我要提问

# Memcached 分布式缓存实现原理

2016-05-21 分类:操作系统、编程开发、首页精华 ံ 1人评论

分享到: 🚺 🔀 🚮 🎑 👢 😰 🛨 更多 🦳 3

#### 摘要

在高并发环境下,大量的读、写请求涌向数据库,此时磁盘IO将成为瓶颈,从而导致过高的响应延迟, 因此缓存应运而生。无论是单机缓存还是分布式缓存都有其适应场景和优缺点,当今存在的缓存产品也 是数不胜数,最常见的有redis和memcached等,既然是分布式,那么他们是怎么实现分布式的呢?本 文主要介绍分布式缓存服务mencached的分布式实现原理。

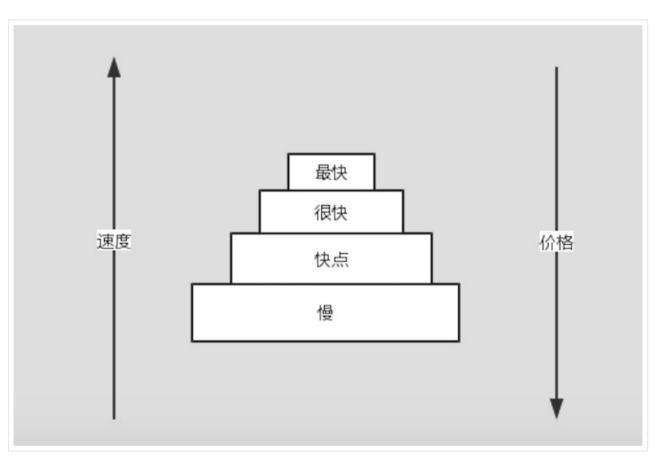
#### 缓存本质

#### 计算机体系缓存

什么是缓存,我们先看看计算机体系结构中的存储体系,根据冯·诺依曼计算机体系结构模型,计算机分 为五大部分:运算器、控制器、存储器、输入设备、输出设备。结合现代计算机, CPU包含运算器和控 制器两个部分,CPU负责计算,其需要的数据由存储提供,存储分为几个级别,就拿我当前的PC举个例 子,我的机器存储清单如下:

- 1. 356G的磁盘
- 2. 4G的内存
- 3. 3MB三级缓存
- 4. 256KB二级缓存(pre core)

除了上述部分,还有CPU内的寄存器,当然有的计算机还有一级缓存等。CPU运算器工作的时候需要数 据,数据哪里来?首先从距离CPU最近的二级缓存去拿,这块缓存速度最快,通常也是体积最小,因为 价格最贵:



## 存储金字塔

如上图所示,存储体系就像个金子塔,最上层最快,价格最贵,最下层最慢,价格也最便宜,CPU的数 据源优先级一层层从上到下去寻找数据。

很显然,除了最慢的那块存储,在计算机体系中,相对较快的那些存储都可以被称为缓存,他们解决的 问题是**让存储访问更快。** 

计算机体系存储系统模型扩展到应用也是一样,应用需要数据,数据哪里来?缓存(更快的存储)->DB

(较慢的存储),他们的工作流程大致如下图所示:

### 缓存应用系统 职场人生

## 编程不需要天赋和激情

热门文章

编程不需要天赋和激情

AngularJS 样式指南介绍

游戏编程十年总结

程序员眼中的古典名画

记一次 Google 面试经历

对优秀程序员的思考

C# Lambda表达式的前世今生

Java 远程通讯技术及原理分析

22 个 Android Studio 优秀插件汇总

80 多个 Linux 系统管理员的监控工具

入门软件工程师所面临的5个挑战

程序员编程的 7 + 1 条小贴士

假如程序员生活在童话里...

一个32岁入门的70后程序员给我的启示

程序员每天都在使用的6个惊讶的软技能

一流程序员完全可以有编程之外的生活

回顾15年程序员生涯, 我总结的7点经验

为什么开源可以提高程序员的编程技能?

程序员的走与留?

## 相关文章

Memcache 内存分配策略和性能(使用)状态...

MemCache超详细解读

使用Memcached改进Java企业级应用性能...

