(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 114268532 B
(45)授权公告日 2024.08.30

(21)申请号 202111409153.8
(22)申请日 2021.11.24
(65)同一申请的已公布的文献号
申请公布号 CN 114268532 A
(43)申请公布日 2022.04.01
(73)专利权人 华人运通(上海)云计算科技有限公司
地址 201100 上海市闵行区苏召路1628号2幢C075室
(72)发明人 丁磊 陈晨
(74)专利代理机构 广州三环专利商标代理有限公司 44202
专利代理师 麦小婵 郝传鑫
(51)Int.Cl.
H04L 41/00(2022.01)
H04L 43/10(2022.01)
H04L 67/10(2022.01)
(56)对比文件
CN 107105032 A,2017.08.29
审查员 高凯

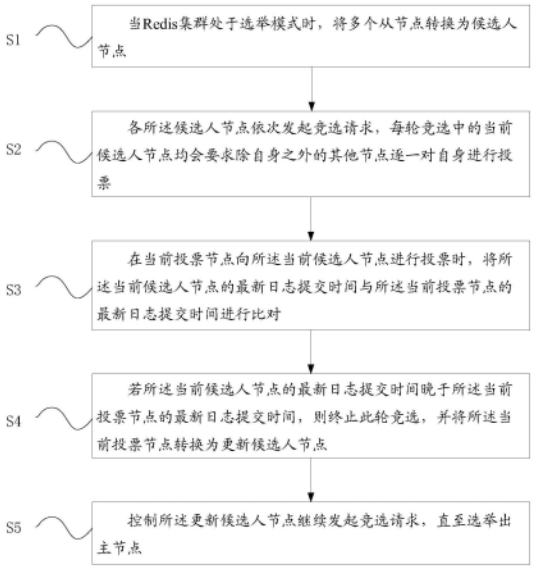
权利要求书2页 说明书6页 附图2页

(54)发明名称

一种基于Raft协议的竞选方法、分布式系统及存储介质

(57)摘要

本发明公开了一种基于Raft协议的竞选方法、分布式系统及存储介质,其中方法包括当Redis集群处于选举模式时,将多个从节点转换为候选人节点;各候选人节点依次发起竞选请求;在当前投票节点向当前候选人节点进行投票时,若当前候选人节点的最新日志提交时间晚于当前投票节点的最新日志提交时间,则终止此轮竞选,将当前投票节点转换为更新候选人节点;控制更新候选人节点继续发起竞选请求,直至选举出主节点。本发明实施例提供的基于Raft协议的竞选方法、分布式系统及存储介质,通过对比最新日志提交时间这一重要参量,提前结束无意义的竞选流程,极大地降低了竞选的数据运算量和运算时间,提高每轮竞选出主节点的概率,保障集群的可用性。



CN 114268532 B

1.一种基于Raft协议的竞选方法,其特征在于,包括:

当Redis集群处于选举模式时,将多个从节点转换为候选人节点;

各所述候选人节点依次发起竞选请求,每轮竞选中的当前候选人节点均会要求除自身之外的其他节点逐一对自身进行投票;

在当前投票节点向所述当前候选人节点进行投票时,将所述当前候选人节点的最新日志提交时间与所述当前投票节点的最新日志提交时间进行比对;

若所述当前候选人节点的最新日志提交时间晚于所述当前投票节点的最新日志提交时间,则终止此轮竞选,并将所述当前投票节点转换为更新候选人节点;

控制所述更新候选人节点继续发起竞选请求,直至选举出主节点;

所述基于Raft协议的竞选方法还包括:

若检测到Redis集群中不存在主节点,则控制Redis集群进入所述选举模式;或,

若当前主节点的任期低于任一从节点的任期,则控制所述当前主节点转换为从节点,并控制Redis集群进入所述选举模式;

所述基于Raft协议的竞选方法还包括:

若Redis集群中存在至少两个更新候选人节点,则各所述更新候选人节点依次生成带时间标志位的特殊竞选请求,其中,所述时间标志位反映更新候选人节点的转换时间;

比较两个所述更新候选人节点的时间标志位,控制时间标志位较晚的所述更新候选人节点向时间标志位较早的所述更新候选人节点进行投票,以及,

将时间标志位较晚的所述更新候选人节点的所有票数转发至时间标志位较早的所述更新候选人节点。

2.如权利要求1所述的基于Raft协议的竞选方法,其特征在于,在所述将多个从节点转换为候选人节点前,所述基于Raft协议的竞选方法还包括:

