

Redis持久化

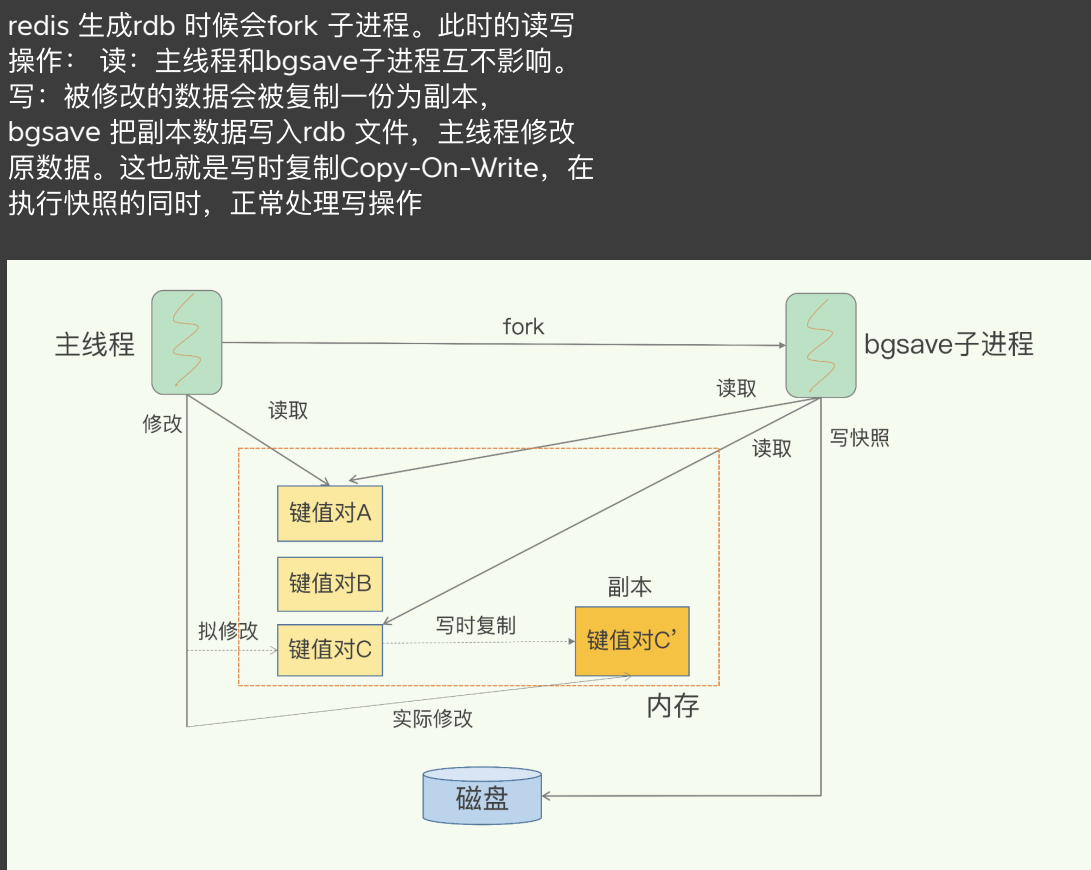
RDB快照

Redis的数据都在内存中。为了提供所有数据的可靠性保证，它执行的是全量快照，也就是说把内存中的所有数据都记录到磁盘中。这类似于给100个人拍合影，把每一个人都拍进照片里。这样做的好处是，一次性记录了所有数据，一个都不少。

给哪些内存数据做快照？

- 两个生成快照的命令
 - save：在主线程中执行，会导致阻塞；
 - bgsave：创建一个子进程，专门用于写入RDB文件，避免了主线程的阻塞，这也是Redis RDB文件生成的默认配置

