

35 | 流量调度与负载均衡

2019-08-23 许式伟

许式伟的架构课

[进入课程 >](#)



讲述：姚迪迈

时长 10:30 大小 9.62M



你好，我是七牛云许式伟。

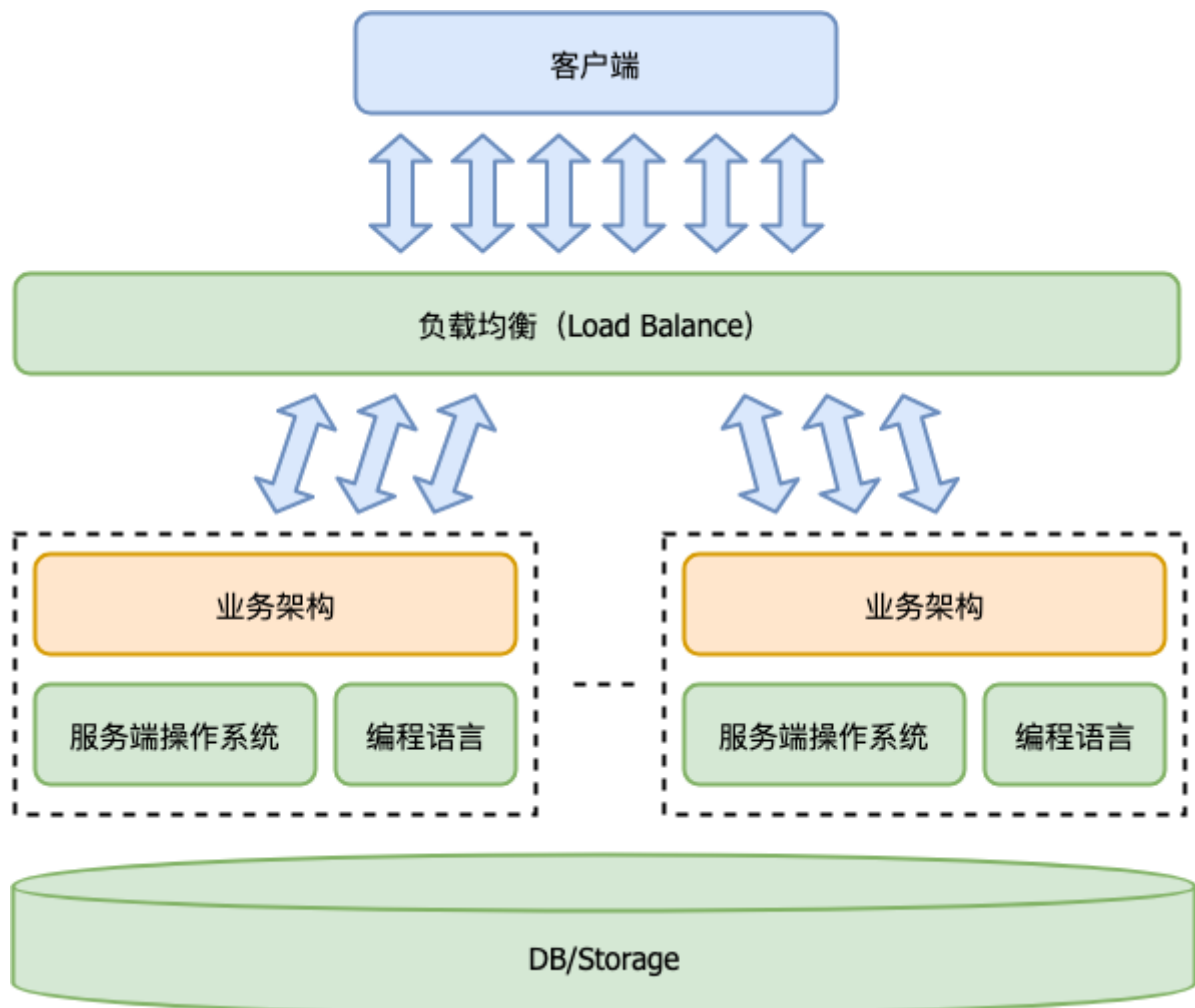
相比桌面程序而言，服务端程序依赖的基础软件不只是操作系统和编程语言，还多了两类：

