封神总结：抖音、腾讯、阿里、美团社招服务端开发岗位硬核面试

原创

站在风口的java 于 2022-02-28 15:59:51 发布 65 收藏 1

分类专栏： 面试 文章标签： 面试 java 程序人生

前言

先报一下身份，笔者的师弟小民同学。小民同学应届校招从事服务端开发，参加校招面试以来，共面四个公司的服务端开发岗位：

- 抖音，IES互娱
- 腾讯，PCG 应用宝数据中台
- 阿里，新零售供应链
- 美团，到店事业部

目前为止，除了阿里还在等交叉，其他均已收到 offer，硬核技术实力。小民说自己是个技术渣，这是个要求很高（有点装 X）的同学。

本篇文章与大家分享一下面试经历，希望能对参与一线互联网大厂面试的同学有些帮助。

腾讯 PCG 应用宝数据中台

PCG 感觉疯狂招人，面试难度忽高忽低，面试形式也很多样，腾讯视频zoom牛客网都有，比较随意。

腾讯三面 20min 算法题+一个基础问题面试就结束了……一度以为自己直接挂了，体验很差。

准备的话操作系统计算机网络多看看。如果会 C++ 多准备下，面试腾讯问题不大。

腾讯一面

- 项目讲了20分钟，问你做了什么，项目细节、你的成长之类的
- Java锁 synchronized volatile
- 快排如何变为稳定的
- 排行榜如何实现
- 如何查排行榜第2000个人的信息
- java游戏服务器 如何通信
- 算法coding，反转链表

腾讯二面

- 工厂模式 场景
- hashmap treemap
- 数据库索引 不用二叉树原因是因为 二叉树可能会变为O(n)
- 数据库如何加快 查询 ：cache 索引 ， 分表分库，
- 网络粘包
- 网络两次握手不可以？
- Kafka水位(high watermark)
- ArrayList 和 LinkedList 遍历操作效率比较？ArrayList更快 操作系统预读

腾讯三面

- 算法：带父节点的二叉树 中序遍历查找的上一个节点？
  zookeeper如何体现AP

抖音，IES互娱

字节面试效率是最高的，一下午一面二面连着面试，之后隔两天就三面。每次面试HR都可以当天晚上给出面试结果，效率极高，给HR小姐姐送爱心～

面试准备，基础点到为止，只要不答得很差就没事。主要还是算法，基本上每轮两个算法题，白板写代码，要运行的那种，还是挺有压力。

抖音后台一面（50min）

线程池 堵塞队列为什么要用堵塞的

一个线程在内存中如何存储

volicate ->内存重排序到底怎么避免的……

http 握手的 wait time

访问一个 url 发生了什么

dns 是什么 原理

跳表 和平衡树区别

平衡树的种类

计算题：扑克牌两张王的概率

手写代码：合并N个链表 -> 优化为 log(n) -> null 判断 -> 不允许修改数据结构怎么实现

抖音后台二面（60min）

微服务注册中心原理

注册服务怎么判断上线下线

如果一个服务版本升级了 其他服务没升级怎么办

为什么要使用spring cloud

jvm参数 为什么要配置-> 8G内存的机器 java进程- 最大配置多少

策略模式 如何解藕 -> 项目中如何使用的

redis 持久化

redis 主从复制

算法题一

```
判断一个IP是否在国内。
输入：数据库中有几十万的国内IP段 (start_ip, end_ip)一个待验证的IP
输出：YES or NO
```

算法题二

```
用户在线波峰计算。
输入： 用户日志(time, user_id, login | logout)
输出：同时在线人数的峰值， 峰段(峰值的90%) eg (19:50到22:10， 峰值3亿，最低2.7亿)
```

抖音后台三面

没啥技术问题，领导约谈人生，职业规划、打算之类的。

阿里，新零售供应链

阿里面试难度最高了，基础每个知识点会问到你直到答不出来为止。时间也是最长的，基本上每次面试都 50min+，面试官问的问题很细节，发现你有含糊其辞的会详细问，基础一定要复习好，还要多看看源码。
面试流程还很长，一周一面，整个流程下来很累，笔者至今还在等交叉面。

阿里一面

- mysql B+ B区别
- mysql 隔离级别 -> MVCC如何保证的 -> 间隙锁怎么使用的
- mysql hash索引？
- redis 为什么快-> 系统设计的时候如何优化的
- jenkins 如何用的
- 策略模式 不同策略怎么转化的
- Spring AOP如何实现的 -> 你项目中如何捕获aop异常 以及记录日志的
- java 枚举类型是否可以继承 (final)？ 注解是否可以继承？
- java内存结构
- 对象创建过程
- 类加载器 -> 双亲委派-> ClassLoad find load的区别(和面向对象有关系)-> JDBC 加载机制 -> 面向对象的原则
- 滑动窗口 -> 客户端和服务器端分别有哪些区域(已确认 传输未确认 未传输)
- volitale 怎么搞

阿里二面

实习的工作？有什么感觉有难度的地方？和团队其他人怎么协调的？

实习之后有哪些成长？

业务可以改进的点？

有没有博客？开源项目？

未来三面职业规划？

看了看我大三的面试记录？问了一下当时面试挂掉的原因？我说算法当时不行，顺便问了下我今年的笔试情况

常用的语言？python和java比较

最近在看什么书？

技术：HTTPS 和HTTP区别是什么？ HTTPS 客户端服务器怎么交互的？

阿里三面

- 项目：介绍 难点 实现细节 和二面差不多
- 项目的平行对比……我们组的项目和平行项目（hadoop spark）相比 优势？
- eureka源码 Hystrix源码
- 分布式锁实现方式？
- zk的原理 源码
- Spring cloud/jdk设计模式 项目中的设计模式
- 序列化方式
- 最近在看什么书？ 平时怎么学习的

美团-到店事业部

美团面试难度一般，基本上在问一些基础知识，多准备基础知识就可以。

美团一面

- MVCC 在读方面有什么用途
- Future的缺陷 CompletionService 在依赖任务之间是如何实现的
- Tomcat框架的 selevet
- 算法稳定性的实际作用
- http 协议是什么 POST请求字段
- 算法：第 K大的元素
- 略去一些基础的问题，比较简单…

美团二面

算法1：第一个从n个数字的数组中等概率的取出m个数字

算法2：后缀树找最长重复的字符串

算法3：反转最后K个节点

http 请求 api 超时如何实现的(定时器有关)

mysql 索引 orderby 之后的字段要不要加进去 -> 以及 mysql orderby如何实现

操作系统两个进程写共享内存中一个位置 会不会出现不一致（和分页分段有关）

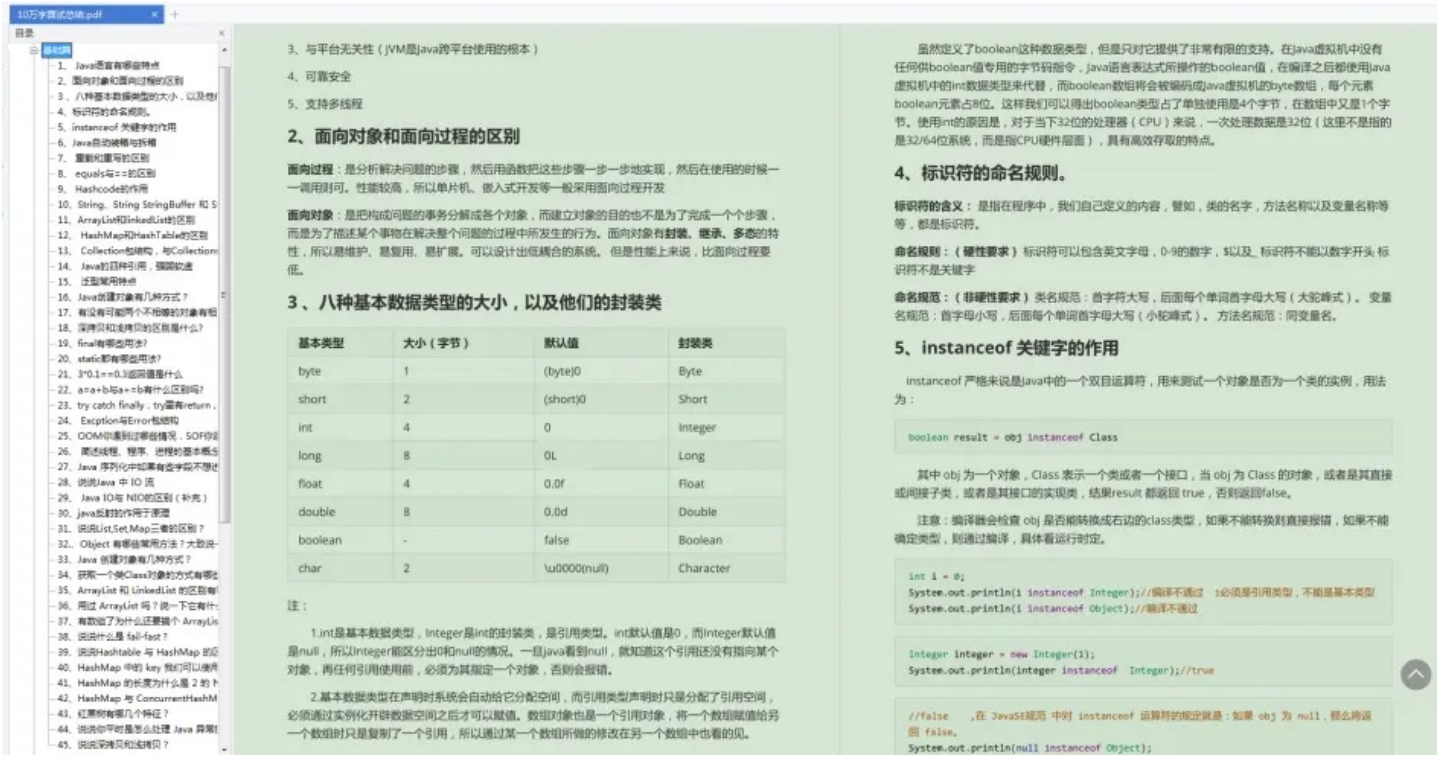
略去一些基础的问题，比较简单…

十万字面经

目录

笔记内容

基础篇



3、与平台无关性（JVM是Java跨平台使用的根本）

4、可靠安全

5、支持多线程

2、面向对象和面向过程的区别

面向过程：是分析解决问题的步骤，然后用函数把这些步骤一步一步地实现，然后在使用的时候一一调用则可。性能较高，所以单片机、嵌入式开发等一般采用面向过程开发

面向对象：是把构成问题的事务分解成各个对象，而建立对象的目的也不是为了完成一个个步骤，而是为了描述某个事物在解决整个问题的过程中所发生的行为。面向对象有**封装、继承、多态**的特性，所以易维护、易复用、易扩展。可以设计出低耦合的系统。但是性能上来说，比面向过程要低。

3、八种基本数据类型的大小，以及他们的封装类

基本类型	大小（字节）	默认值	封装类
byte	1	(byte)0	Byte
short	2	(short)0	Short
int	4	0	Integer
long	8	0L	Long
float	4	0.0f	Float
double	8	0.0d	Double
boolean	-	false	Boolean
char	2	\u0000(null)	Character

注：

1.int是基本数据类型，Integer是int的封装类，是引用类型。int默认值是0，而Integer默认值是null，所以Integer能区分出0和null的情况。一旦java看到null，就知道这个引用还没有指向某个对象，再任何引用使用前，必须为其指定一个对象，否则会报错。

2.基本数据类型在声明时系统会自动给它分配空间，而引用类型声明时只是分配了引用空间，必须通过实例化开辟数据空间之后才可以赋值。数组对象也是一个引用对象，将一个数组赋值给另一个数组时只是复制了一个引用，所以通过某一个数组所做的修改在另一个数组中也看的见。

虽然定义了boolean这种数据类型，但是只对它提供了非常有限的支持。在Java虚拟机中没有任何供boolean值专用的字节码指令，Java语言表达式所操作的boolean值，在编译之后都使用Java虚拟机中的int数据类型来代替，而boolean数组将会被编码成Java虚拟机的byte数组，每个元素boolean元素占8位。这样我们可以得出boolean类型占了单独使用是4个字节，在数组中又是1个字节。使用int的原因是，对于当下32位的处理器（CPU）来说，一次处理数据是32位（这里不是指的是32/64位系统，而是指CPU硬件层面），具有高效存取的特点。

4、标识符的命名规则。

标识符的含义：是指在程序中，我们自己定义的内容，譬如，类的名字，方法名称以及变量名称等等，都是标识符。

命名规则：（硬性要求）标识符可以包含英文字母，0-9的数字，$以及_标识符不能以数字开头 标识符不是关键字

命名规范：（非硬性要求）类名规范：首字符大写，后面每个单词首字母大写（大驼峰式）。 变量名规范：首字母小写，后面每个单词首字母大写（小驼峰式）。 方法名规范：同变量名。

5、instanceof 关键字的作用

instanceof 严格来说是Java中的一个双目运算符，用来测试一个对象是否为一个类的实例，用法为：

```
boolean result = obj instanceof Class
```

其中 obj 为一个对象，Class 表示一个类或者一个接口，当 obj 为 Class 的对象，或者是其直接或间接子类，或者是其接口的实现类，结果result 都返回 true，否则返回false。

