

### Redis ( Remote Dictionary Server ) 即远程字典服务

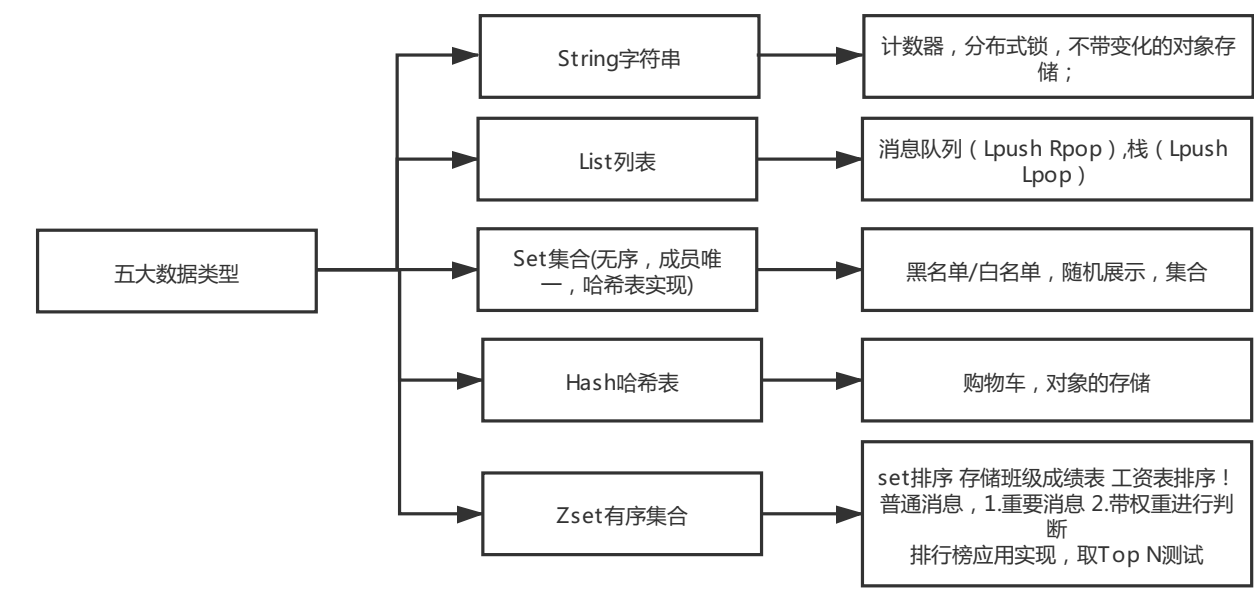
Redis与Memcached的区别：  
与Memcached一样，为了保证效率，两者数据都是保存在**内存**中，区别是Reids会周期性地把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现了master-slave（主从）同步。

Redis的效率：  
Redis是单线程的，基于内存操作，值得注意的是，内存是断电即失的，需要持久化（RDB、AOF），所以redis 的性能瓶颈不是CPU，而是机器内存和网络带宽，由于单线程中没有上下文切换的消耗，多次读写都在一个CPU上，所以Redis的效率高。

**redis默认有16个数据库（DB0~DB15），默认使用DB0**

## 基本命令

命令	功能
select n	切换到DB n
dbsize	查看当前数据库的大小, 与key数量相关
keys *	查看当前数据库中的所有key
flushdb	清空当前数据库中的所有键值对
flushall	清空所有数据库中的所有键值对
exists key	判断键是否存在
del key	删除键值对
move key db	将键值对移动到指定的数据库
expire key second	设置键值对的过期时间
type key	查看key的数据类型
ttl key	查看key的过期剩余时间( -1没有设置过期时间, -2已经过期)
rename key newkey	修改key的名称
renamex key newkey	仅当newkey不存在时, 修改key为新key



**Redis 事务：**  
Redis单条命令是可以保证原子性的，但redis事务不能保证原子性。

**操作过程：**  
1开始事务（multi）  
2命令入队  
3提交事务/执行事务(exec)  
注意：提交事务时才会一次性执行所有命令  
(取消事务 – discard)

**常见的错误：**  
代码语法错误(编译期异常)，所有命令都不会执行  
代码逻辑错误(运行期异常)，其他正确的命名会正常执行，不能保证事务的原子性。

**监控**

悲观锁：可以理解为认为什么事都很悲观，都要上锁  
乐观锁：使用“watch key”来监控指定的key，相当于乐观锁加锁。

**举例：**  
线程一对key上锁并开启事务，可线程二修改了key的value，当线程一提交事务后会失败，保证了事务的原子性。

**注意：**每次提交执行exec后都会自动释放锁，不管是否成功。

**持久化**

**RDB(Redis Databases)**  
在指定时间间隔后，将内存中的数据快照写入数据库。在恢复时，直接读取快照文件,进行数据的恢复。  
默认情况下，Redis将数据库快照保存在名字为dump.rdb的二进制文件中(可以在配置文件中自定义文件)

**工作原理：**在进行RDB的过程中，Redis主线程是不会做io操作的，主线程会fork一个子进程来完成该操作。

主线程调用fork，此时拥有父子进程  
子进程会将数据集写入一个临时的rdb文件中  
当子进程完成对新的rdb文件写入时，redis用新的rdb文件替换旧的rdb文件，并删除旧的rdb文件

**触发机制：**

配置文件中save满足的规则下自动触发  
执行flushall命令，自动触发  
退出redis，自动触发

**Save和Bgsave**  
“save命令”，是一个同步命令，会立即对当前内存中的数据进行持久化，会造成阻塞，当数据非常多的时候，redis命令的执行速度会非常慢，且会阻塞所有客户端的请求。  
“bgsave”命令是一个异步命令，进行持久化时，redis还可以继续响应客户端请求

**AOF(Append Only File)**  
以日志的形式记录所有执行命名的记录(除了读操作)，只会追加文件但不会改写文件

开启AOF，默认是关闭的  
需要修改配置文件，“appendonly yes” 默认为no，不开启AOF  
“appendfilename ‘appendonlyaof’ ” 默认文件名

**appendfsync的三种方式**  
1.always 每次修改都会sync,消耗性能  
2.everysec 每秒执行sync,可能会丢失这一秒的数据  
3.不自动执行sync,手动执行,速度最快

## RDB/AOF对比

类型	RDB	AOF
启动优先级	低	高
体积	小	大
恢复速度	快	慢
数据安全性	丢失数据	决策决定

## 如何选择使用哪种持久化方式？

一般来说， 如果想达到足以媲美 PostgreSQL 的数据安全性， 你应该同时使用两种持久化功能。

如果你非常关心你的数据， 但仍然可以承受数分钟以内的数据丢失， 那么你可以只使用 RDB 持久化。

有很多用户都只使用 AOF 持久化， 但并不推荐这种方式： 因为定时生成 RDB 快照 (snapshot) 非常便于进行数据库备份， 并且 RDB 恢复数据集的速度也要比 AOF 恢复的速度要快。