负载均衡（Load Balance）；

数据库或其他形式的存储（DB/Storage）。

为什么会需要负载均衡（Load Balance）？今天我们就聊一下有关于流量调度与负载均衡的那些事情。

上一讲我们画了服务端程序的体系架构图，如下：



什么是“流量调度”？我们首先要了解这样几个常见的服务端程序运行实例（进程）相关的概念：

连接数；

IOPS；

流量，入向流量和出向流量。

我们知道，一个基本的服务端程序的服务请求，通常是由一个请求包（Request）和一个应答包（Response）构成。这样一问一答就是一次完整的服务。

连接数，有时候也会被称为并发数，指的是同时在服务中的请求数。也就是那些已经发送请求（Request），但是还没有收完应答（Response）的请求数量。

IOPS，指的是平均每秒完成的请求（一问一答）的数量。它可以用来判断服务端程序的做事效率。

流量分入向流量和出向流量。入向流量可以这么估算：

平均每秒收到的请求包（Request）数量 * 请求包平均大小。

同样的，出向流量可以这么估算：

平均每秒返回的应答包（Response）数量 * 应答包平均大小。

不考虑存在无效的请求包，也就是存在有问无答的情况（但实际生产环境下肯定是有的）的话，那么平均每秒收到的请求包（Request）数量、平均每秒返回的应答包（Response）数量就是 IOPS。故此：

入向流量 \approx IOPS * 请求包平均大小

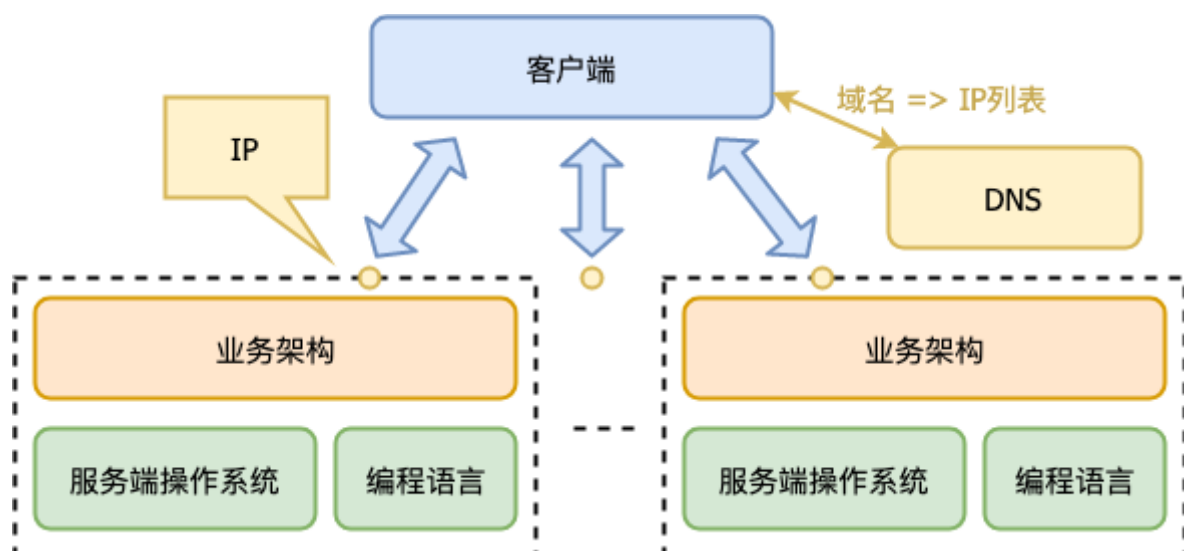
出向流量 \approx IOPS * 应答包平均大小

所谓流量调度，就是把海量客户并发的请求包按特定策略分派到不同的服务端程序实例的过程。

有很多手段可以做流量调度。

DNS 流量调度

最基础的方式，是通过 DNS，如下图所示。



一个域名通过 DNS 解析到多个 IP，每个 IP 对应不同的服务端程序实例。这样就完成了流量调度。这里我们没有用到常规意义的负载均衡（Load Balance）软件，但是我们的确完成了流量调度。

