

# 负载均衡

集群中的应用服务器（节点）通常被设计成无状态，用户可以请求任何一个节点。

负载均衡器会根据集群中每个节点的负载情况，将用户请求转发到合适的节点上。

负载均衡器可以用来实现高可用以及伸缩性：

- 高可用：当某个节点故障时，负载均衡器会将用户请求转发到另外的节点上，从而保证所有服务持续可用；
- 伸缩性：根据系统整体负载情况，可以很容易地添加或移除节点。

负载均衡器运行过程包含两个部分：

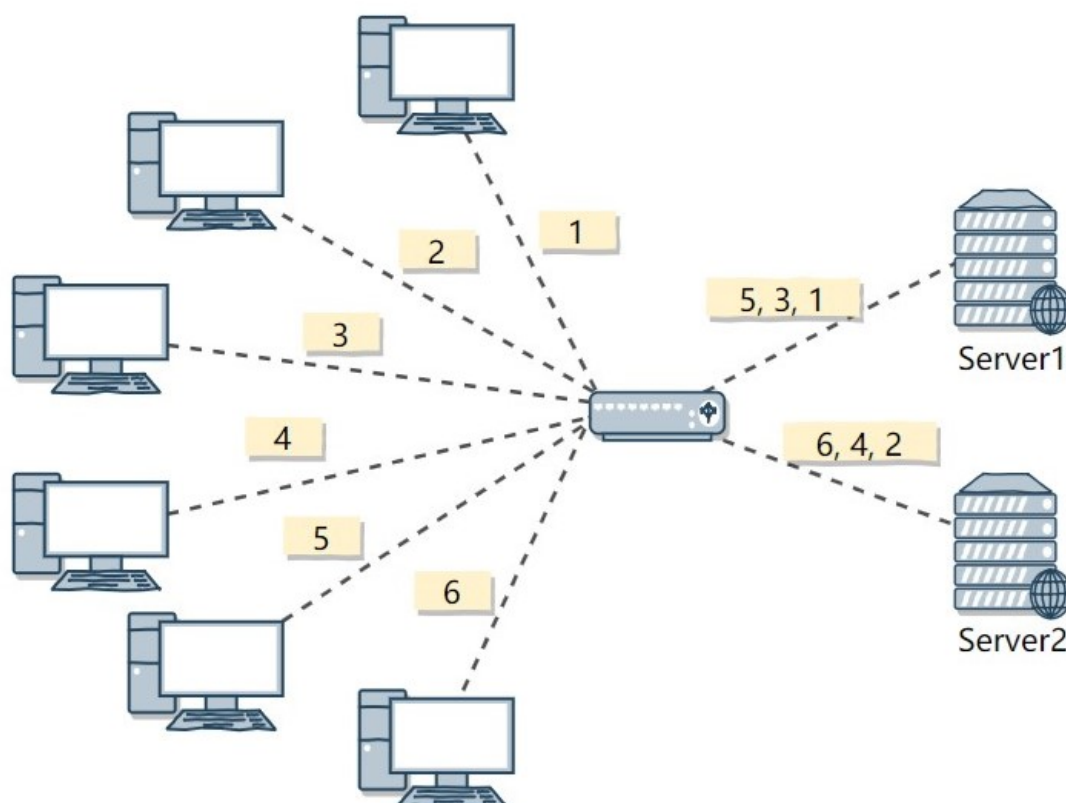
