

# 第19章 MemCache 管理与应用

细说PHP第4版

# 本章目录

19.1 MemCache概述

19.2 memcached的安装及管理

19.3 使用Telnet作为memcached的客户端管理

19.4 PHP的memcached管理接口

19.5 memcached服务器的安全防护

# 19.1 MemCache概述

19.1.1 初识MemCache

19.1.2 MemCache在Web中的应用

## 19.1.1 初识MemCache

MemCache是什么？

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。

## 19.1.2 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。

## 19.2 memcached的安装及管理

19.2.1 Linux下安装MemCache软件

19.2.2 Linux下安装MemCache扩展

19.2.3 Windows下安装MemCache软件

19.2.4 memcached服务器的管理

## 19.2.1 Linux下安装MemCache软件

1.准备二进制包, 这个包在ISO镜像文件里面

```
memcached-1.4.4-3.el6.i686.rpm
```

2.首先挂载ISO镜像文件 DVD1.ISO

```
mount /dev/sr0 /mnt/cdrom/
```

3.安装memcached

```
yum -y install memcached
```

4.添加memcache用户

```
useradd memcache
```

## 19.2.1 Linux下安装MemCache软件

### 5.启动memcache服务

```
memcached -d -m 128 -l 127.0.0.1 -p 11211 -u memcache
```

查看memcache服务进程

```
netstat -tunpl|grep :11211
```

### 6.设置开机自启动

```
chkconfig memcached on
```

### 7.如何杀掉后台进程

```
pkill memcached
```



## 19.2.2 Linux下安装MemCache扩展

首先下载php7.0支持的扩展包 `pecl-memcache-php7.zip`

地址:<https://github.com/websupport-sk/pecl-memcache/archive/php7.zip>

1. 将`pecl-memcache-php7.zip`文件传到服务器上 `/root`

2. 解压文件后进入此目录

```
unzip pecl-memcache-php7.zip
```

```
cd pecl-memcache-php7
```

3. 由于目录中没有`configure`配置环境文件

执行 `/usr/local/php/bin/phpize` 来准备一个编译环境

4. 配置：加配置

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

5. `make && make install` 编译和安装

## 19.2.2 Linux下安装MemCache扩展

6. 安装成功后就会产生一个memcache.so文件

```
/usr/local/php/lib/php/extensions/no-debug-non-zts-20151012/
```

7. 修改php配置文件

```
vim /usr/local/php/etc/php.ini
```

添加以下信息

```
722行 extension_dir='/usr/local/php/lib/php/extensions/no-debug-non-zts-  
20151012/';  
extension="memcache.so";
```

8. 重启apache服务即可(查看phpinfo())

## 19.2.3 Windows下安装MemCache软件

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

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

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。

## 19.2.3 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

## 19.2.3 memcached服务器的管理

- u <username> 绑定使用指定用于运行进程<username>
- 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

## 19.3 使用Telnet作为memcached的客户端管理

19.3.1 连接memcached服务器

19.3.2 基本的memcached客户端命令

19.3.3 查看当前memcached服务器的运行状态信息

19.3.4 数据管理指令

## 19.3.1 连接memcached服务器

连接命令

```
telnet 127.0.0.1 11211 --连接memcache端口11211
```

注意:

如果输入telnet命令的时候 出现telnet不是内部命令,修改操作如下:

windows系统->控制面板->程序和功能->打开和关闭Windows功能->勾选 telnet客户

端

使用quit退出。

## 19.3.2 基本的memcached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



## 19.3.3 查看当前memcached服务器的运行状态信息

Stats这里显示了很多状态信息，下边详细解释每个状态项：

STAT pid 1552 服务进程的进程ID

STAT uptime 3792 服务从启动到当前所经过的时间，单位是秒。

STAT time 1262517674 服务器所在主机当前系统的时间，单位是秒。

STAT version 1.2.6 组件的版本。这里是我当前使用的1.2.6。

STAT pointer\_size 32 服务器所在主机操作系统的指针大小，一般为32或64。

STAT curr\_items 1 表示存放当前的所有缓存对象的数量。不包括已经从缓存中删除的对象。

STAT total\_items 2 表示从启动到当前，系统存储过的所有对象数量，包括已经删除的对象。

STAT bytes 593 表示系统存储缓存对象所使用的存储空间，单位为字节。

STAT curr\_connections 2 表示当前系统打开的连接数。

STAT total\_connections 28 表示从memcached服务启动到当前时间，系统打开过的连接的总数。

### 19.3.3 查看当前memcached服务器的运行状态信息

STAT connection\_structures 9 表示从memcached服务启动到当前时间，被服务器分配的连接结构的数量，这个解释是协议文档给的，具体什么意思，我目前还没搞明白。

STAT cmd\_get 3 累积获取数据的数量，这里是3，因为我测试过3次，第一次因为没有序列化对象，所以获取数据失败，是null，后边有2次是我用不同对象测试了2次。

STAT cmd\_set 2 累积保存数据的树立数量，这里是2.虽然我存储了3次，但是第一次因为没有序列化，所以没有保存到缓存，也就没有记录。

STAT get\_hits 2 表示获取数据成功的次数。

STAT get\_misses 1 表示获取数据失败的次数。

STAT evictions 0 为了给新的数据项目释放空间，从缓存移除的缓存对象的数目。比如超过缓存大小时根据LRU算法移除的对象，以及过期的对象。

### 19.3.3 查看当前memcached服务器的运行状态信息

STAT bytes\_read 1284 memcached服务器从网络读取的总的字节数。

STAT bytes\_written 5362 memcached服务器发送到网络的总的字节数。

STAT limit\_maxbytes 67108864 memcached服务缓存允许使用的最大字节数。这里为67108864字节，也就是64M.与我们启动memcached服务设置的大小一致。

STAT threads 1 被请求的工作线程的总数量。

## 19.3.4 数据管理指令

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

## 19.4 PHP的memcached管理接口

19.4.1 安装PHP中的MemCache应用程序扩展接口

19.4.2 MemCache应用程序扩展接口

19.4.3 MemCache的实例应用

## 19.4.1 安装PHP中的MemCache应用程序扩展接口

Linux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";
```

## 19.4.2 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

## 19.4.2 MemCache应用程序扩展接口

[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



## 19.4.3 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("myobj",new Stu(),MEMCACHE_COMPRESSED,0);
```

### //4. 获取信息

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

```
var_dump($mem->get('myarray'));
```

```
echo "<br/>";
```

```
$mem->get("myobj")->getinfo();
```

## 19.5 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
```



IT兄弟连  
ITXDL.CN

变态严管 让学习成为一种习惯