那么这种做法有什么不足？

第一个问题，是升级不便。

要想升级 IP1 对应的服务端程序实例，必须先把 IP1 从 DNS 解析中去除，等 IP1 这个实例没有流量了，然后我们升级该实例，最后把 IP1 加回 DNS 解析中。

看起来还好，但是我们不要忘记，DNS 解析是有层层缓冲的。我们把 IP1 从 DNS 解析中去除，就算我们写明 TTL 是 15 分钟，但是过了一天可能都还稀稀拉拉有一些用户请求被发送到 IP1 这个实例。

所以通过调整 DNS 解析来实现升级，有极大的不确定性，完成一个实例的升级周期特别长。

假如一个实例升级需要 1 天，我们总共有 10 个实例，那么就需要 10 天。这太夸张了。

第二个问题，是流量调度不均衡。

DNS 服务器是有能力做一定的流量均衡的。比如第一次域名解析返回 IP1 优先，第二次域名解析让 IP2 优先，以此类推，它可以根据域名解析来均衡地返回 IP 列表。

但是域名解析均衡，并不代表真正的流量均衡。

一方面，不是每次用户请求都会对应一次 DNS 解析，客户端自己有缓存。另一方面，DNS 解析本身也有层层缓存，到 DNS 服务器的比例已经很少了。

所以在这样情况下，按域名解析做流量调度均衡，是非常粗糙的，实际结果并不可控。

那么，怎么让流量调度能够做到真正均衡？

网络层负载均衡

第一种做法，是在网络层（IP 层）做负载均衡。

章文嵩博士发起的负载均衡软件 LVS（Linux Virtual Server）就工作在这一层。我们以 LVS 为代表介绍一下工作原理。

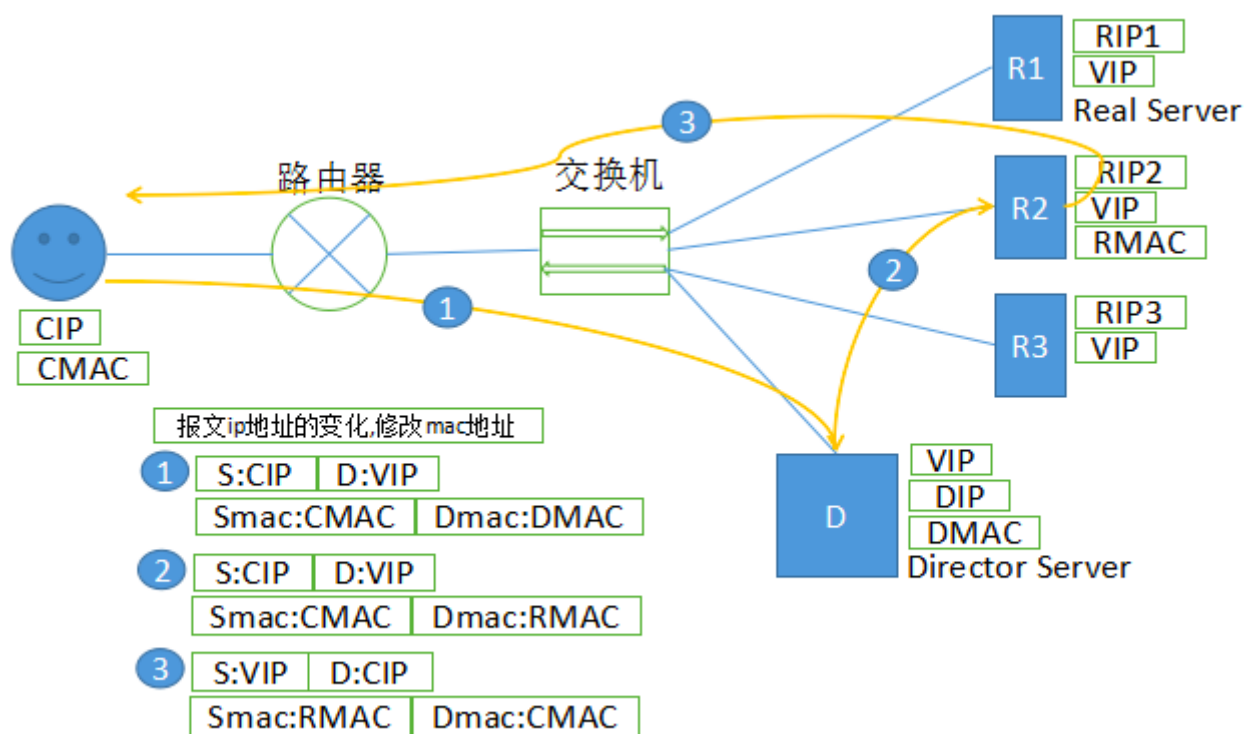
LVS 支持三种调度模式。

VS/NAT：通过网络地址转换（NAT）技术做调度。请求和响应都会经过调度器中转，性能最差。

VS/TUN：把请求报文通过 IP 隧道转发至真实服务器，而真实服务器将响应直接返回给客户，所以调度器只处理请求报文。这种做法性能比 VS/NAT 好很多。

VS/DR：通过改写请求报文的 MAC 地址，将请求发送到真实服务器，真实服务器将响应直接返回给客户。这种做法相比 VS/TUN 少了 IP 隧道的开销，性能最好。

我们重点介绍下 VS/DR 技术。



如上图所示。设客户端的 IP 和 MAC 为 CIP、CMAC。

