

后盾网 人人做后盾

www.houdunwang.com

高并发网站解决
MemCache

后盾网 2011-2016

- Memcache是一个高性能的分布式的内存对象缓存系统。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。
- 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主要用于关系型数据库的缓存，只是说主要并没有说全部，所以盾友不要误解，比如说可以用于session储存处理
- Memcache是将数据储存到内存中，比关系型数据库要快的多，因为内存的访问速度是硬盘的10~100万倍
- Memcache将数据储存到内存中所以当服务器重起时数据全部丢失

MemCache介绍

安装

- `yum -y install memcached.i686`

启动

- `memcached -m 128m -p 11211 -d -u root -P /tmp/memcached.pid -c 256`

启动参数

- `-p` tcp端口号 默认11211
- `-m` 最大内存大小 默认64M
- `-d` 做为守护进程在后台运行
- `-c` 最大并发连接数
- `-P` 保存Memcache的pid文件
- `-l` 监听服务器地址
- `-u` 如果以root用户启动MemCache必须加此选项

通过`netstat -anpl | grep 11211`查看

MemCache安装与启动


```
kill -9 `cat /tmp/memcached.pid`
```

或者

```
kill 进程号
```

关闭MemCache

安装PHP的MemCache扩展

- `yum install -y php-pecl-memcache.i686`

开启PHP中的MemCache支持

1. `$memcache = new Memcache;`
2. `$memcache->connect('127.0.0.1',11211);`
3. `$memcache->set("test","houdunwang.com",0,20);`
4. `echo $memcache->get("test");`
5. `$memcache->close();`

在PHP中操作MemCache缓存

打开memcached服务端连接

- `bool Memcache::connect (string $host [, int $port [, int $timeout]])`
- **Host** 主机地址
- **Port** memcached服务端监听端口
- **Timeout** 连接持续（超时）时间，单位秒。默认值1秒

示例

- `$memcache = new Memcache;`
- `$memcache->connect('192.168.1.21', 11211);`

连接Memcached

```
bool Memcache::add ( string $key , mixed $var [, int $flag [,  
int $expire ]] )
```

Key	变量名
var	将要被存储的变量
Flag	是否压缩
Expire	失效时间。0表明永不过期

```
$memcache_obj->add('web', 'houdunwang', false, 30);
```

增加一个条目到缓存服务器

```
bool Memcache::replace ( string $key , mixed $var  
    [, int $flag [, int $expire ]] )
```

- 通过key来查找元素并替换其值。当key 对应的元素不存在时，Memcache::replace()返回FALSE。

```
$memcache_obj-  
    >replace('web', 'houdunwang.com', false, 50);
```

替换已经存在的元素的值

```
bool Memcache::set ( string $key , mixed $var [, int $flag [, int  
$expire ]] )
```

```
$memcache_obj->set('web', 'houdunwang.com', false, 50);
```

添加或替换一个key

- `string Memcache::get (string $key [, int &$flags])`
- `array Memcache::get (array $keys [, array &$flags])`

1. `$mem->add("a","aa");`
2. `$mem->add("b","bb");`
3. `print_r($mem->get(array("a","b")));`

从服务端检回多个元素

```
bool Memcache::delete ( string $key [, int $timeout ] )
```

- **Key** 要删除的元素的key。
- **Timeout** 删除该元素的执行时间。如果值为0,则该元素立即删除, 如果值为30,元素会在30秒内被删除。
- `$memcache_obj->delete('webname', 10);`

从服务端删除一个元素

bool Memcache::flush (void)

- 1. \$memcache_obj = new Memcache;**
- 2. \$memcache_obj->connect('memcache_host', 11211);**
- 3. \$memcache_obj->flush();**

删除已经存储的所有元素

```
int Memcache::decrement ( string $key [, int $value = 1 ] )
```

- key元素
- value值减小多少。

```
$mem->decrement("a",10);
```

注：

新的元素的值不会小于0。

不要将Memcache::decrement()方法用于压缩存储的元素，那样作会导致Memcache::get()方法获取值会失败。

减小元素的值

```
int Memcache::increment ( string $key [, int $value = 1 ] )
```

Memcache::increment()将指定元素的值增加value。如果指定的key 对应的元素不是数值类型并且不能被转换为数值， 会将此值修改为value. **Memcache::increment()** 不会在key对应元素不存在时创建元素

具体使用参考上一页decrement方法的使用

```
$mem->increment("a",2);
```

增加一个元素的值

bool Memcache::close (void)

Memcache::close()关闭到memcached服务端的连接。这个函数不会关闭持久化连接，持久化连接仅仅会在web服务器关机/重启时关闭。

1. **\$memcache_obj = new Memcache;**
2. **\$memcache_obj->connect('memcache_host', 11211);**
3. **\$memcache_obj->close();**

