

Redis主从复制

主从复制是什么

- 主从复制就是一个主数据库，再复制出多个从数据库，其中主数据库负责写，从数据库负责读，这样做能够提高请求的并发量，提高接收请求的吞吐量。
- 其中主节点将自身的数据复制一份给从节点，保持主从节点的数据一致性。

主从复制的作用

- 主从复制可以让主节点专心写，从节点专心读，提高了系统吞吐量。
- 当主节点崩溃时，数据不会全部丢失，因为从节点中有主节点的数据的备份。
- 同时可以增加哨兵节点，如果主节点崩溃，就可以提升从节点作为主节点，提高了系统的可用性。

主从复制的流程

- 先建立链接，从节点通过slaveof的ip地址和端口号向主节点发送tcp链接，主节点接收后双方完成建立链接。
- 然后根据从节点是否是第一次链接主节点，如果是则进行全量复制，否则进行部分复制。
 - 如果首次同步或者断线了很久再同步，此时进行全量复制。复制过程就是
 - 主节点bgsave创建RDB文件
 - 将RDB文件发送给从节点，从节点清空自己的数据，并利用RDB文件进行恢复。
 - 主节点将生成RDB文件期间接受到的命令发送给从节点，从节点进行同步，保持双方的数据一致性。
 - 如果是断线重连，那么进行部分复制
 - 从节点发送offset和runid给主节点。
 - 主节点判断runid是否和自己匹配，并且从节点的offset是否小于主节点的offset，如果匹配，并且小于，那么就进行部分复制。
- 复制完成之后，就持续同步，主节点每接受一次写命令，就将命令同步到从节点，保证主从节点的一致性。

主从复制的核心机制

offset

主节点和从节点都有一个offset。主节点当把写命令写入到复制缓冲区，就会增加offset；从节点接受主节点的写命令，从节点的offset就增加。

主节点和从节点的offset的差值，就是主节点已经写入到复制缓冲区但是还没有同步给从节点的数据。如果没有主从断连太长时间，这个时候就要重新将差值中间的命令重新同步了，否则可能会出现覆盖的情况。

运行id

每一个Redis节点，都有一个运行ID。

当从节点断线重连的时候，会向主节点发送一个之前记录的主节点的runid，如果runid和当前主节点的runid匹配，说明主节点没有重启过，那么直接进行部分复制；如果runid和当前主节点的runid不匹配，说明主节点重启过了，要重新进行全量复制。

复制缓冲区

主节点在同步数据的时候，如果是全量复制，要经过两个步骤。

首先要生成RDB文件，在生成RDB快照的时候，新来的命令会被放到一个复制缓冲区中，当发送完RDB文件后，才会将这个复制缓冲区里面的命令传给从节点。

但是如果是部分复制，那么说明从节点断线了。如果断线时间短，可以直接用复制缓冲区里面的数据进行同步。但是如果断线时间长，因为复制缓冲区内部是一个环形队列，可能最早的数据会被覆盖，就不能部分复制了，只能全量复制。

RDB和AOF在主从节点的角色

在主从复制中，主要应用的是RDB进行全量复制。而部分复制要用到复制缓冲区。AOF即便打开了也不会使用。并且从节点复制主节点的RDB文件时会清空自己的AOF文件。并且如果后续有复制缓冲区的命令同步，这些命令会被拷贝到AOF文件。

部分复制与全量复制的区别？ 什么时候触发全量复制

- 部分与全量的区别：
 - 首先全量复制是先生成一个RDB文件，然后将RDB文件同步给从节点。在生成RDB文件的过程中，会将接收到的新命令加入到复制缓冲区，同步完RDB文件后，会把复制缓冲区的数据再同步给从节点。
 - 但是部分复制只会同步复制缓冲区的命令给从节点。
- 部分与全量触发的时机
 - 部分复制仅当断开连接后主节点没有重启，即从节点保存的上一次runid和当前主节点匹配，并且主节点复制缓冲区没有把从节点的offset之后的命令覆盖。
 - 全量复制则是第一次进行数据同步时，或者断链后主节点重启，或者断连时间很长，主节点的复制缓冲区中把从节点的offset后的数据覆盖了。