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

Sistemas Distribuídos

Aula Prática: Comunicação Indireta - JMS

Prof. Rodrigo Campiolo

13/09/20

Introdução

- Conceitos
  - **Messaging**: é um método de comunicação entre componentes de software ou aplicações de software por troca de mensagens. É um método de comunicação indireta.
  - **JMS API**: *Java Message Service* é uma API Java que permite a aplicações criar, enviar, receber e ler mensagens. Fornece interfaces e um modelo de programação para acessar provedores (implementações) de serviços de mensagem.

Serviços de Mensagens

- Exemplos de Serviços de Mensagens
  - RabbitMQ (https://www.rabbitmq.com/)
  - Apache ActiveMQ (https://activemq.apache.org/)
  - Apache Qpid (https://qpid.apache.org/)
  - Apache RocketMQ (https://rocketmq.apache.org)
  - Apache Kafka (https://kafka.apache.org/)
  - IBM MQ

Serviços de Mensagens

- Protocolos
  - Advanced Message Queuing Protocol – **AMQP**
    - https://www.amqp.org/
  - Simple Text Oriented Messaging Protocol – **STOMP**
    - https://stomp.github.io/
  - Message Queuing Telemetry Transport – **MQTT**
    - http://mqtt.org

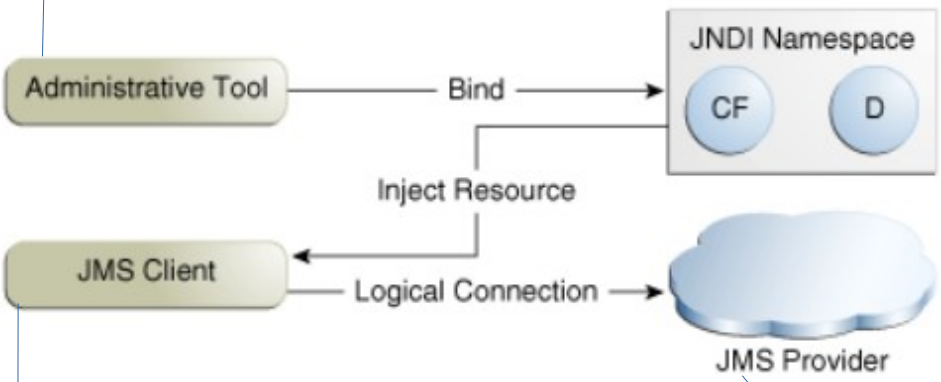
JMS API

- Quando usar a JMS API
  - Situações em que os componentes devem ser independentes de interfaces de outros componentes, logo podem ser facilmente substituídos.
  - Situações que componentes devem continuar executando sem que outros componentes estejam ativos.
  - Situações que necessitam que componentes enviem informações para outros, mas continuem executando sem receber uma resposta imediata.

Arquitetura da JMS API

Permitem mapear objetos administrados no JNDI.

- Connection Factory: cliente usa para criar uma conexão com o provedor.
- Destination: cliente usa para especificar o destino das mensagens ou a origem das mensagens que consome.



API Java para um serviço de diretório.

Programas ou componentes que produzem e consomem as mensagens.

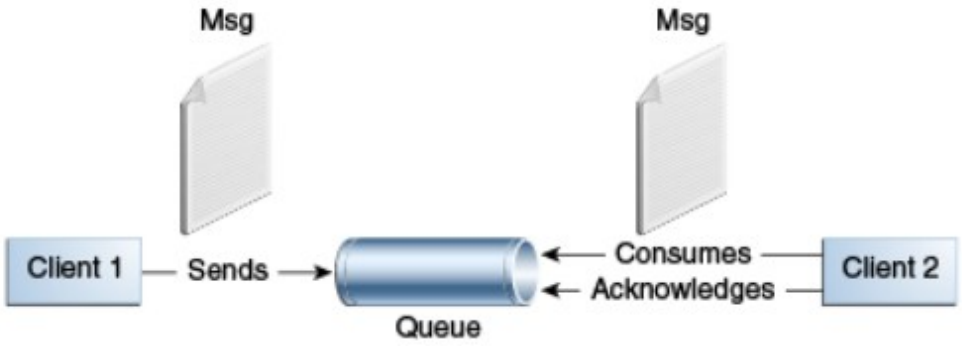
Sistema ou serviço de mensagens que implementa as interfaces JMS e provê funcionalidades para administração e controle.

Messages: objetos para encapsular as mensagens trocadas entre os clientes.

