## 消息队列面试题汇总

#### 1.消息队列的应用场景有哪些？

答：消息队列的应用场景如下。

* 应用解耦，比如，用户下单后，订单系统需要通知库存系统，假如库存系统无法访问，则订单减库存将失败，从而导致订单失败。订单系统与库存系统耦合，这个时候如果使用消息队列，可以返回给用户成功，先把消息持久化，等库存系统恢复后，就可以正常消费减去库存了。
* 削峰填谷，比如，秒杀活动，一般会因为流量过大，从而导致流量暴增，应用挂掉，这个时候加上消息队列，服务器接收到用户的请求后，首先写入消息队列，假如消息队列长度超过最大数量，则直接抛弃用户请求或跳转到错误页面。
* 日志系统，比如，客户端负责将日志采集，然后定时写入消息队列，消息队列再统一将日志数据存储和转发。

#### 2.RabbitMQ 有哪些优点？

答：RabbitMQ 的优点如下：

* 可靠性，RabbitMQ 的持久化支持，保证了消息的稳定性；
* 高并发，RabbitMQ 使用了 Erlang 开发语言，Erlang 是为电话交换机开发的语言，天生自带高并发光环和高可用特性；
* 集群部署简单，正是因为 Erlang 使得 RabbitMQ 集群部署变的非常简单；
* 社区活跃度高，因为 RabbitMQ 应用比较广泛，所以社区的活跃度也很高；
* 解决问题成本低，因为资料比较多，所以解决问题的成本也很低；
* 支持多种语言，主流的编程语言都支持，如 Java、.NET、PHP、Python、JavaScript、Ruby、Go 等；
* 插件多方便使用，如网页控制台消息管理插件、消息延迟插件等。

#### 3.RabbitMQ 有哪些重要的角色？

答：RabbitMQ 包含以下三个重要的角色：

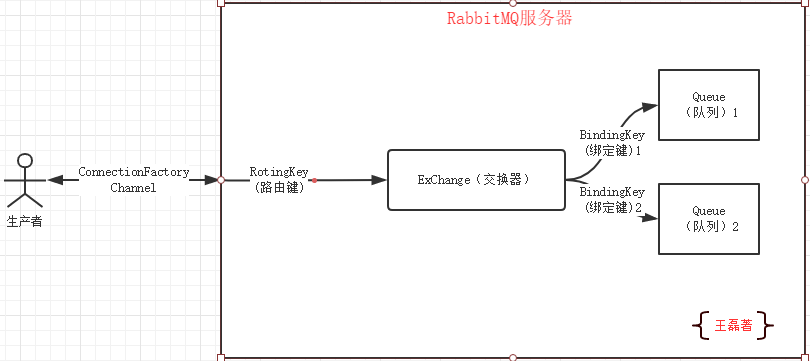
* 生产者：消息的创建者，负责创建和推送数据到消息服务器；
* 消费者：消息的接收方，用于处理数据和确认消息；
* 代理：就是 RabbitMQ 本身，用于扮演“快递”的角色，本身不生产消息，只是扮演“快递”的角色。

#### 4.RabbitMQ 有哪些重要的组件？它们有什么作用？

答：RabbitMQ 包含的重要组件有：ConnectionFactory（连接管理器）、Channel（信道）、Exchange（交换器）、Queue（队列）、RoutingKey（路由键）、BindingKey（绑定键） 等重要的组件，它们的作用如下：

* ConnectionFactory（连接管理器）：应用程序与 RabbitMQ 之间建立连接的管理器，程序代码中使用；
* Channel（信道）：消息推送使用的通道；
* Exchange（交换器）：用于接受、分配消息；
* Queue（队列）：用于存储生产者的消息；
* RoutingKey（路由键）：用于把生成者的数据分配到交换器上；
* BindingKey（绑定键）：用于把交换器的消息绑定到队列上。

