

Redis 从入门到精通

黄健宏(huangz)



RDB 持久化

将数据以二进制文件的形式储存到硬盘里面



因为 Redis 服务器将数据储存在内存里面, 而一旦服务器被关闭、或者运行服务器的主机本身被关闭的话, 储存在内存里面的数据就会消失不见。

如果我们仅仅是将 Redis 用作缓存的话, 那么这种数据丢失带来的问题并不是非常大, 我们只需要重启机器, 然后再次将数据放到缓存里面就可以了; 但如果我们将 Redis 用作数据库的话, 那么这种数据丢失就不能接受了。

为了在 Redis 服务器关闭时, 仍然保留数据库中的数据, Redis 提供了 RDB 和 AOF 两种持久化功能, 这两种功能可以将储存在内存里面的数据库数据以文件的形式保存到硬盘里面, 这样的话, 即使服务器关闭, 已经保存到硬盘里面的数据也不会丢失。

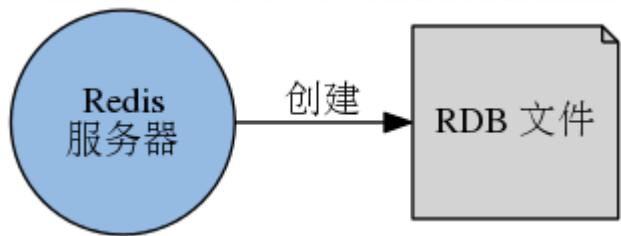
除此之外, 服务器也可以在重新启动时, 通过载入持久化文件来还原服务器在关闭之前的数据库数据。或者使用持久化文件来进行数据备份、数据迁移等工作。

这一课将介绍 RDB 持久化, 而下一课将介绍 AOF 持久化。

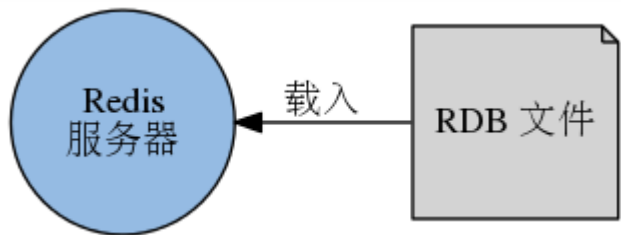


RDB 持久化

RDB 持久化功能可以将服务器包含的所有数据库数据以二进制文件的形式保存到硬盘里面。



而通过在服务器启动时载入 RDB 文件，服务器可以根据 RDB 文件的内容，还原服务器原有的数据库数据。



创建 RDB 文件

那么 Redis 服务器在什么时候才会创建 RDB 文件呢？

在 Redis 服务器创建 RDB 文件的情况中，以下三种是最常见的：

1. 服务器执行客户端发送的 SAVE 命令；
2. 服务器执行客户端发送的 BGSAVE 命令；
3. 使用 save 配置选项设置的自动保存条件被满足，服务器自动执行 BGSAVE 。

在这三种创建 RDB 文件的情况中，前两种情况需要用户手动执行，而第三种情况则是由 Redis 服务器自动执行。

接下来的内容将详细地介绍这三种情况的相同和不同之处。



手动创建 RDB 文件

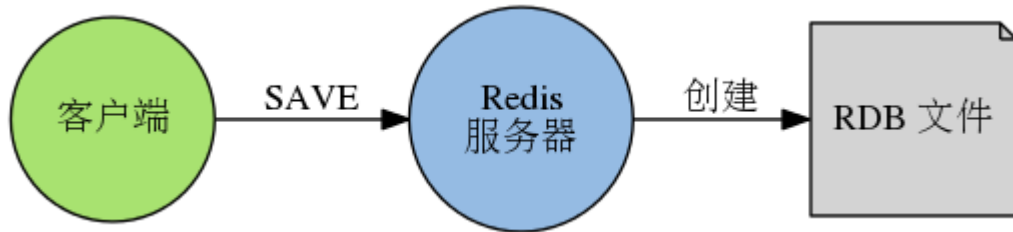
客户端发送 SAVE 或者 BGSAVE



SAVE 命令

通过使用客户端向服务器发送 SAVE 命令, 可以命令服务器去创建一个新的 RDB 文件:

```
redis> SAVE  
OK
```



在执行 SAVE 命令的过程中(也即是创建 RDB 文件的过程中), Redis 服务器将被阻塞, 无法处理客户端发送的命令请求, 只有在 SAVE 命令执行完毕之后(也即是 RDB 文件创建完毕之后), 服务器才会重新开始处理客户端发送的命令请求。

