高性能网站性能优化与系统架构

一个小型的网站，比如个人网站，可以使用最简单的 html 静态页面就实现了，配合一些图片达到美化效果，所有的页面均存放在一个目录下，这样的网站对系统架构、性能的要求都很简单，随着互联网业务的不断丰富，网站相关的技术经过这些年的发展，已经细分到很细的方方面面，尤其对于大型网站来说，所采用的技术更是涉及面非常广，从硬件到软件、编程语言、数据库、WebServer、防火墙等各个领域都有了很高的要求，已经不是原来简单的 html 静态网站所能比拟的。

大型网站，比如门户网站。在面对大量用户访问、高并发请求方面，基本的解决方案集中在这样几个环节：使用高性能的服务器、高性能的数据库、高效率的编程语言、还有高性能的 Web 容器。但是除了这几个方面，还没法根本解决大型网站面临的高负载和高并发问题。

上面提供的几个解决思路在一定程度上也意味着更大的投入，并且这样的解决思路具备瓶颈，没有很好的扩展性，下面我从低成本、高性能和高扩张性的角度来说说我的一些经验。

1、HTML 静态化

其实大家都知道，效率最高、消耗最小的就是纯静态化的 html 页面，所以我们尽可能使我们的网站上的页面采用静态页面来实现，这个最简单的方法其实也是最有效的方法。但是对于大量内容并且频繁更新的网站，我们无法全部手动去挨个实现，于是出现了我们常见的信息发布系统 CMS，像我们常访问的各个门户站点的新闻频道，甚至他们的其他频道，都是通过信息发布系统来管理和实现的，信息发布系统可以实现最简单的信息录入自动生成静态页面，还能具备频道管理、权限管理、自动抓取等功能，对于一个大型网站来说，拥有一套高效、可管理的 CMS 是必不可少的。

除了门户和信息发布类型的网站，对于交互性要求很高的社区类型网站来说，尽可能的静态化也是提高性能的必要手段，将社区内的帖子、文章进行实时的静态化，有更新的时候再重新静态化也是大量使用的策略，像 Mop 的大杂烩就是使用了这样的策略，网易社区等也是如此。目前很多博客也都实现了静态化，我使用的这个 Blog 程序 WordPress 还没有静态化，所以如果面对高负载访问，e.fs-cars.com 一定不能承受

同时，html 静态化也是某些缓存策略使用的手段，对于系统中频繁使用数据库查询但是内容更新很小的应用，可以考虑使用 html 静态化来实现，比如论坛中论坛的公用设置信息，这些信息目前的主流论坛都可以进行后台管理并且存储再数据库中，这些信息其实大量被前台程序调用，但是更新频率很小，可以考虑将这部分内容进行后台更新的时候进行静态化，这样避免了大量的数据库访问请求。

在进行 html 静态化的时候可以使用一种折中的方法，就是前端使用动态实现，在一定的策略下进行定时静态化和定时判断调用，这个能实现很多灵活性的操作，我开发的 51 学 IT 网(www.51xit.com)是使用

了这样的方法，我通过设定一些 html 静态化的时间间隔来对动态网站内容进行缓存，达到分担大部分的压力到静态页面上，可以应用于中小型网站的架构上。

2、图片服务器分离

大家知道，对于 Web 服务器来说，不管是 Apache、IIS 还是其他容器，图片是最消耗资源的，于是我们有必要将图片与页面进行分离，这是基本上大型网站都会采用的策略，他们都有独立的图片服务器，甚至很多台图片服务器。这样的架构可以降低提供页面访问请求的服务器系统压力，并且可以保证系统不会因为图片问题而崩溃。

在应用服务器和图片服务器上，可以进行不同的配置优化，比如 Apache 在配置 ContentType 的时候可以尽量少支持，尽可能少的 LoadModule，保证更高的系统消耗和执行效率。

另外，在处理静态页面或者图片、js 等访问方面，可以考虑使用 lighttpd 代替 Apache，它提供了更轻量级和更高效的处理能力。

3、数据库集群和库表散列

大型网站都有复杂的应用，这些应用必须使用数据库，那么在面对大量访问的时候，数据库的瓶颈很快就能显现出来，这时一台数据库将很快无法满足应用，于是我们需要使用数据库集群或者库表散列。

在数据库集群方面，很多数据库都有自己的解决方案，Oracle、Sybase 等都有很好的方案，常用的 MySQL 提供的 Master/Slave 也是类似的方案，您使用了什么样的 DB，就参考相应的解决方案来实施即可。

