



云数据库Redis版免费送  
最长半年免费用, 轻松上云, 告别繁琐自建

免费领取



This page provides a technical description of Redis persistence, it is a suggested read for all the Redis users. For a wider overview of Redis persistence and the durability guarantees it provides you may want to also read [Redis persistence demystified](#).

## Redis 持久化

Redis 提供了不同级别的持久化方式:

- RDB持久化方式能够在指定的时间间隔能对你的数据进行快照存储.
- AOF持久化方式记录每次对服务器写的操作,当服务器重启的时候会重新执行这些命令来恢复原始的数据,AOF命令以redis协议追加保存每次写的操作到文件末尾.Redis还能对AOF文件进行后台重写,使得AOF文件的体积不至于过大.
- 如果你只希望你的数据在服务器运行的时候存在,你也可以不使用任何持久化方式.
- 你也可以同时开启两种持久化方式, 在这种情况下, 当redis重启的时候会优先载入AOF文件来恢复原始的数据,因为在通常情况下AOF文件保存的数据集要比RDB文件保存的数据集要完整.
- 最重要的事情是了解RDB和AOF持久化方式的不同,让我们以RDB持久化方式开始:

### RDB的优点

- RDB是一个非常紧凑的文件,它保存了某个时间点得数据集,非常适用于数据集的备份,比如你可以在每个小时报保存一下过去24小时内的数据,同时每天保存过去30天的数据,这样即使出了问题你也可以根据需求恢复到不同版本的数据集.
- RDB是一个紧凑的单一文件,很方便传送到另一个远端数据中心或者亚马逊的S3 (可能加密), 非常适用于灾难恢复.
- RDB在保存RDB文件时父进程唯一需要做的就是fork出一个子进程,接下来的工作全部由子进程来做,父进程不需要再做其他IO操作,所以RDB持久化方式可以最大化redis的性能.
- 与AOF相比,在恢复大的数据集的时候, RDB方式会更快一些.

### RDB的缺点

- 如果你希望在redis意外停止工作 (例如电源中断) 的情况下丢失的数据最少的话, 那么RDB不适合你.虽然你可以配置不同的save时间点(例如每隔5分钟并且对数据集有100个写的操作),是Redis要完整的保存整个数据集是一个比较繁重的工作,你通常会每隔5分钟或者更久做一次完整的保存,万一在Redis意外宕机,你可能会丢失几分钟的数据.
- RDB 需要经常fork子进程来保存数据集到硬盘上,当数据集比较大的时候,fork的过程是非常耗时的,可能会导致Redis在一些毫秒级内不能响应客户端的请求.如果数据集巨大并且CPU性能不是

很好的情况下,这种情况会持续1秒,AOF也需要fork,但是你可以调节重写日志文件的频率来提高数据集的耐久度.

## • AOF 优点

- 使用AOF 会让你的Redis更加耐久: 你可以使用不同的fsync策略: 无fsync,每秒fsync,每次写的时候fsync.使用默认的每秒fsync策略,Redis的性能依然很好(fsync是由后台线程进行处理的,主线程会尽力处理客户端请求),一旦出现故障,你最多丢失1秒的数据.
- AOF文件是一个只进行追加的日志文件,所以不需要写入seek,即使由于某些原因(磁盘空间已满,写的过程中宕机等等)未执行完整的写入命令,你也也可使用redis-check-aof工具修复这些问题.
- Redis 可以在 AOF 文件体积变得过大时,自动地在后台对 AOF 进行重写: 重写后的新 AOF 文件包含了恢复当前数据集所需的最小命令集合。整个重写操作是绝对安全的,因为 Redis 在创建新 AOF 文件的过程中,会继续将命令追加到现有的 AOF 文件里面,即使重写过程中发生停机,现有的 AOF 文件也不会丢失。而一旦新 AOF 文件创建完毕,Redis 就会从旧 AOF 文件切换到新 AOF 文件,并开始对新 AOF 文件进行追加操作。
- AOF 文件有序地保存了对数据库执行的所有写入操作, 这些写入操作以 Redis 协议的格式保存, 因此 AOF 文件的内容非常容易被别人读懂, 对文件进行分析 (parse) 也很轻松。导出 (export) AOF 文件也非常简单: 举个例子, 如果你不小心执行了 FLUSHALL 命令, 但只要 AOF 文件未被重写, 那么只要停止服务器, 移除 AOF 文件末尾的 FLUSHALL 命令, 并重启 Redis , 就可以将数据集恢复到 FLUSHALL 执行之前的状态。

