1. 线程池的原理以及各种线程池的应用场景
2. 线程池中使用有限的阻塞队列和无限的阻塞队列的区别
3. 如果你发现你的sql语句始终走另一个索引，但是你希望它走你想要的索引，怎么办
4. mysql执行计划
5. 数据库索引为什么用b+树（磁盘io的深入了解）
6. \* 你在做sql优化主要从哪几个方面做，用到哪些方法工具
7. HashMap 和 ConcurrentHashMap
8. ArrayList 和 LinkedList
9. 接口与抽象类；
10. Exception 与 Error；
11. JVM 的内存结构；
12. Java垃圾回收（器）以及算法；
13. 悲观锁与乐观锁（Java 中）；
14. JVM 调优；
15. 数据库数据结构，索引；
16. \* 数据库隔离级别，锁协议，并发问题等；
17. InnoBD 多版本并发控制；
18. InnoBD 与 MyISAM；
19. Redis
20. 数据库读写分离、分库分表；
21. \* TCP 细节；
22. \* Spring IOC、AOP；
23. RabbitMQ
24. 都看些什么书，什么博客
25. 对技术有什么看法
26. \* 负载均衡，微服务等；
27. 网络方面，怎么影响性能，不同层的问题等；
28. HTTP 和 RPC 等；
29. IO 方面，BIO 和 NIO，内核态，零拷贝，Netty 等；
30. 数据库方面，读写分离，分库分表等，涉及到一个场景题；
31. Java 定位问题；
32. 学 Java 多久了，平时怎么学习，有什么爱好，对未来的规划之类的；
33. 项目介绍(项目中最大的技术挑战和技术难点)
34. Java object类的方法
35. \* Java Map类的子类以及分别的原理
	* concurrentHashMap怎么保证线程安全
	* LinkedHashMap
	* HashMap为什么线程不安全
36. Java线程间的通信
37. 使用的Java并发工具
38. \* MySQL的索引原理
39. MySQL的查询优化方法
40. MySQL的分库分表
41. redis的基本数据类型和使用场景
42. 消息队列的使用
43. 如何学习新技术
44. 代码编写：两个线程，一个线程打印奇数，一个打印偶数，控制交替打印奇偶数。(有多种方法实现，最好能比较优劣)
45. 项目介绍(技术难点和如何做技术选型)
46. \* Java注解的原理
47. spring boot如何启动
48. spring mvc的原理
49. \* 对docker的理解
50. git 常用的命令
51. 设计模式相关
52. 平时如何学习新技术以及最近再看的东西
53. 阅读过哪些源码
54. 询问职业规划
55. 算法题：剑指offer原题，丑数
56. 遇到的最大挑战以及如何解决
57. 大学最大的收获
58. 如何团队合作的
59. 职业规划
60. 项目介绍(项目背景、自己觉得的技术难点)
