中间件技术
Middleware Technology

第七章　消息中间件

赖永炫　博士
厦门大学　软件学院

大纲

- 消息中间件的概念
- 消息中间件的历史
- 消息中间件的架构和要素
- JAVA消息中间件JMS
- 消息驱动Bean （MDB）
- 小结

复习

- 分布式对象调用（如CORBA，RMI和DCOM）提供了一种通讯机制，能透明地在异构的分布式计算环境中传递对象请求。
- 对象可以位于本地或远程机器，分布式对象调用在对象与对象之间提供一种统一的接口，使对象之间的调用和数据共享不再关心对象的位置、实现语言及所驻留的操作系统。

分布式对象调用的局限性

- 同步通信：客户发出调用后，一般需要等待服务对象完成处理并返回结果后才能继续执行。

- 客户和服务对象的生命周期紧密耦合：客户进程和服务对象进程都必须正常运行，如果由于服务对象崩溃或网络故障导致客户的请求不可达，客户会接收到异常。

概 念

- 消息中间件（Message Oriented Middleware）是在分布式系统中完成消息的发送和接收的基础软件。

- 分布式应用程序之间的通信接口由消息中间件提供，消息中间件支持与保障分布式应用程序之间同步或异步的收发消息。

消息中间件的通信方式

- 发送方在发送消息时不必知道接收方的状态，更无需等待接收方的回复。
- 接收方在收到消息时也不必知道发送方的目前状态，更无需进行同步的消息处理。
- 消息的收发双方完全是松耦合的，通信是非阻塞的；收发双方彼此不知道对方的存在，也不受对方影响。
- 消息发送者可以将消息间接传给多个接收者，大大提高了程序的性能、可扩展性及健壮性。