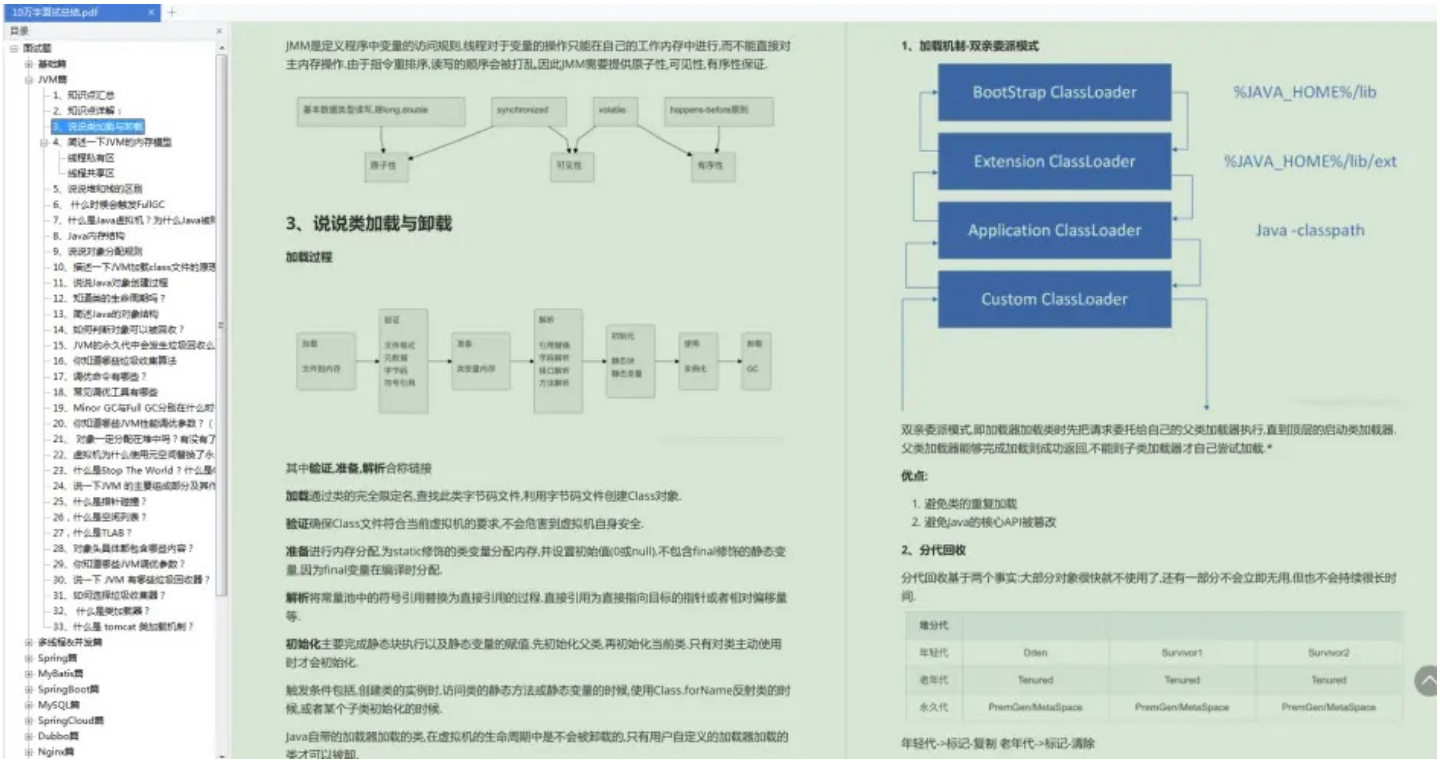
注意：编译器会检查 obj 是否能转换成右边的class类型，如果不能转换则直接报错，如果不能确定类型，则通过编译，具体看运行时定。

```
int i = 0;
System.out.println(i instanceof Integer);//编译不通过  i必须是引用类型，不能是基本类型
System.out.println(i instanceof Object);//编译不通过
```

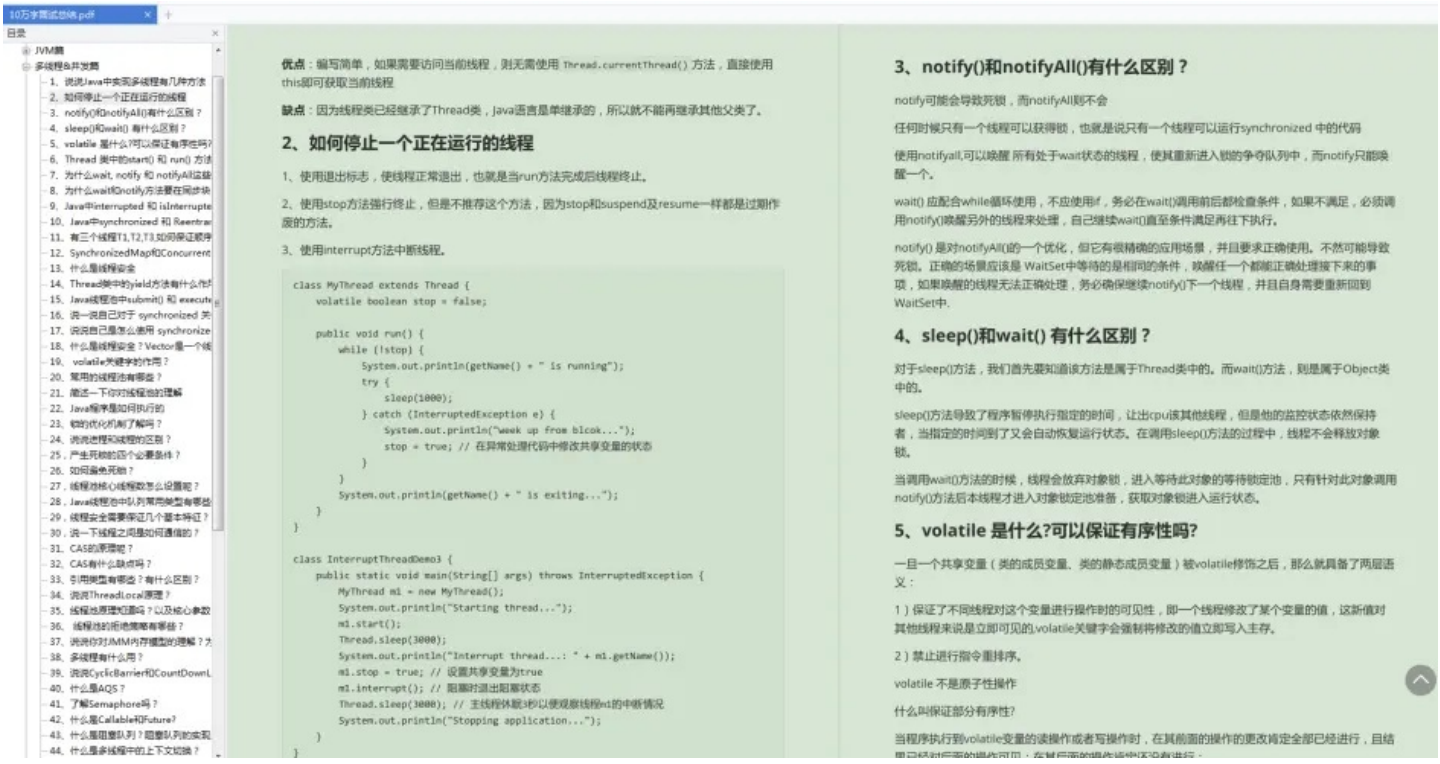
```
Integer integer = new Integer(1);
System.out.println(integer instanceof  Integer);//true
```

```
//false   ,在 JavaSE规范 中对 instanceof 运算符的规定就是：如果 obj 为 null，那么将返回 false。
System.out.println(null instanceof Object);
```