如果 RDB 文件已经存在, 那么服务器将自动使用新的 RDB 文件去代替旧的 RDB 文件。

SAVE 命令的复杂度为 $O(N)$, N 为服务器中, 所有数据库包含的键值对数量总和。



BGSAVE 命令

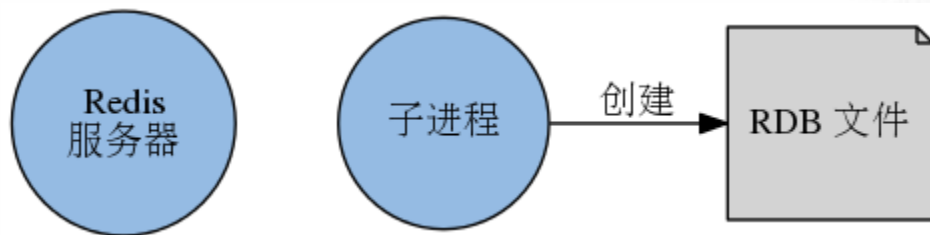
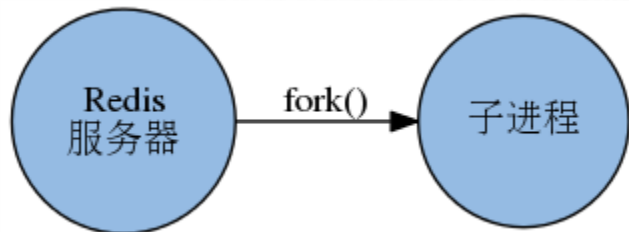
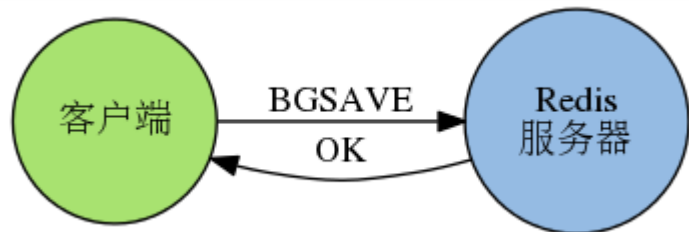
执行 BGSAVE 命令同样可以创建一个新的 RDB 文件, 这个命令和 SAVE 命令的区别在于, **BGSAVE 不会造成 Redis 服务器阻塞**: 在执行 BGSAVE 命令的过程中, Redis 服务器仍然可以正常地处理其他客户端发送的命令请求。

BGSAVE 命令不会造成服务器阻塞的原因在于:

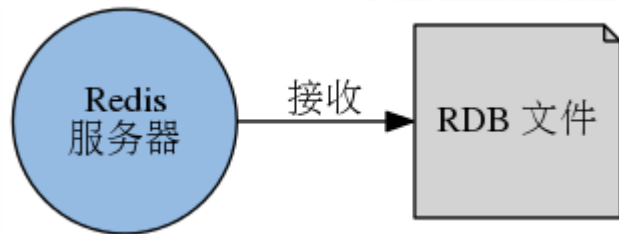
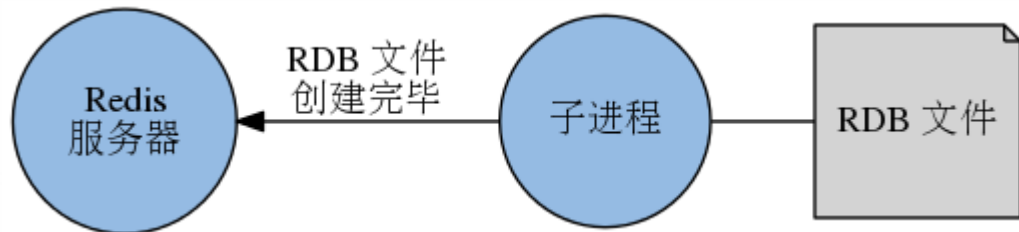
1. 当 Redis 服务器接收到 BGSAVE 命令的时候, 它不会自己来创建 RDB 文件, 而是通过 fork() 来生成一个子进程, 然后由子进程负责创建 RDB 文件, 而自己则继续处理客户端的命令请求;
2. 当子进程创建好 RDB 文件并退出时, 它会向父进程(也即是负责处理命令请求的 Redis 服务器)发送一个信号, 告知它 RDB 文件已经创建完毕;
3. 最后 Redis 服务器(父进程)接收子进程创建的 RDB 文件, BGSAVE 执行完毕。



BGSAVE 命令的执行过程(1/2)



BGSAVE 命令的执行过程(2/2)



BGSAVE 命令

BGSAVE 是一个异步命令，发送命令的客户端会立即得到回复，而实际的操作在回复之后才开始：

```
redis> BGSAVE
```

```
Background saving started
```

