

目录

一、简单型 .....	9
1. 你平时在公司主要做什么? .....	9
2. 你们原来公司的网站架构是怎么样的? .....	9
3. 你对哪一块比较熟练或者精通? .....	9
4. 介绍一下负载均衡? .....	9
5. lvs 内部原理? .....	9
6. nginx lvs haproxy 三个有什么区别 .....	9
7. lvs 主要 3 种工作模式原理 .....	9
8. lvs 工作原理, 算法, 工作模式? .....	10
9. mysql 用的哪个版本, 你们的数据库怎么备份, 什么时间备份, 备份数据量多大 .....	10
10. memcached 工作原理 (内存管理机制) .....	10
11. nginx 状态码 499 是什么意思 .....	11
12. nginx 状态码 502 错误可能的原因有哪些? .....	11
13. 状态码 200, 500 502 304 , 分别表示什么意思 .....	11
14. 维护网站过程中遇到的问题, 处理过什么故障? (状态码: 413、 504 的意思) .....	11
15. FTP 主动和被动的区别 (指的是服务器端的主动和被动)? .....	11

16. apache 两种工作模式区别及优化? .....	11
17. mysql 主从原理? 如果主库不同步报错了怎么恢复? .....	11
18. 如何备份大数据 mysql 数据文件, mysql 都优化哪些? .....	12
19. 你在上家公司用的什么监控软件 (zabbix)? 都监控些什么? 怎么增加模板? .....	12
20. nagios, cacti 维护过吗, 平时都监控什么? .....	12
21. 一台 web 服务器应该监控哪些指标.....	12
22. zabbix 监控数据库监控哪些参数.....	12
23. zabbix 有哪些监控项.....	12
24. 监控一台服务器时, 你觉得应该监控服务器的哪些资源 .....	12
25. 你们监控的时候日志平台会监控吗? .....	12
26. 你们会监控数据库的性能吗, 会定标准吗? 有实际遇到过吗, 如何解决? .....	13
27. 写 shell 脚本监控 mysql 主从的思路 .....	13
28. 写 shell 脚本监控所有 IP 是否存活的思路 .....	13
29. 公司出口带宽? 每天网站 pv? uv? .....	13
30. 你们公司的网站出口带宽是多少? 网站访问量是多少? 是怎么计算的? .....	13
31. 你们公司是什么架构? 都有哪些网上业务? 详细说一下? .....	13
32. 你们公司备份策略是什么? 数据量多大? 备份周期? 备份工具? 备份方式? 是否做数据恢复备份演练? .....	13

33. 你在工作中是怎么做日志分析的？例如：访问日志、错误日志等 .....	13
34. 你们公司日志多久清理一次？ .....	13
35. 你们公司运维部门有多少人？多少服务器？基本优化项目？ .....	13
36. 你在上家公司针对防攻击、扩容、容灾等问题是怎么制定和优化解决方案的？ .....	13
37. 你们公司如何自动化代码上线？ .....	13
38. 你们公司都有哪些应用集群？ .....	13
39. 你们公司用的是物理服务器还是云服务？ .....	13
40. 你们公司安全方面怎么我做的？监控体系结构是怎样的？ .....	14
41. 如何查看 Linux 系统当前的状态，cpu 内存的使用情况及负载 .....	14
42. 写出你所了解的门户网站的服务架构，可用什么方式实现的高可用、负载均衡？ .....	14
43. nginx 日志的路径为 opt/logs/access. logs, 假设日志不能自动分割, 写脚本让日志每天能够定时自动分割 .....	14
44. 写一个脚本，查找 15 天前以 png 结尾的文件并删除 .....	14
45. 你们为什么使用 redis？ .....	14
46. redis 有哪些数据类型 .....	14
47. redis 是怎么做持久化的？ .....	14
48. redis 集群你们怎么做的？ .....	15
49. redis 有哪些应用场景 .....	15

50. 一个字符串类型的值能存储的最大容量是多少? .....	15
51. linux 日常命令操作.....	15
52. 日常工作服务搭建.....	15
53. shell 编程常用哪些脚本 .....	15
54. 写出 Linux 实时查看日志变化的命令? .....	15
55. 写出 Linux 输出查找到某一个服务（如 Java 相关的服务），是否运行的命令 .....	15
56. 自动化部署，自动化运维熟悉哪些? .....	15
57. 简单列出如何做容灾与恢复，保证数据安全? .....	16
58. 公司是什么样的环境，Java 的还是 PHP .....	16
59. 做过集群，MySQL 主从之类的吗? .....	16
60. 源码用的是 git 还是 SVN? .....	16
61. 你们公司做什么业务，有多少服务器，并发多少，怎么分工的，你负责多少服务器..	16
62. 工作中部署的 mysql 主从架构，mysql 全备的数据量有多少? .....	16
63. TCP/IP 三次握手和四次挥手原理.....	16
64. squid 缓存服务器维护过吗? squid 缓存代理的里是什么? 缓存命中率怎么查看及清空缓存? .....	16
65. mysql sleep 线程过多如何解决?.....	17
66. 写 shell 脚本防范 ddos 攻击的思路.....	17

67. 你在工作中都遇到过哪些故障？怎么解决的？日常工作中处理频率最多的是哪方面问题？怎么高效解决的？ .....	17
68. tomcat 多实例怎么配置？ .....	17
69. 请说出这些符号的含义< << >> > 2> 2>> 2>&1 && .....	18
70. http 协议通信原理 .....	18
71. 你们的数据量多大，平时怎么备份的 .....	18
72. 日志分析怎么做的，都关注哪些参数 .....	18
73. MySQL 你都负责维护哪些内容 .....	18
74. shell 熟悉吗，写过哪些 shell 脚本.....	18
75. docker 在公司的应用是啥 .....	18
76. 在工作中遇到过什么棘手的问题，怎么解决的，学到了啥.....	18
77. 你最熟悉的项目是哪个，什么时间做的，做了多久，几个人，你主要在项目当中负责啥 .....	18
78. 有没有做过数据库升级，怎么做的.....	18
79. mysql 用的哪个版本，nginx 哪个版本 .....	19
80. 如果网站出现 500 错误，你的解决思路是？ .....	19
81. 读写分离用的什么工具，原理是啥.....	19
82. mysql 主从原理是啥.....	19
83. DNS 原理是啥 .....	19

84. 当前数据库服务处理速度慢，你认为可能是什么原因导致的，并阐述一下对应的解决办法 .....	19
85. 说一下你知道的存储引擎，建表时如何决定表使用存储引擎 .....	20
86. 单张表过大有什么缺点，如何解决 .....	20
87. 给你 50 台服务器，阐述一下你的拓扑结构如何设计 .....	20
88. 如果你是数据库管理员，阐述一下你如何给服务器添加授权用户 .....	20
89. 若你是数据库管理员，阐述一下你如何对数据做备份 .....	20
90. 目前常见的数据库软件有哪些，搭建数据库集群的时候，你如何选择使用哪种软件 ..	20
91. 配置 mysql 主从同步前，如何保证主库与从库的数据一致 .....	20
92 如何获知一台主机上的网站服务是否在运行 .....	20
93. 如何让系统的普通用户拥有 root 权限，什么情况下需要这么做 .....	20
94. 杀死进程的命令有哪些? .....	20
95. ELK 分别表示什么? .....	20
96. elasticsearch 的特点是什么 .....	21
97. kibana 的特点和作用是什么 .....	21
98. logstash 的特点和作用是什么 .....	21
99. ELK 能做什么? .....	21
100. ELK 在你使用的过程中主要用来做什么? .....	21

101. elasticsearch 中的分片是什么? .....	21
102. es 的 bulk 一次最大的处理数据量是多少 .....	21
103. es 的优化.....	21
104. nginx 优化怎么做的.....	21
105. 用 ansible playbook 编写过什么哪些 .....	21
106. python 你用过哪些模块.....	22
107. 用 docker 部署过哪些项目 .....	22
108. docker 常用的基础命令写 15 个.....	22
109. docker 的网络模式有哪些? .....	22
110. docker 跟虚拟机有什么区别 .....	22
111. 传统的上线流程是怎样的? .....	22
112. redis 持久化你们用的哪种方式, 两种方式有什么区别 .....	22
113. redis 集群你用的哪种方式, 几台服务器 .....	22
114. keepalived 是基于什么协议做的 .....	23
115. 错误码 301 和 302 有什么区别 .....	23
116. 错误码 502 是什么情况, 有什么解决方案 .....	23
117. MySQL 主从原理是什么? .....	23
118. MySQL 数据备份怎么实现的 .....	23

