**面试准备的东西**

**一问 ：一个问题，你工作中遇到过的最印象深刻的问题**

1. **数据库有个新同事将 从服务器 中一个数据删除导致数据库服务器主从同步断开**
2. **系统存储空间占满，查出文件后删除空间并未得到释放**

**一解决：问题怎么解决的**

1. **此问题是由于删除数据操作在从服务器，而再去主服务器执行删除命令时，由于从服务器已经没有对应的数据了。主服务器操作生成了binary\_log后，从服务器的I/O thread 复制二进制日志命令后存储到中继日志relay\_log下，sql thread 线程无法在从服务器执行该命令，导致出现类似bug的现象。**

**方法一：断开主从连接后，将主服务器或从服务器中的表格修改到两边一致后再进行主从连接  
方法二：在从服务器上插入回删除对应的数据，并且重新开启 slave，然后回到主表对需要删除的数据进行删除，即可实现主从同步。**

**2. 删除文件后，使用du -sh 查看空间是否被释放，但是存储空间未被释放。使用lsof 筛选出deleted的文件，查看文件依旧被服务占用。使用echo > 将日志文件清空即可**

**一项 ：一个项目，成为你最熟悉的项目**

**公司主页实现高可用负载均衡（主页是静态页面，无需做动静分离，主要诉求为高可用）  
首先我会搭建LNMP基础架构，web服务器我会选择Nginx（支持高并发，轻量级的服务器），数据库我会使用mysql数据库（开源，支持多平台）。**

**其中，整个项目中我搭建了2台web服务器以及2台调度器，在2台调度器上，我选择了haproxy实现负载均衡以及高可用，其中调度算法我选择了轮询（简单，并且使用的服务器的型号一致）（另外一种方案可以使用lvs+keepalived实现负载均衡+高可用）**

**一台：一个服务器多少台的简称，你们公司服务器多少台，都做什么用**

**IDC（互联网数据中心），u是服务器尺寸，1u 2u 4u 服务器，u越大扩展性越好，1u=4.445cm**

**共6台，2台调度服务器，2台web服务器，2台数据库服务器，2台存储服务器**

**一人：公司多少人，项目组多少人，分工  
3人，一人负责web服务器日常维护，一人负责数据库维护，一人负责存储服务器维护以及调度服务器维护**

**一特长：你最擅长什么，技术方面那块 ceph集群部署、LNMP环境部署、SHELL脚本**

**公司主页实现高可用负载均衡（主页是静态页面，无需做动静分离，主要诉求为高可用）  
首先我会搭建LNMP基础架构，web服务器我会选择Nginx（支持高并发，轻量级的服务器），数据库我会使用mysql数据库（开源，支持多平台）。**

**其中，整个项目中我搭建了2台web服务器以及2台调度器，在2台调度器上，我选择了haproxy实现负载均衡以及高可用，其中调度算法我选择了轮询（简单，并且使用的服务器的型号一致）（另外一种方案可以使用lvs+keepalived实现负载均衡+高可用**

**一口：常用端口，必须要记住**

**20/21 ftp ；22 ssh ; 23 telnet ; smtp 25；53 dns ; 80 web; 443 安全网站；3306 数据库； 8080 tomcat**

**一命：基本命令一定要看要敲要会**

**三原理：mysql主从原理、DNS原理、linux启动原理**

**mysql主从： 主服务器开启二进制日志文件功能，执行命令（除查询）的时候生成一个二进制日志，由从服务器的IO线程对二进制日志的内容进行复制到从服务器的中继日志上relay-log ，再由从服务器的sql线程在从服务器上执行**

**Dns原理：客户端首先在访问域名时会查询hosts文件时候由指定域名的ip地址，如果没有，则会根据/etc/resolv.conf文件中指定的DNS server地址访问dns服务器。由dns服务根据地址库文件解析域名得到一个真实的ip地址，客户端再对ip地址进行访问**

**Linux启动原理：  
内核引导：BIOS开机自检，按照bios中设置的启动设备（通常是硬盘）来启动，操作系统接管硬件后首先读入/boot目录下的内核文件**

**运行init：init进程是所有进程的起点，需要读取配置文件/etc/inittab， init进程根据运行级别(7个级别)运行开机启动的程序  
系统初始化：包括激活交换分区，检查磁盘，加载硬件模块以及其他一些优先执行的任务**

**建立终端：进入用户登录系统**

**一监：会一个监控方式，并且怎么做**

**监控我会使用zabbix软件进行监控服务。**

**Zabbix基于lnmp架构提供服务，在安装zabbix前需要部署LNMP。安装zabbix后进行配置，包括不限于指定数据库的登录用户以及密码，设置日志、指定主机。  
启动zabbix服务，zabbix\_server 占用10051端口，zabbix\_agent 占用10050端口**

**一规：你的职业规划**

**一架构：说下你们公司的产品（平台）架构**

**一全：会2-3个安全相关的东西，比如说DDOS攻击  
ddos攻击就是短期内会有大量的访问请求导致服务器资源被占用并且其他用户对服务器的请求响应时间变长。Nginx通过limit\_req\_zone 配置用于限制请求数**

**一脚：背3-5个shell脚本，用于防止面试官给你出脚本问题**

**三优：nginx、apache、tomcat优化**

**三模式：nginx模式、lvs模式、docker的网络**

**ansible 写过什么playbooks**

Playbooks可以对任务进行编排，就像我们要安装一个程序，写个安装shell脚本一样，在哪一步复制配置文件，最后一步启动服务。虽然/usr/bin/ansible 可

以运行一些临时任务，但是针对复杂的配置，并且可以将配置标准化，这个时候就需要Playbooks了

**nginx无缝升级**

**原理：Nginx服务接收到USR2信号后，将Nginx.pid文件添加.oldbin变成nginx.pid.oldbin文件；然后执行新版本Nginx服务器的二进制文件启动服务，如果新的服务启动成功，系统中将有新旧两个Nginx主进程共同提供Web服务，之后，需要向旧的Nginx服务进程发送WINCH信号，使旧的Nginx服务平滑停止，并删除Nginx.pid.oldbin文件。**

