- **haproxy的工作原理**
- haproxy可以做四层的负载均衡也可以做七层应用层的。四层的话主要是做数据库从服务器的负载均衡，客户端发送请求后，haproxy会根据调度算法指定一个后端服务器，在客户端和该服务端之间进行双向流量的转发，充当一个类似路由器的转发角色。七层一般来做web服务器的负载均衡，客户端发送请求后，haproxy能根据用户的请求信息和特定的规则来指定后端服务池，再根据服务池的调度算法指定后端服务器。
- 四层和七层的tcp三次握手的过程不一样，四层的话首先客户端发送syn请求，haproxy就会直接将请求转发给后端服务器，服务器返回ack信息给haproxy，haproxy再返回给客户端ack信息，这样的话实际上就是客户端与服务端直接建立的连接。而七层是这样的，客户端发送syn请求，haproxy会直接给客户端返回ack信息经过三次握手建立连接，然后分析客户端发送的信息再根据调度算法指定后端服务器，和其进行三次握手建立连接传输信息。这样的话，在面对ddos攻击的时候七层会比四层更加安全，因为ddos攻击的原理是这样的，它利用了tcp实现上的一个缺陷，首先它发送大量的syn请求给服务端，但是在握手的第三步时不给服务端发送确认连接的信息，导致服务器不停地重试并等待一个同步超时的时间，这样就会占用服务器大量的资源和带宽等等，严重影响到正常用户的访问。如果是四层，ddos攻击会越过haproxy直接攻击到后端的服务器，但是七层的话攻击只会影响到haproxy，从而达到保护后端服务器的作用。
- **nginx的try_files是干嘛的，nginx如何设置虚拟目录，那么alias和root有什么区别?**
- try_files可以按给定的顺序检查文件或目录是否存在，会返回第一个找到的文件或目录。如果所有的文件或目录都没找到，则会调用fallback中指定的位置来处理请求。它可以用来替代比较繁琐的rewrite功能。
- 设置虚拟目录的话，可以使用location匹配一个url目录，然后在下面使用alias指定真实目录，这样的话就会用匹配的目录掩盖真实的目录，起到一定的保护作用。
- 除了alias外，root也可以做到一个效果。但是alias和root是有区别的，alias的指定是精确的，有点类似于软连接的ln -s命令，就是匹配的是什么，就把那个给换成alias指定的内容，匹配的url后面加了/表示目录，那么alias指定的也要加上/，不然就会访问失败。root指定的目录是location指定path目录的上一级目录，这个path目录一定是存在于root指定的目录下面的。root目录配置时，location匹配的path加不加/都不会影响到访问。所以我们可以在location匹配的是文件的时候使用alias，在匹配目录的时候使用root。
- **redis了解多少?**

- redis是一个非关系型的数据库。它是基于键值对的形式存储数据的。redis一般用来做缓存服务器，缓存服务器还有memcache，mongodb等等，其中mongodb占用空间过大，在性能上还不如前两者；memcache的话性能很好，但是它不支持持久化；redis是支持持久化的，有RDB和AOF两种方式，现在AOF持久化用的多一点，它还可以设置主从来解决单点故障，使用哨兵模式实现高可用。
- **解释一下https**
- https是http的安全版本，它在http下加入了ssl层，ssl依靠证书来验证服务器的身份，并为浏览器和服务器之间的通信进行加密。https和http的端口不一样，https的端口是443，大部分的https证书需要付费，免费的很少。我们之前有配置过nginx以及haproxy的https，https会先经过443端口进行验证，然后再传输数据。
- 我们做实验用的是x509自签名生成的证书。先生成服务的私钥，openssl genrsa 指定加密方式比如des3 然后-out输出为key文件。之后创建签名请求的证书，openssl req -key指定刚刚生成的密钥，-out输出为csr文件，使用openssl rsa -in可以将原来的key文件-out输出为不要口令的纯文字版，免去麻烦，最后使用x509进行自签名得到证书。openssl x509 -days表示证书有效时间，-in请求的csr文件，-signkey加入key文件 -out就是crt证书了。
- 拿nginx为例，配置的话需要将443端口进行监听，在后面将ssl on打开，还有两个参数要配置，ssl_certificate,ssl_certificate_key分别指定证书路径和密钥key的路径，nginx的https就配置完成了。
- **nginx的epoll模型是什么?**
- 它是一种io模型，与apache的select模型不同，select是通过轮询检测的方式找到所有状态改变的描述符，然后再进行网络io处理，epoll模型是这样的，它无需将所有的描述符集进行检测，只要处理那些被内核io事件异步唤醒而加入ready队列的描述符就行了。因此，随着并发量的提高，select的机制会使性能急速下降，而epoll模型却几乎不会受到影响。
- **nginx服务报错too many open files如何处理**
- 可以在nginx配置里加入work_rlimit_nofile 将值设置为65535，这是文件句柄数的最大值。当然，如果操作系统的文件句柄数没有设置的话这是没有办法实现的。修改操作系统的文件句柄数首先是临时生效的方法，使用ulimit 命令 -n 65535，然后需要将其永久生效，在/etc/security/limits.conf里面加入nofile 指定软硬都是65535就好了。
- **lamp和lnmp有什么区别，lnmp怎么搭建的，软件安装的先后次序?**
- lamp就是linux+apache+mysql+php环境的意思，lnmp就是将apache换成了nginx，有一些服务需要lnmp环境，我之前有做过的zabbix监控就是基于lnmp环境的。现在的话安装次序不是很重要，早先的版本一定要先装mysql再装php，但是现

在不需要了，只需要在php的编译时加入--with-mysqli=mysqlnd --with-pdo-mysql=mysqlnd就行了。

- **描述一下整个项目？以及这个架构最大的优点**
- 我们的架构是这样的，外部是CDN，包括一台智能DNS和三台squid缓存。源站是nginx做七层代理负载均衡动静分离，一台专门的nginx静态服务器，四台tomcat动态服务器，之后是mysql的一主两从，两个从库使用了haproxy的四层负载均衡加keepalived高可用，前面还加入了两台redis存储服务器，一主一从，使用了哨兵模式保证高可用，还使用了aof持久化。这是线上环境，我们还有测试环境，使用了svn+maven+jenkins+ansible+tomcat的架构，还有一个专门的版本发布平台。
- 这个架构最大的优点就是高扩展性和高冗余性。
- **发布平台有哪些功能?**
- 我们可以指定版本和工作组将产品发布到线上环境，我们的四台tomcat分为了两个工作组，分别承担不同的业务；还可以远程将某个工作组的服务重启；还可以将版本进行备份，方便出现问题时能快速回滚。