119. 备份的数据量有多大.....	23
120. gitlab 更新到什么版本了.....	23
121. centos 系统调优怎么做的.....	23
122. DNS 轮询的作用? .....	23
123. 递归查询和迭代查询分别是什么意思? .....	23
124. 在当前目录及子目录中, 查找大写字母开头的 txt 文件.....	24
125. 搜索最近七天内被访问过的所有文件.....	24
126. 搜索超过七天被访问的文件 .....	24
127. nginx 与 apache 比有什么优点.....	24
128. nginx 是如何处理请求的.....	24
129. nginx 的正向代理和反向代理分别表示什么意思.....	24
130. nginx 查看配置文件是否正确怎么查看 .....	24
131. nginx 重新加载配置文件的命令.....	24
132. fastcgi 和 cgi 有什么区别 .....	24
133. nginx 负载均衡的算法有哪些? .....	24
134. nginx 使用中, session 不同步怎么办? .....	24
135. nginx 如何实现防盗链.....	24
136. 为什么要做动静分离? apache 能做动静分离吗? nginx 怎么做动静分离? .....	25

137. mysql 慢查询怎么设置.....	25
138. 简述 MySQL 的常见两种引擎的区别.....	25
139. mysql 主从一致性校验是怎么做的（重要）.....	25
140. truncate delete drop 的区别.....	25
141. 你们公司的 redis 架构.....	25
142. shell 脚本中\$?的用途是做什么.....	25
143. mysql 不同步的处理方法.....	26
144. 如何修改 Linux 的最大文件句柄的限制.....	26
145. 如何验证批量主机的在线情况？.....	26
146. 说说你在工作中最自豪的一件事情.....	26
147. 说下你在工作中遇到的三个最棘手的事情.....	26
148. 你们的服务器在什么地方？.....	26
149. 阿里云用过哪些产品.....	26
150. SLB 是基于几层的负载均衡.....	26
151. 为什么要做日志切割.....	26
152. nginx 有几种工作模式，都是哪几种.....	26
153. tomcat 的版本用的啥.....	26
154. redis 在你们公司用来缓存什么数据.....	27

155. 公司产品是什么，注册数有多少 .....	27
156. 你平时都有学习什么新的技术 .....	27
157. 你们公司的存储用的什么 .....	27
158. nginx 日志切割怎么做的 .....	27
159. nginx 做了哪些优化 .....	27
160. 刚装好的系统做了哪些优化 .....	27
161. 数据库查询很慢排查 .....	27
162. mysql 数据被误删除后，你尝试过使用哪些方法或者手段来挽回数据 .....	27
163. 一主多从的环境，从库宕机，请问你如何恢复？ .....	27
164. mysql 集群和高可用方案有哪些，你在公司用过哪些方案？ .....	27
165. 日常巡检都做什么？ .....	27
166. raid10 raid5 raid1 区别是啥，有什么优势 .....	28
167. 请简述软连接和硬链接的区别 .....	28
168. inode 和 block 是什么东西，请简述解释 .....	28
169. sed 命令的 -n -r 参数均表示什么意思 .....	28
170. 如果采购了一块硬盘，需要做哪些处理，才能正常使用磁盘 .....	28
171. 运维人员的标准工作流程 .....	28
172. 301 和 302 跳转的区别 .....	28

173. 上一家公司是做什么的? .....	28
174. 数据库增长量是多少 .....	28
175. 上家公司你主要负责哪些工作和职责 .....	28
176. 你处理过印象最深刻的故障是什么 .....	28
177. 你最擅长的技术是什么 .....	29
178. 如何查看当前的 Linux 服务器的运行级别? .....	29
179. 如何查看 Linux 的默认网关? .....	29
180. 如何识别 Linux 系统中指定文件 (/etc/fstab) 的关联包? .....	29
181. Linux 系统中的 /proc 文件系统有什么用? .....	29
182. 如何在 /usr 目录下找出大小超过 10MB 的文件? .....	29
183. 如何在 /home 目录下找出 120 天之前被修改过的文件? .....	29
184. 如何在 /var 目录下找出 90 天之内未被访问过的文件? .....	29
185. 在整个目录下查找文件 "core", 如发现则无需提示直接删除它们。 .....	29
186. linux 中的 at 命令有什么用? .....	29
187. 固态硬盘和传统硬盘有什么区别, 并说明它们的优点和缺点。 .....	29
188. 多核 CPU 和单核 CPU 的优点和缺点, 是否所有程序在多核 CPU 上运行速度都快? 为什么? .....	30
189. 有一 web 服务器, 某天某用户投诉上某个网站速度很慢, 你如何查找原因。 .....	30

190. 删除/tmp/oldboy/下除 passwd 以外的其他文件.....	30
191. 请打印/etc/passwd 文件中的第 2-5 行（不低于三种方法）.....	30
192. 把/data 目录及其子目录下所有以扩展名.txt 结尾的文件中包含 oldgirl 的字符串全部替换为 oldboy.....	30
193. 查找/oldboy 下所有 7 天以前以 log 结尾的大于 1M 的文件移动/tmp 下.....	30
194 什么是 linux 的运行级别，请描述 linux 的运行级别不同数字的含义?.....	30
195. 请描述 buffer 和 cache 的区别?.....	30
196. nginx 配合 php 工作的 FastCGI 工作原理.....	30
197. keepalived 高可用服务工作原理.....	30
198. CDN 工作原理.....	31
199. linux 启动过程.....	31
200. linux 系统文件删除原理.....	31
201. linux 默认文件系统介绍及选型.....	31
202. 写出你的技术栈.....	32
203. 一台服务器由两块硬盘，一块 200G，一块 2T，如何分区.....	32
204. 分区命令有哪两个，有什么区别?.....	32
205. 自动化工具用过哪些?.....	32
206. docker 有哪几种网络模式.....	32

207. 查找文件大小在 200M 以上的文件，找到并删除.....	32
208. 使用什么命令可以查看系统的版本信息及当前所开放的端口号.....	32
209. ssh 的配置文件路径.....	32
210. 你常用的性能检测命令.....	33
211. 如何查看端口是否被占用.....	33
212. tomcat 的三个端口的作用.....	33
213. 你们公司有多少台服务器.....	33
214. 你们的服务器放在什么地方.....	33
215. 阿里云用过哪些产品.....	33
216. 如果一个硬盘显示还有 200G 空间，但是无法写入内容，分析原因.....	33
217. 缓存有哪些，消息队列有哪些.....	33
218. raid0 raid1 raid5 有什么区别，使用率是多少.....	33
219. 你处理过最棘手的事情是啥.....	33
220. 读写分离怎么做的.....	34
221. nfs 里面存储什么样的数据.....	34
222. mysql 存储什么样的数据.....	34
223. 你们的代码是如何上线的.....	34
224. 你们上线的周期和时间分别是啥.....	34

225. 你们的数据量多大，如何备份 .....	34
226. 消息队列的作用是啥 .....	34
227. 说下公司的持续集成（CICD）？开发、测试环境、是不是用同一个 jenkins。人员的权限如何分配？ .....	34
228. 有遇到过网络问题吗，是如何解决的？丢包问题如何解决？ .....	34
229. docker 有用到过吗？ .....	34
230. 公司服务器的配置是怎么样的？ .....	34
231. hodoop 有接触过吗？ .....	34
232. 你们的数据安全是怎么管理的？ .....	35
233. 从运维角度做过优化吗？ .....	35
234. 自动化运维做到监控和上线，还有其他的吗？ .....	35
235. 数据库的安全和优化有涉及过吗？ .....	35
236. 数据库的数据量有多少？ .....	35
237. 日志平台有弄过吗，是运维弄的还是开发弄的？ .....	35
238. 如何查看/var/log 目录下的文件数？ .....	35
239. ps aux 中的 VSZ 代表什么意思，RSS 代表什么意思？ .....	35
240. 如何检测并修复/dev/hda5？ .....	35
241. 介绍下 Linux 系统的开机启动顺序 .....	35

242. 符号链接与硬链接的区别 .....	35
243. 如何在文本里面进行复制、粘贴，删除行，删除全部，按行查找和按字母查找? 3 6	
244. 手动安装 grub .....	36
245. 修改内核参数 .....	36
246. 怎么把脚本添加到系统服务里，即用 service 来调用? .....	36
247. 如何让 history 命令显示具体时间? .....	36
248. 固态硬盘和普通硬盘的区别 .....	36
249. SYN 攻击概念和检测是否遭到 syn 攻击? 一也是 ddos 攻击 .....	36
250. 修改 linux 的最大文件打开数（默认 1024） .....	37
251. nginx 日常优化? .....	37
252. 分布式文件存储是否有过了解和使用，了解过的有什么特性 .....	37
254. 500 台 db，在最快时间之内重启 .....	37
255. 你是如何监控你们的数据库的? 你们的慢日志都是怎么查询的? .....	37
256. 你是否做过主从一致性校验，如果有，怎么做的，如果没有，你打算怎么做? .....	37
257. MySQL 中 InnoDB 引擎的行锁是通过加在什么上完成(或称实现)的? 为什么是这样子 的? .....	37
258. 什么是 mysql 多实例? .....	38

