Universidade Regional de Blumenau - FURB
Centro de Ciências Exatas e Naturais – CCEN
Departamento de Sistemas e Computação - DSC

SISTEMAS DISTRIBUÍDOS

MARCOS RODRIGO MOMO
marcos.rodrigomomo@gmail.com

Blumenau, outubro 2024

Middleware Orientado a Mensagens

- RPC/RMI é inadequado para comunicação em alguns cenários de aplicação
  - Cliente e Servidor precisam estar ativos durante a comunicação
  - Implica em espera para estabelecer o sincronismo entre cliente e servidor
  - *Overhead* para manter conexão / sessão
  - Falha de uma das partes impede comunicação
  - Paradigma se limita à comunicação 1->1

Middleware Orientado a Mensagens

- Paradigma de comunicação por mensagens
  - Evita alguns problemas comuns em sistemas baseados em RPC/RMI
  - Outras questões podem ser resolvidas adotando um suporte computacional para comunicação através de mensagens
  - Vários nomes são utilizados para se referir a esse tipo de suporte: Serviço / sistema / barramento / middleware de mensagens / eventos / filas / mailboxes
  - Convencionou-se chamar esse suporte de Middleware Orientado a Mensagens (MOM)

Middleware Orientado a Mensagens

Definição

"Middleware Orientado a Mensagens (MOM) provê suporte para comunicação persistente assíncrona. Esses sistemas oferecem capacidade de armazenamento temporário para mensagens, não exigindo que o emissor e o receptor estejam ativos durante a transmissão da mensagem. Diferentemente de sockets, suportam trocas de mensagens que podem levar vários minutos em vez de alguns segundos ou milissegundos."

Tanenbaum & Van Steen,
Distributed Systems: Principles and Paradigms

Middleware Orientado a Mensagens

Definição

"Middleware Orientado a Mensagens (MOM) provê a abstração de uma fila de mensagens que pode ser acessada através da rede. É uma generalização do mecanismo de Mailbox presente em sistemas operacionais. Apresenta flexibilidade em relação a como programas podem depositar e retirar mensagens da fila. Produtores oferecem filas com persistência, replicação ou desempenho de tempo real.“

David E. Bakken,
Encyclopedia of Distributed Computing

Middleware Orientado a Mensagens

- Tecnologias Relacionadas
  - APIs de comunicação por mensagens (ex.: Sockets)
  - Mecanismos de mailbox
  - Sistemas publish/subscribe
  - Serviços de comunicação por eventos
  - Sistemas de gerenciamento de filas de mensagens

Middleware Orientado a Mensagens

- Vantagens
  - O paradigma de comunicação por mensagens é simples, natural e fácil de entender
  - A reconfiguração de sistemas é simplificada, pois os participantes não precisam conhecer os endereços uns dos outros – basta saberem onde é mantida a fila de mensagens
  - Participantes da comunicação não precisam se sincronizar para trocar dados, o que reduz o tempo ocioso durante a comunicação
  - Participantes não precisam estar permanentemente conectados à rede – basta conectar para enviar/receber mensagens

Middleware Orientado a Mensagens

- Limitações
  - Exigência de um elemento central responsável pelo gerenciamento das filas de mensagens
    - Problemas: ponto único de falha; gargalo na comunicação
    - Solução: replicar esse elemento
  - A comunicação assíncrona pode retardar a entrega de mensagens
    - Problema para aplicações com requisitos de desempenho
    - Solução: filas com prioridades de entrega

Middleware Orientado a Mensagens

- Aplicações
  - Disseminação de informação, em casos nos quais a comunicação síncrona seja inadequada
    - Cotações de ações, status de encomendas, ordens de compra, condições do trânsito, dados meteorológicos, integração da cadeia de produção, auditoria de sistemas, etc.
  - Dispositivos que não possam ficar conectados à rede permanentemente
    - Sensores, celulares, PDAs, RFID, etc.