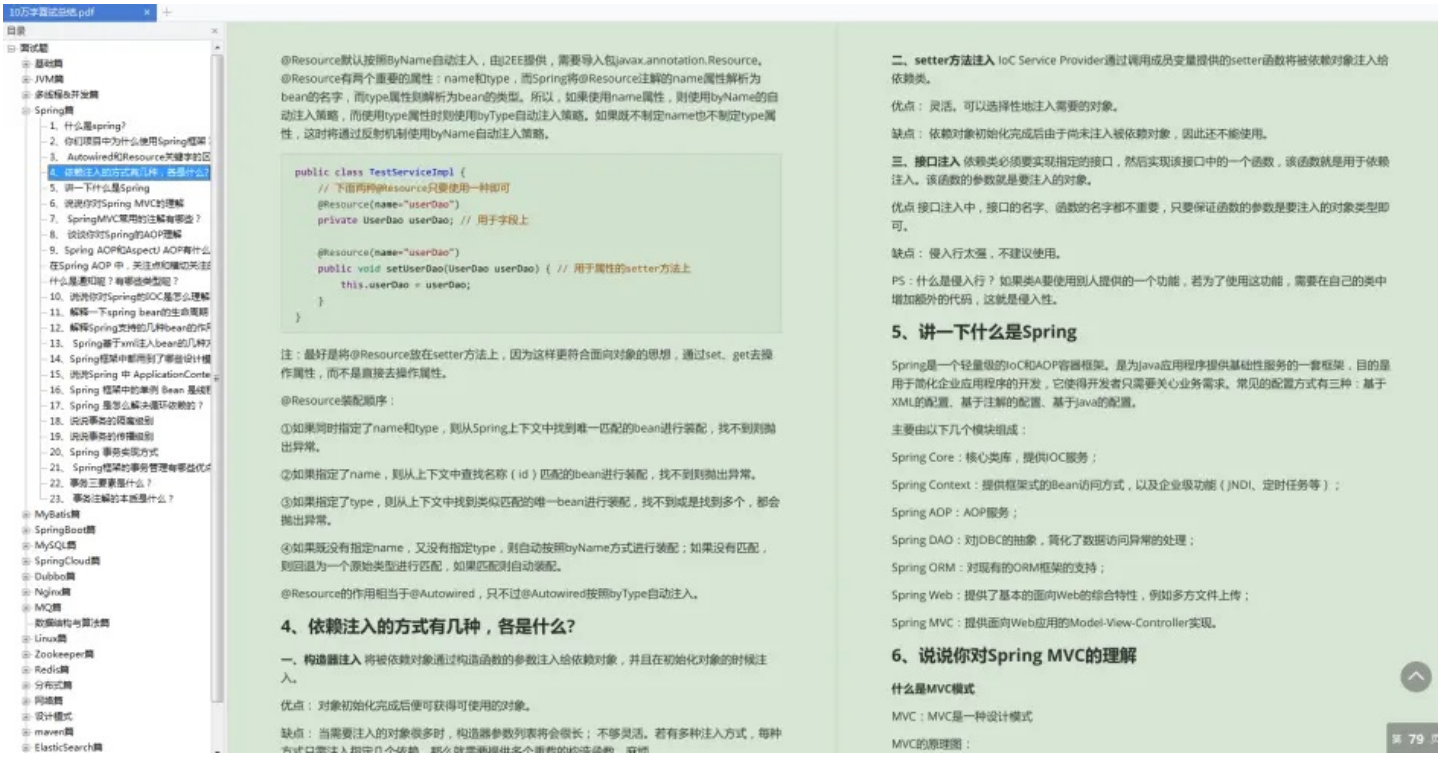
JVM篇



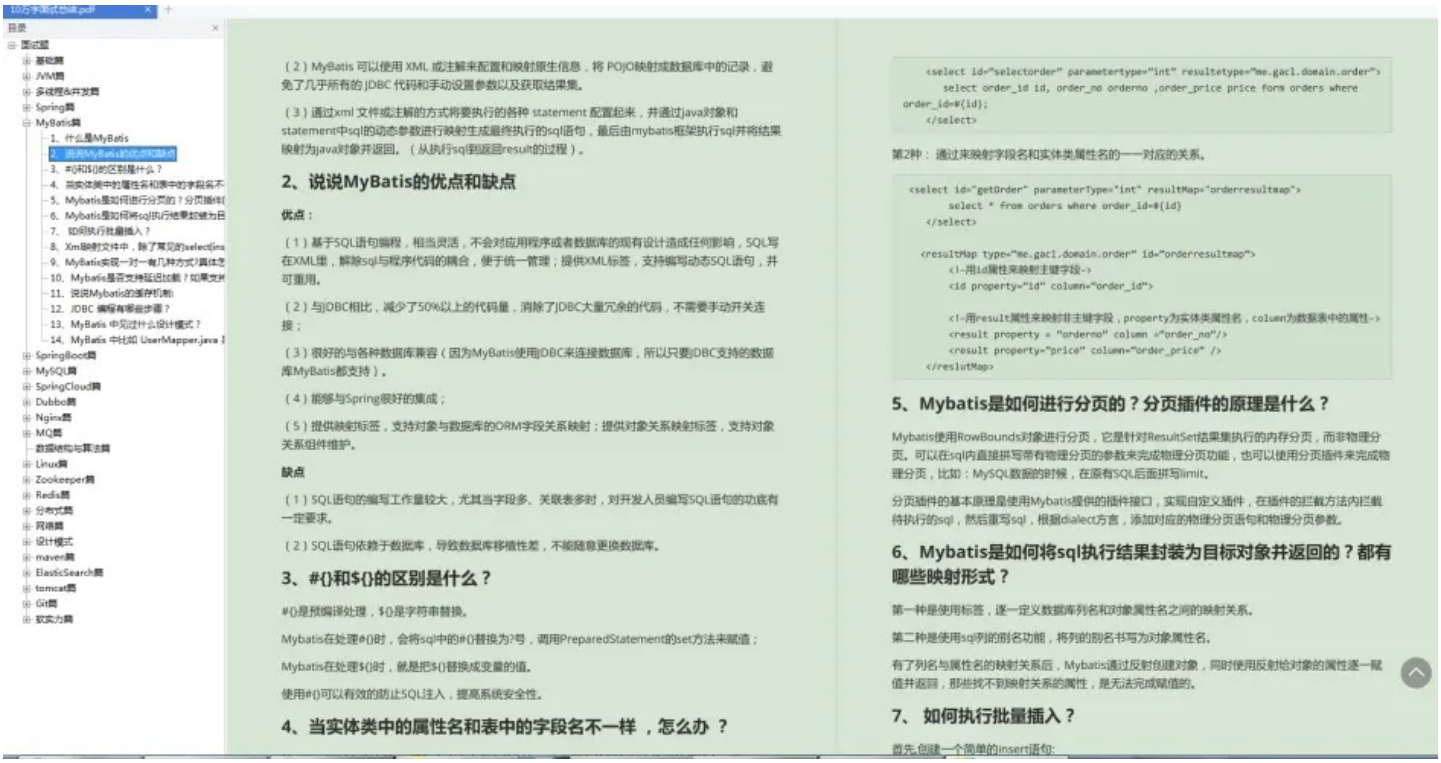
多线程&并发篇



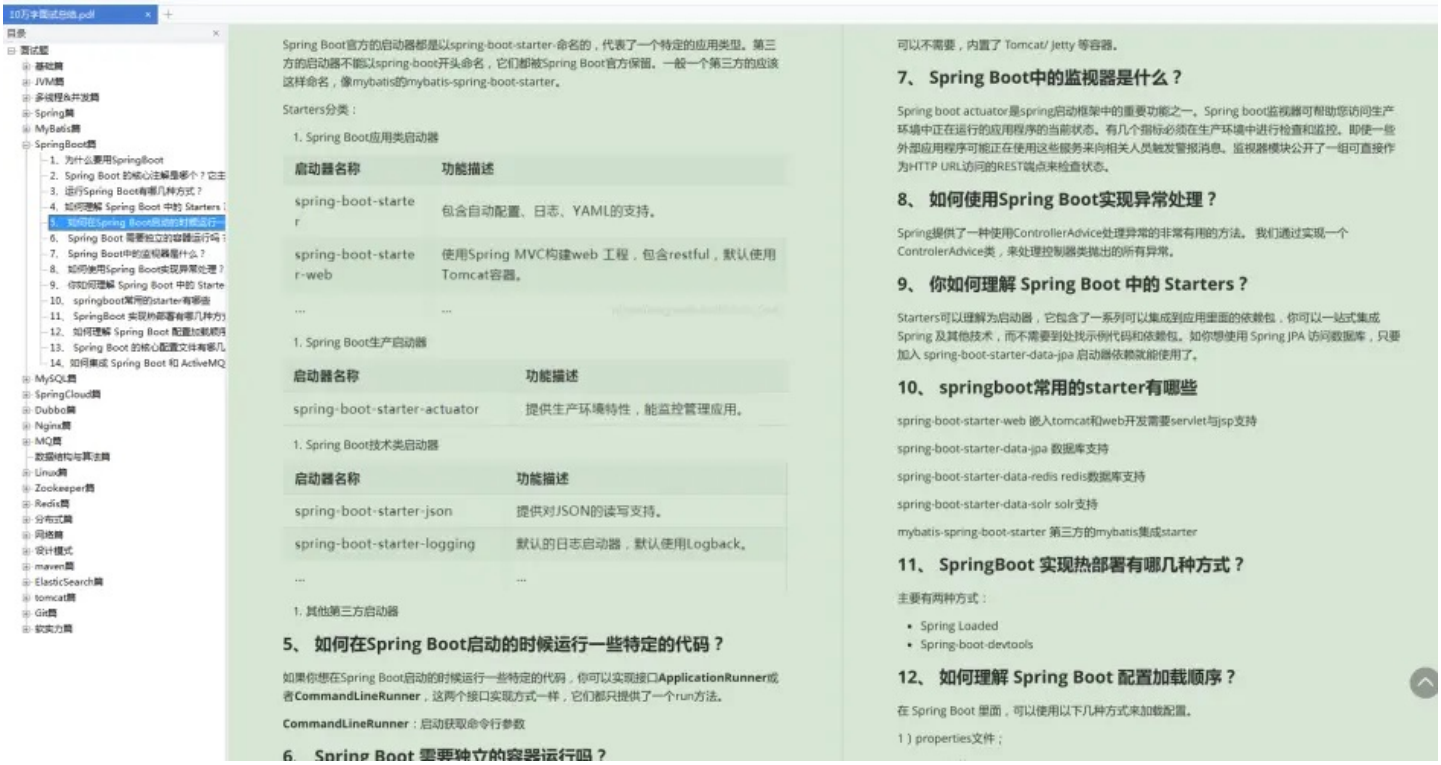
Spring篇



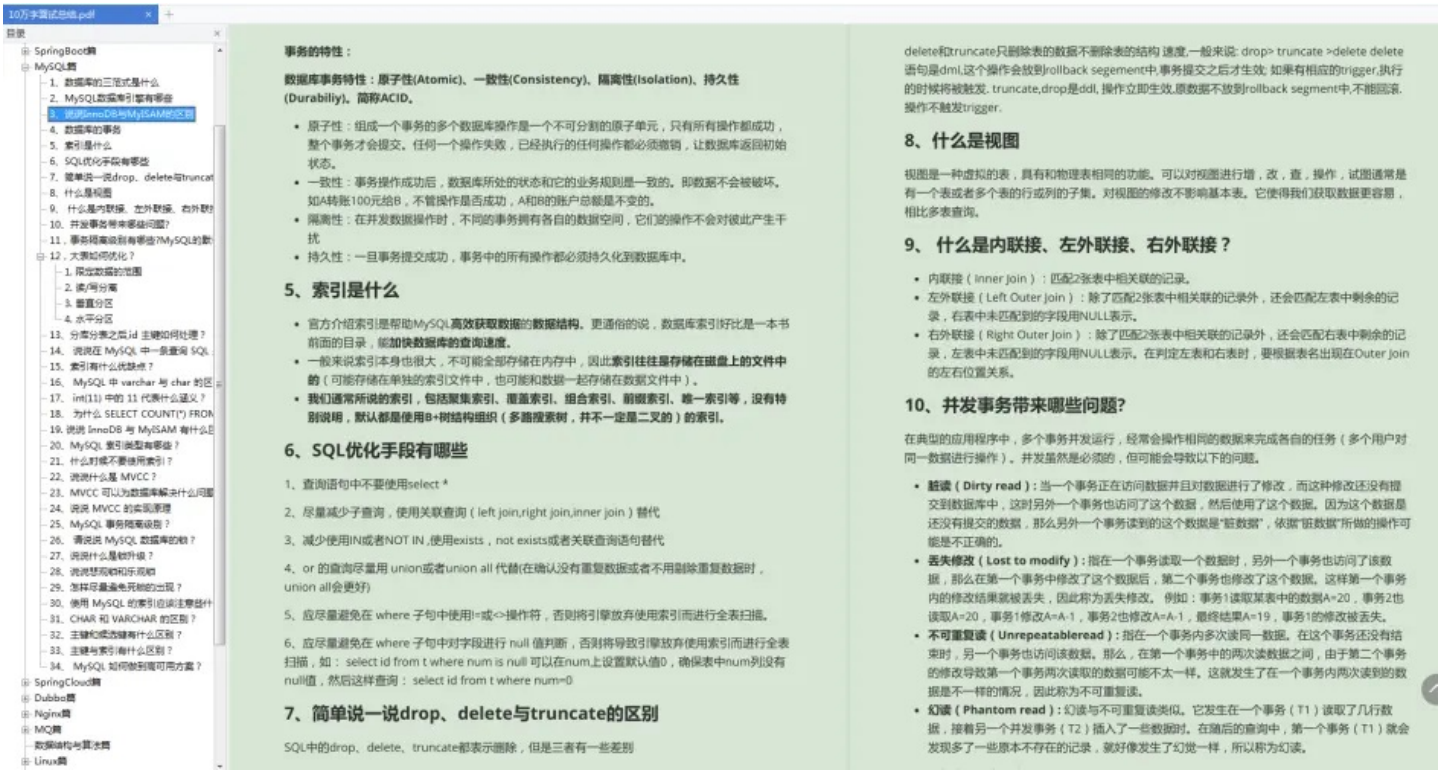
MyBatis篇



SpringBoot篇

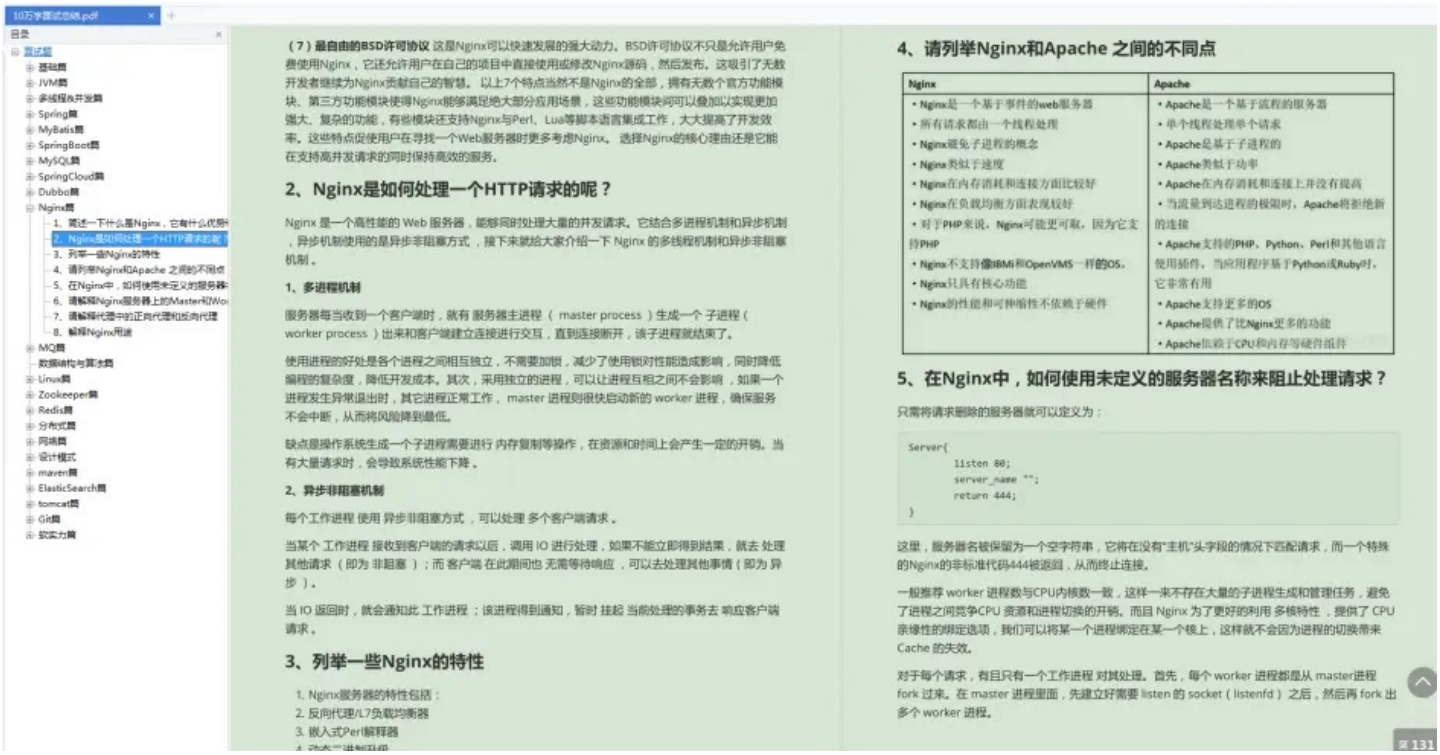

10万字面试总结.pdf
目录
面试篇
基础篇
JVM篇
多线程&并发篇
Spring篇
MyBatis篇
SpringBoot篇
1、为什么要用SpringBoot
2、Spring Boot 的核心注解是哪个？它主
3、运行Spring Boot有哪几种方式？
4、如何理解 Spring Boot 中的 Starters？
5、如何在Spring Boot启动的时候运行一
6、Spring Boot 需要独立的容器运行吗？
7、Spring Boot中的监视器是什么？
8、如何使用Spring Boot实现异常处理？
9、你如何理解 Spring Boot 中的 Starte
10、springboot常用的starter有哪些
11、SpringBoot 实现热部署有哪几种方
12、如何理解 Spring Boot 配置加载顺序
13、Spring Boot 的核心配置文件有哪几
14、如何集成 Spring Boot 和 ActiveMQ
MySQL篇
SpringCloud篇
Dubbo篇
Nginx篇
MQ篇
数据结构与算法篇
Linux篇
Zookeeper篇
Redis篇
分布式篇
网络篇
设计模式
maven篇
ElasticSearch篇
tomcat篇
Git篇
软实力篇
Spring Boot官方的启动器都是以spring-boot-starter-命名的，代表了一个特定的应用类型。第三方的启动器不能以spring-boot开头命名，它们都被Spring Boot官方保留。一般一个第三方的应该这样命名，像mybatis的mybatis-spring-boot-starter。
Starters分类：
1. Spring Boot应用类启动器
启动器名称 功能描述
spring-boot-starter 包含自动配置、日志、YAML的支持。
spring-boot-starter-web 使用Spring MVC构建web 工程，包含restful，默认使用Tomcat容器。
... ...
1. Spring Boot生产启动器
启动器名称 功能描述
spring-boot-starter-actuator 提供生产环境特性，能监控管理应用。
1. Spring Boot技术类启动器
启动器名称 功能描述
spring-boot-starter-json 提供对JSON的读写支持。
spring-boot-starter-logging 默认的日志启动器，默认使用Logback。
... ...
1. 其他第三方启动器
5、 如何在Spring Boot启动的时候运行一些特定的代码？
如果你想在Spring Boot启动的时候运行一些特定的代码，你可以实现接口ApplicationRunner或者CommandLineRunner，这两个接口实现方式一样，它们都只提供了一个run方法。
CommandLineRunner：启动获取命令行参数
6、 Spring Boot 需要独立的容器运行吗？
可以不需要，内置了 Tomcat/ Jetty 等容器。
7、 Spring Boot中的监视器是什么？
Spring boot actuator是spring启动框架中的重要功能之一。Spring boot监视器可帮助您访问生产环境中正在运行的应用程序的当前状态。有几个指标必须在生产环境中进行检查和监控。即使一些外部应用程序可能正在使用这些服务来向相关人员触发警报消息。监视器模块公开了一组可直接作为HTTP URL访问的REST端点来检查状态。
8、 如何使用Spring Boot实现异常处理？
Spring提供了一种使用ControllerAdvice处理异常的非常有用的方法。 我们通过实现一个ControlerAdvice类，来处理控制器类抛出的所有异常。
9、 你如何理解 Spring Boot 中的 Starters？
Starters可以理解为启动器，它包含了一系列可以集成到应用里面的依赖包，你可以一站式集成Spring 及其他技术，而不需要到处找示例代码和依赖包。如你想使用 Spring JPA 访问数据库，只要加入 spring-boot-starter-data-jpa 启动器依赖就能使用了。
10、 springboot常用的starter有哪些
spring-boot-starter-web 嵌入tomcat和web开发需要servlet与jsp支持
spring-boot-starter-data-jpa 数据库支持
spring-boot-starter-data-redis redis数据库支持
spring-boot-starter-data-solr solr支持
mybatis-spring-boot-starter 第三方的mybatis集成starter
11、 SpringBoot 实现热部署有哪几种方式？
主要有两种方式：
Spring Loaded
Spring-boot-devtools
12、 如何理解 Spring Boot 配置加载顺序？
在 Spring Boot 里面，可以使用以下几种方式来加载配置。
1）properties文件；