BGSAVE 命令的复杂度和 SAVE 一样，都是 $O(N)$ ， N 为所有数据库包含的键值对数量总和。



SAVE 和 BGSAVE 的区别

| 命令 | SAVE | BGSAVE |
|-------------|--|---|
| 类型 | 同步操作 | 异步操作 |
| 是否会导致服务器阻塞？ | 是 | 否 |
| 复杂度 | $O(N)$ | $O(N)$ |
| 优点 | 不需要创建子进程，不会消耗额外的内存，可以集中资源来创建 RDB 文件，所以 SAVE 创建 RDB 文件的速度会比 BGSAVE 快。 | 使得 Redis 服务器可以在创建 RDB 文件的过程中，仍然正常地处理客户端的命令请求。 |
| 缺点 | 执行期间会阻塞服务器。 | 需要创建子进程，会耗费额外的内存。 |

SAVE 和 BGSAVE 没有孰好孰坏之分，需要考虑的是哪个更适合你。



自动创建 RDB 文件

save 配置选项



自动 RDB 持久化

以上提到的 SAVE 和 BGSAVE 都是手动操作，它们都需要用户介入。

为了让 Redis 服务器可以自动进行 RDB 持久化操作，Redis 提供了 save 配置选项，通过这个选项，用户可以设置任意多个保存条件，每当保存条件中的任意一个被满足时，服务器就会自动执行 BGSAVE 命令。

save 选项的格式为：

save <seconds> <changes>

这个选项的效果为：如果距离上一次创建 RDB 文件已经过去了 seconds 秒，并且服务器的所有数据库总共已经发生了不少于 changes 次修改(包括添加、删除和更新)，那么执行 BGSAVE 。



save 选项示例

举个例子，设置：

```
save 300 10
```

表示“如果距离上一次创建 RDB 文件已经过去了 300 秒，并且服务器的所有数据库总共已经发生了不少于 10 次修改，那么执行 BGSAVE 命令”。

而设置：

```
save 60 10000
```

则表示“如果距离上一次创建 RDB 文件已经过去了 60 秒，并且服务器的所有数据库总共已经发生了不少于 10000 次修改，那么执行 BGSAVE 命令”。



save 多选项示例

另外, 用户还可以通过设置多个 save 选项来设置多个自动保存条件, 当任意一个条件被满足时, 服务器就会自动执行 BGSAVE 命令。

举个例子, 对于以下设置来说:

```
save 900 1
```

```
save 300 10
```

```
save 60 10000
```

只要三个条件中的任意一个被 满足时, 服务器就会执行 BGSAVE 。

每次创建 RDB 文件之后, 服务器为实现自动持久化而设置的时间计数器和次数计数器就会被清零, 并重新开始计数, 所以多个保存条件的效果是不会叠加的。



RDB 文件概览

揭开 RDB 文件的庐山真面目



RDB 文件概览

| | | | | | | | |
|---------|--------|-----|-----|-----|-----|-----|---------------------|
| "REDIS" | "0006" | db0 | db1 | ... | dbN | EOF | 6265312314761917404 |
|---------|--------|-----|-----|-----|-----|-----|---------------------|



| | | | | | |
|----------|---|-----|-----|-----|-----|
| SELECTDB | 0 | kv0 | kv1 | ... | kvN |
|----------|---|-----|-----|-----|-----|



| | | | | |
|---------------|---------------|-----------------------|---------|---------|
| EXPIRETIME_MS | 1388556000000 | REDIS_RDB_TYPE_STRING | "hello" | "world" |
|---------------|---------------|-----------------------|---------|---------|



复习

回顾本节学习的内容



Redis 提供了 RDB 持久化和 AOF 持久化这两种持久化功能, 用于将储存在内存里面的数据库数据以文件的形式保存到硬盘里面, 以免数据因为服务器关闭而丢失。

RDB 文件是一个二进制文件, 它保存了 Redis 服务器在创建 RDB 文件时, 所有数据库的数据。

三种最常见的创建 RDB 文件的方法是: 1, 执行 SAVE 命令; 2, 执行 BGSAVE 命令; 3, 使用 save 选项设置保存条件, 让服务器自动执行 BGSAVE。

服务器在执行 SAVE 命令时会被阻塞, 导致无法处理客户端发送的命令请求。

BGSAVE 在执行时不会阻塞服务器, 因为创建 RDB 文件的操作是由子进程执行的, 但这也使得执行 BGSAVE 会比执行 SAVE 耗费更多内存, 并且创建 RDB 文件的速度也会比 SAVE 更慢一些。

用户可以设置多个 save 选项, 当任意一个保存条件被满足时, BGSAVE 就会被执行。

