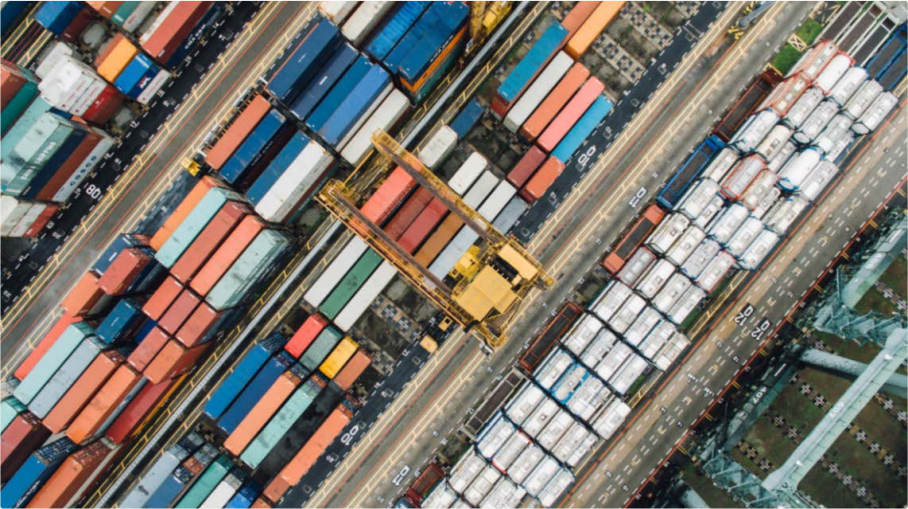


20 | 高性能负载均衡：分类及架构
2018-06-12 李运华

更多一手资源请添加QQ/微信1182316662





20 | 高性能负载均衡：分类及架构

李运华

- 00:00 / 09:14

单服务器无论如何优化，无论采用多好的硬件，总会有一个性能天花板，当单服务器的性能无法满足业务需求时，就需要设计高性能集群来提升系统整体的处理性能。

高性能集群的本质很简单，通过增加更多的服务器来提升系统整体的计算能力。由于计算本身存在一个特点：同样的输入数据和逻辑，无论在哪个服务器上执行，都应该得到相同的输出。因此高性能集群设计的复杂度主要体现在任务分配这部分，需要设计合理的任务分配策略，将计算任务分配到多台服务器上执行。

高性能集群的复杂性主要体现在需要增加一个任务分配器，以及为任务选择一个合适的任务分配算法。对于任务分配器，现在更流行的通用叫法是“负载均衡器”，但这个名称有一定的误导性，会让人潜意识里认为任务分配的目的是要保持各个计算单元的负载达到均衡状态。而实际上任务分配并不只是考虑计算单元的负载均衡，不同的任务分配算法目标是不一样的，有的基于负载考虑，有的基于性能（吞吐量、响应时间）考虑，有的基于业务考虑。考虑到“负载均衡”已经成为了事实上的标准术语，这里我也用“负载均衡”来代替“任务分配”，但请你时刻记住，负载均衡不只是为了计算单元的负载达到均衡状态。

