

基于 Memcached 内存对象缓存技术应用研究

刘 亮¹, 徐步东¹, 谭艳艳²

(1. 山东师范大学 信息技术管理处, 山东 济南 250014;

2. 山东师范大学 信息科学与工程学院, 山东 济南 250014)

摘 要:内存对象缓存技术近年来受到业界和学界的广泛关注, Memcached 既是一种分布式缓存服务器, 也是分布式内存对象缓存系统的典型代表。其以最小的资源消耗获得高效的服务性能优势, 广泛应用于 Web 应用系统中。Memcached 自身的内存管理机制、分布式算法并非完美, 对此, 文中以提升 Web 应用系统性能为目标, 利用分布式算法, 基于 Memcached 对内存对象缓存技术进行深入研究, 通过一致性 Consistent Hashing 算法减轻因添加服务器节点而造成缓存命中率降低的代价, 提高增加服务器节点后的缓存命中率, 并将优化后的 Memcached 应用于某电子商务平台之中。测试表明, 此电子商务应用系统的请求数、流量、TCP 连接数以及 CPU 利用率等技术指标改善明显, 利用 Memcached 能够有效减轻数据库服务器负载压力, 提高 Web 应用系统响应速度, 增强用户体验效果。

关键词:内存对象; 缓存系统; 分布式算法; 电子商务

中图分类号: TP311.52

文献标识码: A

文章编号: 1673-629X(2015)11-0204-05

doi: 10.3969/j.issn.1673-629X.2015.11.041

Research on Memory Object Cache Technology Application Based on Memcached

LIU Liang¹, XU Bu-dong¹, TAN Yan-yan²

(1. Information Technology Management Office of Shandong Normal University,
Jinan 250014, China;

2. College of Information Science and Technology, Shandong Normal University,
Jinan 250014, China)

Abstract: The memory object cache technology has been widely concerned in the industry and academia in recent years. The Memcached is a kind of distributed cache server, it is a typical representative of distributed memory object caching system, and it is getting efficient service performance advantages by minimal consumption of resources, so it is widely used in the Web application system. The memory management mechanism and distributed algorithm owned by Memcached are not perfect. As the important target to enhance the performance of Web applications, using the distributed algorithm, in-depth researching on the memory object caching technology based on Memcached, reduce the cost of the cache hit rate decreased as result of the addition server nodes by Consistent Hashing algorithm, improve the cache hit rate after increased the server node, and the improved Memcached is applied to an e-commerce platform. Tests show that the number of requests, traffic, TCP connections and CPU utilization rate of the e-commerce application systems are improved significantly, using Memcached can effectively reduce database server load pressure and improve Web application system response speed, enhancing the user experience effect.

Key words: memory object; cache system; distributed algorithm; e-commerce

0 引 言

响应速度是基于 Web 技术的应用系统首要面对的关键问题, 经过业界学者的不懈探索, 从最初在线程

内缓存对象, 到共享内存缓存机制, 系统仅仅局限于一台计算机上。内存缓存技术基于操作系统原理中非常经典的理论, 即 20% 的数据通常在 80% 的时间内被访

收稿日期: 2015-03-11

修回日期: 2015-06-17

网络出版时间: 2015-11-04

基金项目: 国家自然科学基金资助项目(61401260)

作者简介: 刘 亮(1984-), 男, 硕士, 软件设计师, 助理工程师, 研究方向为电子商务、信息安全、动态优化算法; 徐步东, 高级实验师, 研究方向为计算机应用、多媒体技术。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20151104.0953.098.html>

问^[1],因此应用系统的目标就可以集中在这 20% 的数据中,当数据被首次访问后,就将其置入内存中,再次访问这些数据时,只需从内存中读取并赋予更高访问级别即可。

目前,国内外对缓存技术的研究现状主要有两方面,一是对缓存技术算法的改进,二是对缓存技术架构的应用创新。GLakshaman 认为,基于客户端的代理服务调度算法能选择最快的代理服务器节点,提高文档获取速度^[2]。Rosu 以 GLakshaman 的算法为基础,将代理加速器设置于代理服务器之前,并缓存代理服务器中的热点对象,提高了代理服务器的响应速度以及整体性能^[3]。在 QoS-Capable 网络的应用中,M. Rezvan 和 K. Pawlikowski 所提出的分布式缓存体系架构降低了 QoS 路由算法的计算量,并缓存已有的计算结果,提高了路由的整体速度^[4]。Sinisa Srbljic 等在分布式网络的网络冲突、代理服务器过载等问题中,采用最佳缓存代理服务器数量的解决策略,促进了缓存技术在分布式网络中的应用^[5]。