**步骤：  
1. /usr/local/nginx/sbin/nginx -v查看版本号，编译配置项；  
2. 备份旧的运行程序；  
3. 编译、make；  
4. 将新版本的运行程序放到程序运行目录下；  
5. Make upgrade;   
6. 向新程序发送启动信号，旧程序发送终结信号.**

nginx 优化

nginx常用优化内容主要包括如下内容：

1. **隐藏版本信息**

**在http{}内添加 server\_tokens off;**

1. **隐藏nginx要修改的源代码**

**vim /usr/src/nginx-1.14.0/src/core/nginx.h #修改软件名和软件版本号**

**vim /usr/src/nginx-1.14.0/src/http/ngx\_http\_header\_filter\_module.c   
#修进入源码文件/src/http/里修改第49行nginx修改为其他服务名**

**vim /usr/src/nginx-1.14.0/src/http/ngx\_http\_special\_response.c**

**##定义对外展示的内容对外展示的NGINX\_VER\_BUILD名字更改为apache**

1. **更改nginx服务的默认用户   
   --user=nginx**
2. **降权启动nginx**
3. **指定Nginx用户为普通用户 A  
   2. Ningx服务部署在A的家目录下  
   3.配置nginx.conf   
   worker\_processes =4; #指定开启最大进程数  
   woker\_cpu\_affinity 0001 0010 0100 1000; #cpu核数+工作模式，几核就是几位  
   woker\_rlimit\_nofile 65535; #进程最大开启文件数**

**Event{}内，  
use epoll #支持大量连接和非活动连接  
accpet\_mutex on #防止惊群现象**

**修改监听端口（不能为80）**

**server{}下指定nginx的根目录**

1. **优化nginx进程个数，且绑定不同的nginx进程到不同的CPU上**

**worker\_processes 4; #cpu核数，高并发可以为cpu的两倍  
worker\_cpu\_affinity 0001 0010 0100 1000;**

1. **Nginx事件处理模型优化  
   nginx事件处理模型有： event{} 下 use epoll;  
   标准方法 select(默认) poll  
   高效方法 kqueue epoll(用于linux)**
2. **调整nginx单个进程允许的客户端最大连接数**
3. **event{}下 worker\_connection = 65535;**

**修改配置/etc/security/limits.conf : #进程最大打开文件数  
 增加： \* soft nofile 65535;   
 \*hard nofile 65535;  
3.修改配置/etc/security/limits.conf : #进程最大打开文件数  
 增加： \* soft nofile 65535;   
 \*hard nofile 65535;**

1. **配置nginx worker进程最大打开文件数**

**1.event{}下 worker\_connection = 65535; worker\_rlimit\_nofile 65535;  
2.修改配置/etc/security/limits.conf : #进程最大打开文件数  
 增加： \* soft nofile 65535;   
 \*hard nofile 65535;**

**3.修改配置/etc/security/limits.conf : #进程最大打开文件数  
 增加： \* soft nofile 65535;   
 \*hard nofile 65535;**

1. **开启高效文件传输模式**

**http{}下插入   
sendfile on ; #开启高效文件传输模式  
tcp\_nopush on; #激活 TCP\_CORK socket 选择；减少网络报文段的数量  
tcp\_nodelay on; #数据传输时不缓存**

1. **Nginx gzip压缩实现性能优化  
   gzip on ; #将配置文件内的这一栏取消注释  
   gzip\_min\_length 1k; #设置允许压缩页面最小字节数  
   gzip\_buffers 4 16k; #设置压缩缓冲区大小，此处设置4个16k内存作为压缩结果缓存  
   gzip\_http\_version 1.1 ; #压缩版本  
   gzip\_comp\_level 2 ; #设置压缩比率，压缩比率越大，压缩速度越慢，9最大，1最小  
   gzip types text/css text/xml application/javascript #指定压缩的类型  
   gzip vary on; #支持vary header ; 可以让前端缓存服务器缓存经过zip压缩的页面**
2. **编写脚本实现日志轮询**

**#!/bin/bash**

**path=/home/nginx**

**time=`date +%Y-%m-%d`**

**mv /home/nginx/logs/access.log /home/nginx/logs/$time\_access.log**

**mv /home/nginx/logs/error.log /home/nginx/logs/$time\_error.log**

1. **不记录不需要的日志  
   在配置文件location ~.\*\.(gif|jpg|png)$ {  
   expires 365d; #缓存365天  
   access\_log off;  
   }**
2. **访问日志的权限设置**

**Chmod a-w -R /home/nginx/logs**

1. **根据扩展名限制程序和文件访问**

**web2.0时代，用户可以向服务器上传一些文件。避免用户上传一些病毒文件或脚本文件。禁止执行用户上传的一些特定文件。  
在fastcgi前面进行设置：**

**location ~^/image/.\*\.(php|php5|.sh|.py|.pl)$ {  
 deny all;**

**}  
location ~^/static/.\*\.(php|php5|.sh|.pl|.py)$ {  
 deny all;  
}**

**location ~^/data/(attachment|avatar)/.\*\.(php|php5)$ {  
 deny all;**

**}**

1. **禁止访问指定目录下的所有文件和目录；且返回404**

**location ~ ^/(static)/ {  
 deny all;  
} #禁止访问单个目录**

**location ~^/static {  
 deny all;  
} #禁止访问多个目录**

**Server{ #访问admin/templates 返回404；  
location /admin/ { return 404; }**

**location /templates/ { return 404; }  
}**

1. **限制网站来源的IP访问  
   location ~^/admin/ {   
   allow 192.168.4.8; #允许4.8访问  
   deny 192.168.4.10; #禁止4.10访问  
   }**
2. **配置nginx禁止非法域名解析访问企业网站**
3. **nginx防爬虫优化**
4. **控制nginx并发连接数量**

