Redis 主从复制原理

一、为什么需要主从复制

在我们日常的业务开发中，经常会用到 Redis，假设我们只有部署了一台 Redis 服务器，某个时刻 Redis 服务挂了，造成 Redis 不可用，在此期间，大量的请求将会直接打到数据库，数据库 cpu飙升，严重的可能导致数据库直接挂掉，这就是我们经常说的单点故障。为了解决单点问题，一般都需要对 redis 配置主从节点，那么 redis 主节点和从节点之间如何进行数据同步呢？Redis 提供了主从复制机制，我个人认为它的作用有以下几点：

1）数据冗余 2）单机故障 3）读写分离 4）负载均衡 5）高可用的基石

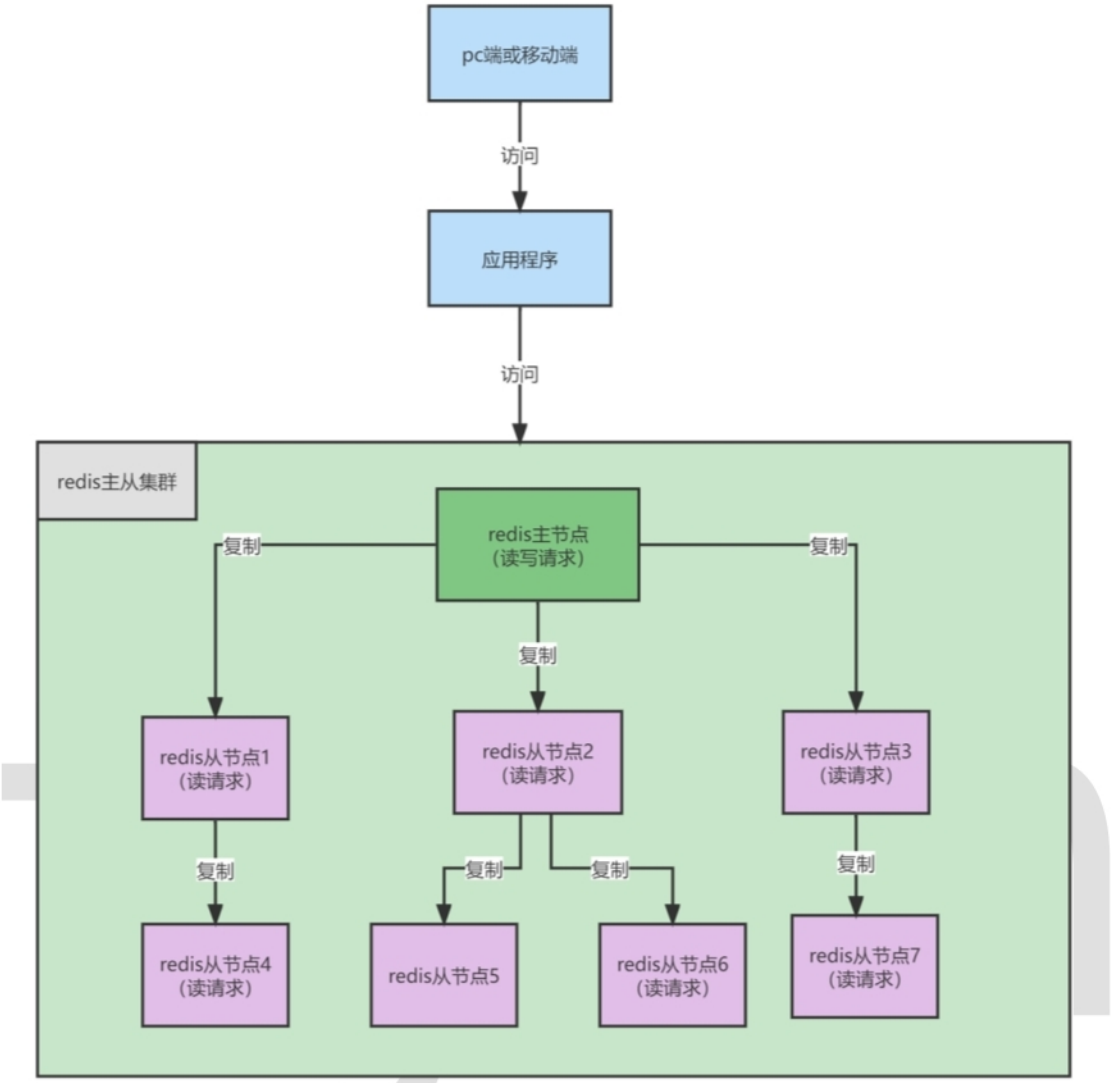
二、什么是 redis 主从复制

主从复制，是指将一台 Redis 服务器的数据，复制到其他的 Redis 服务器。前者称为主节点(master)，后者称为从节点(slave)，数据的复制是单向的，只能由主节点到从节点。一个主节点可以有多个从节点，但一个从节点只能有一个主节点。

