doi:10.3969/j.issn.1672-1292.2023.04.002

基于门限签名及信誉分组的 TRBFT 共识算法

刘金涛,沈丽敏

(南京师范大学计算机与电子信息学院,江苏 南京 210023)

[**摘要**] 基于互联网的快速发展及隐私保护的迫切需求,区块链技术在信息共享及信任领域愈发普及与发展,而共识算法作为区块链的底层关键技术,其效率决定着区块链应用的效果. 自 PBFT 算法提出后,已出现了不少基于该算法的改进方案,但网络实时性差及多节点场景下的各种 PBFT 算法优化方案效果仍不尽人意. 本文基于 Raft 算法进行分层,通过在领导层共识过程中引入 BLS 门限签名,提出一种低延迟、高鲁棒性的拜占庭容错共识算法,并提出一种基于信誉值的动态分组策略,避免了在同一组中出现多数拜占庭节点的情况,提升了方案的安全性,进一步保证了算法的活性. 仿真环境实验测试结果表明,在网络性能差及节点更多的环境下,该算法比 PBFT 算法性能更好且具有更高的扩展性.

[**关键词**] 区块链,共识算法,门限签名,信誉分组,Raft
[**中图分类号**]TP393 [**文献标志码**]A [**文章编号**]1672-1292(2023)04-0010-09

TRBFT Consensus Algorithm Based on Threshold Signature and Reputation Grouping

Liu Jintao, Shen Limin

(School of Computer and Electronic Information, Nanjing Normal University, Nanjing 210023, China)

Abstract: Based on the rapid development of the internet and the growing urgent demand for privacy protection, blockchain technology has become increasingly more popular and developed in the field of information sharing and trust, and the consensus algorithm, as the underlying key technology of blockchain, its efficiency determines the quality of blockchain application. Since the proposal the PBFT algorithm, there have been many improvement schemes based on it, but the effect of various PBFT optimization schemes in poor network real-time and multi-node scenarios is still unsatisfactory. Layering based on Raft algorithm, this paper proposes a low delay and high robustness threshold signature Raft Byzantine fault tolerance (TRBFT) by introducing BLS threshold signature into the leadership consensus process. Meanwhile, a dynamicgrouping strategy based on reputation value is proposed to avoid the situation that most Byzantine nodes appear in the same group, which improves the security of the scheme and further ensures the liveness of the algorithm. Finally, the experimental test results of the simulation environment show that the TRBFT has a better performance and a higher scalability compared with the original PBFT in the environment of poor network performance and more nodes.

Key words: blockchain, consensus algorithm, threshold signature, reputation grouping, Raft

区块链技术是计算机技术、互联网技术与新一代信息技术融合发展形成的新型互联网技术,其本质上是数据和信息技术的集成与融合,是以计算机技术、互联网技术为基础,通过共享交换网络构建起来的一种新的数据处理和存储方式,是信息技术的一种新型应用模式,是一种新的社会经济现象,去中心化、分布式、公开透明和可追溯[1]是其主要特征. 网络技术的发展进步很重要的一点就在于如何克服“信息孤岛”的现象,如何安全高效地共享信息,区块链在此方面做出了很大贡献[2]. 区块链将数据存于互联网中,是通过分布式计算、加密算法、点对点传输、时间戳等技术组成的共识机制,从而保证数据有效且不可篡改. 区块链主要分为 3 种类型:任意节点对任何人开放,可自由参与和退出的公共链;一种由单个机构享有使用权和控制权的私有链;在公共链和私有链之间,参与和退出需要得到联盟授权的联盟链.

