小象科技 ChinaHadoop.cn

Redis 从入门到精通

黄健宏（huangz）

RDB 持久化的缺点

上一课介绍了 RDB 持久化，这种持久化可以将数据库里面的数据以二进制文件的形式储存到硬盘里面。

RDB 持久化有一个缺点，那就是，因为创建 RDB 文件需要将服务器所有数据库的数据都保存起来，这是一个非常耗费资源和时间的操作，所以服务器需要隔一段时间才创建一个新的 RDB 文件，也即是说，创建 RDB 文件的操作不能执行得过于频繁，否则就会严重地影响服务器的性能。

比如说，在 save 配置选项的默认设置下，即使有超过 10000 次修改操作发生，服务器也至少会间隔一分钟才创建下一个 RDB 文件：

```
save 900 1
save 300 10
save 60 10000
```

如果在等待创建下一个 RDB 文件的过程中，服务器遭遇了意外停机，那么用户将丢失最后一次创建 RDB 文件之后，数据库发生的所有修改。

数据丢失例子

小象科技 ChinaHadoop.cn

时间段	服务器动作
T1	执行命令 cmd1 执行命令 cmd2 …… 执行命令 cmdN
T2	满足持久化条件，自动创建 RDB 文件，并且成功。
T3	执行命令 cmdN+1 执行命令 cmdN+2 … （执行了很多命令，但仍未满足触发 RDB 持久化的条件。）
T4	停机，丢失 T3 时间段产生的所有数据。

解决方法

为了解决 RDB 持久化在遭遇意外停机时丢失大量数据的问题，Redis 提供了 AOF 持久化功能。

比起 RDB 持久化，AOF 持久化有一个巨大的优势，那就是，用户可以根据自己的需要对 AOF 持久化进行调整，让 Redis 在遭遇意外停机时不丢失任何数据，或者只丢失一秒钟数据，这比 RDB 持久化遭遇意外停机时，丢失的数据要少得多。