MySQL篇

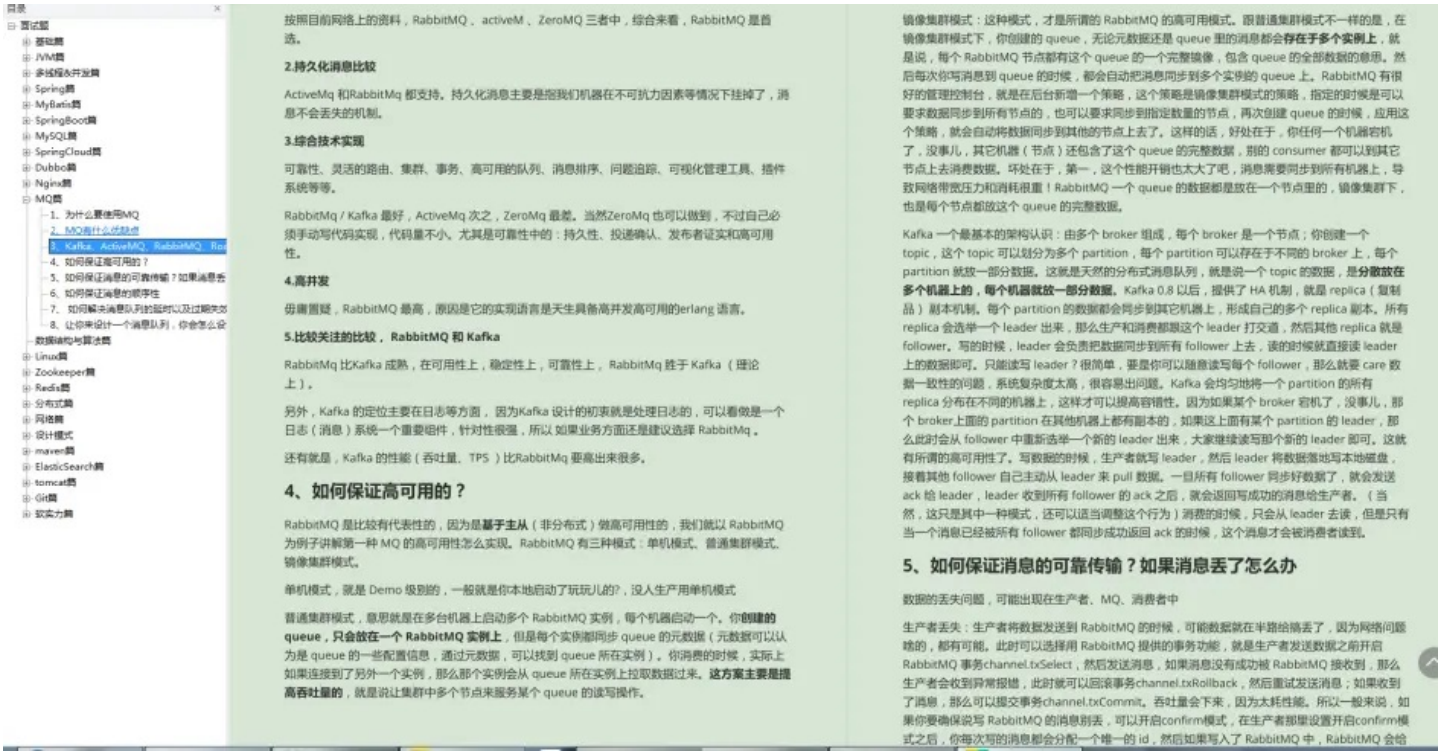

10万字面试总结.pdf
目录
SpringBoot篇
MySQL篇
1、数据库的三范式是什么
2、MySQL数据库引擎有哪些
3、说说InnoDB与MyISAM的区别
4、数据库的事务
5、索引是什么
6、SQL优化手段有哪些
7、简单说一说drop、delete与truncat
8、什么是视图
9、什么是内联接、左外联接、右外联
10、并发事务带来哪些问题?
11、事务隔离级别有哪些?MySQL的默
12、大表如何优化？
1、限定数据的范围
2、读/写分离
3、垂直分区
4、水平分区
13、分库分表之后,id 主键如何处理？
14、说说在 MySQL 中一条查询 SQL
15、索引有什么优缺点？
16、MySQL 中 varchar 与 char 的区
17、int(11) 中的 11 代表什么意义？
18、为什么 SELECT COUNT(*) FROM
19、说说 InnoDB 与 MyISAM 有什么区
20、MySQL 索引类型有哪些？
21、什么时候不要使用索引？
22、说说什么是 MVCC？
23、MVCC 可以为数据库解决什么问题
24、说说 MVCC 的实现原理
25、MySQL 事务隔离级别？
26、请说说 MySQL 数据库的锁？
27、说说什么是锁升级？
28、说说慧观锁和乐观锁
29、怎样尽量避免死锁的出现？
30、使用 MySQL 的索引应该注意些什
31、CHAR 和 VARCHAR 的区别？
32、主键和候选键有什么区别？
33、主键与索引有什么区别？
34、MySQL 如何做到高可用方案？
SpringCloud篇
Dubbo篇
Nginx篇
MQ篇
数据结构与算法篇
Linux篇
事务的特性：
数据库事务特性：原子性(Atomic)、一致性(Consistency)、隔离性(Isolation)、持久性(Durability)。简称ACID。
原子性：组成一个事务的多个数据库操作是一个不可分割的原子单元，只有所有操作都成功，整个事务才会提交。任何一个操作失败，已经执行的任何操作都必须撤销，让数据库返回初始状态。
一致性：事务操作成功后，数据库所处的状态和它的业务规则是一致的。即数据不会被破坏。如A转账100元给B，不管操作是否成功，A和B的账户总额是不变的。
隔离性：在并发数据操作时，不同的事务拥有各自的数据空间，它们的操作不会对彼此产生干扰
持久性：一旦事务提交成功，事务中的所有操作都必须持久化到数据库中。
5、索引是什么
官方介绍索引是帮助MySQL高效获取数据的数据结构。更通俗的说，数据库索引好比是一本书前面的目录，能加快数据库的查询速度。
一般来说索引本身也很大，不可能全部存储在内存中，因此索引往往是存储在磁盘上的文件中的（可能存储在单独的索引文件中，也可能和数据一起存储在数据文件中）。
我们通常所说的索引，包括聚集索引、覆盖索引、组合索引、前缀索引、唯一索引等，没有特别说明，默认都是使用B+树结构组织（多路搜索树，并不一定是二叉的）的索引。
6、SQL优化手段有哪些
1、查询语句中不要使用select *
2、尽量减少子查询，使用关联查询（left join,right join,inner join）替代
3、减少使用IN或者NOT IN ,使用exists，not exists或者关联查询语句替代
4、or 的查询尽量用 union或者union all 代替(在确认没有重复数据或者不用剔除重复数据时，union all会更好)
5、应尽量避免在 where 子句中使用!=或<>操作符，否则将引擎放弃使用索引而进行全表扫描。
6、应尽量避免在 where 子句中对字段进行 null 值判断，否则将导致引擎放弃使用索引而进行全表扫描，如： select id from t where num is null 可以在num上设置默认值0，确保表中num列没有null值，然后这样查询： select id from t where num=0
7、简单说一说drop、delete与truncate的区别
SQL中的drop、delete、truncate都表示删除，但是三者有一些差别
delete和truncate只删除表的数据不删除表的结构 速度,一般来说: drop> truncate >delete delete语句是dml,这个操作会放到rollback segement中,事务提交之后才生效; 如果有相应的trigger,执行的时候将被触发. truncate,drop是ddl, 操作立即生效,原数据不放到rollback segment中,不能回滚. 操作不触发trigger.
8、什么是视图
视图是一种虚拟的表，具有和物理表相同的功能。可以对视图进行增，改，查，操作，试图通常是有一个表或者多个表的行或列的子集。对视图的修改不影响基本表。它使得我们获取数据更容易，相比多表查询。
9、什么是内联接、左外联接、右外联接？
内联接（Inner Join）：匹配2张表中相关联的记录。
左外联接（Left Outer Join）：除了匹配2张表中相关联的记录外，还会匹配左表中剩余的记录，右表中未匹配到的字段用NULL表示。
右外联接（Right Outer Join）：除了匹配2张表中相关联的记录外，还会匹配右表中剩余的记录，左表中未匹配到的字段用NULL表示。在判定左表和右表时，要根据表名出现在Outer Join的左右位置关系。
10、并发事务带来哪些问题?
在典型的应用程序中，多个事务并发运行，经常会操作相同的数据来完成各自的任务（多个用户对同一数据进行操作）。并发虽然是必须的，但可能会导致以下的问题。
脏读（Dirty read）：当一个事务正在访问数据并且对数据进行了修改，而这种修改还没有提交到数据库中，这时另外一个事务也访问了这个数据，然后使用了这个数据。因为这个数据是还没有提交的数据，那么另外一个事务读到的这个数据是“脏数据”，依据“脏数据”所做的操作可能是不正确的。
丢失修改（Lost to modify）：指在一个事务读取一个数据时，另外一个事务也访问了该数据，那么在第一个事务中修改了这个数据后，第二个事务也修改了这个数据。这样第一个事务内的修改结果就被丢失，因此称为丢失修改。 例如：事务1读取某表中的数据A=20，事务2也读取A=20，事务1修改A=A-1，事务2也修改A=A-1，最终结果A=19，事务1的修改被丢失。
不可重复读（Unrepeatableread）：指在一个事务内多次读同一数据。在这个事务还没有结束时，另一个事务也访问该数据。那么，在第一个事务中的两次读数据之间，由于第二个事务的修改导致第一个事务两次读取的数据可能不太一样。这就发生了在一个事务内两次读到的数据是不一样的情况，因此称为不可重复读。
幻读（Phantom read）：幻读与不可重复读类似。它发生在一个事务（T1）读取了几行数据，接着另一个并发事务（T2）插入了一些数据时。在随后的查询中，第一个事务（T1）就会发现多了一些原本不存在的记录，就好像发生了幻觉一样，所以称为幻读。