1 Memcached 的技术研究

Memcached 通过缓存数据对象于内存中,降低数据库访问次数,减轻数据库负载,提高了 Web 应用系统的动态访问速率,增强了 Web 应用系统的灵活性和可扩展性,是分布式内存对象缓存系统的典型代表。

Memcached 也是一种分布式缓存服务器,在高速运行的过程中具备以下特点:

- (1) 协议简单;
- (2) 基于 libevent 程序库的事件处理方式;
- (3) 内置内存存储方式;
- (4) 非互相通信的分布式。

Memcached 并非适用于所有应用,对于非分布、非共享、规模足够小的服务器应用,由于网络连接等资源的限制,Memcached 会拖慢系统效率。在应用中经过测试,其本地读写速度比直接读取内存数据慢 10 倍之多。通常情况,Memcached 多用于数据库前端,以减少 SQL 解析、磁盘读写等数据库操作,并且其在内存中管理数据,比直接读取数据库具有更高的速率和性能。

1.1 Memcached 内存管理机制

Memcached 是一个基于存储键/值对的 Hashmap,它通过三十二位元的循环冗余校验(CRC-32)计算键值后,将数据分散于不同的机器上,进程本身非常轻,仅占用很少的 CPU 资源。

Memcached 采用对象缓存分配机制(Slab Allocator)分配、管理内存。在此之前,操作系统对内存的分配,仅仅简单地对所有数据记录进行动态内存分配(malloc)和释放(free),从而导致内存碎片的增多,增

加了内存管理器的任务,严重情况下,会造成操作系统本身比 Memcached 进程还慢。鉴于此,Slab Allocator 将分配的内存(Page,默认 1 MB)分割成预先规定的特定长度的块来解决内存碎片问题。Slab Allocator 将分配的内存分割成块(Chunk),块的大小不一,然后把相同大小的块组成组(Slab Class,Chunk 的集合),并重复利用已分配的内存^[6]。

Memcached 服务器端获取客户端发送的数据,按照数据大小,选择最适合的 Slab。Slab 中的空闲 Chunk 列表存储于 Memcached 服务器中,服务器则根据该列表选择相应的 Chunk 缓存数据。

尽管 Slab Allocator 机制有效解决了内存碎片问题,但由于分配的内存是特定长度,造成服务器无法加以有效利用。例如,128 字节的 Chunk 缓存 100 字节的数据,会存在 28 字节的浪费^[6]。对此问题的解决方案没有办法做到完美,但在两种情况下,可以使用适合数据大小的 Slab Classes 列表以提高内存利用率,一是客户端发送数据的公用大小在预知的情况下,二是在缓存数据的大小相同的情况下。尽管无法达到最优化,但通过设置 Memcached 的 Growth Factor 因子能调节 Slab 间的差异。Growth Factor 因子默认值 $f = 1.25$ ^[7]的 10 组输出,见表 1。

表 1 Chunk 空间的使用

Slab Class	Chunk Size	Perslab
1	88	11 915
2	112	9 362
3	144	7 281
4	184	5 698
5	232	4 519
6	296	3 542
7	376	2 788
8	472	2 221
9	592	1 771
10	744	1 409

表 1 中输出结果计算误差的设置是为了保持字节数的对齐。因此,将 Memcached 引入 Web 应用中时,数据的预期平均长度需重新计算,从而调整 Growth Factor 获得最优设置。对于已分配的内存,Memcached 并不释放,记录超时后,其存储空间即可被重复利用。Memcached 采用 lazy expiration 技术在获取数据时检查其时间戳,判断记录是否过期,因此系统不会在内部过期监视上消耗 CPU 时间。Memcached 对已超时记录的空间有优先使用权,但也难免在新增记录时会发生分配空间不足的情况,因此 LRU(Least Recently Used)算法经常被用来删除最近最少使用的记录以释放内存

