分布式缓存:Memcached 采用一种集中式的缓存集群管理，也被称作**互不通信**的分布式架构方式。缓存与应用分离部署，缓存系统部署在一组专门的服务器上，应用程序**通过一致性Hash等路由算祛选择缓存服务器**远程访问缓存数据，**缓存服务器之间不通信**，缓存集群的规模可以**很容易地实现扩容**，具有良好的可伸缩性。

内存管理中一个令人头痛的问题就是内存碎片管理。操作系统、虚拟机垃圾回收在  
这方面想了许多办法:压缩、复制等。

Memcached使用了一个非常简单的办法一一固定空间分配。

Memcached将内存空间分为一组slab,每个slab里又包含一组chunk,同一个slab里的每个chunk的大小是固定的，拥有相同大小chunk的slab被组织在一起，叫作slab\_class,如图4.11所示。存储数据时根据数据的Size大小，寻找一个大于Size 的最小  
chunk 将数据写入。这种内存管理方式避免了内存碎片管理的问题，内存的分配和释放都  
是以chunk 为单位的。和其他缓存一样，Memcached 采用LRU 算法释放最近最久未被访  
问的数据占用的空间，释放的chunk 被标记为未用，等待下一个合适大小数据的写入。



分布式哈希一致算法：

普通的hash算法再去mod相应的机器数量，当即扩容时就会导致找到的节点不正确

一致性算法：

具体算法过程为:先构造一个长度为0~2^32的整数环( 这个环被称作一致性Hash 环),  
根据节点名称的Hash 值(其分布范围同样为0~232) 将缓存服务器节点放置在这个Hash  
环上。然后根据需要缓存的数据的KEY 值计算得到其Hash 值(其分布范围也同样为  
0~232 ),然后在**Hash 环上顺时针查找距离这个KEY 的Hash 值最近的缓存服务器节点**，  
完成KEY 到服务器的Hash 映射查找。

要新增node时，也只会影响很小的一部分数据