Dubbo篇

Nginx篇



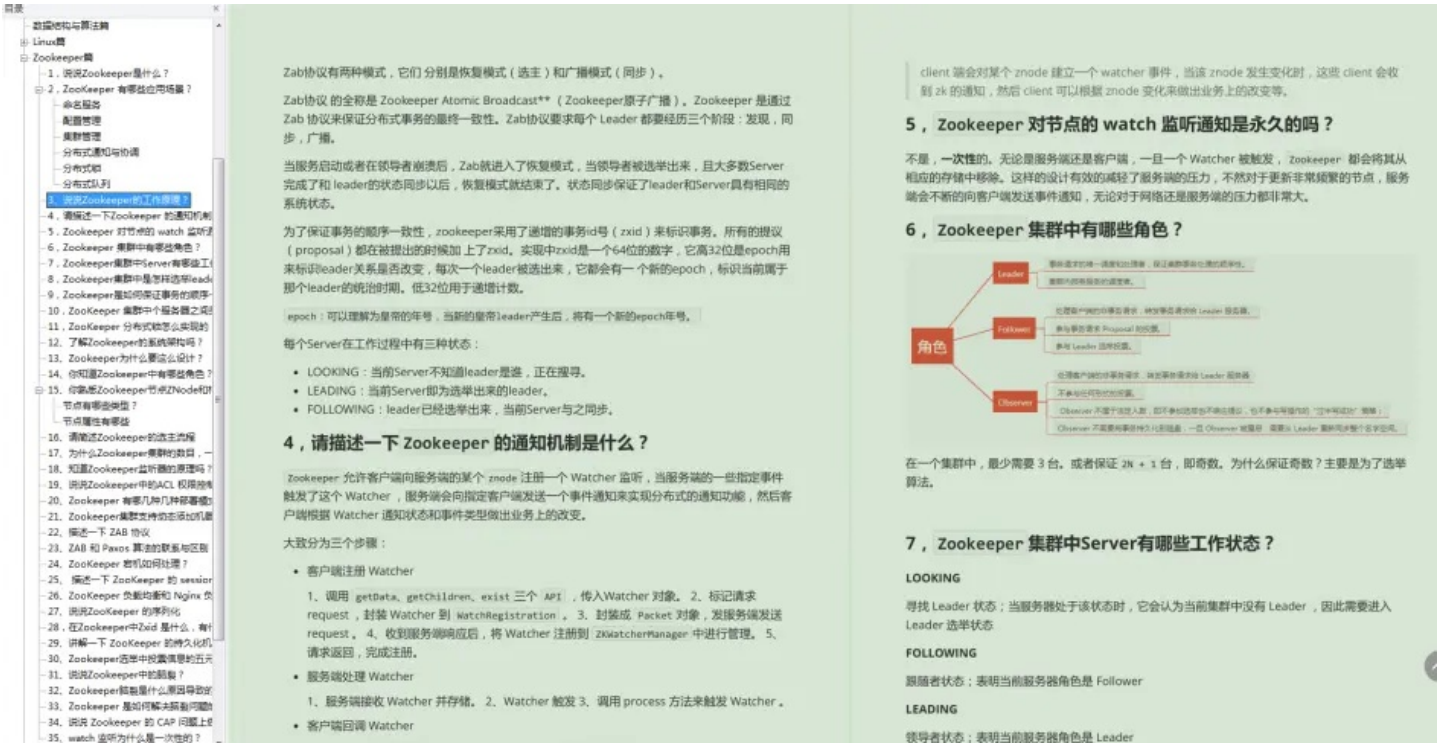
MQ篇



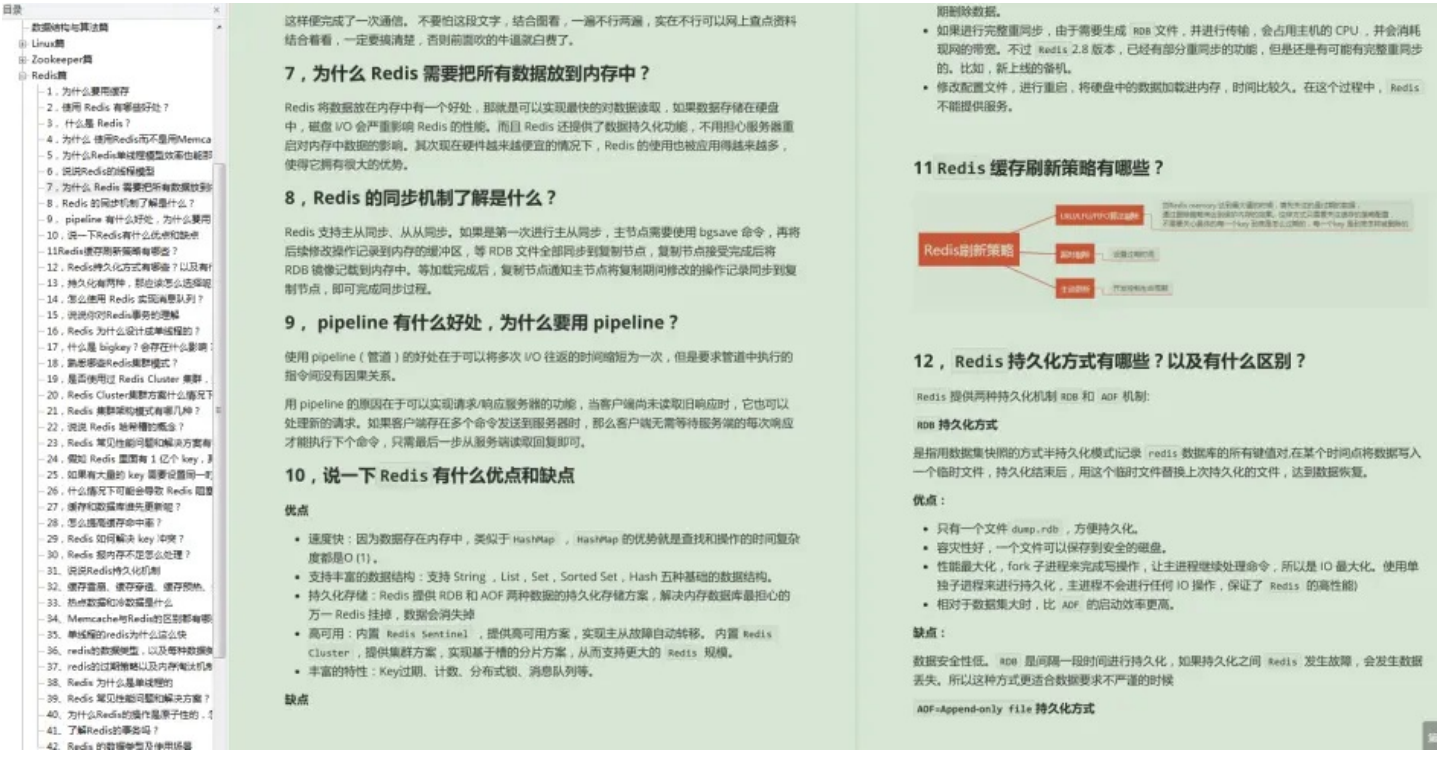
数据结构与算法篇

Linux篇

Zookeeper篇


目录
数据结构与算法篇
Linux篇
Zookeeper篇
1，说说Zookeeper是什么？
2，ZooKeeper 有哪些应用场景？
命名服务
配置管理
集群管理
分布式通知与协调
分布式锁
分布式队列
3，说说Zookeeper的工作原理？
4，请描述一下Zookeeper 的通知机制
5，Zookeeper 对节点的 watch 监听通
6，Zookeeper 集群中有哪些角色？
7，Zookeeper 集群中Server有哪些工
8，Zookeeper集群中是怎样选举leade
9，Zookeeper是如何保证事务的顺序
10，ZooKeeper 集群中个服务器之间
11，ZooKeeper 分布式锁怎么实现的
12，了解Zookeeper的系统架构吗？
13，Zookeeper为什么要这么设计？
14，你知道Zookeeper中有哪些角色？
15，你熟悉Zookeeper节点Znode的
节点有哪些类型？
节点属性有哪些
16，请简述Zookeeper的选主流程
17，为什么Zookeeper集群的数目，一
18，知道Zookeeper监听器的原理吗？
19，说说Zookeeper中的ACL 权限控制
20，Zookeeper 有哪几种部署模式
21，Zookeeper集群支持动态添加机器
22，描述一下 ZAB 协议
23，ZAB 和 Paxos 算法的联系与区别
24，ZooKeeper 宕机如何处理？
25，描述一下 ZooKeeper 的 session
26，ZooKeeper 负载均衡和 Nginx 负
27，说说ZooKeeper 的序列化
28，在Zookeeper中Zxid 是什么，有什
29，讲解一下 ZooKeeper 的持久化机
30，Zookeeper选举中投票信息的五元
31，说说Zookeeper中的脑裂？
32，Zookeeper脑裂是什么原因导致的
33，Zookeeper 是如何解决脑裂问题的
34，说说 Zookeeper 的 CAP 问题上的
35，watch 监听为什么是一次性的？
Zab协议有两种模式，它们 分别是恢复模式（选主）和广播模式（同步）。
Zab协议 的全称是 Zookeeper Atomic Broadcast**（Zookeeper原子广播）。Zookeeper 是通过 Zab 协议来保证分布式事务的最终一致性。Zab协议要求每个 Leader 都要经历三个阶段：发现，同步，广播。
当服务启动或者在领导者崩溃后，Zab就进入了恢复模式，当领导者被选举出来，且大多数Server 完成了和 leader的状态同步以后，恢复模式就结束了。状态同步保证了leader和Server具有相同的系统状态。
为了保证事务的顺序一致性，zookeeper采用了递增的事务id号（zxid）来标识事务。所有的提议（proposal）都在被提出的时候加 上了zxid。实现中zxid是一个64位的数字，它高32位是epoch用来标识leader关系是否改变，每次一个leader被选出来，它都会有一 个新的epoch，标识当前属于那个leader的统治时期。低32位用于递增计数。
epoch：可以理解为皇帝的年号，当新的皇帝leader产生后，将有一个新的epoch年号。
每个Server在工作过程中有三种状态：
- LOOKING：当前Server不知道leader是谁，正在搜寻。
- LEADING：当前Server即为选举出来的leader。
- FOLLOWING：leader已经选举出来，当前Server与之同步。
4，请描述一下 Zookeeper 的通知机制是什么？
Zookeeper 允许客户端向服务端的某个 znode 注册一个 Watcher 监听，当服务端的一些指定事件触发了这个 Watcher，服务端会向指定客户端发送一个事件通知来实现分布式的通知功能，然后客户端根据 Watcher 通知状态和事件类型做出业务上的改变。
大致分为三个步骤：
- 客户端注册 Watcher
  1、调用 getData、getChildren、exist 三个 API，传入Watcher 对象。 2、标记请求 request，封装 Watcher 到 WatchRegistration， 3、封装成 Packet 对象，发服务端发送 request。 4、收到服务端响应后，将 Watcher 注册到 ZKWatcherManager 中进行管理。 5、请求返回，完成注册。
- 服务端处理 Watcher
  1、服务端接收 Watcher 并存储。 2、Watcher 触发 3、调用 process 方法来触发 Watcher。
- 客户端回调 Watcher
client 端会对某个 znode 建立一个 watcher 事件，当该 znode 发生变化时，这些 client 会收到 zk 的通知，然后 client 可以根据 znode 变化来做出业务上的改变等。
5，Zookeeper 对节点的 watch 监听通知是永久的吗？
不是，**一次性的**。无论是服务端还是客户端，一旦一个 Watcher 被触发，Zookeeper 都会将其从相应的存储中移除。这样的设计有效的减轻了服务端的压力，不然对于更新非常频繁的节点，服务端会不断的向客户端发送事件通知，无论对于网络还是服务端的压力都非常大。
6，Zookeeper 集群中有哪些角色？
角色
Leader
Follower
Observer
在一个集群中，最少需要 3 台。或者保证 2N + 1 台，即奇数。为什么保证奇数？主要是为了选举算法。
7，Zookeeper 集群中Server有哪些工作状态？
LOOKING
寻找 Leader 状态；当服务器处于该状态时，它会认为当前集群中没有 Leader，因此需要进入 Leader 选举状态
FOLLOWING
跟随者状态；表明当前服务器角色是 Follower
LEADING
领导者状态；表明当前服务器角色是 Leader

