# 阳光-源泉






## 导航

博客园
首　页
新随笔
管　理

< 2019年2月 >

| 日 | 一 | 二 | 三 | 四 | 五 | 六 |
|---|---|---|---|---|---|---|
| 27 | 28 | 29 | 30 | 31 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |

## 搜索

找找看

## 常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

## 随笔档案(466)

2018年12月 (1)
2018年5月 (6)
2018年4月 (3)
2018年3月 (3)
2018年2月 (24)
2018年1月 (7)
2017年12月 (3)
2017年10月 (7)
2017年9月 (5)
2017年8月 (6)
2017年7月 (3)
2017年6月 (5)
2017年5月 (9)
2017年4月 (7)
2017年3月 (2)
2017年1月 (4)
2016年12月 (6)
2016年11月 (3)


# cache与负载均衡

cache

### 客户端CACHE

客户端CACHE，包括浏览器本身的缓存、FLASH存储等，用于存储一些临时的文件或者变化不大或无变化的数据：
1.如浏览器自动将用户浏览的网页存储在用户的硬盘上，下次再浏览相同的网站的时候，系统会自动从硬盘中调出该网页，既节省了时间也减少了网络的交换；
2.还有就是相同的数据需要以不同的形式在客户端展现，
合理的运用客户端CACHE功能，减少网络的访问次数，这无疑会提升客户体验，给被访问的网站加分，并且不会有硬件的费用投入，理论上讲是一种“零”投入的策略。

### 代理服务器

1.用做FQ
2.还有就是需要访问的站点访问速度比较慢，可以通过代理服务器提高访问的速度
商用的代理服务器，通常都是有CACHE功能的，访问都比较稳定，且个人信息安全性也相对要高得多；
但是普通的代理服务器就不一定有CACHE功能了，访问的稳定性也不一定能够保证，最可怕的个人信息的安全性完全没有保障，因而在考虑使用代理服务器的时候要慎重，否则不要因小失大，那可就亏大了。

### CDN，镜像站点

CDN的全称是Content Delivery Network，即内容分发网络。其目的是通过在现有的Internet中增加一层新的网络架构，将网站的内容发布到最接近用户的网络"边缘"，
使用户可以就近取得所需的内容，解决 Internet网络拥挤的状况，提高用户访问网站的响应速度。
从技术上全面解决由于网络带宽小、用户访问量大、网点分布不均等原因所造成的用户访问网站响应速度慢的问题。
(也就是一个服务器的内容，平均分部到多个服务器上，服务器智能识别，让用户获取离用户最近的服务器，提高速度。）
现阶段，国内流量比较大的网站，如阿里巴巴、淘宝、腾讯、

2016年10月 (6)
2016年9月 (8)
2016年8月 (12)
2016年7月 (20)
2016年6月 (7)
2016年5月 (7)
2016年4月 (10)
2016年3月 (10)
2016年2月 (5)
2016年1月 (1)
2015年12月 (2)
2015年11月 (7)
2015年10月 (2)
2015年9月 (10)
2015年8月 (9)
2015年7月 (15)
2015年6月 (8)
2015年5月 (14)
2015年4月 (30)
2015年3月 (15)
2015年2月 (16)
2015年1月 (15)
2014年12月 (9)
2014年11月 (6)
2014年10月 (15)
2014年9月 (17)
2014年8月 (6)
2014年7月 (11)
2014年6月 (28)
2014年5月 (28)
2014年4月 (9)
2014年3月 (4)

**相册**(7)

故障(3)
知识图(4)

新浪、网易等都有使用CDN技术，这给用户的感觉就是在任何地方访问都比较快。

镜像站点、CDN这二者之间有一点比较类似，都是通过增加服务结点来降低访问的拥挤度，并提高用户访问的速度。他们的差别是镜像服务本身是一个独立的服务器，

内容完全来自于主服务器，通过定期到主服务器去更新内容而保持内容与主服务器基本一致，如华军软件园的实现就是很多的镜像站点，直接访问该镜像的请求都全部去了该镜像服务器，

