Redis 如何保证高可用

一、高可用背景

大家好，我是 tojson，今天我给大家分享一道高频面试题：redis 如何保证高可用，首先，我们看一下什么是高可用？高可用又叫 HA，我们在设计分布式架构的时候，高可用是一个非常的重要的因素，它通常是指，通过设计减少系统不可用的时间，我们经常谈到的 SLA，比如：3 个 9,4 个 9,5 个 9，都是高可用的指标，分别对应系统不可用的时间，5 个 9 基本是天花板级别，全年不可用的时间为 5.26 分钟，目前能做到的，只有头部的几个大厂。

Redis 作为抗高并发流程的神器，很多系统对它产生了强依赖，所以我们必须保证 redis 的高可用，那如果保证 redis 的高可用呢？常见的有三种方案：

1、主从复制方案 2、redis 哨兵方案 3、redis cluster 集群方案

我们先来看下什么是 redis 的主从复制方案。

二、Redis 主从复制模式

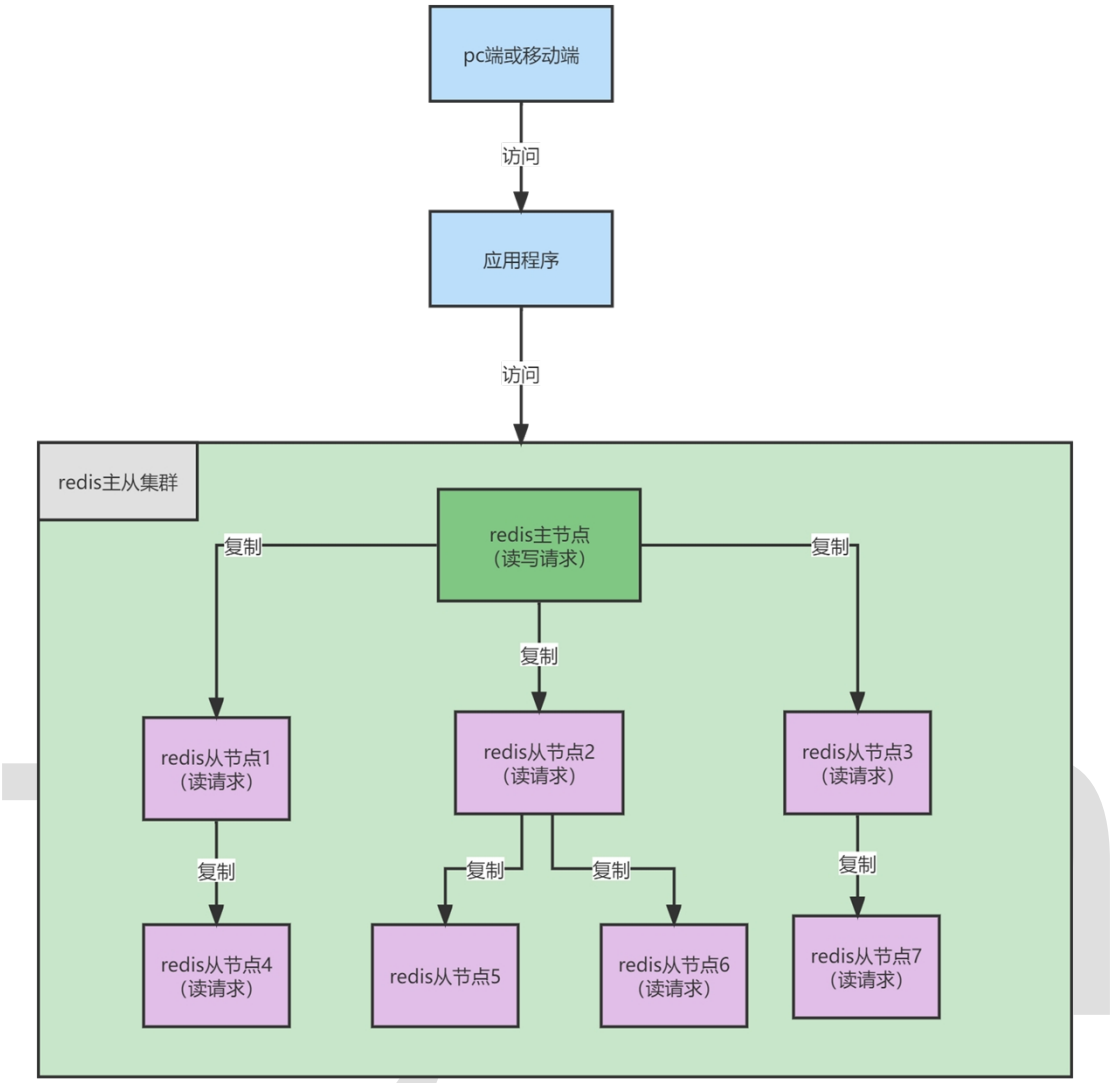
1、原理

Redis 主从复制方案是比较简单的，它有两种类型的节点：

1、master 节点 2、slave 节点

1 个 master 节点可以挂多个 slave 节点，master 节点负责请求的读写，slave 节点负责请求的读，当数据写入 master 节点，通过主从复制将数据同步到 slave 节点，从而实现高可用。

