1. Sentinel
2. 概述

Sentinel哨兵是Redis的高可用解决方案，由一个或多个Sentinel实例组成的Sentinel系统可以监视任意多个主服务器，以及从服务器。并在被监视的主服务器下线时，自动将属下的某个从服务器升级为新的主服务器，由它来继续处理命令请求。

当主服务器的下线时长超过了用户设定的上限时，Sentinel系统就会对它执行故障转移操作，首先，挑选旗下的一个从服务器升级为主服务器，之后向其他的从服务器发送新的复制指令，让他们成为新的主服务器的从服务器，当所有从服务器开始复制新的主服务器时，故障转移操作完成。另外，Sentinel还会监视已下线的主服务器，在它重新上线时，设置为从服务器。

1. 启动并初始化Sentinel

启动一个Sentinel的步骤如下：

1. 初始化服务器

Sentinel本质上只是一个运行在特殊模式的Redis服务器，因此第一步就是初始化一个普通的Redis服务器，但是它不使用数据库，所以不会载入RDB文件和AOF文件。

1. 使用Sentinel专用代码

将一部分普通Redis服务器使用的代码替换为Sentinel专用的代码，如Sentinel不使用数据库，因此没有set等命令，服务器的命令表也是需要替换的。

1. 初始化Sentinel状态

初始阿虎一个Sentinel状态结构，保存了服务器中所有与Sentinel 相关的状态。

1. 初始化Sentinel状态中的master属性

master字典记录了所有被Sentinel监视的主服务器的相关信息，字典的键是主服务器的名字，值是被监视主服务器对应的sentinelRedisInstance结构，即实例结构，代表一个被监视的Redis服务器实例。该字典的初始化是根据被载入的Sentinel配置文件进行的。

1. 创建连向主服务器的网络连接

最后一步是创建连向被监视主服务器的网络连接，Sentinel将成为主服务器的客户端，可以向主服务器发送命令，也可以从命令回复中获取相关的信息。对于每个被Sentinel监视的主服务器来说，Sentinel会创建两个连向主服务器的异步网络连接，一个是命令连接，用于发送和接受命令，另一个是订阅连接，专用于订阅主服务器的\_\_sentinel\_\_:hello频道。

为什么有两个连接呢？因为目前Redis的发布和订阅功能，被发送的信息都不会保存在Redis服务器中，为了不丢失频道的任何信息，专用一个订阅连接接受该频道的消息。

1. 获取主服务器信息

Sentinel会默认以10秒一次的频率，通过命令连接向主服务器发送INFO命令，并通过回复来获取主服务器信息。通过分析该消息回复，可以获取主服务器本身的信息，如运行ID，服务器角色；还可以获取从服务器的信息，如IP，端口，Sentinel无需用户提供从服务器的地址信息，自动发现从服务器。

根据获取的主服务器信息，对主服务器的实例结构进行更新，从服务器信息也会用于更新主服务器实例结构中的slaves字典，字典的键时Sentinel自动设置的从服务器名字，格式为ip:port，值这是从服务器对应的实例结构。

1. 获取从服务器信息

当Sentinel发现主服务器有新的从服务器出现时，除了会为这个新的从服务器创建对应的实例结构外，还会创建连接到从服务器的命令连接和订阅连接。创建命令连接后，Sentinel以10秒一次的频率向从服务器发送INFO命令，获取回复。根据回复Sentinel可以得知从服务器的运行ID，角色，IP和端口，优先级，复制偏移量，然后更新对应的实例结构。

1. 向主服务器和从服务器发送信息

默认情况下，Sentinel会以两秒一次的频率通过命令连接向所有被监视的主从服务器发送命令，这个命令向服务器的\_\_sentinel\_\_:hello频道发送一条消息，该消息的参数记录的Sentinel和主服务器的IP，端口，运行ID，当前的配置纪元。

1. 接受来自主服务器和从服务器的频道信息

Sentinel与主从服务器建立起订阅连接后，就会订阅这个连接直到连接断开，Sentinel既通过命令连接向该频道发送消息，又通过订阅连接从该频道接受消息。对于监视同一个服务器的Sentinel来说，一个Sentinel发送的信息会被其他Sentinel接收到，这些信息会被用于更新其他Sentinel的结构。