## • AOF 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) 。

## • 如何选择使用哪种持久化方式?

一般来说, 如果想达到足以媲美 PostgreSQL 的数据安全性, 你应该同时使用两种持久化功能。如果你非常关心你的数据, 但仍然可以承受数分钟以内的数据丢失, 那么你可以只使用 RDB 持久化。

有很多用户都只使用 AOF 持久化, 但我们并不推荐这种方式: 因为定时生成 RDB 快照 (snapshot) 非常便于进行数据库备份, 并且 RDB 恢复数据集的速度也要比 AOF 恢复的速度要快, 除此之外, 使用 RDB 还可以避免之前提到的 AOF 程序的 bug 。

**Note:** 因为以上提到的种种原因, 未来我们可能会将 AOF 和 RDB 整合成单个持久化模型。(这是一个长期计划。) 接下来的几个小节将介绍 RDB 和 AOF 的更多细节。

## 快照

在默认情况下，Redis 将数据库快照保存在名字为 `dump.rdb` 的二进制文件中。你可以对 Redis 进行设置，让它在“N 秒内数据集至少有 M 个改动”这一条件被满足时，自动保存一次数据集。你可以通过调用 `SAVE` 或者 `BGSAVE`，手动让 Redis 进行数据集保存操作。

比如说，以下设置会让 Redis 在满足“60 秒内有至少有 1000 个键被改动”这一条件时，自动保存一次数据集：

```
save 60 1000
```

这种持久化方式被称为快照 `snapshotting`。

## 工作方式

当 Redis 需要保存 `dump.rdb` 文件时，服务器执行以下操作：

- Redis 调用 `forks`。同时拥有父进程和子进程。
- 子进程将数据集写入到一个临时 `RDB` 文件中。
- 当子进程完成对新 `RDB` 文件的写入时，Redis 用新 `RDB` 文件替换原来的 `RDB` 文件，并删除旧的 `RDB` 文件。

这种工作方式使得 Redis 可以从写时复制 (`copy-on-write`) 机制中获益。

## 只追加操作的文件 (Append-only file, AOF)

快照功能并不是非常耐久 (`durable`)：如果 Redis 因为某些原因而造成故障停机，那么服务器将丢失最近写入、且仍未保存到快照中的那些数据。从 1.1 版本开始，Redis 增加了一种完全耐久的持久化方式：AOF 持久化。

你可以在配置文件中打开 AOF 方式：

```
appendonly yes
```

从现在开始，每当 Redis 执行一个改变数据集的命令时（比如 `SET`），这个命令就会被追加到 AOF 文件的末尾。这样的话，当 Redis 重新启时，程序就可以通过重新执行 AOF 文件中的命令来达到重建数据集的目的。

## 日志重写

因为 AOF 的运作方式是不断地将命令追加到文件的末尾，所以随着写入命令的不断增加，AOF 文件的体积也会变得越来越大。举个例子，如果你对一个计数器调用了 100 次 `INCR`，那么仅仅是

为了保存这个计数器的当前值，AOF 文件就需要使用 100 条记录（entry）。然而在实际上，只使用一条 SET 命令已经足以保存计数器的当前值了，其余 99 条记录实际上都是多余的。