259. 如何配置 mysql 多实例? .....	38
260. 授权 oldboy 用户管理 bbs 库所有表, 127.16.10.0 网段访问, 密码 123456.....	38
261. truncate 和 delete 区别.....	38
262. mysql 忘记 root 密码怎么解决? .....	38
263. 如何从 mysqldump 产生的全库备份中只恢复某一个库、某一张表? .....	38
264. mysql 数据库查询慢排查方法 .....	39
265. mysql 睡眠连接(sleep 线程)过多, 会对 mysql 服务器造成什么影响? .....	39
266. key_buffer_size 参数作用? 如何不重启设置生效? .....	40
267. 主从复制的常见不同步问题 .....	40
268. 如何脱离主从结构? .....	40
269. 判断主从复制到哪一步, 参考查看主从同步情况(了解) .....	40
270. mysql 服务器的主从配置.....	40
271. 主从同步因从库数据冲突导致同步停止的解决方法 .....	41
272. 一主多从的 mysql, 当主库宕机, 将其他的某个从库提升为主的方法步骤(主库故障切换, 验证完毕, 没毛病) .....	41
273. 一主多从, 主库正常, 但有计划的切换从库为主库时候的处理(主库没故障, 有计划的切换, 验证完毕, 没毛病) .....	42
二、复杂的命令型 .....	43
1. 如何使用 iptables 将本地 80 端口的请求转发到 8080 端口, .....	43

2. 限制 apache/nginx 每秒新建连接数为 1，峰值为 3（防火墙限制） .....	43
3. 统计 apache 访问日志中登录服务器次数最多的前 5 个 IP .....	43
4. 统计出 apache 的 access.log 中访问量最多的 5 个 ip .....	43
5. 如何查看 Linux 系统每个 ip 的连接数？（3, 4, 5 本质上差不多） .....	44
6. 如何查看 http 的并发请求数与其 TCP 连接状态？ .....	44
7. 使用 netstat 和 awk 命令统计下网络连接数（6 和 7 本质是一样） .....	44
8. nginx 日志过滤 10 点到 12 点之间访问 IP 排名和统计 .....	44
9. 显示 a.txt 中包含了:一个数字:(即两个冒号中间一个数字)的行.....	44
10. shell 下生成 32 位随机密码 .....	44
11. 在 11 月份内，每天的早上 6 点到 12 点，每隔 2 小时执行一次 usr/bin/httpd.sh 怎么实现 .....	45
三、非技术型 .....	45
1. 你的离职原因是什么 .....	45
2. 你对加班怎么看的 .....	45
3. 你是如何理解运维这个岗位的 .....	45
4. 作为运维工程师，你对该职位的认识和理解有哪些，日常工作中应该怎么做.....	45
5. 你对出差怎么看 .....	45
6. 你们的薪资构成是咋样的.....	45

7. 如果你入职，你打算怎么开展你的工作 .....	45
8. 你对未来有什么职业规划.....	45
9. 你觉得 linux 工程师的职责? .....	45
10. 未来 5-10 年的规划?.....	45
11. 你们公司有什么规章制度? 比如运维要完成某项任务，有什么规范流程? .....	46
12. 在工作中应该做什么? .....	46
13. 先做下自我介绍 .....	46
14. 你在公司中主要负责那一部分 .....	46
15. 你最擅长的是哪一部分 .....	46
16. 你们公司是统招吗，学信网可查吗.....	46
17. 你们的操作系统是 6 还是 7，内核多少.....	46
18. 你工作三年最大的收获是什么 .....	46
19. 你找工作最关注的是什么.....	46
20. 你还有什么想问我的? .....	46
21. 你上家薪资多少，期望薪资多少 怎么说? .....	47
22. 你是如何接触并学习 Linux 的 .....	47
23. 你对未来有什么职业规划.....	47
四、k8s 专题型 .....	47

1. 什么是 kubernetes? .....	47
2. Kubernetes 与 Docker 有什么关系? .....	47
3. 在主机和容器上部署应用程序有什么区别? .....	47
4. Kubernetes 主节点后节点的不同组件有哪些? .....	47
5. Kube-proxy 有什么功能? .....	47
6. kube-apiserver 和 kube-scheduler 的作用是什么? .....	47
7. 什么是 ETCD? .....	47
8. 你对 Kubernetes 的负载均衡器有什么了解? .....	48
9. 什么是 Ingress 网络, 它是如何工作的? .....	48
10. ReplicaSet 和 Replication Controller 之间有什么区别? .....	48
11. 什么是 Headless Service? .....	48
12. 如何看待公司从单一服务转向微服务并部署其服务容器? .....	48
13. 公司希望通过采用新技术来优化其工作负载的分配, 如何有效实现? .....	48
14. Kubernetes 集群数据存储在以下哪个位置? .....	48
15. 哪个是 Kubernetes 控制器? .....	48
16. k8s 各节点的组件和作用? .....	48
17. k8s 中的证书和私钥种类? .....	48
18. k8s 中镜像下载策略 (3 种) .....	49

19. k8s 中 pod 故时的重启策略（3 种） .....	49
20. pv 的几种访问模式？ .....	49
21. pv, pvc 的作用？ .....	49
22. 客户端访问资源需要经过几关？ .....	49
23. k8s 集群 3 主 5 从高可用架构（kubeadmin 方式安装 k8s）, 如下图： .....	49
24. k8s 集群 3 主 5 从高可用架构（二进制方式安装 k8s）, 如下图： .....	50
25. 电子合同 k8s 部署架构 .....	50

## 一、简单型

### 1. 你平时在公司主要做什么？

日常监控维护业务，

服务器巡检，调优，

写一些日常使用的脚本，日常工作形成文档化，

服务优化：nginx 优化，tomcat 优化，系统优化

### 2. 你们原来公司的网站架构是怎么样的？

搭的 LNMP 架构，Nginx 做了集群，做了反向代理负载均衡，

mycat 读写分离，数据库做的一主两从，并且做了 MHA 高可用

### 3. 你对哪一块比较熟练或者精通？

对企业自动化交付 CICD 方面做得比较熟练，

缓存服务 memcached, squid, redis 这种缓存服务器，

还有 lvs, keealived 这种负载均衡集群软件

容器 docker, k8s 方面

mysql 数据库

### 4. 介绍一下负载均衡？

负载均衡我用过 lvs 和 nginx 和阿里云 SLB

nginx 负载均衡：工作在网络第 7 层，支持 http 应用进行分流

lvs: 抗负载能力很强，工作 4 层，配置简单，很稳定，不产生流量

## 5. lvs 内部原理?

lvs 负载均衡模式有 NAT tun dr 默认 DR 模式

lvs 工作在与内核的 ipvs 模块来实现功能, 主要工作于 netfilter 的 INPUT 链上

## 6. nginx lvs haproxy 三个有什么区别

**lvs 优势:** 抗负载能力很强、工作稳定、不占什么流量、pv 超过 1000 万可用 lvs

**nginx:** nginx 工作第 7 层, 支持 http 应用本身分流, lvs 没有这个功能

nginx 对网络依赖很小, nginx 安装简单也稳定, 流量日 pv <=1000 万 nginx 足以能撑住, 一般不是特别大的公司都达不到 lvs 使用级别。官方说并发 50000 都没什么问题, 测试过并发 1-2 万根本没什么问题

**haproxy:**

- 1). haproxy 是支持虚拟主机的, 可以工作在 4、7 层(支持多网段)
- 2). haproxy 的优点能够补充 nginx 的一些缺点, 比如支持 session 的保持, Cookie 的引导; 同时支持通过获取指定的 url 来检测后端服务器的状态。
- 3). haproxy 跟 lvs 类似, 本身就只是一款负载均衡软件; 单纯从效率上来讲 haproxy 会比 nginx 有更出色的负载均衡速度, 在并发处理上也是优于 Nginx 的。
- 4). haproxy 支持 TCP 协议的负载均衡转发, 可以对 MySQL 读进行负载均衡, 对后端的 MySQL 节点进行检测和负载均衡。

## 7. lvs 主要 3 种工作模式原理

### 1). NAT 模式 (NAT)

原理: 就是把客户端发来的数据包 IP 头的目的地址, 在负载均衡器上换成其中一台 RS 的 IP 地址, 并发至此 RS 来处理, RS 处理完成后把数据包交给负载均衡器, 负载均衡器再把数据包的原 IP 地址改为自己的 IP, 将目的地址改为客户端 IP 地址即可。期间, 无论是进来的流量, 还是出去的流量, 都必须经过负载均衡器。

优点: 集群中的物理服务器可以使用任何支持 TCP/IP 操作系统, 只有负载均衡器需要一个合法的 IP 地址。

缺点: 扩展性有限。当服务器节点 (普通 PC 服务器) 增长过多时, 负载均衡器将成为整个系统的瓶颈, 因为所有的请求包和应答包的流向都经过负载均衡器。当服务器节点过多时, 大量的数据包都交汇在负载均衡器那, 速度就会变慢!

