

基于 Memcached 的缓存资源集中管理方法

郭 栋¹, 王 伟^{1,2}, 曾国荪^{1,2}

(1. 同济大学 计算机科学与技术系, 上海 200092;

2. 国家高性能计算机工程技术中心同济分中心, 上海 200092)

摘 要: Memcached 是一个免费开源、高性能的、分布式的内存对象缓存系统, 用于在动态 Web 应用中提升访问速度, 在很多高访问量的大型网站中得到广泛应用。然而却一直没有一个对 Memcached 进行统一集中管理部署的工具, 在实际开发中往往会将 Memcached 模块紧密地和应用程序混在一起, 给缓存的独立维护造成困难。文中从 Memcached 应用特征和 Web 应用体系结构特征两方面分析了现有 Memcached 系统的缺点, 进而提出了一种缓存资源集中管理和多应用共享方案, 并构建了一个 Memcached Manager 应用系统。相对于传统 Web 开发方式, 文中提出的方案可以很大程度上规范和简化应用程序对 Memcached 的使用, 方便缓存资源的统一分配管理。

关键词: 分布式缓存; 缓存共享; 缓存管理; Memcached

中图分类号: TP301

文献标识码: A

文章编号: 1673-629X(2013)12-0062-04

doi: 10.3969/j.issn.1673-629X.2013.12.015

Method of Centralized Management of Cache Resources Based on Memcached

GUO Dong¹, WANG Wei^{1,2}, ZENG Guo-sun^{1,2}

(1. Department of Computer Science and Technology, Tongji University, Shanghai 200092, China;

2. Tongji Branch of National Engineering & Technology Center of High Performance, Shanghai 200092, China)

Abstract: Memcached is a free, open source, high-performance, distributed memory object caching system for dynamic Web applications by improving access speed, it is widely used in many high-traffic large websites. However, there isn't a tool to centrally manage and deploy Memcached, in the actual development the Memcached and application procedure always be hybrid, thus causing difficulties to the maintenance of cache. Analyze the shortcomings of the existing Memcached from the Memcached application characteristics and Web application architecture features, then put forward a cache resources centralized management and application sharing programs, and build a Memcached Manager application system. Compared with traditional Web development methods, the proposed program can standardize and simplify the usage of Memcached, and be convenient for cache resource allocation management.

Key words: distributed cache; shared cache; Memcached management; Memcached

0 引 言

随着现代互联网的兴起, 尤其是 Web2.0 网站的兴起, 在系统复杂度和数据量逐步增长的情况下, 整个应用程序的效率和性能开始下降, 而对于服务器来说,

在海量数据和高并发请求的前提下, 硬盘 I/O 瓶颈是限制整体性能的主要因素之一, 与之对应的则是数据库读写。为了解决这个问题, 数据缓存成为了解决此问题的主要途径。其中, 由 Danga Interactive 开发的

收稿日期: 2013-02-27

修回日期: 2013-06-08

网络出版时间: 2013-09-29

基金项目: 国家自然科学基金资助项目(61103068); 教育部博士点基金项目(20110072120017); 浙江大学 CAD&CG 国家重点实验室开放课题(A1311); 南京大学计算机软件新技术国家重点实验室开放课题(KFKT2012B24); 同济大学中央高校基本科研业务费专项资金项目(0800219208); 中科院模式识别国家重点实验室开放课题(201103187)

作者简介: 郭 栋(1991-), 男, 内蒙古呼和浩特人, 研究方向为分布式系统、云计算; 王 伟, 男, 博士, 副教授, 硕士生导师, CCF YOCSEF 上海学术委员学术秘书, CCF 体系结构专委会委员, IBM 360 精英讲师团成员, 研究方向为分布式并行计算、信息安全和智能信息处理; 曾国荪, 博士, 教授, 博士生导师, 2010 年入选上海市优秀学科带头人, 信息领域“863”和国家自然科学基金上会评委专家, IEEE 和中国计算机学会高级会员, 从事并行分布计算、可信网络软件设计与验证、内容信任与安全、物联网云计算等领域的研究工作。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20130929.1522.016.html>

