# 1.memcache应用说明

### 初识MemCache

Memcache是一个高性能的分布式的内存对象缓存系统，通过在内存里维护一个统一的巨大的hash表，它能够用来存储各种格式的数据，包括图像、视频、文件以及数据库检索的结果等。简单的说就是将数据调用到内存中，然后从内存中读取，从而大大提高读取速度。

Memcache是danga的一个项目，最早是LiveJour  MemCachenal 服务的，最初为了加速 LiveJournal 访问速度而开发的，后来被很多大型的网站采用。

Memcached是以守护程序方式运行于一个或多个服务器中，随时会接收客户端的连接和操作。

### MemCache在Web中的应用

MemCache缓存系统最主要的就是为了提高动态网页应用，分担数据库检索的压力。对于网站流量比较大的，可以使用memcache缓解数据库的压力，主要的焦点集中在以下两个方面：

1. 使用MemCache作为中间缓存层减少数据库的压力。

2. MemCache分布式的应用。

# 2.Memcache的安装

### Liunx下安装MemCached软件

安装memcache源代码

http://memcached.googlecode.com/files/memcached-1.4.10.tar.gz

1.首先安装依赖包libevent

yum –y install libevent \*

2. 主包已经安装,别忘记安装libevent-devel\*,不然./configure过不去

tar xzf /lamp/memcached-1.4.10.tar.gz 解压memcached

cd /lamp/memcached-1.4.10 进入 memcached目录

./configure --prefix=/usr/local/memcache 配置

make && make install 编译与安装

useradd memcache 添加memcache用户

3.因为系统不能用root运行memcache软件

/usr/local/memcache/bin/memcached –umemcache & #后台运行

netstat –tunpl|grep :11211 查看端口

telnet 192.168.10.1 11211 连接测试

stats memcache命令：查看当前状态

4.写入自启动：

vi /etc/rc.local

/usr/local/memcache/bin/memcached –umemcache &

5.如何杀掉后台进程

pkill memcached

### Windows下安装memcached软件

1.1 在网上下载memcached-1.2.1-win32.zip。解压放某个盘下面，

比如在c:\memcached

1.2 在终端（cmd）下：

D:\AppServ>memcached.exe -d install 安装

D:\AppServ>memcached.exe -d uninstall 卸载

D:\AppServ>memcached.exe -d start 启动

D:\AppServ>memcached.exe -d stop 停止

memcached.exe -h 获取所有帮助

1.3 在启动之后连接：

D:\AppServ> telnet 127.0.0.1 11211 --连接memcache端口11211

使用quit退出。

Window7的telnet功能是默认关闭的。

可以输入stats看下状态。

### memcached服务器的管理

启动Memcache 常用参数

-p <num> 设置端口号(默认不设置为: 11211)

-U <num> UDP监听端口(默认: 11211, 0 时关闭)

-l <ip\_addr> 绑定地址(默认:所有都允许,无论内外网或者本机更换IP，有安全隐患，若设置为127.0.0.1就只能本机访问)

-d 独立进程运行

... -d start 启动memcached服务

... -d restart 重起memcached服务

... -d stop|shutdown 关闭正在运行的memcached服务

... -d install 安装memcached服务

... -d uninstall 卸载memcached服务

-u <username> 绑定使用指定用于运行进程<username> 用什么用户名的意思

-m <num> 允许最大内存用量，单位M (默认: 64 MB)

-P <file> 将PID写入文件<file>，可以使得后边进行快速进程终止, 需要与-d 一起使用

-M 内存耗尽时返回错误，而不是删除项

-c 最大同时连接数，默认是1024

-f 块大小增长因子，默认是1.25

-n 最小分配空间，key+value+flags默认是48

-h 显示帮助

常用组合：  
#/usr/local/memcached/bin/memcached -d -m 256 -p 11211 -u www 127.0.0.1  
#/usr/local/memcached/bin/memcached -d -m 256 -p 11211 -u www -l 127.0.0.1

