20 | 高性能负载均衡：分类及架构

2018-06-12 李运华

20 | 高性能负载均衡：分类及架构
朗读人：黄洲君　09’14’’ | 4.23M

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，无论在哪台服务器上执行，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。

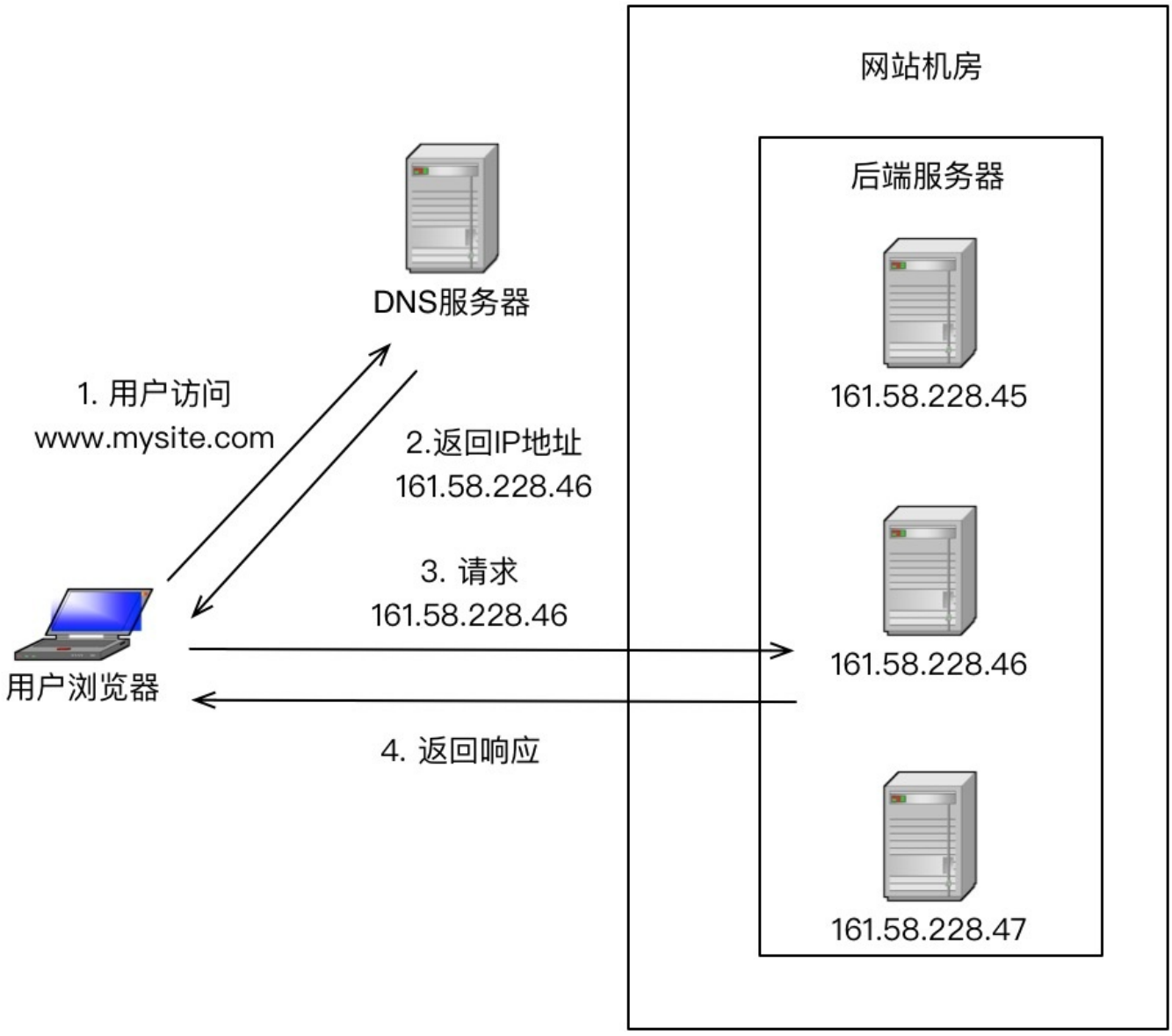
负载均衡分类

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：



（图片来源：https://ww3.sinaimg.cn/large/006tNc79gy1fc7x8apfnyj30pk0ma765.jpg）

DNS 负载均衡实现简单、成本低，但也存在粒度太粗、负载均衡算法少等缺点。仔细分析一下优缺点，其优点有：

- 简单、成本低：负载均衡工作交给 DNS 服务器处理，无须自己开发或者维护负载均衡设备。
- 就近访问，提升访问速度：DNS 解析时可以根据请求来源 IP，解析成距离用户最近的服务器地址，可以加快访问速度，改善性能。