### 2). IP 隧道模式 (TUN)

原理: 首先要知道, 互联网上的大多 Internet 服务的请求包很短小, 而应答包通常很大。那么隧道模式就是, 把客户端发来的数据包, 封装一个新的 IP 头标记 (仅目的 IP) 发给 RS, RS 收到后, 先把数据包的头解开, 还原数据包, 处理后, 直接返回给客户端, 不需要再经过负载均衡器。注意, 由于 RS 需要对负载均衡器发过来的数据包进行还原, 所以说必须支持 IPTUNNEL 协议。所以, 在 RS 的内核中, 必须编译支持 IPTUNNEL 这个选项

优点: 负载均衡器只负责将请求包分发给后端节点服务器, 而 RS 将应答包直接发给用户。所以, 减少了负载均衡器的大量数据流动, 负载均衡器不再是系统的瓶颈, 就能处理很巨大的请求量, 这种方式, 一台负载均衡器能够为很多 RS 进行分发。而且跑在公网上就能进行不同地域的分发。

缺点: 隧道模式的 RS 节点需要合法 IP, 这种方式需要所有的服务器支持 "IP Tunneling" (IP Encapsulation) 协议, 服务器可能只局限在部分 Linux 系统上。

### 3). 直接路由模式 (DR)

原理: 负载均衡器和 RS 都使用同一个 IP 对外服务。但只有 DR 对 ARP 请求进行响应, 所有 RS 对本身这个 IP 的 ARP 请求保持静默。也就是说, 网关会把对这个服务 IP 的请求全部定向给 DR, 而 DR 收到数据包后根据调度算法, 找出对应的 RS, 把目的 MAC 地址改为 RS 的 MAC (因为 IP 一致) 并将请求分发给这台 RS。这时 RS 收到这个数据包, 处理完成之后, 由于 IP 一致, 可以直接将数据返给客户, 则等于直接从客户端收到这个数据包无异, 处理后直接返回给客户端。由于负载均衡器要对二层包头进行改换, 所以负载均衡器和 RS 之间必须在一个广播域, 也可以简单的理解为在同一台交换机上。

优点: 和 TUN (隧道模式) 一样, 负载均衡器也只是分发请求, 应答包通过单独的路由方法返回给客户端。与 VS-TUN 相比, VS-DR 这种实现方式不需要隧道结构, 因此可以使用大多数操作系统做为物理服务器。

缺点: (不能说缺点, 只能说是不足) 要求负载均衡器的网卡必须与物理网卡在一个物理段上。

## 8. lvs 工作原理, 算法, 工作模式?

lvs: 实现负载均衡集群部署的软件, 有一台或多台调度器组成, 通过加载 lvs 内核模块并且生成虚拟 ip, 通过虚拟 ip 接收客户端的请求, 再根据自身配置的调度算法, 实现对请求的转发

**算法:** 大概有 10 种, 主要的有下面几种:

**静态算法:** 只是根据算法进行调度并不考虑后端 REALSERVER 的实际连接情况

rr: 轮询算法: 按照节点顺序一个一个来, 均等地对待每一台服务器, 不管服务器上的实际连接数和系统负载。

wrr: 加权轮询算法: 根据节点权重以及节点顺序分发请求, 调度器可以自动问询真实服务器的负载情况, 并动态调整权重。

sh: 源地址散列调度算法: 匹配客户端最近一次访问的服务节点, 并将请求交给这个服务节点, 根据源地址散列算法进行静态分配固定的服务器资源。

dh: 目标地址散列算法: 根据目标 ip 地址进行哈希计算, 得到散列值, 匹配客户端最近一次访问的服务节点, 并将请求交给这个服务节点, 根据目标 ip 地址通过散列函数将目标 ip 与服务器建立映射关系, 出现服务器不可用或负载过高的情况下, 发往该目标 ip 的请求会固定发给该服务器。

**动态算法:**

lc: 最少连接数算法: 动态地将网络请求调度到已建立的连接数最少的服务器上。

wlc: 加权最少连接数算法: 调度器可以自动问询真实服务器的负载情况, 并动态调整权重。

lblc: 基于局部的最少连接数调度算法: 先根据请求的目标 ip 地址寻找最近的该目标 ip 地址所有使用的服务器, 如果这台服务器依然可用, 并有能力处理该请求, 调度器会尽量选择相同的服务器, 否则会继续选择其他可行的服务器, 即

如果服务节点因为自身故障暂时无法接收请求, 则调度器在服务器集群中找出一台连接量最少的节点来处理请求。

lblcr: 带复制的基于局部的最少连接算法: 记录的并不是要给目标 ip 与一台服务器之间的连接记录, 它会维护一个目标 ip 到一组服务器之间的映射关系, 防止单点服务器负载过高。

**工作模式:** 主要模式有 3 种

**1). DR 模式:** (也是默认模式, 直接路由模式)

客户端访问负载均衡器的 vip, 此时在集群中负载均衡器和后端的 realserver 都具有 vip, 所以负载均衡器通过 vip 进行转发是不实际的。那么只能通过二层协议, 修改数据包二层的目标 mac 地址来进行转发, 后端服务器处理后, 通过本机的 lo 网卡上的 vip 直接返回数据包。

- a). 环境, 为了模拟生产环境, 我们将 dip rip 设置为同一个网段, dr 和 rs 上面的虚拟 ip 均为 vip(生产环境中的公网 ip)。
- b). 注意, arp 静默需要开启, 后端服务器不接受请求, 只处理

**2). NAT 模式:**

客户端访问负载均衡器的 vip, 负载均衡器通过转换目标 ip 地址的方式, 实现负载均衡, 将本该发送给负载均衡器的数据包的目标地址, 修改成后端 real server 的 ip 地址, 然后又 rs 进行处理并返回, 返回过程中, 经过负载均衡器, 负载均衡器将源地址转换为 vip, 完成本次请求。

a). 环境, 为了模拟生产环境, 我们使用一台 centos7 服务器作为负载均衡器, 配置两块网卡, 一块设置 vip(桥接, 生产中是公网 ip), 一块设置 dip(nat 或仅主机模式, 生产中是内网 ip), 后端设置 rip(nat 或仅主机模式, 于 dip 同一网段)

b). 注意, 负载均衡器需要开启路由转发, 所有服务器均关闭防火墙, selinux。如果测试环境中, 使用的是于后端服务器同一网段的客户端访问, 则会丢弃数据包, 超时连接, 这种情况需要在后端 real-server 添加路由规则, route add -host cip gw dip 则可以访问

**3). TUN 模式(隧道模式):**

客户端访问负载均衡器的 vip, 这种模式, 可以跨区域进行负载均衡, vip 和 rip 均为公网 ip, dip 将请求发送给 rip 的过程中是加密的, 后端 rs 上的 vip 直接将请求进行回复, 不经过 dr 负载均衡器。

- a). 环境, 需要两个公网 ip
- b). 添加 vip 在隧道网卡。

**9. mysql 用的哪个版本, 你们的数据库怎么备份, 什么时间备份, 备份数据量多大**

用的 5.7 逻辑备份 每天全量备份 500G 后面改用 xbk 备份。

**10. memcached 工作原理(内存管理机制)**

memcached 是一种内存缓存软件，在工作中经常用来缓存数据库的查询数据，数据被缓存在事先预分配的 memcached 管理的内存中，可以通过 API 或命令的方式存取内存中缓存的这些数据，memcached 服务内存中缓存的数据就像一张巨大的 HASH 表，每条数据都是以 key-value 对的形式存在。每个被缓存的对象或数据都有唯一的标识符 key，存取操作通过这个 key 进行。保存到 Memcached 中的对象或数据放置在内存中，并不会作为文件存储在磁盘上，所以存取速度非常快。由于没有对这些对象进行持久性存储，因此在服务器端的服务重启之后存储在内存中的这些数据就会消失。而且当存储的容量达到启动时设定的值时，就自动使用 LRU 算法删除不用的缓存。

#### 11. nginx 状态码 499 是什么意思

服务端处理时间过长，客户端主动关闭了连接

#### 12. nginx 状态码 502 错误可能的原因有哪些？

502 一般就是后端问题，服务挂了或者进程不够用了大并发时候处理不过来，增加后台服务器

#### 13. 状态码 200, 500 502 304 ， 分别表示什么意思

200 状态：访问正常

500 状态：一般数据库问题

502 状态：一般是后端出现异常，常见处理不过来或者挂了

304 状态：客户端有缓存情况下服务端的一种响应

#### 14. 维护网站过程中遇到的问题，处理过什么故障？（状态码：413、 504 的意思）

web 服务报错 413，查阅后发现是客户端请求的数据量太大，我没有指定 http 请求最大大小，配置文件修改增大了 client\_max\_body\_size 的值完全解决（如果上传文件大小超过 client\_max\_body\_size 时，会报 413 entity too large 的错误）。