空间,并将其分配给新的记录^[8]。

1.2 Memcached 的分布式

Memcached 是“伪分布式”缓存服务器,其实现非常简单,仅包括内存存储功能,而其分布式功能是由客户端实现的。 $node_1$ 、 $node_2$ 和 $node_3$ 表示三台 Memcached 服务器,应用程序将要保存键名为“shijingshan”,“chaoyang”,“haidian”,“dongcheng”和“xicheng”的数据。

首先,Web 应用向 Memcached 中添加“shijingshan”,客户端获取 Web 应用传来的“shijingshan”,然后根据“键”选择存储数据的 Memcached 服务器,并存储“shijingshan”及其值。同理,“chaoyang”,“haidian”,“dongcheng”,“xicheng”都是先选择服务器再保存键值。

其次,Web 应用获取数据时,把数据的键“shijingshan”发送给客户端,客户端依据“键”以及数据存储时的算法选择 Memcached 服务器,然后执行获取命令,只要数据没过期或被删除,就能获得原来保存的数据。

因此,Memcached 分布式,就是根据“键”值的不同把数据存储到不同的服务器上,实现数据的分布式存取。随着 Memcached 服务器的增加,键值 Key 分散于不同的服务器中,因此如果部分 Memcached 服务器发生宕机,对其他缓存服务器没有影响,系统能够正常运行。客户端的分布式使用循环冗余码校验(CRC)算法求得键的整数哈希值,再除以服务器数量,然后根据其余数来选择数据缓存的服务器。示例中,由 CRC 算法求得“键”所对应的服务器为: shijingshan \rightarrow $node_2$, chaoyang \rightarrow $node_3$, chaoyang \rightarrow $node_3$, haidian \rightarrow $node_2$, dongcheng \rightarrow $node_1$, xicheng \rightarrow $node_1$ 。

根据该结果,“shijingshan”分散到 $node_2$,“chaoyang”分散到 $node_3$ 等。如果服务器节点无法连接,则客户端在“键”后添加连接次数,并再次由 CRC 算法计算哈希值并尝试连接,这称之为 rehash。CRC 算法易于理解并具有优秀的分散性能,但是服务器的添加或删除,使得计算后的余数发生变化,造成无法找到保存数据时的服务器,影响缓存命中率。例如,将“a”到“z”的键按照 CRC 算法存储到 Memcached,服务器为 $node_1$ 、 $node_2$ 和 $node_3$,其对应键选择的服务器为:

$node_1$: a, c, d, e, h, j, n, u, w, x

$node_2$: g, i, k, l, p, r, s, y

$node_3$: b, f, m, o, q, t, v, z

输出结果显示,每台服务器存储 8 ~ 10 个数据,然后增加一台服务器 $node_4$,其对应键选择的服务器为:

$node_1$: d, f, m, o, t, v

$node_2$: b, i, k, p, r, y

$node_3$: e, g, l, n, u, w

$node_4$: a, c, h, j, q, s, x, z

可见,服务器命中的仅有 d, i, k, p, r, y 六个字符,增加服务器节点使得缓存重组,键与缓存服务器之间对应关系发生变化,只有六个能够在原服务器获取到,命中率非常低,仅有 23%。

在 Web 应用系统中添加 Memcached 服务器,系统缓存命中率瞬时下降,数据库服务器的访问压力大幅度增大,甚至造成数据库服务器宕机,停止数据服务的严重情况^[9]。对于上述问题,改进后的一致性 Consistent Hashing 分布式算法能避免添加 Memcached 服务器所带来的困扰。

Consistent Hashing 算法如图 1 所示。

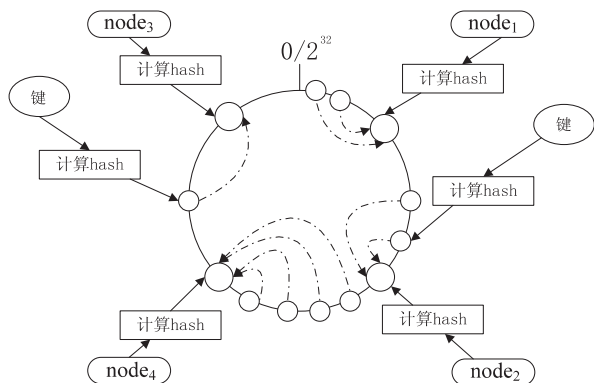


图 1 Consistent Hashing: 基本原理

(1) 在 $0 \sim 2^{32}$ 的圆上,计算并配置 Memcached 服务器节点的哈希值;