**nginx 日志需要怎么进行切割**

**1.定义日志的位置，及存储的位置。**

**2.对日志进行重新命名 移动到指定的位置**

**3.向nginx主进程发信号重新打开日志**

**4.对旧的日志进行删除处理**

**nginx 相对 apache 的优点：**

轻量级，同样起web 服务，比apache 占用更少的内存及资源

抗并发，nginx 处理请求是异步非阻塞的，而apache 则是阻塞型的，在高并发下nginx 能保持低资源低消耗高性能

高度模块化的设计，编写模块相对简单

社区活跃，各种高性能模块出品迅速

**​apache 相对nginx 的优点：**

rewrite ，比nginx 的rewrite 强大

模块超多，基本想到的都可以找到

少bug ，nginx 的bug 相对较多

超稳定

**说明openstack的所有组件内容**

**Horizen(网页界面)、Keynote（认证）、Nova（创建管理虚拟机）、Glance（镜像管理）、**

**Neutron（网络管理）、Swift（对象存储）、Cinder（处理真实物理机与实例间的映射）、  
Heat（编排）、计算（Ceilometer）**

**你了解devops吗**

**tomcat+ansible+jenkins+python+docker+gitlab+k8s+elk+zabbix**

**如何防止服务器被黑**

1.禁止root直接登陆

2.修改ssh远程端口

3.做好控制用户权限

4.修改密码为强密码（大小写字母、数字、特殊符号统统都用上）

5.服务器采用特定加密方式加密密码

6.及时升级相关补丁（中间件、服务器等）

**jenkins（捡肯斯）是什么？CI/CD是什么**

jenkins是开源的，提供友好操作界面的持续集成工具，主要用于持续、自动的构建/测试软件项目

CI （持续集成）是软件开发时间。持续集成强调开发人员提交新代码之后，立刻进行构建、测试

CD（持续交付）是在持续集成的基础上，讲集成后的代码部署更贴近真实运行环境中

**tomcat工作模式**

作为一个servlet（涩符莱特）容器，有三种工作模式： 独立的servlet容器，进程内的servlet容器，进程外的servlet容器

进入tomcat请求可以根据tomcat工作模式分两种：

tomcat作为应用程序服务器：请求来自前端的web服务器，可能是apache，nginx等

tomcat作为独立服务器：请求来自web浏览器

**监控用什么实现**

现在公司业务都在云上，首选云监控，因为自带了ECS（弹性云服务器）RDS（云数据库）等服务的监控模块，可结合自定义报警规则来触发

上家公司业务托管到IDC（互联网数据中心），用的zabbix监控方案，zabbix图形界面丰富，自带很多监控模板，可以多个分区，多个网卡等

自带发现并进行监控，就是需要在客户机安装zabbix agent

（IDC租用一年 1U 7200一年 2U 9500一年 塔式 15000一年）

**说一下用过哪些监控系统**

常见的监控工具有cacti（凯刻太）、nagios（呐给哦死）、zabbix 等，它们各有千秋。

cacti 重图形，有数据历史，需用到数据库支持，支持 web配置，默认不支持告警，可以加插件。cacti 监控网络设备占优势，因为它有rrdtool工具，画

出来的图非常漂亮。

nagios 重状态和结果，没有数据历史，不成图像，不支持 web 配置，可以自己开发脚本定制个性化监控，支持多种插件。没有画图功能，不记录历史数

据。只给出一个状态。不需要存储和mysql。

zabbix 有数据历史，可成图像，支持 web 配置，可以自动发现。

**怎么备份数据，包括数据库备份**

在生成环境下，不管是应用数据，还是数据库数据首先在部署的时候会有主从架构或者集群，这本身就是数据的热备份，其实还考虑冷备份，

门一台服务器做备份服务器，比如可以用resync+inotify配合计划任务实现数据冷备份，发版的包备份，正常情况下有台服务器，每次发版都会保存好发版的包

**LVS负载的原理，nginx负载有啥区别**

LVS提供的负载均衡技术和linux操作系统可实现高性能、高可用的服务器集群，一般LVS都是位于整个集群系统的最前端，由一台或者多台负载调度

器（Director Server）组成，分发给应用服务器（Real Server）,工作在4层（TCP/IP的传输层），LVS基于IP负载均衡技术的IPVS模块实现的，负载

均衡机制有三种： NAT/TUN/DR

NAT：用户到达调度器时，调度器将请求报文的目标地址改成应用服务器菜单地址，报文端口也改成应用服务器端口，最后发送到选定的应用服务器，

服务端得到数据后，返回数据给用户时，还是需要负载调度器改变报文的源地址和源端口改成相应端口，发送用户，然后完成负载调度

TUN： ip隧道技术

它的连接调度和管理NAT一样，只是报文转发方法不一样，调度器采用IP隧道技术将用户请求转发到某个应用服务器，应用服务器直接响应用户的请求

，不在经过前端调度器,(应用服务器没有地域位置要求，可以独立网段，也可以在同一个网段)

DR：直接路由

它的连接调度和管理NAT一样，只是报文转发方法不一样，通过改写报文的MAC地址，请求发送应用服务器，收到后应用服务器直接返回给客户，免去

了通知调度器，但是必须要求是调度器和应用服务器都有一块网卡连在同一个物理网段上

负载调度算法：轮询，加权轮询，最少链接，加权最少链接

LVS优点：抗负载能力强，工作在第四层（传输层），么有流量产生，这个特点决定了在负载均衡软件性能最强；工作稳定，自身有完整的双机热备，如

LVS+keepalived等，应用范围广，可以对所有应用做负载均衡，配置性比较低，减少人为出错几率。

LVS缺点：不支持正则，不能做动静分离，如果网站应用庞大，LVS+Keepalived就复杂，特别是有Windows主机应用机器时，维护和配置会比较麻烦，

相对而言，nginx/haproxy+keepalived就比较简单一点

nginx优点：工作在7层，可以针对应用做分流策略，针对域名，目录结构，它的正则比Haproxy更大强大和灵活，对网络依赖小，理论上ping通就可以

进行负载能力，安装配置简单，测试方便，承担高负载压力且稳定，一般能支撑超哥几万次并发，nginx还是功能强大的web应用服务器，LNMP也是非

常流行的web环境，nginx在处理静态页面和抗高并发方面相对apache有优势，作为web的反向代理加速器缓存越来越成熟。

nginx缺点：仅能支持http和Rmail，这是个弱势，session（塞深）保持和cookie（库肯）引导能力相对欠缺

redis集群原理 redis分片怎么实现，你们公司redis用在哪些环境

其实它的原理不是三两句话能说明白的，redis 3.0版本之前是不支持集群的，官方推荐最大的节点数量为1000，至少需要3(Master)+3(Slave)才能建立

