

码农求职小助手：消息队列

笔记本： 26-消息队列

创建时间： 2019/9/16 11:02

更新时间： 2019/9/16 11:03

作者： pc941206@163.com

更多资料请关注微信公众号：码农求职小助手



消息队列基础知识

1、消息队列的主要作用？

解耦、异步、削峰

2、消息队列的优缺点？

优点：就是在特殊场景下有其对应的好处，解耦、异步、削峰。

缺点有以下几个：

1、**系统可用性降低**：系统引入的外部依赖越多，越容易挂掉。本来你就是 A 系统调用 BCD 三个系统的接口就好了，人 ABCD 四个系统好好的，没啥问题，你偏加个 MQ 进来，万一 MQ 挂了咋整，MQ 一挂，整套系统崩溃的，你不就完了？

2、**系统复杂度提高**：硬生生加个 MQ 进来，你怎么保证消息没有重复消费？怎么处理消息丢失的情况？怎么保证消息传递的顺序性？

3、**一致性问题**：A 系统处理完了直接返回成功了，人都以为你这个请求就成功了；但是问题是，要是 BCD 三个系统那里，BD 两个系统写库成功了，结果 C 系统写库失败了，咋

整？你这数据就不一致了。

3、Kafka、ActiveMQ、RabbitMQ、RocketMQ 有什么优缺点？

特性	ActiveMQ	RabbitMQ	RocketMQ	Kafka
单机吞吐量	万级，比 RocketMQ、Kafka 低一个数量级	同 ActiveMQ	10 万级，支撑高吞吐	10 万级，高吞吐，一般配合大数据类的系统来进行实时数据计算、日志采集等场景
topic 数量对吞吐量的影响			topic 可以达到几百/几千的级别，吞吐量会有较小幅度的下降，这是 RocketMQ 的一大优势，在同等机器下，可以支撑大量的 topic	topic 从几十到几百个时候，吞吐量会大幅度下降，在同等机器下，Kafka 尽量保证 topic 数量不要过多，如果要支撑大规模的 topic，需要增加更多的机器资源
时效性	ms 级	微秒级，这是 RabbitMQ 的一大特点，延迟最低	ms 级	延迟在 ms 级以内
可用性	高，基于主从架构实现高可用	同 ActiveMQ	非常高，分布式架构	非常高，分布式，一个数据多个副本，少数机器宕机，不会丢失数据，不会导致不可用
消息可靠性	有较低的概率丢失数据	基本不丢	经过参数优化配置，可以做到 0 丢失	同 RocketMQ
功能支持	MQ 领域的功能极其完备	基于 erlang 开发，并发能力很强，性能极好，延时很低	MQ 功能较为完善，还是分布式的，扩展性好	功能较为简单，主要支持简单的 MQ 功能，在大数据领域的实时计算以及日志采集被大规模使用

4、如何保证消息队列的高可用？

<https://github.com/doocs/advanced-java/blob/master/docs/high-concurrency/how-to-ensure-high-availability-of-message-queues.md>

5、如何保证消息不被重复消费？或者说，如何保证消息消费的幂等性？

<https://github.com/doocs/advanced-java/blob/master/docs/high-concurrency/how-to-ensure-that-messages-are-not-repeatedly-consumed.md>

6、如何保证消息的可靠性传输？或者说，如何处理消息丢失的问题？

<https://github.com/doocs/advanced-java/blob/master/docs/high-concurrency/how-to-ensure-the-reliable-transmission-of-messages.md>

7、如何保证消息的顺序性？

<https://github.com/doocs/advanced-java/blob/master/docs/high-concurrency/how-to-ensure-the-order-of-messages.md>

8、如何解决消息队列的延时以及过期失效问题？消息队列满了以后该怎么处理？有几百万消息持续积压几小时，说说怎么解决？

<https://github.com/doocs/advanced-java/blob/master/docs/high-concurrency/mq-time-delay-and-expired-failure.md>

9、如果让你写一个消息队列，该如何进行架构设计？说一下你的思路。

<https://github.com/doocs/advanced-java/blob/master/docs/high-concurrency/mq-design.md>

