Memcache 技术笔记

一、基本概念

1. 缘起:

在数据驱动的 web 开发中，经常要重复从数据库中取出相同的数据，这种重复极大的增加了数据库负载。缓存是解决这个问题的好办法。但是 Web 中的虽然已经可以实现对页面局部进行缓存，但还是不够灵活。此时 Memcached 或许是你想要的。

2. Memcached 是什么?

Memcached 是由 Danga Interactive 开发的，高性能的，分布式的内存对象缓存系统，用于在动态应用中减少数据库负载，提升访问速度。

3. Memcached 能缓存什么？

通过在内存里维护一个统一的巨大的 hash 表，Memcached 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。

4. Memcached 快么？

非常快。Memcached 使用了 libevent（如果可以的话，在 linux 下使用 epoll）来均衡任何数量的打开链接，使用非阻塞的网络 I/O，对内部对象实现引用计数(因此，针对多样的客户端，对象可以处在多样的状态)，使用自己的页块分配器和哈希表，因此虚拟内存不会产生碎片并且虚拟内存分配的时间复杂度可以保证为 O(1)。

Danga Interactive 为提升 Danga Interactive 的速度研发了 Memcached。

5. Memcached 的特点

Memcached 的缓存是一种分布式的，可以让不同主机上的多个用户同时访问，因此解决了共享内存只能单机应用的局限，更不会出现使用数据库做类似事情的时候，磁盘开销和阻塞的发生。

缓存数据:

1.在内存中缓存数据　　2.数据形态以 key->value 结构　　3.安全度非常差的

软件:

1.memcache 模块

让 php 支持 memcache 的函数，可以去连接 memcache 服务，进行增、删、改、查

2.memcached　提供 memcache 服务,11211

学习 memcache:

1.memcache 服务器操作

2.用 php 去操作 memcache

二、安装：

1. Windows 下操作：

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。

1.4 其他命令参数：

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>

-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

2. Linux 下安装操作：

2.1 #安装 memcache 源代码

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

a. **首先安装依赖包 libevent**

Yum –y install libevent*

b. **#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

c. **#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

d. **写入自启动：**

vi /etc/rc.local

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

e. **#如何杀掉后台进程**

pkill memcached

2.2 编译安装 memcache

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";
```

1.安装 memcache

2.连接 memcache 服务器，并进行增、删、改、查操作

3.用 php 连接 memcache 服务器，并进行增、删、改、查操作

三、memcached 的命令

连接　telnet 127.0.0.1 11211

1.【错误指令】

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

2.【数据操作命令】

格式：<命令> <键> <标记> <有效期> <数据长度>

其中：

<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

2.1 添加数据

格式：add 变量名 标记位　时间　长度

例:　add　one 1 100000 10

1234567890

3. 获取 get 变量名

get one

get <键>*

<键>* - key

key 是是一个不为空的字符串组合，发送这个指令以后，等待服务器的返回。如果服务器端没有任何数据，则是返回：

END

4. 修改　set|replace　变量名 标记位　时间（秒）　长度

5. delete <键> <超时时间>

<键> - key，希望在服务器上删除数据的 key 键

<超时时间> - timeout

按照秒为单位，这个是个可选项，如果没有指定这个值，那么服务器上 key 数据将马上被删除，

如果设置了这个值，那么数据将在超时时间后把数据清除，该项缺省值是 0，就是马上被删除

删除数据后，服务器端会返回：

DELETED　删除数据成功

NOT_FOUND　这个 key 没有在服务器上找到.

6. flush_all 这个指令执行后，服务器上所有缓存的数据都被删除，并且返回：OK

7. stats items 看选项号

stats cachedump 选项号　(0|n)

只能看到变量， 值使用 get 获取

如果查看 memcache 已经启动:

1.服务列表 services.msc

2.查看进程 tasklist|find “11211”

3.查看端口 netstat –ano|find “11211”

netstat -ano |find "11211"　查看端口

tasklist　　进程树

启动参数:

memcached.exe -d -m 2048 -u root -l 192.168.1.20 -p 12111 -c 1024

-m 设置内存

-u 登录用户

-l 监听网卡

-p 监听端口

-c 并发用户

memcache 服务操作:

存数据

set my1 1 3600 4

aaa

//过期时间为 0 永久不过期,存储 4 个字节数据，第二个参数是个标记,设置为 0,不为 0，php 页面取不到从 cmd 中插入的数据

取数据

get my1

增:

1.add

2.set

删:

1.delete my1

2.flush_all

改:

1.set

2.replace

查:

1.get

2.stats

3.stats items

4.stats cachedump 1 0

5.stats sizes

php 操作 memcache:

连接 memcache 服务:

$mem=new Memcache;

$mem->connect("localhost","11211");

增:

$mem->add($key,$value,是否压缩,过期时间);

$mem->set($key,$value,是否压缩,过期时间);

删:

$mem->delete($key);

$mem->flush();

改:

curr_items 1

$mem->set($key,$value,是否压缩,过期时间);

查:

$mem->get($key);

$mem->getStats();

$mem->getVersion();

参考:

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。

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。

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。

END