504 错误：请求后端服务的时候，后端服务很长时间才响应，查看错误日志后修改增大了 proxy\_read\_timeout 和 proxy\_read\_time 的值后完美解决。

#### 15. FTP 主动和被动的区别（指的是服务器端的主动和被动）？

**主动：**

FTP 客户端向服务器的 FTP 控制端口（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，客户端告诉服务器我开放了某>1024 的大端口，你过来连接我。于是服务器从 20 端口向客户端的大端口发送连接请求，建立一条数据链路来传送数据。在数据链路建立过程中是服务器主动请求，所以称为主动模式。

**被动：**

FTP 客户端向服务器的 FTP 控制端口（默认 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，服务器告诉客户端，我打开了某端口(>1024 的大端口)，你过来连我。于是客户端从>1024 的大端口向服务器的该大端口发送连接请求，建立一条数据链路来传送数据。在数据链路建立的过程中是服务器被动等待客户机的请求，所以称被动模式。

#### 16. apache 两种工作模式区别及优化？

apache 的三种工作模式

1). prefork: 预派生多进程，当 apache 启动时，启动一个主进程，并根据配置由该主进程派生多个子进程，这些子进程等待客户端发送来的请求并处理，由于已经有预先派生好的进程，所以在处理客户端请求时效率比较高，如果客户端的请求数量超过预派生的进程数量，则该主机会派生新的进程来处理后来的请求，支持的并发量低，就是说不能同时处理客户端发来的请求，但是稳定

2) worker: 预派生多线程，当 apache 启动时，启动一个主进程，并根据配置由该主进程派生多个子进程，每个子进程又派生出多个线程，这些线程等待客户端发送来的请求并处理由于已经有预先派生好的线程所以在处理客户端的请求时效率比较高，如果客户端的请求数量超过预派生的线程，则每个子进程会派生出新的线程来处理后来的请求，支持的并发量相较于 prefork 模式要多，但是没有 prefork 模式稳定，但是在长链接的模式下，线程在连接时间范围内会一直等待客户端发来的请求，线程利用率不高

3). event: worker 的升级版，通过管理线程，可以对处理长连接请求的线程进行调度，当客户端暂时没有请求发来时，则该处理长连接的线程可以接受别的请求提高了线程的利用率

优化：控制 maxclients 最大连接数的设置，避免服务器产生太多子进程而发生交换

选择更好的 cpu 内存 硬盘

hostnamelookups 设置为 off 尽量减少 dns 查询的次数

修改 apache 最大并发连接数

### 17. mysql 主从原理？如果主库不同步报错了怎么恢复？

**原理：**

主库：需要开启二进制日志文件，并且对数据库进行授权，允许从库访问，例如主库使用的二进制文件名是 Mysql-bin.01，授权的用户密码分别是 slave,123.com

从库：需要指定自己的主库是谁，一个从库只能有一个主库，一个主库可以有多个从库，指定主库时需要得信息是 master\_host 主库的主机地址，master\_user 访问主库时使用的用户，master\_log\_file 同步主库哪个二进制日志文件的操作，master\_Log\_pos 从二进制日志哪个位置的操作开始同步，从库还需要开启中继日志来记录从主库获取的日志信息

当主从部署完成之后，从节点会派生出两个线程分别是 I/O 线程（用来连接主库并将主库二进制日志获取到的信息写入到中继日志），SQL 线程（从中继日志中提取 sql 语句并执行，实现从节点与主节点的数据同步），主节点会派生 DUMP 线程，当主库数据更新时，主节点的 DUMP 线程会通知从节点的 I/O 线程，I/O 线程通过 slave 用户和密码 123.com 连接到主节点之后会告诉 DUMP 线程自己要从二进制日志文件 Mysql-bin.01 中指定的位置读取数据，获取到数据之后写入到自己的中继日志中，I/O 线程任务完成，SQL 线程检测到中继日志更新之后会从中继日志中提取 SQL 语句并执行这些语句完成数据同步

