1. 发布与订阅
2. 概述

Redis的发布与订阅功能由PUBLISH, SUBSCRIBE, PSUBSCRIBE等命令组成。通过订阅命令，客户端可以订阅一个或多个频道，每当有其他客户端向被订阅的频道发送消息时，频道的所有订阅者都会收到这条消息。除了订阅频道，客户端还可以通过PSUBSCRIBE命令订阅模式，模式以模式匹配的规则匹配到频道，只要有消息被发送到频道上，与这个频道匹配的模式的订阅者就能够收到消息。

1. 频道的订阅和退订

Redis将所有频道的订阅关系都保存在了服务器状态的pubsub\_channels字典中，这个字典的键是某个被订阅的频道，而键的值则是一个链表，保存了所有订阅了这个频道的客户端。

1. 模式的订阅和退订

同样的，模式的订阅关系也被保存在了服务器状态的pubsub\_Patterns链表中，每个节点就是一个pubsub\_Pattern结构，该结构保存了被订阅的模式，以及订阅该模式的客户端。

1. 发送消息

当一个Redis客户端执行PUBLISH <channel> <message>命令将消息message发送给频道channel时，服务器会将消息发送给频道的所有订阅者，同时如果有多个模式与频道匹配，模式的订阅者也会收到消息。

1. 查看订阅消息

PUBSUB命令是Redis2.8新增的命令，客户端可以通过这个命令来查看频道或者模式的信息，如订阅者数量。

1. PUBSUB CHANNELS [pattern]

返回服务器当前被订阅的所有频道，如果添加了模式参数，则还得满足与pattern模式匹配这个条件。

1. PUBSUB NUMSUB

接受任意多个频道作为输入参数，返回这些频道的订阅者数量。

1. PUBSUB NUMPAT

返回服务器当前被订阅模式的数量。

1. 事务
2. 概述

Redis通过MULTI, EXEC, WATCH,等命令来实现事务，事务将一个或多个命令请求打包，然后按照顺序执行这些命令。

1. 事务的实现
2. 事务开始

MULTI命令标识着事务的开始，该命令可以将命令的客户端从非事务状态切换至事务状态，通过在客户端状态的flags属性中打开REDIS\_MULTI标识来完成的。

1. 命令入队

当一个客户端处于非事务状态时，这个客户端发送的命令会被立即执行；而客户端转为事务状态后，如果客户端发送的命令时EXEC, DISCARD, WATCH, MULTI四个中的一个，那么服务器立即执行这个命令，否则不会执行，而是将该命令放入一个队列中，向客户端返回QUEUED回复。

1. 事务队列

每个Redis客户端都有自己的事务状态，事务状态包含了一个事务队列，以及计数器，队列中保存了已入队命令的函数指针，参数，以及参数的数量。

1. 执行事务

当一个处于事务状态的客户端向服务器发送EXEC命令，则会被立即执行，服务器会遍历这个客户端的事务队列，执行队列中保存的所有命令，并将执行结果全部返回给客户端。

1. WATCH命令
2. 概述

WATCH命令是一个乐观锁，它可以在EXEC命令执行之前，监视任意数量的数据库键，并在EXEC命令执行时，检查被监视的键是否至少有一个已经被修改过了，如果是的话，则拒绝执行事务，向客户端返回代表事务执行失败的空回复，这确保了事务的安全性。

1. 实现

每个Redis数据库中都保存了一个watched\_keys字典，这个字典的键是某个被WATCH命令监视的数据库键，而字典的值是一个链表，记录了所有监视该数据库键的客户端。通过该字典，服务器可以知道哪些数据库键正在被监视，哪些客户端正在监视着这些数据库键。

1. 监视机制的触发

所有对数据库进行修改的命令，在执行之后都会调用函数对watched\_keys字典进行检查，查看是否有客户端正在监视刚刚被命令修改过的数据库键，如果有的话，则会将监视该数据库键的所有客户端的REDIS\_DIRTY\_CAS标识打开，标识该客户端的事务安全性已破坏。

