1. 说一下你们公司平台的架构
2. es用在哪
3. es存储的数据量有多大
4. kafka与redis区别
5. kafka默认缓存时间 7天
6. kakka配置 分区
7. kafka有五个分区 没有指定分区 会写到哪一个分区
8. 如果是读数据的话 会读哪一个分区 会读默认的0
9. kafka数据存在哪
10. redis和kafka队列的区别
11. redis有几种类型 string set zet list hash
12. Redis的数据存在哪 内存
13. set和zset区别
14. 集合有什么特性 确定性 互异性 无序性
15. 用户量，pv数据，存在哪
16. hadoop干嘛的 是一个大数据生态圈 用什么存储数据 HDFS 用来存海量数据的 hive hbase 进行高级应用
17. Hadoop
18. mysql数据量多大
19. mysql最左匹配原则的左是怎么定义的
20. 索引的建立 如果不根据业务 DBA会根据什么来建立索引
21. 索引怎么实现的
22. B+和b-有什么区别 谁查询的更快 为什么 树是怎么构建的
23. TCP/IP协议 三次握手 两个客户端同时建立连接 如何判断 ACK
24. 分布式集群怎么部署的 部署了什么东西
25. PHP工厂模式 一般需要几个基类
26. 什么是观察者模式
27. Laravel和YII和TP有什么区别 event job command
28. PHP7有什么新特性
29. linux命令 只知道文件内内容 如何查找文件位置
30. 更改用户主 chown 更改所有 -R
31. PHP想把一维数组分割成二维数组

**珍岛**

1. **风控是怎么实现的**

风控：现代经济管理学中指控制企业财务损失风险的一种职称（风险控制）

**内部人员管理层**

因打款都是人为操作，难免会产生很多风险；包含操作失误，流程把控上等；

操作失误：可以从培训、验证等方式上减少失误；例如我们就是新上一个功能，都会经过培训，这样就基础层面把人为的操作失误减少啦；同时我们在付款确认时增加账号后4位验证，以确保将打款的对象搞错；

流程把控：可以从权限控制，复核流程，验证等方面减少风险；比如我们在“新增供应商”、“新增收款账号”等，都是只交给财务内部人员才拥有的权限；同时我们在新增付款单、确认付款单、复核付款单等步骤上，都是交给不同的角色来复核，以通过流程上来多步骤减少风险；同时我们在复核是，也是指定的手机号；

内部人员管理层，除去以上2种情况，其实最重要的，还是财务部门的UI内部人员的职业素养的培训；如果人出现问题，那其实很多措施都是无效的摆设而已；

这部分，产品层面会做很多复核流程及复核的设计；

**系统技术层面**

这部分其实更多的是防止黑客攻入，需要技术考虑的比较多；我们都知道黑客现在已经有成套的体系来攻击网站，大致流程如：拖库——洗库——撞库

接下来，我 通过问答的方式将这些

Q：若黑客通过截取服务器接收用户请求、密码、甚至是服务器的私钥怎么办？

A：可增强服务器的防护

Q：若黑客通过SQL注入，修改数据库怎么办呢？

A：写后台程序时，数据库需要加上用户数据进行验证

Q：若黑客暴力破解，怎么办呢？

A：增加密码的长度，同时增加登录密码通过下载本地证书或动态密码

第六种风措施：名单数据（ip名单：比如只限制办公区）

Q：若黑客通过虚拟ip，攻击我方系统怎么办呢？

A：可设置来源IP，只限制财务办公区域ip等白名单

其实，我们还做了登录口令的变换，30分钟进行手动变更一次；

**外部协助层**

此部分，其实是需要商务与合作方进行洽谈，看看是否配合一起做好风控措施的；我们这一次，主要在2各方面，需要合作方进行配合的；一方面是，在支付渠道方维护我们的收款账户，也即支付渠道方只接收我们已维护的收款账户；一方面，也是做ip来源限制，让对方只允许我们财务部门的ip发起的付款申请才能通过；

以上就是我这次关于风控措施想到的措施，当然可能也还有更好的方法，比如设备绑定、数字证书等，但是若按照将以上措施完善的话，风险可以保证是能减少大部分的；

1. **Redis分布式锁**

使用redis命令 set key value NX EX max-lock-time 实现加锁

使用redis命令 EVAL 实现解锁

https://www.cnblogs.com/cmyxn/p/9047848.html

1. **手机消息推送怎么实现的**

https://zhuanlan.zhihu.com/p/19801751

1. **对比几个mvc框架的优缺点TP,Laravel,YII**

**一、ThinkPHP**

ThinkPHP(FCS)是一个轻量级的中型框架，是从Java的Struts结构移植过来的中文PHP开发框架。它使用面向对象的开发结构和MVC模式，并且模拟实现了Struts的标签库，各方面都比较人性化，熟悉J2EE的开发人员相对比较容易上手，适合php框架初学者。 ThinkPHP的宗旨是简化开发、提高效率、易于扩展，其在对数据库的支持方面已经包括MySQL、MSSQL、Sqlite、PgSQL、 Oracle，以及PDO的支持。ThinkPHP有着丰富的文档和示例，框架的兼容性较强，但是其功能有限，因此更适合用于中小项目的开发。