运行流程，如下图所示：



1

#### 5.什么是消息持久化？

答：消息持久化是把消息保存到物理介质上，以防止消息的丢失。

#### 6.RabbitMQ 要实现消息持久化，需要满足哪些条件？

答：RabbitMQ 要实现消息持久化，必须满足以下 4 个条件：

* 投递消息的时候 durable 设置为 true，消息持久化，代码：channel.queueDeclare(x, true, false, false, null)，参数 2 设置为 true 持久化；
* 设置投递模式 deliveryMode 设置为 2（持久），代码：channel.basicPublish(x, x, MessageProperties.PERSISTENT *TEXT* PLAIN,x)，参数 3 设置为存储纯文本到磁盘；
* 消息已经到达持久化交换器上；
* 消息已经到达持久化的队列。

#### 7.消息持久化有哪些缺点？如何缓解？

答：消息持久化的缺点是很消耗性能，因为要写入硬盘要比写入内存性能较低很多，从而降低了服务器的吞吐量。可使用固态硬盘来提高读写速度，以达到缓解消息持久化的缺点。

#### 8.如何使用 Java 代码连接 RabbitMQ？

答：使用 Java 代码连接 RabbitMQ 有以下两种方式：

方式一：

public static Connection GetRabbitConnection() {  
 ConnectionFactory factory = new ConnectionFactory();  
 factory.setUsername(Config.UserName);  
 factory.setPassword(Config.Password);  
 factory.setVirtualHost(Config.VHost);  
 factory.setHost(Config.Host);  
 factory.setPort(Config.Port);  
 Connection conn = null;  
 try {  
 conn = factory.newConnection();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return conn;  
}

方式二：

public static Connection GetRabbitConnection2() {  
 ConnectionFactory factory = new ConnectionFactory();  
 // 连接格式：amqp://userName:password@hostName:portNumber/virtualHost  
 String uri = String.format("amqp://%s:%s@%s:%d%s", Config.UserName, Config.Password, Config.Host, Config.Port,  
 Config.VHost);  
 Connection conn = null;  
 try {  
 factory.setUri(uri);  
 factory.setVirtualHost(Config.VHost);  
 conn = factory.newConnection();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return conn;  
}

#### 9.使用 Java 代码编写一个 RabbitMQ 消费和生产的示例？

答：代码如下：

public static void main(String[] args) {  
 publisher(); // 生产消息  
 consumer(); // 消费消息  
}  
  
/\*\*  
 \* 推送消息  
 \*/  
public static void publisher() {  
 // 创建一个连接  
 Connection conn = ConnectionFactoryUtil.GetRabbitConnection();  
 if (conn != null) {  
 try {  
 // 创建通道  
 Channel channel = conn.createChannel();  
 // 声明队列【参数说明：参数一：队列名称，参数二：是否持久化；参数三：是否独占模式；参数四：消费者断开连接时是否删除队列；参数五：消息其他参数】  
 channel.queueDeclare(Config.QueueName, false, false, false, null);  
 String content = String.format("当前时间：%s", new Date().getTime());  
 // 发送内容【参数说明：参数一：交换机名称；参数二：队列名称，参数三：消息的其他属性-routing headers，此属性为MessageProperties.PERSISTENT\_TEXT\_PLAIN用于设置纯文本消息存储到硬盘；参数四：消息主体】  
 channel.basicPublish("", Config.QueueName, null, content.getBytes("UTF-8"));  
 System.out.println("已发送消息：" + content);  
 // 关闭连接  
 channel.close();  
 conn.close();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
}  
  
/\*\*  
 \* 消费消息  
 \*/  
public static void consumer() {  
 // 创建一个连接  
 Connection conn = ConnectionFactoryUtil.GetRabbitConnection();  
 if (conn != null) {  
 try {  
 // 创建通道  
 Channel channel = conn.createChannel();  
 // 声明队列【参数说明：参数一：队列名称，参数二：是否持久化；参数三：是否独占模式；参数四：消费者断开连接时是否删除队列；参数五：消息其他参数】  
 channel.queueDeclare(Config.QueueName, false, false, false, null);  
  