缺点有：

- 更新不及时：DNS 缓存的时间比较长，修改 DNS 配置后，由于缓存的原因，还是有很多用户会继续访问修改前的 IP，这样的访问会失败，达不到负载均衡的目的，并且也影响用户正常使用业务。
- 扩展性差：DNS 负载均衡的控制权在域名商那里，无法根据业务特点针对其做更多的定制化功能和扩展特性。
- 分配策略比较简单：DNS 负载均衡支持的算法少；不能区分服务器的差异（不能根据系统与服务的状态来判断负载）；也无法感知后端服务器的状态。

针对 DNS 负载均衡的一些缺点，对于时延和故障敏感的业务，有一些公司自己实现了 HTTP-DNS 的功能，即使用 HTTP 协议实现一个私有的 DNS 系统。这样的方案和通用的 DNS 优缺点正好相反。

硬件负载均衡

硬件负载均衡是通过单独的硬件设备来实现负载均衡功能，这类设备和路由器、交换机类似，可以理解为一个用于负载均衡的基础网络设备。目前业界典型的硬件负载均衡设备有两款：F5 和 A10。这类设备性能强劲、功能强大，但价格都不便宜，一般只有“土豪”公司才会考虑使用此类设备。普通业务量级的公司一是负担不起，二是业务量没那么大，用这些设备也是浪费。

硬件负载均衡的优点是：

- 功能强大：全面支持各层级的负载均衡，支持全面的负载均衡算法，支持全局负载均衡。
- 性能强大：对比一下，软件负载均衡支持到 10 万级并发已经很厉害了，硬件负载均衡可以支持 100 万以上的并发。
- 稳定性高：商用硬件负载均衡，经过了良好的严格测试，经过大规模使用，稳定性高。
- 支持安全防护：硬件均衡设备除具备负载均衡功能外，还具备防火墙、防 DDoS 攻击等安全功能。