主从模式下，redis 采用读写分离模式：

读操作：主节点与从节点均可执行读操作，主要从节点读为主。

写操作： 主节点可以执行写操作，然后把数据同步给各个从节点，保证主从数据一致性。



三、如何搭建 redis 主从模式

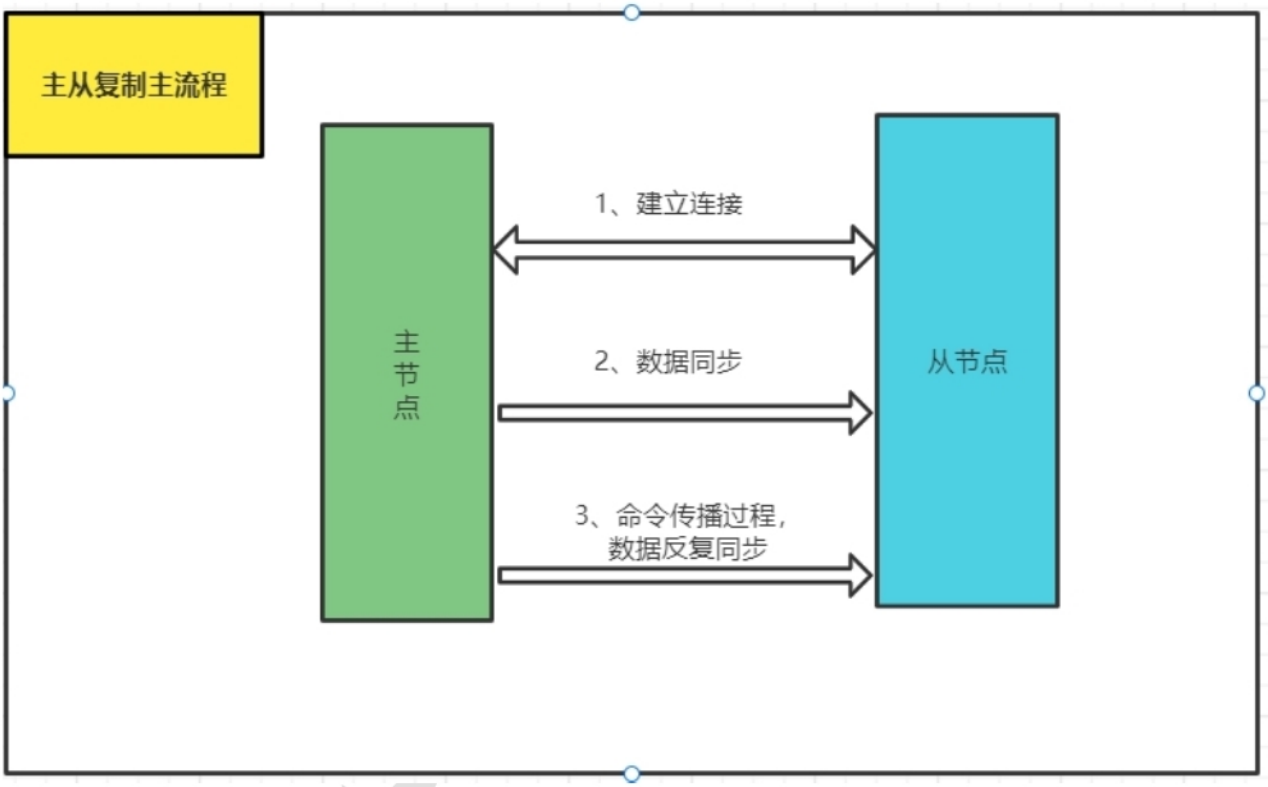
参考文档：https://www.cnblogs.com/cmyxn/p/9414457.html

