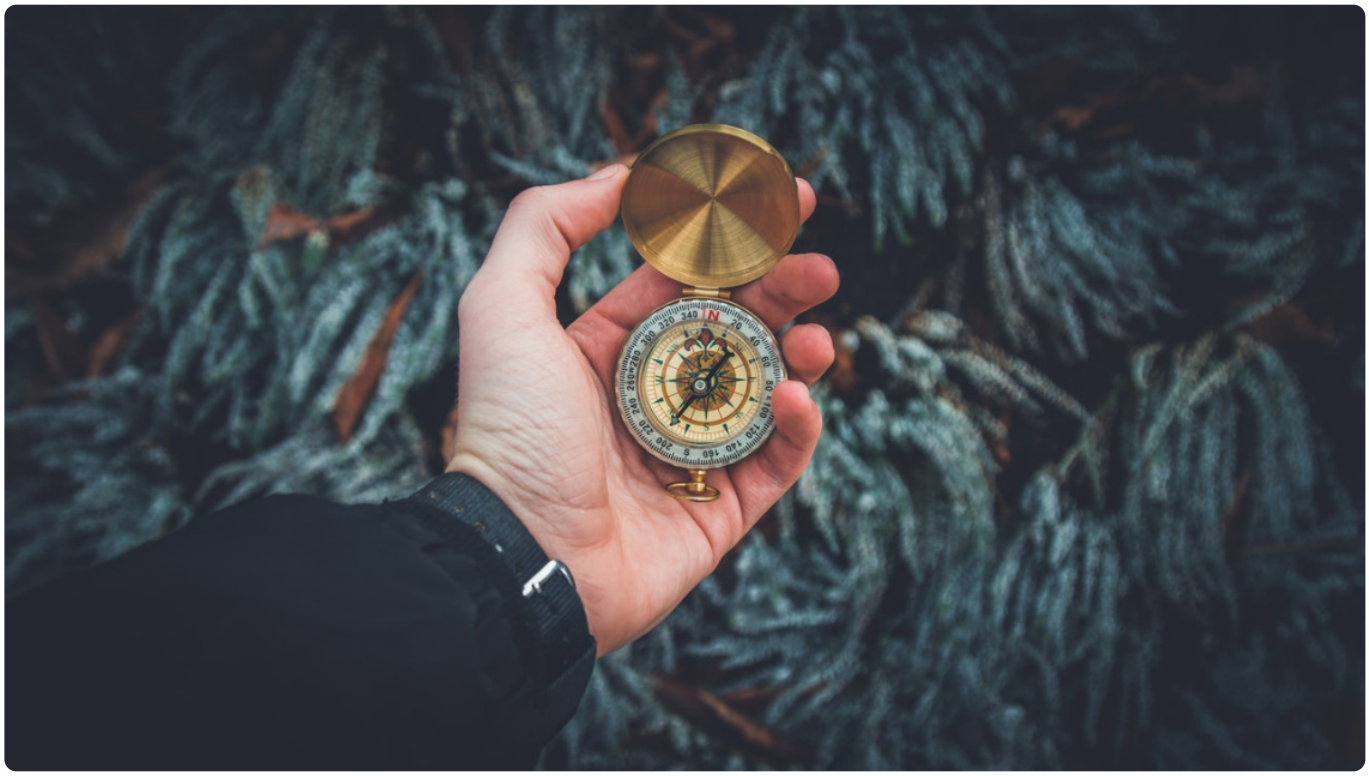
第35讲 | 如何用网关服务器进行负载均衡?

2018-08-14 蔡能

从0开始学游戏开发 进入课程 >



讲述：蔡能
时长 07:45 大小 3.56M

我们费劲心思做了一款游戏，那肯定希望有很多人来玩。一旦玩家数量开始多起来，服务器需要承担的压力就会变大，我们就需要做负载均衡了。

所谓的负载均衡，就是对外有一个公共地址，在请求过来的时候，通过轮询、随机分配等方式，定点到不同的服务器，以分摊服务器的压力。

两种常用的负载均衡技术

常用的负载均衡技术有**软件模式**和**硬件模式**。

其中，硬件模式用得比较多的是 F5。F5 是 F5 Networks 公司提供的一个负载均衡器专用设备， F5 BIG-IP LTM 的官方名称叫本地流量管理器，可以做 4～7 层负载均衡，具有负