今天我先来讲讲[负载均衡的分类及架构](#)，下一期会讲负载均衡的算法。

负载均衡分类

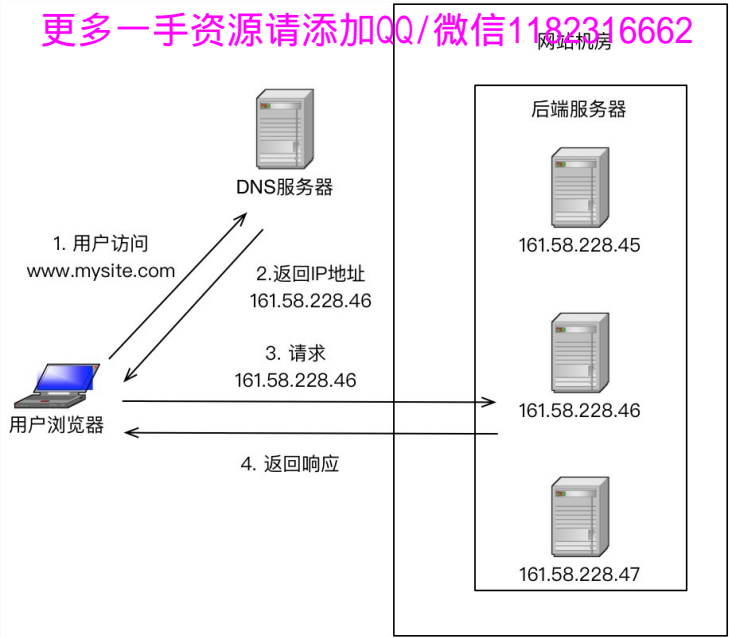
常见的负载均衡系统包括3种：DNS负载均衡、硬件负载均衡和软件负载均衡。

DNS负载均衡

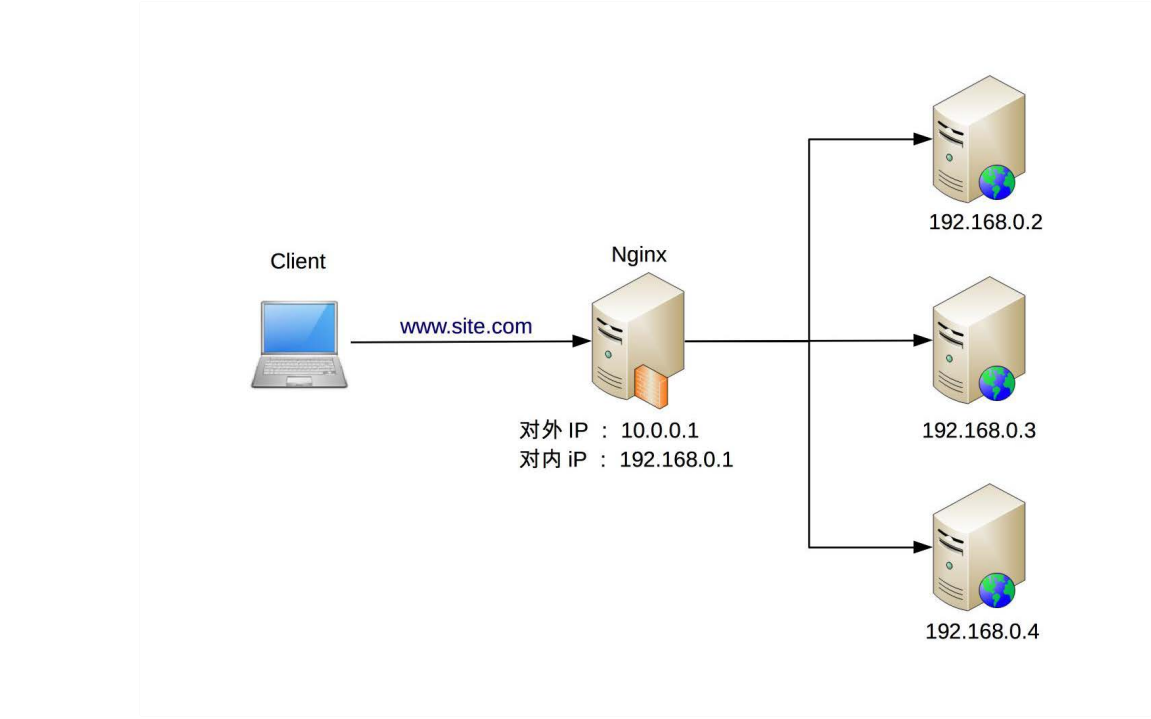
DNS是最简单也是最常见的负载均衡方式，一般用来实现地理级别的均衡。例如，北方的用户访问北京的机房，南方的用户访问深圳的机房。DNS负载均衡的本质是DNS解析同一个域名可以返回不同的IP地址。例如，同样是www.baidu.com，北方用户解析后获取的地址是61.135.165.224（这是北京机房的IP），南方用户解析后获取的地址是14.215.177.38（这是深圳机房的IP）。

下面是DNS负载均衡的简单示意图：

更多一手资源请添加QQ/微信1182316662



更多一手资源请添加QQ/微信1182316662



软件负载均衡的优点:

- 简单: 无论是部署还是维护都比较简单。
- 便宜: 只要买个Linux服务器, 装上软件即可。
- 灵活: 4层和7层负载均衡可以根据业务进行选择; 也可以根据业务进行比较方便的扩展, 例如, 可以通过Nginx的插件来实现业务的定制化功能。