#/usr/local/memcached/bin/memcached -d -m 1024 -p 11211 -u root -c 1000

这个是ppt里面的隐藏部分

### Stats内容

stats 命令的功能正如其名：转储所连接的 memcached 实例的当前统计数据。在下例中，执行 stats 命令显示了关于当前 memcached 实例的信息：

STAT pid 22459                             进程ID   
STAT uptime 1027046                        服务器运行秒数   
STAT time 1273043062                       服务器当前unix时间戳   
STAT version 1.4.4                         服务器版本   
STAT pointer\_size 64                       操作系统字大小(这台服务器是64位的)   
STAT rusage\_user 0.040000                  进程累计用户时间   
STAT rusage\_system 0.260000                进程累计系统时间   
STAT curr\_connections 10                   当前打开连接数   
STAT total\_connections 82                  曾打开的连接总数   
STAT connection\_structures 13              服务器分配的连接结构数   
STAT cmd\_get 54                            执行get命令总数   
STAT cmd\_set 34                            执行set命令总数   
STAT cmd\_flush 3                           指向flush\_all命令总数   
STAT get\_hits 9                            get命中次数   
STAT get\_misses 45                         get未命中次数   
STAT delete\_misses 5                       delete未命中次数   
STAT delete\_hits 1                         delete命中次数   
STAT incr\_misses 0                         incr未命中次数   
STAT incr\_hits 0                           incr命中次数   
STAT decr\_misses 0                         decr未命中次数   
STAT decr\_hits 0                           decr命中次数   
STAT cas\_misses 0    cas未命中次数   
STAT cas\_hits 0                            cas命中次数   
STAT cas\_badval 0                          使用擦拭次数   
STAT auth\_cmds 0   
STAT auth\_errors 0   
STAT bytes\_read 15785                      读取字节总数   
STAT bytes\_written 15222                   写入字节总数   
STAT limit\_maxbytes 1048576                分配的内存数（字节）   
STAT accepting\_conns 1                     目前接受的链接数   
STAT listen\_disabled\_num 0                   
STAT threads 4                             线程数   
STAT conn\_yields 0   
STAT bytes 0                               存储item字节数   
STAT curr\_items 0                          item个数   
STAT total\_items 34                        item总数   
STAT evictions 0                           为获取空间删除item的总数

# 3.Memcache的常用命令操作

### 基本的memecached客户端命令

5个常用的命令

stats: 当前所有memcached服务器运行的状态信息

add: 添加一个数据到服务器

set: 替换一个已经存在的数据，如果数据不存在，则和add命令相同。

get: 从服务器端提取指定的数据。

delete: 删除指定的单个数据，如果要清除所有数据，可以使用flush\_all指令。

Memcache的协议的错误部分主要是三个错误提示之提示指令：

ERROR -- 普通错误信息，比如指令错误之类的

CLIENT\_ERROR <错误信息> -- 客户端错误

SERVER\_ERROR <错误信息> --服务器端错误

### 数据管理命令

格式：<命令> <键> <标记> <有效期> <数据长度>

其中：

命令：add(添加)、set(修改)、delete(删除)、get(获取)

<键> -key，是发送过来指令的key内容

<标记> - flags，是调用set指令保存数据时候的flags标记

有效期：是数据在服务器上的有效期限，如果是0，则数据永远有效，单位是秒

数据的长度，block data 块数据的长度，一般在这个个长度结束以后下一行跟着block data数据内容，

发送完数据以后，客户端一般等待服务器端的返回，服务器端的返回：

STORED 数据保存成功

NOT\_STORED 数据保存失败，是因为服务器端这个数据key已经存在

执行stats items命令，可以看到出现 很多的items行。   
  
执行stats cachedump 3 0命令。这里的3表示上面图中items后面的数字，0标示显示全部的数据，如果是1就标示只显示1条。

# 4.PHP的memcache功能扩展模块安装

### 安装PHP中的MemCache应用程序扩展接口

Liunx系统下：

tar  zxvf memcache-2.2.5.tgz 解压软件包

cd  memcache-2.2.5 进入目录

/usr/local/php/bin/phpize 生成配置环境

./configure --with-php-config=/usr/local/php/bin/php-config 配置

make  &&  make  install 编译和安装

修改php.ini

extension\_dir = "/usr/local/php//lib/php/extensions/no-debug-non-zts-20060613/“

extension="memcache.so";