这就会出现不同的镜像站点会有不同的访问压力，也就是没有导流、负载均衡的作用，但如果镜像站点出现了问题，是没有被其它服务器代理的，

就相当于你访问的其它网站出现问题一样：挂了；CDN其实也是代理，只是他比镜像服务更智能的代理，如果当前服务器访问量过大时，可以将流量导到其它的服务器，提升响应的时间，

它的优势在于几乎涵盖国内所有线路，而在可靠性上，CDN 在结构上实现了多点的冗余，即使某一个节点由于意外发生故障，对网站的访问能够被自动导向其他的健康节点进行响应。

CDN能轻松实现网站的全国铺设，不必考虑服务器的投入与托管、不必考虑新增带宽的成本、不必考虑多台服务器的镜像同步、不必考虑更多的管理维护技术人员。

## squid

## 中心数据CACHE服务器 (内存cache)

以上谈到的都是应用以外的CACHE，以增加服务器节点提升用户访问速度，有没有办法使系统在获取数据时不是每次都和数据库打交道，减少I/O的开销，从而提访问速度呢？答案是肯定的，

阿里使用的是内存CACHE（memcached），将经常访问、不易变化且CACHE命中率比较大的数据，放到CACHE服务器中，如阿里助手的菜单，这些数据都是不怎么变化，放到CACHE中应用在获取菜单数据时，

速度就会提高，从而提高应用响应时间，也可以说是提高了用户的访问速度。

在选择，需要注意几点：

1、CACHE的可靠性，在大量访问及使用的情况，能够保证数据的可靠性；水平伸缩性比较好，这方便于快速提高CACHE的系统容量；CACHE要能够快速数据恢复，

当其中某台CACHE出现问题的时候，其它的CACHE能够在短时间之内代替它的工作；

2、CACHE的有效性，要支持定期的数据失效策略，为其它需要使用CACHE的数据提供存储空间。
使用CACHE的时候，也需要注意CACHE的命中率要高，意思就是CACHE中要存放经常访问、不易变化的数据，这才能够体现CACHE的价值，如果将阿里每个用户的个人信息都放到CACHE中，这是没有意义的。

**JVM CACHE (另一种内存cache)**

这里说的JVM CACHE，也是将数据放到内存中，算是另外一种内存CACHE，如我们在代码中写的静态对象，只是这种CACHE是没有过期策略的，除了绝对不变的数据以外，
如配置文件及参数可以保存JVM中，不推荐使用JVM CACHE；
阿里助手的菜单使用的是MEMCACHE，没有使用JVM CACHE，因为菜单有些时候也需要做变更的，这个时候要使变更立马的反映出来，
可以通过手动清理MEMCACHE的方式，而JVM CACHE是不方便清理的，并且如果又是在集群环境中，完全清理的成本将是非常大的，而使用中心CACHE服务器，只需清理一次就完全清除了；
如果菜单出不来或者数据不正确，那至少是A类故障。
另外还有一些数据库中间件，如IBATIS、HIBERNATE等，也都支持数据缓存，支持数据自动过期策略，可以设定CACHE的数量，如下是IBATIS的CACHE配置实现：

**数据库CACHE**

数据库本身也是有CACHE的，这里我们就不继续究下去了，有兴趣的可以去研究研究。

本篇介绍CACHE，是从离用户最近到用户最远的顺序进行介绍的，让你可以一层层的去了解合适的优化策略，比较泛泛，没有什么深入的东西，都只是从表面上谈到了一些东西，具体的场景和使用，还需要根据具体情况而定。

http://www.blogjava.net/DLevin/archive/2013/10/15/404770.html

硬件的缓存系统，如cpu的一级，二级，硬盘的缓存等
在软件的缓存系统中，一般是为了解决内存的访问速率和磁盘、网络、数据库（属于磁盘或网络访问，单独列出来因为它的应用比较广泛）等访问速率不匹配的问题（对于内存缓存系统来说）。
但是由于内存大小和成本的限制，我们又不能把所有的数据先