四、主从复制工作原理

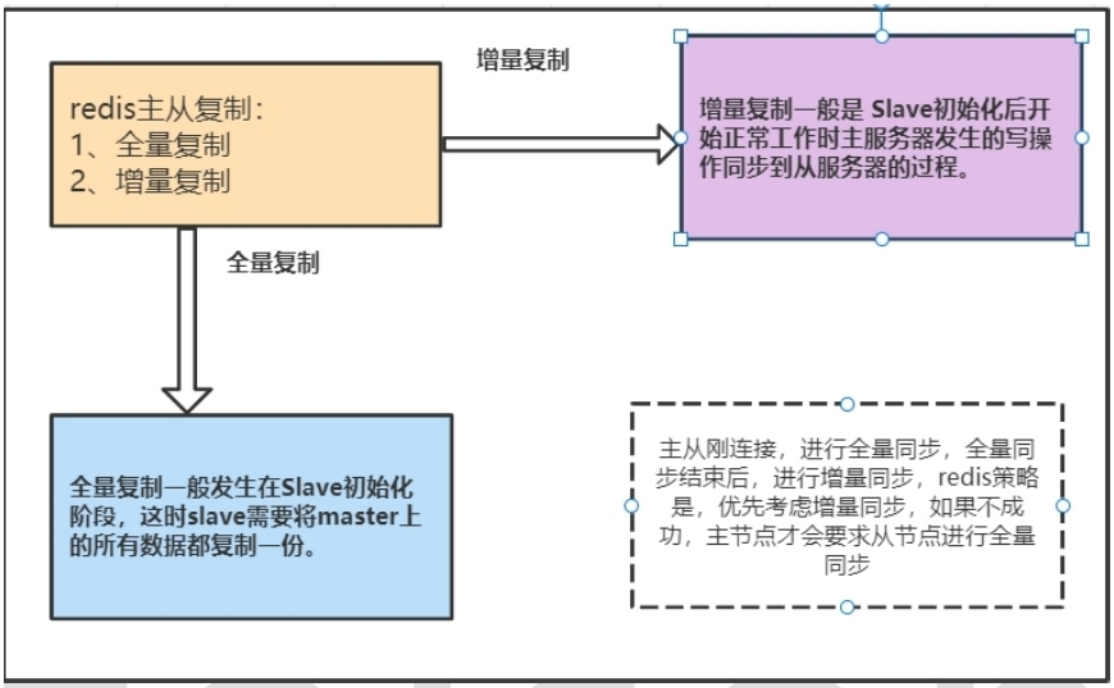
slave 节点初次连接 master 节点，会发送 psync 命令并触发全量复制。此时 master 节点fork 一个后台进程，**开始**生成一份 RDB 快照，同时将那些从外面接收到的写命令缓存到缓冲区中。RDB 文件生成完毕后，将此文件发送给 slave 节点，slave 先写入磁盘，**再**从磁盘加载到内存，接着 master 会将新增加的缓

冲区的写命令发送给 slave，slave 执行写命令并同步数据。如果 slave 节点和 master 节点因网络故障断开连接，会自动重连，连接之后 master 节点会复制缺少的数据给 slave 节点。

1) 主从同步流程



2) 主从同步类型



3) 主从复制名词解释

- runId-----主节点的运行 id

  redis 在启动时会自动生成一个随机的 id（这里需要注意的是每次启动的 id 都会不一样），是由 40 个随机的十六进制字符串组成，用来唯一识别一个 redis 节点。