控制各从节点进入待机状态,其中,所述待机状态持续150ms～300ms。

3.如权利要求1所述的基于Raft协议的竞选方法,其特征在于,若所述当前候选人节点的最新日志提交时间早于所述当前投票节点的最新日志提交时间,则控制任期和日志长度小于所述当前候选人节点的所述当前投票节点向所述当前候选人节点进行投票。

4.如权利要求1所述的基于Raft协议的竞选方法,其特征在于,若当前候选人节点的票数满足下列关系式,则将其选举为主节点:

$$a > \frac{b}{2}$$

其中,a为当前候选人节点的票数,b为Redis集群的节点数量。

5.如权利要求1所述的基于Raft协议的竞选方法,其特征在于,在选举出主节点后,结束所述选举模式,控制Redis集群进入正常模式。

6.一种分布式系统,其特征在于,包括多个节点,所述节点是运行在集群模式下的Redis服务器,且所述节点被分为主节点和从节点;

所述分布式系统用于在所述主节点异常时执行如权利要求1～5中任一项所述的基于Raft协议的竞选方法。

7.如权利要求6所述的分布式系统,其特征在于,所述节点包括N个主节点和N×n个从

节点,任意一个主节点分别对应于n个从节点。

8.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质存储计算机程序,其中,在所述计算机程序运行时控制所述计算机可读存储介质所在设备执行如权利要求1至5中任意一项所述的基于Raft协议的竞选方法。

一种基于Raft协议的竞选方法、分布式系统及存储介质

技术领域

[0001] 本发明涉及分布式集群管理技术领域,尤其是涉及一种基于Raft协议的竞选方法、分布式系统及存储介质。

背景技术

[0002] Raft是一个管理replicated log的算法,在分布式容错系统中有着重要应用。在Redis集群中,任何时刻每个服务器都只会处于主节点、候选人、从节点三种状态中的一种。主节点从客户端接收日志条目,复制给其他服务器,并控制其他服务器将日志条目应用到自己的状态机上;从节点是消极的,他们不会发出请求,只会对来自主节点和候选人的请求作出回应;候选人是集群从服务器中随机选出,用来选举出一个主节点。

[0003] 现如今,在候选人要求其他从节点向自身投票时,会对比候选人和投票节点拥有的任期和日志长度,获得票数的条件是候选人拥有的任期和日志长度大于投票节点拥有的任期和日志长度,以此方式来选举出新的主节点。在一些情况下,选票可能会被瓜分,导致没有主节点产生,这个term将会以没有主节点结束,一个新的term将会很快产生。Raft协议会确保每个term至多有一个主节点,极端情况下,集群将经历多轮投票才能竞选出主节点。

[0004] 由于分布式容错系统会出现网络延迟、分区、丢包、重复和失序等多种复杂的异常状况,会导致出现日志长度小于其他节点,但任期高于其他节点的“缺陷节点”,此类“缺陷节点”若一开始被集群随机确立为候选人,将会发起一轮势必无法产生主节点的竞选,带来毫无意义的时间消耗,提高了数据的计算量,延长了算法的运算时间,大大降低集群的可用性。

发明内容

[0005] 本发明提供一种基于Raft协议的竞选方法、分布式系统及存储介质,以解决现有的Redis集群会因“缺陷节点”导致算法运算时间延长,集群可用性降低的技术问题,通过对比最新日志提交时间这一重要参量,提前结束“缺陷节点”发起的无意义竞选,从而加快了选举主节点的进程,提高了集群的可用性。

[0006] 为了解决上述技术问题,本发明实施例提供了一种基于Raft协议的竞选方法,包括:

[0007] 当Redis集群处于选举模式时,将多个从节点转换为候选人节点;

[0008] 各所述候选人节点依次发起竞选请求,每轮竞选中的当前候选人节点均会要求除自身之外的其他节点逐一对自身进行投票;

[0009] 在当前投票节点向所述当前候选人节点进行投票时,将所述当前候选人节点的最新日志提交时间与所述当前投票节点的最新日志提交时间进行比对;