加载进内存来。因而如CPU中的缓存，我们只能先将一部分数据保存在缓存中。此时，对于缓存，我们一般要解决如下需求：

记得我最早接触Cache是在大学学计算机组成原理的时候，由于CPU的速度要远大于内存的读取速度，为了提高CPU的效率，CPU会在内部提供缓存区，该缓存区的读取速度和CPU的处理速度类似，CPU可以直接从缓存区中读取数据，从而解决CPU的处理速度和内存读取速度不匹配的问题。缓存之所以能解决这个问题是基于程序的局部性原理，即”程序在执行时呈现出局部性规律，即在一段时间内，整个程序的执行仅限于程序中的某一部分。相应地，执行所访问的存储空间也局限于某个内存区域。局部性原理又表现为：时间局部性和空间局部性。时间局部性是指如果程序中的某条指令一旦执行，则不久之后该指令可能再次被执行；如果某数据被访问，则不久之后该数据可能再次被访问。空间局部性是指一旦程序访问了某个存储单元，则不久之后。其附近的存储单元也将被访问。”在实际工作中，CPU先向缓存区读取数据，如果缓存区已存在，则读取缓存中的数据（命中），否则（失效），将内存中相应数据块载入缓存中，以提高接下来的访问速度。由于成本和CPU大小的限制，CPU只能提供有限的缓存区，因而缓存区的大小是衡量CPU性能的重要指标之一。
使用缓存，在CPU向内存更新数据时需要处理一个问题（写回策略问题），即CPU在更新数据时只更新缓存的数据（write back，写回，当缓存需要被替换时才将缓存中更新的值写回内存），还是CPU在更新数据时同时更新缓存中和内存中的数据（write through，写通）。在写回策略中，为了减少内存写操作，缓存块通常还设有一个脏位（dirty bit），用以标识该块在被载入之后是否发生过更新。如果一个缓存块在被置换回内存之前从未被写入过，则可以免去回写操作；写回的优点是节省了大量的写操作。这主要是因为，对一个数据块内不同单元的更新仅需一次写操作即可完成。这种内存带宽上的节省进一步降低了能耗，因此颇适用于嵌入式系统。写通策略由于要经常和内存交互（有些CPU设计会在中间提供写缓冲器以缓解性能），因而性能较差，但是它实现简单，而且能简单的维持数据一致性。

Java中的Cache库

在Java社区中已经提供了很多Cache库实现，具体可以参考http://www.open-open.com/13.htm，这里只关注自己用到的几个Cache库而且这几个库都比较具有代表性：

Guava中提供的Cache是基于单JVM的简单实现；
EHCache出自Hibernate，也是基于单JVM的实现，是对单JVM Cache比较完善的实现；
而Gemfire则提供了对分布式Cache的完善实现。
这一系列的文章主要关注在这几个Cache系统的实现上，因而不探讨关于Cache的好处、何时用Cache等问题，由于他们都是基于内存的Cache，
因而也仅局限于这种类型的Cache（说实话，我不知道有没有其他的Cache系统，比如基于文件？囧）。

http://www.iteye.com/topic/21810

1、Varnish缓存安装比较简单，主要是注意修改Varnish的端口号为80，然后把Apache和Nginx的端口号修改为非80，Memcached缓存各大VPS控制面板都会提供直接安装方法，安装后在探针中可以检测到。
2、总得来说Varnish和Memcached都属于内存级别的缓存加速，Varnish即使服务器重启也会原有的缓存也不丢失，因此适合页面缓存，而Memcached更适合MysqL数据库缓存，可以大大减少数据库压力。

