# [讲讲亿级PV的负载均衡架构](https://www.cnblogs.com/liliuguang/p/10370671.html)

## 引言

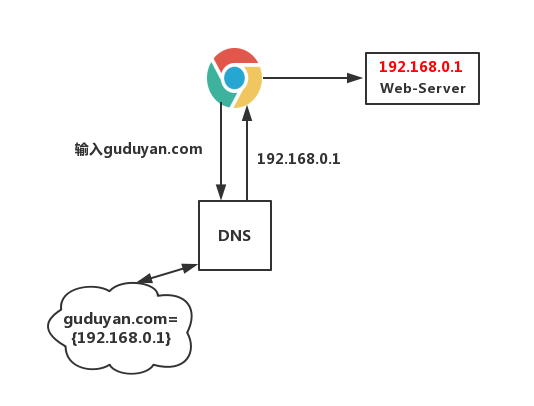
本来没想写这个题材的，为了某某童鞋能够更好的茁壮成长，临时写一篇负载均衡的。负载均衡，大家可能听过什么3层负载均衡、4层负载均衡、7层负载均衡什么的？那这是怎么分的呢，ok，是根据osi七层网络模型来分的，例如nginx是工作在应用层，应用层刚好是在第7层，因此nginx又可以称为7层负载均衡。  
我本来想一层层慢慢讲，从最基础的网络协议开始讲起，想了想又觉得这种讲法不适合速成。因此我改变思路，直接讲负载均衡架构的演进，最后的成品就可以在面试中侃一侃，因为现在负载均衡基本都是这套架构！。

## 正文

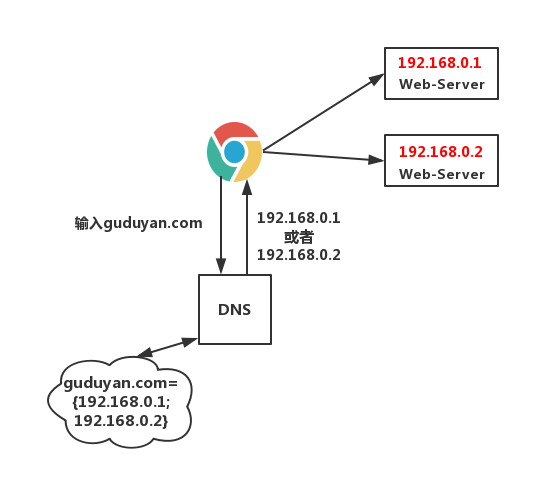
### DNS

开始呢，我们的应用只有一台web-server。

那么你希望:输入guduyan.com就能定位到该server

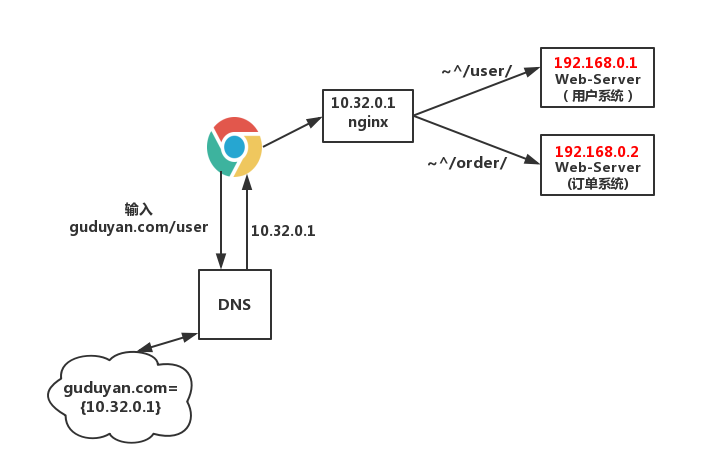
那很简单，只要在DNS里配上域名和你的server映射关系，就能访问到啦！  
流程如下图所示  


好，现在呢，多了一台web-server，你就可以通过在DNS里加一条配置，以DNS轮询方式进行负载均衡。

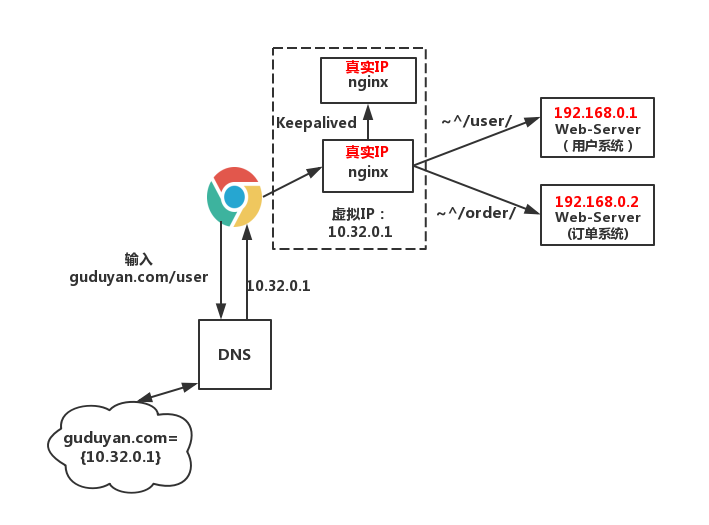
如下图所示  


### Nginx+DNS

现在假设，我们多了一些需求啊。你的系统按照功能模块拆成两个系统:用户系统和订单系统。那么你希望输入guduyan.com/user/的时候定位到用户系统。输入guduyan.com/order/的时候定位到订单系统。