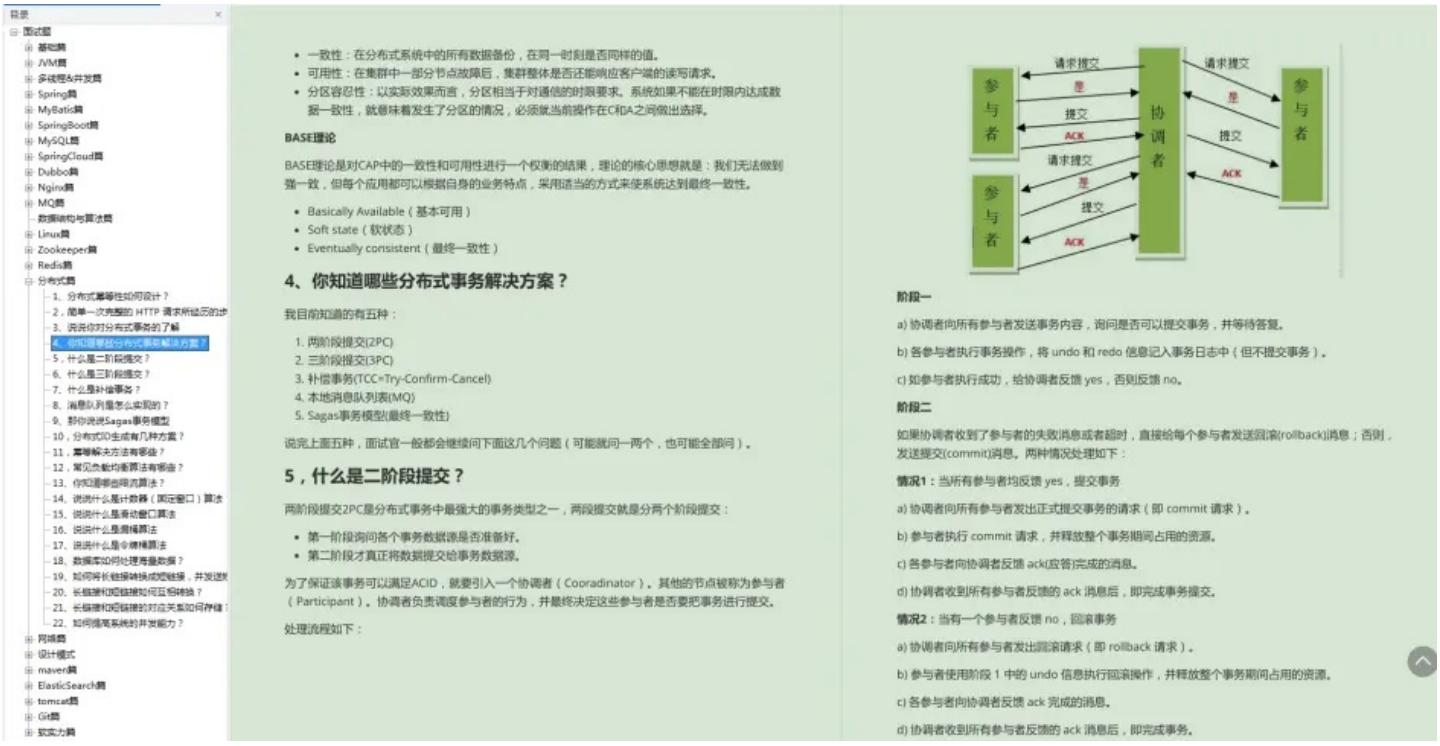
Redis篇


目录
数据结构与算法篇
Linux篇
Zookeeper篇
Redis篇
1，为什么要用缓存
2，使用 Redis 有哪些好处？
3，什么是 Redis？
4，为什么 使用Redis而不是用Memca
5，为什么Redis单线程模型效率也能那
6，说说Redis的线程模型
7，为什么 Redis 需要把所有数据放到
8，Redis 的同步机制了解是什么？
9，pipeline 有什么好处，为什么要用
10，说一下Redis有什么优点和缺点
11Redis缓存刷新策略有哪些？
12，Redis持久化方式有哪些？以及有什
13，持久化有两种，那应该怎么选择呢
14，怎么使用 Redis 实现消息队列？
15，说说你对Redis事务的理解
16，Redis 为什么设计成单线程的？
17，什么是 bigkey？会存在什么影响
18，集群哪些Redis集群模式？
19，是否使用过 Redis Cluster 集群，
20，Redis Cluster集群方案什么情况下
21，Redis 集群架构模式有哪几种？
22，说说 Redis 哨兵模式的概念？
23，Redis 常见性能问题和解决方案有
24，假如 Redis 里面有 1 亿个 key，其
25，如果有大量的 key 需要设置同一时
26，什么情况下可能会导致 Redis 阻塞
27，缓存和数据库谁先更新呢？
28，怎么提高缓存命中率？
29，Redis 如何解决 key 冲突？
30，Redis 报内存不足怎么处理？
31，说说Redis持久化机制
32，缓存雪崩、缓存穿透、缓存预热、
33，热点数据和冷数据是什么
34，Memcache与Redis的区别都有哪些
35，单线程的redis为什么这么快
36，redis的数据类型，以及每种数据类
37，redis的过期策略以及内存淘汰机制
38，Redis 为什么是单线程的
39，Redis 常见性能问题和解决方案？
40，为什么Redis的操作是原子性的，怎
41，了解Redis的事务吗？
42，Redis 的数据类型及使用场景
这样便完成了一次通信。不要怕这段文字，结合图看，一遍不行两遍，实在不行可以网上查点资料结合着看，一定要搞清楚，否则前面吹的牛逼就白费了。
7，为什么 Redis 需要把所有数据放到内存中？
Redis 将数据放在内存中有一个好处，那就是可以实现最快的对数据读取，如果数据存储在硬盘中，磁盘 I/O 会严重影响 Redis 的性能。而且 Redis 还提供了数据持久化功能，不用担心服务器重启对内存中数据的影响。其次现在硬件越来越便宜的情况下，Redis 的使用也被应用得越来越多，使得它拥有很大的优势。
8，Redis 的同步机制了解是什么？
Redis 支持主从同步、从从同步。如果是第一次进行主从同步，主节点需要使用 bgsave 命令，再将后续修改操作记录到内存的缓冲区，等 RDB 文件全部同步到复制节点，复制节点接受完成后将 RDB 镜像记载到内存中。等加载完成后，复制节点通知主节点将复制期间修改的操作记录同步到复制节点，即可完成同步过程。
9，pipeline 有什么好处，为什么要用 pipeline？
使用 pipeline（管道）的好处在于可以将多次 I/O 往返的时间缩短为一次，但是要求管道中执行的指令间没有因果关系。
用 pipeline 的原因在于可以实现请求/响应服务器的功能，当客户端尚未读取旧响应时，它也可以处理新的请求。如果客户端存在多个命令发送到服务器时，那么客户端无需等待服务端的每次响应才能执行下个命令，只需最后一步从服务端读取回复即可。
10，说一下 Redis 有什么优点和缺点
优点
- 速度快：因为数据存在内存中，类似于 HashMap，HashMap 的优势就是查找和操作的时间复杂度都是O (1)。
- 支持丰富的数据结构：支持 String，List，Set，Sorted Set，Hash 五种基础的数据结构。
- 持久化存储：Redis 提供 RDB 和 AOF 两种数据的持久化存储方案，解决内存数据库最担心的万一 Redis 挂掉，数据会消失掉
- 高可用：内置 Redis Sentinel，提供高可用方案，实现主从故障自动转移。 内置 Redis Cluster，提供集群方案，实现基于槽的分片方案，从而支持更大的 Redis 规模。
- 丰富的特性：Key过期、计数、分布式锁、消息队列等。

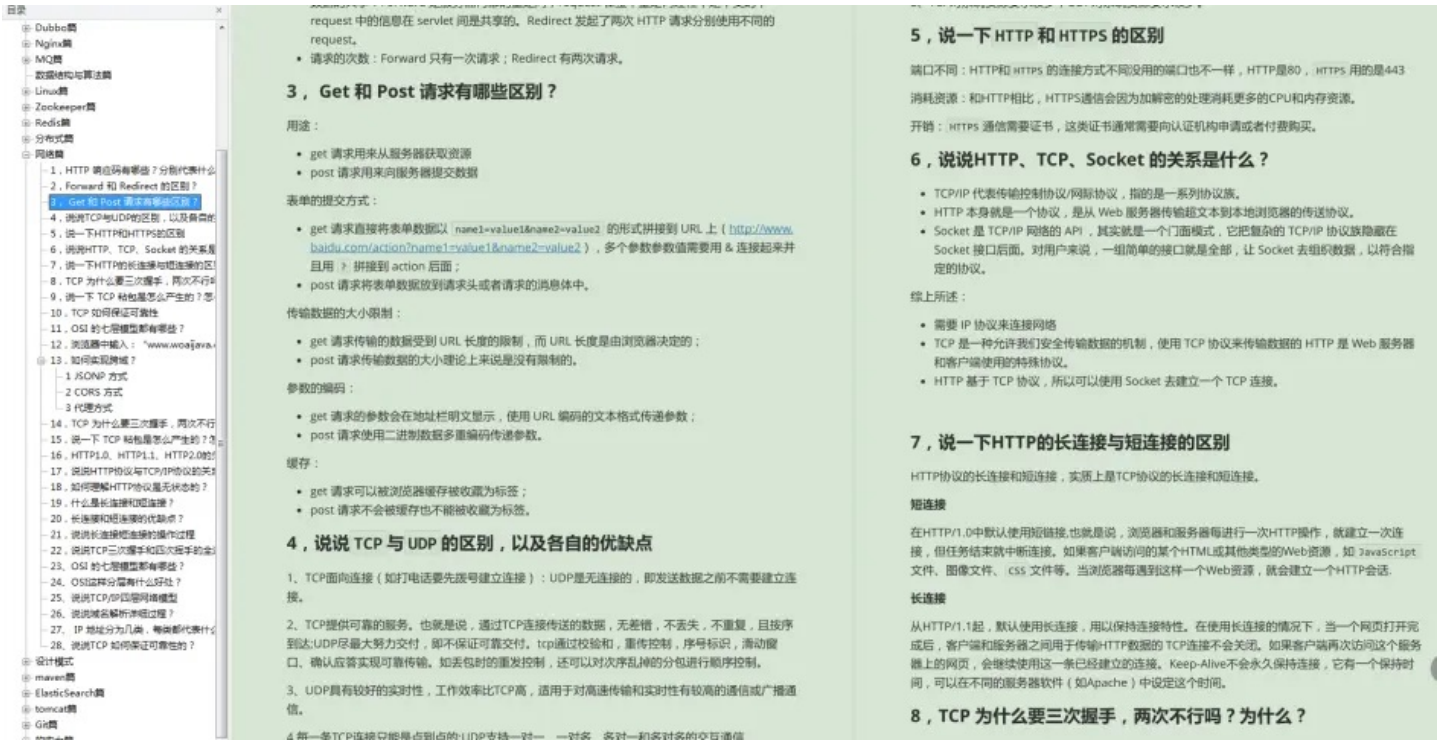
缺点
期删除数据。
- 如果进行完整重同步，由于需要生成 RDB 文件，并进行传输，会占用主机的 CPU，并会消耗现网的带宽。不过 Redis 2.8 版本，已经有部分重同步的功能，但是还是有可能有完整重同步的。比如，新上线的备机。
- 修改配置文件，进行重启，将硬盘中的数据加载进内存，时间比较久。在这个过程中，Redis 不能提供服务。
11 Redis 缓存刷新策略有哪些？
Redis刷新策略
12，Redis 持久化方式有哪些？以及有什么区别？
Redis 提供两种持久化机制 RDB 和 AOF 机制:
RDB 持久化方式
是指用数据集快照的方式半持久化模式记录 redis 数据库的所有键值对,在某个时间点将数据写入一个临时文件，持久化结束后，用这个临时文件替换上次持久化的文件，达到数据恢复。
优点：
- 只有一个文件 dump.rdb，方便持久化。
- 容灾性好，一个文件可以保存到安全的磁盘。
- 性能最大化，fork 子进程来完成写操作，让主进程继续处理命令，所以是 IO 最大化。使用单独子进程来进行持久化，主进程不会进行任何 IO 操作，保证了 Redis 的高性能)
- 相对于数据集大时，比 AOF 的启动效率更高。

缺点：
数据安全性低。RDB 是间隔一段时间进行持久化，如果持久化之间 Redis 发生故障，会发生数据丢失。所以这种方式更适合数据要求不严谨的时候
AOF=Append-only file 持久化方式

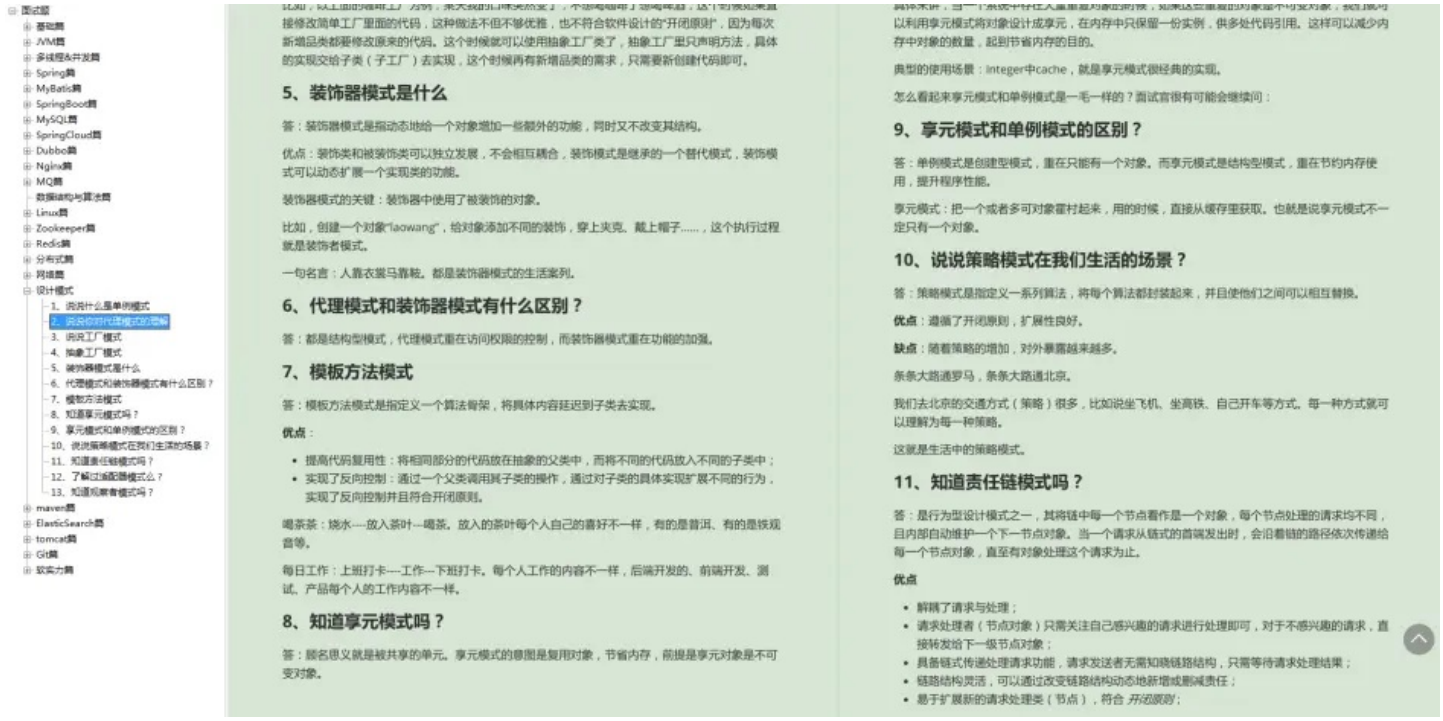
分布式篇


目录
面试题
基础篇
JVM篇
多线程&并发篇
Spring篇
MyBatis篇
SpringBoot篇
MySQL篇
SpringCloud篇
Dubbo篇
Nginx篇
MQ篇
数据结构与算法篇
Linux篇
Zookeeper篇
Redis篇
分布式篇
1、分布式幂等性如何设计？
2、简单一次完整的 HTTP 请求所经历的步
3、说说你对分布式事务的了解
4、你知道哪些分布式事务解决方案？
5、什么是二阶段提交？
6、什么是三阶段提交？
7、什么是补偿事务？
8、消息队列是怎么实现的？
9、那你说说Sagas事务模型
10、分布式ID生成有几种方案？
11、幂等解决方法有哪些？
12、常见负载均衡算法有哪些？
13、你知道哪些限流算法？
14、说说什么是计数器（固定窗口）算法
15、说说什么是滑动窗口算法
16、说说什么是漏桶算法
17、说说什么是令牌桶算法
18、数据库如何处理海量数据？
19、如何将长链接转换成短链接，并发送
20、长链接和短链接如何互相转换？
21、长链接和短链接的对应关系如何存储
22、如何提高系统的并发能力？
网络篇
设计模式
maven篇
ElasticSearch篇
tomcat篇
Git篇
软实力篇
一致性：在分布式系统中的所有数据备份，在同一时刻是否同样的值。
可用性：在集群中一部分节点故障后，集群整体是否还能响应客户端的读写请求。
分区容忍性：以实际效果而言，分区相当于对通信的时限要求。系统如果不能在时限内达成数据一致性，就意味着发生了分区的情况，必须就当前操作在C和A之间做出选择。
BASE理论
BASE理论是对CAP中的一致性和可用性进行一个权衡的结果，理论的核心思想就是：我们无法做到强一致，但每个应用都可以根据自身的业务特点，采用适当的方式来使系统达到最终一致性。
Basically Available（基本可用）
Soft state（软状态）
Eventually consistent（最终一致性）
4、你知道哪些分布式事务解决方案？
我目前知道的有五种：
1. 两阶段提交(2PC)
2. 三阶段提交(3PC)
3. 补偿事务(TCC=Try-Confirm-Cancel)
4. 本地消息队列表(MQ)
5. Sagas事务模型(最终一致性)
说完上面五种，面试官一般都会继续问下面这几个问题（可能就问一两个，也可能全部问）。
5，什么是二阶段提交？
两阶段提交2PC是分布式事务中最强大的事务类型之一，两段提交就是分两个阶段提交：
第一阶段询问各个事务数据源是否准备好。
第二阶段才真正将数据提交给事务数据源。
为了保证该事务可以满足ACID，就要引入一个协调者（Cooradinator）。其他的节点被称为参与者（Participant）。协调者负责调度参与者的行为，并最终决定这些参与者是否要把事务进行提交。
处理流程如下：
请求提交
是
提交
ACK
参与者
协调者
阶段一
a) 协调者向所有参与者发送事务内容，询问是否可以提交事务，并等待答复。
b) 各参与者执行事务操作，将 undo 和 redo 信息记入事务日志中（但不提交事务）。
c) 如参与者执行成功，给协调者反馈 yes，否则反馈 no。
阶段二
如果协调者收到了参与者的失败消息或者超时，直接给每个参与者发送回滚(rollback)消息；否则，发送提交(commit)消息。两种情况处理如下：
情况1：当所有参与者均反馈 yes，提交事务
a) 协调者向所有参与者发出正式提交事务的请求（即 commit 请求）。
b) 参与者执行 commit 请求，并释放整个事务期间占用的资源。
c) 各参与者向协调者反馈 ack(应答)完成的消息。
d) 协调者收到所有参与者反馈的 ack 消息后，即完成事务提交。
情况2：当有一个参与者反馈 no，回滚事务
a) 协调者向所有参与者发出回滚请求（即 rollback 请求）。
b) 参与者使用阶段 1 中的 undo 信息执行回滚操作，并释放整个事务期间占用的资源。
c) 各参与者向协调者反馈 ack 完成的消息。
d) 协调者收到所有参与者反馈的 ack 消息后，即完成事务。