第 1 步，客户端发起请求，其 IP 报文中，源 IP 为用户的 CIP，目标 IP 是 VIP；源 MAC 地址为 CMAC，目标 MAC 地址为 DMAC。

第 2 步，请求包到达 LVS 调度器（Director Server）。我们保持源 IP 和目标 IP 不变，仅仅修改目标 MAC 地址为 RMAC，将请求转发到真实的业务服务器实例 RS（Real Server）。

第 3 步，RS 收到数据包并经过处理，直接响应发送给客户端。

这里面的关键技巧，是 VIP 绑定在多台机器上，所以我们把它叫做虚拟 IP（Virtual IP）。它既绑定在 LVS 调度器（Director Server）上，也绑定在所有的业务服务器实例 RS（Real Server）上。

当然这里有一个很重要的细节是，ARP 广播查询 VIP 对应的 MAC 地址得到什么？答案当然是 LVS 调度器（Director Server）。在真实的业务服务器实例 RS（Real Server）上，我们把 VIP 绑定在 lo 接口上，并对 ARP 请求作了抑制，这样就避免了 IP 冲突。

LVS 这种在网络层底层来做负载均衡，相比其他负载均衡技术来说，其特点是通用性强、性能优势高。

但它也有一些缺点。假如某个业务服务器实例 RS 挂掉，但 LVS 调度器（Director Server）还没有感知到，在这个短周期内转发到该实例的请求都会失败。这样的失败只能依赖客户端重试来解决。

应用层负载均衡

有办法避免出现这种请求失败的情况吗？

可以。答案是：服务端重试。

怎么做服务端重试？应用层负载均衡。有时候我们也把它叫做应用网关。

HTTP 协议是应用最为广泛的应用层协议。当前应用网关，绝大多数都是 HTTP 应用网关。

Nginx 和 Apache 都是大家最为耳熟能详的 HTTP 应用网关。因为知道应用层协议的细节，所以 HTTP 应用网关的能力通常非常强大。这一点我们后面还会进一步进行探讨，今

天我们先聊负载均衡（Load Balance）相关的内容。

HTTP 网关收到一个 HTTP 请求（Request）后，根据一定调度算法把请求转发给后端真实的业务服务器实例 RS（Real Server），收到 RS 的应答（Response）后，再把它转发给客户端。