集群，是无中心的分布式存储架构，可以在多个节点之间进行数据共享，解决了Redis高可用、可扩展等问题。

分片部署一般三种：1客户端做分片，确定连接redis实例，然后直接访问相应redis实例

2代理中做分片，客户端不能直接访问redis实例，其工作过程为：客户端先将请求发送给代理，代理通过分片算法确定要访问的是

哪个redis实例，然后将请求发送给相应的redis实例，redis实例将结果返回给代理，代理最后将结果返回给客户端

3在redis服务器端做分片，客户端随机选择一个redis实例发送请求，如果所请求的内容不再当前redis实例中它会负责将请求转交

给正确的redis实例，也有的实现中，redis实例不会转发请求，而是将正确redis的信息发给客户端，由客户端再去向正确的redis实例发送请求

redis用在了哪些环境：java，php环境，主要缓存登录用户信息数据，设备详细数据等

**脚本：**

**统计当前访问IP并排列**

**用awk结合unip、sort过滤access.log日志进行统计**

你会使用哪些虚拟化技术

VMware及kvm，用到较多的vmware虚拟化，基本上生成环境都用vmware，kvm在测试环境中使用，vmware属于原生架构虚拟化技术可以直接硬件上

运行，kvm是寄居架构的虚拟化技术，依托在系统之上运行，管理上，图形管理界面功能很强大，稳定性强，kvm界面稍微差点

**假如有人反应，调取后端接口时特别慢，你会如何排查？**

问清楚反应的人哪个服务应用或者页面调取哪个接口慢，叫他把页面或相关的URL发给你，首先，最直观的分析就是用浏览器按F12，看下是哪一块的内

容过慢（DNS解析、网络加载、大图片、还是某个文件内容等），如果有，就对症下药去解决（图片慢就优化图片、网络慢就查看内网情况等）。其次

，看后端服务的日志，其实大多数的问题看相关日志是最有效分析，最好用tail -f 跟踪一下日志，当然你也要点击测试来访问接口日志才会打出来。最后

，排除sql，，找到sql去mysql执行一下，看看时间是否很久，如果很久，就要优化SQL问题了，expain一下SQL看看索引情况啥的，针对性优化。数据

量太大的能分表就分表，能分库就分库。如果SQL没啥问题，那可能就是写的逻辑代码的问题了，一行行审代码，找到耗时的地方改造，优化逻辑。

**elk中的logstash是怎么收集日志的，在客户端的logstash配置文件主要有哪些内容**

input、output两大块配置；input中指定日志（type、path）等，output指定日志输出的目标（host、port）等。

给你一套环境，你会如何设计高可用、高并发的架构？

如果这套环境是部署在云端(比如阿里云)，你就不用去考虑硬件设计的问题。可直接上阿里云的SLB+ECS+RDS（负载均衡服务器+弹性云服务器+云

数据库）这套标准的高可用、高并发的架构。对外服务直接上SLB负载均衡技术，由阿里的SLB分发到后端的ECS主机；ECS主机部署多台，应用拆分在

不同的ECS主机上，尽量细分服务。数据库用RDS高可用版本（一主一备的经典高可用架构）、或者用RDS金融版（一主两备的三节点架构）。在结合

阿里其它的服务就完全OK，业务量上来了，主机不够用了，直横向扩容ECS主机搞定。（华为云是ELB负载均衡）

如果这套环境托管在IDC，那么你就要从硬件、软件（应用服务）双面去考虑了。硬件要达到高可用、高并发公司必须买多套网络硬件设备（比如负载设

备F5、防火墙、核心层交换、接入层交换）都必须要冗余，由其是在网络设计上，设备之间都必须有双线连接。设备如果都是跑的单机，其中一个设备

挂了，你整个网络都瘫痪了，就谈不上高可用、高并发了。其次在是考虑应用服务了，对外服务我会采用成熟的开源方案LVS+Keepalived或者Nginx+

Keepalived，缓存层可以考虑redis集群及Mongodb集群，中间件等其它服务可以用kafka、zookeeper，图片存储可以用fastDFS或MFS，如果数据量

大、又非常多，那么可采用hadoop这一套方案。后端数据库可采用 “主从+MHA”。这样一套环境下来是绝对满足高可用、高并发的架构。

TCP/IP原理说一下？TCP有哪几个状态，分别是什么意思

以tcp/ip协议为核心,分五层。tcp工作在第4层，主要有tcp和udp协议。其中tcp是可靠协议，udp是不可靠协议。 tcp传输之前，需要建立连接，通过三次