1. 判断事务是否可执行

当服务器接收到一个客户端发来的EXEC命令时，服务器根据这个客户端的REDIS\_DIRTY\_CAS标识来决定是否执行事务。

1. 事务的ACID
2. 原子性

事务中多个操作当做一个整体执行，但还是Redis中不支持事务的回滚，即时某个命令在执行期间出现了错误，整个事务也会继续执行下去，直到所有命令执行完毕。

1. 一致性

事务具有一致性指的是如果数据库在执行事务之前是一致的，那么在事务执行之后，无论事务会否执行成功，数据库也应该是一致的。一致指的是数据符合数据库本身的定义和要求，没有包含非法或者无效的错误数据，因此Redis通过错误检测和简单地设计来保证事务的一致性。

1. 入队错误

一个事务在入队命令中，出现了命令不存在或者命令格式不正确，则拒绝执行这个事务，因此不会影响一致性。

1. 执行错误

事务在执行过程中发生错误，服务器也不会中断事务的执行，因此不会影响一致性。

1. 服务器停机

如果Redis服务器在执行事务的过程中停机，根据服务器所使用的持久化模式判断：

如无持久化，则重启后数据库是空白的，因此一定一致；

如有RDB持久化，则使用RDB恢复数据，一定一致；

如有AOF持久化，则使用AOF恢复数据，一定一致；

1. 隔离性

Redis采用单线程执行事务，具有隔离性。

1. 耐久性

根据Redis的持久化模式决定，如RDB模式下 ，由于BGSAVE命令是异步执行的，不能保证事务数据被第一时间保存到硬盘中；在AOF模式下，根据appendfsync选项的值，决定耐久性。

注意，也可以在每一个事务的最后加上一个SAVE命令来保证事务的持久性，不过效率不高。

1. Lua脚本
2. 概述

Redis从2.6版本引入对Lua脚本的支持，通过在服务器中嵌入Lua环境，Redis客户端可以使用Lua脚本，直接在服务器端原子性地执行多个Redis命令。

1. Lua环境协作组件
2. 伪客户端

因为执行Redis命令必须有相应的客户端状态，所以为了执行Lua脚本中包含的Redis命令，Redis服务器专门为Lua环境创建了一个伪客户端，由它负责处理Redis命令。Lua脚本使用redis.call函数执行一个Redis命令，发送给伪客户端，伪客户端将命令传给命令执行器，然后将结果返回给Lua环境。

1. lua\_scripts字典

这个字典的键为某个Lua脚本的SHA1校验和，而字典的值则是SHA1校验和对应的Lua脚本，Redis会将所有被EVAL命令执行过的Lua脚本，以及所有被SCRIPT LOAD命令载入过的Lua脚本都保存到这个字典中。

1. EVAL命令

根据客户端执行的Lua脚本，在Lua环境中定义一个与这个脚本对应的Lua函数，Lua函数的名字由f\_前缀加上脚本的SHA1校验和组成，然后将给定的脚本保存到lua\_scripts字典中，最后执行刚刚定义的函数。

1. EVALSHA命令

该命令根据脚本的SHA1校验和来调用脚本对应的函数，要求是该脚本必须被EVAL命令执行过或者被SCRIPT LOAD命令载入过。

1. 脚本管理命令
2. SCRIPT FLUSH

清除服务器中所有与Lua脚本有关的信息，释放并重建lua\_scripts字典，重建新的Lua环境。

1. SCRIPT EXISTS

根据输入的SHA1校验和，检查对应的脚本是否在服务器中。

1. SCRIPT LOAD

载入脚本命令，命令首先在Lua环境中为脚本创建对应的函数，然后再将脚本保存到lua\_scripts字典中，但是不执行。

1. 慢查询日志
2. 概述

Redis的慢查询日志用于记录执行时间超过给定时长的命令请求，用户可以通过这个功能产生的日志来监视和优化查询速度。

1. 配置

服务器配置有两个和慢查询相关的配置：

1. slowlog-log-slower-than

执行执行事件超过多少微妙的命令请求会被记录到日志上。

1. slowlog-max-len

执行服务器最多保存多少条慢查询日志，采用FIFO的方式保存日志。

1. 查阅和删除
2. SLOWLOG GET

查阅慢查询日志。

1. SLOWLOG LEN

查看日志数量。

1. SLOWLOG RESET

清除所有的慢查询日志。