[0010] 若所述当前候选人节点的最新日志提交时间晚于所述当前投票节点的最新日志提交时间,则终止此轮竞选,并将所述当前投票节点转换为更新候选人节点;

[0011] 控制所述更新候选人节点继续发起竞选请求,直至选举出主节点。

[0012] 作为其中一种优选方案,所述基于Raft协议的竞选方法还包括:

[0013] 若检测到Redis集群中不存在主节点,则控制Redis集群进入所述选举模式;或,

[0014] 若当前主节点的任期低于任一从节点的任期,则控制所述当前主节点转换为从节点,并控制Redis集群进入所述选举模式。

[0015] 作为其中一种优选方案,在所述将多个从节点转换为候选人节点前,所述基于Raft协议的竞选方法还包括:

[0016] 控制各从节点进入待机状态,其中,所述待机状态持续150ms～300ms。

[0017] 作为其中一种优选方案,若所述当前候选人节点的最新日志提交时间早于所述当前投票节点的最新日志提交时间,则控制任期和日志长度小于所述当前候选人节点的所述当前投票节点向所述当前候选人节点进行投票。

[0018] 作为其中一种优选方案,若当前候选人节点的票数满足下列关系式,则将其选举为主节点:

[0019] $$a > \frac{b}{2}$$

[0020] 其中,a为当前候选人节点的票数,b为Redis集群的节点数量。

[0021] 作为其中一种优选方案,所述基于Raft协议的竞选方法还包括:

[0022] 若Redis集群中存在至少两个更新候选人节点,则各所述更新候选人节点依次生成带时间标志位的特殊竞选请求,其中,所述时间标志位反映更新候选人节点的转换时间;

[0023] 比较两个所述更新候选人节点的时间标志位,控制时间标志位较晚的所述更新候选人节点向时间标志位较早的所述更新候选人节点进行投票,以及,

[0024] 将时间标志位较晚的所述更新候选人节点的所有票数转发至时间标志位较早的所述更新候选人节点。

[0025] 作为其中一种优选方案,在选举出主节点后,结束所述选举模式,控制Redis集群进入正常模式。

[0026] 本发明另一实施例提供了一种分布式系统,包括多个节点,所述节点是运行在集群模式下的Redis服务器,且所述节点被分为主节点和从节点;

[0027] 所述分布式系统用于在所述主节点异常时执行如上所述的基于Raft协议的竞选方法。

[0028] 作为其中一种优选方案,所述节点包括N个主节点和N×n个从节点,任意一个主节点分别对应于n个从节点。

[0029] 本发明又一实施例提供了一种计算机可读存储介质,所述计算机可读存储介质存储计算机程序,其中,在所述计算机程序运行时控制所述计算机可读存储介质所在设备执行如上所述的基于Raft协议的竞选方法。

[0030] 相比于现有技术,本发明实施例的有益效果在于以下所述中的至少一点:在Redis集群处于选举模式时,除了要求提供任期和日志长度这两个参量之外,每一轮候选人节点所发起的竞选都会要求候选人节点和投票节点提供自身节点拥有的最新日志提交时间,在投票节点发现自身拥有更新的最新日志提交时间之后,集群会马上结束此轮竞选,并将投票节点转换为更新后的候选人节点继续发起竞选请求,以此方式,剔除某些日志长度小于其他节点,但任期高于其他节点的“缺陷节点”所发起的毫无意义的竞选,极大地降低了竞

选的数据运算量和运算时间,提高每轮竞选出主节点的概率,保障集群的可用性。

附图说明

[0031] 图1是本发明其中一种实施例中的基于Raft协议的竞选方法的流程示意图;

[0032] 图2是本发明其中一种实施例中的特殊竞选环节的流程示意图。

具体实施方式

[0033] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0034] 在本申请描述中,术语“第一”、“第二”、“第三”等仅用于描述目的,而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量。由此,限定有“第一”、“第二”、“第三”等的特征可以明示或者隐含地包括一个或者更多个该特征。在本申请的描述中,除非另有说明,“多个”的含义是两个或两个以上。

[0035] 在本申请的描述中,需要说明的是,除非另有明确的规定和限定,术语“安装”、“相连”、“连接”应做广义理解,例如,可以是固定连接,也可以是可拆卸连接,或一体地连接;可以是机械连接,也可以是电连接;可以是直接相连,也可以通过中间媒介间接相连,可以是两个元件内部的连通。对于本领域的普通技术人员而言,可以具体情况理解上述术语在本申请中的具体含义。