Middleware Orientado a Mensagens

- Aplicações
  - Integração de Sistemas Legados
    - MOM permite a troca de dados sem que haja forte acoplamento entre os sistemas
    - Exige alterações mínimas nos sistemas legados para enviar/receber mensagens
    - O impacto da comunicação no desempenho é mínimo, devido ao assincronismo
  - Sistemas com interações mais complexas que aquelas permitidas com RPC/RMI
    - Comunicação de grupo (1->N ou M->N)
    - Interação conversacional

Comunicação

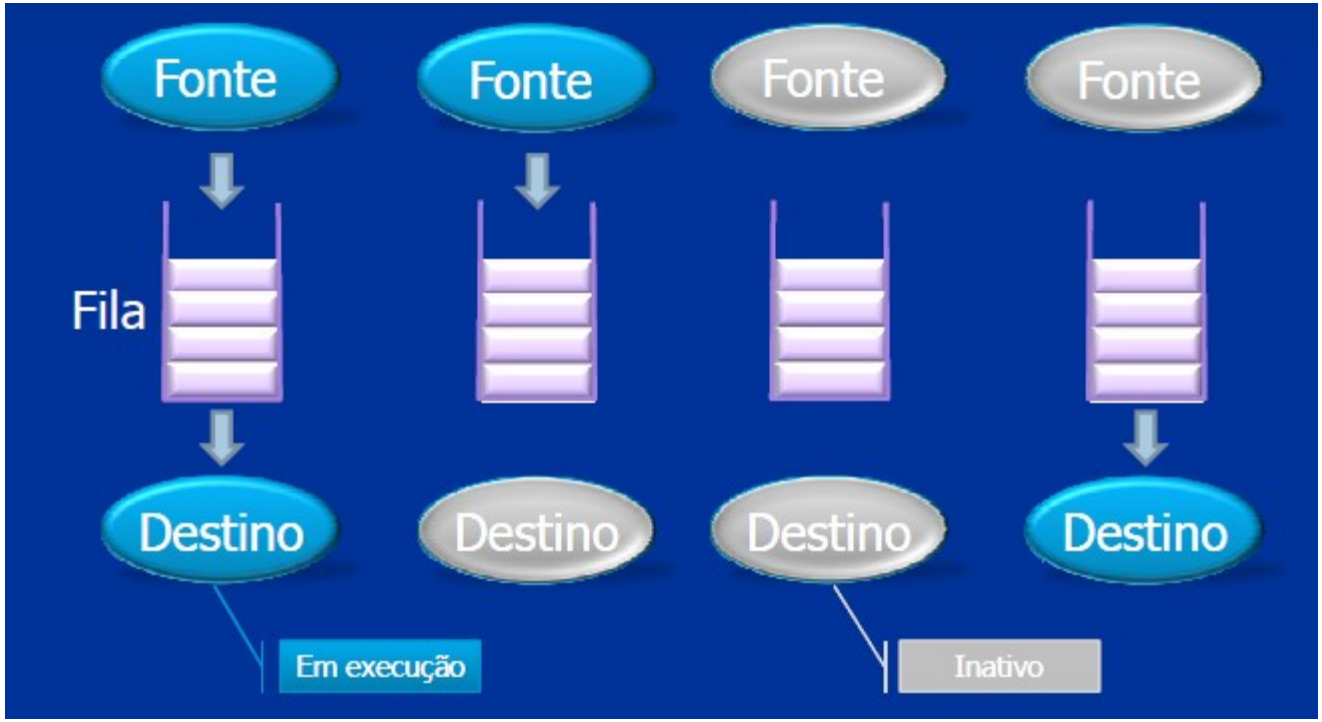
- Características Principais
  - A unidade de comunicação é uma mensagem, semelhante ao que é chamado de evento em mecanismos de eventos
  - Comunicação ocorre de forma assíncrona
  - Um elemento centralizador, possivelmente replicado, gerencia as filas de mensagens

Comunicação

- Primitivas de Comunicação
  - PUT: adiciona uma mensagem a uma determinada fila
  - GET: obtém uma mensagem de uma certa fila, bloqueando caso a mesma esteja vazia
  - POLL: verifica a fila sem bloquear, obtendo uma mensagem caso a fila não esteja vazia
  - NOTIFY: fornece *handler* para ser chamado quando uma mensagem for colocada em uma determinada fila