收稿日期:2023-03-26.
基金项目:国家自然科学基金青年基金项目(61802195).
通讯作者:沈丽敏,副教授,研究方向:网络安全. E-mail:shenlimin@njnu.edu.cn

联盟链是一个面向全球的区块链网络,旨在建立和维护一个去中心化、开源、自治的分布式区块链系统. 联盟链具有安全性高、可扩展性强等特点,可满足不同用户和业务场景的需求. 以联盟链为基础架构建立起的底层区块链生态不仅提供了数据共享、交易验证等基础功能,还提供了区块信息和数据服务. 在各行业广泛部署和应用联盟链,将加速数字经济发展. 联盟链通过与其他区块链网络互联互通,建立了统一的数据交换标准,在提升用户体验的同时,通过开放共享数据源和服务等方式提升产业数字化水平,基于联盟链建立数字资产交易平台,实现资产交易流通的闭环. 联盟链目前在数据存储[3]、版权管理[4]、隐私保护[5]、数据溯源[6]等方面已有众多的应用. 联盟链中应用最广泛的通用平台是 Hyperledger Fabric,其为一个分布式网络,是完全去中心化的,且可同时运行多个任务[7]. Hyperledger Fabric 使用一个共享数据库来存储分布式数据,该数据库包含了与用户网络上运行的程序相关的所有数据;同时使用“无服务器”设计,允许每个节点均可完全独立地运行任务且不会影响到其他节点.

共识算法作为区块链中关键的底层技术,不少学者对其提出了不同的优化策略,主要分为以下 5 种[8]:策略 1,优化共识轮次,如巫史政[9]提出新增子轮优化视图更换复杂度,并通过结合阈值签名、引入乐观响应等方式来进一步优化共识机制;策略 2,结合可信硬件,如 A2M-PBFT[10]选择使用可信硬件 TPM 来实现一个单调计数器的功能,可将一个拜占庭节点转化为一个非拜占庭节点;策略 3,投机与乐观,如 Obft[11]、CheaPBFT[12]和 FastBFT[13]等,这类算法认为系统运行在较为理想的环境中先进行简单共识,只有出现错误时才会触发其他节点参与共识;策略 4,选用更高效签名算法,在共识过程中需要收集到足够数量的非拜占庭节点信息才能确保共识,门限签名算法完美地契合了该过程所需的安全性及高效性,如 Sbft[14]和 HotStuff[15]等都通过引入门限签名将系统的通信复杂度降至 $O(n)$;策略 5,选举领导者,因为共识过程需要节点间信息交互,所以参与共识的节点数量对系统复杂度有直接的影响,选取出领导者参与共识再将共识结果通知给从节点将大大减少系统的通信复杂度.

本文结合策略 4 和策略 5,基于 Raft 算法进行分层,通过在领导层共识过程中引入 BLS 门限签名,提出一种更高效的拜占庭容错共识算法(threshold signature Raft Byzantine fault tolerance,TRBFT). 通过对节点添加信誉值属性进行动态分组,提出基于信誉值的节点动态分组策略和监督节点的监督策略,以增加组内安全性. 通过对 TRBFT 算法的安全性、活性、可用性、通信开销进行分析,以证明 TRBFT 算法的有效性,最后通过仿真实验对 TRBFT 算法进行评估,并与 PBFT 算法在单次共识通信次数、共识时延、吞吐量等方面进行比较分析.

1 背景知识

1.1 PBFT 共识算法

Castro 和 Liskov[16]在 1999 年首次提出了 PBFT 算法. PBFT 算法是解决拜占庭问题的一种更实用的算法,由于其高效性和可实现性常常被应用于支持拜占庭容错的分布式系统中. 储劲松等[17]通过性能建模验证了 PBFT 算法满足多节点区块链网络的性能需求. 假设系统内参与共识节点总数量为 N,则 PBFT 算法能够满足拜占庭节点(f 表示拜占庭节点数量)不超过总节点数三分之一情况下的共识,即 $f<=(N-1)/3$. 当今主流的 3 种区块链中,PBFT 算法由于其容错特性常被用于联盟链中节点的共识. PBFT 共识算法中的节点一共存在客户端、主节点和从节点 3 种身份,以及主从节点共识和视图更换 2 个重要流程.

