聊天室架构说明

本项目聊天室功能使用RabbitMQ中间件进行实现。

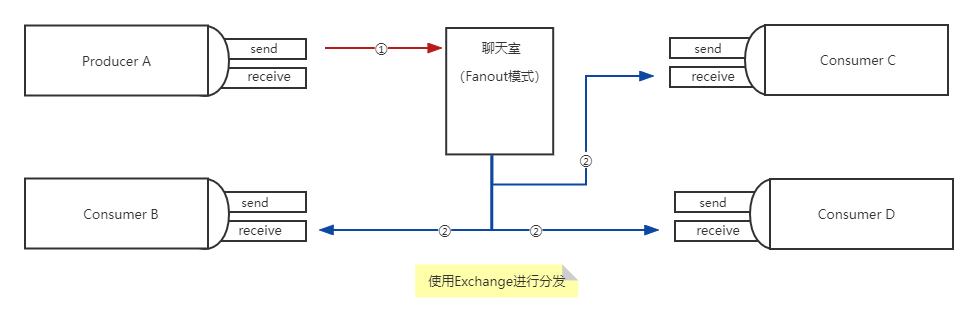
RabbitMQ中间件具有以下几个功能特性：

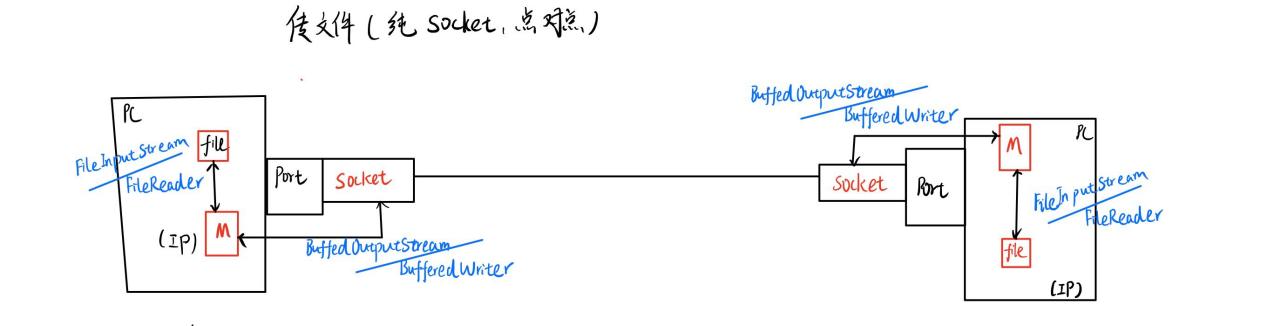
1. RabbitMQ中间件支持延迟消息，在生产者将消息发送到MQ服务端时，可以选择延迟一段时间后再投放给消费者。
2. RabbitMQ可以为消息或者队列设置属性TTL，即所有消息或队列的最大存活时间，如果这条消息在TTL时间内没有被消费则变成死信，进而配合死信队列实现任务的延迟。
3. RabbitMQ发送消息具有一定的可靠性，其表现在：1、RabbitMQ中具有Exchange（交换机）的概念，AMQP协议中，生产者将信息发送到交换机，交换机按照特定的策略转发给Queue进行存储，进而保证了生产者消息发送的可靠性。2、RabbitMQ为消费者提供了消息确认机制，采用消息确认机制后，只要令noAck=false，消费者就有足够的时间处理消息(任务)，不用担心处理消息过程中消费者进程挂掉后消息丢失的问题，因为RabbitMQ会一直持有消息直到消费者显式调用basicAck为止，至此，中间件保证了消费者接受的可靠性。
4. RabbitMQ保证了消息的幂等性。由于RabbitMQ存在重试机制，它可以避免由于网络原因或客端户延迟而造成的重复消费。
5. RabbitMQ可以实现远程RPC的调用。在RabbitMQ中，每条消息都具有correlationId的属性，进而实现关联response和request，在我们进行消息回复的时候，可以根据correlationId找到该回复对应的请求消息。
6. RabbitMQ具有权限系统，该权限系统下存在五种角色：超级管理员（administrator）、监控者（monitor）、决策制定者（policymaker）、普通管理者（management）和其他（none）。权限系统保证了信息队列的安全性。
7. RabbitMQ可以搭建集群。​ 当单台 RabbitMQ 服务器的处理消息的能力达到瓶颈时，此时可以通过 RabbitMQ 集群来进行扩展，从而达到提升吞吐量的目的。

聊天室使用RabbitMQ中间件实现聊天需要以下几个步骤：

1. 创建群聊：此时中间件创建了交换机，并设定进入广播模式，在广播模式下，交换机会把收到的信息发送给所有绑定该交换机的队列，并忽略routingKey，进而实现聊天室的每个成员都可以收到同一条消息。
2. 拉人：此时中间件加入队列绑定，每个聊天室使用者的消息队列会被放入监听队列当中。
3. 交流：此时各个消费者发送消息，由于RabbitMQ本身并不会对异常消息进行处理，因此消费者确认采用manual模式进行异常处理，除非人工进行调用，否则异常会一直堵塞，从而保证了消息的可靠性。

当A发言时，A发送的消息会发送到广播模式下的exchange（交换机），交换机会分发给每一个消费者的队列当中，示意图如下：





文件传输只使用socket实现点对点上传。一端读出file到内存中的数组，socket从内存获取信息，传输到另一端；另一端从socket读到内存，生成file，写入。

下一次迭代：

使用MQ完成对文件的点对点，乃至群发传输。