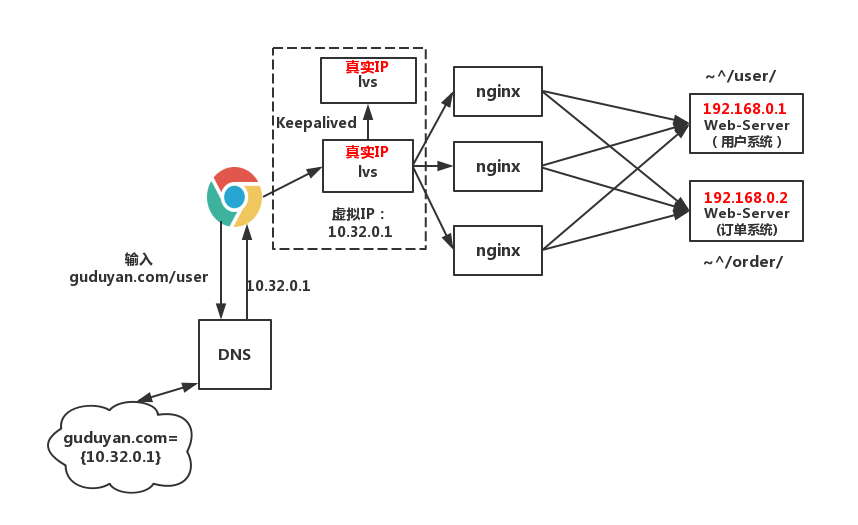
那这时候，光靠DNS就不行了，就需要采用DNS+nginx进行负载均衡！如下图所示  


ps:nginx还可以做动静分离哦，大家应该懂的！

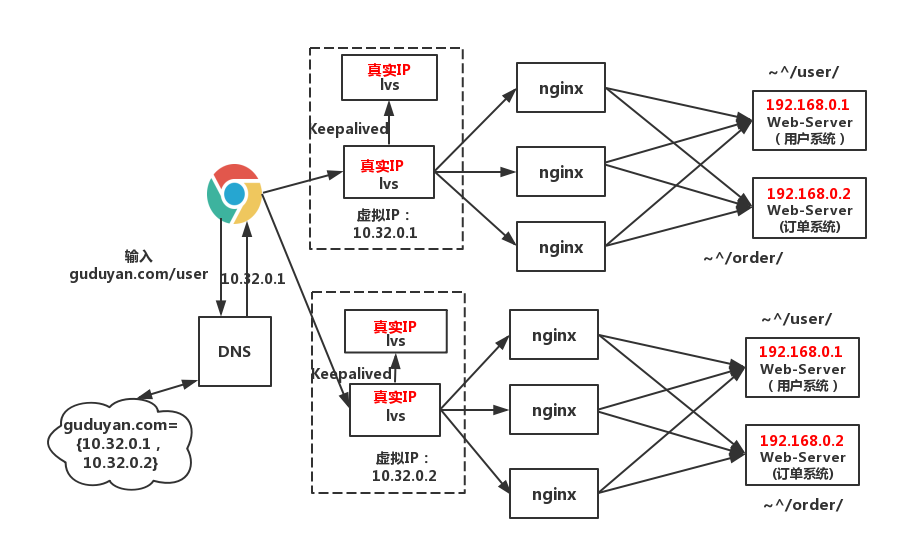
那如果系统的访问压力进一步加大，万一nginx挂了怎么办？如何给nginx引入热备？  
这里就要用keepalived了，用两台nginx组成一个集群，分别部署上keepalived，设置成相同的虚IP，这样一个节点在崩溃的情况下，另一个节点能够自动接替其工作，如下图所示  


### Lvs+Nginx+DNS

接下来随着系统规模的继续增大，你会慢慢的发现nginx也扛不住了！nginx工作在网络的第7层，所以它可以针对http应用本身来做分流策略，比如针对域名、目录结构等。  
而Lvs工作在网络4层，抗负载能力强，性能高，能达到F5的60%，对内存和CPU资源消耗比较低，且稳定，可靠性高。它利用linux的内核进行转发，不产生流量。它能撑的并发量取决于机器的内存大小，一般来说撑个几十万并发问题不大！  
ps:好好思考为什么会出现nginx+Lvs被同时使用，注意看我演变的过程，面试必问！注意了，如果是比较小的网站（日pv<1000万），用nginx就完全可以了，如果机器也不少，可以用dns轮询，Lvs所耗费的机器还是比较多的；大型网站或者重要的服务，机器不发愁的时候要多多考虑利用Lvs。

那么，在这种情况下的架构图如下所示  


可能有个疑问，为什么nginx层不用keepalived做热备？  
主要原因是:  
在这种架构下，nginx不是单台，如果nginx挂了，Lvs会帮你转发到其他可用的nginx上！

最后，为了应对亿级的PV，一般会在DNS端配多个Lvs集群的地址。如下所示  


方案扩展到了这一步，Lvs层就没有必要再进行扩展新的节点了。这套架构已经能扛得住亿级的PV。当然，前提是你的应用没问题！