1. 根据负载均衡算法得到转发的节点；
2. 进行转发。

## 负载均衡算法

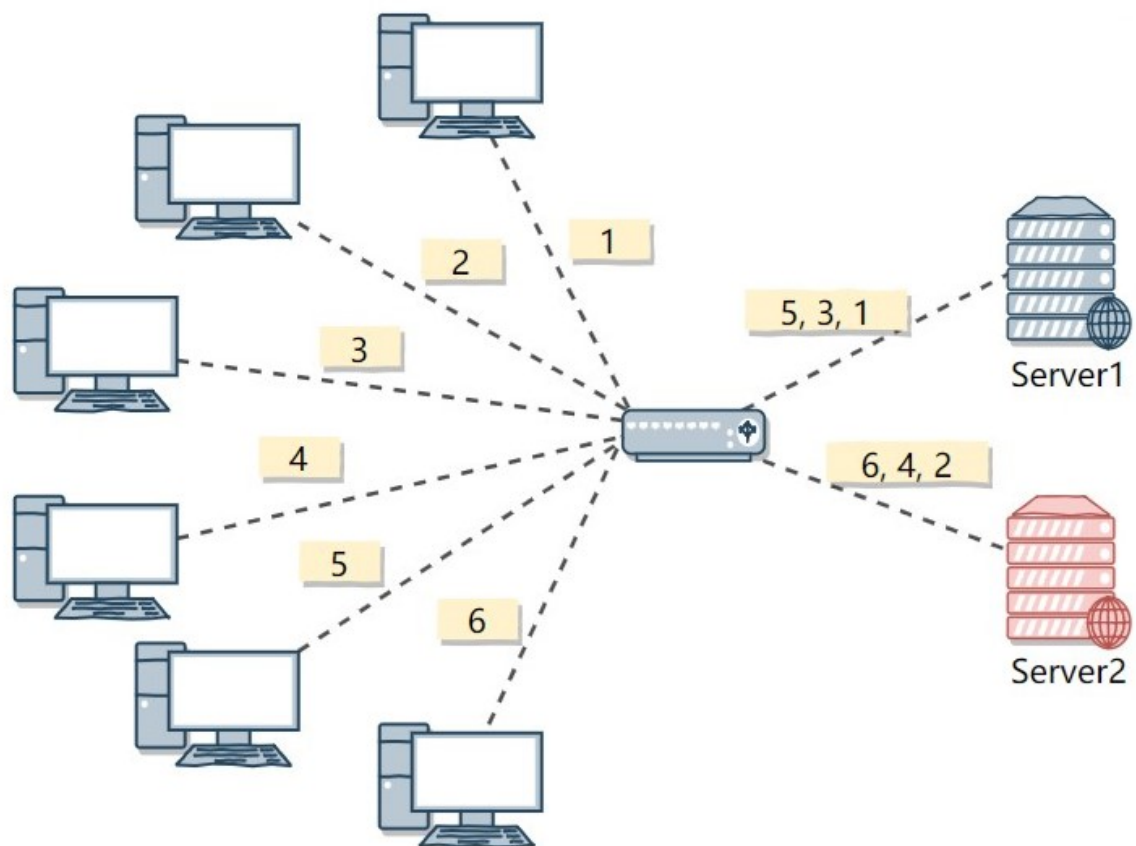
### 1. 轮询 (Round Robin)

轮询算法把每个请求轮流发送到每个服务器上。

下图中，一共有 6 个客户端产生了 6 个请求，这 6 个请求按 (1, 2, 3, 4, 5, 6) 的顺序发送。(1, 3, 5) 的请求会被发送到服务器 1，(2, 4, 6) 的请求会被发送到服务器 2。