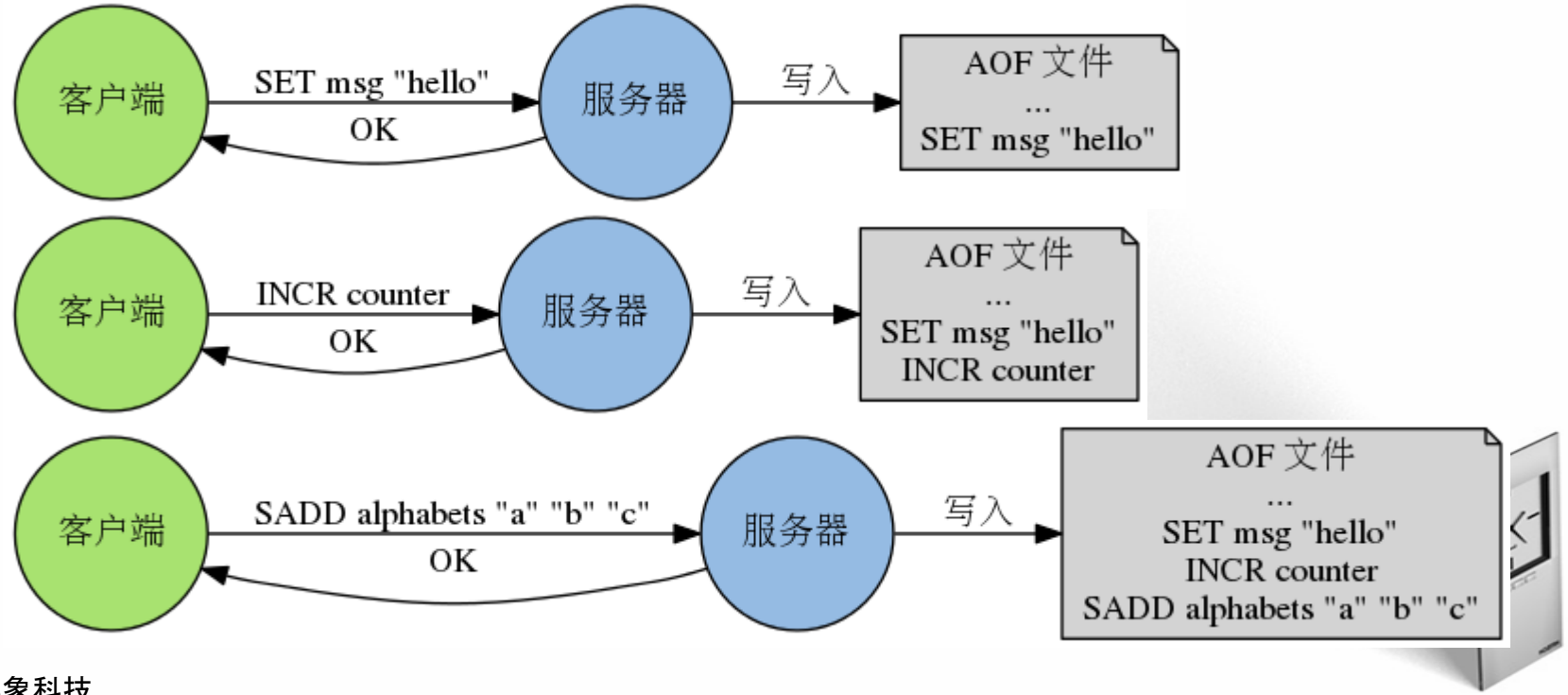
小象科技
ChinaHadoop.cn

AOF 持久化的运作原理

数据保存和数据还原

保存数据

AOF 持久化保存数据库数据的方法是:每当有修改数据库的命令被执行时,服务器就会将被执行的命令写入到 AOF 文件的末尾。



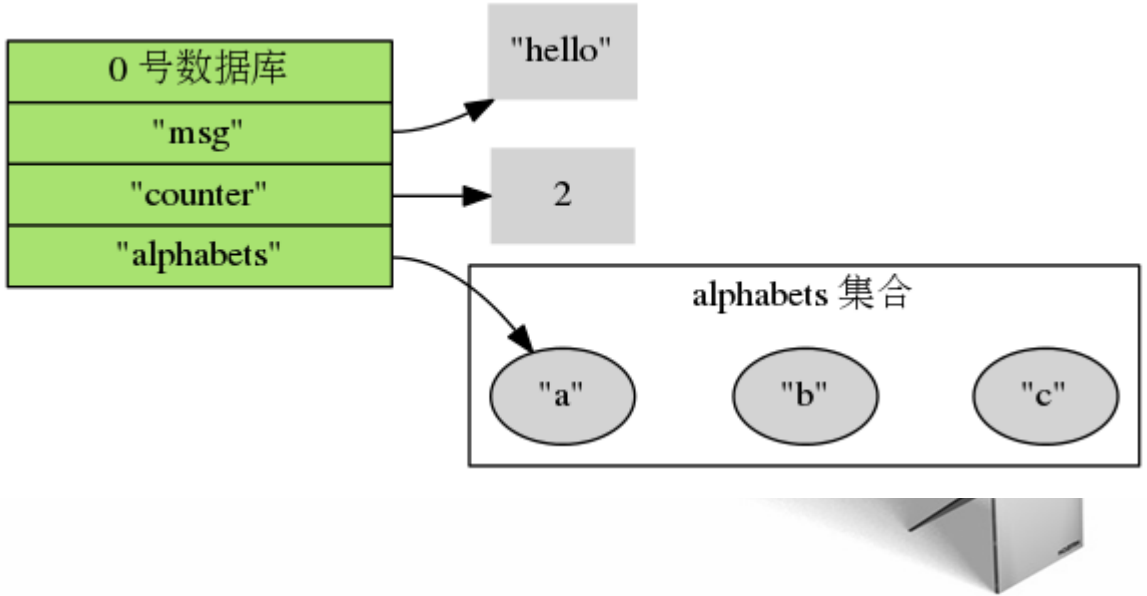
数据还原

因为 AOF 文件里面储存了服务器执行过的所有数据库修改命令，所以给定一个 AOF 文件，服务器只要重新执行一遍 AOF 文件里面包含的所有命令，就可以达到 还原数据库数据的目的。