Fonte: Oracle

JMS – Filas de Mensagens

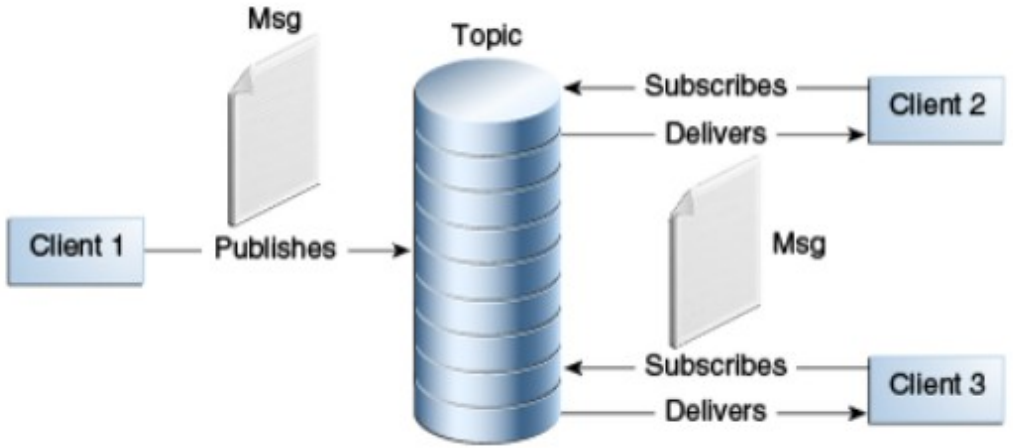
- Uso do JMS para comunicação por meio de filas de mensagens (point-to-point (PTP)).
- Mensagens destinadas para uma fila específica.
- Filas mantêm as mensagens até serem consumidas ou expiradas.



Fonte: Oracle

JMS – Publish/Subscribe

- Uso do JMS para comunicação via publicador/ assinante.
- Clientes associam mensagens a um *tópico*.
- Cada mensagem tem múltiplos consumidores.