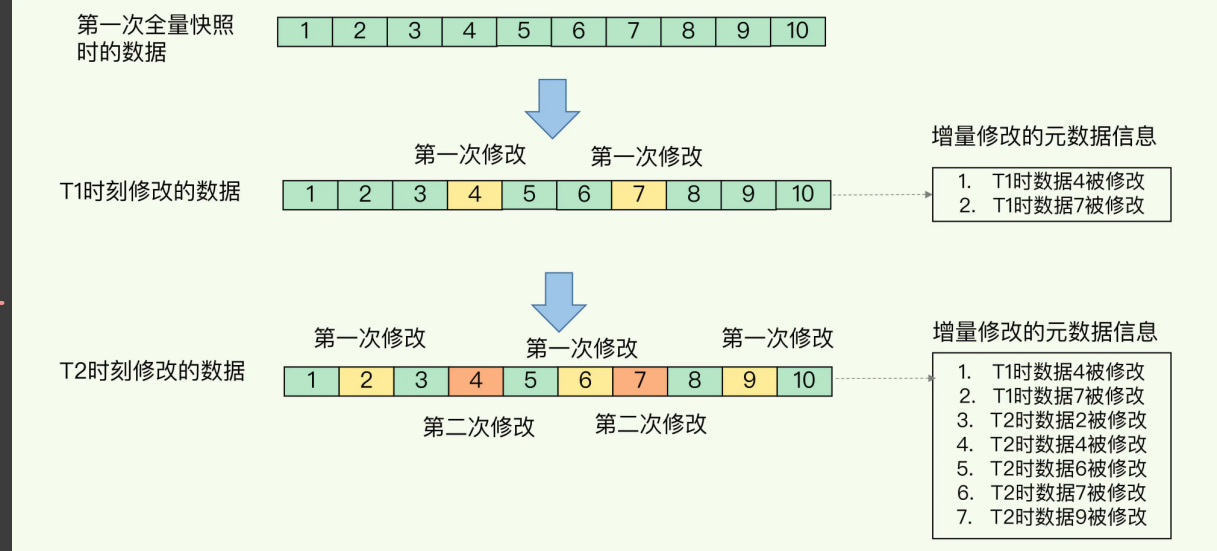
快照时数据能修改吗？



写时复制机制保证快照期间数据可修改

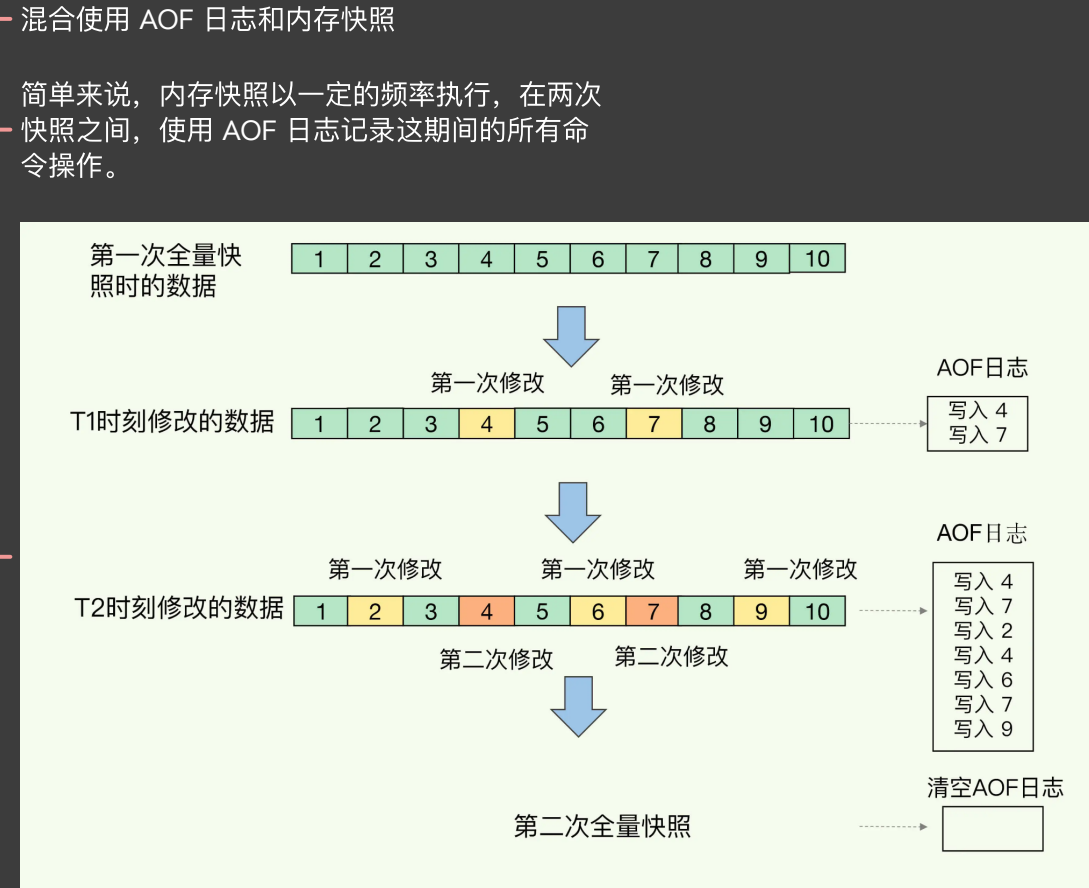
可以每秒做一次快照吗？

- 一方面，频繁将全量数据写入磁盘，会给磁盘带来很大压力，多个快照竞争有限的磁盘带宽，前一个快照还没有做完，后一个又开始了，容易造成恶性循环。
- 另一方面，bgsave子进程需要通过fork操作从主线程创建出来。虽然，子进程在创建后不会再阻塞主线程，但是，fork这个创建过程本身会阻塞主线程，而且主线程的内存越大，阻塞时间越长。如果频繁fork出bgsave子进程，这就会频繁阻塞主线程了（所以，在Redis中如果有一个bgsave在运行，就不会再启动第二个bgsave子进程）



使用增量快照来对全量快照的问题进行优化