**主库不同步的原因有很多：**

网络的延迟？主从两台机器负载不一致？ max\_allowed\_packet 设置不一致？ 自增键不一致？

最大的问题就是 io 线程和 sql 线程不同步

可以先进主库锁表，再数据备份，把备份文件传到从库，在停止从库的主从状态，从库在导入书数据备份，设置从库同步，重新开启从同步即可。

### 18. 如何备份大数据 mysql 数据文件，mysql 都优化哪些？

xtrabackup 热备份， xtrabackup 备份时候是热备，不会锁表。

mysql 优化：主要有两个主要方面：**安全、性能、参数、架构、sql 语句**

**安全方面：**可以把默认端口 3306 改为其他的，对用户降权，以比普通用户运行 mysql 即可，开启二进制日志文件，最好在禁止一下 root 账户远程访问

**性能方面：**可以给主机选择合理的 cpu，内存调大，硬盘，可以对 Linux 系统内核参数优化，数据库优化的话可以对执行计划，索引，sql 改写等等优化

**参数优化：** innodb 的 buffer 参数调大，连接数调大，缓存的参数优化

**架构优化：**主从，读写分离，一主多从，架构调整。

**sql 语句优化：**协助开发优化，开启慢查询 sql，是否没有索引，是否需要借助 redis 缓存，es 等。

### 19. 你在家公司用的什么监控软件（zabbix）？都监控些什么？怎么增加模板？

zabbix 对系统层监控和应用层监控

系统层：如 cpu、内存、硬盘、磁盘 i/o、流量、连接数

应用层：进程、mysql 主从状态、tomcat 性能监控、mysql 性能监控

网站状态

zabbix+grafana 监控可视化

故障报警，预警，邮件或微信

自动注册，自动安装客户端

### 20. nagios, cacti 维护过吗，平时都监控什么？

系统层监控：用阿里云对 cpu、内存、硬盘、带宽、磁盘 i./o

应用层监控：各种进程监控、应用性能监控、mysql 主从监控、并发连接监控

### 21. 一台 web 服务器应该监控哪些指标

网站状态 cpu 内存 磁盘 i0 连接数

## 22. zabbix 监控数据库监控哪些参数

主从状态 进程 性能指标比如 查询吞吐量 查询执行性能 连接情况 缓冲池情况，模板导入监控

## 23. zabbix 有哪些监控项

模板自带监控项

自定义监控项

## 24. 监控一台服务器时，你觉得应该监控服务器的哪些资源

系统层：cpu 内存 硬盘 磁盘 i/o 连接数

应用层：进程状态、服务性能状态、主从同步状态

对称加密和非对称加密的区别

## 25. 你们监控的时候日志平台会监控吗？

只监控日志监控平台服务器性能监控，并实现预警，其他就没有做

## 26. 你们会监控数据库的性能吗，会定标准码？有实际遇到过吗，如何解决？

数据库监控我们采用 zabbix 进行监控，监控 mysql 进程、各种配置参数监控、性能、主从监控

## 27. 写 shell 脚本监控 mysql 主从的思路

判断 io 和 sql 线程是否为 yes,

## 28. 写 shell 脚本监控所有 IP 是否存活的思路

使用 ping 命令+for 循环，循环 ping 所有 ip 的列表，能通的就是在线，不能通的就是离线

## 29. 公司出口带宽？每天网站 pv？ uv？

带宽：100M

pv:100W

uv:5W

## 30. 你们公司的网站出口带宽是多少？网站访问量是多少？是怎么计算的？

我们对外服务器带宽 100M 每天 pv 100 万 UV 大概 2 万

## 31. 你们公司是什么架构？都有哪些网上业务？详细说一下？

前端使用 SLB 负载均衡 负载 3 台应用服务器 缓存层使用 redis 数据库层 mysql 主从架构

## 32. 你们公司备份策略是什么？数据量多大？备份周期？备份工具？备份方式？是否做数据恢复备份演练？

我们线上服务器数据库采用数据库逻辑备份 mysqldump 每天进行全量备份，binlog 日志保留 7 天

我们线上服务器异地也有一套环境就是做灾备用和故障演练用的，x bk 备份或逻辑备份，数据量 500G 大小。

## 33. 你在工作中是怎么做日志分析的？例如：访问日志、错误日志等

我们公司使用 ELK 日志分析系统做的，日志进行可视化，服务器日志每周备份一次进行日志切割

## 34. 你们公司日志多久清理一次？

定期删除一个月之前的日志。

**35. 你们公司运维部门有多少人？多少服务器？基本优化项目？**

我们现在就我一个人，测试环境和生产环境一起是 20 台服务器 日常系统内核优化

**36. 你在上家公司针对防攻击、扩容、容灾等问题是怎么制定和优化解决方案的？**

防攻击我们公司使用脚本做的实时探测异常 ip 多次请求，拉黑机制

扩容制作运维扩充方法，发送邮件通知有关人员知悉，先写文档，在测试环境测试好后，再正式环境再做

容灾我们公司没有做，我们架构基本都是有高可用。

**37. 你们公司如何自动化代码上线？**

我们公司代码上线使用 jenkins+mave 自动构建编译打包推送到应用服务器

如果是手动方法：我们有制定手动代码上线流程

根据流程上线：如果测试环境测试没有问题时候，使用灰度方法发布上线

**38. 你们公司都有哪些应用集群？**

我们公司集群只有 tomcat, redis 集群

**39. 你们公司用的是物理服务器还是云服务？**

阿里云服务器，物理机也有

**40. 你们公司安全方面怎么我做的？监控体系结构是怎样的？**

1). 服务器有外网 ip 全部去掉，采用内网通信除了对外服务器必须对外，只有 nginx 对外开放外网 ip

2). 关闭密码认证登录服务器，采用秘钥认证登录服务器

3). 服务器权限不对外开放

4). 利用防火墙和阿里云安全组关闭没必要的对外端口

5). jumpserver 跳板机使用

6). 更改 ssh 端口

**41. 如何查看 Linux 系统当前的状态，cpu 内存的使用情况及负载**

```
top free -h uptime
```

**42. 写出你所了解的门户网站的服务架构，可用什么方式实现的高可用、负载均衡？**

lvs+keepalived

nginx+keepalived

nginx +tomcat nginx 负责转发和处理静态资源 tomcat 负责动态

**43. nginx 日志的路径为 opt/logs/access. logs, 假设日志不能自动分割, 写脚本让日志每天能够定时自动分割**

```
cp access. logs /tmp/"%Y%m%d". access. logs
```

```
echo > access. log
```

**44. 写一个脚本，查找 15 天前以 png 结尾的文件并删除**

```
#!/bin/bash
```

```
find /picture/*.png* -type f -mtime +15 -exec rm {} \;
```

```
find /mnt -type f -mtime +15 -name *.png -exec rm {} \;
```

#### 45. 你们为什么使用 redis?

答: 面试官好, 我们的数据库主要用的是 MySQL, 目前的业务发展设计到高并发场景下的库存扣减, 网站首页访问流量高峰等等, 数据库已经支撑不住如此高的并发, 于是我们引入了缓存中间件, 市面上的中间件 redis 和 memcached 我们有在用。我在使用过程中对 reids 和 memcached 做了下对比:

- 1). redis 相比 memcached 来说用拥有更多的数据结构和更丰富的数据操作。所以像我们公司设计复杂的结构和操作, 我们选择了 redis
- 2). 在内存使用效率上做了下对比, 单纯使用 key-value 存储的话, memcached 的内存利用率更高, 但 redis 采用 hash 结构来做 key-value 存储, 由于其组合式的压缩, 让内存的利用率高于了 memcached
- 3). 在性能上, redis 只使用单核, 而 memcached 可以使用多核, 所以平均每一个核上的 redis 在存储小数据时比 memcached 性能要高。在使用过程中, 有时候也设计到 100K 以上的数据, 这个时候, redis 的性能就稍逊于 memcached
- 4). 在数据持久化支持上, redis 提供了两种持久化策略, 分别是 RDB 快照和 AOF 日志, 而 memcached 不支持数据持久化操作。所以综合下来我们选择了 redis。

#### 46. redis 有哪些数据类型

答: 关于 redis 的数据类型, 我知道的有这么几个: 字符串 string, 字典 hash, 列表 list, 集合 set, 有序集合 SortedSet, 因为公司涉及到高并发, 所以还接触过 Bloom Filter, 以及支持全文搜索的 RediSearch 等。

#### 47. redis 是怎么做持久化的?

答: 持久化有两种方式 RDB 和 AOF, RDB 做镜像全量持久化, AOF 做增量持久化。我们公司一般 RDB 和 AOF 会配合起来使用。在 redis 实例重启的时候, 会使用 RDB 持久化文件重构内存, 再使用 AOF 存放近期的操作指令来恢复重启前的状态。当然 reids 的机制是 AOF 开启且存在 AOF 文件时候, 优先加载 AOF 文件。AOF 关闭或者 AOF 文件不存在的时候, 加载 RDB 文件。

大流量的时候不要开 AOF, 开 RDB 即可。

如果服务器突然断电, 可以使用 AOF 恢复, AOF 日志有 sync 属性配置, 在高性能场景下使用定时 sync, 比如 1s1 次, 这样即使断电最多丢失 1s 的数据。

#### 48. redis 集群你们怎么做的?

答: redis 集群我接触过的有这么几种,

- 1). 主从模式, 一主多从。虽然 redis 单机的 qps 可以达到 10W+, 但是在某些高访问场景下依然不够用, 一般通过读写分离来增加 slave, 减少主机压力。

这种方式需要用户自行开发主从切换功能, 也就是使用哨兵去探测每个实例的健康状况, 然后通过指令进行集群状态的改变。当集群规模变大可以加入类似 zk 的主动通知功能, 降低风险。一般可以维护几千个 redis 节点。

- 2). 代理模式, 比如 codis, 代理层通过把自己模拟成一个 redis, 接受来自客户端的请求, 然后按照自定义的路由逻辑进行数据分片和迁移。加入了 proxy 层, 对性能有一定的损耗, 多个 proxy 使用 lvs 进行负载均衡

- 3). redis cluster 是社区提供的原生集群方案, 官方当前推荐的方案, 主要特点是去中心化, 不需要 proxy 代理。主要的用途是实现数据分片 (Data Sharding), 同样可以实现 HA。

redis cluster 采用了 hash 槽的概念, 集群会预先分配 16384 个槽, 并将这些槽分配给具体的服务节点, 通过 Key 进行 CRC16(KEY)%16384 运算得到对应的槽是哪一个, 从而将读写操作转发到该槽对应的服务节点。当有新的节点加入或者一处的时候, 再来迁移这些槽以及对应的数据。更方便的动态扩容和缩容。

#### 49. redis 有哪些应用场景

- 1). 存储 json 类型的对象
- 2). 排行榜, 计数器
- 3). 优酷视频的点赞
- 4). 朋友圈点赞
- 5). 会话缓存

redis 能支持 10 万并发等

#### 50. 一个字符串类型的值能存储的最大容量是多少?

512M (官网数据)

#### 51. linux 日常命令操作

ls , pwd, chown, find, iftop top, ps, atop sed

#### 52. 日常工作服务搭建

nginx, nfs, ftp, apache, nginx 动静分离, nginx+keepalived , lvs+keepalived

mysql 主从搭建, mysql 主从加 mha 故障切换, tomcat, zabbix, redis 等 docker, k8s 等。

#### 53. shell 编程常用哪些脚本

自动备份, lnmp 一键安装, for 循环, case 语句, if 判断, 数据库自动备份脚本, 日常日志切割, 自动化安装 lnmp, 安装 nginx, redis, tomcat, 自动化一键安装, 远程部署脚本, 自动化发布脚本, 批量处理, 批量建站, 故障探测脚本

#### 54. 写出 Linux 实时查看日志变化的命令?

```
#tail -f *.log
```

#### 55. 写出 Linux 输出查找到某一个服务 (如 Java 相关的服务), 是否运行的命令

```
#ps -ef | grep tomcat
```

#### 56. 自动化部署, 自动化运维熟悉哪些?

自动化运维工具 熟悉 jenkins 、 gitlab、 salt 、 ansible

#### 57. 简单列出如何做容灾与恢复, 保证数据安全?

容灾我们在异地也部署一套模拟真实环境, 数据库每天进行全量备份, 推送异地服务器上, 以便于恢复使用

可以从三个层面入手: 操作系统; 应用系统; 数据库; 比较常用的是: 应用系统和数据库层面的安全保障措施

操作系统的设置, 防火墙使用

应用系统层次: 比如密码设置不能使用明文、登录权限分配、

数据库层面: 权限控制

#### 58. 公司是什么样的环境, Java 的还是 PHP

java php 都有

### 59. 做过集群，MySQL 主从之类的吗？

我们公司数据库架构

1 主 2 从，开启 mha 高可用

或者 1 主 2 从，用 mycat 进行读写分离，开启 1 写 2 台负责读

### 60. 源码用的是 git 还是 SVN？

代码是 git： 文件资料管理：svn

### 61. 你们公司做什么业务，有多少服务器，并发多少，怎么分工的，你负责多少服务器

20 台服务器 pv 200 万 UV3 万

### 62. 工作中部署的 mysql 主从架构，mysql 全备的数据量有多少？

这个没有工作的经验的说几十个 G 就行，说太大了不好。或者说 500G 内

### 63. TCP/IP 三次握手和四次挥手原理

#### 三次握手原理：

三次握手的本质是确认通信双方收发数据的能力

首先，我让信使运输一份信件给对方，对方收到了，那么他就知道了我的发件能力和他的收件能力是可以的。

于是他给我回信，我若收到了，我便知我的发件能力和他的收件能力是可以的，并且他的发件能力和我的收件能力是可以。

然而此时他还不知道他的发件能力和我的收件能力到底可不可以，于是我最后回馈一次，他若收到了，他便清楚了他的发件能力和我的收件能力是可以的。这就是三次握手

第一次握手：Client 将标志位 SYN 置为 1，随机产生一个值 seq=J，并将该数据包发送给 Server，Client 进入 SYN\_SEND 状态，等待 Server 确认。

