多个memcached服务的连接：

$mem\_servers=[

'192.168.28.1:11211', #服务器1

'192.168.250.129:20001' #服务器2

];

$mem=new memcache();

foreach ($mem\_servers as $value) {

$mem->addServer($value);

}

当向这两个服务器set数据时，如果不做任何处理的话会随机存储到两个服务器，但不会两个服务器都储存—即分布式存储/缓存。

memcached可以存储数组,那是因为PHP的mecached的扩展才存储之前自动对数组进行了序列化处理；实际存储在memcached中的序列化后的数据。

$k='cccc';

$v=[1,2,3,'a','b'];

// $flag=$mem->set($k,$v); //返回bool

var\_dump(is\_array($mem->get($k))); //true，证明memcached可以存储数组

即使此时拿掉一个memcached服务器，该集群任然能很好地工作（只是存储在该memcached服务器上的数据消失了客户端请求不到对应的数据了，需要从新加载到其他的memcached上-在内存足够大的情况下）

推荐的硬盘配置

CPU要求

命令

Memcached处理一小部分基本命令。

可以在协议文档中找到更全的说明。

基本协议

Memcached的基本协议部分包含对一个元素（item）运行一条命令。一个元素（item）的组成：

1. key （最大250字节，在ASCII模式下不能包含空格和换行）
2. 一个32bit的名为flag的数值
3. 以秒为单位的过期时间。0表示永远不过期。最长30天，30天之后被看做是一个确定日趋的unix时间戳
4. 64bit的CAS值，报纸唯一
5. 任何数据

CAS（Memcached的一种协议）是可选的（能够通过参数-C而完全禁止，而且内部有很多的空间来合成一个item，但是这些是与你的客户端交互的）

无回复

大多数的ASCII命令允许一个 noreply版本。在ASCII协议下不应该像通常一样使用。因为它不能够随着请求列出错误。意图避免在执行一个交互式命令（例如set或者add）后对回复的等待。

二进制协议可能恰当的实现了无回复声明。如果你有一个支持或者使用了二进制协议，也许你希望利用这个优势。

存储命令

set

非常常用的命令。存储指定的数据，该命令会覆盖已经存在的数据。新的元素将会在顶部（更具LRU近期最少使用算法）

add

该命令也是存储数据，但与set不同的是只有在数据不存在的时候才会存储，如果数据已经存在则添加失败，但是同时会将这个元素移动到LRU的前面。新的元素将会在顶部（根据LRU算法-近期最少使用算法）

replace

正如名字，替换。只有数据应存在的时候才会存储。该命令很少使用，并且在完全协议（protocol completeness）下存在。

append

将数据附加到一个已经存在的元素的最后字节的后面。这条命令不允许你扩展元素的限制。对操作list很有用。

prepend

与append类似，只是在已经存在的元素之前附加数据。

cas

检查并且设置（check and set或者compare and swap）。一个存储数据的命令，但是只操作那些自从你上次读取用来为更新过的数据。这个命令对于在竞态条件下更新数据很有帮助。

检索命令

get

该命令用于获取数据。获取一个或多个key，然后返回找到的所有元素（items）

译注：该命令可以跟多个key，key之间用空格隔开

gets

一个可选择的搭配CAS使用的命令。返回一个CAS标识符（一个唯一的64bit的数字）和该元素。如果元素的cas值自从你gets这个元素后已经改变了，它将不会被保存。

delete

从缓存中移除一个key，如果该key存在的话。

incr/decr

增加和减少。如果一个被存储的元素是64bit整型值的字符串表示或者就是数字，你可能执行incr后者decr命令来改变这个数字。你只允许增加或减少正的数值。该命令不接受负的数值（即不能增加或减少负数）。

如果元素不存在，那么增加或减少的命令将失败。

统计命令

stats

基本的统计命令

stats items

该命令返回一些信息，slab分块信息关于存储在memcached中的元素的信息。

stats slabs

返回更多的信息，被slab分块信息，关于存储在memcached中的元素的信息。更加聚焦于slab的性能而不是特殊元素的数量。

stats sizes

一个特殊的命令，该命令显示如果slabs被分成32字节buckets而不是你当前的slabs，元素将会怎么被分配/分布（distributed）。这条命令在可以检测你的slab分块的性能。

warning:这个是一个开发版（development）命令。对于1.4版本，它仍然是在某些时候锁住/阻塞 你的memcached实例的唯一命令。如果你有数百万的元素存储在memcached，执行这条命令可能导致几分钟内没有回复。只在你能容忍的风险内执行这条命令。是否支持此特性或者加速它的执行速度还在开发计划中。

flush\_all

使所有得缓存元素失效。后面可以跟一个可选的参数，这个参数意味着在多少秒后是所有的key失效。

这条命令不会暂停服务器因为它立即返回。它根本不会释放或者清空内存，它只是将所有的元素都标记为过期。

stats cachedump slab\_id limit\_num

显示某个slab中的前limit\_num个key列表，如果limit\_num为0表示显示所有

stats detail [on | off | dump]:设置或者显示详细操作记录

参数为on：打开详细操作记录

参数为off:关闭详细操作记录

参数dump:显示详细操作记录（每一个键值get set hit del的次数）