为了处理这种情况，Redis 支持一种有趣的特性：可以在不中断服务客户端的情况下，对 AOF 文件进行重建（rebuild）。执行 BGREWRITEAOF 命令，Redis 将生成一个新的 AOF 文件，这个文件包含重建当前数据集所需的最少命令。Redis 2.2 需要自己手动执行 BGREWRITEAOF 命令；Redis 2.4 则可以自动触发 AOF 重写，具体信息请查看 2.4 的示例配置文件。

## AOF有多耐用？

你可以配置 Redis 多久才将数据 fsync 到磁盘一次。有三种方式：

- 每次有新命令追加到 AOF 文件时就执行一次 fsync：非常慢，也非常安全
- 每秒 fsync 一次：足够快（和使用 RDB 持久化差不多），并且在故障时只会丢失 1 秒钟的数据。
- 从不 fsync：将数据交给操作系统来处理。更快，也更不安全的选择。
- 推荐（并且也是默认）的措施为每秒 fsync 一次，这种 fsync 策略可以兼顾速度和安全性。

## 如果AOF文件损坏了怎么办？

服务器可能在程序正在对 AOF 文件进行写入时停机，如果停机造成了 AOF 文件出错（corrupt），那么 Redis 在重启时会拒绝载入这个 AOF 文件，从而确保数据的一致性不会被破坏。当发生这种情况时，可以用以下方法来修复出错的 AOF 文件：

- 为现有的 AOF 文件创建一个备份。
- 使用 Redis 附带的 redis-check-aof 程序，对原来的 AOF 文件进行修复：  

```
$ redis-check-aof -fix
```
- （可选）使用 diff -u 对比修复后的 AOF 文件和原始 AOF 文件的备份，查看两个文件之间的不同之处。
- 重启 Redis 服务器，等待服务器载入修复后的 AOF 文件，并进行数据恢复。

## 工作原理

AOF 重写和 RDB 创建快照一样，都巧妙地利用了写时复制机制：

- Redis 执行 fork()，现在同时拥有父进程和子进程。
- 子进程开始将新 AOF 文件的内容写入到临时文件。
- 对于所有新执行的写入命令，父进程一边将它们累积到一个内存缓存中，一边将这些改动追加到现有 AOF 文件的末尾，这样即使在重写中途发生停机，现有的 AOF 文件也还是安全的。
- 当子进程完成重写工作时，它给父进程发送一个信号，父进程在接收到信号之后，将内存缓存中的所有数据追加到新 AOF 文件的末尾。
- 搞定！现在 Redis 原子地用新文件替换旧文件，之后所有命令都会直接追加到新 AOF 文件的末尾。

## 怎样从RDB方式切换为AOF方式

在 Redis 2.2 或以上版本，可以在不重启的情况下，从 RDB 切换到 AOF：

- 为最新的 `dump.rdb` 文件创建一个备份。
- 将备份放到一个安全的地方。
- 执行以下两条命令：
- `redis-cli config set appendonly yes`
- `redis-cli config set save ""`
- 确保写命令会被正确地追加到 AOF 文件的末尾。
- 执行的第一条命令开启了 AOF 功能：Redis 会阻塞直到初始 AOF 文件创建完成为止，之后 Redis 会继续处理命令请求，并开始将写入命令追加到 AOF 文件末尾。

执行的第二条命令用于关闭 RDB 功能。这一步是可选的，如果你愿意的话，也可以同时使用 RDB 和 AOF 这两种持久化功能。

重要:别忘了在 `redis.conf` 中打开 AOF 功能！否则的话，服务器重启之后，之前通过 `CONFIG SET` 设置的配置就会被遗忘，程序会按原来的配置来启动服务器。

## AOF和RDB之间的相互作用

在版本号大于等于 2.4 的 Redis 中，`BGSAVE` 执行的过程中，不可以执行 `BGREWRITEAOF`。反过来说，在 `BGREWRITEAOF` 执行的过程中，也不可以执行 `BGSAVE`。这可以防止两个 Redis 后台进程同时对磁盘进行大量的 I/O 操作。