Memcached^[1]是目前应用比较广泛的一个免费开源、高性能的、分布式的内存对象缓存系统^[1]。

目前,很多的大型 Web 应用程序包括 Facebook、YouTube、Wikipedia、Yahoo! 等都在使用 Memcached 来支持它们每天数亿级的页面访问。通过把 cache 层与它们的 Web 架构集成,应用程序在提高了性能的同时,还大大降低了数据库的负载^[2-3]。然而,在 Memcached 很好地提高系统性能的同时,也存在 Memcached 滥用的问题,缺乏一个对 Memcached 进行统一集中管理部署的工具,对 Memcached 的使用与应用程序的编程紧密相关,何时、何地以及如何用 Memcached 完全决定于应用程序开发者,Memcached 无法独立出来进行维护。Memcached 作为一项核心功能组件,是不允许随意更改的,这无疑为后期维护困难埋下了隐患^[4-5]。

针对目前大量使用的缓存系统架构,文中提出了一种 Memcached 集中分配管理和多应用共享方法,旨在将缓存作为一项服务提供,与应用程序在架构上独立开来。通过在 Memcached 分布式服务和应用程序之间建立一个代理服务系统,向下对所有 Memcached 服务器进行管理,向上对应用程序提供统一的编程接口,使得 Memcached 对应用程序开发者完全透明,开发者只需要关心应用是否使用了 Memcached,而不需要关心 Memcached 的具体实现^[6-7]。

1 Memcached 的原理及其问题

1.1 Memcached 的基本原理

Memcached 虽然称为“分布式”缓存服务器,但服务器端并没有“分布式”功能。服务器端仅包括存储功能。至于 Memcached 的分布式,则是完全由客户端程序实现的^[1]。

在一个 Memcached 的查询中,客户端先通过计算 key 的 hash 值来确定数据对所处的服务器位置。当对应的服务器确定后,客户端就会发送一个查询请求给对应的 Memcached 服务器,让它来查找确切的数据。常用的 Memcached 的分布算法一般有两种,元素 key hash 算法和一致性哈希算法。

元素 key hash 算法比较简单,它根据 hash(key)的结果,模连接数的余数决定存储到哪个节点,也就是 $\text{hash}(\text{key}) \% \text{servers.size}()$ 。Consistent Hashing^[8](一致性哈希算法)相对复杂,它首先求出 Memcached 服务器(节点)的哈希值,并将其配置到 $0 \sim 2^{32}$ 的圆上。然后用同样的方法求出存储数据的键的哈希值,并映射到圆上,连接相应的服务器读取或存入数据。

1.2 Memcached 存在的问题

Memcached 以使用方便、协议简单为宗旨^[1],方便

的同时也会带来一些问题。现有的 Memcached 系统主要存在的问题体现在以下几方面。

(1) 缺乏一套 Memcached 的统一集中资源管理系统。开发者如果需要使用 Memcached,必须手动一个一个去部署 Memcached 服务器;

(2) 传统的网站所使用的 Memcached 一般都是单应用服务。无法做到多个不相关的应用程序共享 Memcached;

(3) Memcached 服务往往与应用程序编程紧密相关。开发者需要花费大量时间组织 Memcached 连接方式并维护其运行;

(4) Memcached 没有用户验证功能。

总结起来,造成上述问题的根源在于没有将缓存系统作为一项服务,而是作为应用系统的功能组件。文中即是以缓存作为服务,独立于应用系统为思想,解决上述问题。

2 实现资源集中管理和多应用共享的主要手段

实现多应用共享需要解决并发访问、身份认证、数据冲突三个问题。

2.1 并发访问

作为一项缓存服务,缓存系统会被很多应用程序共享访问,在短时间内会有大量缓存请求,必然要对缓存系统进行并发处理编程,使其能够同时应对多个应用程序的并发请求。一种可行的解决方案是:应用程序和 Memcached Manager 进行通信,Memcached Manager 的服务端在 PHP 的 cli 模式下以后台程序方式运行,使用一种非阻塞式循环监听的仿多线程编程方法,实现并发访问。