该算法比较适合每个服务器的性能差不多的场景，如果有性能存在差异的情况下，那么性能较差的服务器可能无法承担过大的负载（下图的 Server 2）。

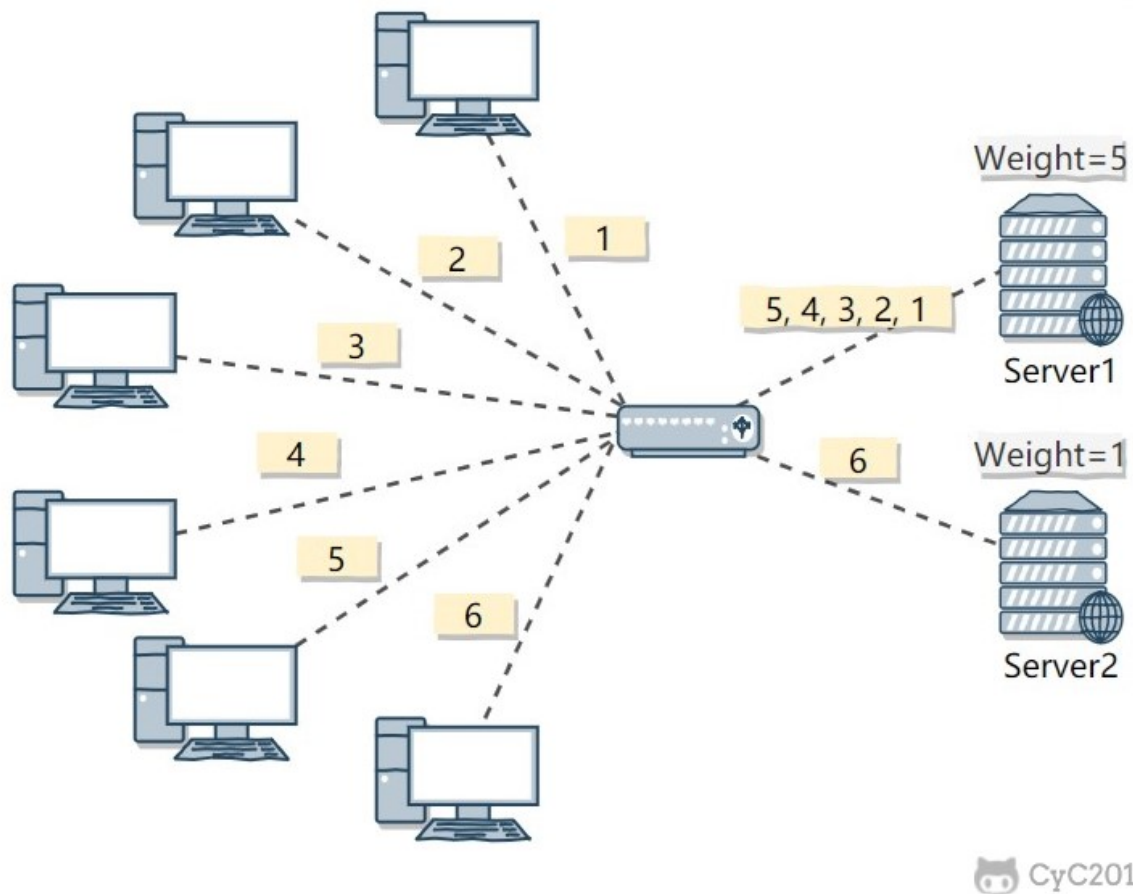


CyC2018

## 2. 加权轮询 (Weighted Round Robin)