硬件负载均衡的缺点是：

- 价格昂贵：最普通的一台 F5 就是一台“马 6”，好一点的就是“Q7”了。

- 扩展能力差：硬件设备，可以根据业务进行配置，但无法进行扩展和定制。

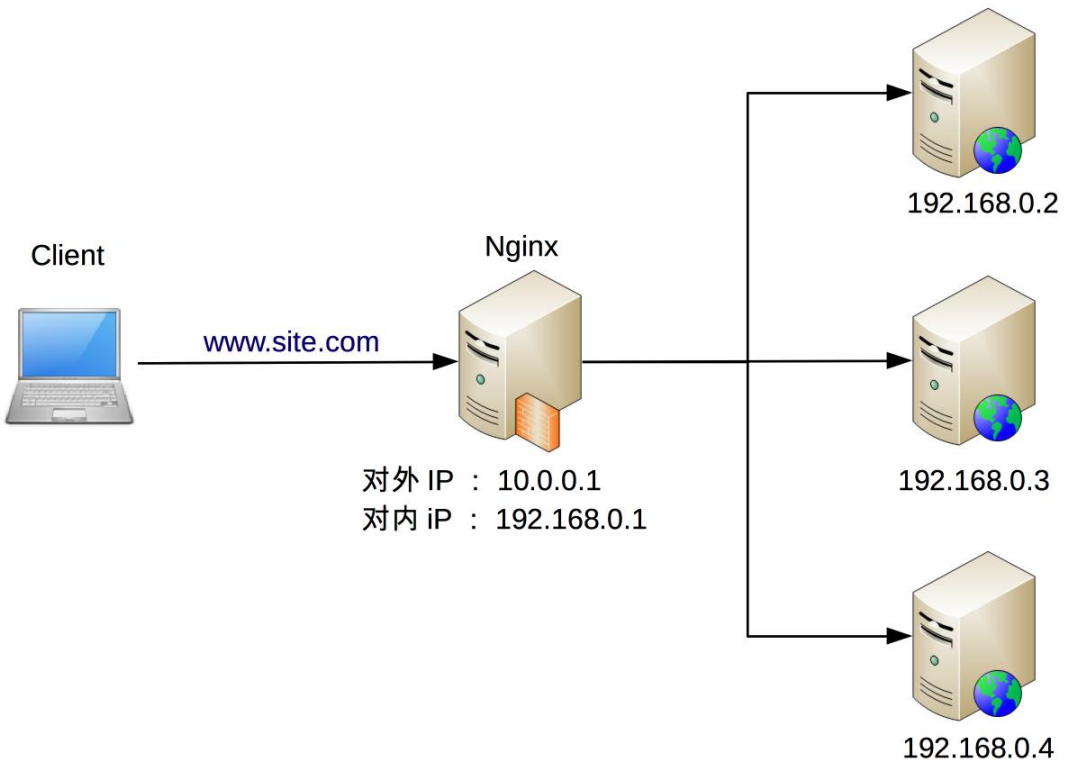
软件负载均衡

软件负载均衡通过负载均衡软件来实现负载均衡功能，常见的有 Nginx 和 LVS，其中 Nginx 是软件的 7 层负载均衡，LVS 是 Linux 内核的 4 层负载均衡。4 层和 7 层的区别就在于协议和灵活性，Nginx 支持 HTTP、E-mail 协议；而 LVS 是 4 层负载均衡，和协议无关，几乎所有应用都可以做，例如，聊天、数据库等。

软件和硬件的最主要区别就在于性能，硬件负载均衡性能远远高于软件负载均衡性能。Ngxin 的性能是万级，一般的 Linux 服务器上装一个 Nginx 大概能到 5 万 / 秒；LVS 的性能是十万级，据说可达到 80 万 / 秒；而 F5 性能是百万级，从 200 万 / 秒到 800 万 / 秒都有（数据来源网络，仅供参考，如需采用请根据实际业务场景进行性能测试）。当然，软件负载均衡的最大优势是便宜，一台普通的 Linux 服务器批发价大概就是 1 万元左右，相比 F5 的价格，那就是自行车和宝马的区别了。