第二次握手：Server 收到数据包后由标志位 SYN=1 知道 Client 请求建立连接，Server 将标志位 SYN 和 ACK 都置为 1，ack=J+1，随机产生一个值 seq=K，并将该数据包发送给 Client 以确认连接请求，Server 进入 SYN\_RECV 状态。

第三次握手：Client 收到确认后，检查 ack 是否为 J+1，ACK 是否为 1，如果正确则将标志位 ACK 置为 1，ack=K+1，并将该数据包发送给 Server，Server 检查 ack 是否为 K+1，ACK 是否为 1，如果正确则连接建立成功，Client 和 Server 进入 ESTABLISHED 状态，完成三次握手，随后 Client 与 Server 之间可以开始传输数据了

在三次握手过程中，Server 发送 SYN-ACK 之后，收到 Client 的 ACK 之前的 TCP 连接称为半连接（half-open connect），此时 Server 处于 SYN\_RECV 状态，当收到 ACK 后，Server 转入 ESTABLISHED 状态。

#### 四次挥手原理：

所谓四次挥手（Four-WayWavehand）即终止 TCP 连接，就是指断开一个 TCP 连接时，需要客户端和服务端总共发送 4 个包以确认连接的断开。在 socket 编程中，这一过程由客户端或服务端任一方执行 close 来触发，

由于 TCP 连接时全双工的，因此，每个方向都必须单独进行关闭，这一原则是当一方完成数据发送任务后，发送一个 FIN 来终止这一方向的连接，收到一个 FIN 只是意味着这一方向上没有数据流动了，即不会再收到数据了，但是在这个 TCP 连接上仍然能够发送数据，直到这一方向也发送了 FIN。首先进行关闭的一方将执行主动关闭，而另一方则执行被动关闭。

### 64. squid 缓存服务器维护过吗？squid 缓存代理的里是什么？缓存命中率怎么查看及清空缓存？

**squid 优化过：**cache\_mem 缓存占内存大小

Maximum\_object\_size 最大缓存块

Reply\_body\_max\_size 限定过下载文件大小

Access\_log /var/log/squid/access.log 指定一下日志存放文件

**squid 缓存代理：**

高性能的 web 缓存服务器，数据缓存在硬盘，缓存静态数据，接收客户端的请求，并在本地存储查找有没有客户端请求的数据，如果有，则直接响应给客户端，如果没有，则由 squid 代理服务器向服务端发送请求，请求的数据为客户端需要的数据，服务端响应代理服务器后，代理服务器将数据缓存到本地后响应给客户端

正向代理：面向客户端，接收客户端发送给服务器的数据，并响应给客户端，可以节省客户端的网络带宽，提高响应速度

反向代理：面向服务端，代理服务节点接收客户端的请求并响应给客户端，可以缓解服务器节点压力，提高服务端的安全性

squid 查看缓存命中率： # squidclient -p 3128 mgr:info

squid 清空缓存 # squidclient -h 主机名 -p 3128 -m PURGE 域名

补充：nginx 可开启缓存模块做缓存。

## 65.mysql sleep 线程过多如何解决?

### 1). 什么是 sleep 线程?

sleep 线程长时间保持可客户端与服务端的连接状态

导致 sleep 过多的原因:

使用太多持久连接(高并发系统中 不适合使用持久连接)

程序中 没有及时关闭 MySQL 连接

数据库优化不完善 导致执行 sql 语句过慢

### 2). 解决方法:

wait\_timeout 和 interactive\_timeout 的参数应该调整小一点。

1). wait\_timeout: 服务器关闭非交互连接之前等待活动的秒数。参数默认值: 28800 秒 (8 小时)

2). interactive\_timeout: 服务器关闭交互式连接前等待活动的秒数。参数默认值: 28800 秒 (8 小时)

想当时就生效利用直接在数据库里设置:

```
set global wait_timeout = 120 #临时全局生效
```

```
set wait_timeout = 120 #临时生效
```

永久生效: vim /etc/my.cnf 配置文件里进行配置

#### 【mysqld】

```
interactive_timeout = 120 #设置服务器关闭交互式连接前等待活动的秒数. 系统默认是 8 小时
```

```
wait_timeout = 120 #设置服务器关闭非交互连接之前等待活动的秒数. 系统默认是 8 小时
```

说明:

mysql 服务器所支持的最大连接数是有上限的，因为每个连接的建立都会消耗内存，因此我们希望客户端在连接到 mysql-server 处理完相应的操作后，应该断开连接并释放占用的内存。如果你的 mysql-server 有大量的闲置连接，他们不仅会白白消耗内存，而且如果连接一直在累加而不断开，最终肯定会达到 mysql-server 的连接上限数，这会报 'too many connections' 的错误。对于 wait\_timeout 的值设定，应该根据系统的运行情况来判断。在系统运行一段时间后，可以通过 showprocesslist 命令查看当前系统的连接状态，如果发现大量的 sleep 状态的连接进程，则说明该参数设置的过大，可以进行适当的调整小些。

## 66. 写 shell 脚本防范 ddos 攻击的思路

日志获取异常 ip 然后使用防火墙拉入黑名单，也能用 nginx 限制 ip 的并发连接数，防止 ddos 攻击

## 67. 你在工作中都遇到过哪些故障? 怎么解决的? 日常工作中处理频率最多的是哪方面问题? 怎么高效解决的?

### 网站访问慢排查故障:

1). 首先要确定是用户端还是服务端的问题。当接到用户反馈访问慢，那边自己立即访问网站看看，如果自己这边访问快，基本断定是用户端问题，就需要耐心跟客户解释，协助客户解决问题。

2). 如果自己访问也慢，那么可以利用浏览器的调试功能，看看加载那一项数据消耗时间过多，是图片加载慢，还是某些数据加载慢。排查静态资源问题还是动态某些数据加载慢问题。

- 3). 针对服务器负载情况。查看服务器硬件(网络、CPU、内存)的消耗情况。如果是购买的云主机，比如阿里云，可以登录阿里云平台提供各方面的监控，比如 CPU、内存、带宽的使用情况。有无异常进程，有无攻击等。
- 4). 如果发现硬件资源消耗都不高，那么就需要通过查日志，比如看看 MySQL 慢查询的日志，看看是不是某条 sql 语句查询慢，导致网站访问慢。

#### 怎么去解决：

- 1). 如果是出口带宽问题，那么就申请加大出口带宽。
- 2). 如果慢查询比较多，那么就要开发人员或 DBA 协助进行 SQL 语句的优化。
- 3). 如果数据库响应慢，考虑可以加一个数据库缓存，如 Redis 等等。然后也可以搭建 MySQL 主从和读写分离，一台 MySQL 服务器负责写，其他几台从数据库负责读。
- 4). 申请购买 CDN 服务，加载用户的访问。
- 5). 如果访问还比较慢，那就需要从整体架构上进行优化。做到专角色专用，多台服务器提供同一个服务。

#### 68. tomcat 多实例怎么配置？

tomcat 多实例：可以不同端口号，多虚拟主机，自定义 web 应用访问位置 想要多实例 在编译安装一次 tomcat 就行，只不过和第一个 tomcat 的路径不同，端口不同

#### 69. 请说出这些符号的含义 < << >> > 2> 2>> 2>&1 &&

- < 或 << #标准输入
- >, >>, 1> 或 1>> #标准输出
- >> #标准输出追加在后面
- > #标准输出覆盖添加
- 2> 或 2>> #标准错误输出
- 2>&1 #将标准错误输出重定向到标准输出
- && #一个表达式不成立的话，后面的表达式不运算，前面执行成功，后面才执行

#### 70. http 协议通信原理

- 1). web 浏览器请求 DNS 服务器把域名 www.etiantian.org 解析成 web 服务器的 IP 地址，此处就是 DNS 解析过程。
- 2). web 浏览器通过解析后的 IP 地址及端口号与 web 服务器之间建立一条 TCP 连接。
- 3). 建立连接后浏览器向 web 服务器发送一条 HTTP 报文请求，web 服务器读取浏览器的请求信息，并返回一条 HTTP 响应。
- 4). 客户端接收服务器返回的信息通过浏览器显示在用户显示屏上，然后客户机与服务器断开 http 连接，断开 TCP 连接。