加权轮询是在轮询的基础上，根据服务器的性能差异，为服务器赋予一定的权值，性能高的服务器分配更高的权值。

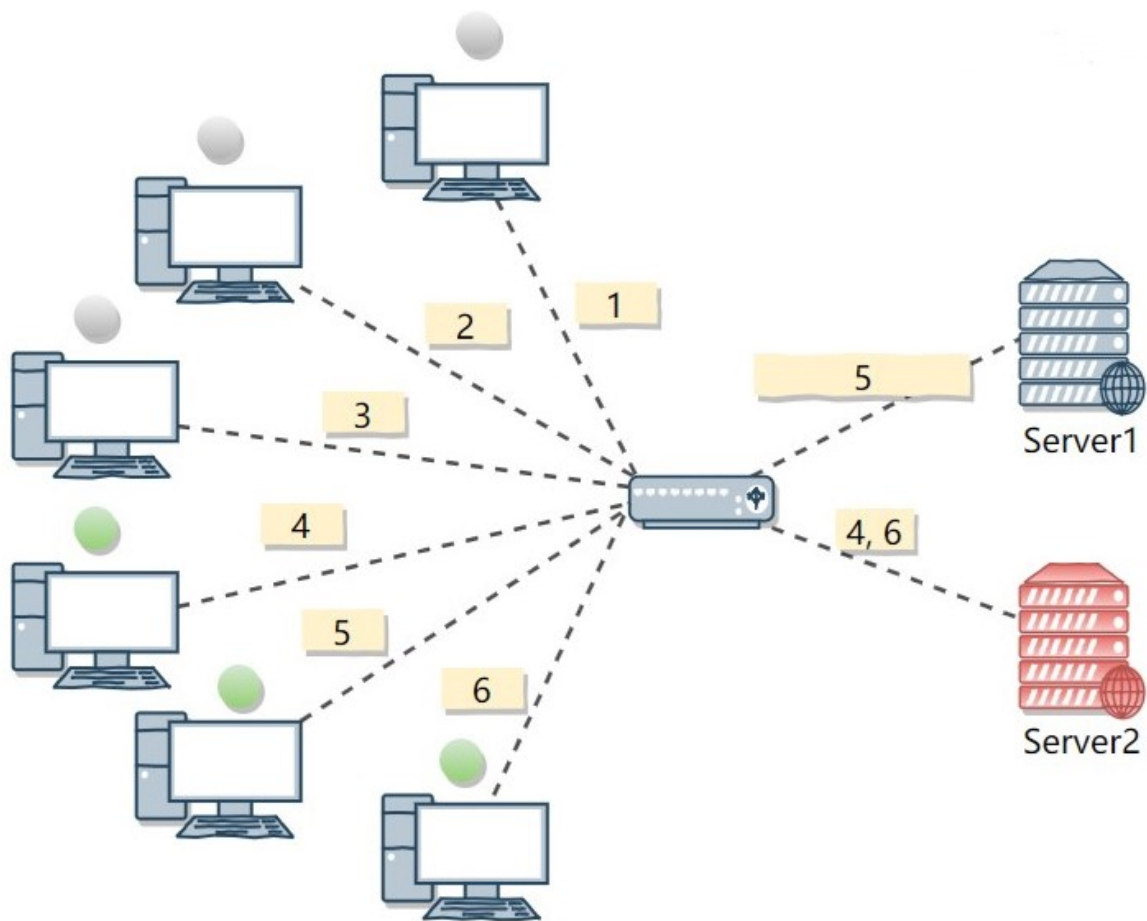
例如下图中，服务器 1 被赋予的权值为 5，服务器 2 被赋予的权值为 1，那么 (1, 2, 3, 4, 5) 请求会被发送到服务器 1，(6) 请求会被发送到服务器 2。



