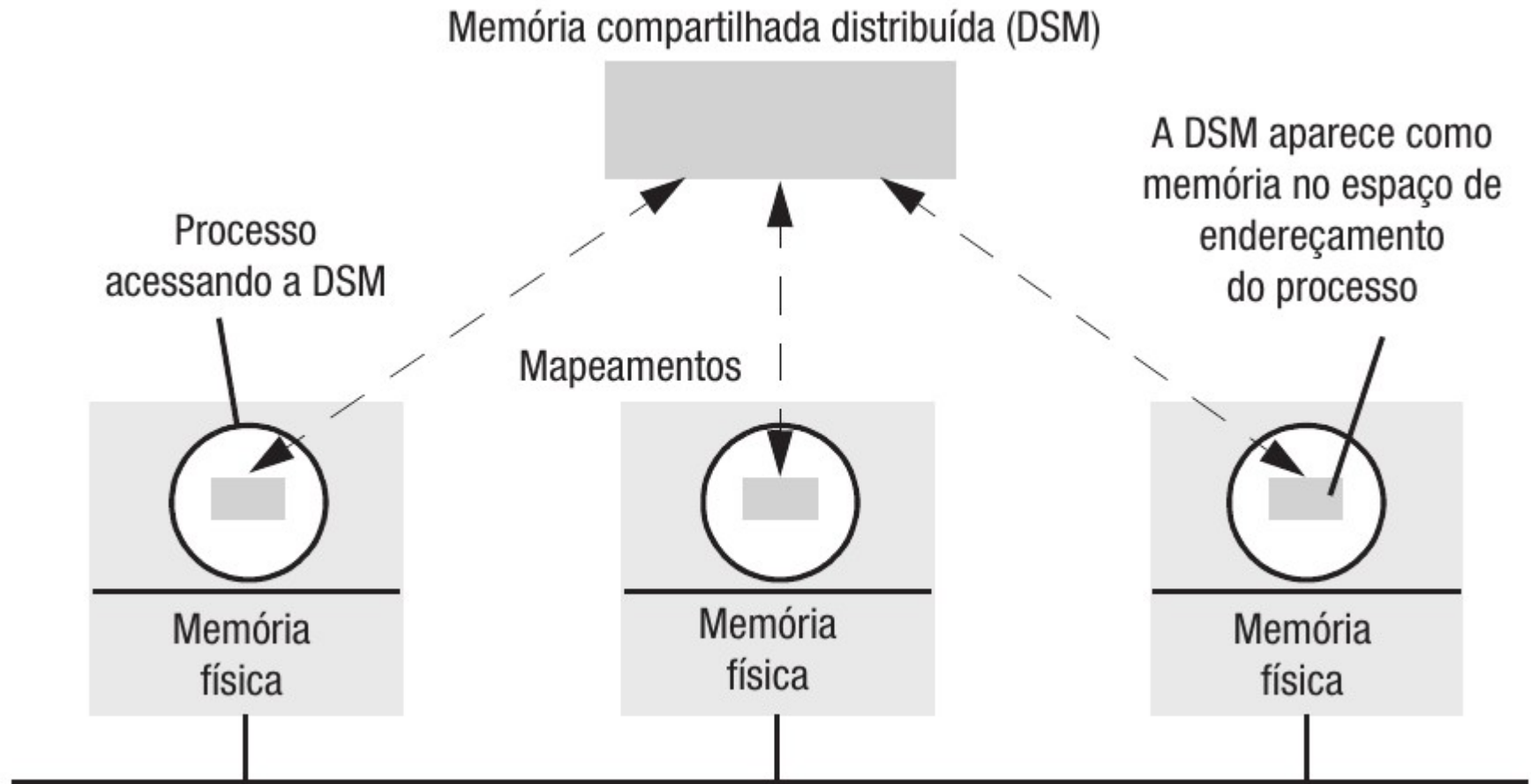


Figura: Abordagem de memória compartilhada distribuída.

Fonte: Coulouris

Memória Compartilhada

- Espaço de tuplas (*Tuple spaces*)
 - Processos comunicam indiretamente colocando tuplas em um espaço de tuplas enquanto outros processos podem ler e remover essas tuplas.
 - Operações: read, write, take
 - Processos buscam especificações de tuplas e o espaço de tuplas retornam as tuplas com as especificações (tipo de endereçamento associativo).
 - Exemplo: Linda, JavaSpaces

