### 一、 redis与memcache的区别

**1.1** redis在数据支持上要比memecache多的多，Redis除K-V之外，还支持list,set,sorted set,hash等众多数据结构，

**1.2** Redis和Memcache都是将数据存放在内存中，都是内存数据库。不过memcache还可用于缓存其他东西，例如图片、视频等等。

**1.3** 虚拟内存–Redis当物理内存用完时，可以将一些很久没用到的value 交换到磁盘 。Redis采用hash结构来做K-V存储，由于其组合式的压缩，其内存利用率会高于Memcache。

**1.4** redis支持数据持久化，redis有部份存在硬盘上，这样能保证数据的持久性；memcache不支持数据持久化，断电就要重启。

**1.5** 性能:由于Redis只使用单核,而Memcache可以使用多核，所以平均每一个核上Redis在存储小数据时比Memcache性能更高。而在100K以上的数据时，Memcache性能要高于Redis。

**1.6** 数据持久化和主从复制时，只有redis拥有这两个特性。如果目标是构建一个缓存在升级或者重启后之前的数据不会丢失的话，那也只能选择redis。

**1.7** 如果需要缓存的数据只是key-value这样简单的结构时，采用memcache，它也足够的稳定可靠。如果涉及到存储，排序等一系列复杂的操作时，毫无疑问选择redis。

### 二、redis和MongoDB区别

**2.1** MongoDB与Redis都依赖内存，但redisTPS高于MongoDB

**2.2** MongoDB有丰富的数据表达,索引;最类似于关系型数据库,支持丰富的查询语句，redis数据丰富,较少的IO

**2.3** MongoDB适合大数据量存储,依赖系统虚拟内存,采用镜像文件存储;内存占用率比较高,官方建议独立部署在64位系统，Redis2.0后支持虚拟内存特性(VM) 突破物理内存限制;数据可以设置时效性,类似于memcache

**2.4** MongoDB支持master-slave,replicatset(内部采用paxos选举算法,自动故障恢复),auto sharding机制,对客户端屏蔽了故障转移和切片机制，redis依赖客户端来实现分布式读写;主从复制时,每次从节点重新连接主节点都要依赖整个快照,无增量复制;不支持auto sharding,需要依赖程序设定一致性hash机制

**2.5** MongoDB从1.8版本后,采用binlog方式(类似Mysql) 支持持久化，redis依赖快照进行持久化;AOF增强可靠性;增强性的同时,影响访问性能

**2.6** MongoDB不支持事务,靠客户端保证，redis支持事务,比较脆,仅能保证事务中的操作按顺序执行，而且MongoDB内置数据分析功能，redis不支持数据分析，所以， MongoDB海量数据的访问效率提升，相比之下redis较小数据量的性能和运算

**2.7** 当TPS(每秒传输的事物处理个数)非常高的时候，可以用到redis中数据持久化和分布式锁的内容，通过redis数据持久化，我们可以将缓存的数据保存到本地中来。此时使用redis较好。

**2.8** 当拥有海量数据但要求有限查询时间可以采用mongodb中的分片来实现，通过mongodb的分片机制，我们可以将海量的数据查询分别负载到不同的分片服务器上面，最后将数据查询的数据结果整合到一起。基于这种情况，不管数据量有多大，我们都可以实现快速的查询功能，查询时间约等于（数据量/分片数量）。分片其实本身就是一种高可用性的方案，因为每一个分片都保留着完整的一份数据，每次插入数据的时候，先插入一个主分片中，然后同步复制到所有从分片中，即使一个分片挂了，其余分片也能自动升级为主分片，继续工作。