握手实现

TCP三次握手 A建立连接发送syn包等待B确认 B收到A的syn包确认A的syn和ack在发送自己的syn A收到B的syn和ack后发送自己ack给B

TCP 四次断开

客户端A发送一个FIN，用来关闭客户A到服务器B的数据传送。

服务器B收到这个FIN，它发回一个ACK，确认序号为收到的序号加1。

服务器B关闭与客户端A的连接，发送一个FIN给客户端A。

客户端A发回ACK报文确认，并将确认序号设置为收到序号加1

两者区别：

1） TCP提供面向连接的传输，通信前要先建立连接（三次握手机制）；UDP提供无连接的传输，通信前不需要建立连接。

2） TCP提供可靠的传输（有序，无差错，不丢失，不重复）； UDP提供不可靠的传输。

3） TCP面向字节流的传输，因此它能将信息分割成组，并在接收端将其重组； UDP是面向数据报的传输，没有分组开销。

4） TCP提供拥塞控制和流量控制机制； UDP不提供拥塞控制和流量控制机制。

有个客户说访问不到你们的网站，但是你们自己测试内网和外网访问都没问题。你会怎么排查并解决客户的问题？

我们自己测了都没问题，只是这个客户访问有问题，那肯定是要先联系到这个客户，能远程最好，问一下客户的网络是不是正常的，访问其它的网站有没有

问题（比如京东、百度什么的）。如果访问其它网站有问题，那叫客户解决本身网络问题。如果访问其它网站都没问题，用ping和nslookup解析一下我们

的网站是不是正常的，让客户用IP来访问我们的网站是否可行，如果IP访问没问题，那就是客户的DNS服务器有问题或者DNS服务器解析不到我们的网站。

还有一种可能就是跨运营商访问的问题，比如我们的服务器用的是北方联通、而客户用的是南方移动，就也有可能突然在某个时间段访问不到，这种情况在

庞大的中国网络环境中经常发生（一般是靠CDN解决）。还有可能就是我们的网站没有SSL证书，在公网是使用的是http协议，这种情况有可能就是没有用

https协议网站被运营商劫持了。

你会用什么方法查看某个应用服务的流量使用情况

如果是单一应用的服务器，只需要用iftop、sar等工具统计网卡流量就可以。如果服务器跑了多个应用，可以使用nethogs（耐特号歌死）工具实现，它的特

别之处在于可以显示每个进程的带宽占用情况，这样可以更直观获取网络使用情况（nethogs是一个全面的类似top的命令，用来按进程或程序实时统计网络

带宽使用率）

系统信息查看基础命令

uname -a # 查看主机名/内核/操作系统/CPU信息 -r 查看系统版本

head -n 1 /etc/issue # 查看操作系统版本

cat /proc/cpuinfo # 查看CPU信息 ?怎么查看有多少CPU

cat /proc/cpuinfo |grep "physical id"|sort -u |wc -l

getconf LONG\_BIT --查看CPU多少位

hostname # 查看计算机名

lspci -tv # 列出所有PCI设备

lsusb -tv # 列出所有USB设备

lsmod # 列出加载的内核模块 modprobe--加载模块

env # 查看环境变量

set # 查看环境变量

系统资源查看基础命令

free -m # 查看内存使用量和交换区使用量 '-m':单位Mb

cat /proc/meminfo #查看内存相关信息

grep MemTotal /proc/meminfo # 查看内存总量

grep MemFree /proc/meminfo # 查看空闲内存量

du -sh <目录名> # 查看指定目录的大小

--查看系统负载

uptime # 查看系统运行时间、用户数、负载

tload #显示系统负载

cat /proc/loadavg # 查看系统负载

-查看磁盘

df -h # 查看各分区使用情况

fdisk -l # 查看磁盘空间

磁盘和分区

mount | column -t # 查看挂接的分区状态 'column -t'使个字段对齐

mount -o loop rhel.iso /mnt #?'-o loop'参数

fdisk -l # 查看所有分区

swapon -s # 查看所有交换分区

hdparm -i /dev/hda # 查看磁盘参数(仅适用于IDE设备)

dmesg | grep IDE # 查看启动时IDE设备检测状况

网络

ifconfig # 查看所有网络接口的属性

iptables -L # 查看防火墙设置 service iptables status

route -n # 查看路由表

netstat -lntp # 查看所有监听端口

netstat -antp # 查看所有已经建立的连接

netstat -s # 查看网络统计信息

netstat -nat | grep -i "80" |wc -l #查看80端口连接数

lsof -i :port，使用lsof -i :port就能看见所指定端口运行的程序，同时还有当前连接。

进程

ps -ef # 查看所有进程

top # 实时显示进程状态

ps -ef | grep httpd |wc -l #显示web服务进程数

ps -u root -N #显示非root用户身份运行的进程

ps -C httpd -o pid= #显示httpd相关的进程号。一般杀死主进程号即可

用户

w # 查看活动用户

id <用户名> # 查看指定用户信息

last # 查看用户登录日志

cut -d: -f1 /etc/passwd # 查看系统所有用户

cut -d: -f1 /etc/group # 查看系统所有组

crontab -l # 查看当前用户的计划任务 -e 编辑 -u 用户的crond -r删除

\*全值`每` -范围 ,指定的 /5 每隔5 #注释掉

tty # 查看当前终端

ac -p # 查看每个用户的连接时间

ac -d # 查看用户每天的连接时间

lastlog # 查看每个用户最后登录时间

finger zhang # 查看用户的详细信息

pkill -kill -t pts/12 #踢掉用户

服务

chkconfig --list # 列出所有系统服务

chkconfig --list telnet

chkconfig --list | grep on # 列出所有启动的系统服务

程序

rpm -qa |grep httpd # 查看所有安装的软件包

rpm -ivh RealPlayer10 GOLD.rpm

rpm -evh --卸载软件包

rpm -qp

rpm -qf `which df` --查看df命令由那个包安装生成的 `which df`中'`'符号是1左边的

rpm -qf /etc/dhcpd.conf --查看文件是由哪个包安装生成的

源码安装

cd httpd-2.2.9

./configure --prefix=/opt/apache2.2.9

make

make install

/ /网络设置

hostname

lspci --列出所有PCI设备，确定机器有一块以太网卡。

vi /etc/sysconfig/network --修改主机名

vi /etc/sysconfig/network-scripts/ifcfg-eth0 --网卡配置 动态IP

/ /开启linux代理转发功能

# echo“1”> /proc/sys/net/ipv4/ip\_forward

或者

vi /etc/sysctl.conf

net.ipv4.ip\_forward=0

sysctl -p --使修改马上生效

//网络服务重启

service network reload/restart

$ DNS配置

cat /etc/hosts --将主机名和IP地址进行匹配的简单数据库

cat /etc/resolv.conf --指定域名服务器和默认的DNS搜索域

ansible你用过它的哪些模块，ansbile同时分发多台服务器的过程很慢（它是逐台分发的），你想过怎么解决吗？

用过ansible的（copy file yum ping command shell）等模块；ansible默认只会创建5个进程,所以一次任务只能同时控制5台机器执行.那如果你有大量的

机器需要控制,或者你希望减少进程数,那你可以采取异步执行.ansible的模块可以把task放进后台,然后轮询它.这使得在一定进程数下能让大量需要的机器同时

运作起来.

nginx有哪几种调度算法，解释一下ip hash和轮询有啥不一样

常用的有3种调度算法（轮询、ip hash、权重）。

轮询：upstream按照轮询（默认）方式进行负载，每个请求按时间顺序逐一分配到不同的后端服务器，如果后端服务器down掉，能自动剔除。

ip hash：每个请求按访问ip的hash结果分配，这样每个访客固定访问一个后端服务器，可以解决session的问题。

权重：指定轮询几率，权重（weight）和访问比率成正比，用于后端服务器性能不均的情况。

nginx你用到了哪些模块，在proxy模块中你配置过哪些参数

用到过（负载均衡upstream、反向代理proxy\_pass、location、rewrite等）。

proxy模块中配置过:proxy\_set\_header、proxy\_connect\_timeout、proxy\_send\_timeout、proxy\_buffer\_\*

