

普通余数分散法

余数计算的方法简单，数据的分散性也相当优秀，但也有其缺点。那就是当添加或移除服务器时，缓存重组的代价相当巨大。添加服务器后，余数就会产生巨变，这样就无法获取与保存时相同的服务器，从而影响缓存的命中率。

magent的hash算法

magent采用的是:Consistent Hashing原理, Consistent Hashing如下所示：首先求出memcached服务器（节点）的哈希值，并将其配置到0~232的圆（continuum）上。然后用同样的方法求出存储数据的键的哈希值，并映射到圆上。然后从数据映射到的位置开始顺时针查找，将数据保存到找到的第一个服务器上。如果超过232仍然找不到服务器，就会保存到第一台memcached服务器上。

Memcache集群的实现

memcached尽管是“分布式”缓存服务器，但服务器端并没有分布式功能。各个memcached不会互相通信以共享信息。那么，怎样进行分布式呢？这完全取决于客户端的实现。

magent是一个memcached代理软件(memcached agent)，又叫memagent，其项目网址为：

<https://github.com/wangmh/memagent>，防止单点现象，缓存代理也可以做备份，通过客户端连接到缓存代理服务器，缓存代理服务器连接缓存服务器。

它提供的功能及特点有：

- 1、和每个memcache server保持多个长连接，效果是减少memcache server保持的连接数量及创建销毁连接的开销。不过，memcache本身就支持大并发连接，这个功能也就没什么特别的说道。
 - 2、支持memcache的binary协议命令，实现请求的转发。
 - 3、和memcache一样，基于libevent的事件驱动来处理IO。
 - 4、支持ketama 的一致性hash算法。
 - 5、支持memcache backup集群，当memcache集群有机器挂了，memagent会将get请求转向memcache backup集群。这个功能对于cache的稳定性要求高的场景下会有有用武之地。
- 就提供的功能而言，memagent是个很简单的东西。对于较大的memcache集群，可以考虑搭一套memagent作为proxy使用。