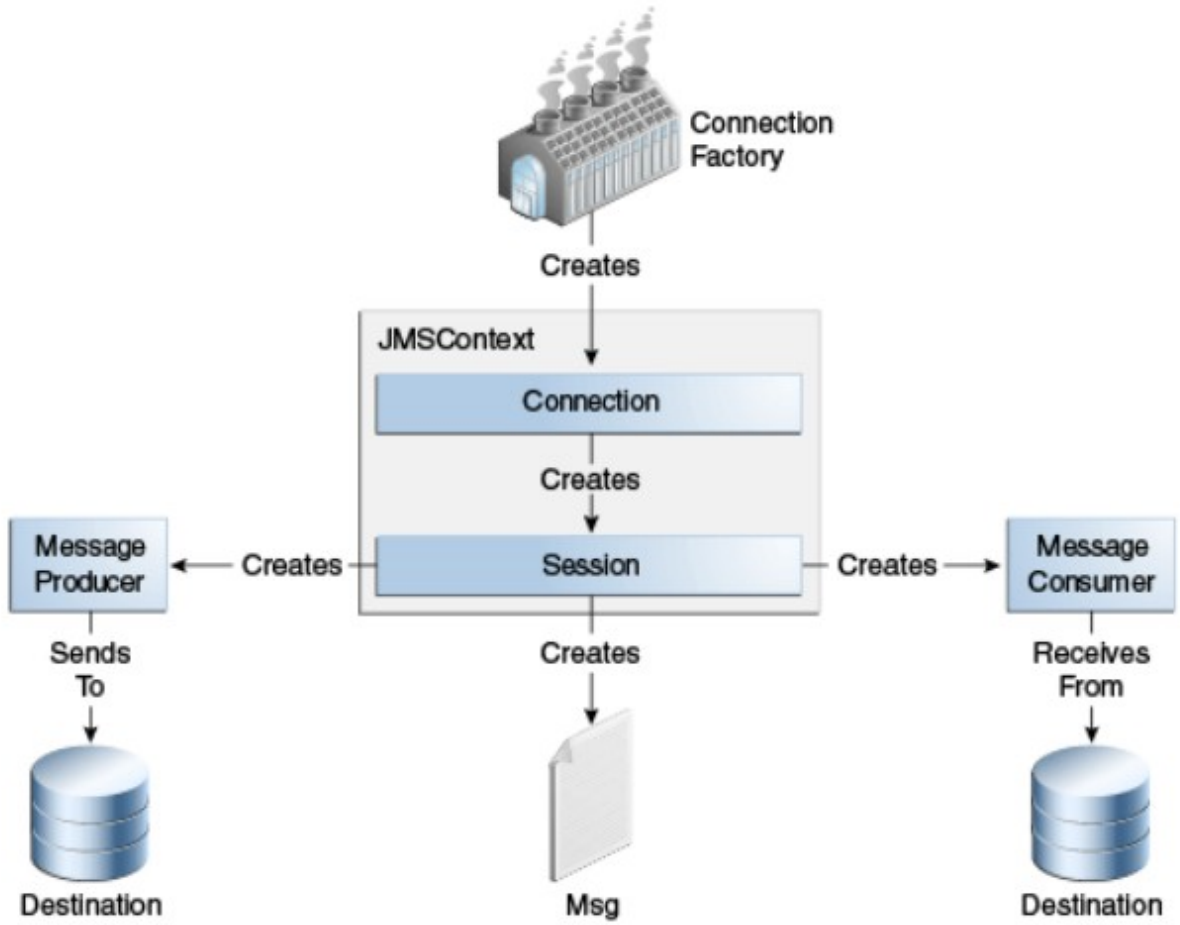
Fonte: Oracle

Consumo da Mensagem

- Síncrona: consumidor explicitamente obtém a mensagem por um método *receive* que bloqueia o receptor até a chegada da mensagem ou após a expiração de um tempo (timeout).

- Assíncrona: registro de um **message listener** com um consumidor. O provedor anuncia a chegada da mensagem e o receptor recebe o conteúdo.

JMS – Modelo de Programação



Fonte: Oracle

JMS

- **Connection Factory:** objeto que o cliente usa para criar uma conexão com o provedor.
- **Destination:** objeto que o cliente usa para especificar o destino das mensagens produzidas ou a origem das mensagens consumidas.
- **Connection:** encapsula uma conexão virtual com um provedor JMS (*middleware*).
- **Session:** contexto *single-threaded* para produzir e consumir mensagens.
- **JMSContext:** objeto que combina uma conexão e sessão em um único objeto.

JMS

- **Message Producer:** objeto que é criado por um JMSContext ou uma sessão e usado para enviar mensagens para um destino.
- **Message Consumer:** objeto que é criado por um JMSContext ou uma sessão e é usado para receber mensagens que foram enviadas para um destino.
- **Message Listener:** objeto que atua como um tratador de eventos assíncrono para mensagens.
- **Message Selector:** permite um consumidor de mensagens especificar as mensagens de interesse.

Apache Qpid

- É um sistema de mensagens (*middleware*) que implementa o protocolo **AMQP - Advanced Message Queuing Protocol**. (https://www.amqp.org/)
- AMPQ é um protocolo aberto para enviar e receber mensagens de forma confiável.
- Provê dois tipos de componentes:
  - Messaging APIs
  - Messaging Servers (Message Brokers)

JMS Exemplo – Apache Qpid

- Sender

```
Context context = new InitialContext();
ConnectionFactory factory = (ConnectionFactory) context.lookup("myFactoryLookup");
Destination queue = (Destination) context.lookup("myQueueLookup");

Connection connection = factory.createConnection(USER, PASSWORD);
connection.setExceptionListener(new MyExceptionListener());
connection.start();

Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);

MessageProducer messageProducer = session.createProducer(queue);

TextMessage message = session.createTextMessage("Olá Fila!");

messageProducer.send(message, DELIVERY_MODE, Message.DEFAULT_PRIORITY,
Message.DEFAULT_TIME_TO_LIVE);
```

JMS Exemplo – Apache Qpid

- Receiver

```
Context context = new InitialContext();
ConnectionFactory factory = (ConnectionFactory) context.lookup("myFactoryLookup");
Destination queue = (Destination) context.lookup("myQueueLookup");

Connection connection = factory.createConnection(USER, PASSWORD);
connection.setExceptionListener(new MyExceptionListener());
connection.start();

Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);

MessageConsumer messageConsumer = session.createConsumer(queue);

int timeout = 1000;
Message message = messageConsumer.receive(timeout);
```

JMS Exemplo – Apache Qpid

- jndi.properties

```
# Set the InitialContextFactory class to use
java.naming.factory.initial = org.apache.qpid.jms.jndi.JmsInitialContextFactory

# Define the required ConnectionFactory instances
# connectionfactory.<JNDI-lookup-name> = <URI>
connectionfactory.myFactoryLookup = amqp://localhost:5672

# Configure the necessary Queue and Topic objects
# queue.<JNDI-lookup-name> = <queue-name>
# topic.<JNDI-lookup-name> = <topic-name>
queue.myQueueLookup = queue
topic.myTopicLookup = topic
```

Referências

JMS API – The Java EE Tutorial. Disponível em https://docs.oracle.com/javaee/7/tutorial/jms-concepts.htm. Acessado em 27/10/2016.

Apache Qpid. Chapter 4. Examples. Disponível em https://qpid.apache.org/releases/qpid-0.32/jms-client-0-8/book/JMS-Client-0-8-Examples.html. Acessado em 27/10.2016.