说一下iptables的原理，有哪些表、哪些链？怎么修改默认策略全部为DROP

iptables是工作在TCP/IP的2、3、4层。你要说它的原理也不是几话能概括的，当主机收到一个数据包后，数据包先在内核空间中处理，若发现目的地址是

自身，则传到用户空间中交给对应的应用程序处理，若发现目的不是自身，则会将包丢弃或进行转发。

4张表（raw表、mangle表、net表、filter表）

5条链（INPUT链、OUTPUT链、PORWARD链、PREROUTING链、POSTROUTING链）。

如何开启linux服务器路由转发功能？

echo "1" > /proc/sys/net/ipv4/ip\_forward

你在shell脚本中用过哪些语法，case语法会用到哪些地方？

一般会用到if语句、for语句、while语句、case语句以及function函数的定义；case语句为多选择语句，可以用case语句匹配一个值与一个模式，

如果匹配成功，执行相匹配的命令。最典型的case语法会用到启动服务脚本的处理

linux系统中你会用到什么命令查看硬件使用状态信息？

lscpu(查看cpu信息)、free -m（查看内存信息）、df -h（查看硬盘分区信息）、top（还可以动态查看cpu、内存使用情况的信息），/proc/目录下也可以

查看很多硬件信息

zabbix如何修改其中监控的一台服务器中内存阈值信息，比如正常内存使用到了80%报警，我想修改为60%报警？

正常来说，一般会把监控的服务器统一加入到一个模板中，修改模板的其是某一项的监控项参数和告警阈值后，加入模板中的所有主机都会同步。如果单独想

修改其中某一台服务器内存告警阈值，需要进入这台主机，单独创建一个告警Triggers，关联这台主机监控内存的项，配置好告警的阈值为60%即可实现。其

实，zabbix一切都为图形化操作

mysql主从复制原理说一下？

(1) master将改变记录到二进制日志(binary log)中（这些记录叫做二进制日志事件，binary log events）；

(2) slave将master的binary log events拷贝到它的中继日志(relay log)；

(3) slave重做中继日志中的事件，将改变反映它自己的数据。

办公室同事反馈网站打不开，请简述一下排查故障的思路或解决方法

先ping一下域名，看看网络是否通

Telnet 一下服务器的80端口，看是否通

查看服务器流量是否正常，借助sar或者nload

查看服务器是否有负载，用w，查看服务是否开启，用ps

抓包查看数据包是否正常，看是否有攻击，用tcpdump

查看配置是否正确

rsync+inotify是实现文件实时同步的，加什么参数才能实现实时同步，--delete参数又是什么意思？

rsync是远程同步工具、inotify是一种强大的异步文件系统系统监控机制。通过inotifywait 中的-m参数可以实现“始终保持事件监听状态”。rsync

中的-delete参数是指“ 删除那些DST中SRC没有的文件”。

简述DNS进行域名解析的过程？

用户要访问http://www.baidu.com，会先找本机的host文件，再找本地设置的DNS服务器，如果也没有的话，就去网络中找根服务器，根服务器反馈结

果，说只能提供一级域名服务器.cn，就去找一级域名服务器，一级域名服务器说只能提供二级域名服务器.com.cn,就去找二级域名服务器，二级域服务器只能

提供三级域名服务器.http://baidu.com.cn，就去找三级域名服务器，三级域名服务器正好有这个网站http://www.baidu.com，然后发给请求的服务器，保

存一份之后，再发给客户端

你常用的Nginx模块，用来做什么

rewrite模块，实现重写功能

access模块：来源控制

ssl模块：安全加密

ngx\_http\_gzip\_module：网络传输压缩模块

ngx\_http\_proxy\_module 模块实现代理

ngx\_http\_upstream\_module模块实现定义后端服务器列表

ngx\_cache\_purge实现缓存清除功能

如何优化 Linux系统（可以不说太具体）

不用root，添加普通用户，通过sudo授权管理

更改默认的远程连接SSH服务端口及禁止root用户远程连接

定时自动更新服务器时间

配置国内yum源

关闭selinux及iptables（iptables工作场景如果有外网IP一定要打开，高并发除外）

调整文件描述符的数量

精简开机启动服务（crond rsyslog network sshd）

内核参数优化（/etc/sysctl.conf）

更改字符集，支持中文，但建议还是用英文字符集，防止乱码

锁定关键系统文件

清空/etc/issue，去除系统及内核版本登录前的屏幕显示

如何对一台Linux服务器进行系统性能调优，列举出参数

主机的存活状态、系统负载值、CPU的使用率、磁盘IO量，swap进出量，网络流量，数据库的主从复制状态、LVS+keepalived的VIP

RAID0，1，5，0+1的含义

raid0 就是把多个（最少2个）硬盘合并成1个逻辑盘使用，数据读写时对各硬盘同时操作，不同硬盘写入不同数据，速度快。

raid1就是同时对2个硬盘读写（同样的数据）。强调数据的安全性。比较浪费。

raid5也是把多个（最少3个）硬盘合并成1个逻辑盘使用，数据读写时会建立奇偶校验信息，并且奇偶校验信息和相对应的数据分别存储于不同的磁盘上。

当RAID5的一个磁盘数据发生损坏后，利用剩下的数据和相应的奇偶校验信息去恢复被损坏的数据。相当于raid0和raid1的综合。

raid10就是raid1+raid0，比较适合速度要求高，又要完全容错，最少需要4块硬盘（注意：做raid10时要先作RAID1，再把数个RAID1做成RAID0，这样

