牛客大学高薪加成系列课

哨兵的实现原理

牛客大学

- 专业求职辅导 -

1．定期获取拓扑结构

每隔10秒，哨兵节点会向主节点和从节点发送info命令，获取最新的拓扑结构。

- 通过向主节点执行info命令，可以获取从节点的信息，当有新的从节点加入时能立刻感知到；
- 当节点不可达或者故障转移后，可以通过info命令实时更新节点的拓扑结构；

2．定期交换节点信息

每隔2秒，哨兵节点会向数据节点的_sentinel_:hello频道发送它对主节点的判断及自身的信息，同时每个哨兵节点也会订阅该频道。

- 通过订阅数据节点的_sentinel_:hello频道，可以了解其他哨兵节点的信息，并且发现新的哨兵节点；
- 可以用作哨兵节点之间交换主节点的状态，作为后面客观下线以及领导者选举的依据；

3．定期进行心跳检测

每隔1秒，哨兵节点会向主节点、从节点、其他哨兵节点发送一条ping命令，做一次心跳检测。

- 通过该定时任务，哨兵节点与各个节点都建立了连接，实现了对每个节点的监控，是节点失败判定的依据。

- 主观下线
  1. 每隔1秒，哨兵节点对主节点、从节点、其他哨兵节点发送ping命令做心跳检测；
  2. 若超过一定时间（down-after-milliseconds）没有得到回复，当前哨兵就会对这个节点做出失败的判定，这叫主观下线；
- 客观下线
  1. 当哨兵节点主观下线的是主节点时，它就会向其他哨兵节点发送命令，询问它们对主节点的判断；
  2. 当超过quorum个哨兵节点都认为主节点存在问题时，当前哨兵就会对主节点做出客观下线的决定。

Redis使用Raft算法实现领导者选举，其大致思路如下：

1．任意哨兵节点确认主节点客观下线的时候，就会向其他哨兵节点发送命令，要求将自己设置为领导者；

2．收到这项命令的哨兵节点，如果还没有同意其他哨兵节点的投票请求，就将同意本次请求，否则拒绝；

3．若哨兵发现自己的票数已经大于等于max(quorum,num(sentinels/2)+1)，那么它将成为领导者；

4．若此过程没有选举出领导者，则进入下一轮选举。

一旦某个哨兵节点获得了max(quorum,num(sentinels)/2+1)票数，就没必要再去确认其他哨兵节点的票数了，因为每个哨兵节点只有一票。

选举胜出的哨兵节点负责故障转移，具体步骤如下：

1．哨兵领导者会在从节点列表中选出一个从节点作为新的主节点；

2．哨兵领导者会对选出来的从节点执行“replicaof no one”命令，让其成为主节点；

3．哨兵领导者会向剩余的从节点发送命令，让它们去复制新的主节点；

4．哨兵节点集合会将原来的主节点更新为从节点，当它恢复后命令它去复制新的主节点。

从节点的选择方案：

1．先过滤掉不健康的（主观下线、断线）、5秒内没有回复过ping命令的、与主节点失联超过down-after-milliseconds*10秒的从节点；

2．选择replica-priority最高的从节点，若存在则返回，不存在则继续；

3．选择复制偏移量最大的从节点（最完整），若存在则返回，不存在则继续；

4．选择runid最小的从节点。

牛客大学
-专业求职辅导-

THANKS

关注【牛客大学】公众号
回复“牛客大学”获取更多求职资料