

和众多其他数据库一样，redis做为Nosql数据库也同样提供了事务机制，在redis中，MULTI/EXEC/DISCARD/WATCH这四个命令是我们实现事务的基石。相信对有关系型数据库开发经验的开发者而言对这一概念并不陌生。下面列出redis事务的实现特征：

1. 在事务中的所有命令都将会被串行化的顺序执行，事务执行期间，redis不会再为其他客户端的请求提供任何服务，从而保证了事务中的所有命令被原子的执行。
2. 和关系型数据库中的事务相比，在redis事务中如果有一条命令执行失败，其后的命令仍然会继续执行。
3. 我们可以通过MULTI命令开启一个事务，有关系型数据库开发经验的人可以将其理解为“BEGIN TRANSACTION”语句。在该语句之后执行的命令都将被视为事务之内的操作，最后我们可以通过执行EXEC/DISCARD命令来提交/回滚该事务内的所有操作。这两个redis命令可被是等同于关系型数据库中的commit/rollback语句。
4. 在事务开启之前，如果客户端与服务器之间出现通讯故障并导致网络断开，其后所有待执行的语句都将不会被服务器执行，然而如果网络中断时间是发生在客户端执行exec命令之后，那么该事务中的所有命令都会被服务器执行。
5. 当使用append-only模式时，redis会通过调用系统函数write将该事务内的所有写操作在本次调用中全部写入磁盘。然而如果写入的过程中出现系统崩溃，如电源故障导致的宕机，那么此时也许只有部分数据被写入磁盘，而另一部分数据却已经丢失。redis服务器会在重新启动时执行一系列必要的一致性检查，一旦发现类似问题，就会立即退出并给出相应的错误提示。此时，我们就要充分利用redis工具包中提供的redis-check-aof工具，该工具可以帮助我们定位到数据不一致的错误，并将已经写入的部分数据进行回滚。修复之后我们就可以再次重新启动redis服务器了。