[0036] 在本申请的描述中,需要说明的是,除非另有定义,本发明所使用的所有的技术和科学术语与属于本的技术领域的技术人员通常理解的含义相同。本发明中说明书中所使用的术语只是为了描述具体的实施例的目的,不是旨在于限制本发明,对于本领域的普通技术人员而言,可以具体情况理解上述术语在本申请中的具体含义。

[0037] 本发明一实施例提供了一种基于Raft协议的竞选方法,具体的,请参见图1,图1示出为本发明其中一种实施例中的基于Raft协议的竞选方法的流程示意图,其中包括步骤S1～S5:

[0038] S1、当Redis集群处于选举模式时,将多个从节点转换为候选人节点;

[0039] S2、各所述候选人节点依次发起竞选请求,每轮竞选中的当前候选人节点均会要求除自身之外的其他节点逐一对自身进行投票;

[0040] S3、在当前投票节点向所述当前候选人节点进行投票时,将所述当前候选人节点的最新日志提交时间与所述当前投票节点的最新日志提交时间进行比对;

[0041] S4、若所述当前候选人节点的最新日志提交时间晚于所述当前投票节点的最新日志提交时间,则终止此轮竞选,并将所述当前投票节点转换为更新候选人节点;

[0042] S5、控制所述更新候选人节点继续发起竞选请求,直至选举出主节点。

[0043] Raft协议使用心跳机制来触发leader(主节点)选举,例如,某一服务器启动的时候是处于follower(从节点)状态,当它可以收到来自leader或者candidate(候选人)的有效RPC请求时就会保持follower的状态。Leader发送周期性的心跳(不含日志的AppendEntries RPC)给所有的follower来确保自己的权威,当出现异常时,它就会假定

leader失效并开始新的选举。

[0044] 为了开始新一轮选举,集群会在诸多服务器中随机选取多个从节点,将其状态进行转换,被选取的follower会提高自己当前的term并转为candidate状态。它会先给自己投一票然后并行向集群中的其他服务器发出竞选请求,candidate会保持这个状态,直至自身赢得选举、另一个服务器被确立为leader、或没有获胜者产生。

[0045] 应当说明的是,每一个服务器在给定的term内至多只能投票给一个candidate,先到先得,收到多数节点的选票可以确保在一个term内至多只能有一个leader选出。一旦一个candidate赢得选举,它就会成为leader。它之后会发送心跳消息来建立自己的权威,并阻止新的选举。若candidate没有选举成功或者失败,candidate将会超时并触发新一轮的选举,提高term并发送新的竞选请求,若选票被再次瓜分,则会导致仍然没有服务器被确立为leader的结果。

[0046] 由此可见,极端情况下,集群将经历多轮投票才能选出主节点,每轮无法选出主节点的概率接近50%。为了缩短整个竞选时间,发明人经研究发现,由于服务器状态异常或者网络不稳定的原因,集群中存在日志长度小于其他节点,但任期高于其他节点的“缺陷节点”,这些缺陷节点的竞选过程无疑是毫无意义的,因此为了优化集群竞选主节点的过程,本发明实施例通过比对最新日志提交时间这一重要参量,当发现发起投票请求的候选人节点的最新日志提交时间落后于投票节点的最新日志提交时间时,立即将发起投票请求的候选人节点确立为“缺陷节点”,此时不再逐一延续“缺陷节点”要求其他节点向其投票的竞选流程,而是直接结束此轮竞选流程,并剥夺这一候选人节点的候选人资格,转而将投票节点升级为候选人节点,继续推进后续的竞选流程,以此方式,降低了竞选数据的运算量,提高了集群的可用性。

[0047] 需要说明的是,leader可以决定将新的日志条目放在什么位置,而无需询问其他节点,且数据总是简单的从leader流向其他节点。因此,在Leader异常或者断开连接的情况下,集群需要选举出一个新的leader,在本实施例中,leader异常包括下列两种情况:

[0048] 检测到Redis集群中不存在主节点,或当前主节点的任期低于任一从节点的任期,在这两种情况下,集群会停止日志收发复制等正常工作,进入选举模式,选举出一个新的主节点。