载均衡、应用交换、会话交换、状态监控等等全备的强大网络功能。

F5 硬件负载均衡的优点是，负载均衡能直接通过智能交换机实现, 处理能力更强，与操作系统无关，负载性能强，适用于一大堆设备、大访问量，其缺点也是很明显的，那就是购买成本高，设备的配置冗余，有些用不上的都给默认配置了。另外，由于设备只有单个，所以单个负载均衡配置无法有效掌握所有服务器及应用状态。

硬件的负载均衡，是从网络层来判断负载规则，所以有时候软件的系统处理能力出现问题，网络硬件还可以作出负载的动作。

软件模式的话，比较常用的软件的有 LVS、Nginx、HAProxy。

LVS 是四层负载均衡，根据目标地址和端口选择内部服务器。Nginx 这种负载均衡工具是七层负载均衡。而 HAProxy 同时支持四层、七层负载均衡，还可以根据报文内容选择内部服务器。

因此，LVS 分发路径优于 Nginx 和 HAProxy，性能上也要高些。但 Nginx 和 HAProxy 则更具配置性，比如说可以用来做动静分离。所谓动静分离，就是根据请求协议的特征，去选择静态资源服务器还是应用服务器。

很久以前，游戏服务器只是简单的对应客户端的服务，就像使用编程语言写了一个多线程的程序，每个线程接收一个客户端，然后把该存储的数据放到数据库去保存。到了后期，大量的网游开始兴起，玩的人越来越多，所以一些老式游戏服务器框架已经无法满足更大更现代化的网络游戏的需求了。

到了 2005 年左右，这种情况愈演愈烈，不改变现状不行了。于是，程序员和游戏开发厂商设计出了新的一种服务器的框架模型。这种模型几乎是延用到今天，这种模型甚至延伸到各行各业的服务框架。

我们甚至可以说 Nginx 反向代理的想法也是类似这种模型的一种表现形式。尽管我们不能说 Nginx 学的就是这种模型，但是与这种反向代理的模型的做法实在太类似了。

这种服务器模型的最大改变，就是加了一个 gateway，可以称作网关。这当然不是传统意义上的网关路由器，只是在服务器的应用层面，做的事情类似网关路由器，所以我们仍然把它称为网关。

我们可以在 Web 端称它为**会话**（**Session**），也可以称它为**Link Server**，总之道理是一样的。

这个网关服务器所做的工作可以分为两种，对应网关服务器实现不同功能的服务。每一种功能不同，后台逻辑服务器的传输数据内容也会不同，不能相互混合使用。

网关服务器有哪些功能?

1. 中转功能

网关服务器作为一种代理，所有玩家从客户端传输到真正的游戏逻辑服务器的内容，都需要通过网关服务器，用该服务器作为中转。也就是说，假设有 A 客户端到 B 服务器，网关为 G 的话，就是 A 到 G 到 B，然后 B 服务器完成逻辑计算后，返回给 G 网关，网关再一次返回给 A、B 到 G 到 A。

这样做的好处是，网关可以随时询问它底下的真实逻辑服务器到底哪一台趋于饱和，可以将玩家移动到不饱和的游戏服务器，但是缺点也是很明显的，那就是玩家和服务器之间隔了一层网关，需要消耗更长的时间，传输速率相对低。

2. 负载均衡

网关服务器作为查询网关，也就是说，网关服务器会和底下所有服务器做一个长连接，或者随时询问的连接，这个连接所询问的内容，可以放到一个缓存里面，所查询的内容就是它底下所有服务器哪一台有空，在这种功能模式下，网关服务器只做了负载均衡的工作。

那么当客户端 A 要连接到游戏服务器的时候，需要先询问网关服务器 G，模型看起来会是这样：

A- 询问 G，G 通过查询缓存表，告知 A 客户端，C 服务器有空，于是通知 A，你去连 C 服务器，IP 地址和端口号是多少多少，于是 A 从网关 G 关闭连接，去连接 C 服务器。如果连接失败（因为是缓存查询，从逻辑上讲有可能滞后），那么再次询问网关，直到成功连接某一台服务器为止。