对于内存缓存，比较常用的有两种memcache和memcached扩展。而memcached和memcache的守护进程memcached同名， 比较容易引起混淆，甚至提到memcached，有些人第一想到的是后台的守护进程，这里还是有必要分析一下两者之间的区别

memcache是完全在PHP框架内开发的

memecached是使用libmemcached的

从手册上看，memcached 会比 memcache 多几个方法，使用方式上都差不多。

memcache是原生实现的，但是使用libmemcached的memached只支持OO接口，而 memcache则是OO和非OO两套接口并存,以后随着memcached服务器端的改进，这个lib也必定会马上跟进的。

而memcache却不一定 能做到按时跟进。

memcached，还有个非常称赞的地方，就是flag不是在操作的时候设置了。而是有一个统一的setOption()。memcached 实现了更多的 memcached 协议（毕竟是基于 libmemcached 库的）。

# 5.在PHP中操作memcache

### MemCache应用程序扩展接口

Memcache — Memcache类

Memcache::add — 增加一个条目到缓存服务器

Memcache::addServer — 向连接池中添加一个memcache服务器

Memcache::close — 关闭memcache连接

Memcache::connect — 打开一个memcached服务端连接

Memcache::decrement — 减小元素的值

Memcache::delete — 从服务端删除一个元素

Memcache::flush — 清洗（删除）已经存储的所有的元素

Memcache::get — 从服务端检回一个元素

Memcache::getExtendedStats — 缓存服务器池中所有服务器统计信息

Memcache::getServerStatus — 获取一个服务器的在线/离线状态

Memcache::getStats — 获取服务器统计信息

Memcache::getVersion — 返回服务器版本信息

Memcache::increment — 增加一个元素的值

Memcache::pconnect — 打开一个到服务器的持久化连接

Memcache::replace — 替换已经存在的元素的值

Memcache::set — Store data at the server

Memcache::setCompressThreshold — 开启大值自动压缩

Memcache::setServerParams — 运行时修改服务器参数和状态

Memcache 函数memcache\_debug — 转换调试输出的开/关

### MemCache的实例应用

//1. 创建对象

$mem = new Memcache();

//2. 添加服务

$mem->addServer("192.168.150.250",11211);

$mem->addServer("192.168.150.138",11211);

$mem->addServer("192.168.112.128",11211);

//3. 放置信息

$mem->add("mystr","hello memcache!",MEMCACHE\_COMPRESSED,0);

$mem->add("myarray",array(10,20,30,40),MEMCACHE\_COMPRESSED,0);

$mem->add("myob",new Stu(),MEMCACHE\_COMPRESSED,0);

//4. 获取信息

echo $mem->get("mystr")."<br/>";

var\_dump($mem->get('myarray'));

echo "<br/>";

$mem->get("myob")->getinfo();

# 6.在PHP中设置分布式memcache存取

//1. 创建对象

$mem = new Memcache();

//2. 添加服务

$mem->addServer("192.168.150.250",11211);

$mem->addServer("192.168.150.138",11211);

$mem->addServer("192.168.112.128",11211);

//3. 放置信息

$mem->add("mystr","hello memcache!",MEMCACHE\_COMPRESSED,0);

$mem->add("myarray",array(10,20,30,40),MEMCACHE\_COMPRESSED,0);

$mem->add("myob",new Stu(),MEMCACHE\_COMPRESSED,0);

//4. 获取信息

echo $mem->get("mystr")."<br/>";

var\_dump($mem->get('myarray'));

echo "<br/>";

$mem->get("myob")->getinfo();

### memcached服务器的安全防护

1. 内网访问

memcached -d -m 1024 -u root –l 192.168.0.10 –p 11211 –c 1024 start

2. 设置防火墙

iptables –F

iptables –P INPUT DROP

iptables –A INPUT –p tcp –s 192.168.0.10 –-dport 11211 –j ACCEPT

iptables –A INPUT –p udp –s 192.168.0.10 –-dport 11211 –j ACCEPT