61. Java常见的集合类(包括区别、实现原理)
62. \* Java反射机制
63. Java原始类型\*的装箱
64. Java类加载机制(为什么双亲委派?)
65. Java常用的多线程工具
66. Java的锁
67. Java的io模型
68. MySQL存储引擎和索引原理和优化
69. **以下是Java**
70. JVM内存分配机制、内存模型
71. GC（minor GC、major GC、触发条件、GC算法等）
72. 类加载机制（类隔离和类共享）
73. Java集合（List、set、HashMap、HashTable、ConcurrentHashMap等）
74. \* socket编程及NIO
75. synchronize、final、static理解
76. 异常（检测异常和未检测异常、继承关系等）
77. **以下是数据库**
78. \* 第一、第二、第三范式之间的理解和比较
79. 数据库的事务、ACID及隔离级别
80. 索引优化（组合索引、最左匹配原则）、优缺
81. 手动写创建索引的语句
82. 并发访问场景和所有可能出现的结果、锁作用和实现
83. 主主复制、主从复制
84. B-tree的应用
85. int和varchar
86. io优化
87. 分表分库设计
88. **以下是操作系统/linux/集群**
89. shell脚本、查找文件命令
90. top命令、netstat命令、ifconfig和ipconfig
91. 乐观锁和悲观锁
92. session集群共享
93. Hadoop集群扩容问题
94. 集群负载均衡
95. 虚拟路由
96. Hadoop集群单点故障
97. **以下是计算机网络**
98. 传输层协议TCP和UDP
99. http和https（对称加密、非对称加密）
100. ftp和sftp
101. 从访问一个网址到页面出现，描述中间发生的所有事情。
102. **以下是数据结构/算法/设计模式**
103. \* Hash及冲突解决
104. 二叉搜索树
105. 手写快速排序算法、单链表反转、字符串部分逆序（如moc.anis.www转为[www.sina.com](http://www.sina.com)）
106. 手写二叉树层序遍历、二分查找、递归算法实现
107. 超大文件寻找top K算法设计（单机1M内存、Hadoop集群、外部排序+uniq命令）
108. 算法设计：给定函数a可以等概率产生1-5这5个数，仅调用a实现等概率产生1-7这7个数
109. 订单超大并发访问-队列批量处理
110. 观察者模式、工厂模式、适配器模式
111. 每个项目从整体架构设计、扩展性、后续优化到任何实现细节都有可能会问。对做过的事情需要多总结和优化思考。项目这块是需要花大量心思准备的。
112. 登陆鉴权服务是怎么做的（用户信息，利用https加密后明文变为秘文，直接存储）
113. session存什么信息
114. \* session和cookie的区别和关系
115. \* 线程和协程的区别
116. 联合索引，如何创建+联合索引，联合索引的数据结构
117. 分库分表
118. 分布式系统的设计
119. 4G内存，2T硬盘，有1T数据。这1T数据是一行一个字符串，字符串里面有字母数字符号；求出现频率最高的10个字符串。
120. 面向对象的原则
121. 单例模式
122. 为什么uniqueInstance声明使用volitie修饰
123. 你还会什么其他模式吗
124. 对称加密，非对称加密，https加密流程
125. 计算机网络的statusCode
126. volatile的底层如何实现，怎么就能保住可见性了？
127. 三个线程如何实现交替打印ABC
128. 线程池有哪些创建方式和安全性问题
129. 有哪些线程池的类型
130. 线程池中LinkedBlockingQueue满了的话，线程会怎么样
131. 线程池的底层原理和实现方法
132. 线程之间的交互方式有哪些？有没有线程交互的封装类 （join）
133. Java锁机制
134. \* 异常捕获处理
135. Spring MVC的原理和流程
136. HashMap和ConcurrentHashMap哪个效率更高？为什么？
137. Redis的缓存淘汰策略有哪些？
138. mybatis如何进行类型转换
139. mybatis的xml有什么标签
140. MySQL锁机制
141. 如何修改linux的文件权限
142. 你会怎么设计数据库表结构
143. 数据库有哪些索引？
144. 如何防止sql注入
145. myql间歇锁的实现原理
146. future的底层实现异步原理
147. SpringBoot Starter原理
148. rpc原理
149. 多个服务端上下线怎么感知
150. \* 缓存和数据一致性，怎么处理。流式计算
