原文链接：<http://os.51cto.com/art/201107/278292.htm>

**Redis能干啥？细看11种Web应用场景**

Redis的一个很大好处就是可以不用整个转入到这个数据库，而是可以沿用之前的MySQL等数据库，而仅在一些特定的应用场景通过Redis的特性提高效率。本文列出了11个这样的Web应用场景，如显示最新的项目列表、删除和过滤、排行榜等相关需求。

AD：

【51CTO精选译文】在[“怎样让Redis在你的系统中发挥作用”一文](http://database.51cto.com/art/201107/276333.htm)中，Salvatore 'antirez’ Sanfilippo告诉我们如何利用Redis独有的数据结构处理能力来解决一些常见问题。一些Redis原语命令比如LPUSH、LTRIM和LREM等等能够用来帮助开发者完成需要的任务——这些任务在传统的数据库存储中非常困难或缓慢。这是一篇非常有用并且实际的文章。那么要如何在你的框架中完成这些任务呢？

下面列出11种Web应用场景，在这些场景下可以充分的利用Redis的特性，大大提高效率。

**1．在主页中显示最新的项目列表。**

Redis使用的是常驻内存的缓存，速度非常快。LPUSH用来插入一个内容ID，作为关键字存储在列表头部。LTRIM用来限制列表中的项目数最多为5000。如果用户需要的检索的数据量超越这个缓存容量，这时才需要把请求发送到数据库。

**2．删除和过滤。**

如果一篇文章被删除，可以使用LREM从缓存中彻底清除掉。

**3．排行榜及相关问题。**

排行榜（leader board）按照得分进行排序。ZADD命令可以直接实现这个功能，而ZREVRANGE命令可以用来按照得分来获取前100名的用户，ZRANK可以用来获取用户排名，非常直接而且操作容易。

**4．按照用户投票和时间排序。**

这就像Reddit的排行榜，得分会随着时间变化。LPUSH和LTRIM命令结合运用，把文章添加到一个列表中。一项后台任务用来获取列表，并重新计算列表的排序，ZADD命令用来按照新的顺序填充生成列表。列表可以实现非常快速的检索，即使是负载很重的站点。

**5．过期项目处理。**

使用unix时间作为关键字，用来保持列表能够按时间排序。对current\_time和time\_to\_live进行检索，完成查找过期项目的艰巨任务。另一项后台任务使用ZRANGE...WITHSCORES进行查询，删除过期的条目。

**6．计数。**

进行各种数据统计的用途是非常广泛的，比如想知道什么时候封锁一个IP地址。INCRBY命令让这些变得很容易，通过原子递增保持计数；GETSET用来重置计数器；过期属性用来确认一个关键字什么时候应该删除。

**7．特定时间内的特定项目。**

这是特定访问者的问题，可以通过给每次页面浏览使用SADD命令来解决。SADD不会将已经存在的成员添加到一个集合。

**8．实时分析正在发生的情况，用于数据统计与防止垃圾邮件等。**

使用Redis原语命令，更容易实施垃圾邮件过滤系统或其他实时跟踪系统。

**9．Pub/Sub。**

在更新中保持用户对数据的映射是系统中的一个普遍任务。Redis的pub/sub功能使用了SUBSCRIBE、UNSUBSCRIBE和PUBLISH命令，让这个变得更加容易。

**10．队列。**

在当前的编程中队列随处可见。除了push和pop类型的命令之外，Redis还有阻塞队列的命令，能够让一个程序在执行时被另一个程序添加到队列。你也可以做些更有趣的事情，比如一个旋转更新的RSS feed队列。

**11．缓存。**

Redis缓存使用的方式与memcache相同。

网络应用不能无休止地进行模型的战争，看看这些Redis的原语命令，尽管简单但功能强大，把它们加以组合，所能完成的就更无法想象。当然，你可以专门编写代码来完成所有这些操作，但Redis实现起来显然更为轻松。

原文：[11 Common Web Use Cases Solved In Redis](http://highscalability.com/blog/2011/7/6/11-common-web-use-cases-solved-in-redis.html)