Memória Compartilhada

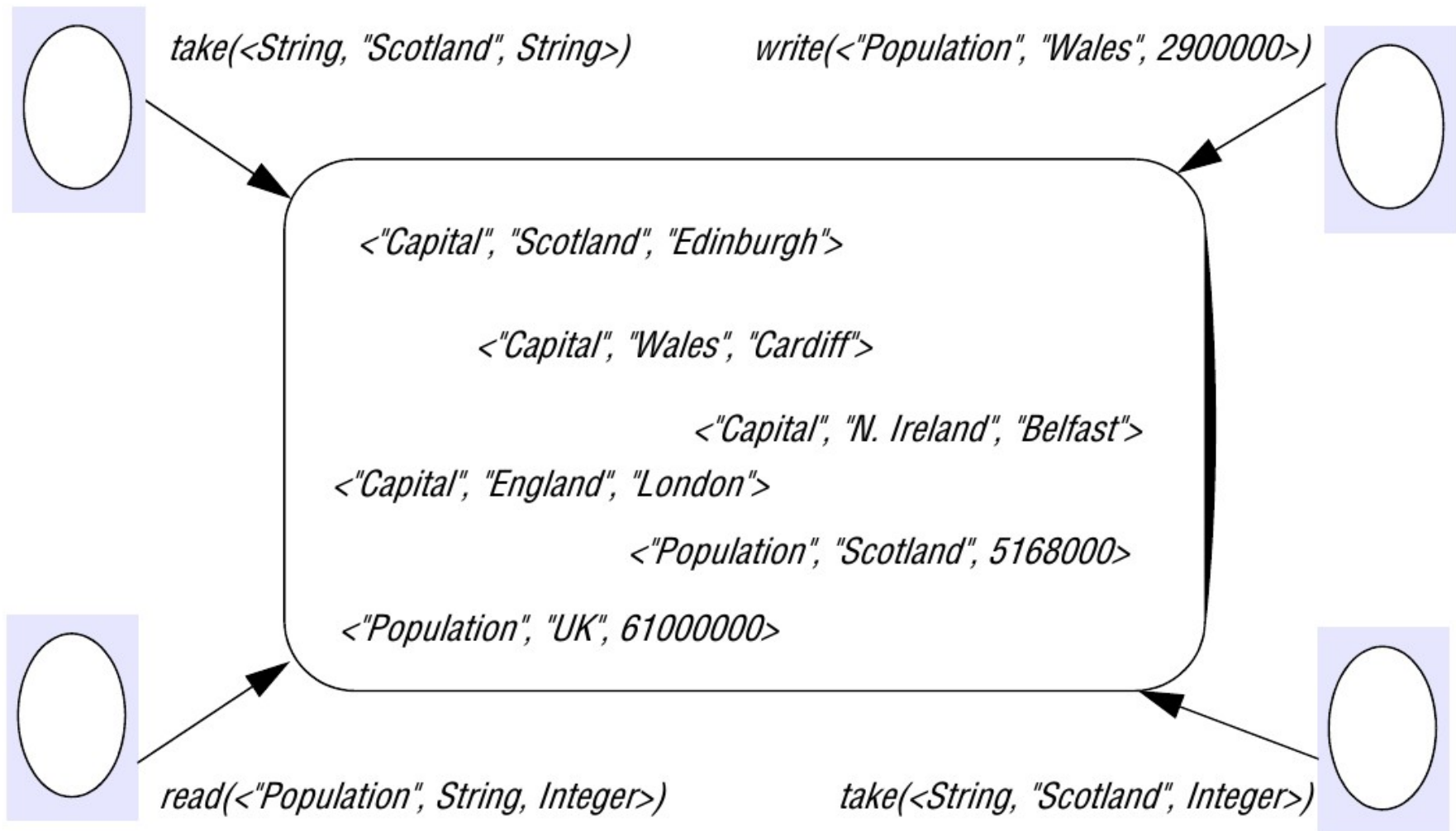


Figura: Abordagem de espaço de tuplas.

Fonte: Coulouris

Resumo

	<i>Grupos</i>	<i>Sistemas publicar- -assinar</i>	<i>Filas de mensagem</i>	<i>DSM</i>	<i>Espaços de tuplas</i>
<i>Desacoplado no espaço</i>	Sim	Sim	Sim	Sim	Sim
<i>Desacoplado no tempo</i>	Possível	Possível	Sim	Sim	Sim
<i>Estilo de serviço</i>	Baseado em comunicação	Baseado em comunicação	Baseado em comunicação	Baseado em estado	Baseado em estado
<i>Padrão de comunicação</i>	Um para muitos	Um para muitos	Um para um	Um para muitos	Um para um ou um para muitos
<i>Principal objetivo</i>	Computação distribuída	Disseminação de informações ou EAI; sistemas móveis e ubíquos	Computação distribuída Disseminação de informações ou EAI; processamento de transações comerciais	Computação paralela e distribuída	Computação paralela e distribuída; sistemas móveis e ubíquos
<i>Escalabilidade</i>	Limitada	Possível	Possível	Limitada	Limitada
<i>Associativo</i>	Não	Somente publicar- -assinar baseada em conteúdo	Não	Não	Sim

Atividades

- Dividir trabalhos práticos na turma sobre os modelos de comunicação indireta.

Referências

COULOURIS, George F; DOLLIMORE, Jean; KINDBERG, Tim; BLAIR, Gordon. **Sistemas distribuídos: conceitos e projeto**. 5. ed. Porto Alegre: Bookman, 2013.