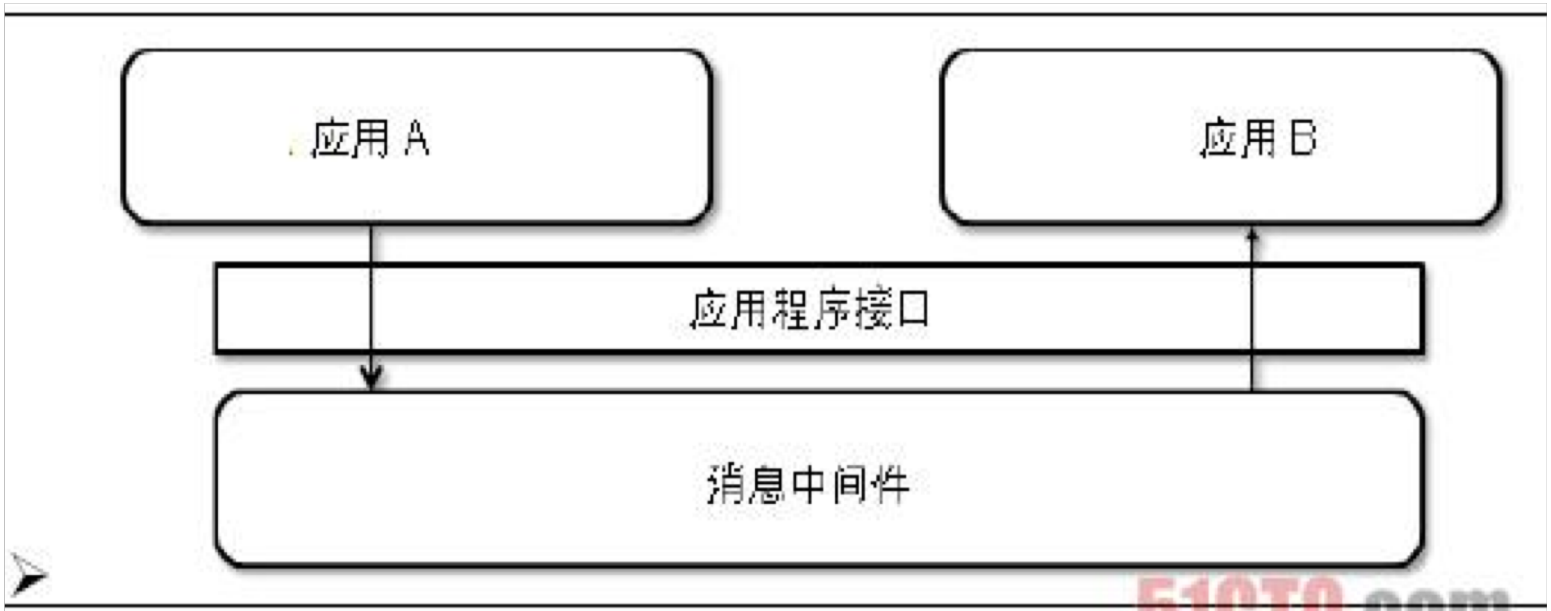


图 2-25　消息中间件

发展历史

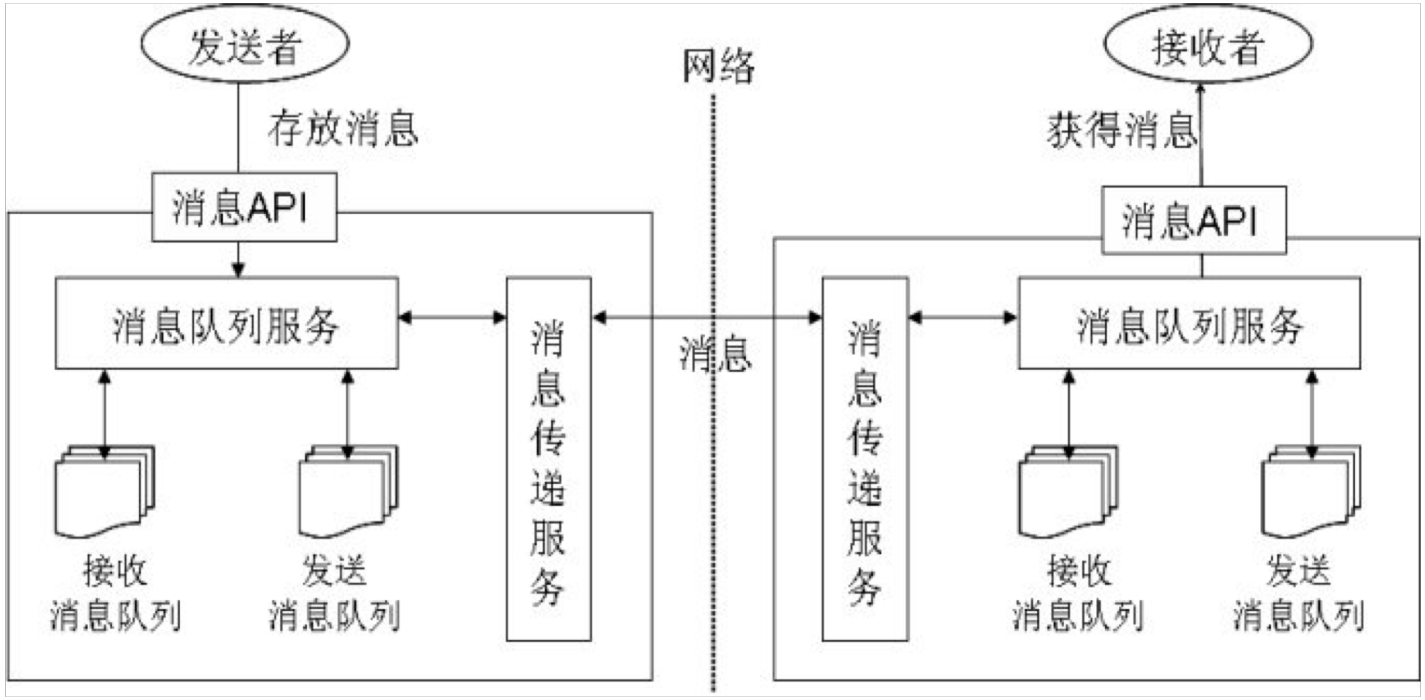
- 80年代后期，IBM推出了消息中间件产品MQSeries
- 90年代，OMG制定了公共对象服务标准（COSS），其中对消息服务进行了规范。
- 90末期，消息中间件开始向发布/订阅架构转变，并成为企业应用集成中间件的一种核心机制

发展历史

- 进入21世纪，由于J2EE技术的广泛应用，J2EE的消息服务规范JMS（Java Message Service）得到消息中间件厂商的广泛采纳，并逐渐成为消息中间件的事实标准
- W3C组织定义了Web服务的可靠消息传送规范（WS-Reliable Messaging）
- 典型的消息中间件包括IBM WebSphere MQSeries、Active MQ 、SonicMQ 、Tibco TIB/Rendezvous和Microsoft MSMQ等。