一、RabbitMQ

1、RabbitMQ 的使用场景有哪些？

1、**跨系统的异步通信**，所有需要异步交互的地方都可以使用消息队列。就像我们除了打电话（同步）以外，还需要发短信，发电子邮件（异步）的通讯方式；

2、**多个应用之间的耦合**，由于消息是平台无关和语言无关的，而且语义上也不再是函数调用，因此更适合作为多个应用之间的松耦合的接口。基于消息队列的耦合，不需要发送方和接收方同时在线。在企业应用集成（EAI）中，文件传输，共享数据库，消息队列，远程过程调用都可以作为集成的方法；

3、**应用内的同步变异步**，比如订单处理，就可以由前端应用将订单信息放到队列，后端应用从队列里依次获得消息处理，高峰时的大量订单可以积压在队列里慢慢处理掉。由于

同步通常意味着阻塞，而大量线程的阻塞会降低计算机的性能；

4、**消息驱动的架构**（EDA），系统分解为消息队列，和消息制造者和消息消费者，一个处理流程可以根据需要拆成多个阶段（Stage），各阶段之间用队列连接起来，前一个阶段处理的结果放入队列，后一个阶段从队列中获取消息继续处理；

5、应用需要更灵活的耦合方式，如发布订阅，比如可以指定路由规则；

6、跨局域网，甚至跨城市的通讯（CDN 行业），比如北京机房与广州机房的应用程序的通信。

2、RabbitMQ 有哪些重要的角色？

RabbitMQ 中重要的角色有：**生产者、消费者 和 代理**：

1、生产者：消息的创建者，负责创建和推送数据到消息服务器；

2、消费者：消息的接收方，用于处理数据和确认消息；

3、代理：就是 RabbitMQ 本身，用于扮演“快递”的角色，本身不生产消息，只是扮演“快递”的角色。

3、RabbitMQ 有哪些重要的组件？

1、ConnectionFactory（连接管理器）：应用程序与 rabbit 之间建立连接的管理器，程序代码中使用；

2、Channel（信道）：消息推送使用的通道；

3、Exchange（交换器）：用于接受、分配消息；

4、Queue（队列）：用于存储生产者的消息；

5、RoutingKey（路由键）：用于把生成者的数据分配到交换器上；

6、BindingKey（绑定键）：用于把交换器的消息绑定到队列上。

4、RabbitMQ 中 vhost 的作用是什么？

vhost 可以理解为虚拟 broker，即 mini-RabbitMQ server。其内部均含有独立的 queue、exchange 和 binding 等，但最重要的是，其拥有独立的权限系统，可以做到 vhost 范围的用户控制。当然，从 RabbitMQ 的全局角度，vhost 可以作为不同权限隔离的手段（一个典型的例子就是不同的应用可以跑在不同的 vhost 中）。

5、RabbitMQ 的消息是怎么发送的？

首先客户端必须连接到 RabbitMQ 服务器才能发布和消费消息，客户端和 rabbit server 之间会创建一个 tcp 连接，一旦 tcp 打开并通过了认证（认证就是你发送给 rabbit 服务器的用户名和密码），你的客户端和 RabbitMQ 就创建了一条 **amqp 信道**（channel），信道是创建在“真实” tcp 上的虚拟连接，amqp 命令都是通过信道发送出去的，每个信道都会有一个唯一的 id，不论是发布消息，订阅队列都是通过这个信道完成的。

6、RabbitMQ 怎么保证消息的稳定性？

提供了事务的功能。通过将 channel 设置为 confirm（确认）模式。

7、RabbitMQ 怎么避免消息丢失？

- 1、消息持久化；
- 2、ACK 确认机制；
- 3、设置集群镜像模式；
- 4、消息补偿机制。

8、要保证消息持久化成功的条件有哪些？

- 1、声明队列必须设置持久化 durable 设置为 true；
- 2、消息推送投递模式必须设置持久化，deliveryMode 设置为 2（持久）；
- 3、消息已经到达持久化交换器；
- 4、消息已经到达持久化队列。

