

redis是一种高级key-value数据库，可以对数据进行持久化处理，而且支持的数据类型非常丰富，redis的数据都存放在内存中，如果没有配置持久化，redis重启后数据就全丢失了，于是需要开启redis持久化功能，将数据保存在磁盘上，当redis重启时，可以从磁盘中恢复数据。redis提供两种方式进行持久化。

1. RDB持久化，原理是将redis在内存中的数据库记录定时dump到磁盘上的RDB持久化。
2. AOF持久化，原理是将redis的操作日志以追加的方式写入文件。

二者的区别：

RDB持久化是指在指定的时间间隔内将内存中的数据集写入磁盘，实际操作过程是fork一个子进程，先将数据集写入临时文件，然后再替换之前的文件，用二进制压缩存储。

AOF持久化是以日志的形式记录服务器所处理的每一个写、删除操作，查询操作不会记录，以文本的方式记录，可以打开文件看到详细的操作记录。

二者优缺点：

RDB优势：

1. 一旦采用该方式，那么redis数据库将只包含一个文件，这对于文件备份而言是非常完美的，通过这样的备份策略，一旦系统出现灾难性故障，可以非常容易的恢复。
2. 对于灾难恢复而言，RDB是非常不错的选择，因为我们可以将单独的压缩文件转移到其他存储介质中。
3. 性能最大化，对于redis的服务进程而言，在开始持久化时，唯一需要做的是fork出子进程，之后再由子进程完成这些持久化工作，这样就可以极大的避免服务进程执行IO操作。
4. 相比于AOF机制，如果数据集很大，RDB的启动效率会更高。

缺点：

1. 如果想保证数据的高可用性，即最大限度的避免数据丢失，那么RDB将不是一个很好的选择。因为系统一旦在定时持久化之前出现宕机现象，此前没有来得及写入磁盘的数据都将丢失。
2. 由于RDB是通过fork子进程来协助完成数据持久化工作的，因此，当数据集较大时候，可能会导致整个服务器停止服务几百毫秒。

AOF的优势：

1. 该机制可以带来更高的数据安全性，即数据持久性。比较差的就是一旦系统出现宕机现象，那么这一秒钟之内修改的数据将会丢失。
2. 由于该机制对日志文件的写入操作采用的是append模式，因此在写入过程中即使出现宕机的现象，也不会破坏日志文件中已经存在的内ring，然而如果我们本次操作只是写入了一半数据就出现了系统奔溃问题，不用担心，在redis下一次启动之前，我们可以通过redis-check-aof工具来帮助我们解决数据一致性问题。
3. AOF包含一个格式清晰、易于理解的日志文件用于记录所有的修改操作。

缺点：

1. 对于相同数量的数据集而言，AOF文件通常要比RDB文件，RDB在恢复大数据集时的速度要比RDB的恢复速度要快。
2. 根据同步策略不同，AOF在运行效率上慢于RDB。

TCP是如何确定网络进入了拥塞状态的，TCP认为网络拥塞的主要依据是它重传了一个报文段。TCP对每一个报文段都有一个定时器，称为重传定时器(RTO)，当RTO超时且还没有得到数据确认，那么TCP就会对该报文段进行重传，当发生超时，那么出现拥塞的可能性就很大。