[0049] 进一步地,在上述实施例中,所述基于Raft协议的竞选方法还包括:

[0050] 控制各从节点进入待机状态,其中,所述待机状态持续150ms～300ms之间的任意数值,当然,除上述范围外,待机状态的持续时间可以根据实际的服务器集群性能进行设置,在此不做限定。在待机之后,需要增加每一节点的自由任期,保障集群的时序性,在此不再赘述。

[0051] 进一步地,在本实施例中,在候选人节点要求投票节点向自身投票后,会判断候选人节点是否具有投票资格,若所述当前候选人节点的最新日志提交时间早于所述当前投票节点的最新日志提交时间,则控制任期和日志长度小于所述当前候选人节点的所述当前投票节点向所述当前候选人节点进行投票。

[0052] 本实施例中,当选票的数量达到一定值时,此时立即将其转换为主节点,并结束竞选流程,具体的,若当前候选人节点的票数满足下列关系式,则将其选举为主节点:

[0053] $$a > \frac{b}{2}$$

[0054] 其中,a为当前候选人节点的票数,b为Redis集群的节点数量。

[0055] 具体的,请参见图2,图2示出为本发明其中一种实施例中的特殊竞选环节的流程示意图,在本实施例中,在投票节点发现自身拥有更新的最细日志提交时间后,立即转换为更新候选人节点,并发起竞选投票请求,此时,这一请求在本实施例中被称为“特殊竞选请求”,请求中附带特殊标志位(IrregularVote)表示此次为特殊竞选请求,以及本次发送时间(RequestTime)信息,也即,“特殊竞选请求”是带有时间标志位的请求,其中,所述时间标志位反映更新候选人节点的转换时间。

[0056] 收到“特殊竞选请求”的对象节点包括两大类,一类为普通的从节点,普通的从节点在收到上述“特殊竞选请求”后触发投票机制,投票给候选人,然后等待后续指令;另一类为其他候选人节点,在其他候选人节点收到“特殊竞选请求”后,会判断自身是否也发出过上述“特殊竞选请求”,如未发出过上述“特殊竞选请求”,则触发投票机制,自身投票给候选人,如其他候选人节点也发出过上述“特殊竞选请求”,其也可被称为更新候选人节点,集群会比较两个更新候选人节点的时间标志位,控制时间标志位较晚的所述更新候选人节点向时间标志位较早的所述更新候选人节点进行投票,以及,将时间标志位较晚的所述更新候选人节点的所有票数转发至时间标志位较早的所述更新候选人节点。当然,对整个集群来说,倾向于将更早发出“特殊竞选请求”的节点选举为主节点,因此若收到“特殊竞选请求”的更新候选人节点发现自身之前发出的“特殊竞选请求”的时间标志位较晚,则会忽略当前收到的“特殊竞选请求”,发送/等待主节点的信号,在此不再赘述。

[0057] 由于集群处于选举模式时,正常的工作流程会被中断,因此处于选举模式的集群是不可用的状态,在本实施例中,在选举出主节点后,结束所述选举模式,控制Redis集群进入正常模式,从而优化集群的可用性。

[0058] 本发明另一实施例提供了一种分布式系统,包括多个节点,所述节点是运行在集群模式下的Redis服务器,且所述节点被分为主节点和从节点;

[0059] 所述分布式系统用于在所述主节点异常时执行如上所述的基于Raft协议的竞选方法。

[0060] 在上述实施例中,所述节点包括N个主节点和N×n个从节点,任意一个主节点分别对应于n个从节点。当然,节点的数量和对应关系需要结合实际的分布式系统,考虑灾备、容错、异地/本地机房等网络安全方面的因素,在此不再赘述。

[0061] 相应地,本发明实施例提供一种计算机可读存储介质,所述计算机可读存储介质包括存储的计算机程序,其中,在所述计算机程序运行时控制所述计算机可读存储介质所在设备执行如上述实施例的基于Raft协议的竞选方法中的步骤,例如图1中所述的步骤S1～S5。

[0062] 本发明实施例提供的基于Raft协议的竞选方法、分布式系统及存储介质,有益效果在于以下所述中的至少一点:

[0063] 在Redis集群处于选举模式时,除了要求提供任期和日志长度这两个参量之外,每一轮候选人节点所发起的竞选都会要求候选人节点和投票节点提供自身节点拥有的最新日志提交时间,在投票节点发现自身拥有更新的最新日志提交时间之后,集群会马上结束