以上四个条件都满足才能保证消息持久化成功。

9、RabbitMQ 持久化有什么缺点？

持久化的缺点就是降低了服务器的吞吐量，因为使用的是磁盘而非内存存储，从而降低了吞吐量。可尽量使用 ssd 硬盘来缓解吞吐量的问题。

10、RabbitMQ 有几种广播类型？

- 三种广播模式：

- 1、**fanout**：所有 bind 到此 exchange 的 queue 都可以接收消息（纯广播，绑定到 RabbitMQ 的接受者都能收到消息）；
- 2、**direct**：通过 routingKey 和 exchange 决定的那个唯一的 queue 可以接收消息；

3、**topic**：所有符合 routingKey (此时可以是一个表达式)的 routingKey 所 bind 的 queue 可以接收消息；

11、RabbitMQ 怎么实现延迟消息队列？

- 1、通过消息过期后进入死信交换器，再由交换器转发到延迟消费队列，实现延迟功能；
- 2、使用 RabbitMQ-delayed-message-exchange 插件实现延迟功能。

12、RabbitMQ 集群有什么用？

集群主要有以下两个用途：

- 1、高可用：某个服务器出现问题，整个 RabbitMQ 还可以继续使用；
- 2、高容量：集群可以承载更多的消息量。

13、RabbitMQ 节点的类型有哪些？

- 1、**磁盘节点**：消息会存储到磁盘；
- 2、**内存节点**：消息都存储在内存中，重启服务器消息丢失，性能高于磁盘类型。

14、RabbitMQ 集群搭建需要注意哪些问题？

- 1、各节点之间使用 “--link” 连接，此属性不能忽略；
- 2、各节点使用的 erlang cookie 值必须相同，此值相当于“密钥”的功能，用于各节点的认证；
- 3、整个集群中必须包含一个磁盘节点。

15、RabbitMQ 每个节点是其他节点的完整拷贝吗？为什么？

不是，原因有以下两个：

- 1、**存储空间的考虑**：如果每个节点都拥有所有队列的完全拷贝，这样新增节点不但没有新增存储空间，反而增加了更多的冗余数据；
- 2、**性能的考虑**：如果每条消息都需要完整拷贝到每一个集群节点，那新增节点并没有提升处理消息的能力，最多是保持和单节点相同的性能甚至是更糟。

16、RabbitMQ 集群中唯一的一个磁盘节点崩溃了会发生什么情况？

如果唯一磁盘的磁盘节点崩溃了，不能进行以下操作：

- 1、不能创建队列；
- 2、不能创建交换器；
- 3、不能创建绑定；
- 4、不能添加用户；
- 5、不能更改权限；
- 6、不能添加和删除集群节点

唯一磁盘节点崩溃了，集群是可以保持运行的，但你不能更改任何东西。

17、RabbitMQ 对集群节点停止顺序有要求吗？

RabbitMQ 对集群的停止的顺序是有要求的，应该先关闭内存节点，最后再关闭磁盘节点。如果顺序恰好相反的话，可能会造成消息的丢失。

二、Kafka

1、kafka 可以脱离 zookeeper 单独使用吗？为什么？

kafka 不能脱离 zookeeper 单独使用，因为 kafka 使用 zookeeper 管理和协调 kafka 的节点服务器。

2、kafka 有几种数据保留的策略？

kafka 有两种数据保存策略：**按照过期时间保留** 和 **按照存储的消息大小保留**。

3、kafka 同时设置了 7 天和 10G 清除数据，到第五天的时候消息达到了 10G，这个时候 kafka 将如何处理？

这个时候 kafka 会执行数据清除工作，时间和大小不论那个满足条件，都会清空数据。

4、什么情况会导致 kafka 运行变慢？

- 1、cpu 性能瓶颈；
- 2、磁盘读写瓶颈；
- 3、网络瓶颈。

5、使用 kafka 集群需要注意什么？

集群的数量不是越多越好，最好不要超过 7 个，因为节点越多，消息复制需要的时间就越长，整个群组的吞吐量就越低。集群数量最好是单数，因为超过一半故障集群就不能用了，设置为单数容错率更高。

三、消息队列常见问题及其解决方案

- 1、https://blog.csdn.net/qq_36236890/article/details/81174504
- 2、https://blog.csdn.net/A_BlackMoon/article/details/85197785