memcache是内存型的数据库 ，因为内存的关闭释放的特性，memcache也无法持久化存储内容；

memcache内部是分块存储，所以大于1M的数据也无法存储。

memcache依赖libevent库，安装前需确认已经安装了libevent库。

memcache是一个轻量级的内存型数据库，只支持key-value型的存储。

memcache中没有关于用户，密码的设置，所以在配置时要配置防火墙的端口限制连接，以达到安全的目的。

使用repcached也能轻易实现memcache的单master单slave主从复制。

大家都知道，当有一个request过来后，web服务器交给app服务器，app处理并从db中存取相关数据，但db存取的花费是相当高昂的。

特别是每次都取相同的数据，等于是让数据库每次都在做高耗费的无用功，数据库如果会说话，肯定会发牢骚，你都问了这么多遍了，难道还记不住吗？

是啊，如果 app拿到第一次数据并存到内存里，下次读取时直接从内存里读取，而不用麻烦数据库，这样不就给数据库减负了？而且从内存取数据必然要比从数据库媒介取快 很多倍。

因此，我们可以在web/app层与db层之间加一层cache层，主要目的：1. 减少数据库读取负担；2. 提高数据读取速度。

而且，cache存取的媒介是内存，而一台服务器的内存容量一般都是有限制的，不像硬盘容量可以做到TB级别。

所以，可以考虑采用分布 式的cache层，这样更易于破除内存容量的限制，同时又增加了灵活性。

**2、使用memcached的场景**

想象有这么一个场景，一个电子商务网站，在网站的左侧会是商品的分类，中间是商品搜索结果的列表，可以查看商品信息和商家的基本信息和相关商家的信誉度信息。

在这个场景下，因为一个商场的类别不会经常改变的。实时性不是很高，这样应该放到缓存中取的。

 当这个过程首次发生时，数据将正常地从数据库或其他数据源加载，然后再存储到 memcached 内。

当下一次访问此信息时，它就会从 memcached 中取出，而不是从数据库加载，节省了时间和 CPU 循环。

但是要是数据中的数据改变怎么来更新memcached中的数据呢？

过程为：更新数据库中分类的信息------->找到memcached中key值，删除------>重新插入到你的memcached中就可以了

**适用memcached的业务场景：**

1) 存储大量不需要持久存储或数据库内已存在不会变动的数据。

2）如果数据库服务器的负载比较低但CPU使用率很高，这时可以缓存计算好的结果和渲染后的网页模板。

3）利用memcached可以缓存 session数据 、临时数据以减少对他们的数据库写操作。

4）缓存一些很小但是被频繁访问的文件。

**不适用memcached的业务场景：**

1）缓存对象的大小大于1MB Memcached本身就不是为了处理庞大的多媒体（large media）和巨大的二进制块（streaming huge blobs）而设计的。

2）key的长度大于250字符

3）虚拟主机不让运行memcached服务

如果应用本身托管在低端的虚拟私有服务器上，像vmware, xen这类虚拟化技术并不适合运行memcached。Memcached需要接管和控制大块的内存，如果memcached管理的内存 被OS或 hypervisor交换出去，memcached的性能将大打折扣。

4）应用运行在不安全的环境中

Memcached为提供任何安全策略，仅仅通过telnet就可以访问到memcached。如果应用运行在共享的系统上，需要着重考虑安全问题。

业务本身需要的是持久化数据或者说需要的应该是database

# **memcached的应用场景**

**http://www.cnblogs.com/literoad/archive/2012/12/23/2830178.html**

**<http://www.cnblogs.com/ggjucheng/p/3349240.html>**

**http://www.linuxidc.com/Linux/2012-03/56501.htm**

**<http://blog.csdn.net/yumengkk/article/details/7902103>**