- offset-------复制偏移量

  复制偏移量是指命令的字节长度，例如：16000，通过对比主从节点的复制偏移量，可以判断主从节点数据是否一致。

- replication buffer--------内部队列缓冲区

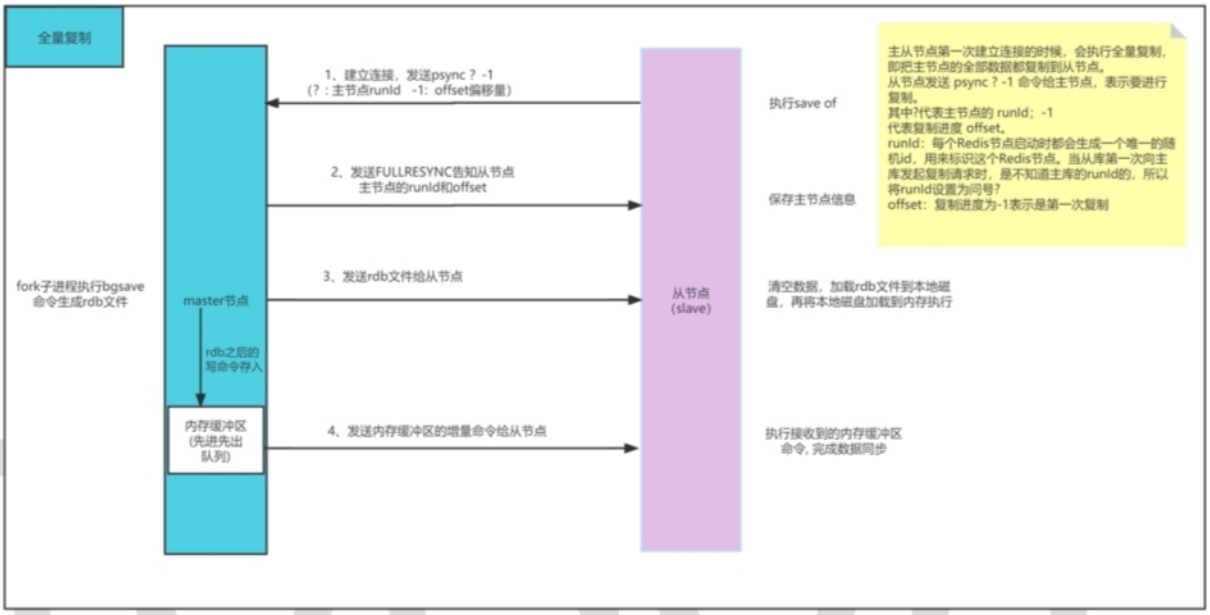
  它是在从节点和主节点建立连接成功后创建的，在主从断开后，这个缓冲区也会被主节点进行删除，主从节点之间复制命令的传输，都会经过这个 buffer，而且这个 buffer 是每个从节点独有的。

- repl_backlog_buffer-------环形缓冲区

  开始进行命令传输之前，就会建立好这个 buffer，这个 buffer 记录当前

master 接收到的新的写操作命令 offset 和命令本身，是所有 slave 公用的 buffer，slave 发送 psync 之后，会和 master 的 offset 进行比较，来决定是否进行增量复制。

