

ActiveMQ 入门

在 RabbitMQ 的章节中，我们已经介绍了消息队列的微服务场景，消息队列作为微服务应用之间的联系方式，在微服务中占有重要的地位。目前流行的消息队列产品很多，这节我们主要讲解 ActiveMQ。

ActiveMQ 是 Apache 出品，最流行的，功能强大的开源消息总线。ActiveMQ 是一个完全支持 JMS1.1 和 J2EE 1.4 规范的 JMS Provider 实现。目前，也是华为公有云提供的完全托管的高性能分布式消息服务（DMS）的一种。

ActiveMQ 的内核是 Java 编写的，也就是说服务端需要 Java 运行环境启动 ActiveMQ。

ActiveMQ 启动时，启动脚本使用 wrapper 包装器来启动 JVM。启动目录的“wrapper.conf”文件中配置了 JVM 相关信息，如 JVM 的初始内存大小和最大内存大小等，我们可以根据需要进行修改。

ActiveMQ 基本概念

JMS（Java Messaging Service）是 Java 平台上有关面向消息中间件(MOM)的技术规范，旨在通过提供标准的产生、发送、接收和处理消息的 API 简化企业应用的开发，类似于 JDBC 和关系型数据库通信方式的抽象。

MOM（Message-oriented middleware），是用于以分布式应用或系统中的异步、松耦合、可靠、可扩展和安全通信的一类软件。MOM 的总体思想是它作为消息发送器和消息接收器之间的消息中介，这种中介提供了一个全新水平的松耦合。

ActiveMQ 持久化存储方案

当 ActiveMQ 接收到 PERSISTENT Message 消息后就需要借助持久化方案来完成消息存储。可以通过多种介质完成存储：磁盘文件系统、ActiveMQ 内置数据库或第三方关系型数据库。

1. KahaDB

在 ActiveMQ V5.4 版本及后续版本 KahaDB 都作为 ActiveMQ 的默认持久化存储方案。

Apache ActiveMQ 官方表示它用来替换之前的 AMQ Message Store 存储方案。

相比于原来的 AMQ 存储方式，官方认为 KahaDB 使用了更少的文件描述符，设计目标是支持事务日志、可靠、可扩展、速度快等。可扩展体现在 KahaDB 支持其他三种存储器的外接扩展，也就是说可以同时用不止一种，这样可以取长补短，适合更广的应用场景，达到性能最佳。KahaDB 等存储类型，需要在 `conf/activemq.xml` 中进行配置。

如果安装了 ActiveMQ V5.4+ 的版本，即使不配置任何的 KahaDB 参数信息，ActiveMQ 也会启动 KahaDB。此时，KahaDB 文件所在位置是您的 ActiveMQ 安装路径下的 `/data/KahaDB` 子目录下。

2. AMQ

AMQ 与 KahaDB 存储相似，不依赖于第三方数据库，用户能够快速启动和运行。当消息吞吐量是应用程序的主要需求时，该存储方案是最佳选择，在此方案下消息以日志的形式实现持久化，存放在 Data Log 里。

对于每个 destination 都要建一个索引，所以不适于很多 destination 并发的场合，而这恰恰是 KahaDB 的优势（KahaDB 一个索引文件存储所有的 destination），它可以支持最大 10000 个 queue 的同时等待。

3. JDBC

从 ActiveMQ V4 版本开始，ActiveMQ 就支持使用关系型数据库进行持久化存储——通过 JDBC 实现的数据库连接。可以支持的关系型数据库包括：Apache Derby、DB2、HSQL、

Informix、MySQL、Oracle、Postgresql、SQLServer、Sybase 等。

选择 JDBC 方案，通常是因为这些团队对关系型数据库有更丰富的使用经验，且有专门的数据库管理人员，但是它在性能上不比上述消息存储方案要好。事实是，许多企业使用关系数据库作为存储，是因为他们更愿意充分利用这些数据库资源，比如已有的热备和负载方案。

4. 内存存储

内存消息存储器将所有持久消息保存在内存中。在仅存储有限数量 Message 的情况下，内存消息存储会很有用，因为 Message 通常会被快速消耗。一般用于实时消息的缓存，只针对非持久订阅的消费者提供了 5 种订阅恢复策略，可以极大程度增强非持久订阅的可用性。