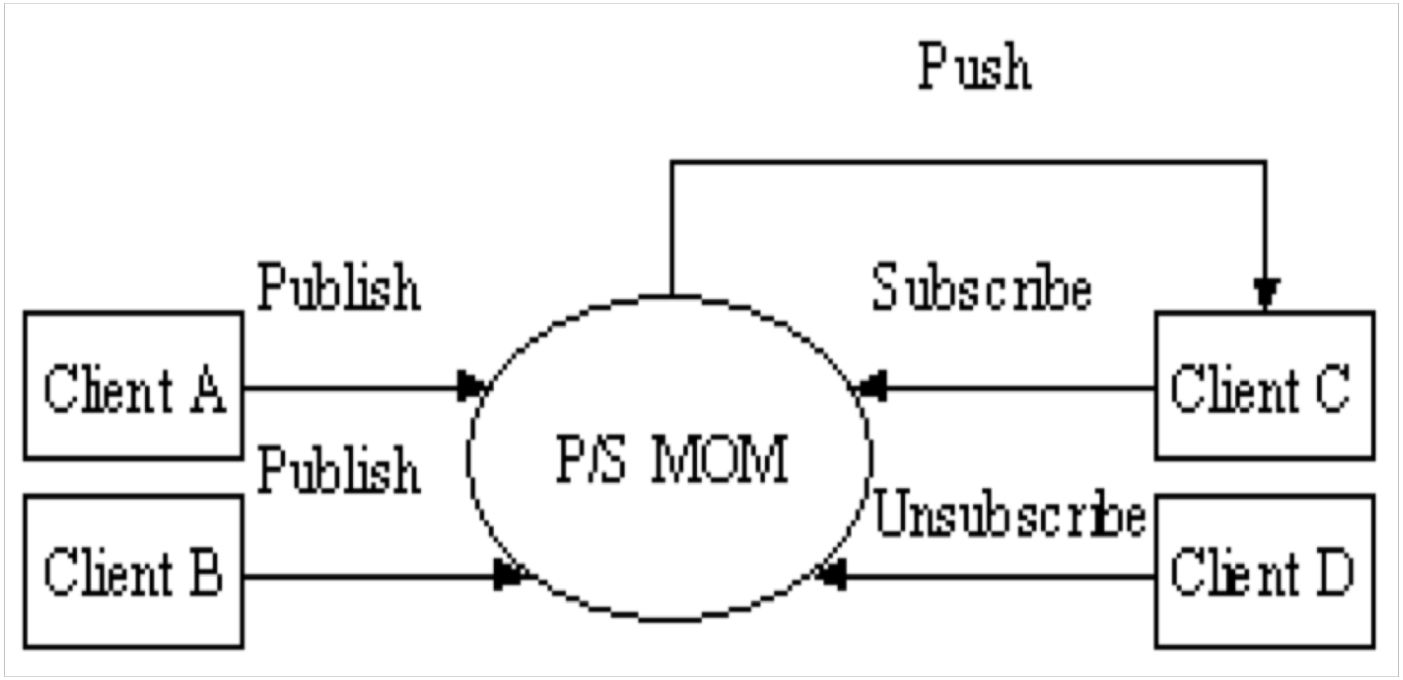
消息中间件的架构

- 传统的点对点消息中间件通常由消息队列服务、消息传递服务、消息队列和消息应用程序接口API组成



- 消息发送者调用发送消息的API函数，将需要发送的消息经消息队列服务存储到发送消息队列中；
- 通过双方消息传递服务之间的交互，消息队列服务将需要发送的消息从发送队列取出，并送到接收方；
- 接收方的消息队列服务将接收到的消息存放到接收消息队列中；
- 消息接收者调用接收消息的API函数，同样经过消息队列服务，将需要的消息从接收队列中取出，并进行处理。
- 消息在发送或接收成功后，消息队列服务将对相应的消息队列进行管理。

- 在基于消息代理的分布式应用系统中，消息的发送方称为出版者（publisher），消息的接收方称为订阅者（subscriber），不同的消息通过不同的主题进行区分。



消息代理的工作流程

- 消息发布者和订阅者分别同消息代理进行通信。消息发布者将包含主题的消息发布到消息代理；消息订阅者向消息代理订阅自己感兴趣的主题。
- 消息代理对双方的主题进行匹配后，不断将订阅者感兴趣的消息推(Push)给订阅者，直到订阅者向消息代理发出取消订阅的消息。

解耦

- 消息代理实现了发布者和订阅者的解耦：
- 时间解耦：发布方和订阅方无需同时在线就能够进行消息传输，消息中间件通过存储转发提供了这种异步传输的能力；
- 空间解耦：发布方和订阅方都无需知道对方的物理地址、端口，甚至无需知道对方的逻辑名字和个数；
- 流程解耦：发布方和订阅方在发送和接收数据时并不阻塞各自的控制流程。

消息中间件的要素

- 无论是点对点消息中间件还是消息代理，消息中间件的体系结构都是非常清晰简单的。
- 但由于分布式应用及其环境的多样性和复杂性，导致了消息中间件的复杂性：
  - ✓跨平台、跨语言
  - ✓存储转发或消息路由
  - ✓消息传输的安全性、事务性、时限等质量要求
  - ✓持久存储能力

消息的表示

- 消息头：用于描述消息发送者和接收者的地址或消息主题，以及消息的服务质量要求，例如，消息传输的时限、优先级、安全属性等；
- 消息体：用于描述消息中具体携带的信息内容。
- 目前多采用XML作为消息表示的格式。