主从节点共识流程主要是为了对发来的请求进行共识之后使得节点统一接收,共识过程主要分为请求、预准备、准备、提交及回复 5 个阶段. 客户端向主节点发送区块共识请求,主节点接收到客户端请求后将预准备消息对所有从节点进行广播. 从节点接收到主节点发送的预准备消息后,首先会对该消息进行验证,并在验证通过后将准备消息发送给所有参与共识的节点(包括主节点). 当一个从节点(包括其自身)接收到至少 $2f+1$ 条准备消息时,该节点将进入提交阶段并广播提交消息. 此时,节点将收集所有提交消息,在收到来自不少于 $2f+1$ 个不同节点的提交消息后,节点将认为共识过程成功结束,并向客户端发送回复消息. PBFT 算法的共识过程如图 1 所示.

视图替换流程是为了确保当主节点出现故障或是成为共识过程中的拜占庭节点时,算法可以继续运行. 根据协议定制的编号依次选择主节点,若触发视图替换协议,则根据编号替换主节点,并重新启动下一轮共识.

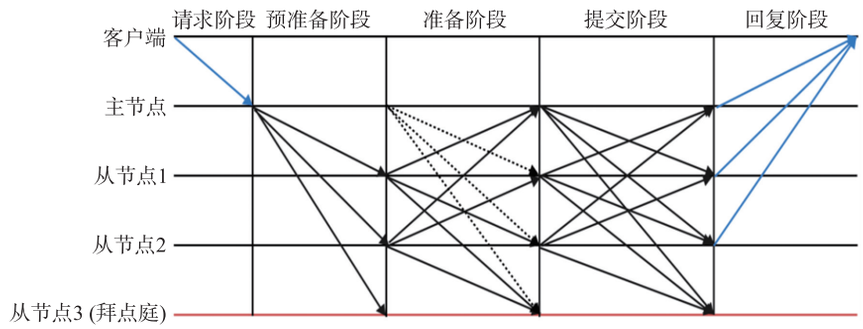

图 1　PBFT 算法共识流程图

Fig. 1　Consensus flowchart of PBFT algorithm

1.2　Raft 共识算法

对于简化后的拜占庭问题,Lamport[18] 于 1998 年首次公开 Paxos 算法,但由于该算法晦涩难懂且难以实现,所以 Ongaro 等[19] 于 2014 年提出了一个新的分布式协议算法 Raft. Raft 算法主要由两部分构成,分别是领导者选举和日志复制.

Raft 算法从始至终都只存在 3 种角色,分别是领导者(Leader)、候选者(Candidate)、跟随者(Follower),算法中的每个节点在某一时刻都只能是 3 个角色中的一个. 每个小组在正常运行过程中都只有一个领导者,负责处理客户端发来的请求、日志复制及向跟随者定期发送心跳请求,说明数据是从领导者向跟随者单向流动的. 候选者状态存在于领导者选举阶段,通过任期号和所得票数进行领导者身份竞争,得票多者将成为下一任期的领导者. 跟随者是被动的,正常情况下不会主动发出请求,当超过一定时间没有收到来自领导者的心跳请求,就会触发超时事件,身份转变为候选者. Raft 算法中,任期充当了逻辑时钟的作用,用来让各节点检测过期信息. 若一个节点发现自己的任期比其他节点小,要立即更新自己的任期;一个候选者或领导者发现了新的任期,就要从当前状态切换到跟随者状态;一个节点接收到了比自己的任期编号小的请求,则会拒绝这个请求. 节点详细的状态转换过程如图 2 所示.

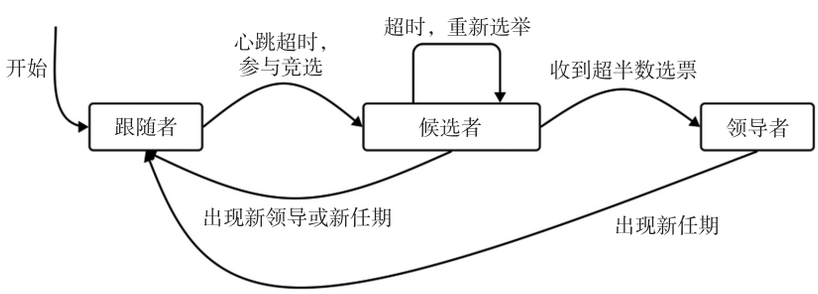

图 2　Raft 算法节点状态转换图

Fig. 2　Node state transition graph of Raft algorithm

领导者选举完成后,将进入日志复制阶段. 领导者接收所有来自客户端的请求,并将其以日志复制命令的形式广播给所有跟随者. 跟随者在接收到日志复制命令后会先拿自己的数据和其中的任期号和索引进行对比,只有数据一致才进行接收并回复确认消息,若拒绝接收则领导者就会发送上一个日志的任期号和索引,直至收到确认消息.