2.2 身份认证

Memcached 没有提供身份认证功能,立足于程序的完备性和系统化,对于实现多应用程序共享,出于安全考虑,身份认证是必要的。对于上述问题,一种可行的办法是:Memcached Manager 服务器持久化存储用户的认证信息,应用程序在使用 Memcached 时,Memcached Manager 对用户进行验证,如果是合法用户,则将相关的操作信息发送给相应的 Memcached 服务器,Memcached 服务器处理后返回结果,由 Memcached Manager 封装后统一返回给应用程序^[9]。

2.3 数据冲突

由于整个缓存系统是被应用程序共享的,必须保证不同的用户之间的数据相互独立,避免发生相互覆盖等数据冲突现象^[10]。一种可行的方案是:通过统一控制,为每个用户设置一个唯一的前缀,作为存入到 Memcached 的索引前缀^[11],这样可以保证所有用户的

数据不会发生冲突,达到了多用户共享资源的效果。

有了上述 3 个问题的解决思想后,基本上就可以构建实际的应用系统,下面就以具体应用系统 Memcached Manager 为例剖析实现方法。

3 Memcached Manager 实现方法

此节从具体应用系统角度介绍 Memcached Manager 缓存服务硬件架构和软件设计方案,构成文中思想的具体实现。

3.1 Memcached Manager 缓存服务物理架构

该系统旨在将 Memcached 服务独立于应用系统,所以在架构上也要进行物理分离。一个简单的系统架构如图 1 所示。Node-1 到 Node-n 构成分布式 Memcached 环境,负责提供整个缓存系统的存储基础,它们通过高速 LAN 连接到 Memcached Manager 所在的服务器;所有的应用服务器也连接到 Memcached Manager 服务器,由 Memcached Manager 的服务程序进行统一访问控制和对并发进行处理。

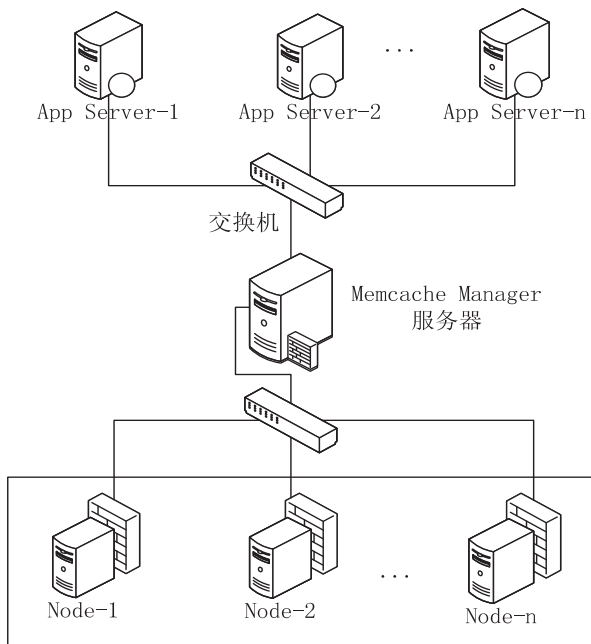


图 1 Memcached Manager 缓存服务物理架构

3.2 Memcached Manager 逻辑架构

Memcached Manager 逻辑系统主要由分布式 Memcached 环境、Parasite 寄宿程序、Memcached Web 管理门户、Memcached Manager 主服务器以及持久化存储系统(如数据库系统)组成。

分布式 Memcached 由多台服务器构成,每台服务器上运行一到多个 Memcached 服务程序,构成分布式缓存基础。

Parasite 寄宿程序运行在缓存服务器上,其主要功能有:管理 Memcached 服务、监控 Memcached 状态,统计资源使用情况以及服务器负载。

Memcached Web 管理门户使系统管理员更方便地管理整个分布式 Memcached 系统,实现信息的可视化操作。

Memcached Manager 主服务器接收来自应用服务器的读写请求,然后将请求转发给相应节点的 Memcached 服务程序,最后将 Memcached 返回的结果返回给应用程序。