其实下面的缺点都是和硬件负载均衡相比的, 并不是说软件负载均衡没用了。

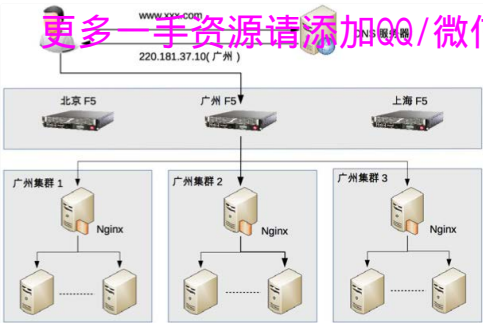
- 性能一般: 一个Nginx大约能支撑5万并发。
- 功能没有硬件负载均衡那么强大。
- 一般不具备防火墙和防DDoS攻击等安全功能。

负载均衡典型架构

前面我们介绍了3种常见的负载均衡机制: DNS负载均衡、硬件负载均衡、软件负载均衡, 每种方式都有一些优缺点, 但并不意味着在实际应用中只能基于它们的优缺点进行非此即彼的选择, 反而是基于它们的优缺点进行组合使用。具体来说, 组合的基本原则为: DNS负载均衡用于实现地理级别的负载均衡; 硬件负载均衡用于实现集群级别的负载均衡; 软件负载均衡用于实现机器级别的负载均衡。

我以一个假想的实例来说明一下这种组合方式, 如下图所示。

更多一手资源请添加QQ/微信1182316662



整个系统的负载均衡分为三层。

- 地理级别负载均衡：www.xxx.com部署在北京、广州、上海三个机房，当用户访问时，DNS会根据用户的地理位置来决定返回哪个机房的IP，图中返回了广州机房的IP地址，这样用户就访问到广州机房了。
- 集群级别负载均衡：广州机房的负载均衡用的是F5设备，F5收到用户请求后，进行集群级别的负载均衡，将用户请求发给3个本地集群中的一个，我们假设F5将用户请求发给了“广州集群2”。
- 机器级别的负载均衡：广州集群2的负载均衡用的是Nginx，Nginx收到用户请求后，将用户请求发送给集群里面的某台服务器，服务器处理用户的业务请求并返回业务响应。

需要注意的是，上图只是一个示例，一般在大型业务场景下才会这样用，如果业务量没这么大，则没有必要严格照搬这套架构。例如，一个大学的论坛，完全可以不需要DNS负载均衡，也不需要F5设备，只需要用Nginx作为一个简单的负载均衡就足够了。

小结

今天我为你讲了负载均衡的常见分类以及典型架构，希望你有所帮助。

这就是今天的全部内容，留一道思考题给你吧，假设你来设计一个日活跃用户1000万的论坛的负载均衡集群，你的方案是什么？设计理由是什么？

欢迎你把答案写到留言区，和我一起讨论。相信经过深度思考的回答，也会让你对知识的理解更加深刻。（编辑乱入：精彩的留言有机会获得丰厚福利哦！）



鲍米豆发

日活千万的论坛，这个流量不低了。
1、首先，流量评估。
1000万DAU，换算成秒级，平均约等于116。
考虑每个用户操作次数，假定10，换算成平均QPS=1160。
考虑峰值是均值倍数，假定10，换算成峰值QPS=11600。
考虑静态资源、图片资源、服务拆分等，流量放大效应，假定10，QPS*10=116000。
2、其次，容量规划。
考虑高可用、异地多活，QPS*2=232000。
考虑未来半年增长，QPS*1.5=348000。
3、最后，方案设计。
三级导流。
第一级，DNS，确定机房，以目前量级，可以不考虑。
第二级，确定集群，扩展优先，则选Haproxy/LVS，稳定优先则选F5。
第三级，Nginx+KeepAlived，确定实例。

作者回复

思路不错◆◆◆◆

食指可爰多

请问老师后面会有容量规划方面文章吗？日活用户1000w转换成日请求量（这一步我没啥经验），再计算平均qps，考虑请求放大倍数，按峰值qps的5倍。1000x10000x10*24*60*60x5=5700得到qps峰值5700。不考虑后端应用层和更下层数据库缓存这些，接入层Nginx就可以搞定？