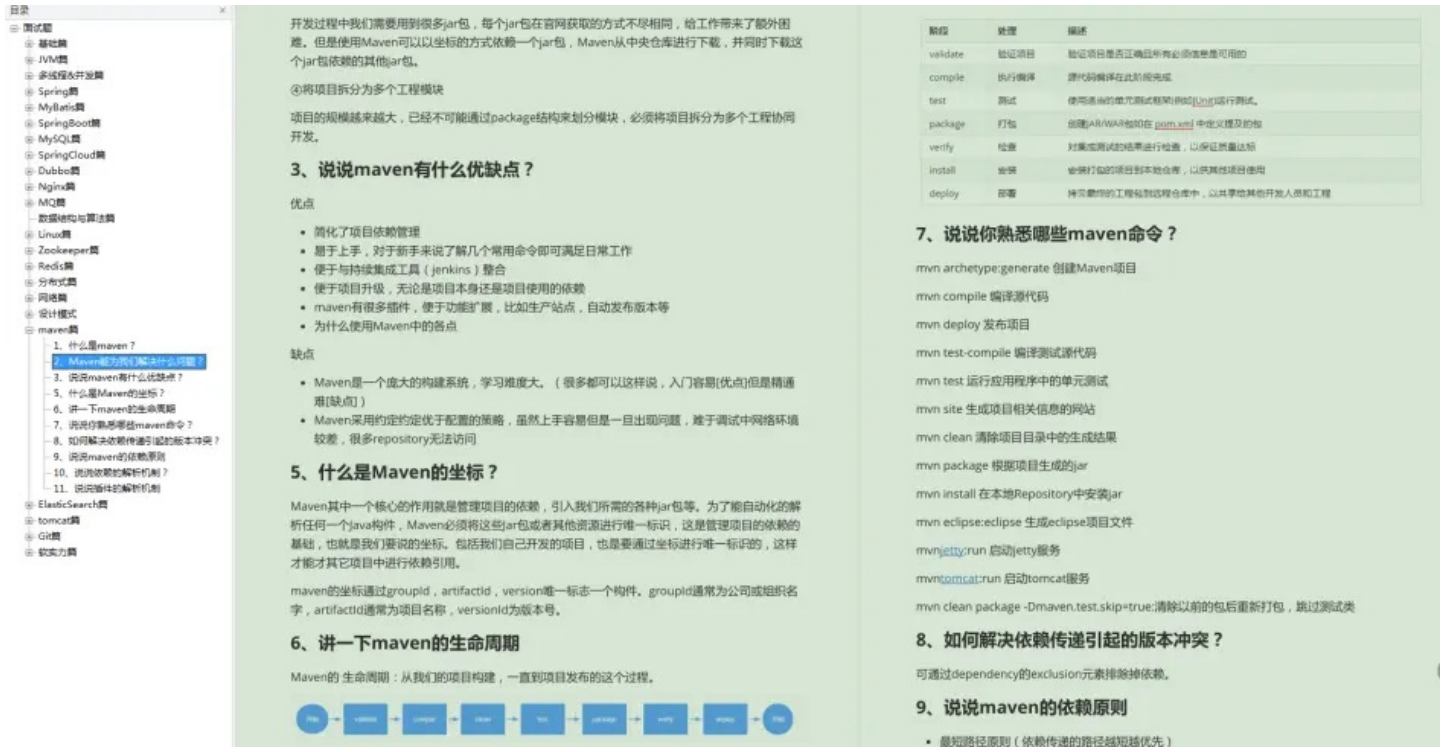
网络篇


目录
Dubbo篇
Nginx篇
MQ篇
数据结构与算法篇
Linux篇
Zookeeper篇
Redis篇
分布式篇
网络篇
1，HTTP 响应码有哪些？分别代表什么
2，Forward 和 Redirect 的区别？
3，Get 和 Post 请求有哪些区别？
4，说说TCP与UDP的区别，以及各自的
5，说一下HTTP和HTTPS的区别
6，说说HTTP、TCP、Socket 的关系是
7，说一下HTTP的长连接与短连接的区
8，TCP 为什么要三次握手，两次不行
9，说一下 TCP 粘包是怎么产生的？怎
10，TCP 如何保证可靠性
11，OSI 的七层模型都有哪些？
12，浏览器中输入：“www.woaijava.
13，如何实现跨域？
1 JSONP 方式
2 CORS 方式
3 代理方式
14，TCP 为什么要三次握手，两次不行
15，说一下 TCP 粘包是怎么产生的？怎
16，HTTP1.0、HTTP1.1、HTTP2.0的
17，说说HTTP协议与TCP/IP协议的关
18，如何理解HTTP协议是无状态的？
19，什么是长连接和短连接？
20，长连接和短连接的优缺点？
21，说说长连接短连接的操作过程
22，说说TCP三次握手和四次握手的全
23，OSI 的七层模型都有哪些？
24，OSI监视分层有什么好处？
25，说说TCP/IP四层网络模型
26，说说域名解析详细过程？
27，IP 地址分为几类，每类都代表什么
28，说说TCP 如何保证可靠性的？
设计模式
maven篇
ElasticSearch篇
tomcat篇
Git篇
request 中的信息在 servlet 间是共享的。Redirect 发起了两次 HTTP 请求分别使用不同的 request。
请求的次数：Forward 只有一次请求；Redirect 有两次请求。
3， Get 和 Post 请求有哪些区别？
用途：
get 请求用来从服务器获取资源
post 请求用来向服务器提交数据
表单的提交方式：
get 请求直接将表单数据以 name1=value1&name2=value2 的形式拼接到 URL 上（http://www.baidu.com/action?name1=value1&name2=value2），多个参数参数值需要用 & 连接起来并且用 ？ 拼接到 action 后面；
post 请求将表单数据放到请求头或者请求的消息体中。
传输数据的大小限制：
get 请求传输的数据受到 URL 长度的限制，而 URL 长度是由浏览器决定的；
post 请求传输数据的大小理论上来说是没有限制的。
参数的编码：
get 请求的参数会在地址栏明文显示，使用 URL 编码的文本格式传递参数；
post 请求使用二进制数据多重编码传递参数。
缓存：
get 请求可以被浏览器缓存被收藏为标签；
post 请求不会被缓存也不能被收藏为标签。
4，说说 TCP 与 UDP 的区别，以及各自的优缺点
1、TCP面向连接（如打电话要先拨号建立连接）：UDP是无连接的，即发送数据之前不需要建立连接。
2、TCP提供可靠的服务。也就是说，通过TCP连接传送的数据，无差错，不丢失，不重复，且按序到达;UDP尽最大努力交付，即不保证可靠交付。tcp通过校验和，重传控制，序号标识，滑动窗口、确认应答实现可靠传输。如丢包时的重发控制，还可以对次序乱掉的分包进行顺序控制。
3、UDP具有较好的实时性，工作效率比TCP高，适用于对高速传输和实时性有较高的通信或广播通信。
4 每一条TCP连接只能是点到点的;UDP支持一对一，一对多，多对一和多对多的交互通信
5，说一下 HTTP 和 HTTPS 的区别
端口不同：HTTP和 HTTPS 的连接方式不同没用的端口也不一样，HTTP是80， HTTPS 用的是443
消耗资源：和HTTP相比，HTTPS通信会因为加解密的处理消耗更多的CPU和内存资源。
开销： HTTPS 通信需要证书，这类证书通常需要向认证机构申请或者付费购买。
6，说说HTTP、TCP、Socket 的关系是什么？
TCP/IP 代表传输控制协议/网际协议，指的是一系列协议族。
HTTP 本身就是一个协议，是从 Web 服务器传输超文本到本地浏览器的传送协议。
Socket 是 TCP/IP 网络的 API，其实就是一个门面模式，它把复杂的 TCP/IP 协议族隐藏在 Socket 接口后面。对用户来说，一组简单的接口就是全部，让 Socket 去组织数据，以符合指定的协议。
综上所述：
需要 IP 协议来连接网络
TCP 是一种允许我们安全传输数据的机制，使用 TCP 协议来传输数据的 HTTP 是 Web 服务器和客户端使用的特殊协议。
HTTP 基于 TCP 协议，所以可以使用 Socket 去建立一个 TCP 连接。
7，说一下HTTP的长连接与短连接的区别
HTTP协议的长连接和短连接，实质上是TCP协议的长连接和短连接。
短连接
在HTTP/1.0中默认使用短链接,也就是说，浏览器和服务器每进行一次HTTP操作，就建立一次连接，但任务结束就中断连接。如果客户端访问的某个HTML或其他类型的Web资源，如 JavaScript 文件、图像文件、 CSS 文件等。当浏览器每遇到这样一个Web资源，就会建立一个HTTP会话.
长连接
从HTTP/1.1起，默认使用长连接，用以保持连接特性。在使用长连接的情况下，当一个网页打开完成后，客户端和服务器之间用于传输HTTP数据的 TCP连接不会关闭。如果客户端再次访问这个服务器上的网页，会继续使用这一条已经建立的连接。Keep-Alive不会永久保持连接，它有一个保持时间，可以在不同的服务器软件（如Apache）中设定这个时间。
8，TCP 为什么要三次握手，两次不行吗？为什么？

设计模式



maven篇



ElasticSearch篇

tomcat篇

Git篇

等，事情会处于混乱的状态。问题是，你不想把完成了一半的工作的提交，以便你以后就可以回到当前的工作。解决这个问题的答案是 git stash。

再解释什么是git stash。

stash 会将你的工作目录，即修改后的跟踪文件和暂存的更改保存在一堆未完成的更改中，你可以随时重新应用这些更改。

10、什么是git stash drop？

通过说明我们使用 `git stash drop` 的目的来回答这个问题。

`git stash drop` 命令用于删除隐藏的项目。默认情况下，它将删除最后添加的存储项，如果提供参数的话，它还可以删除特定项。

下面举个例子。

如果要从隐藏项目列表中删除特定的存储项目，可以使用以下命令：

git stash list：它将显示隐藏项目列表，如：

stash@{0}: WIP on master: 049d078 added the index file stash@{1}: WIP on master: c264051 Revert “added file_size” stash@{2}: WIP on master: 21d80a5 added number to log

如果要删除名为 stash@{0} 的项目，请使用命令 **git stash drop stash@{0}**。

11.、如何找到特定提交中已更改的文件列表？

对于这个问题，不能仅仅是提供命令，还要解释这个命令究竟做了些什么。

要获取特定提交中已更改的列表文件，请使用以下命令：

git diff-tree -r {hash}

给定提交哈希，这将列出在该提交中更改或添加的所有文件。 `-r` 标志使命令列出单个文件，而不是仅将它们折叠到根目录名称中。

你还可以包括下面提到的内容，虽然它是可选的，但有助于给面试官留下深刻印象。

输出还将包含一些额外信息，可以通过包含两个标志把它们轻松的屏蔽掉：

git diff-tree –no-commit-id –name-only -r {hash}

这里 `-no-commit-id` 将禁止提交哈希值出现在输出中，而 `-name-only` 只会打印文件名而不是它们的路径。

12、git config 的功能是什么？

git 使用你的用户名将提交与身份相关联。 `git config` 命令可用来更改你的 git 配置，包括你的用户名。

下面用一个例子来解释。

假设你要提供用户名和电子邮件 ID 用来将提交与身份相关联，以便你可以知道是谁进行了特定提交。为此，我将使用：

git config –global user.name "Your Name": 此命令将添加用户名。

git config –global user.email "Your E-mail Address": 此命令将添加电子邮件ID。

13、提交对象包含什么？

Commit 对象包含以下组件，你应该提到以下这三点：

- 一组文件，表示给定时间点的项目状态
- 引用父提交对象
- SHA1 名称，一个40个字符的字符串，提交对象的唯一标识。

14、 如何在Git中创建存储库？

这可能是最常见的问题，答案很简单。

要创建存储库，先为项目创建一个目录（如果该目录不存在），然后运行命令 **git init**。通过运行此命令，将在项目的目录中创建 .git 目录。

15、怎样将 N 次提交压缩成一次提交？

将N个提交压缩到单个提交中有两种方式：

- 如果要从头开始编写新的提交消息，请使用以下命令：

```
git reset –soft HEAD~N &&
git commit
```

- 如果你想在新的提交消息中串联现有的提交消息，那么需要提取这些消息并将它们传给 git commit，可以这样：

```
git reset –soft HEAD~N &&
git commit –edit -m"$(git log –format=%B –reverse .HEAD@{N})"
```