上面提到的数据库集群由于在架构、成本、扩张性方面都会受到所采用 DB 类型的限制，于是我们需要从应用程序的角度来考虑改善系统架构，库表散列是常用并且最有效的解决方案。我们在应用程序中安装业务和应用或者功能模块将数据库进行分离，不同的模块对应不同的数据库或者表，再按照一定的策略对某个页面或者功能进行更小的数据库散列，比如用户表，按照用户 ID 进行表散列，这样就能够低成本的提升系统的性能并且有很好的扩展性。sohu 的论坛就是采用了这样的架构，将论坛的用户、设置、帖子等信息进行数据库分离，然后对帖子、用户按照板块和 ID 进行散列数据库和表，最终可以在配置文件中进行简单的配置便能让系统随时增加一台低成本的数据库进来补充系统性能。

4、缓存

缓存一词搞技术的都接触过，很多地方用到缓存。网站架构和网站开发中的缓存也是非常重要。这里先讲述最基本的两种缓存。高级和分布式的缓存在后面讲述。

架构方面的缓存，对 Apache 比较熟悉的人都能知道 Apache 提供了自己的 mod_proxy 缓存模块，也可以使用外加的 Squid 进行缓存，这两种方式均可以有效的提高 Apache 的访问响应能力。

网站程序开发方面的缓存，Linux 上提供的 Memcached 是常用的缓存方案，不少 web 编程语言都提供 memcache 访问接口，php、perl、c 和 java 都有，可以在 web 开发中使用，可以实时或者 Cron 的把数据、对象等内容进行缓存，策略非常灵活。一些大型社区使用了这样的架构。

另外，在使用 web 语言开发的时候，各种语言基本都有自己的缓存模块和方法，PHP 有 Pear 的 Cache 模块和 eAccelerator 加速和 Cache 模块，还要知名的 Apc、XCache（国人开发的，支持！）php 缓存模块，Java 就更多了，.net 不是很熟悉，相信也肯定有。

5、镜像

镜像是大型网站常采用的提高性能和数据安全性的方式，镜像的技术可以解决不同网络接入商和地域带来的用户访问速度差异，比如 ChinaNet 和 EduNet 之间的差异就促使了很多网站在教育网内搭建镜像站点，数据进行定时更新或者实时更新。在镜像的细节技术方面，这里不阐述太深，有很多专业的现成的解决架构和产品可选。也有廉价的通过软件实现的思路，比如 Linux 上的 rsync 等工具。

6、负载均衡

负载均衡将是大型网站解决高负荷访问和大量并发请求采用的终极解决办法。

负载均衡技术发展了多年，有很多专业的服务提供商和产品可以选择，我个人接触过一些解决方法，其中有两个架构可以给大家做参考。另外有关初级的负载均衡 DNS 轮循和较专业的 CDN 架构就不多说了。

6.1 硬件四层交换

第四层交换使用第三层和第四层信息包的报头信息，根据应用区间识别业务流，将整个区间段的业务流分配到合适的应用服务器进行处理。 第四层交换功能就象是虚 IP，指向物理服务器。它传输的业务服从的协议多种多样，有 HTTP、FTP、NFS、Telnet 或其他协议。这些业务在物理服务器基础上，需要复杂的载量平衡算法。在 IP 世界，业务类型由终端 TCP 或 UDP 端口地址来决定，在第四层交换中的应用区间则由源端和终端 IP 地址、TCP 和 UDP 端口共同决定。

在硬件四层交换产品领域，有一些知名的产品可以选择，比如 Alteon、F5 等，这些产品很昂贵，但是物有所值，能够提供非常优秀的性能和很灵活的管理能力。Yahoo 中国当初接近 2000 台服务器使用了三四台 Alteon 就搞定了。

6.2 软件四层交换

大家知道了硬件四层交换机的原理后，基于 OSI 模型来实现的软件四层交换也就应运而生，这样的解决方案实现的原理一致，不过性能稍差。但是满足一定量的压力还是游刃有余的，有人说软件实现方式其实更灵活，处理能力完全看你配置的熟悉能力。

软件四层交换我们可以使用 Linux 上常用的 LVS 来解决，LVS 就是 Linux Virtual server，他提供了基于心跳线 heartbeat 的实时灾难应对解决方案，提高系统的鲁棒性，同时可供了灵活的虚拟 VIP 配置和管理功能，可以同时满足多种应用需求，这对于分布式的系统来说必不可少。

一个典型的使用负载均衡的策略就是，在软件或者硬件四层交换的基础上搭建 squid 集群，这种思路在很多大型网站包括搜索引擎上被采用，这样的架构低成本、高性能还有很强的扩张性，随时往架构里面增减节点都非常容易。这样的架构我准备空了专门详细整理一下和大家探讨。

总结：

对于大型网站来说，前面提到的每个方法可能都会被同时使用到，Michael 这里介绍得比较浅显，具体实现过程中很多细节还需要大家慢慢熟悉和体会，有时一个很小的 squid 参数或者 apache 参数设置，对于系统性能的影响就会很大，希望大家一起讨论，达到抛砖引玉之效。

This entry is filed under C / C++ / 其他技术, 技术交流. You can follow any responses to this entry through the RSS 2.0 feed. You can leave a response, or trackback from your own site.