2、架构图



3、缺点

正常情况下，这个方案是没有什么问题的。但很多时候，我们必须考虑各种各样的异常场景的处理，因为这些异常场景往往对系统造成的负面影响是巨大的，那我们思考下有哪些异常的场景呢？我这里想到了两点：

1）master 节点或 slave 节点出现异常了如何实时监控。

2）如果 master 节点出现异常，如何从 slave 节点选一个节点作为新的 master 节点，然后让其它的 slave 节点指向这个新的 master 节点。

针对以上两边，我们如何解决呢？

我们可以写一个监控脚本或者集成业界比较流行的监控工具，例如 zabbix 或者

prometheus，当这些监控工具监听到异常节点，及时发出告警通知，人工收到通知后开始干预，将异常的 master 节点对应的某一个 slave 节点升级为 master 节点，同时，让其它的 slave 节点指向新的 master 节点，另外通知上游的客户端更换新的 master 节点地址。

我们可以看一下这整个过程，节点异常发现和故障自动转移，需要人工参数，并且耗时比较场，在修复的过程中，服务对外是不可用的，这对流量比较大的互联网公司来说影响比较大，那我们有没有一种方案能做到自动发现节点异常和故障自动转移呢？这就是接下来我要介绍的 redis 哨兵方案。

三、Redis 哨兵方案

1、原理

redis 是 2.8 版本之后采用的哨兵方案，它是在主从复制方案的基础上引入了一组哨兵节点来实现高可用的，哨兵节点是独立的进程，独立运行的，它主要有以下几个作用：

1) 监控(它通过向所有的 redis 发送 ping 命令，并等待他们的响应，来判断节点是否运行正常)

2) 告警提醒

当监控的 redis 节点出现问题时，哨兵通过 api 向指定人发送通知 告警

3) 自动故障转移

假如某个 master 节点宕机，sentinelleader 节点先检测到，系统不会马上进行 failover 选出新的 master 节点，而是先标记为主观下线。当其它的哨兵也检测到 master 节点不可用，并且数量达到一定值时，sentinel 领导者就会发起一次投票，然后根据投票结果进行 failover。这个时候这个 master 节点会被标记为客观下线。同时哨兵会选择一个 master 对应的 slave 节点作为新的 master 节点，然后通过发布订阅模式通知其他的 slave 节点指向新的 master 节点。哨兵和集群节点之间通过 gossip 二进制协议进行通信。

slave 被选为 master 节点，需要具备以下优先级：

a. 跟 master 断开连接的时长(时间越短优先级越高)

b. slave 优先级(值越小优先级高)