此轮竞选,并将投票节点转换为更新后的候选人节点继续发起竞选请求,以此方式,剔除某些日志长度小于其他节点,但任期高于其他节点的“缺陷节点”所发起的毫无意义的竞选,极大地降低了竞选的数据运算量和运算时间,提高每轮竞选出主节点的概率,保障集群的可用性。

[0064] 以上所述是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也视为本发明的保护范围。

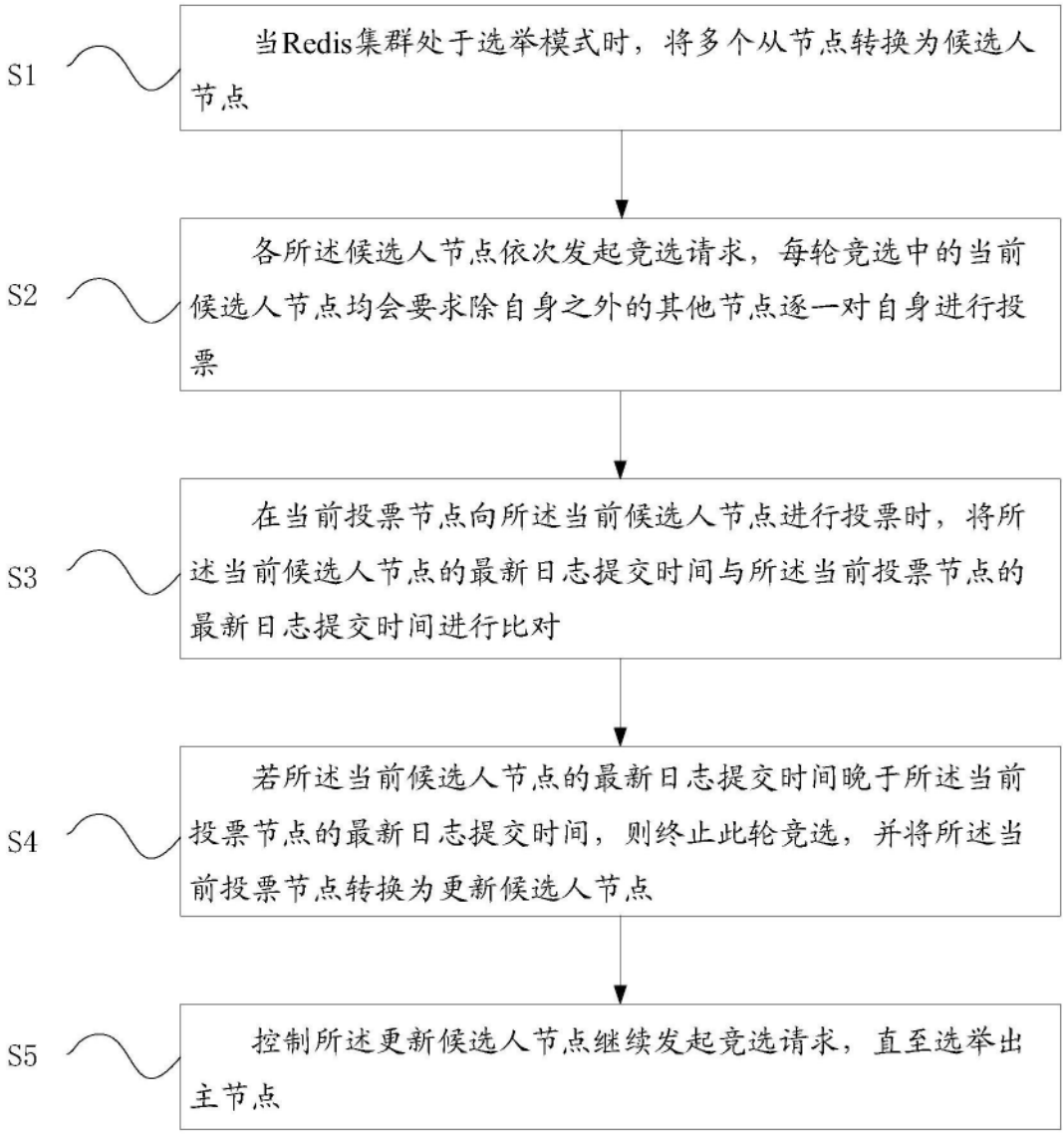

S1
当Redis集群处于选举模式时，将多个从节点转换为候选人节点
S2
各所述候选人节点依次发起竞选请求，每轮竞选中的当前候选人节点均会要求除自身之外的其他节点逐一对自身进行投票
S3