关闭memcache连接

通过memcache.php可以方便的监控memcache状态，这是一个PHP文件，要求系统有PHP运行环境

下载地址：<http://livebookmark.net/memcachephp>

下载后需要编辑这个PHP文件来指定memcache的ip地址与端口号，同时要设置用户名与密码项即可

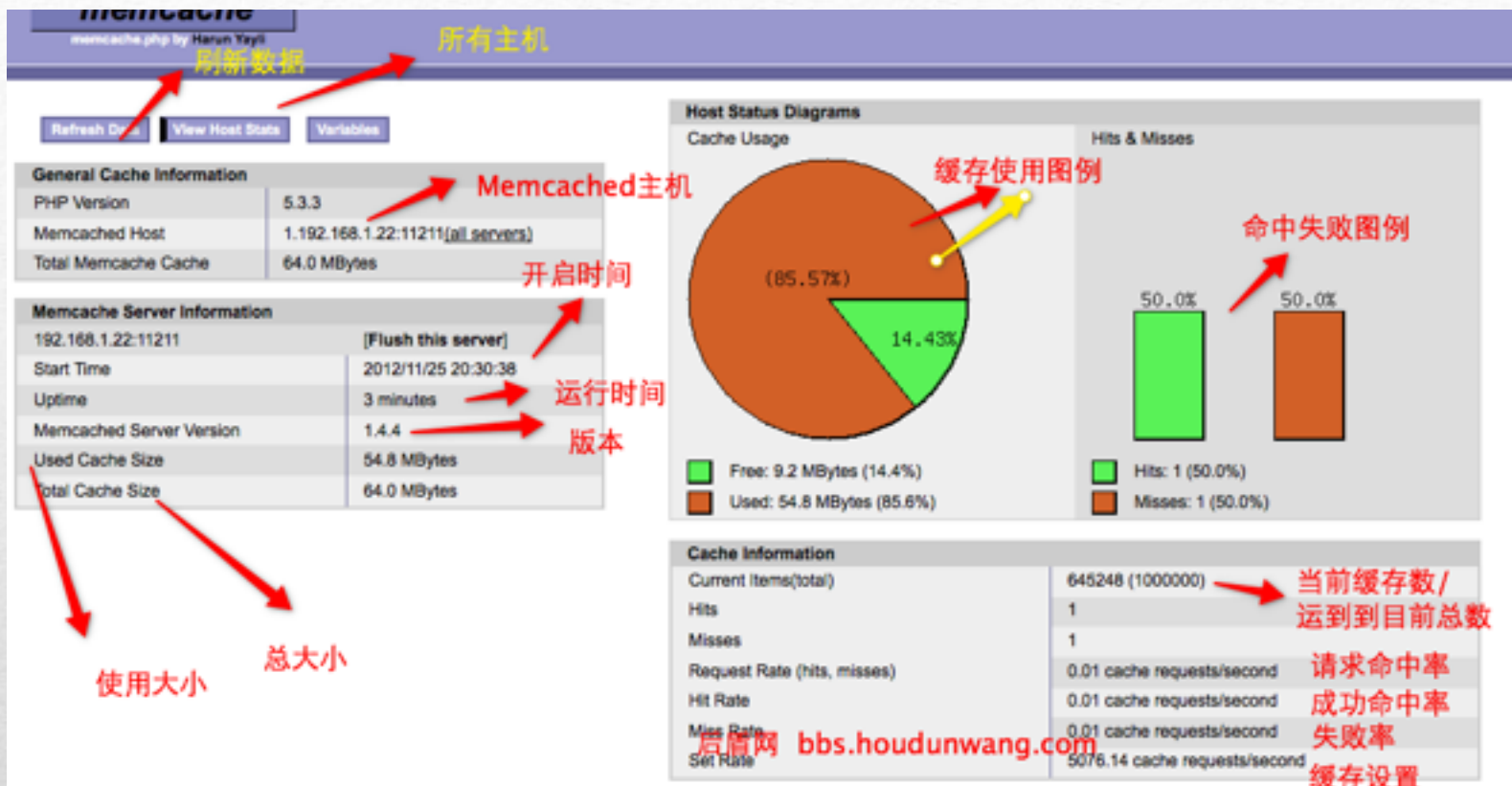
安装：

1.wget <http://livebookmark.net/memcachephp/memcachephp.zip>

2.unzip memcachephp.zip

3.解压后修改相应配置项

监控MemCache



监控MemCache

修改php.ini配置文件

- `session.save_handler = memcache`
- `session.save_path = "tcp://127.0.0.1:11211"`

Memcached操作session