(2) 依据(1)中同样的算法,计算并映射存储数据 Key 的哈希值;

(3) 在此数据所映射的位置顺时针查找第一个 Memcached 服务器节点,存储数据,若超出 2^{32} ,则将数据存储到第一个节点上。

把图 1 作为初始状态,在此之上增加一个服务器节点,“键”值的存储采用 CRC 算法,缓存的命中率会变得非常低,而对于 Consistent Hashing 算法,影响的只是增加服务器节点的逆时针方向的第一个服务器节点,如图 2 所示。

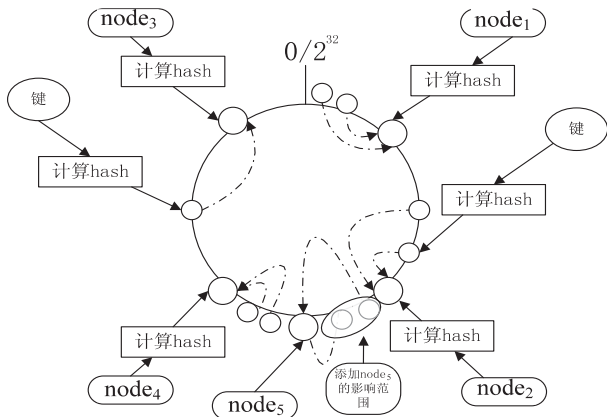


图 2 Consistent Hashing: 添加服务器

因此,Consistent Hashing 算法对键 Key 的重新分布进行了最大限度抑制,从而减轻添加服务器节点造成缓存命中率降低的困扰。使用 Consistent Hashing 算法,设服务器节点数为 n ,增加的服务器节点数为 m ,则增加服务器节点后的缓存命中率为:

$$\left(1 - \frac{n}{m + n}\right) \times 100\% \tag{1}$$

而且,如果采用虚拟节点的思想,在圆上为每个服务器节点分配 100~200 个点,就能最大限度地抑制分布不均匀,从而减小服务器节点增减的代价^[10]。

2 Memcached 的应用研究

2.1 Memcached 在电子商务系统架构中的应用

近年来,电子商务的蓬勃发展给电子商务应用平台带来了巨大考验,用户访问量的暴增使得数据库服务器压力倍增。将 Memcached 技术应用于多数据库、多服务器的某个电子商务平台架构中,以缓解数据库服务器压力,提高用户访问速度^[11]。

在电子商务平台架构方案之初,就考虑到随着平台客户访问量和用户数据量的急剧增加,仅仅通过添加 slave 的方式无法缓解目前的问题,故将 Memcached 技术引入方案之中。经验证,Memcached 的引入使得系统的速度和稳定性都得到了很大改善。引入 Memcached 的电子商务平台体系架构^[12]如图 3 所示。

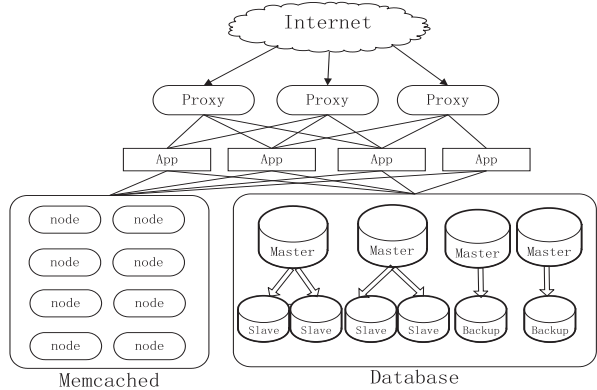


图 3 测试项目平台的系统组件

由于 32 位操作系统的进程至多占用 2 GB 内存,因此电子商务平台架构中的 Memcached 服务器采用普通商用 PC,4 GB 内存,X86_64 操作系统,分配给 Memcached 的内存为 3 GB,以防止可能导致的内存交换,导致数据访问命中率的降低^[13]。

在电子商务平台系统中,将 m 台 Memcached 服务器作为一个大的缓冲池,则整个系统有 $3m$ GB 的内存缓存。Memcached 客户端与服务器的交互采用 Consistent Hashing 分布式算法^[14]。