### 3. 最少连接 (least Connections)

由于每个请求的连接时间不一样，使用轮询或者加权轮询算法的话，可能会让一台服务器当前连接数过大，而另一台服务器的连接过小，造成负载不均衡。

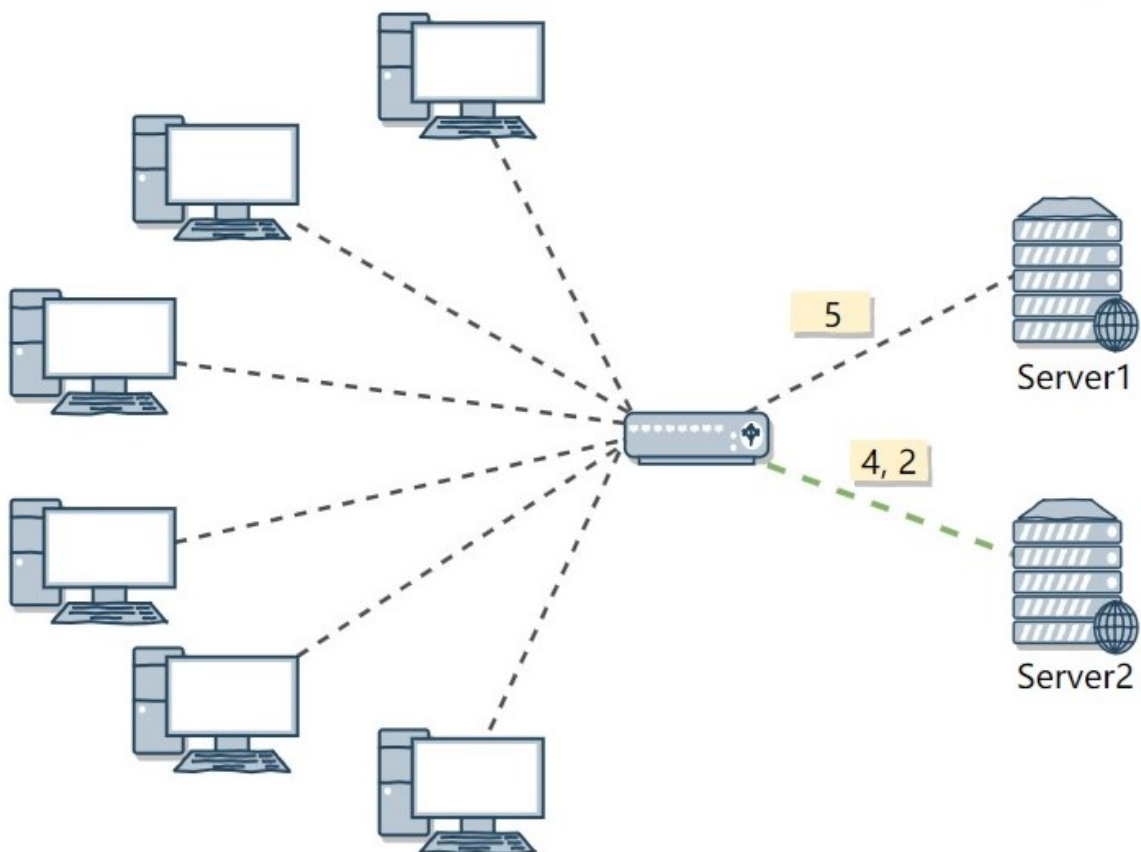
例如下图中，(1, 3, 5) 请求会被发送到服务器 1，但是 (1, 3) 很快就断开连接，此时只有 (5) 请求连接服务器 1；(2, 4, 6) 请求被发送到服务器 2，只有 (2) 的连接断开，此时 (6, 4) 请求连接服务器 2。该系统继续运行时，服务器 2 会承担过大的负载。