151. 多线程讲一下，FutureTask
152. Java和mysql的锁介绍，乐观锁和悲观锁
153. 分布式一致性讲一讲
154. 分布式锁的实现方式，zk实现和redis实现哪个比较好
155. 多点登陆怎么实现
156. 把乐观锁加在数据库上面，怎么实现
157. 降级处理hystrix了解过么
158. 两次点击，怎么防止重复下订单
159. JUC说说你知道的东西
160. B+树的叶子节点
161. full gc怎么触发
162. 高吞吐量的话用哪种gc算法
163. JDK8的stream的操作
164. volatile原理 线程池原理，拒绝策略，核心线程数
165. 1亿个手机号码，判断重复
166. 单元测试介绍一下，多模块依赖怎么单元测试。Mockito
167. 消息队列原理介绍
168. ClassLoader原理和应用
169. Java的新技术，以及技术最新进展
170. Java三大特性
171. 工厂模式
172. 集合类的结构：Iterator、Collection（List、Set、Queue）、Map
173. ArrayList和LinkedList的区别
174. HashMap底层结构
175. HashMap Put 的过程
176. HashMap 为什么长度是2的倍数：找索引时 key 的 hash 值与数组的长度值减 1 进行与运算，长度为 2 的倍数时能减少碰撞
177. JDK 1.7 和 1.8 中 HashMap 的区别：1.8 增加红黑树、头插变为尾插、扩容后元素位置要么在原位置，要么在原位置 + 扩容前旧容量
178. HashMap 为什么线程不安全：扩容时链表可能形成闭环
179. ConcurrentHashMap 怎么保证线程安全
180. ConcurrentHashMap 和 HashMap 区别
181. 线程池常用参数：核心线程数、最大线程数、阻塞队列、线程最大存活时间
182. 线程池工作流程
183. 线程池的类别和区别
184. 阻塞队列的类别和区别
185. **内存结构和各自存放的内容：**方法区、堆、虚拟机栈、本地方法栈、程序计数器
186. 新生代和老年代垃圾回收算法：新生代 -> 复制，老年代 -> 标记-整理（CMS用的标记-清除）
187. CMS/G1的流程：初始标记 -> 并发标记 -> 重新标记 -> 并发清除
188. CMS/G1的特点和区别：CMS：尽可能减少停顿时间，采用标记-清除，会造成内存碎片化；G1：可以预测停顿时间，采用标记-整理，减少内存碎片化， 将内存划分为 region 粒度
189. 判断垃圾是否可以回收的方法：引用计数法、可达性分析法
190. 哪些对象可以作为GC ROOT
191. Spring IOC 和 AOP 的含义和原理
192. Spring、SpringBoot、SpringMVC 的区别
193. SpringBoot 的启动流程
194. Bean 的生命周期
195. SpringMVC 的流程
196. 什么是事务
197. 事务的特性（ACID），详细说明各个特性的含义：原子性、一致性、隔离性、持久性
198. 事务隔离级别：未提交读 -> 已提交读 -> 可重复读 -> 串行化
199. 脏读、不可重复读、幻读的区别
200. 事务隔离级别怎么实现预防上述问题（eg: 可重复读怎么实现预防不可重复读）
201. 常用索引（B树、Hash）
202. 聚簇索引和非聚簇索引的区别
203. 联合索引和单个索引的区别，联合索引的最左匹配原则
204. **B树和B+树的区别：1.** B+树的非叶子节点不保存数据，所以磁盘页能容纳更多节点元素，更“矮胖”；2. B+树查询必须查找到叶子节点，B树只要匹配到即可不用管元素位置，因此B+树查找更稳定（也不慢）；3. 对于范围查找来说，B+树只需遍历叶子节点链表即可，B树却需要重复地中序遍历
205. Redis 的五种数据结构：String、Hash、List、Set、Zset（SortSet）
206. Redis 为什么快：纯内存、单线程、IO多路复用（一般会继续问IO多路复用相关问题）
207. \* Redis 持久化：RDB、AOF
208. Redis 高可用：哨兵机制、Redis Cluster、持久化（上面那个）
209. 点击一个URL到页面返回，发生了什么：客户端传给服务端时经过 TCP/IP 四层模型，中途依次加入HTTP报文、TCP报文、IP报文、MAC报文，然后逆序依次拆包
210. OSI 七层模型 & TCP/IP 四层模型
211. cookie 和 session 的区别
212. TCP 三次握手,四次挥手的流程（最好能画图，说明各阶段状态）
213. 为什么需要三次握手：防止已过期的连接再次传到被连接的主机
214. 为什么需要四次挥手 / 断开连接时为什么比建立连接多一次：服务端的 ACK 和 FIN 一般都会分开发送，ACK 表示自己收到了客户端的断开请求（但还有数据没传输完），FIN 表示自己传输完了数据