整个过程的逻辑非常简单，而且重试也非常好做。

在发现某个 RS 实例挂了后，HTTP 网关可以将同一个 HTTP 请求（Request）重新发给其他 RS 实例。

当然一个重要的细节是为了能够支持重试，HTTP 请求（Request）需要被保存起来。不保存 HTTP 请求做重试是有可能的，但是只能支持业务实例完全挂掉 HTTP 请求一个字节都没发过去的场景。但在断电或异常崩溃等情况，显然会有很多进行中的请求是不符合这个前提的，它们就没法做重试。

大部分 HTTP 请求不大，直接在内存中存储即可，保存代价不高。但是文件上传型的请求，由于请求包中包含文件内容，可能就需要依赖临时文件或其他手段来保存 HTTP 请求。

优雅升级

有了负载均衡，不只是可以实现了流量的均衡调度，连带业务服务器的升级也会方便多了。

对于前端是 LVS 这种网络层负载均衡的场景，升级的核心步骤为：

升级系统通知 LVS 调度器（Director Server）下线要升级的业务服务器（Real Server）实例。

LVS 调度器（Director Server）将该实例从 RS 集合中去除，这样就不再调度新流量到它。

升级系统通知要升级的 RS 实例退出。

要升级的 RS 实例处理完所有处理中的请求，然后主动退出。

升级系统更新 RS 实例到新版本，并重启。

升级系统将 RS 实例重新加回 RS 集合参与调度。

对于前端是 HTTP 应用网关这种负载均衡的场景，升级的过程可以更加简单：

升级系统通知升级的业务服务器（Real Server）实例退出。

要升级的 RS 实例进入退出状态，这时新请求进来直接拒绝（返回一个特殊的 Status Code）；处理完所有处理中的请求后，RS 实例主动退出。

升级系统更新 RS 实例到新版本，并重启。

可以看出，因 HTTP 应用网关支持重试，业务服务器的升级过程就变得简单很多。

结语

今天我们从流量调度谈起，聊了几种典型的调度手段和负载均衡的方式。

从流量调度角度来说，负载均衡的最大价值是让多个业务服务器的压力均衡。这里面隐含的一个前提是负载均衡软件的抗压能力往往比业务服务器强很多（为什么？欢迎留言讨论）。

这表现在：其一，负载均衡的实例数 / 业务服务器的实例数往往大大小于 1；其二，DNS 的调度不均衡，所以负载均衡的不同实例的压力不均衡，有的实例可能压力很大。

当然，负载均衡的价值并不只是做流量的均衡调度，它也让我们的业务服务器优雅升级成为可能。

如果你对今天的内容有什么思考与解读，欢迎给我留言，我们一起讨论。下一讲我们将聊聊存储中间件。


如果你觉得有所收获，也欢迎把文章分享给你的朋友。感谢你的收听，我们下期再见。

许式伟的架构课

从源头出发, 带你重新理解架构设计

许式伟
七牛云 CEO



新版升级: 点击「 请朋友读」, 20位好友免费读, 邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有, 未经许可不得传播售卖。页面已增加防盗追踪, 如有侵权极客邦将依法追究其法律责任。

上一篇 34 | 服务端开发的宏观视角

精选留言 (14)

写留言



leslie

2019-08-23

老师今天说的都是前端的: 可是好像负载均衡不止这些吧, 软件一旦并发高了不是从整体的去做均衡么, 不仅仅是这些吧? 就像数据库方面我经常会去做一主多从、读写分离, 甚至说软硬件很多时候都会做相应的事情, 可是总觉得这个如何去整体的把握这种均衡确实觉得不容易把握。

老师的课程一路断断续续努力学到现在整体的收获还是让我感觉不一样: 如果可以...
展开

作者回复: 存储的扩展不是基于负载均衡, 这个下一讲就会有所涉及。



2

3



Geek_4b2920

2019-08-24

讲到lvs时说到"有办法避免出现这种请求失败的情况吗？"，接着就说nginx是怎么去做的，感觉这里不太衔接呢，lvs不能做服务端重试？还是什么原因？没太明白

展开 ∨

作者回复: lvs 不太好做，在 VS/DR 模式下应答包（response）根本就不经过它，所以它连请求包（request）是否已经应答都不知道，就别提失败后重试了。如果在 VS/NAT 模式下，它也需要理解应用层的协议后才能重试，那它就不是网络层负载均衡，而是应用层负载均衡了。



