为什么要用Nginx？

因为跨平台，配置简单，支持高并发，耗费内存少。

为什么Nginx性能这么高？

因为nginx是异步非阻塞事件处理机制：运用了epoll模型，提供了一个队列，排队解决

使用**异步非阻塞**模式（类似多路复用器）。  
在Linux 环境中使用 **epoll 模型**，来处理工作进程。能处理6-8万个请求

Nginx怎么处理请求的？

nginx接受一个请求后，首先由listen和server\_name 匹配server模块，在匹配的server模块里location，location就是实际地址

什么是正向代理和反向代理？

正向代理：客户端向代理服务器发送一个请求并指定目标，然后代理服务器向目标服务器转交请求并将获得的内容返回给客户端（访问国外网站）

反向代理：代理服务器来接受请求，然后将请求转发给内网服务器，再将获得的内容返回给客户端。

反向代理的用途

隐藏服务器真实IP

负载均衡

正向代理和反向代理的区别

正向代理代理的是客户端，反向代理代理的是服务器

正向代理解决访问限制问题，反向代理解决了高并发问题

Nginx负载均衡的算法怎么实现的?策略有哪些?

1.轮询：如果有服务器宕机，能够自动剔除

2.权重：将服务器性能高的权值设置大点。

3.ip绑定：使同一ip的访客固定访问一台服务器

Nginx 进程模型，如何修改worker 进程数？

当Nginx 启动后，默认会启动两个进程。  
1.Master 进程  
2.Work 进程

在 conf/nginx.conf 文件中修改 worker\_processes 数字即可。

nginx的常用算法实现？

轮询（默认）：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。

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

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。

Fair:按后端服务器的响应时间来分配请求，响应时间短的优先分配。

动态资源、静态资源分离的原因

动态资源、静态资源分离是让动态网站里的动态网页根据一定规则把不变的资源和经常变的资源区分开来，动静资源做好了拆分以后，我们就可以根据静态资源的特点将其做缓存操作，这就是网站静态化处理的核心思路

动态资源、静态资源分离简单的概括是：动态文件与静态文件的分离

二者分离的原因

在我们的软件开发中，有些请求是需要后台处理的（如：.jsp,.do等等），有些请求是不需要经过后台处理的（如：css、html、jpg、js等等文件）

这些不需要经过后台处理的文件称为静态文件，否则动态文件。因此我们后台处理忽略静态文件。这会有人又说那我后台忽略静态文件不就完了吗

当然这是可以的，但是这样后台的请求次数就明显增多了。在我们对资源的响应速度有要求的时候，我们应该使用这种动静分离的策略去解决

动、静分离将网站静态资源（HTML，JavaScript，CSS，img等文件）与后台应用分开部署，提高用户访问静态代码的速度，降低对后台应用访问

这里我们将静态资源放到nginx中，动态资源转发到tomcat服务器中

## Nginx常用的功能模块

·1.upstream负载均衡模块

## upstream模块的负载均衡算法主要有三种，轮调(round-robin)、ip哈希(ip\_hash)和 最少连接(least\_conn)三种。

## ip\_hash：基于客户端IP地址完成请求的分发，它可以保证来自于同一个客户端的请求始终被转发至同一个upstream服务器；

## round-robin ： 基于 权重的轮询。不需要明确注明，在各个节点中设置 权重 weight 即可。

## least\_conn： 最少连接调度算法；

## 2.proxy-pass代理转发模块

## 在nginx中配置proxy\_pass代理转发时，如果在proxy\_pass后面的url加/，表示绝对根路径；如果没有/，表示相对路径，把匹配的路径部分也给代理走。

## Nginx的核心组成

·nginx二进制可执行文件

·nginx.conf配置文件

·error.log错误的日志记录

·access.log访问日志记录