Varnish是一款高性能的缓存加速器，Varnish把数据存放在服务器的内存中，利用内存可以极大的提高PHP页面执行速度，可以设置0～60秒的精确缓存时间，32位的机器支持的缓存文件最大为2 GB。

Varnish采用VCL的配置，而且具有强大的管理功能，如top、stat、admin、lis，管理方式比较灵活。Varnish的状态机设计不仅巧妙，结构也很清晰，利用二叉堆管理缓存文件，即可达到随时删除的目的。

Memcached是一个高性能的分布式内存对象缓存系统，通过在内存中缓存数据和对象来减少读取数据库的次数，从而提高动态、数据库驱动网站的速度。Memcached对于减少MysqL数据查询压力非常有帮助。

由于Varnish采用了Visual Page Cache技术，所有缓存的数据都直接从内存读取，而Squid从硬盘读取缓存的数据，所以

Varnish在访问速度方面会更快一些。但是Varnish在高并发状态下，CPU、I/O和内存等资源的开销高于Squid。

缓存分类
代理缓存：客户端请求代理，先去找缓存，缓存没有，代理会去上游服务器找到资源，并缓存在代理，然后返回给客户端
旁路缓存：客户端去缓存找缓存，缓存没命中返回客户端，客户端去上游服务器找到资源返回到本地，然后再把资源缓存到缓存

Memcache适用的场景
memcache的缺点：不能适应实时更新，如果实时更新，缓存不命中，命中率低。
memcache支持分布式缓存，有mysql主从就不需要memcache，memcache适合多台mysql集群环境，此时直接到mysql缓存取查询性能较好

总是有人在问cache用什么，有varnish，squid，apache，nginx这几种，我们到底用什么架构的cache。

1、从这些功能上。varnish和squid是专业的cache服务，而apache，nginx这些都是第三方模块完成。
2、要做cache服务的话，我们肯定是要选择专业的cache服务，优先选择squid和varnish。
varnish本身的技术上优势要高于squid，它采用了“Visual Page Cache”技术，在内存的利用上，Varnish比Squid具有优势，它避免了Squid频繁在内存、磁盘中交换文件，性能要比Squid高。varnish是不能cache到本地硬盘上的。
还有强大的通过Varnish管理端口，可以使用正则表达式快速、批量地清除部分缓存
squid的优势在于完整的庞大的cache技术资料，和很多的应用生产环境（这应该与squid早出来有关）。
3、谈谈nginx，nginx是用第三方模块ncache做的缓冲，其性能基本达到varnish，但在架构中nginx一般作为反向（静态文件现在用nginx的很多，并发能支持到2万+）。在静态架构中，如果前端直接面对的是cdn或者前端了4层负载的话，完全用nginx的cache就够了。
4、本人觉得如果是在apache服务上提升性能，做一些本地cache是完全可以的，但如果在系统架构中用apache做cache服

务，那就有点牛头不对马尾了。

http://blog.csdn.net/gzh0222/article/details/8540604 lvs、haproxy、nginx 负载均衡的比较分析

分类: linux概念



« 上一篇：shell之here文档
» 下一篇：linux服务之varnish

posted on 2014-06-27 17:06 阳光-源泉 阅读(737) 评论(0) 编辑 收藏

**相关博文：**

· nginx 负载均衡-反向代理+cache浅谈
· Nginx网络负载均衡,负载均衡,网络负载,网络均衡
· 负载均衡配置与使用
· Tomcat集群与负载均衡
· lighttpd 负载均衡-反向代理+cache浅谈

**最新新闻**：

· 张一鸣豪赌千亿营收，但字节跳动仍将面临三重难关
· 马斯克私有化推文影响犹在 特斯拉还在应付股东集体诉讼案
· 字节跳动新增商标“字节锤子”
· 记者探访FF美国总部：贾跃亭没忘记乐视债务，也不愿放弃造车梦
· MIT和微软的新成果，能否帮自动驾驶摆脱成长的烦恼？
» 更多新闻...