215. \* TCP 和 UDP 的区别
216. **TIME\_WAIT 的作用：1.** 保证客户端发送的最后一个ACK报文能够到达服务器 2. 防止类似与“三次握手”中提到了的“已经失效的连接请求报文段”出现在本连接中。客户端发送完最后一个确认报文后，在这个2MSL时间中，就可以使本连接持续的时间内所产生的所有报文段都从网络中消失
217. HTTPS的优势：CA证书认证、加密传输
218. HTTPS建立连接流程
219. 为什么传输过程中采用对称加密，协商时采用非对称加密：对称加密快
220. 进程和线程的区别
221. 进程间通信方式：管道、消息队列、信号量、共享内存
222. 线程间通信方式：消息队列、全局变量
223. Linux 查找某个进程（ps aux |grep threadId）
224. Linux 杀掉某个进程（kill，参数添加-9表示强制执行）
225. Linux 查看某个端口是否被占用（netstat -anp |grep threadId）
226. Linux 实时查看进程状态（top）
227. 数组中查找两数之和为K的序列对
228. 硬盘中有10G数据，内存1G，如何排序（多路归并排序）
229. 常见的单例写法
230. 新生代和老年代的区别
231. 大量大对象进入老年代会有什么问题（频繁触发full GC）
232. Jvm 有很多个大对象，应该如何改进
233. \* ThreadLocal
234. 讲一下servlet的生命周期
235. Mybatis的一二级缓存，#和$的区别
236. 给你N个已经排序好的长度为L的数组，将其排序成长度为N\*L的数组。
237. 数据库的锁机制。
238. 讲一下synchronized锁。（互斥锁、jdk1.5以后的偏向锁、轻量级锁、重量级锁）
239. 讲一下互斥锁的底层原理
240. 怎么查看消失的一个进程
241. Set和List的区别
242. 打印GC日志
243. 死锁的条件以及预防死锁。
244. ARP协议
245. 局域网，城域网，广域网
246. 说说mybatis
247. Java是值传递还是引用传递，为什么
248. \* new一个对象的过程
249. 青蛙跳台阶问题（剑指offer原题）
250. 传输层与网络层的区别
251. ping使用的协议，处于哪一层
252. last\_ack阶段
253. java互斥锁（synchronized和lock的区别）
254. Linux命令（find命令）
255. TCP传输过程中出现丢包怎么处理
256. 内存泄漏举例
257. 实现一个后端的异步服务框架（实现思路）
258. 迪杰斯特拉最短路径算法
259. 矩阵连乘问题（动态规划中的矩阵连乘问题）
260. ThreadLocal使用场景，原理
261. BFS和DFS的区别
262. 线程如何等待别的线程执行任务？
263. ping通反应很快，调用远程服务反应时间很长的原因（远程服务端阻塞，本地TCP粘包）
264. 高并发极小任务，如何选择线程池？换成长短不一的任务呢？（自定义任务队列）
265. 红黑树
266. 各种内部类 哪儿见过 怎么用
267. beanFactory 和 factoryBean
268. MySQL搜索引擎 InnoDB MyISAM
269. TCP粘包，保护消息边界和流？
270. 客户端和服务器最多能发送和接收多少TCP连接数？
271. 为什么数据库索引中用B+数而不用B树
272. B+树相比B树的优势？我说了查询性能更稳定，他说这样不是降低查询效率嘛，那为什么不用B树呢？
273. 什么时候索引会失效？
274. 怎么知道有没有加索引
275. B+树的叶子结点上存了哪些信息呢
276. 事务的底层原理是什么呢
277. MVCC详细说一下
278. 说一下并发编程的几个特性
279. volatile关键字的两层语义
280. \* 说一下四种引用状态
281. \* 介绍SpringBoot https://www.jianshu.com/p/24add3c5fedb
282. SpringBoot中可以借鉴的设计模式 https://www.cnblogs.com/xiufengchen/p/11609691.html
283. \* 大小端 https://www.jianshu.com/p/f9436f605299
284. 字符集与字符编码的区别 https://www.cnblogs.com/xdyixia/p/9114145.html
285. 介绍TCP/IP头，其中哪些首部涉及滑动窗口，以及拥塞控制 https://www.cnblogs.com/zlingh/p/6161088.html