Comunicação

- Estados possíveis durante a comunicação



Comunicação

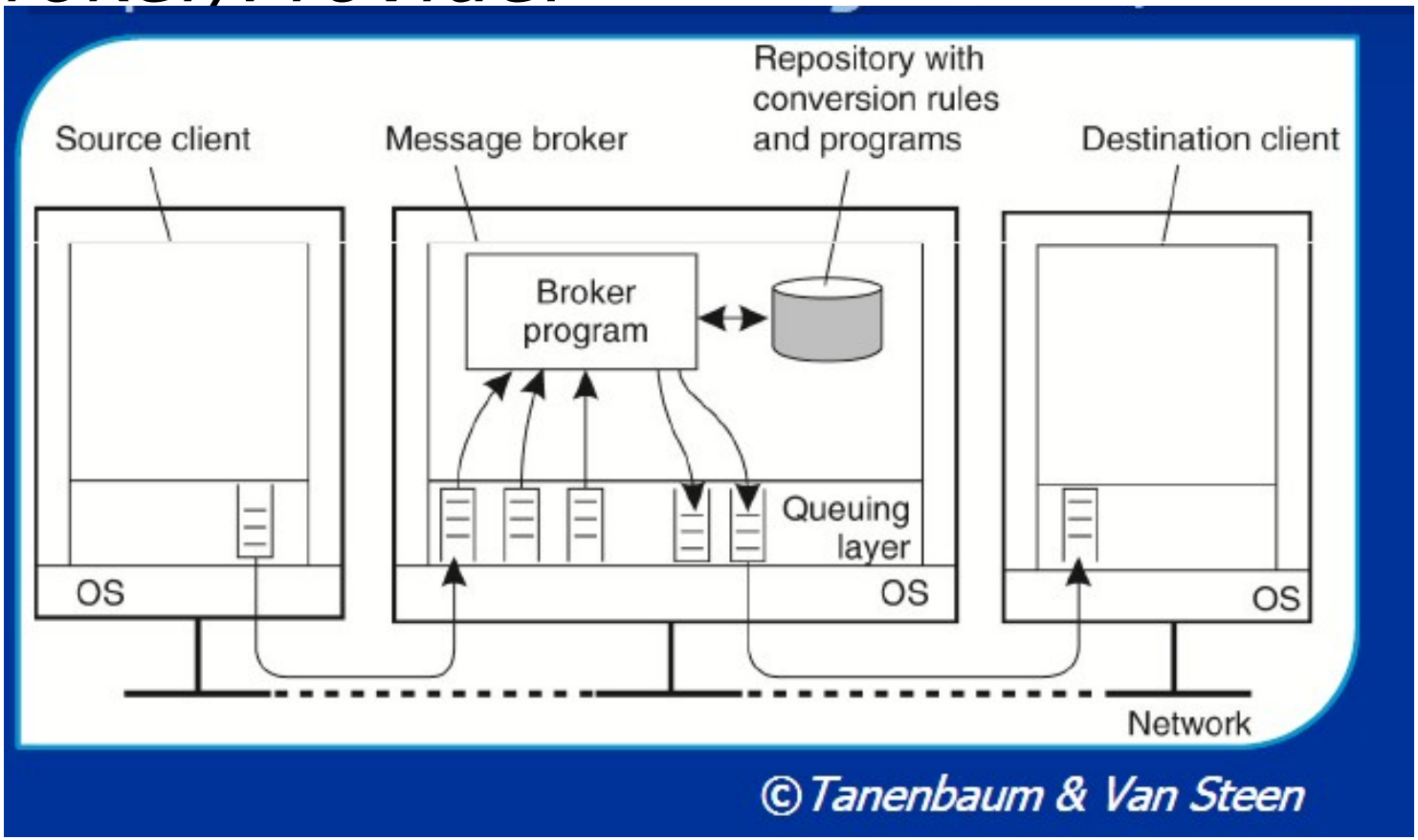
- Formato das mensagens
  - Mensagens podem ter os mais diversos formatos, podendo seguir um formato padrão (string, XML, ...) ou ter formato livre (binário)
    - Essa flexibilidade contrasta com RPC/RMI, onde os parâmetros são em geral tipados
  - Cada fila pode adotar um formato próprio
  - Mensagens podem ter um assunto/tópico, que pode ser usado por clientes para filtragem
  - Regras de conversão podem ser aplicadas às mensagens antes de serem colocadas na fila

Comunicação

- Suporte de Comunicação
  - O MOM pode ser construído sobre os mais diversos mecanismos de comunicação, desde os de mais baixo nível (ex.: Sockets) até os de mais alto nível (ex.: RMI/RPC, Web Services)
  - Os participantes da comunicação utilizam uma API simples para enviar/receber mensagens
  - O elemento principal envolvido na comunicação é o *Message Broker/Provider*, que intermedia a interação entre os participantes e gerencia as filas de mensagens

Comunicação

- Arquitetura de um *Message Broker/Provider*



Comunicação

- Confiabilidade
  - Requisitos de confiabilidade podem ser impostos no envio de mensagens
  - Protocolos de entrega confiável de mensagens podem ser usados
  - Acordo e ordenação podem ser observados na entrega das mensagens a grupos de destinatários
  - Coordenação é efetuada pelo *Message Broker*