4 实验结果和分析

4.1 实验环境和数据

文中的实验环境采用一台主机和多个虚拟机进行搭建。该主机配置 Intel Core2 T6570 2.1 GHz 处理器,2 GB 内存,操作系统为 Microsoft Windows XP 32 位操作系统(Service Pack 3),使用 3 个 VMware 虚拟机作为分布式测试环境。

文中采用对比实验的方法,分别对 Web 应用系统直接使用数据库作为数据层和采用 Memcached Manager 作为数据层的运行效率进行比较。测试数据表结构为(Id, Username, Password)。分别在测试数据表里插入 100、500、1 000、5 000、10 000、50 000 条测试数据。为了使实验更符合实际情况,这里对每次查询循环 100 次,模拟实际应用系统一次请求可能出现的大量查询。程序 1 所示的代码为直接查询数据库的代码片段。

程序 1:直接查询数据库(测试代码)。

```
for( $n=0; $n<100; $n++) {
    perform_mysql_query();
}
```

程序 2 所示的代码为使用了 Memcached Manager 的相应查询。

程序 2:使用 Memcached Manager(测试代码)。

```
for( $n=0; $n<100; $n++) {
    if ( ! $mcm->get('row'. $n) ) { $row = perform_mysql_query();
        $mcm->set('row'. $n, $row[0]);
    }
}
```

4.2 实验结果分析

根据上面的实验配置进行具体实验后,得到的测试结果如图 2 所示。

从实验结果可以得出以下结论以及相应的分析:使用 Memcached Manager 可以有效提高应用程序性能,尤其是在大数据量的情况下,效果更加明显;通过读数据库来获取数据,查询时间会随着数据表的数据量增大而逐渐上升,而通过 Memcached Manager 获取数据所需的时间却基本维持在 60 到 90 毫秒之间;第

一次访问 Memcached Manager 所需要的时间要比直接读数据库长,因为第一次读取需首先读取数据库,然后将数据存入缓存系统,第二次访问则直接从 Memcached 中获取。

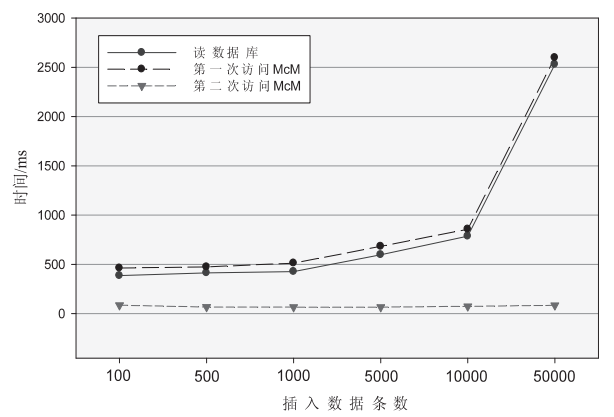


图 2 不同的查询方式所需的时间

经过上边的结果分析,在实际的应用系统中,相对于传统的缓存系统,文中提出的方案同样可以有效提升系统的运行效率,尤其是在大数据量的情况下,其优势更加明显。

5 结束语

文中首先分别从 Memcached 应用特征和 Web 应用体系结构特征两方面分析了现有 Memcached 系统的缺点,从而提出了一种缓存资源集中管理分配方法和多应用共享方案,并介绍相应的 Memcached Manager 应用系统。实验结果表明,相对于传统 Web 开发方式,文中提出的方案能够获得很高的性能加速,同时可以有效提高开发人员的开发效率。

4 结束语

文中基于邻接表存储结构和深度优先搜索遍历,以面向对象的方法构造了域邻接表和搜索算法,给出了详细的运算流程,并且在电子政务跨域数据交换实际场景中实现了该方法,为以稀疏图形式存储的电子政务域的跨域数据交换最短路径查找提供了一条新途径。

参考文献:

[1] 邓磊,吴健,张昌利,等.电子政务中跨域可信数据交换模型设计与实现[J].计算机工程,2007,33(12):4-6.