286. 数据库的ACID以及怎么实现 https://blog.csdn.net/tb3039450/article/details/70168243
287. 怎么判断一个点是否在一个多边形的内部 https://www.cnblogs.com/anningwang/p/7581545.html
288. 栈与堆的不同 https://blog.csdn.net/K346K346/article/details/80849966
289. 事务ACID的实现 https://blog.csdn.net/tb3039450/article/details/70168243
290. synchronized 关键字底层原理 https://www.jianshu.com/p/570410236ff5
291. 什么是幂等性 https://blog.csdn.net/miachen520/article/details/91039661
292. 两种动态代理方式及其区别（代理模式） https://blog.csdn.net/lz1170063911/article/details/79835248
293. 高并发下如何删减库存 https://www.jianshu.com/p/9a04aaaf6e87
294. sso单点登录 https://www.jianshu.com/p/75edcc05acfd
295. \* servlet单实例多线程还是多实例多线程 https://www.cnblogs.com/horsen/p/6918640.html
296. 选举算法 https://www.jianshu.com/p/f0c717501cee
297. 用语言描述一颗二叉树 资料2
298. \* 解释了为什么MyISAM不支持行锁，而InnoDB支持 https://blog.csdn.net/silyvin/article/details/80140153
299. 深拷贝浅拷贝 https://blog.csdn.net/weixin\_41910848/article/details/82144671
300. 继承和组合的区别 https://blog.csdn.net/zymx14/article/details/79605926
301. 观察者模式 https://www.cnblogs.com/adamjwh/p/10913660.html
302. 自旋锁和阻塞锁的区别 https://www.cnblogs.com/1013wang/p/11806580.html
303. 关于公平锁和非公平锁 https://www.jianshu.com/p/d86faec4baa1
304. 栅栏和闭锁的区别 https://blog.csdn.net/x329357842/article/details/53239351
305. \* 类的编译过程 https://www.cnblogs.com/xing-yao/p/9322998.html
306. 僵尸进程 https://www.cnblogs.com/Anker/p/3271773.html
307. MySQL日志类型 https://blog.csdn.net/xiamiflying/article/details/80960598
308. Java终止线程的三种方式 https://www.cnblogs.com/liyutian/p/10196044.html
309. HTTP报文结构 <https://www.cnblogs.com/myseries/p/11239662.html>
310. Spring的事务
311. \* Spring的事件
312. \* Spring下的注解
313. Spring DI 的三种方式
314. Spring主要使用了什么模式
315. IOC，AOP的实现原理
316. SpringMvc 的控制器是不是单例模式，如果是，有什么问题，怎么解决
317. SpringMvc 中控制器的注解
318. @RequestMapping 注解用在类上的作用
319. 前台多个参数，这些参数都是一个对象，快速得到对象？
320. SpringMvc中函数的返回值？
321. SpringMvc和Ajax之间的相互调用？
322. \* Struts2 和 SpringMvc的区别?
323. \* 什么是Mybatis的接口绑定，有什么好处？
324. 什么情况用注解，什么情况用xml绑定？
325. Mybatis在核心处理类叫什么?
326. \* 查询表名和返回实体Bean对象不一致，如何处理？
327. \* Mybatis的好处？
328. \* Mybatis配置一对多，一对一？
329. 获取上一次自动生成的主键值？
330. \* Mybatis如何分页，分页原理？
331. Mybatis工作原理？
332. \* 非关系型数据库和关系型数据库区别，优势比较?