另外,电子商务平台统一命名规范,系统将符合规则的用户共享数据自动存储到多台 Memcached 服

器中,建立数据冗余,获取数据时仅从中选取一台服务器即可,极大地减少了由于共享数据集中于一台服务器而导致的因服务器故障命中率降低的问题^[15]。

2.2 Memcached 效果评价

Apache JMeter 是一个基于 Java 平台、对 Web 应用系统进行压力测试的工具,也可用于模拟服务器、网络等承载巨大负载的情况,分类测试应用系统的强度和性能,并给出数据分析报告。

利用 JMeter 对电子商务平台进行测试,将测试结果抽样,用户请求访问的 Memcached 服务器监控情况如图 4 所示。

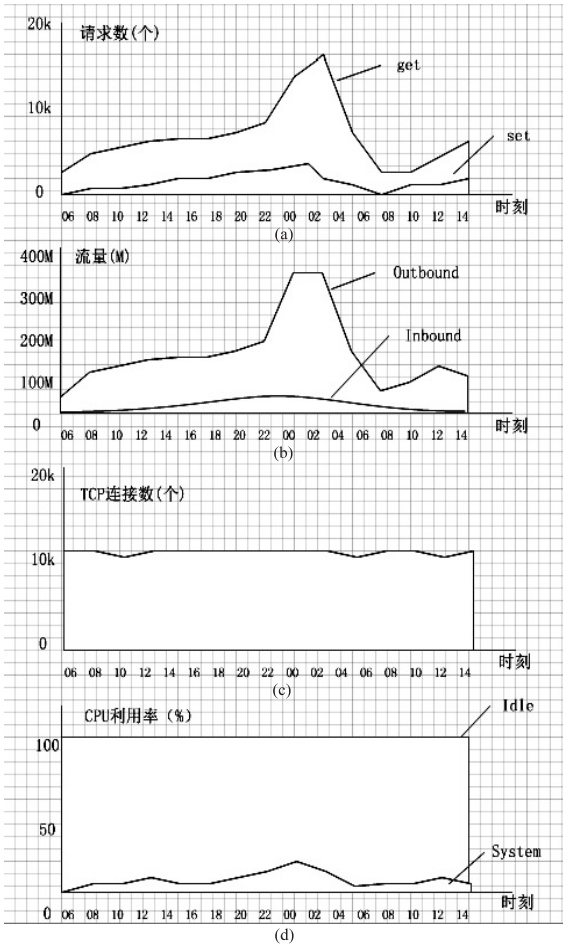


图 4 测试结果

由此,请求数峰值为 15 000 qps,流量达 400 Mbps,连接数超过 10 000 个,此时的 CPU 利用率如图 4 所示。

可见,虽然 CPU 仍有闲置,但 Memcached 服务器性能已经非常高,非常适合缓存 Web 应用系统数据。使用 Memcached 和不使用 Memcached 的电子商务平台测试结果如表 2 所示。

表 2 中,Label 为定义的 HTTP 请求名称;Samples 为此次测试中发出的请求总数;Average 为访问页面的平均响应时间;Min 为访问页面的最小响应时间;Max 为访问页面的最大响应时间;Error 为错误的请求数量

比请求总数的百分比。

表 2 测试结果

测试参数	使用 Memcached	未使用 Memcached
Label	Http://10. 5. 113. 235/	Http://10. 5. 113. 235/
Samples	100	100
Average/ms	27	91
Min/ms	11	40
Max/ms	45	200
Error/%	5	17

由测试结果可见,使用 Memcached 能够减少用户访问响应时间,提高访问命中率,减少错误请求出现率,有效地提高电子商务平台的运行效率。

3 结束语

文中基于 Memcached 对 Web 应用中的内存对象缓存技术进行深入研究,并将其应用于某电子商务平台应用系统中,减轻了数据库访问压力,提高了数据访问效率,增强了用户体验。未来 Memcached 的发展,将以二进制协议的实现和外部引擎加载机制为目标,使得基于 Memcached 的内存对象缓存技术性能更高,应用范围更为广泛。

参考文献:

[1] 林 恺. 基于内存缓存技术的社区网站设计与实现[D]. 厦门:厦门大学,2011.

[2] Li Fajie. Research on using memcached in call center[C]//Proc of international conference on computer science and network technology. [s. l.]:[s. n.],2011:1721-1723.