Comunicação

- Confidencialidade e controle de acesso
  - Mensagens podem ser criptografadas, de modo a impedir acesso não-autorizado
  - Filas podem ter controle de acesso, impondo restrições quanto a quem pode produzir e consumir mensagens
  - Controle é feito pelo *Message Broker*

Gerenciamento de Filas

- A principal função do *Message Broker* é gerenciar filas de mensagens
  - Filas podem não ter ordem definida ou ter ordem FIFO, LIFO (pilha), por prioridade, ...
  - Filas podem ser persistentes ou não
  - Quando lidas, as mensagens podem ser mantidas ou retiradas da fila
  - Mensagens podem ter um 'prazo de validade'

Gerenciamento de Filas

- Semelhanças entre *Message Brokers* e SGBDs
  - Armazena persistentemente mensagens/dados
  - Permite a criação de filas/tabelas
  - Executa transações para adição/remoção de mensagens/dados das filas/tabelas
  - Efetua indexação para agilizar o acesso às mensagens/dados
  - Provê mecanismos avançados de busca
  - Dispara gatilhos quando uma mensagem/dado for adicionado a uma fila

Gerenciamento de Filas

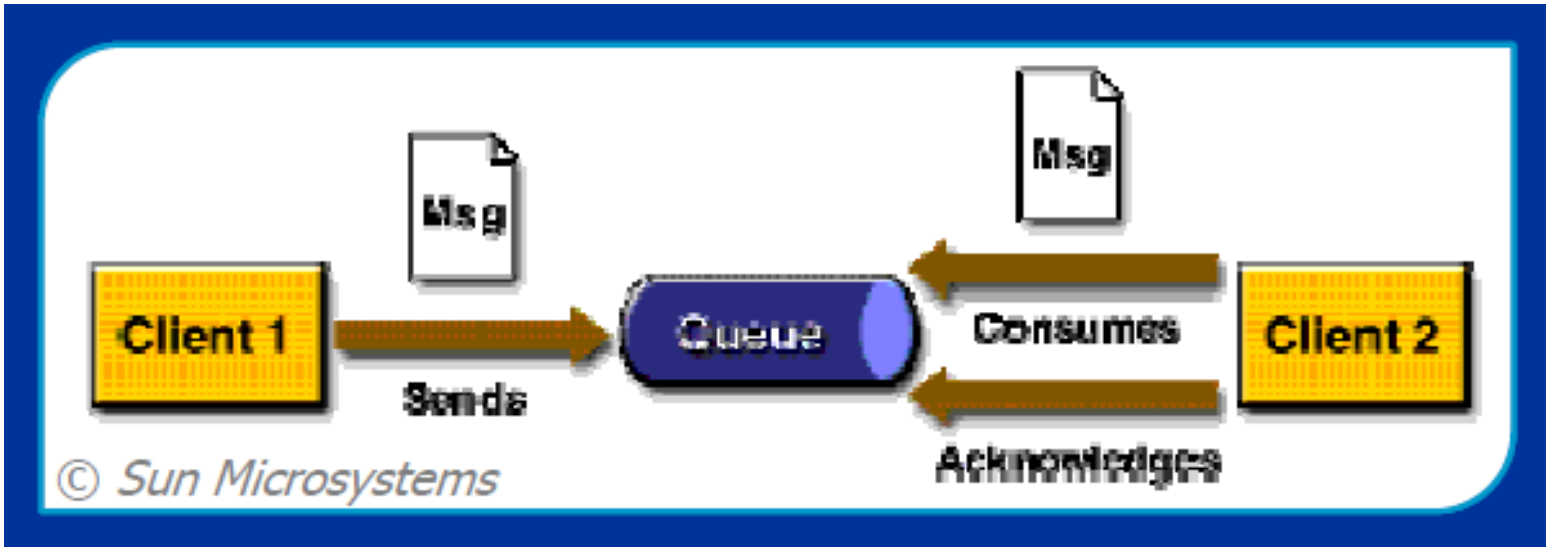
- Filtragem por tópicos
  - O receptor pode filtrar as mensagens que recebe com base em tópicos/assuntos
  - Benefícios:
    - Reduz o tráfego na rede
    - Elimina a necessidade de tratar mensagens que não interessam ao receptor
  - Processo de filtragem é efetuado pelo *broker* com base nos parâmetros de filtragem especificados pelos receptores