虽然跟AOF相比，快照的恢复速度快，但是，快照的频率不好把握。如果频率太低，两次快照间一旦宕机，就可能丢失较多的数据。如果频率太高，又会产生额外开销，那么，还有什么方法既能利用RDB的快速恢复，又能以较小的开销做到尽量少丢数据呢？



AOF和RDB的选择问题，三点建议

- 数据不能丢失时，内存快照和AOF的混合使用是一个很好的选择；
- 如果允许分钟级别的数据丢失，可以只使用RDB；
- 如果只用AOF，优先使用everysec的配置选项，因为它在可靠性和性能之间取了一个平衡。

AOF日志

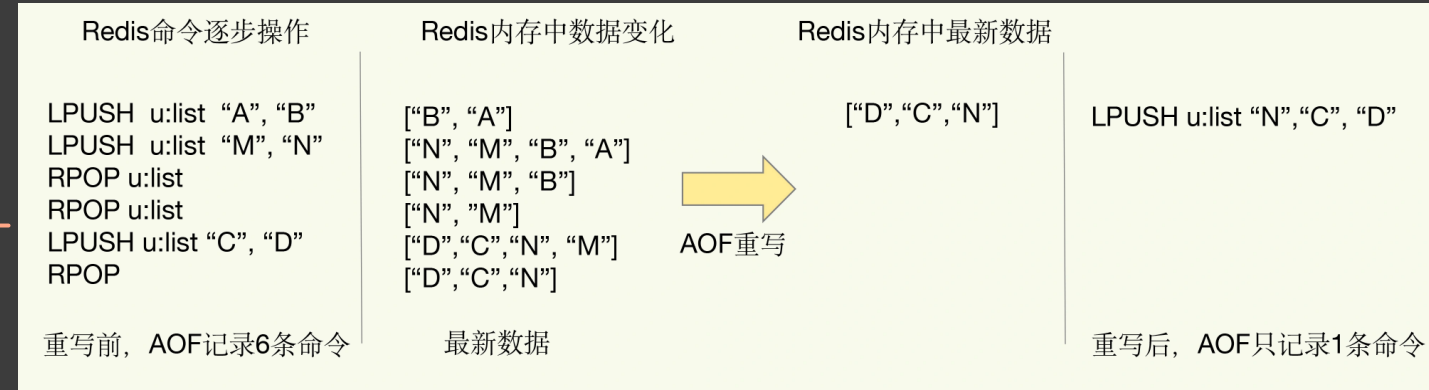
AOF日志写入磁盘是比较影响性能的，为了平衡性能与数据安全，有三种写回策略

- Always，同步写回：每个写命令执行完，立马同步地将日志写回磁盘；
- Everysec，每秒写回：每个写命令执行完，只是先把日志写到AOF文件的内存缓冲区，每隔一秒把缓冲区中的内容写入磁盘；
- No，操作系统控制的写回：每个写命令执行完，只是先把日志写到AOF文件的内存缓冲区，由操作系统决定何时将缓冲区内容写回磁盘。