这个模型，网关服务器只做了负载均衡的动作，客户端和网关之间不会保持一个长连接，在这个基础上，一台网关服务器支撑同时七千人以上都不是什么太大的问题。但是它的缺点也很明显，那就是一台游戏逻辑服务器只能负责一个游戏世界，不能进行分块。如果要进行分块，则需要其他模型的服务器模块，这个我一会儿会说。

Nginx 的反向代理也是类似这种负载均衡的网关模型，这种模型大量运用在很多应用服务器、HTTP 连接的网络服务器上。但是，这项技术到了上升时期开始遇到了瓶颈，人们发现就算加上网关，也无法负担体量更大的游戏地图。于是，我们需要对这样的模型进行修改。

如何优化负载均衡的网关模型?

首先，需要将网关服务器增加为几个网关服务器。每个网关服务器都做相同的工作，也就是管理它所下属的所有逻辑服务器。客户端在启动的时候，随机抽取某一个网关服务器，进行询问，使用网关服务器做代理进行中转。

如果游戏地图特别大，这样的模型可以将游戏地图分割成几块，分割好的地图放到下属的各个逻辑服务器中，网关做中转服务，比如服务器 A 负责浙江省，服务器 B 负责安徽省等等。

客户端在连接到网关服务器后，随着游戏进度的走向，网关服务器可以选择连接负责哪一块地图的逻辑服务器，这样在玩家看来就像是连接了一台服务器，而客户端并不用考虑切换服务器的工作。

当然为了减轻服务器的压力，增加更多的人流量，后期这样的模型被逐步细分。比如可以将聊天服务放到一台独立的服务器进行存放，把用户数据独立到一台数据服务器存放，把商品交易放到另一个独立的服务器，或者把私信等等这些和主游戏逻辑无关的内容都放到一个独立的服务器上。

这样一来，主游戏逻辑的服务器的负载就会减轻，然而客户端就不得不多连接几台服务器，要不停获取用户数据或者聊天信息等等，某些负载就转嫁到客户端上了。

这样的游戏逻辑服务器的模型一直沿用到现在。某一些稍微轻量级的，只是使用网关当成负载均衡使用，有一些重量级的，加上地图分割，就会增加网关服务器，但是付出的代价就是，如果要加一台新的游戏逻辑服务器的话，势必会增加部署难度。

不仅网关服务器的配置文件要重新部署，每个游戏节点服务器和被分割的诸如聊天等服务都需要进行重新配置，这样付出的代价也是巨大的，当然很多游戏公司靠着这样的服务器框架使用了好多年，其思想也被延伸到各个行业领域的服务器架构中。

小结

这节内容差不多了，我来总结一下。

> 我首先讲了两种常用的负载均衡技术，软件模式和硬件模式。硬件模式用得比较多的是 F5。软件模式的话，比较常用的软件的有 LVS、Nginx、HAProxy。
>
> 网关服务器有中转功能和负载功能。Nginx 的反向代理用的是负载均衡的网关模型，但是这种模型无法负担更大体量的内容。为了减轻服务器的压力，也为了增加更多的人流量，可以通过增加网关，分割业务逻辑到独立的服务器，分摊服务器压力，这种经典类型的服务器模型被大量沿用并使用至今。

现在给你留一个小问题吧。

我们使用网关服务器这样的模型，如果网关服务器宕机了，或者网关服务器很久没有响应的情况下，有什么办法让客户端能顺利连上网关服务器之下的逻辑服务器呢？

欢迎留言说出你的看法。我在下一节的挑战中等你！

上一篇　第34讲 | 热点剖析（九）：谈谈独立开发者的未来发展

下一篇　第36讲 | 如何制作游戏内容保存和缓存处理？

下一篇 第36讲：如何制作游戏内容保存和缓存处理？

精选留言 (1)

写留言

以往

2018-09-10

2

每次将网关返回的结果在本地做缓存，如果连接网关超时，使用最近一次使用的逻辑服务器的IP和端口。

或者在前端存个网关列表，一台出问题在备选机器之间切换。

作者回复: 可以