333. \* 什么是 内连接、外连接、交叉连接、笛卡尔积等?
334. varchar和char的使用场景?
335. SQL语言分类
336. like %和-的区别
337. count(\*)、count(1)、count(column)的区别
338. 什么是索引？
339. 索引的作用？它的优点缺点是什么？
340. 什么样的字段适合建索引
341. 嵌套事务
342. 查询语句不同元素（where、join、limit、group by、having等等）执行先后顺序
343. 主从复制的几种方式
344. 数据库主从复制分析的 7 个问题?
345. 数据库连接池怎么实现？https://www.cnblogs.com/zhuleixiao/p/8974982.html
346. \* Java web过滤器的生命周期？https://blog.csdn.net/hanziang1996/article/details/78967400
347. \* 慢查询https://blog.csdn.net/qq\_35571554/article/details/82800463
348. \* RPC简介及框架选择https://www.jianshu.com/p/b0343bfd216e
349. \* ThreadLocal 应用场景https://blog.csdn.net/liqingtx/article/details/80530633
350. 系统从单机到分布式到演变https://blog.csdn.net/arthuskingofc/article/details/104210191
351. \* 设计一个秒杀系统https://www.jianshu.com/p/d789ea15d060
352. \* 线程工厂https://www.cnblogs.com/bjlhx/p/7609100.html
353. 数据库水平切分，垂直切分的设计思路和切分顺序https://blog.csdn.net/qianfeng\_dashuju/article/details/93876986
354. 用Java反射机制实现对象的浅copy方法，即将srcObj对象的属性的值copy到对象destObj的同名属性中
355. 日志切分在运维中扮演着重要角色，现有1个500G的日志文件A，时间跨度为2019年1月25日到4月28日。怎么获取其中1月某天时段的记录生成文件B，注：操作服务器系统资源有限，请使用一条命令实现
356. Mysql的主从同步
357. 加索引需要注意哪些方面 https://www.cnblogs.com/suncan0/p/4767859.html
358. Redis源码
359. Redis和MySQL数据同步：<https://www.cnblogs.com/xiaozengzeng/p/10872290.html>
360. NIO是怎么从核心态到用户态完成IO的
361. Spring流程 https://www.jianshu.com/p/280c7e720d0c
362. Spring boot原理 https://baijiahao.baidu.com/s?id=1621999890737938965&wfr=spider&for=pc
363. JVM源码分析之安全点safepoint：https://www.jianshu.com/p/c79c5e02ebe6
364. SafePoint是什么 <https://www.jianshu.com/p/c79c5e02ebe6>
365. 几种常用的内存调试工具：<https://blog.csdn.net/imooc_000000/article/details/80305438>
366. 什么是Apache Kafka：<https://zhuanlan.zhihu.com/p/37405836>
367. Spring Boot中的监视器是什么：<https://www.cnblogs.com/programb/p/12430516.html>
368. 什么是JavaConfig：https://www.iteye.com/blog/xiaoxiaoher-2428037
369. Spring Boot 错误（异常）处理：<https://www.jianshu.com/p/b1681ce07663>
370. 如何监视所有spring boot微服务：<https://www.cnblogs.com/programb/p/12430620.html>
371. MySQL常用监控命令：<https://www.cnblogs.com/alan6/p/11589165.html>
372. Redis的全称 Remote Dictionary Server(远程数据服务)
373. redis集群之间是如何复制的：<https://blog.csdn.net/meism5/article/details/104258024>
374. Redis事务实现原理：<https://baijiahao.baidu.com/s?id=1620089432702209171&wfr=spider&for=pc>
375. Hello World 是如何运行的 <https://www.cnblogs.com/suncoolcat/p/3367632.html>
376. 字节码和二进制码的区别