 // 创建订阅器，并接受消息  
 channel.basicConsume(Config.QueueName, false, "", new DefaultConsumer(channel) {  
 @Override  
 public void handleDelivery(String consumerTag, Envelope envelope, AMQP.BasicProperties properties,  
 byte[] body) throws IOException {  
 String routingKey = envelope.getRoutingKey(); // 队列名称  
 String contentType = properties.getContentType(); // 内容类型  
 String content = new String(body, "utf-8"); // 消息正文  
 System.out.println("消息正文：" + content);  
 channel.basicAck(envelope.getDeliveryTag(), false); // 手动确认消息【参数说明：参数一：该消息的index；参数二：是否批量应答，true批量确认小于index的消息】  
 }  
 });  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
}

#### 10.RabbitMQ 交换器类型有哪些？

答：RabbitMQ 消费类型也就是交换器（Exchange）类型有以下四种：

* direct：轮询方式
* headers：轮询方式，允许使用 header 而非路由键匹配消息，性能差，几乎不用
* fanout：广播方式，发送给所有订阅者
* topic：匹配模式，允许使用正则表达式匹配消息

RabbitMQ 默认的是 direct 方式。

#### 11.RabbitMQ 如何确保每个消息能被消费？

答：RabbitMQ 使用 ack 消息确认的方式保证每个消息都能被消费，开发者可根据自己的实际业务，选择 channel.basicAck() 方法手动确认消息被消费。

#### 12.RabbitMQ 接收到消息之后必须消费吗？

答：RabbitMQ 接收到消息之后可以不消费，在消息确认消费之前，可以做以下两件事：

* 拒绝消息消费，使用 channel.basicReject(消息编号, true) 方法，消息会被分配给其他订阅者；
* 设置为死信队列，死信队列是用于专门存放被拒绝的消息队列。

#### 13.topic 模式下发布了一个路由键为“com.mq.rabbit.error”的消息，请问以下不能接收到消息的是？

A：cn.mq.rabbit.\*  
B：#.error  
C：cn.mq.\*  
D：cn.mq.#

答：C

题目解析：“\*”用于匹配一个分段（用“.”分割）的内容，“#”用于匹配 0 和多个字符。

#### 14.以下可以获取历史消息的是？

A：topic 交换器  
B：fanout 交换器  
C：direct 交换器  
D：以上都不是

答：C

题目解析：fanout 和 topic 都是广播形式的，因此无法获取历史消息，而 direct 可以。

#### 15.RabbitMQ 包含事务功能吗？如何使用？

答：RabbitMQ 包含事务功能，主要是对信道（Channel）的设置，主要方法有以下三个：

* channel.txSelect() 声明启动事务模式；
* channel.txComment() 提交事务；
* channel.txRollback() 回滚事务。

#### 16.RabbitMQ 的事务在什么情况下是无效的？

答：RabbitMQ 的事务在 autoAck=true 也就是自动消费确认的时候，事务是无效的。因为如果是自动消费确认，RabbitMQ 会直接把消息从队列中移除，即使后面事务回滚也不能起到任何作用。

#### 17.Kafka 可以脱离 ZooKeeper 单独使用吗？

答：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。

#### 18.Kafka 有几种数据保留的策略？

答：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。

#### 19.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？

答：这个时候 Kafka 会执行数据清除工作，时间和大小不论哪个满足条件，都会清空数据。

#### 20.什么情况会导致 Kafka 运行变慢？

答：以下情况可导致 Kafka 运行变慢：

* CPU 性能瓶颈
* 磁盘读写瓶颈
* 网络瓶颈

#### 21.使用 Kafka 集群需要注意什么？

答：Kafka 集群使用需要注意以下事项：

* 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。

## 更多资源下载交流请加微信：Morstrong,加入永久会员,网盘更新更快捷！

# 本资源由微信公众号：光明顶一号，提供支持