如果 `BGSAVE` 正在执行，并且用户显示地调用 `BGREWRITEAOF` 命令，那么服务器将向用户回复一个 OK 状态，并告知用户，`BGREWRITEAOF` 已经被预定执行：一旦 `BGSAVE` 执行完毕，`BGREWRITEAOF` 就会正式开始。当 Redis 启动时，如果 RDB 持久化和 AOF 持久化都被打开了，那么程序会优先使用 AOF 文件来恢复数据集，因为 AOF 文件所保存的数据通常是最完整的。

## 备份redis数据

在阅读这个小节前，请牢记下面这句话: 确保你的数据由完整的备份. 磁盘故障，节点失效，诸如此类的问题都可能让你的数据消失不见，不进行备份是非常危险的。

Redis 对于数据备份是非常友好的，因为你可以服务器运行的时候对 RDB 文件进行复制：RDB 文件一旦被创建，就不会进行任何修改。当服务器要创建一个新的 RDB 文件时，它先将文件的内容保存在一个临时文件里面，当临时文件写入完毕时，程序才使用 `rename(2)` 原子地用临时文件替换原来的 RDB 文件。

这也就是说，无论何时，复制 RDB 文件都是绝对安全的。

- 创建一个定期任务 (cron job)，每小时将一个 RDB 文件备份到一个文件夹，并且每天将一个 RDB 文件备份到另一个文件夹。

- 确保快照的备份都带有相应的日期和时间信息，每次执行定期任务脚本时，使用 `find` 命令来删除过期的快照：比如说，你可以保留最近 48 小时内的每小时快照，还可以保留最近一两个月的每日快照。
- 至少每天一次，将 RDB 备份到你的数据中心之外，或者至少是备份到你运行 Redis 服务器的物理机器之外。

## 容灾备份

Redis 的容灾备份基本上就是对数据进行备份，并将这些备份传送到多个不同的外部数据中心。容灾备份可以在 Redis 运行并产生快照的主数据中心发生严重的问题时，仍然让数据处于安全状态。因为很多 Redis 用户都是创业者，他们没有大把大把的钱可以浪费，所以下面介绍的都是一些实用又便宜的容灾备份方法：

- Amazon S3，以及其他类似 S3 的服务，是一个构建灾难备份系统的好地方。最简单的方法就是将你的每小时或者每日 RDB 备份加密并传送到 S3。对数据的加密可以通过 `gpg -c` 命令来完成（对称加密模式）。记得把你的密码放到几个不同的、安全的地方去（比如你可以把密码复制给你组织里最重要的人物）。同时使用多个储存服务来保存数据文件，可以提升数据的安全性。
- 传送快照可以使用 SCP 来完成（SSH 的组件）。以下是简单并且安全的传送方法：买一个离你的数据中心非常远的 VPS，装上 SSH，创建一个无口令的 SSH 客户端 key，并将这个 key 添加到 VPS 的 `authorized_keys` 文件中，这样就可以向这个 VPS 传送快照备份文件了。为了达到最好的数据安全性，至少要从两个不同的提供商那里各购买一个 VPS 来进行数据容灾备份。
- 需要注意的是，这类容灾系统如果没有小心地进行处理的话，是很容易失效的。最低限度下，你应该在文件传送完毕之后，检查所传送备份文件的体积和原始快照文件的体积是否相同。如果你使用的是 VPS，那么还可以通过比对文件的 SHA1 校验和来确认文件是否传送完整。

另外，你还需要一个独立的警报系统，让它在负责传送备份文件的传送器（transfer）失灵时通知你。

关于[topics/persistence](#)互动的最新评论

geelou 发布于 2016-8-31 17:55:29

[topics/persistence](#)互动

发表评论

联系Email:[admin@redis.cn](mailto:admin@redis.cn), [redis](#)交流群: 187095642 京ICP备15003959号 站长统计

友情链接: [阿里云](#) [DBA的罗浮宫](#) [VIP-陈群博客](#) [Redis-知识库](#) [Kubernetes](#)