

消息如何保证100%投递成功

什么是生产端可靠性投递？

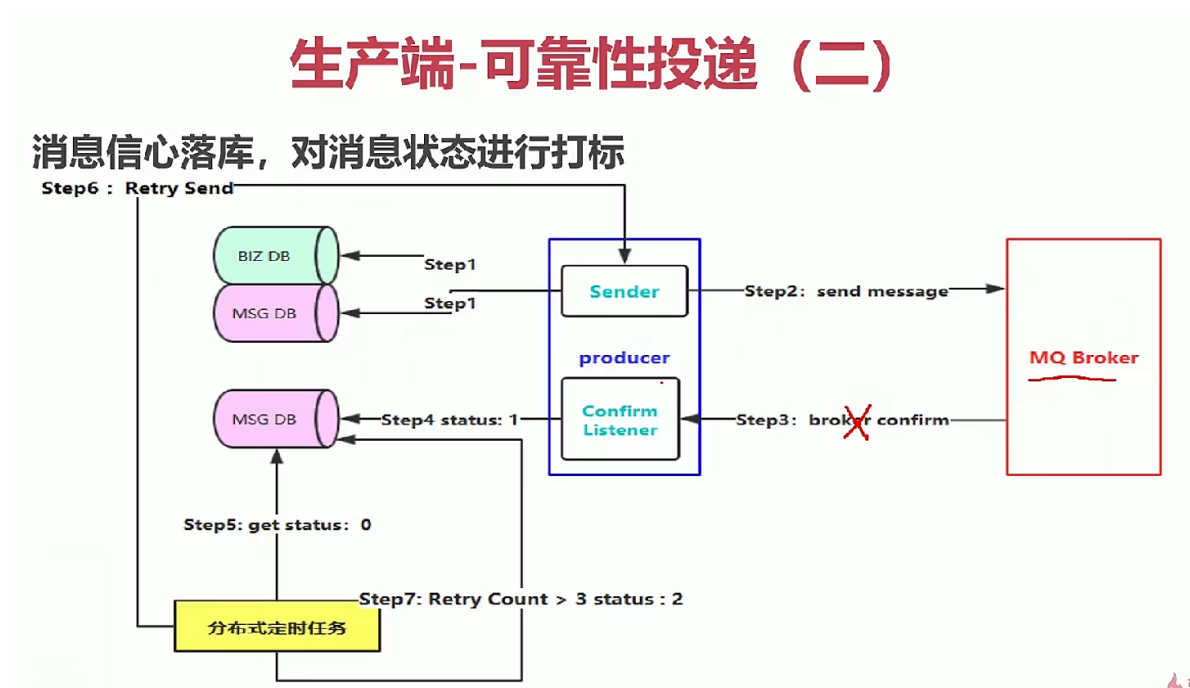
- 保障消息的成功发出
- 保障 MQ 节点成功接收
- 生产者收到 MQ 节点（Broker）确认应答
- 消息补偿机制

互联网大厂解决方案

1. 消息先入库，对消息的状态进行标记。
2. 消息的延迟投递，做二次确认，回调检查。

方案一

消息先入库，对消息的状态进行标记。



1. 消息先入库，对消息的状态进行标记。例如：status
2. 向 MQ 发送消息。
3. MQ 将应答发送给生产端，生产端要监听 MQ 的应答。
4. 如果收到了确认的应答。那么修改消息的状态为发送成功。
5. 如果没有收到 MQ 应答，需要有一个定时任务捞取 status 状态的消息再次发送。这个时候消息需要有最大的重试次数的限制。

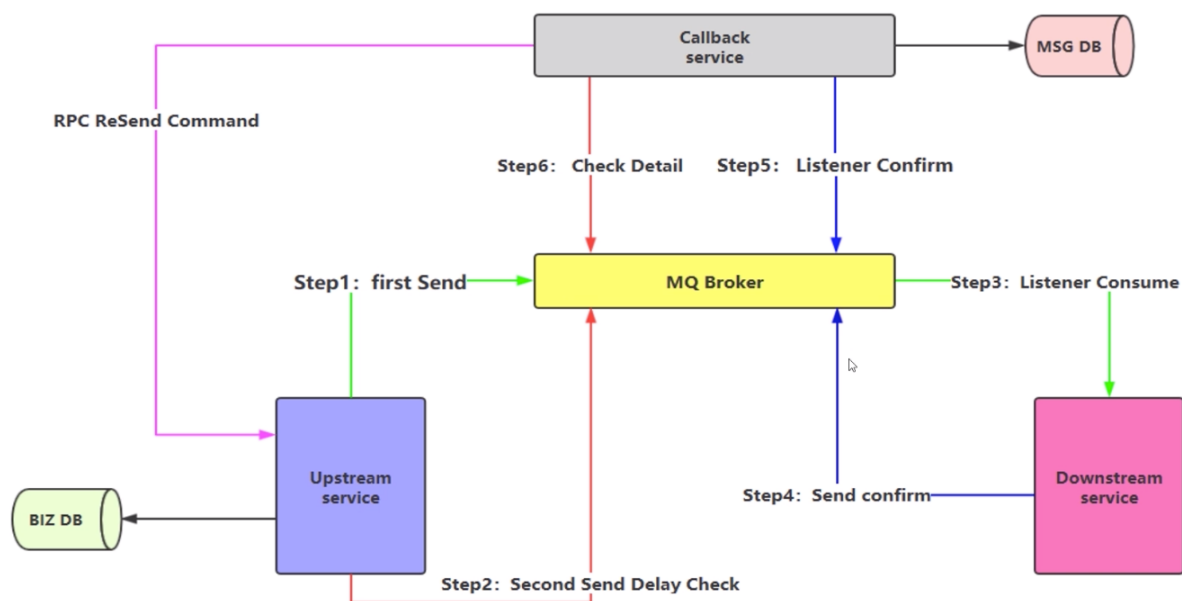
方案二

适用场景：适合在高并发场景下使用，

消息的延迟投递，做二次确认，回调检查。

目的：核心业务上减少数据库操作。减少消息的持久化，减少核心业务功能持久化时间，提高效率，提高性能

（最关键的点不是100%投递成功，最关键的点就是一定要保证性能，保证能看住这么大的订单量，能节省数据库操作就节省，可以异步的去补偿，例如：如果把订单数据和消息都直接保存在数据库，可能一台服务器只能抗住1000单（单点的），那个如果只保存订单数据，那么你的性能可能抗住2000单，这样的话你给节省一台服务器了）



1. 先把消息发送给 MQ 的队列 A。
2. 延迟一定的时间第二次发送一样的消息给 MQ 队列 B。B 是 A 的备份。
3. 有一个单独的服务（callback service）监听 MQ 队列 A 的响应。并且接受队列 B 的消息。
4. 如果 callback service 收到了队列 A 的消息确认，那么入库标记。
5. 延迟投递的消息开始发送到 B 队列。
6. callback service 收到队列 B 中的数据，去数据库里查询这个消息的状态
7. 如果状态不是成功则 callback service 则通知生产者，该消息没有收到响应。
8. 生产者如果收到 callback service 的消息。那么就走计数，补偿，告警等步骤。