配置项	写回时机	优点	缺点
Always	同步写回	可靠性高，数据基本不丢失	每个写命令都要落盘，性能影响较大
Everysec	每秒写回	性能适中	宕机时丢失1秒内的数据
No	操作系统控制的写回	性能好	宕机时丢失数据较多

三种写回策略的优缺点

AOF文件是以追加的方式，逐一记录接收到的写命令的。当一个键值对被多条写命令反复修改时，AOF文件会记录相应的多条命令。但是，在重写的时候，是根据这个键值对当前的最新状态，为它生成对应的写入命令。这样一来，一个键值对在重写日志中只用一条命令就行了，而且，在日志恢复时，只用执行这条命令，就可以直接完成这个键值对的写入了。



图例

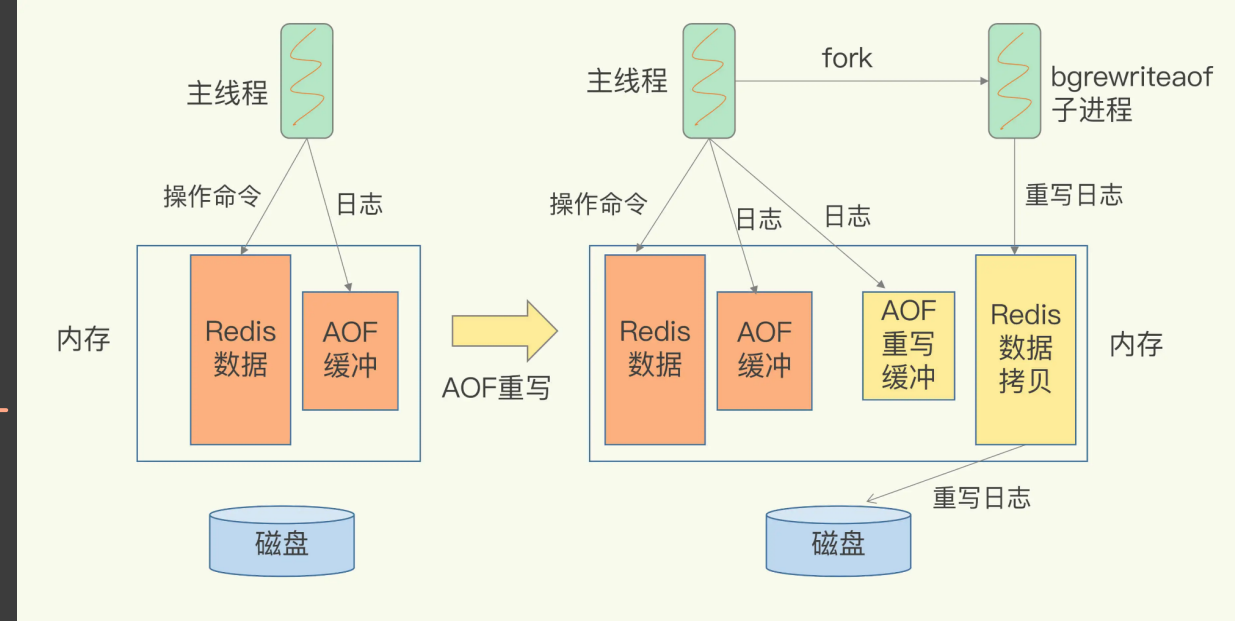
和AOF日志由主线程写回不同，重写过程是由后台子进程bgrewriteaof来完成的，这也是为了避免阻塞主线程，导致数据库性能下降。

AOF重写会阻塞吗？

“一个拷贝”就是指，每次执行重写时，主线程fork出后台的bgrewriteaof子进程。此时，fork会把主线程的内存拷贝一份给bgrewriteaof子进程，这里面就包含了数据库的最新数据。然后，bgrewriteaof子进程就可以在不影响主线程的情况下，逐一把拷贝的数据写成操作，记入重写日志。