作者回复	2018-06-13
同样1000万日活用户，不同业务特点的QPS差异很大，例如抖音的访问量会明显高于支付业务，论坛业务明显高于工具类业务	
公号-Java大后端	2018-06-12
通过容量规划，并考虑到高性能、高可用的要求，Web前端可采用HAProxy+Keepalived双机(实现故障转移，保证高可用)作为负载均衡器；后端的数据库架构采用MySQL一主多从，读写分离的方式，可采用LVS+Keepalived的方式实现读数据库的负载均衡与故障转移。	
piflying	2018-06-12
1、首先分析论坛系统的需求：高可用、扩展性、常规安全性、高性能。以上需求优先级依次降低。 2、开发计算： 1)、首先计算每秒并发量：1000万/(10*60*60)=278qps。（此处每天按照10个小时计算） 2)、计算每秒最大并发量：278*5=1390。（此处的5为经验值，按照论坛的用户使用特点多集中晚上小部分时段，该值已尽量取大。同时网上也有按照时间和并发数的二/八原则计算，本人按照第一种计算） 3、容量规划： 1、前端2台nginx负载均衡，利用keepalive保证高可用，同时可用做静态资源缓存服务器。 2、后端tomcat部署应用，按照单台tomcat支撑1200并发，需要2台。考虑冗余，此处配置3台。 3、考虑高性能应用可以集成缓存，也可以使用统一缓存。 4、数据库采用mysql，使用2台进行主从复制和读写分离。一方面满足读多写少的应用场景，另一方面在1台出现故障时，能保证高可用。 以上内容请老师指正！	
作者回复	2018-06-13
1000万是用户数量，不是访问次数，访问次数会多很多，其它分析都可以	
ant	2018-06-12
日活跃用户1000万应该就是国家级应用了，面向全国活跃或者全球用户，比如最大的Xxx网站github，这个时候钱应该都不是问题了。我觉得可以考虑异地多机房部署。这样导流之后每个机房的日活就少很多。其实我在想如果在每个机房不加入负载均衡用多个nginx集群来实现，每个nginx上会有我们自定义的路由算法。nginx也架设多层，逐层导流，比如我们一个机房设计承受200万，那么我们可以架设3层nginx，第一层基于自己的路由算法导流到第2层nginx。第2层又导流到第3层。为了避免nginx单点，每一层nginx部署多。这样导流下流每台服务器所承认的访问不会很多。不知道这样的设计能不能达到要求，老师点评下	
作者回复	2018-06-13
可以达到，但有点复杂，nginx做级联不太合适，因为顶层的nginx性能是瓶颈，多级导流一般用在处理能力有差异的系统上，例如一级用F5，二级用LVS，三级用nginx	
黄金的太阳	2018-06-12
假设论坛的用户平均分布在全国各地(东、西、南、北四个区域)，1000万的日活跃用户平均分散到每个区域后可近似估计并发量在2.5万~5万用户，可以采用两级嵌套的负载均衡架构 1.利用DNS达到四个地域的负载均衡 2.利用Nginx的方式达到本区域内的负载均衡 此方案未考虑西部地区用户少，东部地区用户多的情况，在并发量尚可接受的范围内，可以考虑将单台Nginx集群化以增强并发负载支持能力	
不知道理解的对不对	
作者回复	2018-06-13
基本正确，中国一般分南北北区域接入，西部用户确实少很多	
何国平	2018-06-14
nginx也支持4层反向代理了	
作者回复	2018-06-14
我宁愿用LVS，久经考验，性能强大◆◆	
felfei	2018-06-13
日活跃用户千万，按14小时折算，每秒并发198，但这是平均，还有高峰时段，并发按平均并发5倍来估算，即每秒1千，然后来对比方案：	
Dns负载，目前单机房能够满足，没跨机房的必要，dns目前不适用。 硬件负载，每秒几百万级的并非，很显然系统没有这么高的并发，硬件负载不适合。 软件负载，nginx单台能支持到5万的并发，当前系统.折算最高的并发也不过千级别。	
经过方案的对比，软件负载使用nginx可以完全满足业务的要求，所以使用软件负载即可	
作者回复	2018-06-14
日活用户数！= 用户访问数，论坛类业务，一个用户一天可能访问几十个页面	
汇通科技	2018-06-13
并发测试如何做，怎么知道自己设计的数据库，或者架构能支撑多少并发	
作者回复	2018-06-13
基于业务场景进行性能压测，了解大概量级即可，不需要很精确	
张伟(大圣)	2018-06-12
看了大家的各种计算，估容量，估机器，	
想说的是：根据之前专栏讲到的单台高性能模式等知识，先把单台机器做到最优化，同时做好负载均衡层，然后进行压测，一步一步添加机器，均衡层 Nginx 够了，另外，要考虑成本啊，F5尽量不用，稳定性用双主克服下	
作者回复	2018-06-13
最好算一下，当然有经验的架构师确实能够凭感觉预估	
肖一林	2018-06-12
峰值大概就是5000/s ~ 20000/s，要看论坛活跃度。所以一个ng就够了。dns负载均衡也不一定就要支持异地多活吧，同一个机房多台主机也是可以的，所以最多dns+ng就可以很完美。需要异地多活的项目应该非常少。	
作者回复	2018-06-13
这种方式也可以，dns做同机房多入口负载均衡	