比先做raid0，再做raid1有更高的可靠性）

1.中间件类

中间件名称 默认端口

tomcat 8080

wildfly 8080

jetty 8080

jboss 8080

nginx 80

apache 80

weblogic 7001(管理节点)/7011（从节点）

2.数据库类

mysql 3306

MariaDB 3306

oracle 1521

SQL Server 1443

PostgreSQL 5432

Sybase 5000

MongoDB 27017

Redis 6379

Memcached 11211

3.ELK

ElasticSearch 9200

Kibana 5601

Logstash 5044

4.存储类

fastdfs

Tracker（跟踪器） 22122

Storage（存储节点） 23000

5.监控软件类

zabbix\_server 10051

zabbix\_agentd 10050

6.linux服务类

ftp 20/21

sftp 115

ssh 22

telnet 23

smtp(邮件传输协议) 25

pop3 (邮件传输协议) 110

dns 53

http 80

https 443

ntp 123

VNC 5900

7.hadoop

组件 角色 端口作用 默认端口

HDFS NameNode NameNode 端口 8020

HDFS NameNode NameNode Web UI 端口 50070

HDFS DataNode DataNode 协议端口 50020

HDFS DataNode DataNode 收发器端口 1004

HDFS DataNode DataNode HTTP Web UI 端口 1006

HBase HMaster HBase Master 端口 60000

HBase HMaster HBase Master Web UI 端口 60010

HBase RegionServer HBase Region Server 端口 60020

HBase RegionServer HBase Region Server Web UI 端口 60030

Zookeeper zookeeper Server 客户端端口 2181

yarn ResourceManager ResourceManager Web 应用程序 HTTP 端口 8088

yarn NodeManager NodeManager Web 应用程序 HTTP 端口 8042

yarn JobHistory MapReduce JobHistory Web 应用程序 HTTP 端口 19888

hive HiveServer2 HiveServer2 端口 10000

kafka kafka broker TCP Port 9092

kafka kafka broker JMX Port 9393

impala Impala Daemon Impala Daemon HiveServer2 端口 21050

impala Impala Daemon Impala Daemon HTTP 服务器端口 25000

impala Catalog Server Catalog Server HTTP 服务器端口 25020

impala StateStore StateStore HTTP 服务器端口 25010

solr Solr Server Solr HTTP 端口 8983

solr Solr Server Solr 管理端口 8984

spark History Server History Server WebUI Port 18088

基础平台管理控制台 cloudera-scm-server Admin Console 的 HTTP 端口 7180

基础平台管理控制台 cloudera-scm-server Admin Console 的 HTTPS端口 7183

mysql 用了什么引擎

Innodb引擎

MyIASM引擎

主从同步，主服务器宕机怎么处理

可用keepalived做主从，主宕机可以切到优先级高的服务器，让优先级高的服务器做主。

mysql 优化做了哪些

MySQL内核优化

MySQL配置参数优化，此优化需要进行压力测试来进行参数调整

对语句以及表优化

Linux系统中病毒怎么解决

1.最简单有效的方法就是重装系统（不过一般要小心谨慎而行）

2.查找病毒内容并处理

中毒后服务器的现象：

（1）cpu、内存使用率会炒鸡高；

（2）网络带宽存在异常，多表象为：机器向外发包等异常情况，网络带宽占用高于正常时期；

（3）僵尸进程

查找内容：

（1）异常进程

（2）僵尸进程

（3）异常文件

（4）异常脚本

（5）异常定时任务

（6）其他

一般在实际工作中，即使删除病毒文件也不排除会有有潜伏病毒及激活病毒的情况，所以建议最好是把机器中的重要数据备份之后重装系统。

FTP的主动模式和被动模式

FTP协议有两种工作方式：PORT方式和PASV方式，中文意思为主动式和被动式。

PORT（主动）方式的连接过程是：客户端向服务器的FTP端口（默认是21）发送连接请 求，服务器接受连接，建立一条命令链路。

当需要传送数据时，客户端在命令链路上用PORT 命令告诉服务器：“我打开了XX端口，你过来连接我”。于是服务器从20端口向

客户端的 XX端口发送连接请求，建立一条数据链路来传送数据。

PASV（被动）方式的连接过程是：客户端向服务器的FTP端口（默认是21）发送连接请 求，服务器接受连接，建立一条命令链路。

当需要传送数据时，服务器在命令链路上用PASV 命令告诉客户端：“我打开了XX端口，你过来连接我”。于是客户端向服务器的

XX端口 发送连接请求，建立一条数据链路来传送数据。

从上面可以看出，两种方式的命令链路连接方法是一样的，而数据链路的建立方法就完 全不同。

ansible中的常用模块及作用

ping # 主机连通性测试

command # 在远程主机上执行命令,不支持管道

shell # 在远程主机上调用shell解析器,支持管道命令个

copy # 用于将文件复制到远程主机,支持设定内容和修改权限.

file # 创建文件,创建连接文件,删除文件等

fetch # 从远程复制文件到本地

cron # 管理cron计划任务

yum # 用于模块的安装

service # 管理服务

user # 管理用户账号

group # 用户组管理

script # 将本地的脚本在远端服务器运行

setup # 该模块主要用于收集信息，是通过调用facts组件来实现的,以变量形式存储主机上的信息

有一web服务器，某天某用户投诉上某个网站速度很慢，如果你是服务器的管理员，请问

你如何查找原因。假设服务器有任意操作系统和软件。

1、首先检测服务器状态，排除硬件故障和系统故障 排除机器故障

2、检查底层网络状态（包括互联情况，PING包延迟、流量情况）排除网络故障和网卡等设备故障。

3、检查WEB服务器软件运行情况（并发连接、服务软件配置情况、数据库连接情况等）排除应用服务故障。

**什么叫网站灰度发布？**

**灰度发布是指在黑与白之间，能够平滑过渡的一种发布方式**

**AB test就是一种灰度发布方式，让一部用户继续用A，一部分用户开始用B**

**如果用户对B没有什么反对意见，那么逐步扩大范围，把所有用户都迁移到B上面 来**

**灰度发布可以保证整体系统的稳定，在初始灰度的时候就可以发现、调整问题，以保证其影响度**

简述DNS进行域名解析的过程？

用户要访问www.baidu.com，会先找本机的host文件，再找本地设置的DNS服务器，如果也没有的话，就去网络中找根服务器，根服务器

反馈结果，说只能提供一级域名服务器.cn，就去找一级域名服务器，一级域名服务器说只能提供二级域名服务器.com.cn,就去找二级域名服

务器，二级域服务器只能提供三级域名服务器.baidu.com.cn，就去找三级域名服务器，三级域名服务器正好有这个网站www.baidu.com

，然后发给请求的服务器，保存一份之后，再发给客户端

讲一下Keepalived的工作原理？

在一个虚拟路由器中，只有作为MASTER的VRRP路由器会一直发送VRRP通告信息,

BACKUP不会抢占MASTER，除非它的优先级更高。当MASTER不可用时(BACKUP收不到通告信息)

多台BACKUP中优先级最高的这台会被抢占为MASTER。这种抢占是非常快速的(<1s)，以保证服务的连续性

由于安全性考虑，VRRP包使用了加密协议进行加密。BACKUP不会发送通告信息，只会接收通告信息

mysql的innodb如何定位锁问题，mysql如何减少主从复制延迟？

mysql的innodb如何定位锁问题:

