负载均衡本质上是用于将用户流量进行均衡减压的

**负载均衡的方案：**

（1）基于DNS负载均衡

（2）基于硬件负载均衡

（3）基于软件负载均衡

（1）基于DNS负载均衡

解决方案：在DNS服务器上做一个简单的配置

原理：用户访问域名时，在DNS服务器上根据用户的IP将域名解析成不同的IP地址（北方用户解析成北方服务器的IP地址）

优点：配置简单

缺点：由于DNS有多级缓存，则修改DNS配置后，负载均衡效果不明显

基于地域或直接做IP轮询，没有更高级的路由策略

（2）基于硬件负载均衡

**F5 Network Big-IP**

是一个网络设备，类似于交换机，每秒处理请求达到百万级，性能好

优点：在负载均衡算法方面支持灵活的策略，具有防火墙安全功能

缺点：贵

（3）基于软件负载均衡

分为第七层协议和第四层协议

1）第4层协议：用七层网络协议中的第4层传输层来做流量分发

例如：LVS

性能较高：处理量达到几十万/秒

2）第7层协议：用七层网络协议中的第7层应用层来做流量分发 例如：Nginx

性能较低：处理量达到几万/秒

**常用的负载均衡算法**

**轮询策略 负载度策略 响应策略 哈希策略**

1.轮询策略

当用户请求来了之后，「负载均衡器」将请求轮流的转发到后端不同的业务服务器上

思考：轮询浪费不必要的时间，可以在负载均衡器上弄一个监听器，监听各个服务器的状态，达到有目的的访问服务器

轮询方式：按顺序轮询，随机轮询，按权重轮询（性能高的服务器分配的权重高一些）

2. 负载度策略

负载度策略是指当「负载均衡器」往后端转发流量的时候，会先去评估后端每台服务器的负载压力情况，对于压力比较大的后端服务器转发的请求就少一些，对于压力比较小的后端服务器可以多转发一些请求给它

优点：根据服务器状态动态分配流量比较好

缺点：负载均衡器处理转发请求外，还需要做其他工作（采集连接数，请求数，CPU负载指标，IO负载指标），增加负载均衡器的实现难度和维护成本。

3.响应策略

不停的统计每台服务器的响应时间（处理请求的速度），根据后台服务器的响应情况来分配

优点：用户体验好

缺点：不断的统计消耗成本，难以实现和维护

4.哈希策略

根据请求中的某个信息进行hash运算，值相同的被送到一个服务器中

例子：根据用户的IP或ID分配，这样同一用户就会访问相同的服务器，处理缓存和会话等功能十分好用。