**优点**

1.借助成熟的Java思想  
2.易于上手，有丰富的中文文档；学习成本低，社区活跃度高  
3.框架的兼容性较强，PHP4和PHP5完全兼容、完全支持UTF8等。  
4.适合用于中小项目的开发  
5.从thinkphp3.2.2引入composer包管理工具

**缺点**

1.对Ajax的支持不是很好；  
2.目录结构混乱，相比其他框架目录结构要差一点；  
3.上手容易，但是深入学习较难。

**二、Yii**

Yii 是一个基于组件的高性能php框架，用于开发大型Web应用。Yii采用严格的OOP编写，并有着完善的库引用以及全面的教程。从 MVC，DAO/ActiveRecord，widgets，caching，等级式RBAC，Web服务，到主题化，I18N和L10N，Yii提供了 今日Web 2.0应用开发所需要的几乎一切功能。事实上，Yii是最有效率的PHP框架之一。

**优点**

1.纯OOP  
2.用于大规模Web应用  
3.模型使用方便  
4.开发速度快，运行速度也快。性能优异且功能丰富  
5.使用命令行工具。  
6.支持composer包管理工具

**缺点：**

1.对Model层的指导和考虑较少  
2.文档实例较少  
3.英文太多  
4.要求PHP技术精通，OOP编程要熟练！  
5.View并不是理想view，理想中的view可能只是html代码，不会涉及PHP代码。

**三、laravel**

**优点**

1.laravel的设计思想是很先进的，非常适合应用各种开发模式TDD, DDD 和BDD  
2.支持composer包管理工具  
3.集合了php 比较新的特性，以及各种各样的设计模式，Ioc 容器，依赖注入等。

**缺点**

1.基于组件式的框架，所以比较臃肿

1. **jquery取出标签里面的一段纯文本怎么做**

两种方法：

**（1）正则表达式**

n.content = JSON.parse(JSON.stringify(n.content).replace(/<\/?.+?\/?>/g,""));

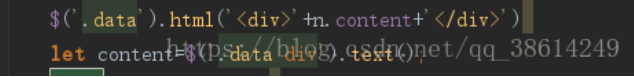
其中：

/<\/?.+?\/?>/g，匹配全局的html标签。

**（2）text（）方法**

由于text（）方法，需要对DOM节点操作。

思路：新建一个html标签，设置为display:none，将拿到的数据插入到新建的html标签中，然后获取该节点调用text（）方法，



代码：

$('.data').html('<div>'+n.content+'</div>')

let content=$('.data div').text();

我自己使用的是正则的匹配，可以减少对DOM节点的操作。

1. 跨域怎么产生的 jsonp原理是什么 两个一样的域名会不会参数跨域 什么样的情况下会产生跨域
2. 单点登录如何实现的
3. mysql连接用的什么连接的 为什么用PDO连接
4. Mysql长连接，短连接是什么以及应用场景
5. 除了PHP其他语言有接触过吗 GO
6. Redis有什么优势
7. 负载均衡原理是什么
8. 如何抵御短时高并发 比如活动期间 服务器
9. 用什么做缓存 redis redis一台扛不住怎么办
10. PHP优化做了哪些东西
11. 已有的一些模块是如何实现 速度的提升优化 比如
12. 服务器安全
13. XSS攻击
14. cookie如何保证安全性
15. Session回收机制
16. PHP垃圾回收机制
17. 关闭浏览器session还存在吗
18. nginx反向代理 代理别的站点时 那个站点通过什么找到反向代理服务器
19. 日志回收怎么处理的
20. Mysql优化
21. 新闻模块需要几张表
22. 大文件上传
23. PHP Curl
24. 做项目中的难点 推荐系统的实现 推荐算法
25. 对996有什么想法
26. 你们团队多少人 你觉得你属于什么水平
27. 有没有排查过线上比较难的bug
28. 技术难点 有挑战的地方
29. 对加班有什么想法 这个行业如果不加班的话 不太现实 最大的承受程度
30. 遇到自己解决不了的问题 怎么办
31. 提前完成了任务 一般会怎么做
32. 测试测出来你的bug多不多