Padrões e Produtos

- *Java Message Service* (JMS)
  - Padrão de interface para acesso a MOMs
  - Independente de fornecedor, mas não de linguagem
  - Suportado por diversos MOMs e por grande parte dos servidores de aplicação
  - Elementos
    - Provedor JMS
    - Clientes JMS
      - Produtores
      - Consumidores

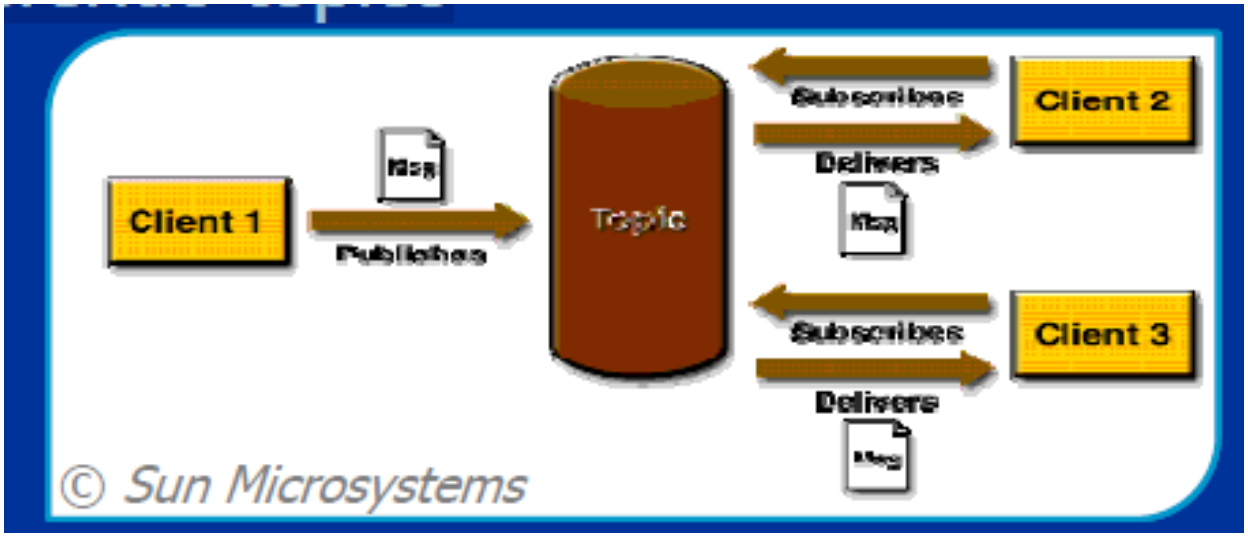
Padrões e Produtos

- *Java Message Service* (JMS)
  – Modelo de comunicação ponto-a-ponto: mensagem é endereçada a uma fila e é lida por apenas um consumidor (dentre vários)



Padrões e Produtos

- *Java Message Service (JMS)*
  - Modelo de comunicação *Publish/Subscribe*: mensagens são associadas a um tópico, e podem ser lidas por vários assinantes que optarem por receber mensagens sobre o referido tópico



Padrões e Produtos

- Advanced Message Queuing Protocol (AMQP)
  - Proposto pelo AMQP Working Group, que reúne grandes instituições financeiras (Credit Suisse, JPMorgan, Goldman Sachs, etc.) e empresas da área de informática (Cisco, Novell, Red Hat, etc.)
  - Define o comportamento do provedor de mensagens e de seus clientes
  - Permite que implementações de diferentes fabricantes interoperem
  - Independente de linguagem e de plataforma

Padrões e Produtos

- IBM MQ
  - Middleware de suporte à sistemas de mensagens
  - Muito utilizando no mercado
  - Sistemas operacionais suportados: AIX, i5/OS, HP UX, Linux, Solaris, Windows, z/OS.
  - Linguagens suportadas:
    - Java através da interface JMS C, C++ e C# via Multi-Language Message Service (XMS) C, COBOL e Assembly via MQ Interface
  - Integração com:
    - HTTP / AJAX (Web 2.0)
    - SOAP / Web Services

Padrões e Produtos

- **Apache ActiveMQ**
- BEA WebLogic JMS
- JBoss Messaging
- Microsoft Message Queue Server (MSMQ)
- Oracle Advanced Queueing (AQ)
- Sun Java System Message Queue (SJS MQ)
- ...

Referências

- Prof. Frank Siqueira, Departamento de Informática e Estatística, UFSC.

- http://www.coderpanda.com/jms-example-using-apache-activemq/