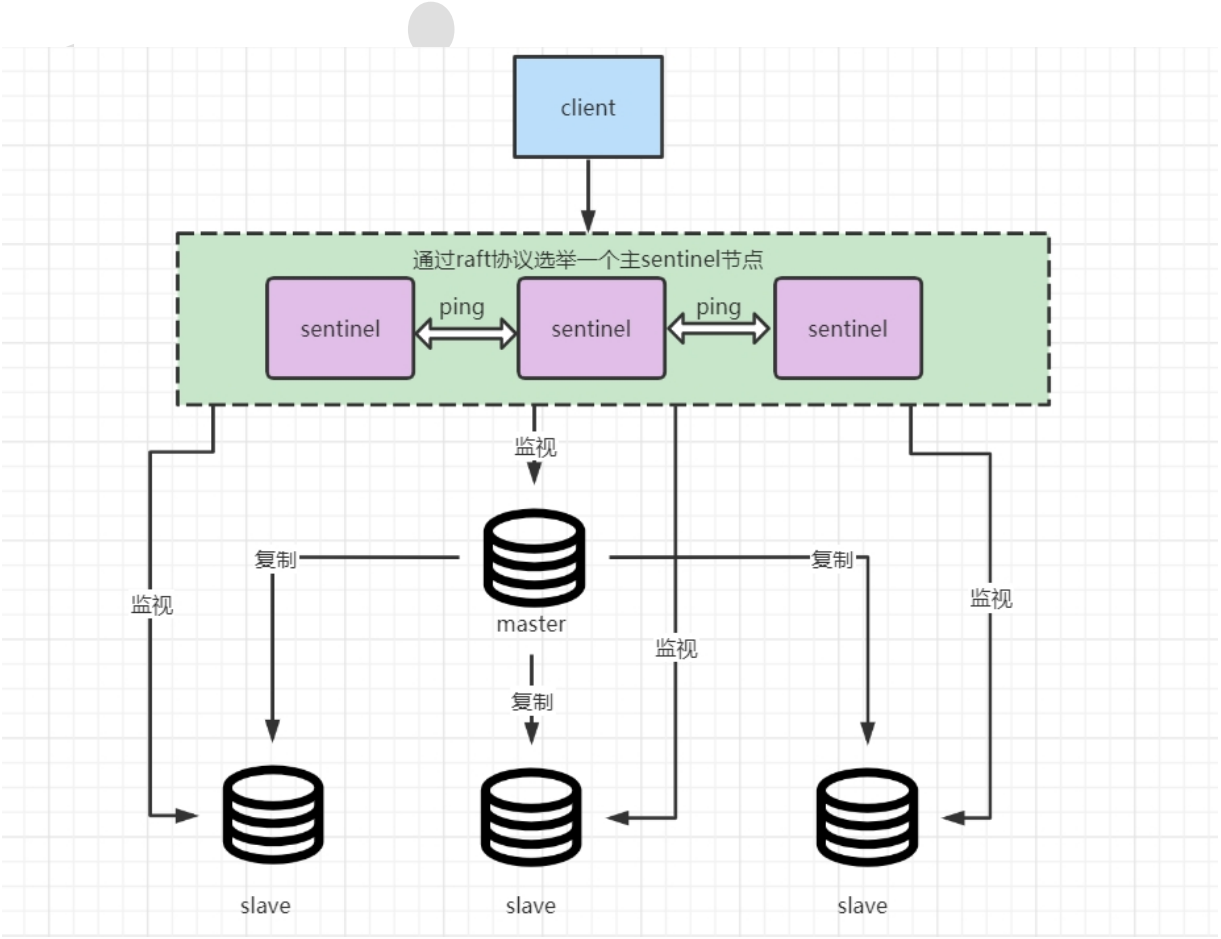
c. 复制 offset(值越大优先级越高)

d. run id(值越小优先级越高)

4) 通知客户端连接新的 master

使用 redis sentinel，客户端不用直接连接 redis 主节点，而且连接 sentinel 获取主节点，类似 nginx 代理，这里有一个问题，如果 sentinel 宕机了，客户端就找不到对应的 redis 节点，所以 sentinel 节点必须实现高可用，sentinel 通过集群部署，节点之间采用 raft 协议保证数据的一致性，通过 ping 的方式进行探活，如此一来，主从服务和 sentinel 服务都实现了高可用。

2、架构图



3、缺点

Redis 哨兵方案已经实现了故障自动转移，无需人工干预，切换时间也非常短，

但是它有一个问题，我们回过头来看一下，这个方案有没有什么问题？

1）内存利用率低

主从节点存储的都是相同的数据，内存利用率很低，如果我们因为业务量暴增，需要进行在线扩容，这个操作变得非常复杂，而且受操作系统的限制，很容易就达到瓶颈。

2）高并发写入支持 QPS 有限

因为只有一个 master 节点可以写，当并发量非常高的时候，master 节点就会成为瓶颈，影响 redis 整体吞吐量。

那我们有什么方案解决这些问题呢？这就是接下来我要介绍的 redis 集群方案。

四、Redis cluster 集群模式

1、原理

Redis 在 3.0 版本以后官方推出了 cluser 集群方案，通过对 redis 数据分片，实现 redis 的分布式存储。cluser 集群采用去中心化的思想，它只有两种类型的节点：master 节点和 slave 节点，一个 master 节点可以挂多个 slave 节点，cluser 集群有多组主从节点，每组主从节点通过集群总线(cluster bus)进行通信，节点之间的通信采用 gossip 二进制协议。master 节点负责请求的读写，slave 节点不参与请求的处理，只作为 master 的备份，下面我们来了解下 redis cluster 运行机制:

1）redis 如何实现数据分片

Redis 通过引入 hash 槽的概念，集群预先分配 16384 个槽 slot，并将槽点分配具体的服务节点，每个节点负责一部分槽点数据的存储，这样数据就分散到多个节点，突破了 redis 单机内存的限制，存储容量大大增加。

2）数据如何进行存取

当接受到客户端的请求，通过对应进行 crc16(key) % 16384 取模运算得到对应的槽，从而将读写操作转发到具体的服务节点，当数据写入对应的 master 节点后，数据会同步到这个 master 的所有 slave 节点。

3）如何实现高可用

采用主从复制模式来保证高可用，一个主节点对应多个从节点，主节点提供存取，从节点从主节点复制数据进行备份，当这个主节点挂掉以后，选取一个从节点充当主节点，从而保证集群节点不会挂掉。节点之间通过 gossip 协议交换信息，每个节点除了存取数据还会维护整个集群的节点信息。