2



Void_seT

2019-08-23

- 1、首先，因为绝大多数情况下负载均衡服务器的简单转发消耗的系统资源更少，而业务逻辑的处理往往需要更多的系统资源，那么，在服务器配置相当的情况下，负载均衡服务器就比业务处理服务器能处理更多的请求；
- 2、如果，负载均衡服务器的处理能力与业务处理服务器的处理能力相当，那这种依靠负载均衡服务器来做负载均衡的方式效率就极低（约为50%），资源使用率也很低（约为5...

展开 ∨



1



黄伟洪

2019-08-23

Docker是基于应用层的负载均衡？

展开 ∨

作者回复: 我猜想你说的docker应该是指k8s。k8s应该是四层（传输层）和七层（应用层）。我们这里谈的是三层（网络层）和七层。



Aaron Cheung

2019-08-23

学习了 业务流量不大.....肿么办

展开 ∨



许童童

2019-08-23

老师这一节讲得很好，服务优雅升级配合负载均衡确实是很不错的解决方案。



觉

2019-08-23

一门深入 长时薰修

展开 ▾



Jxin

2019-08-23

1.课后题：假如网关层负载率小于应用层，同时本次请求是需要rsp的。那么网关层该干嘛还是能干嘛，问题是整个应用集群的负载量将受到网关层的约束，也就是说水平扩容无状态应用服务并无法增长负载量。（单机瓶颈依然存在，java现流行的可编程网关负载感觉就会存在这方面问题）

2.存储请求和计算请求是两码事。存储请求的请求分发往往意味着业务数据模型的拆分...

展开 ▾



Charles

2019-08-23

从文章的描述，负载均衡软件之所以性能高，是因为相对业务服务器，少了高级语言层面的编译解析执行、复杂业务逻辑处理、IO读写存储、以及响应过程的处理，而是直接通过网络层以及CPU和内存解决了一切需求



CoderLim

2019-08-23

负载均衡软件的抗压能力往往比业务服务器强很多，为什么？

负载均衡的功能只是转发，相对简单，没有耗时操作，主要的瓶颈应该是最大连接数和内存



歌在云端

2019-08-23

请问一下多机房的是怎么处理的，比方说在深圳，上海，北京各部署对应的服务，怎么保证广东那边的请求优先进入到深圳的机房里面去啊？是通过DNS吗。还有假如客户端增加，是不是可以三种均衡一起弄

作者回复: 1、是通过dns；2、三个均衡方式是可以一起的





借我一生

2019-08-23

1, 在硬件层做负载均衡, 比如F5 也是一种常见且高性能的做法。
2, 要避免最外接入层单点, 且需要解决超高并发下单台负载均衡服务器性能瓶颈, 只能上DNS轮询技术



Linuxer

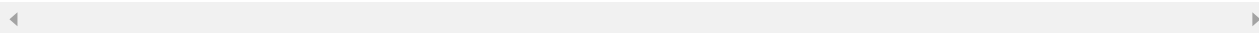
2019-08-23

这里面隐含的一个前提是负载均衡软件的抗压能力往往比业务服务器强很多

负载均衡软件按照文章所说应该有更高的iops, 处理时间短, 逻辑相对简单。
有一个问题请教, 负载均衡考虑流量, 是不是还需要保存到每个RS目前流量的统计信息呢?

展开 ▾

作者回复: 严谨考虑的话比较复杂, 负载均衡也是一个集群, 相互之间是否需要协同。目前负载均衡的调度策略大都比较简单, 详细可以参考lvs、nginx等软件的文档。



业余爱好者

2019-08-23

负载均衡软件就是为了流量调度而生的, 它主要是将请求路由到应用服务器, 相比而言, 应用服务器多了负载的业务处理这一步, 所以抗压能力不如负载均衡软件。

展开 ▾