[2] Cherkassky B V,Goldberg A V,Radzik T. Shortest paths algorithms:theory and experimental evaluation[J]. Mathematical programming,1996,73:129-174.

[3] Lu Feng,Zhou Chenghu,Wan Qin. An optimum path algorithm for traffic network based on hierarchical spatial reasoning[J].

参考文献:

[1] Memcached: High-performance, distributed memory object caching system[EB/OL]. 2011. <http://memcached.org/>

[2] Vaidyanathan K,Narravula S,Balaji P,et al. Designing efficient systems services and primitives for next-generation data centers[C]//Proc of workshop on NSF next generation software (NGS) program. [s. l.]:[s. n.],2007.

[3] Tanenbaum A S,Steen M V. Distributed systems:Principles and paradigms[M]. 2nd ed. [s. l.]:Pearson Prentice-Hall,2006.

[4] Petrovic J. Using Memcached for data distribution in industrial environment[C]//Proceedings of the third international conference on systems (ICONS). [s. l.]:[s. n.],2008:368-372.

[5] Menace D A. Scaling Web sites through caching[J]. IEEE Internet computing,2003,7(4):86-89.

[6] 秦秀磊,张文博,魏峻,等.云计算环境下分布式缓存技术的现状与挑战[J].软件学报,2013,24(1):50-66.

[7] 陈康,郑纬民.云计算:系统实例与研究现状[J].软件学报,2009,20(5):1337-1348.

[8] Karger D,Lehman E,Leighton T,et al. Consistent hashing and random trees: Distributed cachine protocols for relieving hot spots on the World Wide Web[C]//Proc of 29th annu ACM symp on theory of computing. [s. l.]:[s. n.],1997:654-663.

[9] 林海略,韩燕波.多租户应用的性能管理关键问题研究[J].计算机学报,2010,33(10):1881-1895.

[10] 王伟,曾国荪.基于 Bayes 认知信任模型的 MANETs 自聚集算法[J].中国科学 E 辑,2010,40(2):228-239.

[11] 王伟,曾国荪.一种基于 Bayes 信任模型的可信动态级调度算法[J].中国科学 E 辑,2007,37(2):285-296.

(上接第 61 页)

Geo-spatial information science,2000,3(4):36-42.

[4] 陆锋.最短路径算法:分类体系与研究进展[J].测绘学报,2001,30(3):269-275.

[5] 宫恩超,李鲁群.基于 Bellman-Ford 算法的动态最优路径算法设计[J].测绘通报,2011(8):26-28.

[6] 鲍培明.距离寻优中 Dijkstra 算法的优化[J].计算机研究与发展,2001,38(3):307-311.

[7] Höfner P,Möller B. Dijkstra,Floyd and Warshall meet Kleene[J]. Formal aspects of computing,2012,24(4-6):459-476.

[8] 王银燕,余镇危,曹怀虎,等.基于二度量的单播最短路径算法[J].计算机工程,2007,33(5):89-90.

[9] 许国志.系统科学[M].上海:上海科技教育出版社,2000.

[10] 徐振宁,黄凯歌,张维明,等. Ontology 建模方法研究[J]. 计算机科学,2002,29(1):68-71.

[11] 徐孝凯.数据结构[M].北京:电子工业出版社,2004.

[12] 温晓磊.基于混合算法的最短路径优化算法[J].天津理工大学学报,2009,25(1):37-40.

基于Memcached的缓存资源集中管理方法

作者:

郭栋, 王伟, 曾国荪, [GUO Dong](#), [WANG Wei](#), [ZENG Guo-sun](#)

作者单位:

郭栋, [GUO Dong](#) (同济大学 计算机科学与技术系, 上海, 200092), [王伟](#), 曾国荪, [WANG Wei](#), [ZENG Guo-sun](#) (同济大学 计算机科学与技术系, 上海 200092; 国家高性能计算机工程技术中心同济分中心, 上海 200092)

刊名:

计算机技术与发展

ISTIC

英文刊名:

[Computer Technology and Development](#)

年, 卷(期):

2013(12)

本文链接: http://d.wanfangdata.com.cn/Periodical_wjz201312015.aspx