(2 votes, average: 6.5 out of 10)

Loading ...

65 Responses to “说说大型高并发高负载网站的系统架构”

1

pi1ot says:

April 29th, 2006 at 1:00 pm

Quote

各模块间或者进程间的通信普遍异步化队列化也相当重要，可以兼顾轻载重载时的响应性能和系统压力,数据库压力可以通过 file cache 分解到文件系统，文件系统 io 压力再通过 mem cache 分解，效果很不错.

3

guest says:

May 1st, 2006 at 8:13 am

Quote

完全胡说八道!

“大家知道，对于 Web 服务器来说，不管是 Apache、IIS 还是其他容器，图片是最消耗资源的”,你以为是在

内存中动态生成图片啊.无论是什么文件,在容器输出时只是读文件,输出给 response 而已,和是什么文件有什么关系.

关键是静态文件和动态页面之间应该采用不同策略,如静态文件应该尽量缓存,因为无论你请求多少次输出内容都是相同的,如果用户页面中有二十个就没有必要请求二十次,而应该使用缓存.而动态页面每次请求输出都不相同(否则就应该是静态的),所以不应该缓存.

所以即使在同一服务器上也可以对静态和动态资源做不同优化,专门的图片服务器那是为了资源管理的方便,和你说的性能没有关系.

4

Michael says:

May 2nd, 2006 at 1:15 am

Quote

动态的缓存案例估计楼上朋友没有遇到过，在处理 inktomi 的搜索结果的案例中，我们使用的全部是面对动态的缓存，对于同样的关键词和查询条件来说，这样的缓存是非常重要的，对于动态的内容缓存，编程时使用合理的 header 参数可以方便的管理缓存的策略，比如失效时间等。

我们说到有关图片影响性能的问题，一般来说都是出自于我们的大部分访问页面中图片往往比 html 代码占用的流量大，在同等网络带宽的情况下，图片传输需要的时间更长，由于传输需要花很大开销在建立连接上，这会延长用户 client 端与 server 端的 http 连接时长，这对于 apache 来说，并发性能肯定会下降，除非你的返回全部是静态的，那就可以把 httpd.conf 中的 KeepAlives 为 off ，这样可以减小连接处理时间，但是如果图片过多会导致建立的连接次数增多，同样消耗性能。

另外我们提到的理论更多的是针对大型集群的案例，在这样的环境下，图片的分离能有效的改进架构，进而影响到性能的提升，要知道我们为什么要谈架构？架构可能为了安全、为了资源分配、也为了更科学的开发和管理，但是终极目都是为了性能。

另外在 RFC1945 的 HTTP 协议文档中很容易找到有关 Mime Type 和 Content length 部分的说明，这样对于理解图片对性能影响是很容易的。

楼上的朋友完全是小人作为，希望别用 guest 跟我忽悠，男人还害怕别人知道你叫啥？再说了，就算说错了也不至于用胡说八道来找茬！大家重在交流和学习，我也不是什么高人，顶多算个普通程序员而已。

5

Ken Kwei says:

June 3rd, 2006 at 3:42 pm

Quote

Michael 您好，这篇文章我看几次了，有一个问题，您的文章中提到了如下一段：

“对于交互性要求很高的社区类型网站来说，尽可能的静态化也是提高性能的必要手段，将社区内的帖子、文章进行实时的静态化，有更新的时候再重新静态化也是大量使用的策略，像 Mop 的大杂烩就是使用了这样的策略，网易社区等也是如此。”

对于大型的站点来说，他的数据库和 Web Server 一般都是分布式的，在多个区域都有部署，当某个地区的用户访问时会对应到一个节点上，如果是对社区内的帖子实时静态化，有更新时再重新静态化，那么在节点之间如何立刻同步呢？数据库端如何实现呢？如果用户看不到的话会以为发帖失败？造成重复发了，那么如何将用户锁定在一个节点上呢，这些怎么解决？谢谢。

6

Michael says:

June 3rd, 2006 at 3:57 pm

Quote

对于将一个用户锁定在某个节点上是通过四层交换来实现的，一般情况下是这样，如果应用比较小的可以通过程序代码来实现。大型的应用一般通过类似 LVS 和硬件四层交换来管理用户连接，可以制定策略来使用户的连接在生命期内保持在某个节点上。