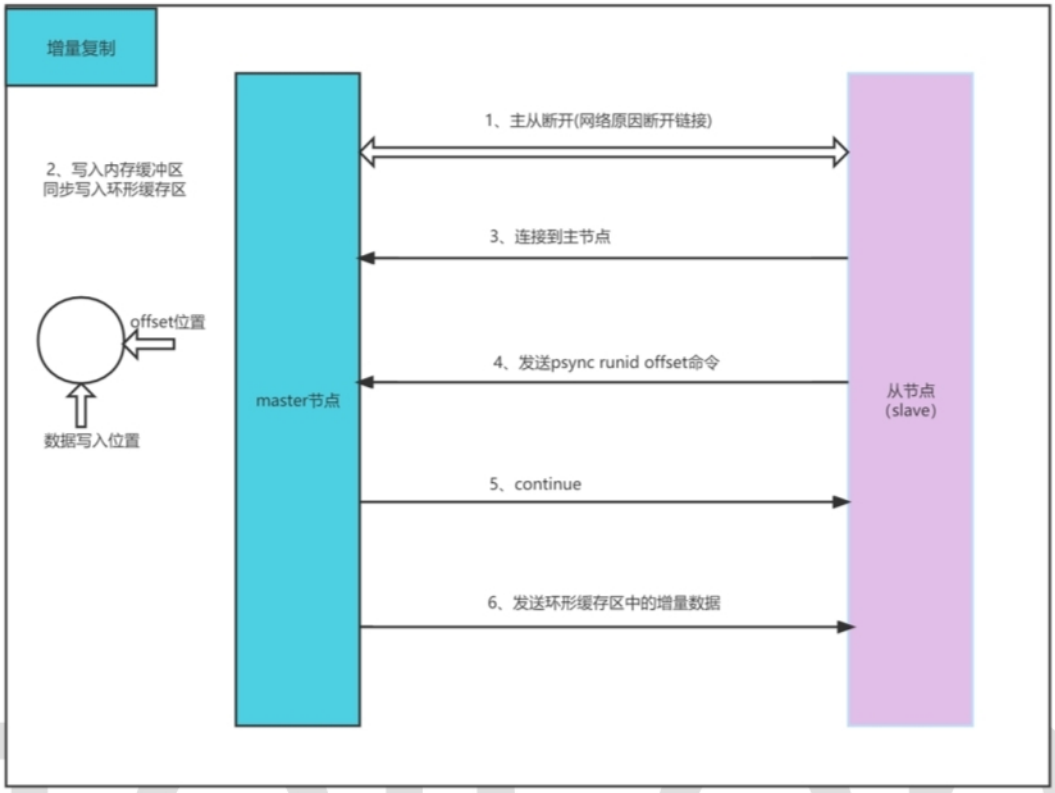
4) 全量复制流程



- 从服务器连接主服务器，发送 psync 命令
- 主服务器接收到 SYNC 命名后，开始执行 BGSAVE 命令生成 RDB 文件并使用 replication buffer 缓冲区记录此后执行的所有写命令
- 主服务器 BGSAVE 执行完后，向所有从服务器发送快照文件，并在发送期间继续记录被执行的写命令
- 从服务器收到快照文件后丢弃所有旧数据，载入收到的快照
- 主服务器快照发送完毕后开始向从服务器发送缓冲区中的写命令
- 从服务器完成对快照的载入，开始接收命令请求，并执行来自主服务器缓冲区的写命令

5) 增量复制流程

从 redis 2.8 开始，如果主从连接因为网络原因断开以后，重新连接之后可以从中断处继续进行复制，而不用全量复制，大大提升了 redis 的性能。



如果主从断开连接了，redis 主节点会根据从节点发送过来的 runId 和从节点的 offset 进行判断是进行全量复制还是增量复制。判断逻辑如下：

增量复制：

如果 runId 和主节点的 id 相同，并且主从的 offset 差距没有超过 repl_backlog_buffer 缓冲区的长度，主节点就会复制 offset 之间的 repl_backlog_buffer 的命令给 slave。

全量复制：

如果 runId 和主节点的 id 不同或者主从的 offset 差距超过 repl_backlog_buffer 缓冲区的长度，则进行全量复制。

五、大厂常见面试题

1) redis 主从节点是长连接还是短连接？

长连接

2) 怎么判断 redis 某个节点是否正常工作

一般集群判断节点是否正常工作，常用的方法都是通过互相的 ping-pong 心跳检测机制，如果有一半以上的节点去 ping 一个节点的时候没 pong 回应，集群就会认为这个节点宕机，会断掉这个节点的连接。

redis 主节点默认每隔 10s 发送一次心跳----判断从节点是否在线。

redis 从节点每隔 1s 发送一次心跳------给主节点发送自己的复制偏移量，从主节点获取到最新的数据变更命令，还做一件事情就是判断主节点是否在线。

3) 过期 key 如何处理

主节点处理了一个 key 或者通过淘汰算法淘汰了一个 key，这个时候主节点模拟一条 del 命令发送给从节点，从节点接收到命令删除 key。

4) redis 是同步复制还是异步复制

redis 主节点每次接收到写命令之后，先写到内部的缓冲区，然后异步发送给从节点。

5) redis 主从切换如何减少数据丢失

- **异步复制同步丢失**

对于 Redis 主节点与从节点之间的数据复制，是异步复制的，当客户端发送写请求给 master 节点的时候，客户端会返回 OK，然后同步到各个 slave 节点中。如果此时 master 还没来得及同步给 slave 节点时发生宕机，那么 master 内存中的数据会丢失。