#### 71. 你们的数据量多大，平时怎么备份的

现在 500G 还是逻辑备份 准备使用 xbk 备份，节约备份时间。

#### 72. 日志分析怎么做的，都关注哪些参数

答：efk 搜集：nginx, tomcat, redis, docker 等日志。

#### 73. MySQL 你都负责维护哪些内容

- 1). 权限管理
- 2). mysql 慢日常抓取协助开发分析

3).mysql 优化

**74. shell 熟悉吗，写过哪些 shell 脚本**

答：mysql 备份脚本、tomcat 故障自愈脚本、nginx 日志切割脚本、自动化安装环境脚本、自动发布项目脚本、批量建站脚本、远程部署服务、一键安装

**75. docker 在公司的应用是啥**

跑 java 应用

**76. 在工作中遇到过什么棘手的问题，怎么解决的，学到了啥**

tomcat 内存溢出

临时解决：写脚本实时探测如果发现内存快要溢出时候就重启 tomcat

永久解决：分析原因，通过监控和分析日志，找到到底什么问题造成，我们当时经过分析是

有几个原因 1). 系统内核需要优化 2). tomcat 需要优化 3). 2 台 tomcat 处理不过来 4). 开发代码也有问题, 最好借助消息队列使用 MQ

**77. 你最熟悉的项目是哪个，什么时间做的，做了多久，几个人，你主要在项目当中负责啥**

gitlab+jenkins 自动发布项目 我一个人慢慢改进 脚本改进的

**78. 有没有做过数据库升级，怎么做的**

先测试环境升级，并写升级文档，

正式环境，做好系统快照和备份 然后根据写的升级文档升级，

docker 运行的 mysql 升级到高版本，先备份数据目录，再换成高版本镜像重新运行

**79. mysql 用的哪个版本，nginx 哪个版本**

答：mysql 5.7.29 nginx 1.16.1

**80. 如果网站出现 500 错误，你的解决思路是？**

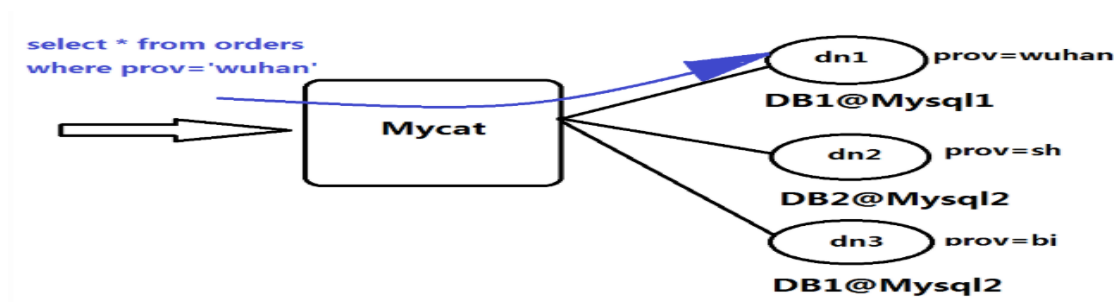
网站报 500 错误一般数据库出现问题

检查下为啥连接不上数据库，端口不通还是账号和密码问题，防火墙问题，数据库慢查询是否太多，有无锁表, 数据库是否挂了

**81. 读写分离用的什么工具，原理是啥**

mycat

原理：MyCAT 主要是通过对 SQL 的拦截，然后经过一定规则的分片解析、路由分析、读写分离分析、缓存分析等，然后将 SQL 发给后端真实的数据块，并将返回的结果做适当处理返回给客户端



## 82. mysql 主从原理是啥

答: mysql 主从同步就是异步复制过程, 主库开启 i/o 线程, 从库开启 i/o 线程和 sql 线程

主库, 增、删、改 生成二进制日志

从库 i/o 线程去请求主库 binlog 日志, 并把主库 binlog 日志复制到从库, 然后写到中继日志里面, SQL 线程会读取 relay log 文件中的日志, 进行 sql 语句回放操作, 来实现主从的操作一致, 而最终数据一致。

## 83. DNS 原理是啥

访问网站过程 (以访问百度为例)

答案: 假定想访问 www.baidu.com。下面是的几个查询步骤:

主机先查询本地 DNS 缓存及 hosts 文件是否有域名对应 IP 地址, 如果有, 就先调用这个 IP 地址映射, 完成域名解析。

如果 hosts 与本地 DNS 解析器缓存都没有相应的网址映射关系, 首先会找 TCP/IP 参数中设置的首选 DNS 服务器, 在此我们叫它本地 DNS 服务器, 此服务器收到查询时, 如果要查询的域名, 包含在本地配置区域资源中, 则返回解析结果给客户机, 完成域名解析, 此解析具有权威性; 如果要查询的域名, 不由本地 DNS 服务器区域解析, 但该服务器已缓存了此网址映射关系, 则调用这个 IP 地址映射, 完成域名解析, 此解析不具有权威性。

如果本地 DNS 服务器本区域文件与缓存解析都失效, 则根据本地 DNS 服务器的设置 (是否设置转发器) 进行查询, 如果未用转发模式, 本地 DNS 就把请求发至 13 台根 DNS, 根 DNS 服务器收到请求后会判断这个域名 (.com) 是谁来授权管理, 并会返回一个负责 .com 域名解析的顶级域名服务器的一个 IP。

这台负责 .com 域的服务器收到请求后, 如果自己无法解析, 它就会找一个负责 .baidu.com 域名解析的权限 DNS 服务器地址给本地 DNS 服务器

本地域名服务器然后向权限域名服务器 dns.baidu.com 进行查询。权限域名服务器 dns.baidu.com 告诉本地域名服务器所查询的域名的 IP 地址。本地域名服务器最后把查询结果告诉本地主机。完成访问过程

## 84. 当前数据库服务处理速度慢, 你认为可能是什么原因导致的, 并阐述一下对应的解决办法

硬件配置低、网络带宽窄、

网络架构不合理 (数据包的进出有瓶颈, 是否有读写分离, 是不是读写都在一台扛不住, 是不是一主两从)  
程序代码有问题 (开启慢查询日志)

是否有缓存 redis 了

服务软件版本低 show variables; show status;

软件优化: 打开最大文件数..., 是不是参数连接数需要调大, keybuffer 参数是否合理, 存储引擎是否合适。

## 85. 说一下你知道的存储引擎, 建表时如何决定表使用存储引擎

innodb myisam

## 86. 单张表过大有什么缺点, 如何解决

缺点: 查询会很慢

对表进行分区、分表、分库

实在太大了 只能上大数据库 hoooop 家族、hbase

**87. 给你 50 台服务器，阐述一下你的拓扑结构如何设计**

前端使用 5 台多 dns 轮训负载均衡

中间：使用 10 台应用

中间件： redis 集群 6 台 kafka 集群 6 台

db 层：3 台 mycat 10 台数据库

图片存储服务器 2 台： nfs+rsync

日志收集系统： elk 3 台

监控 2 台

代码仓库 1 台

cicd 一台

**88. 如果你是数据库管理员，阐述一下你如何给服务器添加授权用户**

用户设置 ip 授权 和 读取权限

**89. 若你是数据库管理员，阐述一下你如何对数据做备份**

数据量小时使用逻辑备份，数据库大时候使用 xbk，比较快，不会锁表

**90. 目前常见的数据库软件有哪些，搭建数据库集群的时候，你如何选择使用哪种软件**

mgr 集群或者 mha

**91. 配置 mysql 主从同步前，如何保证主库与从库的数据一致**

从库配置高于主库，因为从库压力可能比较大，而且还得及时将主库的数据同步过来。

开启多线程复制

主库快速 d11 执行

或者 架构调整 使用 MGR 集群模式使用组复制

**92 如何获知一台主机上的网站服务是否在运行**

#ps -ef | grep 进程 或 #netstat -anput |grep 端口

**93. 如何让系统的普通用户拥有 root 权限，什么情况下需要这么做**

visudo 里面授权

#www ALL=(ALL) ALL

www ALL= (root) NOPASSWD: /sbin/mount, (root) NOPASSWD: /bin/umount, (root) NOPASSWD: /bin/touch, (root) NOPASSWD: /bin/rm, (root) NOPASSWD: /usr/bin/make, (root) NOPASSWD: /bin/ln, (root) NOPASSWD: /bin/sh, (root) NOPASSWD: /bin/mv, (root) NOPASSWD: /bin/chown, (root) NOPASSWD: /bin/chgrp, (root) NOPASSWD: /bin/cp, (root)

**94. 杀死进程的命令有哪些？**

#kill -9 进程号或名

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：

<https://d.book118.com/858070001121006052>