举个例子，对于包含以下内容的 AOF 文件来说：

SELECT 0
SET msg “hello”
INCR counter
SADD alphabets “a” “b” “c”
INCR counter

服务器只要重新执行这些命令，
就可以还原出右图所示的数据库。



安全性问题

通过配置选项来调整 AOF 持久化的安全性

AOF 持久化的安全性问题

虽然服务器每执行一个修改数据库的命令，就会将被执行的命令写入到 AOF 文件，但这并不意味着 AOF 持久化不会丢失任何数据。

在目前常见的操作系统中，执行系统调用 write 函数，将一些内容写入到某个文件里面时，为了提高效率，系统通常不会直接将内容写入到硬盘里面，而是先将内容放入到一个内存缓冲区（buffer）里面，等到缓冲区被填满，或者用户执行 fsync 调用和 fdatasync 调用时，才将储存在缓冲区里面的内容真正地写入到硬盘里面。

对于 AOF 持久化来说，当一条命令真正地被写入到硬盘里面时，这条命令才不会因为停机而意外丢失。

因此，AOF 持久化在遭遇停机时丢失命令的数量，取决于命令被写入到硬盘的时间：

- 越早将命令写入到硬盘，发生意外停机时丢失的数据就越少；
- 而越迟将命令写入到硬盘，发生意外停机时丢失的数据就越多。

安全性控制

为了控制 Redis 服务器在遇到意外停机时丢失的数据量，Redis 为 AOF 持久化提供了 appendfsync 选项，这个选项的值可以是 always 、everysec 或者 no ，这些值的意思分别为：

always ：服务器每写入一个命令，就调用一次 fdatasync ，将缓冲区里面的命令写入到硬盘里面。在这种模式下，服务器即使遭遇意外停机，也不会丢失任何已经成功执行的命令数据。

everysec ：服务器每秒钟调用一次 fdatasync ，将缓冲区里面的命令写入到硬盘里面。在这种模式下，服务器遭遇意外停机时，最多只丢失一秒钟内执行的命令数据。

no ：服务器不主动调用 fdatasync ，由操作系统决定何时将缓冲区里面的命令写入到硬盘里面。在这种模式下，服务器遭遇意外停机时，丢失命令的数量是不确定的。

运行速度：always 的速度慢， everysec 和 no 都很快。

默认值：everysec 。

停机示例

小象科技 ChinaHadoop.cn

时间	执行的命令
00 秒	SET msg “hello” INCR counter SADD alphabets “a” “b” “c”
01 秒	INCR counter SADD alphabets “d” HMSET profile “name” “peter” “age” 35 SADD alphabets “e”
02 秒	SET date “2014.9.24” HSET profile “phone-number” 123456123 停机

在不同模式下，停机 丢失的数据量是不同的。

AOF 重写

创建一个没有冗余内容的新 AOF 文件

AOF 文件中的冗余命令

随着服务器的不断运行，为了记录数据库发生的变化，服务器会将越来越多的命令写入到 AOF 文件里面，使得 AOF 文件的体积不断地增大。

为了让 AOF 文件的大小控制在合理的范 围，避免它胡乱地增 长，Redis 提供了 AOF 重写功能，通过这个功能，服务器可以产生一个新的 AOF 文件：

- 新 AOF 文件记录的数据库数据和原有 AOF 文件记录的数据库数据完全一样；
- 新的 AOF 文件会使用尽可能少的命令来 记录数据库数据，因此新 AOF 文件的体积通常会比原有 AOF 文件的体积要小得多。
- AOF 重写期间，服务器不会被阻塞，可以正常 处理客户端发送的命令请求。