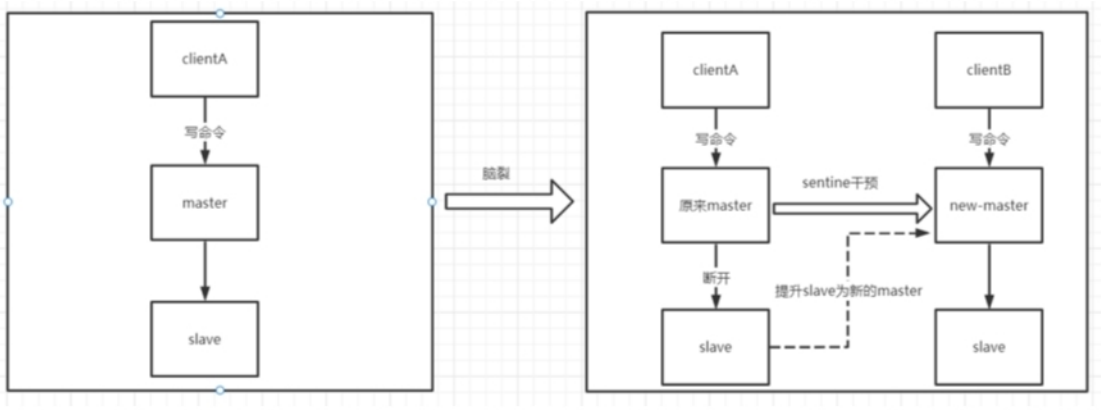
解决方案：

client 端我们可以采取降级措施，将数据暂时写入本地缓存和磁盘中，在一段时间后重新写入 master 来保证数据不丢失。也可以将数据写入 rocketmq 消息队列，发送一个延时消费消息去写入 master。

- **集群产生脑裂数据丢失**

首先我们需要理解集群的脑裂现象，这就好比一个人有两个大脑，那么到底受谁来控制呢？在分布式集群中，zookeeper 很好地解决了这个问题，通过控制半数以上的机器来解决。

那么在 Redis 中，集群脑裂产生数据丢失的现象是怎么样的呢？



解决方案：

在 redis 的配置文件中有两个参数我们可以设置：

min-slaves-to-write 默认是 0，min-slaves-max-lag 默认是 10

```
min-slaves-to-write 2
min-slaves-max-lag 5
```

两个参数表示至少有 2 个 salve 与 master 的同步复制延迟不能超过 5s，一旦所有的 slave 复制和同步的延迟达到了 5s,那么此时 master 就不会接受任何请求。我们可以减小 min-slaves-max-lag 参数的值，这样就可以避免在发生故障时大量的数据丢失，一旦发现延迟超过了该值就不会往 master 中写入数据。

6) redis 主从如何做到故障自动切换

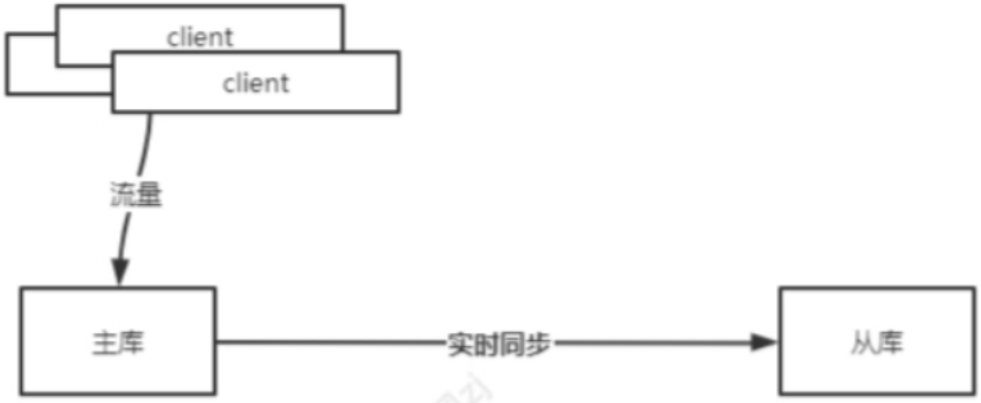
主节点挂了，从节点是无法自动升级成主节点的，这个过程需要人工处理，在此期间，Redis 无法对外提供写操作。此时，Redis 哨兵模式就登场了。

哨兵模式：当主节点出现故障时，由 Redis Sentinel 自动完成故障发现和转移，并通知应用方，实现高可用性。

六、数据备份方式(扩展)