CyC2018

最少连接算法就是将请求发送给当前最少连接数的服务器上。

例如下图中，服务器 1 当前连接数最小，那么新到来的请求 6 就会被发送到服务器 1 上。



CyC2018

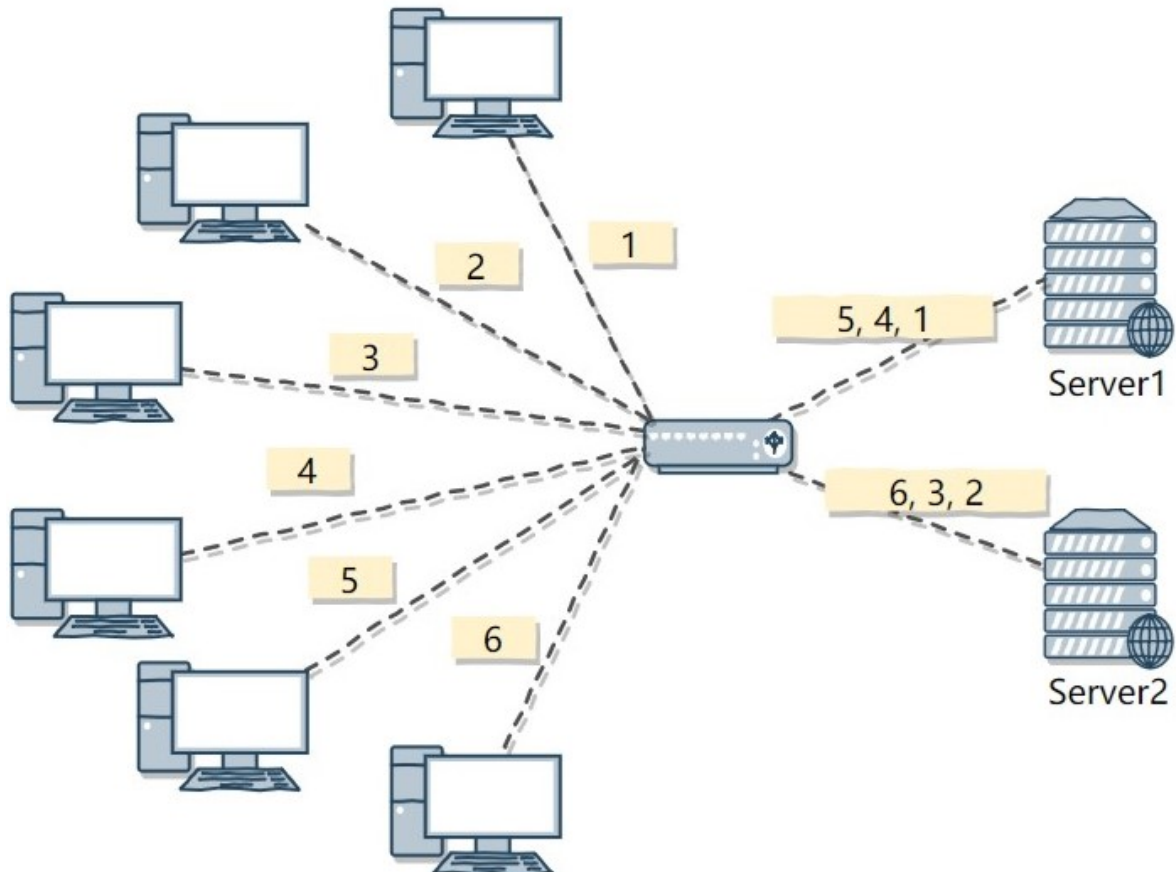
## 4. 加权最少连接 (Weighted Least Connection)