4）集群如何进行扩缩蓉

扩容：从集群添加一组或多组主从节点

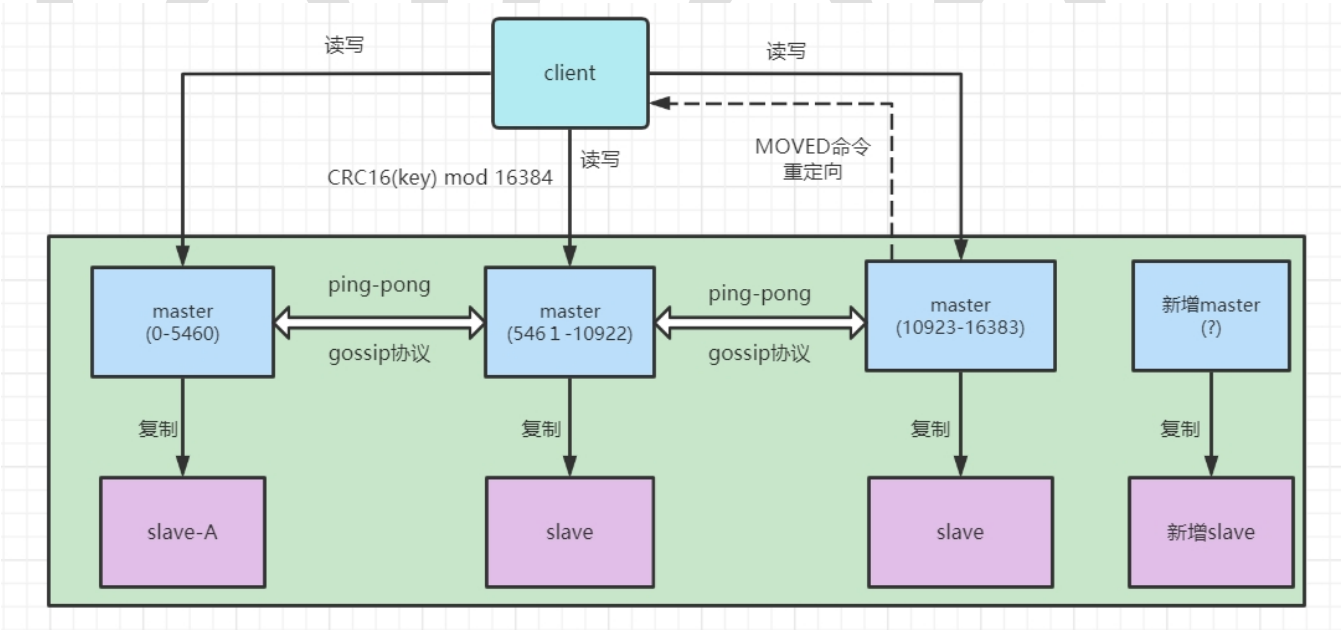
缩容：从集群中删除一组或多组主从节点

Redis cluster 通过集群管理工具 redis-tri.rb 进行扩缩容操作：

扩容:，分三步：1、准备新节点　2、加入集群　3、迁移槽或数据

缩容: 确认下线节点是否迁移数据，如果迁移，借助集群工具处理，如果不迁移，可以直接下线

2、架构图



3、缺点

1）Redis Cluster 是无中心节点的集群架构，依靠 Gossip 协议协同自动化修复集群的状态,在集群节点数量过多的时候，节点之间需要不断进行 ping/pong 通讯，不必须要的流量占用了大量的网络资源。

2) 数据迁移问题

Redis Cluster 可以进行节点的动态扩容缩容，这一过程，在目前实现中，还处于半自动状态，需要人工介入。在扩缩容的时候，需要进行数据迁移。而 Redis 为了保证迁移的一致性，迁移所有操作都是同步操作，执行迁移时，两端的 Redis 均会进入时长不等的阻塞状态，对于小 Key，该时间可以忽略不计，但如果一旦 Key 的内存使用过大，严重的时候会接触发集群内的故障转移，造成不必要的切换。

五、总结

主从模式：

需要手动进行故障转移，耗时长，可用性比较差，基本不推荐

哨兵模式:

通过哨兵自动完成故障转移，但是存储数据比较冗余，利用率低不高，在线扩容难，也不是太推荐

集群模式:

完美继承哨兵模式的所有优点，通过数据分片，实现在线扩容，同时每个 master 节点支持可读可写，可以支持超高的并发，强烈推荐这种方案。

注意：不管使用哪种 redis 高可用方案，都不能保证数据不丢失。 因为三种方案底层都是依赖的主从复制原理，而主从复制是采用的异步复制，而异步复制是肯定会丢数据的，例如：节点宕机

今天 redis 高可用方案就讲到这里，谢谢大家！