



章节

问答

笔记

资料

4-15 附：LVS的负载均衡算法

LVS的负载均衡算法

静态算法

静态：根据LVS本身自由的固定的算法分发用户请求。

1. 轮询（Round Robin 简写'rr'）：轮询算法假设所有的服务器处理请求的能力都一样的，调度器会把所有的请求平均分配给每个真实服务器。（同Nginx的
2. 加权轮询（Weight Round Robin 简写'wrr'）：安装权重比例分配用户请求。权重越高，被分配到处理的请求越多。（同Nginx的权重）
3. 源地址散列（Source Hash 简写'sh'）：同一个用户ip的请求，会由同一个RS来处理。（同Nginx的ip_hash）
4. 目标地址散列（Destination Hash 简写'dh'）：根据url的不同，请求到不同的RS。（同Nginx的url_hash）

动态算法

动态：会根据流量的不同，或者服务器的压力不同来分配用户请求，这是动态计算的。

1. 最小连接数（Least Connections 简写'lc'）：把新的连接请求分配到当前连接数最小的服务器。
2. 加权最少连接数（Weight Least Connections 简写'wlc'）：服务器的处理性能用数值来代表，权重越大处理的请求越多。Real Server 有可能会存在性能上动态获取不同服务器的负载状况，把请求分发到性能好并且比较空闲的服务器。
3. 最短期望延迟（Shortest Expected Delay 简写'sed'）：特殊的wlc算法。举例阐述，假设有ABC三台服务器，权重分别为1、2、3。如果使用wlc算法的话求进来，它可能会分给ABC中的任意一个。使用sed算法后会进行如下运算：

- A： $(1+1)/1=2$
- B： $(1+2)/2=3/2$
- C： $(1+3)/3=4/3$

最终结果，会把这个请求交给得出运算结果最小的服务器。

4. 最少队列调度（Never Queue 简写'nq'）：永不使用队列。如果有Real Server的连接数等于0，则直接把这个请求分配过去，不需要在排队等待运算了（s

总结：

LVS在实际使用过程中，负载均衡算法用的较多的分别为wlc或wrr，简单易用。

附1：

参考文献：[官方文档](#)

附2：

作为架构师，对lvs集群的负载算法有一定的了解即可，因为你要和运维人员进行有效沟通；但是如果作为运维的话那么是要深入钻研LVS了，一个企业如果使用到LVS了，那么业务量是十分巨大的，并且也会有专门的运维团队来负责网络架构的。

检测到您还没有关注慕课网服务号，无法接收课程更新通知。请扫描二维码即可绑定



下一节

4-15 附：LVS的负载均衡算法

播放下一节

重新观看