更多一手资源请添加QQ/ 微信1182316662

三水	2018-06-12
老师，流云说SLB还有HAProxy，我们用LVS做DNS的LB，Nginx在HAProxy做HTTP的LB。 作者回复	2018-06-13
HAProxy也很成熟，可以用	2018-06-13
星火燎原	2018-06-12
不差钱的话可以考虑文中DNS + F5+ nginx ，一般这种日活还是考虑DNS + LVS + Nginx 作者回复	2018-06-13
论坛不怎么赚钱啊💎💎	2018-06-12
姜洋昌	2018-06-12
首先可以增加DNS负载均衡，如果部分地区并发访问量仍然大大。或无法使用DNS负载均衡，则可以搭建一个F5的集群，然后再通过另外一台F5对这个F5集群进行任务分配，成本虽昂贵，但相信对于这样有海量用户的系统也不会有太大的成本压力。另外老师啥时候介绍一下负载均衡的算法？ 作者回复	2018-06-13
下一章就介绍算法	2018-06-13
老北	2018-07-08
千万日活，论坛的时间相对比较集中，同时在线预计会达到一百万。 这时候会有一半的人在操作(查看帖子之类)，每个用户操作可能会调用2-3个接口。那并发数大约就是50w*2.5=125w？ 这时候nginx的5w并发就不行了。需要多个dns到不同主机，再进行nginx.lvs之类转发。 另外像tomcat一般支持2000左右连接数。这样就需要600多台tomcat？ 总感觉哪里算的不对💎💎 作者回复	2018-07-09
确实有点吓人，千万日活转换为百万同时在线这里有问题，一般把日活转换为pv，例如平均每个用户会访问100个页面，日访问量就是10亿，每秒就是大约1.2万的并发访问量，再按照峰值等于均值3倍来算，也就3.6万，远远没有125万那么夸张	2018-07-03
啮yi啃泥	2018-07-04
我也希望老师能对并发数，qps，日活这些能够举个经典例子计算下，网上虽然有，可是总觉得讲解地不是很清楚，对这块一直都有点迷糊。 作者回复	2018-07-04
并发数：同一时刻的接收的访问数量，时间单位一般用秒，也可以用分钟和小时，常见有并发请求数，并发连接数 QPS：query per second，指查询类请求 日活：每日活跃用户数，指当天来访问过系统的用户，同一用户，无论用户访问多少功能和页面都只算一个用户	2018-06-29
低调的大老刘	2018-06-29
华哥，看到很多DNS+Nginx做负载，但是这种方式没办法预防DDOS攻击，你的Ip都暴露了 作者回复	2018-06-29
谁都没法防DDOS攻击呀，不暴露ip，正常用户也访问不了啊💎💎	2018-06-27
楼剑波	2018-06-28
老师，能回答下，政府部门的域名也是通过域名服务商那里买的吗？比如说万网，他们也需要dns，把域名解析到IP吧？感觉dns是所有架构都要采用的，要么你的网站就只要IP访问。 作者回复	2018-06-28
买不买不一定，用是肯定要用的💎💎	2018-06-25
孙晓明	2018-06-25
老师能讲一下设计思路吗？以前听说是计算并发的话一般是用户数的1%，但是看各位同学的分析都很复杂，而且感觉除以时间的计算方式也不合理，就像您说的，不同系统的访问请求量也是不同的，不能一刀切的除以时间来计算并发。求答案。	2018-06-25
李牧之	2018-06-18
ld hash的方法只能用着7层负载均衡上吧？另外，请问一下数据库层的负载均衡一般使用什么算法	2018-06-18
交叉路口	2018-06-15
论坛这种业务的接口响应应该比较短，预计平均100ms，超时限制500ms。日活千万，预计峰值QPS 4000/s，按照超时500ms，并发估算2000。采取dns+nginx 足够，具体实例根据staging 压测来评估。dns 一是为了地理位置负载均衡，还有为了扩展性（客户端通过域名访问，后端需要拓展机器对客户端透明）。Nginx：应用负载均衡，打到某个服务实例，利用其故障转移（可检测实例状态进行剔除）、并发限制等特性来保证服务稳定性，同时还可以利用它做其他事情（access log 做日志行为分析）。希望华哥指出不足💎💎 作者回复	2018-06-15
基本OK	2018-06-15
迟博💎💎	2018-06-15
我看到好多评论说多机房，请问一下多机房的数据一致一般怎么解决的？数据库通过专线组成跨dc的多主集群？这个后面会讲到吗？ 作者回复	2018-06-15
会有异地多活章节讲	2018-06-14
Mask	2018-06-14
老师，这个负载均衡会涉及到数据同步吗？还是说我没搞懂💎💎，多台服务器其实访问的是同一个数据库服务器，不存在数据实时同步的问题。 作者回复	2018-06-14
负载均衡不涉及数据同步，是指将请求分发到不同服务器	2018-06-14