消息队列

- 为了有效控制消息收发过程而在消息中间件中内置的存储消息的数据结构。
- 由于消息多采用先进先出的控制方式，通常采用队列作为消息的存储结构。
- 队列的分类：
  - ✓消息的内容：发送队列、接收队列、死信队列
  - ✓消息发送的质量：优先队列、普通队列。
  - ✓队列存储介质：持久消息队列、内存队列和高速缓存队列。

队列的分类

- 持久消息队列：
  - ✓基于数据库或文件系统，提供消息持久存储功能，同时又具有最小的内存开销，适合于消息需要可靠传输的应用环境。
- 内存队列：
  - ✓基于内存的消息队列。不提供消息持久功能，完全基于内存来进行消息的缓存和分发。适合对性能要求非常苛刻，但是消息无需可靠持久的应用环境。
- 高速缓存队列：
  - ✓基于数据库和内存Cache。提供可靠持久功能，同时又使用内存作为Cache，具有最大的资源开销，同时又具有很高的性能。适合于大部分应用场合。

消息路由

- 消息路由借用了IP层的路由和路由器中的路由的概念
- 但不同之处在于：消息路由属于应用层的概念
  - ✓它是为了保证应用之间的消息交换处于可控的状态而设计的软件功能模块，其机制是按照消息路由规则将消息从发送者传送到目标应用，并提供消息流量控制功能。
- 消息路由有时也称为"流量控制"、"基于内容的路由"、"智能路由"。

消息QoS机制

- 服务质量(Quality of Service，简称QoS)是指与用户对服务满意程度相关的各种性能效果。
- 消息QoS机制是指消息中间件提供的消息传送过程中在性能、安全、可靠性等方面的各种非功能型需求约束。
  - ✓ 可靠性/事务性/安全性/优先级/时间约束/队列管理

JAVA消息中间件JMS

- JMS即Java消息服务（Java Message Service），是Java平台上有关面向消息中间件(MOM)的技术规范
- 便于消息系统中的Java应用程序进行消息交换,并且通过提供标准的产生、发送、接收消息的接口简化企业应用的开发。

- JMS 通过 MOM 产品为 Java 程序提供了一个发送和接收消息的标准的、便利的方法。用 JMS 编写的程序可以在任何实现 JMS 标准的 MOM 上运行。

消息分类

- 消息有下面几种类型，他们都是派生自 Message 接口。
  - ✓ StreamMessage：一种主体中包含 Java 基元值流的消息。其填充和读取均按顺序进行。
  - ✓ MapMessage：一种主体中包含一组名-值对的消息。没有定义条目顺序。
  - ✓ TextMessage：一种主体中包含 Java 字符串的消息（例如，XML 消息）。
  - ✓ ObjectMessage：一种主体中包含序列化 Java 对象的消息。
  - ✓ BytesMessage：一种主体中包含连续字节流的消息

JMS模型

- JMS 支持两种消息传递模型：
  - ✓ 点对点（point-to-point，简称 PTP）
  - ✓ 发布/订阅（publish/subscribe，简称 pub/sub）

- 这两种模型都通过扩展公用基类来实现
  - ✓ javax.jms.Queue 和javax.jms.Topic 都扩展自 javax.jms.Destination 类。

投递方式

- JMS现在有两种传递消息的方式.
- 标记为NON_PERSISTENT的消息最多投递一次
- 标记为PERSISTENT的消息将使用暂存后再转送的机理投递。
  - ✓ 如果一个JMS服务离线，那么持久性消息不会丢失但是得等到这个服务恢复联 机时才会被传递。
- JMS定义了从0到9的优先级路线级别，0是最低的优先级而9则是最高的。

- JMS 编程案例

消息驱动bean

- (Message Driven Bean)是用来转换处理基于消息请求的组件。
- MDB负责处理消息，而EJB容器则负责处理服务(事务、安全、资源、并发、消息确认,等等),使bean开发者把精力集中在消息处理的业务逻辑上。
- MDB它和无状态Session Bean一样也使用了实例池机制,容器可以为它创建大量的实例,用来并发处理成百上千个JMS消息。

MessageListener 接口

- MDB 通常要实现 MessageListener 接口，该接口定义了 onMessage()方法。
- 当容器检测到 bean 守候的管道有消息到达时，容器调用 onMessage()方法，将消息作为参数传入 MDB。
- onMessage()中决定如何处理该消息：
  - ✓ 你可以使用注解指定 MDB 监听哪一个目标地址(Destination)。当 MDB 部署时，容器将读取其中的配置信息。

- MDB 编程案例

本章小结

- 消息中间件（Message Oriented Middleware）是在分布式系统中完成消息的发送和接收的基础软件。
- 本章介绍了：
  - ✓ 消息中间件的架构和要素
  - ✓ JAVA消息中间件JMS
  - ✓ 消息驱动Bean （MDB）