

redis持久化的方式

- RDB：Redis DataBase 持久化内存数据到磁盘
 - redis主进程Fork一个子进程，定时并且达到出发修改次数时，进行写磁盘，写到RDB文件中
- AOF：Append Only File 持久化修改数据的命令
 - 以 每秒/每个命令./不同步 等方式进行

RDB和AOF的区别

REDIS持久化 <small>📖</small>		
	RDB	AOF
实现方式	内存快照：RDB持久化是指在指定的时间间隔内将内存中的数据集快照写入磁盘，实际操作过程是fork一个子进程，先将数据集写入临时文件，写入成功后，再替换之前的文件，用二进制压缩存储	日志：AOF持久化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
配置文件名	dump.rdb	appendonly.aof
配置文件和意义	save 900 1: 900秒内有一次更改 save 300 10: 300秒内有10次更改 save 60 10000.60秒内有1万次更改 子主题 15	appendfsync always：同步持久化，每次发生数据变化会立刻写入到磁盘中。性能较差当数据完整性比较好（慢，安全） appendfsync everysec：出厂默认推荐，每秒异步记录一次（默认值） appendfsync no：将数据交给操作系统来处理，由操作系统来决定什么时候同步数据。更快，也更不安全的选择
恢复数据方法	将dump.rdb 文件拷贝到 redis的安装目录的bin目录下，重启redis服务即可	将appendonly.aof 文件拷贝到 redis的安装目录的bin目录下，重启redis服务即可。 但在实际开发中，可能因为某些原因导致appendonly.aof 文件格式异常，从而导致数据还原失败，可以通过命令redis-check-aof --fix appendonly.aof 进行修复
优点	1 适合大规模的数据恢复。 2 如果业务对数据完整性和一致性要求不高，RDB是很好的选择。	数据的完整性和一致性更高
缺点	1 数据的完整性和一致性不高，因为RDB可能在最后一次备份时宕机了。 2 备份时占用内存，因为Redis 在备份时会独立创建一个子进程，将数据写入到一个临时文件（此时内存中的数据是原来的两倍哦），最后再将临时文件替换之前的备份文件。	因为AOF记录的内容多，文件会越来越大，数据恢复也会越来越慢
是否默认开启	是	否 appendonly yes：开启aof日志持久化
触发机制	save 900 1: 900秒内有一次更改 save 300 10: 300秒内有10次更改 save 60 10000.60秒内有1万次更改	auto-aof-rewrite-percentage 100：触发AOF文件执行重写的增长率 auto-aof-rewrite-min-size 64mb：当AOF文件大小是上次rewrite后大小的一倍且文件大于64MB时触发。一般都设置为3G，64M太小了

如何能够最大化保证恢复数据的速度及数据的完整性

混合持久化