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

吴天	更多一手资源请添加QQ/微信1182316662	2018-06-14
可以选用DNS做区域划分 在使用Nginx做负载均衡。 日活1000万，每秒开发115，峰值按照10~100 也只在 1150~11500，Nginx 和DNS成本较低 也足够使用		
作者回复		
日活1000万不是日pv 1000万，整体计算逻辑是OK的		2018-06-19
learn more		
活跃用户数，并发数，每秒开发量，qps，峰值估算，最大连接数等等，这些名词充满了整篇文章和评论，可是计算结果千差万别，恳请作者帮帮新手如何计算这几个值，以及确定概念		2018-06-13
作者回复		
名词可以上网查查，计算方法参考前面高性能复杂度的内容		2018-06-13
zhou		
你好，后续可介绍下有些偏业务的规则引擎相关的知识吗？		2018-06-13
作者回复		
规则引擎和设计模式类似，都是单机具体实现的一种方式，和架构关系不大，所以没有进行阐述		2018-06-14
sunlight001		
感觉这种设计初期，考虑用户体验，可以用dns+ha+nginx来做负载均衡，一般的可能不用dns也差不多，用了dns肯定是要快些的		2018-06-13
作者回复		
论坛业务其实可以不用dns做地理位置负载均衡，用CDN效果更好		2018-06-13
一位不愿透露姓名的王先生		
多台LVS+ECMP 做前端负载均衡，Nginx做Web服务器，其实主要压力还是在应用逻辑和数据库上，中间再加上一层缓存		2018-06-13
大P		
日活1000万。平均每秒活跃就是1000万/86400≈115，高峰放大10倍也就是1150，这是每秒活跃用户，不知能否当作QPS？ 个人觉得论坛读多写少类的，加上老师之前说的mysql读写分离或分库分表(需要结合实际情况)+高缓存结合，前面Nginx做负载均衡（Nginx理论上每秒5万QPS），MySQL读写分离，后端三台WebServer，根据三原则之简单原则不知是否可以扛得住？		2018-06-12
作者回复		
用户数不能当QPS，每个用户会访问很多页面，论坛用户可能评论访问20页面以上		2018-06-13
其它分析基本OK		
三月沙@wecatch		
不考虑多机房，不考虑不同地区，一个配置好点的nginx 就够了，防止单点，可以再加一台		2018-06-12
作者回复		
1000万日活的业务，你真的要这么节省么？💎💎💎💎		2018-06-13