在当前投票节点向所述当前候选人节点进行投票时，将所述当前候选人节点的最新日志提交时间与所述当前投票节点的最新日志提交时间进行比对
S4
若所述当前候选人节点的最新日志提交时间晚于所述当前投票节点的最新日志提交时间，则终止此轮竞选，并将所述当前投票节点转换为更新候选人节点
S5
控制所述更新候选人节点继续发起竞选请求，直至选举出主节点

图1

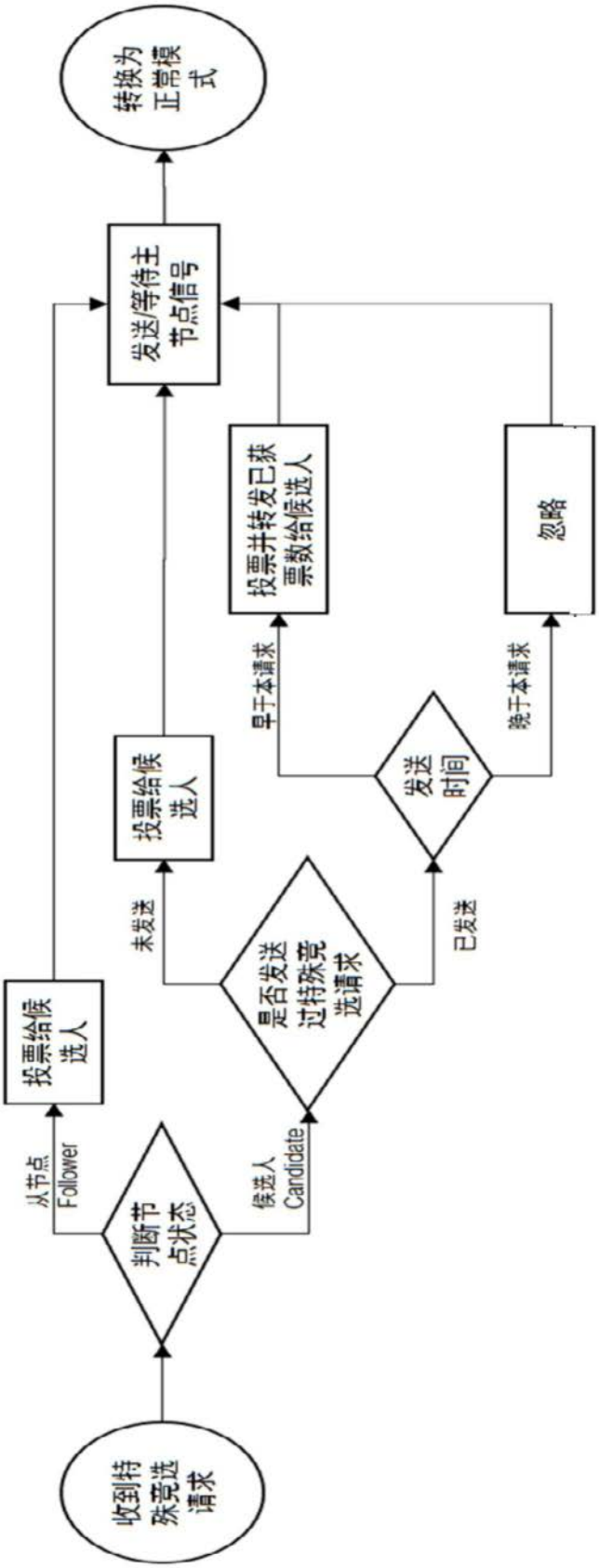

转换为正常模式
发送/等待主节点信号
投票并转发已获票数给候选人
忽略
早于本请求
晚于本请求
发送时间
投票给候选人
未发送
是否发送过特殊竞选请求
已发送
投票给候选人
从节点 Follower
判断节点状态
候选人 Candidate
收到特殊竞选请求

图2