在使用 show engine innodb status检查引擎状态时，发现了死锁问题

在5.5中，information\_schema 库中增加了三个关于锁的表（MEMORY引擎）

innodb\_trx ## 当前运行的所有事务

innodb\_locks ## 当前出现的锁

innodb\_lock\_waits ## 锁等待的对应关系

mysql如何减少主从复制延迟:

如果延迟比较大，就先确认以下几个因素：

1. 从库硬件比主库差，导致复制延迟

2. 主从复制单线程，如果主库写并发太大，来不及传送到从库就会导致延迟。更高版本的mysql可以支持多线程复制

3. 慢SQL语句过多

4. 网络延迟

5. master负载

主库读写压力大，导致复制延迟，架构的前端要加buffer及缓存层

6. slave负载

一般的做法是，使用多台slave来分摊读请求，再从这些slave中取一台专用的服务器

你常用的Nginx模块，用来做什么

rewrite模块，实现重写功能

access模块：来源控制

ssl模块：安全加密

ngx\_http\_gzip\_module：网络传输压缩模块

ngx\_http\_proxy\_module 模块实现代理

ngx\_http\_upstream\_module模块实现定义后端服务器列表

ngx\_cache\_purge实现缓存清除功能

如何优化 Linux系统（可以不说太具体）？

不用root，添加普通用户，通过sudo授权管理

更改默认的远程连接SSH服务端口及禁止root用户远程连接

定时自动更新服务器时间

配置国内yum源

关闭selinux及iptables（iptables工作场景如果有外网IP一定要打开，高并发除外）

调整文件描述符的数量

精简开机启动服务（crond rsyslog network sshd）

内核参数优化（/etc/sysctl.conf）

更改字符集，支持中文，但建议还是用英文字符集，防止乱码

锁定关键系统文件

清空/etc/issue，去除系统及内核版本登录前的屏幕显示

redis基本部署

常用的 ssentinel（哨兵模式）

官方推荐高可用方案

当用Redis做Master-slave的高可用方案时，假如master宕机了，Redis本身(包括它的很多客户端)都没有实现自动进行主备切换，而Redis-sentine

本身也是一个独立运行的进程，它能监控多个master-slave集群，发现master宕机后能进行切换

sentinel哨兵如下功能实现

1.监控（monitoring）：监控redis情况

2.通知（notification）：通知错误信息

3.故障转移（failover）：当某个master死掉，选择另外一个slave升级为master,更 新master-slave关系

4.提供服务商（configurationprovider）：client通过sentinel获取redis地址，并在failover时更新地址（客户通过哨兵获取redis地址，故障转移时更新）

Redis主从复制模式和Redis哨兵模式采用的都是复制Master节点的数据，实现读写分离

但是这种设计存在一个严重的问题，它没有真正意义上实现数据分片。两个模式都有一个问题，不能水平扩容，并且这两个模式的高可用特性都会受到

Master主节点内存的限制。Redis较难支持在线扩容，在集群容量达到上限时在线扩容会变得很复杂。所有Redis集群模式就是用来解决这个问题

Redis Cluster集群模式（外地死 夸死特）

3.0版本开始正式提供

结构特点

所有的redis节点彼此互联(PING-PONG机制),内部使用二进制协议优化传输速度和带宽。

节点的fail是通过集群中超过半数的节点检测失效时才生效。

客户端与redis节点直连,不需要中间proxy层.客户端不需要连接集群所有节点,连接集群中任何一个可用节点即可。

redis-cluster把所有的物理节点映射到[0-16383]slot上（不一定是平均分配）,cluster 负责维护node<->slot<->value。

Redis集群预分好16384个桶，当需要在 Redis 集群中放置一个 key-value 时，根据 CRC16(key) mod 16384的值，决定将一个key放到哪个桶中。

投票过程是集群中所有master参与,如果半数以上master节点与master节点通信超时(cluster-node-timeout),认为当前master节点挂掉

项目遇到问题：

自定义Zabbix监控Mysql性能 , zabbix-3.4.4 Agent没有内置MySQL的监控功能。

解决方案：

监控MySQL 是否还活着，自定义的key ，命令mysql -V 。建立mysql检查脚本：自定义数据库连接用户IP地址端口号以及检查的的命令存储于变量里，在使用case语句位置传参，

下面是脚本内容：

1.查询当前MySQL 本次启动后的运行统计时间

2.查看当前慢查询语句的个数

3.执行回滚的个数

4.发送的字节数

5.查看 当前打开的表数量

6.查看线程缓存内的线程数量

7.查看select语句的执行数

8.查看insert语句的执行数

9.查看update 语句的 执行数

10.查看delete语句的执行数

11.查看试图连接到MySQL(不管是否连接成功)的连接数

12.查看立即获得的表的锁的次数

13.查看不能立即获得的表的锁的次数

14.查看创建用来处理连接的线程数。

项目遇到问题：

一、配置Haproxy的过程中遇到的问题，对于页面分离请求不够明确。

二、编写rsync+inotify脚本，用Ansible脚本模快去执行实时同步，当Ansible使用ssh登陆执行时会产生一个ssh的进程， 由于此脚本必须是放入后台执行的，Ansible执行完任务之后退出被管理机,ssh下面所有的子进程都会退出，放入后台的脚本就无法执行 了。

三、Keepalived脑裂(裂脑)，Keepalived的 BACKUP主机在收到不 MASTER 主机报文后就会切换成为master，如果是它们 之间的通信线路出现问题，无法接收到彼此的组播通知，但是两个节点实际都处于正常工作状态，这时两个节点均为master强行绑定虚 拟IP，导致不可预料的后果。

解决方案：

一，分离页面配置：frontend内，定义静态检查标，定义动态检查标，为主页专门定制，当请求的内容是php内容时，将请 求转交给php动态资源服务器的acl规则， 当请求的内容是静态内容时（图片、视频、js、css、html），将请求转交给静态资源服务 器的acl规则。

二，rsync+inotify脚本：1，使用Ansible自带的模块rsync此模板必有安装rsync包才可以使用。使用Ansible 自带的模块 at在被管理机，定时去执行此脚本。

三，Keepalived脑裂(裂脑)：做好对裂脑的监控报警，在问题发生时第一时间发送报警消息到管理员手机上