**Redis的持久化策略**：2种

* **RDB**：快照形式是直接把内存中的数据保存到一个 dump 文件中，定时保存，保存策略。
* **AOF**：把所有的对Redis的服务器进行修改的命令都存到一个文件里，命令的集合。

**RDB 持久化**

**默认 Redis 是会以快照 “RDB” 的形式将数据持久化到磁盘的，一个二进 制文件，dump.rdb**

**工作原理简单介绍一下**：当 Redis 需要做持久化时，Redis 会 fork 一个子进程，子进程将数据写到磁盘上一个临时 RDB 文件中。当子进程完成写临时文件后，将原来的 RDB 替换掉，这样的好处就是可以 copy-on-write。

Redis默认情况下，是快照 RDB 的持久化方式，将内存中的数据以快照的方式写入二进制文件中，默认的文件名是 dump.rdb 。当然我们也可以手动执行 save 或者 bgsave（异步）做快照。

**Redis.conf配置** ：默认是如下配置

save 900 1

save 300 10

save 60 10000

复制代码

* 900秒之内，如果超过1个key被修改，则发起快照保存；
* 300秒内，如果超过10个key被修改，则发起快照保存；
* 1分钟之内，如果1万个key被修改，则发起快照保存；

**RDB 的优点:**

这种文件非常适合用于进行备份： 比如说，你可以在最近的 24 小时内，每小时备份一次 RDB 文件，并且在每个月的每一天，也备份一个 RDB 文件。 这样的话，即使遇上问题，也可以随时将数据集还原到不同的版本。**RDB 非常适用于灾难恢复**（disaster recovery）。

**RDB 的缺点:**

**如果你需要尽量避免在服务器故障时丢失数据**，**那么 RDB 不适合你**。 虽然 Redis 允许你设置不同的保存点（save point）来控制保存 RDB 文件的频率， 但是， 因为RDB 文件需要保存整个数据集的状态， 所以它并不是一个轻松的操作。 因此你可能会至少 5 分钟才保存一次 RDB 文件。 在这种情况下， **一旦发生故障停机， 你就可能会丢失好几分钟的数据**。

**AOF 持久化**

使用 AOF 做持久化，每一个写命令都通过write函数追加到 appendonly.aof 中,配置方式：启动 AOF 持久化的方式

**Redis.conf配置**

appendfsync yes

appendfsync always #每次有数据修改发生时都会写入AOF文件。

appendfsync everysec #每秒钟同步一次，该策略为AOF的缺省策略。

复制代码

AOF 就可以做到全程持久化，只需要在配置文件中开启（默认是no），appendonly yes开启 AOF 之后，Redis 每执行一个修改数据的命令，都会把它添加到 AOF 文件中，当 Redis 重启时，将会读取 AOF 文件进行“重放”以恢复到 Redis 关闭前的最后时刻。

**AOF 的优点**

使用 AOF 持久化会让 Redis 变得非常耐久（much more durable）：**你可以设置不同的 fsync 策略**，比如无 fsync ，每秒钟一次 fsync ，或者每次执行写入命令时 fsync 。 **AOF 的默认策略为每秒钟 fsync 一次**，在这种配置下，Redis 仍然可以保持良好的性能，并且就算发生故障停机，也**最多只会丢失一秒钟的数据**（ fsync 会在后台线程执行，所以主线程可以继续努力地处理命令请求）。

**AOF 的缺点**

对于相同的数据集来说，AOF 文件的体积通常要大于 RDB 文件的体积。**根据所使用的 fsync 策略，AOF 的速度可能会慢于 RDB**。 **在一般情况下， 每秒 fsync 的性能依然非常高**， 而关闭 fsync 可以让 AOF 的速度和 RDB 一样快， 即使在高负荷之下也是如此。 不过在处理巨大的写入载入时，RDB 可以提供更有保证的最大延迟时间（latency）。

作者：关注公众号\_搜云库\_每天更新  
链接：https://juejin.im/post/5ab5f08e518825557f00dfac  
来源：掘金  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。