在最少连接的基础上，根据服务器的性能为每台服务器分配权重，再根据权重计算出每台服务器能处理的连接数。

## 5. 随机算法 (Random)

把请求随机发送到服务器上。

和轮询算法类似，该算法比较适合服务器性能差不多的场景。

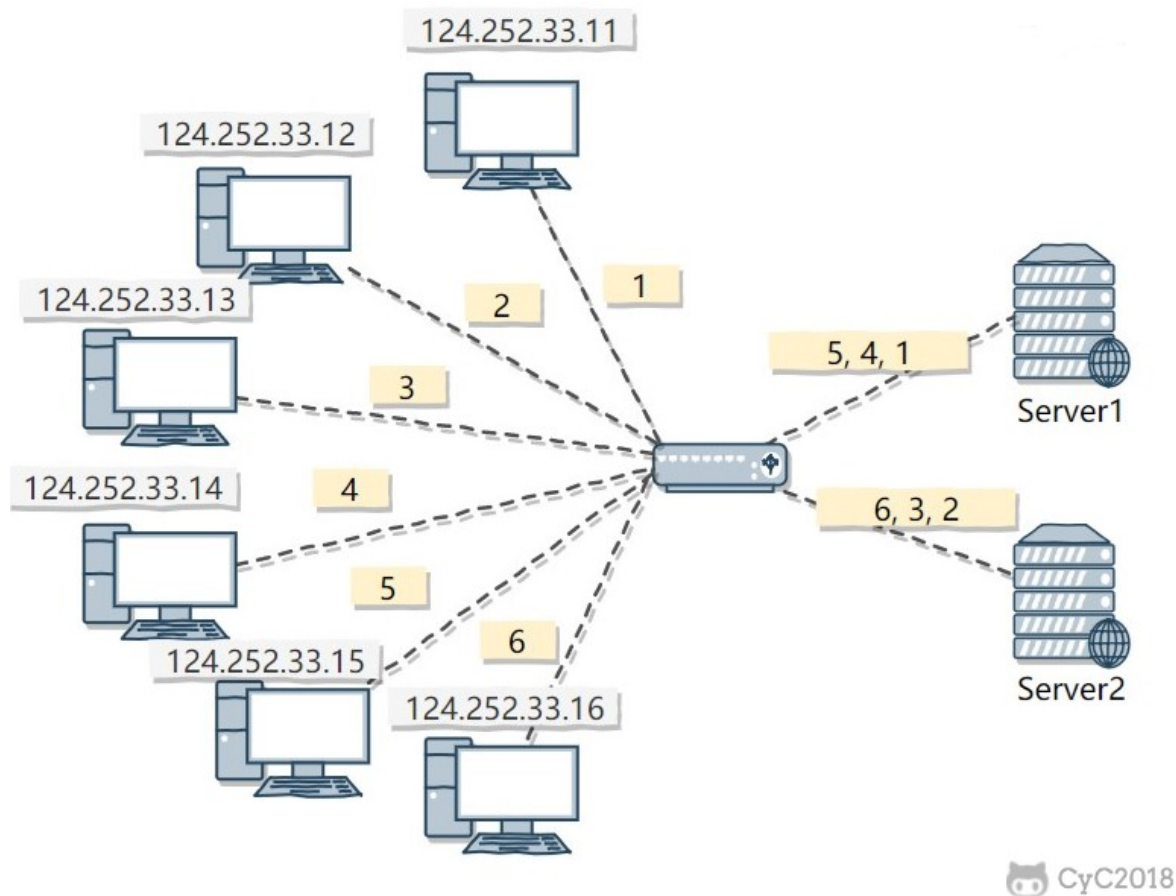


CyC2018

## 6. 源地址哈希法 (IP Hash)

源地址哈希通过对客户端 IP 计算哈希值之后，再对服务器数量取模得到目标服务器的序号。

可以保证同一 IP 的客户端的请求会转发到同一台服务器上，用来实现会话粘滞 (Sticky Session)



## 负载均衡基础技术

### 数据链路层负载均衡

通过在数据链路层修改mac地址进行负载均衡。

把所有机器的虚拟IP配置成跟负载均衡机器的IP一致，不需要进行地址转换就能直接返回响应数据，避免负载均衡服务器网卡成为瓶颈，常用于大型站点的外网入口，LVS。

### IP负载均衡

通过在在网络层修改目标地址进行负载均衡。

负载均衡服务器在内核进程获取网络数据包，修改目的IP，待真实响应服务器处理完成后，负载均衡服务器再将数据包源地址修改成自身IP地址。负载均衡服务器网卡容易成为瓶颈，并且没有反向代理灵活，不常用。

### 反向代理负载均衡

利用反向代理服务器进行负载均衡。

在应用层将请求转发到web服务器，待web服务器处理完成后，再将响应返回给用户。部署简单可灵活配置，常位于内网的web服务器前，nginx。

### HTTP重定向负载均衡

利用HTTP重定向协议实现负载均衡。

重定向服务器将真实服务器的地址写入重定向响应中，返回302进行负载均衡。需要两次请求，性能较差，不常用。

### DNS域名解析负载均衡

利用dns处理域名解析请求的同时进行负载均衡的一种方案。

在DNS服务器配置多个A记录，每次域名解析根据负载均衡算法返回不同的IP地址。常用于大型网站的第一级负载均衡。