1. Sentinels的自动发现

Sentinel为主服务器创建的实例结构中的sentinels字典保存了其他同样监视该主服务器的Sentinels的资料，键时Sentinel的名字，格式是ip:port，值是该Sentinels对应的实例结构。当目标Sentinel接受到源Sentinel发送来的信息后，会提取出与Sentinel相关的参数和主服务器相关的参数，然后跟新master字典中的主服务器实例结构，以及实例结构中的sentinels字典，也就是说一个Sentinels可以通过接受到频道信息来获知其他Sentinels的存在，并通过发送频道信息让其他Sentinels直到自己的存在，因此Sentinels也是可以自动发现的。

1. 创建连向其他Sentinels的命令连接

当Sentinels通过频道信息发现了一个新的Sentinel时，不仅会创建一个新的实例结构，还会创建一个连向新的Sentinel的命令连接，而新的Sentinel也会创建连向这个Sentinel的命令连接，形成互联的网络，进行信息的交换，但是它们之间不会创建订阅连接，因为订阅连接是用来发现未知的Sentinel的，而他们互相都是已知的。

1. 检测主观下线状态

默认Sentinel以每秒一次的频率向所有与它创建了命令连接的实例发送PING命令，通过返回的PONG命令回复判断实例是否在线。Sentinel配置文件中的down-after-milliseconds选项指定了Sentinel判断实例主观下线的限制时间，如果某实例在该时间内连续向Sentinel返回无效回复，则Sentinel会修改这个实例结构的flags属性，添加SRI\_S\_DOWN标识，表示该实例进行了主观下线状态。

1. 检测客观下线状态

然而监视同一个主服务器的多个Sentinel设置的主观下线时间可能是不同的，其他Sentinel可能认为该服务器没有主观下线。为了确认这个服务器是否真的下线，如果一个Sentinel将一个服务器判断为主观下线后，它会向按其它同样监视这个服务器的Sentinel发送询问，看他们是否也认为该服务器进入了下线状态（主观或客观），如果该Sentinel从其他的Sentinel那里接受到了足够数量的已下线判断，Sentinel就会对该服务器判断为客观下线，并进行故障转移操作。

客观下线的判断条件是Sentinel配置中设置的quorum参数的值，只要包括当前的Sentinel在内的quorum个Sentinel认为该服务器已下线，则当前Sentinel就会将该服务器判断为客观下线。

1. 选举领头Sentinel

当一个主服务器被判断为客观下线后，监视这个服务器的各个Sentinel会进行协商，选举出一个领头Sentinel，对下线服主服务器进行故障转移操作。所有在线的Sentinel都有被选为领头Sentinel的资格，每次进行领头Sentinel选举之后，不论是否成功，所有Sentinel的选举纪元都会+1，它其实就是一个计数器。在一个配置纪元里，所有Sentinel都有一次将某个Sentinel设置为局部领头Sentinel的机会，一旦局部领头设置，在这个配置纪元中就无法更改了。

每个发现主服务器进入客观下线的Sentinel都会要求其他Sentinel将自己设置为局部领头Sentinel，设置局部领头Sentinel的规则是先到先得，目标Sentinel在接收到一条设置命令后，会回复自己的局部领头Sentinel的运行ID和配置纪元。源Sentinel在接收到目标Sentinel发送的的命令回复后，检查回复中的配置纪元和自己是否相同，以及运行ID与自己是否一致，如果都是一致的，说明目标Sentinel将源Sentinel设置为了局部领头Sentinel。

如果有某个Sentinel被半数以上的Sentinel设置为了局部领头Sentinel，那么这个Sentinel成为领头Sentinel，如果在给定时限内，没有一个成为领头的，那么在一段时间后再次选举。

1. 故障转移

领头Sentinel将对已下线的主服务器进行故障转移操作。

1. 选出新的主服务器

挑选一个状态良好，数据完整的从服务器，然后向这个服务器发送 SLAVEOF no one命令，转为为主服务器。选择规则是将从服务器都保存到一个列表汇中，删除已下线，断开连接的，再删除最近没有通信的，再删除保存数据比较老的，最后排序选择优先级较高的。

1. 修改从服务器的复制目标

向从服务器发送SLAVEOF命令来实现。

1. 将已下线主服务器变为从服务器

修改已下线的主服务器的实例结构，当它重新上线时，Sentinel就会向它发送SLAVEOF命令，让他成为从服务器。