**比较好的面试问题**

<https://www.nowcoder.com/discuss/198717?type=0&order=0&pos=22&page=1>

**头条后端一面**

1.自我介绍，介绍简历上的项目

2.问我为什么要转开发？

5.Java里synchronized是怎么实现的？synchronized是可重入锁吗，怎么实现？

6.JVM调优

7.线程与进程的区别，线程切换与进程切换？进程切换一定比线程切换开销大吗？（进程间线程切换）

8.进程通信，线程同步方法

9.操作系统内存是怎么管理的，虚拟内存是什么？

10.操作系统用户态和内核态

11.计算机网络模型有哪些，分别介绍

12.TCP四次挥手过程？第三次挥手客户端收到服务端发来的FIN后处于什么状态（time\_wait），为什么要time wait？

13.MySQL索引有哪些，基于数据结构的分类一下？基于hash和B+树的有什么区别、复杂度？什么时候要建立索引？

14.索引最左原则，模糊查询一定索引失效?

15.数据库的事务隔离级别? 脏读、不可重复读、幻读？

16.安全隔离级别设置为序列化，删除更新操作会产生幻读吗（问题记的有点模糊）

17.sql语句的执行过程，写完一条sql语句之后过程是怎样的。

18.怎么加快sql查询？sql优化方法？

vivo提前批Android面经

1.自我介绍，包括项目经历；

2.项目中遇到的问题，怎么解决的？

例如三级目录的排序问题。可以牵扯到如何设计数据库表

全局异常处理的优点

3.数据类型有哪些？

4.什么时候会自动装箱、自动拆箱？

5.接口和抽象类的区别？

6.线程和进程的区别？

7.线程有哪些状态？

8.创建线程有哪几种方法？

9.调用哪些方法实现线程阻塞？

10.wait和sleep方法的区别？

11.异常处理简单说一下？哪些类型异常必须放到finally处理？

16.手撕多线程代码：两个线程交替打印0-100；

# 并发

你用过Java的J.U.C并发包吧，给我讲一下AQS的原理

能讲下ConcurrentHashMap的实现原理么 JDK7或者8都行

CAS的原理给我讲一下

他是怎么保证内存的可见性的

CAS会产生什么问题

# 多线程

你平时是怎么创建线程的？

为什么用线程池，线程池有什么好处？

创建线程池需要的关键参数有哪些？

线程池有哪几种任务拒绝策略

# Redis

我看了一下你简历上写了很多redis字眼，那你知道redis的什么东西

Redis的key的写入和删除的原理

怎么保证Redis的高可用

# RabbitMQ

给你一道进阶题，rabbitMQ是怎么保证消息不丢的，从客户端—消息队列， 消息队列—服务器端的角度考虑

答：

<https://blog.csdn.net/qq_36236890/article/details/81174504>

**生产者到MQ**：

1. 可以选择用RabbitMQ提供的事务功能，就是生产者发送数据之前开启RabbitMQ事务（channel.txSelect），然后发送消息，如果消息没有成功被RabbitMQ接收到，此时就会抛出异常，捕获异常后就可以回滚事务（channel.txRollback），然后重试发送消息；如果正常发送了消息，那么可以提交事务（channel.txCommit）。但是问题是，RabbitMQ事务机制一搞，基本上吞吐量会下来，因为太耗性能。

2. 开启confirm模式，在生产者那里设置开启confirm模式之后，你每次写的消息都会分配一个唯一的id，然后如果写入了RabbitMQ中，RabbitMQ会给你回传一个ack消息，告诉你说这个消息ok了。如果RabbitMQ没能处理这个消息，会回调你一个nack接口，告诉你这个消息接收失败，你可以重试。而且你可以结合这个机制自己在内存里维护每个消息id的状态，如果超过一定时间还没接收到这个消息的回调，那么你可以重发。（有串行、批量、异步三种confirm方式）

总结：事务机制和cnofirm机制最大的不同在于，事务机制是同步的，你提交一个事务之后会阻塞在那儿，但是confirm机制是异步的，你发送个消息之后就可以发送下一个消息，然后那个消息RabbitMQ接收了之后会异步回调你一个接口通知你这个消息接收到了。

所以一般在生产者这块避免数据丢失，都是用confirm机制的。

**RabbitMQ弄丢了数据**

设置持久化有两个步骤，第一个是创建queue的时候将其设置为持久化的，这样就可以保证RabbitMQ持久化queue的元数据，但是不会持久化queue里的数据；第二个是发送消息的时候将消息的deliveryMode设置为2，就是将消息设置为持久化的，此时RabbitMQ就会将消息持久化到磁盘上去。必须要同时设置这两个持久化才行，RabbitMQ哪怕是挂了，再次重启，也会从磁盘上重启恢复queue，恢复这个queue里的数据。

而且持久化可以跟生产者那边的confirm机制配合起来，只有消息被持久化到磁盘之后，才会通知生产者ack了，所以哪怕是在持久化到磁盘之前，RabbitMQ挂了，数据丢了，生产者收不到ack，你也是可以自己重发的。

**消息队列—服务器：**

原因：为了保证RabbitMQ能够感知消费者正确取到了消息，RabbitMQ提供了消息确认机制，当队列发送一条消息给消费者时，会记录一个unack标志，当消费者拿到消息之后，会回复一个ack标志，从而抵消了原来的unack标志。默认情况是：当消费者拿到消息之后立即回复ack而不管消息是否正确被处理，就回复。这会出现的问题是，当消费者一拿到消息，消费者异常并与服务器断开，此时已经回复了ack标志，服务器会以为消费者已经处理完了这条消息，这时这条消息就丢失了。

解决方法：

关闭自动ack（basicConsume方法中的autoAck设置为false），在消费完消息后手动发送一个ack标志给RabbitMQ，RabbitMQ收到消息回执（Message acknowledgment）后才将该消息从Queue中移除。

如果RabbitMQ没有收到回执并检测到消费者的RabbitMQ连接断开，则RabbitMQ会将该消息重新放到队列中发送给其他消费者（如果存在多个消费者）进行处理。这里不存在timeout，一个消费者处理消息时间再长也不会导致该消息被发送给其他消费者，除非它的RabbitMQ连接断开。

# JVM

知道Java中的内存模型吧，它有8个指令你给我说一下

双亲委派模型本质是解决了什么问题？安全性

有哪几种类加载器？

# SQL

MySQL InnoDB存储引擎中的MVCC解决了什么问题，能说下MVCC的实现原理么

SQL语句经常写吧，那我给你出一道SQL题（分组求和排序）

你认为什么情况下不应该建立索引

# 其他

RPC你了解过吗？