静态化和同步的策略比较多，一般采用的方法是集中或者分布存储，但是静态化却是通过集中存储来实现的，然后使用前端的 proxy 群来实现缓存和分担压力。

一般对于一个中型网站来说，交互操作非常多，日 PV 百万左右，如何做合理的负载？

交互如果非常多，可以考虑使用集群加 Memory Cache 的方式，把不断变化而且需要同步的数据放入 Memory Cache 里面进行读取，具体的方案还得需要结合具体的情况来分析。

11

donald says:

June 27th, 2006 at 5:39 pm

Quote

请问，如果一个网站处于技术发展期，那么这些优化手段应该先实施哪些后实施哪些呢？

或者说从成本（技术、人力和财力成本）方面，哪些先实施能够取得最大效果呢？

12

Michael says:

June 27th, 2006 at 9:16 pm

Quote

donald on June 27, 2006 at 5:39 pm said:

请问，如果一个网站处于技术发展期，那么这些优化手段应该先实施哪些后实施哪些呢？

或者说从成本（技术、人力和财力成本）方面，哪些先实施能够取得最大效果呢？

先从服务器性能优化、代码性能优化方面入手，包括 webserver、dbserver 的优化配置、html 静态化等容易入手的开始，这些环节争取先榨取到最大化的利用率，然后再考虑从架构上增加投入，比如集群、负载均衡等方面，这些都需要在有一定的发展积累之后再做考虑比较恰当。

16

echonow says:

September 1st, 2006 at 2:28 pm

Quote

赞一个先，是一篇很不错的文章，不过要真正掌握里面的东西恐怕还是需要时间和实践！

先问一下关于图片服务器的问题了！

我的台球网站故人居 9tmd.com 也使用了图片服务器架构上的分离，目前是仅仅是架构上分离，物理上没有分离，由于没有钱买更多的服务器:)，大家可以看到故人居上的图片连接都是类似 img.9tmd.com 或者 img1.9tmd.com 的 URL。

这个，楼主这个 img.9tmd.com 是虚拟主机吧，也就是说是一个 apache 提供的服务吧，这样的话对于性能的提高也很有意义吗？还是只是铺垫，为了方便以后的物理分离呢？

17

Michael says:

September 1st, 2006 at 3:05 pm

Quote

echonow on September 1, 2006 at 2:28 pm said:

赞一个先，是一篇很不错的文章，不过要真正掌握里面的东西恐怕还是需要时间和实践！

先问一下关于图片服务器的问题了！

我的台球网站故人居 9tmd.com 也使用了图片服务器架构上的分离，目前是仅仅是架构上分离，物理上没有分离，由于没有钱买更多的服务器:)，大家可以看到故人居上的图片连接都是类似 img.9tmd.com 或者 img1.9tmd.com 的 URL。

这个，楼主这个 img.9tmd.com 是虚拟主机吧，也就是说是一个 apache 提供的服务吧，这样的话对于性能的提高也很有意义吗？还是只是铺垫，为了方便以后的物理分离呢？

这位朋友说得很对，因为目前只有一台服务器，所以从物理上无法实现真正的分离，暂时使用虚拟主机来实现，是为了程序设计和网站架构上的灵活，如果有了一台新的服务器，我只需要把图片镜像过去或者同步过去，然后把 img.9tmd.com 的 dns 解析到新的服务器上就自然实现了分离，如果现在不从架构和程序上实现，今后这样的分离就会比较痛苦:)

18

echonow says:

September 7th, 2006 at 4:59 pm

Quote

谢谢 lz 的回复，现在主要实现问题是如何能在素材上传时直接传到图片服务器上呢，总不至于每次先传到 web，然后再同步到图片服务器吧

19

Michael says:

September 7th, 2006 at 11:25 pm

Quote

echonow on September 7, 2006 at 4:59 pm said:

谢谢 lz 的回复，现在主要实现问题是如何能在素材上传时直接传到图片服务器上呢，总不至于每次先传到 web，然后再同步到图片服务器吧

通过 samba 或者 nfs 实现是比较简单的方法。然后使用 squid 缓存来降低访问的负载，提高磁盘性能和延长磁盘使用寿命。

20

echonow says:

September 8th, 2006 at 9:42 am

Quote

多谢楼主的耐心指导，我先研究下，用共享区来存储确实是个不错的想法!

21

Michael says:

September 8th, 2006 at 11:16 am

Quote

echonow on September 8, 2006 at 9:42 am said:

多谢楼主的耐心指导，我先研究下，用共享区来存储确实是个不错的想法!

不客气，欢迎常交流！