除了使用开源的系统进行负载均衡，如果业务比较特殊，也可能基于开源系统进行定制（例如，Nginx 插件），甚至进行自研。

下面是 Nginx 的负载均衡架构示意图：



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。

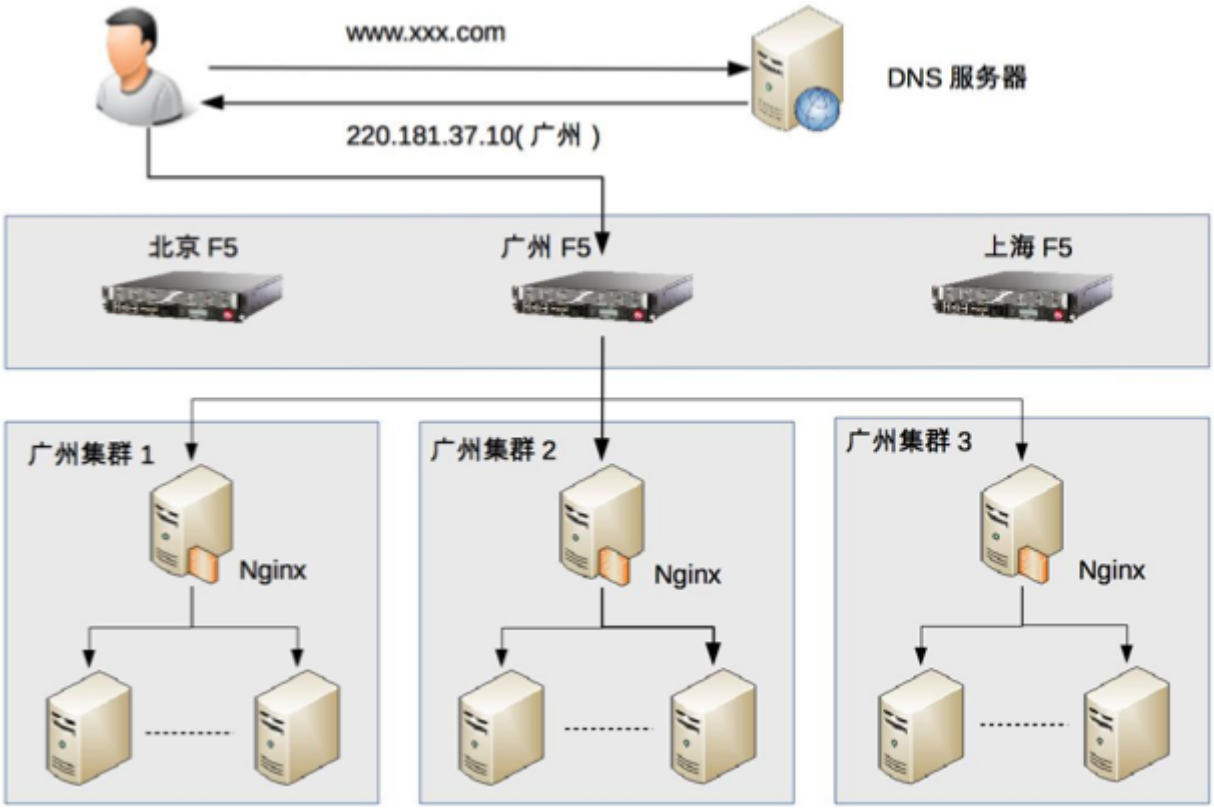
其实下面的缺点都是和硬件负载均衡相比的，并不是说软件负载均衡没法用。

- 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。



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，希望对你有所帮助。

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！）

精选留言

鹅米豆发 49

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。

2018-06-12

作者回复

思路不错👍👍

2018-06-12

食指可爱多 6

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，考虑请求的波峰波谷，波峰取qps均值的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，接入层一个nginx就可以搞定了？

2018-06-12

作者回复

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

2018-06-13

公号-Java大后端 2

通过容量规划，并考虑到高性能、高可用的要求，Web最前端可采用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。

2018-06-12

plflying 2

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、并发计算：
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！

2018-06-12

作者回复

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

2018-06-13

ant 2

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硬件用多个ngnix集群来实现，每个ngnix上会有我们自定义的路由算法。ngnix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ngnix，第一层基于自己的路由算法导流到第2层ngnix。第2层又导流到第3层。为了避免ngnix单点，每一层ngnix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

2018-06-12

作者回复

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

2018-06-13

黄金的太阳 2

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

2018-06-12

作者回复

基本正确，中国一般分南北区域接入，西部用户确实少很多

2018-06-13

何国平 1

nginx也支持4层反向代理了

2018-06-14

作者回复

我宁愿用LVS，久经考验，性能强大😄

2018-06-14

feifei 1

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

2018-06-13

作者回复

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

2018-06-14

汇通科技 1

并发测试如何来做，怎么知道自己设计的数据库，或者架构能支撑多少并发

2018-06-13

作者回复

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

2018-06-13

张玮(大圣) 1

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

2018-06-12

作者回复

最好算一下，当然有经验的架构师确实能够凭感觉预估

2018-06-13

肖一林 1

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。

2018-06-12

作者回复

这种方式也可以，dns做同机房多入口负载均衡

2018-06-13

三水 1

老师，流行的SLB还有HAProxy，我们用LVS做DNS的LB，Nginx和HAProxy做HTTP的LB。

2018-06-12

作者回复

HAProxy也很成熟，可以用

2018-06-13

星火燎原 1

不差钱的话可以考虑文中DNS +F5+ ngnix ，一般这种日活还是考虑DNS+LVS+Nginx

2018-06-12

作者回复

论坛不怎么赚钱啊😂

2018-06-13

姜泮昌 1

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，成本虽略贵，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?

2018-06-12

作者回复

下一章就介绍算法

2018-06-13

小橙橙 0

老师，有个问题一直不是很清楚，如果用nginx轮询做负载均衡，下游某个服务挂掉了，那就会导致某些请求无法响应，这样的问题要如何解决呢?

2018-07-27

作者回复

查询nginx官方文档，里面有介绍health check

2018-07-30

孙振超 0

这篇文章最大的收获是分析问题的思路，从dau出发，结合业务的特点，计算出来总的qps和tps，而后再根据通常规律计算出qps和tps的峰值，加上一定的未来发展空间和高可用冗余，结合单机能够支撑的qps和tps量，就可以计算出来整个集群的规模，有了这些数据就可以制定出比较合理的负载均衡的策略，而不是无的放矢，凭空猜测。

2018-07-21

作者回复

最常用的方式

2018-07-22

Ezekiel 0

还是要具体看业务，这个论坛是关于什么内容的论坛，是否有区域量级不同的情况，如果存在则考虑下DNS均衡，论坛应该都不怎么赚钱的，硬件均衡可以不考虑，请求量较大，LVS做集群均衡，Nginx做机器均衡感觉就可以了。论坛个人觉得做好读写分离和缓存设计才是重点。

2018-07-19

老北 0

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😂

2018-07-08

作者回复

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

2018-07-09

啃yi嘴泥 0

我也希望老师能对并发数，qps，日活等这些能够举个经典例子计算下，网上虽然有，可是总觉着讲解地不是很清楚，对这块一直都有点迷糊。

2018-07-03

作者回复

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

2018-07-04

低调的大老刘 0

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

2018-06-29

作者回复

谁都没法防DDOS攻击呀，不暴露ip，正常用户也访问不了啊😁

2018-06-29