热门栏目订阅

Java RSS 微博

程序员 RSS 微博

Android RSS

RSS

Linux RSS

JavaScript

php PHP



付费投稿计划 点击查看详情

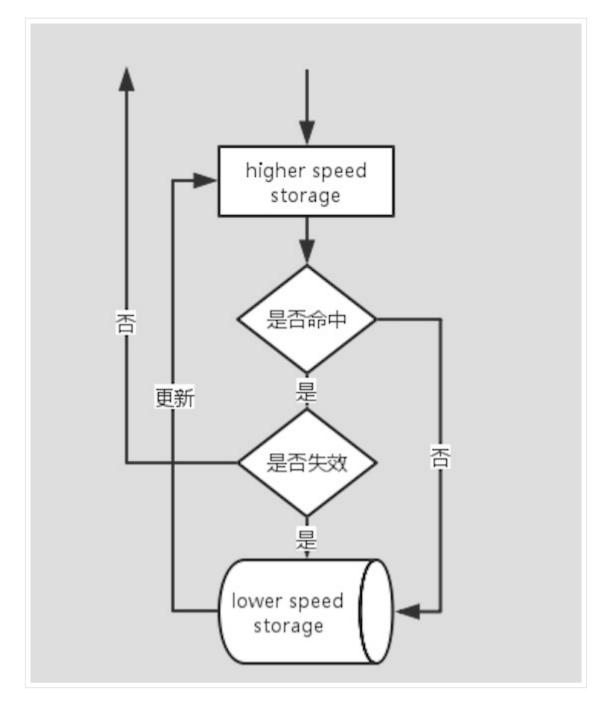
C#客户端Redis服务器的分布式缓存

5个强大的Java分布式缓存框架推荐

Java分布式键-值缓存系统Voldemort

Cacheonix: Java分布式集群缓存框架

Hadoop DistributedCache分布式缓存的使..



#### 带缓存的存储访问一般模型

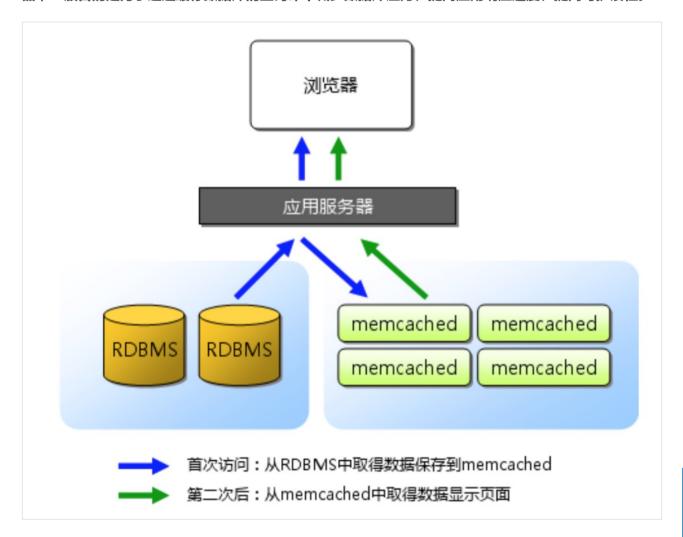
如上图所示,缓存应用系统一般存储访问流程:首先访问缓存较快的存储介质,如果命中且未失效则返回内容,如果未命中或失效则访问较慢的存储介质将内容返回同时更新缓存。

## memcached简介

#### 什么是memcached

memcached是LiveJournal旗下的Danga Interactive公司的Brad Fitzpatric为首开发的一款软件。现在已经成为mixi、hatena、Facebook、Vox、LiveJournal等众多服务中提高Web应用扩展性的重要因素。传统的Web应用都将数据保存到RDBMS中,应用服务器从RDBMS中读取数据、处理数据并在浏览器中显示。但是随着数据量增大、访问的集中、就会出现RDBMS的负担加重、数据库响应变慢、导致整个系统响应延迟增加。

而memcached就是为了解决这个问题而出现的,**memcached是一款高性能的分布式内存缓存服务** 器,一般目的是为了通过缓存数据库的查询命中减少数据库压力、提高应用响应速度、提高可扩展性。



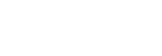
## memcached缓存应用

## memcached缓存特点

- 1. 协议简单
- 2. 基于libevent的事件处理
- 3. 内置内存存储方式
- 4. memcached不相互通信的分布式

## memcached分布式原理

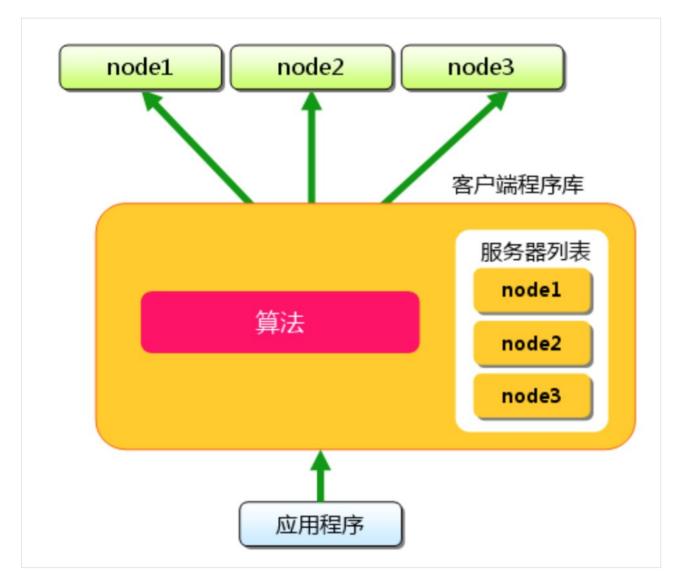
今天的内容主要涉及memcached特点的第四条,memcached不相互通信,那么memcached是如何实现分布式的呢?memcached的分布式实现主要依赖客户端的实现:







