



链滴

Redis 进阶

作者: [kevin2020](#)

原文链接: <https://ld246.com/article/1514913931379>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

事务

multi

sadd key value

sadd key value

...

exec

(还能保证一个事务内的命令不被其他命令插入，如A执行几条命令，B执行1条命令，使用事务防止B入)

语法错误，若包含语法错误则exec后一个命令也不会执行（2.6.5前会执行正确的命令）

运行时错误，会执行正确的命令

redis没有提供回滚功能，所以执行出错需要手动处理错误

watch命令

对一个或多个键进行监控，持续到exec命令，一旦其中一个键被修改或删除，则事务不会执行

如果想取消监控，可以使用unwatch

过期时间

expire key seconds 为一个键设置过期时间 成功返回1 键不存在或失败返回0

ttl key 返回键剩余时间 -2表示不存在 -1 表示永久存在（2.6版本中，不存在还是没有过期时间都是-）

取消过期时间

1 persist key 成功清除返回1 否则0

2 为键重新赋值

3 再次expire会覆盖

其它 incr hset lpush等均无效

pexpire pttl 用法一样 单位为毫秒

Tips 如果watch监测了一个有过期时间的键 键到期被删除 不会认为键被改变

两个不常用的命令 expireat pexpireat 以unix时间为单位 后者为毫秒级

可以用来实现访问频率限制

缓存功能

大量使用缓存且过期时间过长会导致redis占满内存

可以修改配置文件中maxmemory 单位为字节

根绝maxmemory-policy删除不需要的键直到redis内存小于指定内存

volatile-lru随机lru删带过期时间 allkeys-lru 随机lru删

v-random随机删带过期时间 a-random随机删 v-ttl删过期时间最近 noeviction报错

LRU 最近最少使用算法（感谢操作系统中学到过）

（maxmemory samples设置实际数量，默认为3 Redis并不会准确删，每次随机3个按上述方案删）

Sort命令

如果集合，列表 有序集合中元素可以排序

可使用sort进行排序 对有序集合排序会忽略分数

sort key alpha参照字典顺序排序非数字元素

如果不加alpha则尝试转换为双精度浮点数来比较，转换失败报错

sort key desc limit 1 2 默认从小到大 desc倒序 limit同mysql

By命令

sort key by post:->time (表示为键名->字段名) 必须在->前，之后的会被当做常量处理，但并不会判定为常量参考键，会按元素本身大小排列。

对每个元素的值替换参考键中的第一个并获取值在排序

也可以是字符串

sort key by item:

当参考键不包含*即为常量时 返回结果与lrange相同

如果参考值相同会再比较本身的值

参考键不存在默认为0

未完待续

更了更了

Get命令

使sort命令返回结果不在是元素自身的值，而是get指定的键值。

sort tag:ruby:post by post:->time desc get post:->title get post:*->time get #

get #返回元素本身的值

Store命令

默认sort会直接返回结果 可使用store保存结果

sort tag:ruby:post by post:->time desc get post:->title get post:*->time get # store sort.result

保存后键为列表类型 返回结果的个数 键已存在 会覆盖

sort是Redis中最强大复杂的命令之一，复杂度为 $O(n+m\log(m))$

n表示排序列表个数，m表示返回的个数 会建立一个长度为n的临时容器来储存排序元素

当n太大时，会影响性能

注意3点 1使集合中元素尽可能少2使用limit限制获取结果数量3排序的数据量较大时，使用store保存

Redis任务队列

任务队列 相当于生产/消费模式

可以做到松耦合 方便系统进行扩展 降低单台服务器负载

BRPOP 与RPOP类似，区别为当 列表中没有元素时，会阻塞连接 直到有新元素加入 或超时
超时返回nil

BRPOP queue 0

0代表永不超时

BLPOP 类似

优先级队列

BRPOP可同时接受多个键 如

BRPOP queue1 queue2 queue3

如果不止一个键存在元素

则 从左往右取第一个

发布/订阅模式

publish channel hi

返回接受到消息的订阅者数量

subscribe channel1 channel2

可同时订阅多个频道

执行后进入订阅模式

只可以执行 subscribe unsubscribe psubscribe punsubscribe

返回3个值

1 subscribe订阅成功反馈 第二个值为订阅成功频道名 第三个为当前客户端订阅数量

2 message 第二个值表示产生消息的频道名 第三个为内容

3 unsubscribe 取消成功反馈 第二个值为对应频道名 第三个为当前订阅数量 0时会退出订阅模式

unsubscribe 可以取消多个频道 不指定参数取消所有

前面多两个p命令 指定参数时可以使用glob风格通配符，可以重复订阅多个频道

p命令与s命令订阅 退订 互不干扰

p * 无法匹配到 p xxx.* 匹配时为严格字符串匹配

管道

减少往返时延

当一组命令不依赖之前命令结果时 可使用管道发出

JRedis时可采用

节省空间

精简键名键值 但不可过简 不易维护 也容易冲突

内部编码优化

object encoding key 可查看

Redis启动时会预先建立0到9999数字的对象

之后存id时会指向这些地址 节约空间

当配置参数maxmemory后 不会使用共享对象

因为每个键值都要单独记录lru

到此为止，绝大多数命令已经介绍完毕。

下期预告

JRedis实战