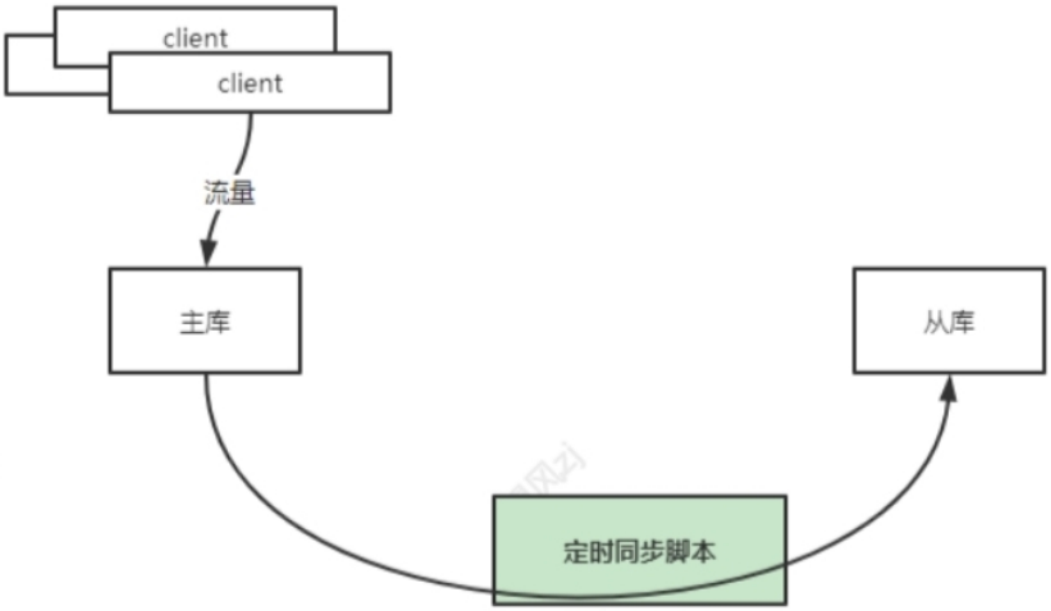
1) 热备

由主库承担业务流量，同时会实时的备份数据到从库。



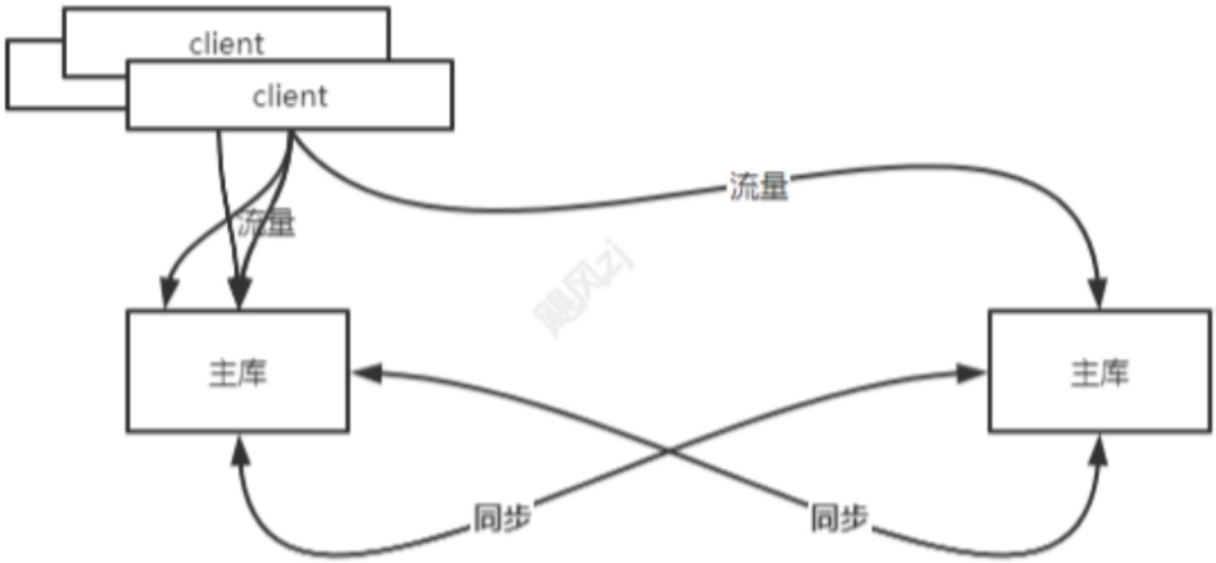
2) 冷备

由主库承担业务流量，通过定时或者离线手动执行脚本备份数据到从库。



3) 多活

由两个数据中心承担业务流量，数据中心互为主备，一般主数据中心会承担大部分流量，备数据中心会承担小部分流量。



那我们思考一个问题，redis 是属于哪种备份呢？

冷备，为了防止 redis 服务器磁盘出现问题，我们定时将 redis rdb 文件按日期备份到阿里云的 oss。