[3] 符青云,刘心松. 基于分布式协同缓存机制的 Web Proxy 研究[J]. 计算机工程,2007,33(11):120-122.

[4] Chen G, Wang C L, Lau F C M. Building a scalable web

server with global object space support on heterogeneous clusters[C]//Proceedings of IEEE international conference on cluster computing. Newport Beach,California;IEEE,2001:313-320.

[5] Xiao Li,Chen Xin,Zhang Xiaodong,et al. On scalable and locality-aware web document sharing[J]. Journal of Parallel and Distributed Computing,2003,63:945-962.

[6] 徐剑强,邹伟平. Memcached 应用研究[J]. 科技广场,2012(7):95-97.

[7] 南 轶,李先国. 基于. NET Cache+Memcached Web 缓存技术的研究与应用[J]. 科学技术与工程,2011,11(31):7808-7811.

[8] 宗小忠. 基于 Memcached 构建 Web 缓存服务器[J]. 电脑知识与技术,2011,7(5):1044-1045.

[9] 杨立身,曹志义. 内存缓存技术在门户网站开发中的应用研究[J]. 电脑知识与技术,2008,3(7):1415-1416.

[10] Wang X,Zhou B,Li W. A streaming protocol for memcached[J]. Information Technology Journal,2012,11(12):1776-1780.

[11] Xu Chuncong,Huang Xiaomeng,Wu Nuo,et al. Using memcached to promote read throughput in massive small-file storage system[C]//Proc of 9th international conference on grid and cooperative computing. Nanjing:IEEE,2010:24-29.

[12] Jose J,Subramoni H,Luo Miao,et al. Memcached design on high performance RDMA capable interconnects[C]//Proc of international conference on parallel processing. [s. l.]:[s. n.],2011:743-752.

[13] 王心妍. Memcached 和 Redis 在高速缓存方面的应用[J]. 无线互联科技,2012(9):8-9.

[14] 俞华锋. Memcached 在大型网站中的应用[J]. 科技信息,2008(1):70-70.

[15] 涂婷婷,王 晶,王 纯. Memcached 在彩印 SNS 网站上的应用[J]. 电信工程技术与标准化,2012,25(9):86-90.

(上接第 203 页)

检测[J]. 计算机工程与应用,2003,39(18):75-76.

[11] 张桂林,陈益新,李 强,等. 基于灰度与边缘的图像分割方法[J]. 华中理工大学学报,1994,22(5):7-10.

[12] 李 军,丁明跃. 超声图像中胎儿股骨的自动检测与测量[J]. 红外与激光工程,2000,29(4):22-25.

[13] 罗家燕,卢瑞祥,林 珠,等. 计算机化骨龄评价系统的研制[J]. 第四军医大学学报,1998,19(1):115-116.

[14] Fan Bo-Chun,Heish Chi-Wen,Jong Tai-Lang,et al. Automatic bone age estimation based on carpal-bone image-a pre-

liminary report[J]. Chinese Medical Journal (Taipei),2001,64:203-208.

[15] Somkantha K,Theera-Umpun N,Auephanwiriyakul S. Bone age assessment in young children using automatic carpal bone feature extraction and support vector regression[J]. Journal of Digital Imaging,2011,24(6):1044-1058.

[16] Hsieh C W,Chen C Y,Jong T L,et al. Automatic segmentation of phalanx and epiphyseal metaphyseal region[J]. Measurement Science Review,2012,12:128-135.

基于Memcached内存对象缓存技术应用研究

作者：[刘亮](#)，[徐步东](#)，[谭艳艳](#)，[LIU Liang](#)，[XU Bu-dong](#)，[TAN Yan-yan](#)
作者单位：[刘亮, 徐步东, LIU Liang, XU Bu-dong\(山东师范大学 信息技术管理处, 山东 济南, 250014\)](#)
，[谭艳艳, TAN Yan-yan\(山东师范大学 信息科学与工程学院, 山东 济南, 250014\)](#)
刊名：[计算机技术与发展](#)
英文刊名：[Computer Technology and Development](#)
年，卷(期)：2015, 25(11)

引用本文格式：[刘亮. 徐步东. 谭艳艳. LIU Liang. XU Bu-dong. TAN Yan-yan 基于Memcached内存对象缓存技术应用研究\[期刊论文\]-计算机技术与发展 2015\(11\)](#)