来看一个 AOF 重写的示例。

AOF 重写示例

小象科技 ChinaHadoop.cn

原有的 AOF 文件	重写产生的新 AOF 文件
SELECT 0 SADD fruits “apple” SADD fruits “banana” SADD fruits “cherry” SADD fruits “apple” INCR counter INCR counter DEL counter SET msg “hello world” SET msg “goodbye” SET msg “hello world again!” RPUSH lst 1 3 5 RPUSH lst 7 9 LPOP lst RPOP lst	SELECT 0 SADD fruits “apple” “banana” “cherry” SET msg “hello world again!” RPUSH lst 3 5 7

AOF 重写的触发

有两种方法可以触发 AOF 重写：

1. 客户端向服务器发送 BGREWRITEAOF 命令。
2. 通过设置配置选项来让服务器自动执行 BGREWRITEAOF 命令，它们分别是：

- auto-aof-rewrite-min-size <size>，触发 AOF 重写所需的最小体积：只有在 AOF 文件的体积大于等于 size 时，服务器才会考虑是否需要进行 AOF 重写。这个选项用于避免对体积过小的 AOF 文件进行重写。
- auto-aof-rewrite-percentage <percent>，指定触发重写所需的 AOF 文件体积百分比：当 AOF 文件的体积大于 auto-aof-rewrite-min-size 指定的体积，并且超过上一次重写之后的 AOF 文件体积的 percent% 时，就会触发 AOF 重写。（如果服务器刚刚启动不久，还没有进行过 AOF 重写，那么使用服务器启动时载入的 AOF 文件的体积来作为基准值。）将这个值设置为 0 表示关闭自动 AOF 重写。

例子：

auto-aof-rewrite-percentage 100

auto-aof-rewrite-min-size 64mb

复习

回顾本节的内容

复习

创建 RDB 文件需要将服务器包含的所有数据全部写入到硬盘里面，这是一个非常耗费资源和时间的操作，因此服务器通常需要每隔一段时间才创建一个新的 RDB 文件，这使得服务器在遭遇意外停机时，可能会丢失大量数据。

AOF 持久化会将每个修改了数据库的命令都写入到 AOF 文件末尾，在启动服务器的时候，只要重新执行 AOF 文件包含的命令，就可以还原服务器原有的数据库数据。

因为写入缓冲区的存在，AOF 持久化的安全性取决于缓冲区里面的命令何时会被真正地写入到硬盘里面，通过设置 appendfsync 配置选项，用户可以让 AOF 持久化不丢失任何已经成功执行的命令数据，或者只丢失一秒钟内被执行的命令数据，又或者不主动执行 fdatasync 调用，将写入硬盘的时机交给操作系统来管理。

随着服务器的运行，AOF 文件会产生越来越多冗余命令，使得文件的体积不断增大，而通过执行 AOF 重写操作，服务器可以创建一个保存相同数据库数据，但不包含任何冗余命令的新 AOF 文件，并使用这个新 AOF 文件来代替原有的 AOF 文件。

持久化对比

小象科技 ChinaHadoop.cn

RDB 持久化	AOF 持久化
全量备份，一次保存整个数据 库。	增量备份，一次保存一个修改数据 库的命令。
保存的间隔较长。	保存的间隔默认为一秒钟。
数据还原速度快。	数据还原速度一般。冗余命令越多，还原速度越慢。
执行 SAVE 命令时会阻塞服务器，但手动或者自动触发的 BGSAVE 都不会阻塞服务器。	无论是平时还是进行 AOF 重写时，都不会阻塞服务器。
更适合数据备份。	更适合用来保存数据，通常意义上的数据持久化。在 appendfsync always 模式下运行时，Redis 的持久化方式和一般的 SQL 数据库的持久化方式是一样的。

好消息:可以同时使用两种持久化，根据你的需求来判断。还原数据优先使用 AOF 文件。