软实力篇

虽说面试和工作是两回事，但是你要想要获得自己满意的 offer，你自身的实力必须要强。

4、必知必会的几点

大部分公司的HR都说我们不看重学历（骗你的！），但是如果你的学校不出众的话，很难在一堆简历中脱颖而出，除非你的简历上有特别的亮点，比如：某某大厂的实习经历、获得了某某大赛的奖等等。

大部分应届生找工作的硬伤是没有工作经验或实习经历，所以如果你是应届生就不要错过秋招和春招。一旦错过，你后面就极大可能会面临社招，这个时候没有工作经验的你可能就会面临各种碰壁，导致找不到一个好的工作

写在简历上的东西一定要慎重，这是面试官大量提问的地方；

将自己的项目经历完美的展示出来非常重要。

5、必须了解的两大法则

STAR法则（Situation Task Action Result）

- **Situation**：事情是在什么情况下发生；
- **Task:**：你是如何明确你的任务的；
- **Action**：针对这样的情况分析，你采用了什么行动方式；
- **Result**：结果怎样，在这样的情况下你学习到了什么。

简而言之，STAR法则，就是一种讲述自己故事的方式，或者说，是一个清晰、条理的作文模板。不管是什么，合理熟练运用此法则，可以轻松的对面试官描述事物的逻辑方式，表现出自己分析阐述问题的清晰性、条理性和逻辑性。

FAB 法则（Feature Advantage Benefit）

- **Feature**：是什么；
- **Advantage**：比别人好在哪些地方；
- **Benefit**：如果雇佣你，招聘方会得到什么好处。

简单来说，这个法则主要是让你的面试官知道你的优势、招了你之后对公司有什么帮助。

6、项目经历怎么写

简历上有一两个项目经历很正常，但是真正能把项目经历很好的展示给面试官的非常少。对于项目经历大家可以考虑从如下几点来写：

1. 对项目整体设计的一个感受
2. 在这个项目中你负责了什么、做了什么、担任了什么角色
3. 从这个项目中你学会了那些东西，使用到了那些技术，学会了那些新技术的使用

者在遇到某一个棘手的问题的时候你是如何解决的又或者说你在这个项目用了什么技术实现了什么功能比如:用redis做缓存提高访问速度和并发量、使用消息队列削峰和降流等等。

7、专业技能怎么写

先问一下你自己会什么，然后看看你意向的公司需要什么。一般HR可能并不太懂技术，所以他在筛选简历的时候可能就盯着你专业技能的关键词来看。对于公司有要求而你不会的技能，你可以花几天时间学习一下，然后在简历上可以写上自己了解这个技能。比如你可以这样写(下面这部分内容摘自我的简历，大家可以根据自己的情况做一些修改和完善)：

- 计算机网络、数据结构、算法、操作系统等课内基础知识：掌握
- Java 基础知识：掌握
- JVM 虚拟机（Java内存区域、虚拟机垃圾算法、虚拟垃圾收集器、JVM内存管理）：掌握
- 高并发、高可用、高性能系统开发：掌握
- Struts2、Spring、Hibernate、Ajax、Mybatis、JQuery：掌握
- SSH 整合、SSM 整合、SOA 架构：掌握
- Dubbo： 掌握
- Zookeeper: 掌握
- 常见消息队列: 掌握
- Linux：掌握
- MySQL常见优化手段：掌握
- Spring Boot +Spring Cloud +Docker:了解
- Hadoop 生态相关技术中的 HDFS、Storm、MapReduce、Hive、Hbase：了解
- Python 基础、一些常见第三方库比如OpenCV、wxpy、wordcloud、matplotlib：熟悉

8、排版注意事项

1. 尽量简洁，不要太花里胡哨；
2. 一些技术名词不要弄错了大小写比如MySQL不要写成mysql，Java不要写成java。这个在我看来还是比较忌讳的，所以一定要注意这个细节；
3. 中文和数字英文之间加上空格的话看起来会舒服一点；

9、其他一些小tips

1. 尽量避免主观表述，少一点语义模糊的形容词，尽量要简洁明了，逻辑结构清晰。
2. 如果自己有博客或者个人技术栈点的话，写上去会为你加分很多。
3. 如果自己的Github比较活跃的话，写上去也会为你加分很多。
4. 注意简历真实性，一定不要写自己不会的东西，或者带有欺骗性的内容
5. 项目经历建议以时间倒序排序，另外项目经历不在于多，而在于有亮点。
6. 如果内容过多的话，不需要非把内容压缩到一页，保持排版干净整洁就可以了。

面试突击手册

目录

部分内容展示

不论是校招还是社招都避免不了各种面试、笔试，如何去准备这些东西就显得格外重要。不论是笔试还是面试都是有章可循的，我这个“有章可循”说的意思只是说应对技术面试是可以提前准备。 我其实特别不喜欢那种临近考试就提前背啊记啊各种题的行为，非常反对！我觉得这种方法特别极端，而且在稍有一点经验的面试官面前是根本没有用的。建议大家还是一步一个脚印踏踏实实地走。

1.1 如何准备面试

1.1.1 如何获取大厂面试机会?

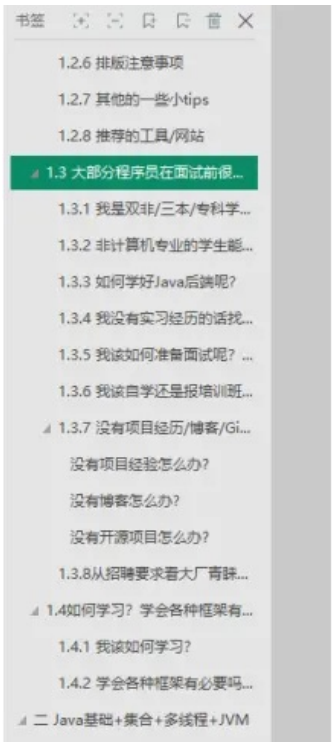
在讲如何获取大厂面试机会之前，先来给大家科普/对比一下两个校招非常常见的概念—春招和秋招。

1. **招聘人数** ： 秋招多于春招 ；
2. **招聘时间** ： 秋招一般7月左右开始，大概一直持续到10月底。但是大厂（如BAT）都会早开始早结束，**所以一定要把握好时间。**春招最佳时间为3月，次佳时间为4月，进入5月基本就不会再有春招了（金三银四）。
3. **应聘难度** ： 秋招略大于春招；
4. **招聘公司**： 秋招数量多，而春招数量较少，一般为秋招的补充。

综上，一般来说，秋招的含金量明显是高于春招的。

下面我就说一下我自己知道的一些方法，不过应该也涵盖了大部分获取面试机会的方法。

1. **关注大厂官网，随时投递简历（走流程的网申）；**
2. **线下参加宣讲会，直接投递简历；**
3. **找到师兄师姐/认识的人，帮忙内推（能够让你避开网申简历筛选、笔试筛选、还是挺不错的。**



1.3 大部分程序员在面试前很关心的一些问题

身边的朋友或者公众号的粉丝很多人都向我询问过:“我是双非/三本/专科学校的，我有机会进入大厂吗？”、“非计算机专业的学生能学好吗？”、“如何学习Java？”、“Java学习该学那些东西？”、“我该如何准备Java面试？”.....这些方面的问题。我会根据自己的一点经验对大部分人关心的这些问题进行答疑解惑。

希望这篇可以给已经在Java方向走了几年的朋友或者正在准备往Java后端方向发展的朋友们一点帮助。道理懂了如果没有实际行动，那这篇文章对你或许没有任何意义。

如果觉得内容不错的话，可以分享给到朋友圈让你的朋友看到，感谢!

1.3.1 我是双非/三本/专科学校的，我有机会进入大厂吗?

我自己也是非985非211学校的，结合自己的经历以及一些朋友的经历，我觉得让我回答这个问题再好不过。

首先，我觉得学校歧视很正常，真的太正常了，如果要抱怨的话，你只能抱怨自己没有进入名校。但是，千万不要动不动说自己学校差，动不动拿自己学校当做自己进不了大厂的借口，学历只是筛选简历的很多标准中的一个而已，如果你够优秀，简历够丰富，你也一样可以和名校同学一起同台竞争。

企业HR肯定是更喜欢高学历的人，毕竟985，211优秀人才比例肯定比普通学校高很多，HR团队肯定会优先在这些学校里选。这就好比相亲，你是愿意在很多优秀的人中选一个优秀的，还是愿意在很多普通的人中选一个优秀的呢？ 双非本科甚至是二本、三本甚至是专科的同学也有很多进入大厂的，不过比率相比于名校的低很多而已。从大厂招聘的结果上看，高学历人才的数量占据大头，那些成功进入

2.1 Java基础

1. 面向对象和面向过程的区别

- **面向过程 ： 面向过程性能比面向对象高。** 因为类调用时需要实例化，开销比较大，比较消耗资源，所以当性能是最重要的考量因素的时候，比如单片机、嵌入式开发、Linux/Unix 等一般采用面向过程开发。但是，**面向过程没有面向对象易维护、易复用、易扩展。**
- **面向对象 ： 面向对象易维护、易复用、易扩展。** 因为面向对象有封装、继承、多态性的特性，所以可以设计出低耦合的系统，使系统更加灵活、更加易于维护。但是，**面向对象性能比面向过程低。**

参见 issue ： 面向过程 ： 面向过程性能比面向对象高？？

> 这个并不是根本原因，面向过程也需要分配内存，计算内存偏移量，Java 性能差的主要原因并不是因为它是面向对象语言，而是 Java 是半编译语言，最终的执行代码并不是可以直接被 CPU 执行的二进制机械码。
>
> 而面向过程语言大多都是直接编译成机械码在电脑上执行，并且其它一些面向过程的脚本语言性能也并不一定比 Java 好。

2. Java 语言有哪些特点?

1. 简单易学；
2. 面向对象（封装，继承，多态）；
3. 平台无关性（ Java 虚拟机实现平台无关性）；

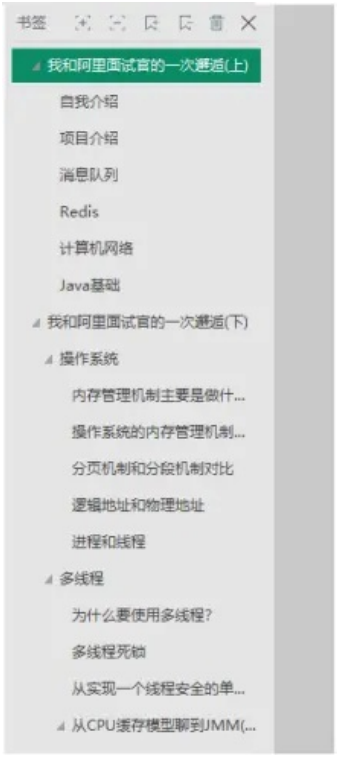
前言

在接触 Java 之前我接触的比较多的是硬件方面，用的比较多的语言就是C和C++。到了大三我才正式选择 Java 方向，到目前为止使用Java到现在大概有一年多的时间，所以Java算不上很好。刚开始投递的时候，实习刚辞职，也没准备笔试面试，很多东西都忘记了。所以，刚开始我并没有直接就投递阿里，毕竟心里还是有一点点小害怕的。于是，我就先投递了几个不算大的公司来练手，就是想着刷刷经验而已或者说是练练手（ps：还是挺对不起那些公司的）。面了一个月其他公司后，我找了我实验室的学长内推我，后面就有了这5次面试。

下面简单的说一下我的这5次面试：4次技术面+1次HR面，希望我的经历能对你有所帮助。

一面(技术面)

1. 自我介绍（主要讲自己会的技术细节，项目经验，经历那些就一语带过，后面面试官会问你的）。
2. 聊聊项目（就是一个很普通的分布式商城，自己做了一些改进），让我画了整个项目的架构图，然后针对项目抛了一系列的提高性能的问题，还问了我做项目的过程中遇到了那些问题，如何解决的，差不读就这些吧。
3. 可能是我前面说了我会数据库优化，然后面试官就开始问索引、事务隔离级别、悲观锁和乐观锁、索引、ACID、MVVC这些问题。
4. 浏览器输入URL发生了什么？TCP和UDP区别？TCP如何保证传输可靠性？
5. 讲下跳表怎么实现的?哈夫曼编码是怎么回事？非递归且不用额外空间（不用栈），如何遍历二叉树
6. 后面又问了很多JVM方面的问题，比如Java内存模型、常见的垃圾回收器、双亲委派模型这些
7. 你有什么问题要问吗?



我和阿里面试官的一次邂逅(上)

本文的内容都是根据读者投稿的真实面试经历改编而来，首次尝试这种风格的文章，花了几天晚上才总算写完，希望对你有帮助。

本文主要涵盖下面的内容：

1. 分布式商城系统：架构图讲解；
2. 消息队列相关：削峰和解耦；
3. Redis 相关：缓存穿透问题的解决；
4. 一些基础问题：
    - 网络相关：1.浏览器输入 URL 发生了什么？2.TCP 和 UDP 区别？3.TCP 如何保证传输可靠性？
    - Java 基础：1. 既然有了字节流,为什么还要有字符流？2.深拷贝 和 浅拷贝有啥区别呢？

下面是正文！

面试开始，坐在我前面的就是这次我的面试官吗？这发量看着根本不像程序员啊？我心里正嘀咕着，只听见面试官说：“小伙，下午好，我今天就是你的面试官，咱们开始面试吧！”。

自我介绍