付费投稿计划点击查看详情



#### memcached分布式

如上图所示,我们看下缓存的存储的一般流程:

当数据到达客户端,客户端实现的算法就会根据"键"来决定保存的memcached服务器,服务器选定后,命令他保存数据。取的时候也一样,客户端根据"键"选择服务器,使用保存时候的相同算法就能保证选中和存的时候相同的服务器。

#### 余数计算分散法

余数计算分散法是memcached标准的memcached分布式方法,算法如下:

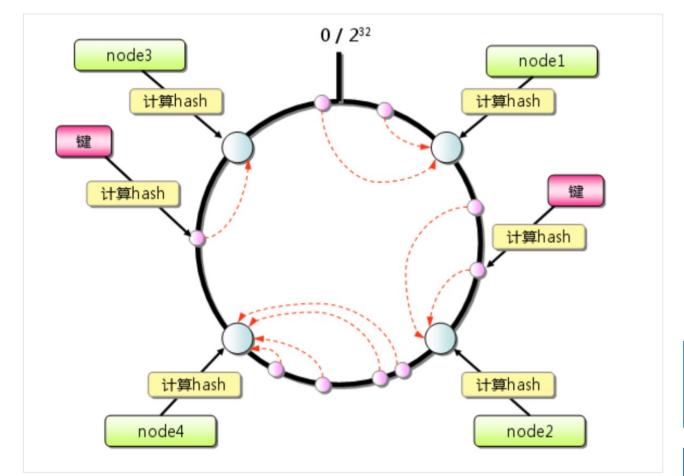
CRC(\$key)%N

该算法下,客户端首先根据key来计算CRC,然后结果对服务器数进行取模得到memcached服务器节点,对于这种方式有两个问题值得说明一下:

- 1. 当选择到的服务器无法连接的时候,一种解决办法是将尝试的连接次数加到key后面,然后重新进行hash,这种做法也叫rehash。
- 2. 第二个问题也是这种方法的致命的缺点,尽管余数计算分散发相当简单,数据分散也很优秀,当添加或者移除服务器的时候,缓存重组的代价相当大。

## Consistent Hashing算法

Consistent Hashing算法描述如下:首先求出memcached服务器节点的哈希值,并将其分配到0~2^3 2的圆上,这个圆我们可以把它叫做值域,然后用同样的方法求出存储数据键的哈希值,并映射到圆上。然后从数据映射到的位置开始顺时针查找,将数据保存到找到的第一个服务器上,如果超过0~2^32仍找不到,就会保存在第一台memcached服务器上:



## memcachd基本原理

再抛出上面的问题,如果新添加或移除一台机器,在consistent Hashing算法下会有什么影响。上图中假设有四个节点,我们再添加一个节点叫node5:

## 关注我们的微博



付费投稿计划 点击查看详情

#### 添加了node节点之后

node5被放在了node4与node2之间,本来映射到node2和node4之间的区域都会找到node4,当有node5的时候,node5和node4之间的还是找到node4,而node5和node2之间的此时会找到node5,因此当添加一台服务器的时候受影响的仅仅是node5和node2区间。

#### 优化的Consistent Hashing算法

上面可以看出使用consistent Hashing最大限度的抑制了键的重新分配,且有的consistent Hashing的实现方式还采用了虚拟节点的思想。问题起源于使用一般hash函数的话,服务器的映射地点的分布非常不均匀,从而导致数据库访问倾斜,大量的key被映射到同一台服务器上。为了避免这个问题,引入了虚拟节点的机制,为每台服务器计算出多个hash值,每个值对应环上的一个节点位置,这种节点叫虚拟节点。而key的映射方式不变,就是多了层从虚拟节点再映射到物理机的过程。这种优化下尽管物理机很少的情况下,只要虚拟节点足够多,也能够使用得key分布的相对均匀。

### 总结

本文介在理解缓存基本概念的情况下介绍了memcached的分布式算法实现原理, memcached的分布式是由客户端函数库实现的。