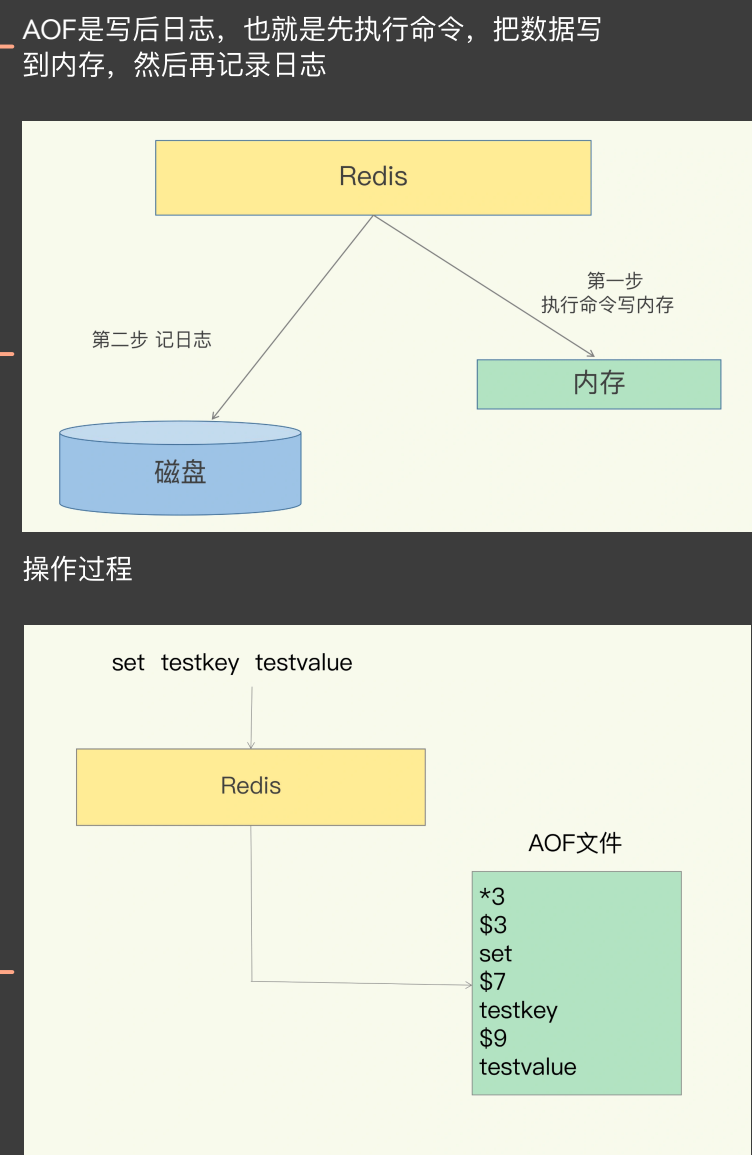
因为主线程未阻塞，仍然可以处理新来的操作。此时，如果有写操作，第一处日志就是指正在使用的AOF日志，Redis会把这个操作写到它的缓冲区。这样一来，即使宕机了，这个AOF日志的操作仍然是齐全的，可以用于恢复。

而第二处日志，就是指新的AOF重写日志。这个操作也会被写到重写日志的缓冲区。这样，重写日志也不会丢失最新的操作。等到拷贝数据的所有操作记录重写完成后，重写日志记录的这些最新操作也会写入新的AOF文件，以保证数据库最新状态的记录。此时，我们就可以用新的AOF文件替代旧文件了。



总结来说，每次AOF重写时，Redis会先执行一个内存拷贝，用于重写；然后，使用两个日志保证在重写过程中，新写入的数据不会丢失。而且，因为Redis采用额外的线程进行数据重写，所以，这个过程并不会阻塞主线程。

AOF日志是如何实现的？



传统数据库的日志，例如redo log（重做日志），记录的是修改后的数据，而AOF里记录的是Redis收到的每一条命令，这些命令是以文本形式保存的。