Raft 共识算法是一种强一致性、去中心化、高可用的分布式协议,在系统中不超过一半的节点发生宕机时仍能正常工作,但其缺点也十分明显,即不支持拜占庭节点存在. 为了解决这个问题,在分组内设置监督节点并制定了相关监督策略.

1.3　门限签名

门限签名是秘密共享和数字签名的一种结合,大意是指 n 个成员各自拥有一部分不同的完整私钥,但只需 t 个或 t 个以上的成员即可生成完整私钥,其中 t 不大于 n. 当前已有许多门限签名方案被提出,例如 RSA、BLS、SM2 等. 由于其生成签名的方式与共识算法过程的完美契合,本文考虑将其中一种门限签名方案——BLS,加入到本文的 TRBFT 共识算法中.

BLS 签名算法是由斯坦福大学的 Boneh 等[20]提出的,主要实现原理为将待签名的消息散列到椭圆曲线上的点,再利用双线性函数的性质在不泄露私钥的情况下进行签名. BLS 算法已有多方面应用,例如刘琪等[21]将其应用于平行链研究,Abdulrahman 等[22]将 BLS 结合云计算提出了一套更加安全的公共审计系统. 将 BLS 签名算法直接应用于门限签名,其实现步骤如下:

(1)初始化:生成系统所需公开参数.

G_1、G_2、G_T 是阶为素数 p 的乘法循环群,生成元分别是 g_1 和 g_2,双线性对 $e:G_1\times G_2\to G_T$,H 是映射到 G_1 的哈希函数,公共参数 $\eta=(G_1,G_2,G_T,p,g_1,g_2,e,H)$.

(2)密钥生成:由密钥生成中心完成,选择系统主私钥计算系统主公钥,分别计算节点私钥和公钥.

系统主私钥:$MSK=x$,x 为系统选择的随机数.

系统主公钥:$MPK=v=g_2^x\in G_2$.

随机选择一个 Z_p 上的 $t-1$ 阶多项式 P,满足 $P(0)=x$,计算 $x_i=P(i)$ 作为签名者 u_i 的私钥,$v_i=g_2^{x_i}$ 为 u_i 公钥.

公开主公钥 MPK 和所有用户公钥.

(3)签名:用户对消息进行签名并进行广播.

用户 u_i 计算对消息 m 的签名:$\sigma_i=H(m)^{x_i}$,广播 σ_i.

(4)验证单个签名:收到签名消息先进行验证并记录通过的签名.

用户 u_i 收到来自 u_j 的签名 σ_j 后,首先验证签名正确性:

$$e(\sigma_j,g_2)=e(H(m),v_j),$$

若等式成立则验证通过,记录下来.

(5)验证门限签名:将收集到的 t 个及以上的部分签名进行合成,验证完整门限签名.

用户 u_i 收到至少 t 个正确签名后计算完整签名:

$$\sigma=\prod_{i=1}^{t}\sigma_i^{\varphi_i},\text{其中 }\varphi_i=\frac{\prod_{j=1,j\neq i}^{t}(0-j)}{\prod_{j=1,j\neq i}^{t}(i-j)}\ (\mathrm{mod}\ p).$$

由拉格朗日插值公式可知,任意 t 个及以上用户所产生的完整签名相同. 对签名进行验证:

$$e(\sigma,g_2)=e(H(m),MPK),$$

等式成立则为正确门限签名,反之则验证失败.

2 TRBFT 共识机制

为了应对 PBFT 共识算法出现网络性能较差及节点较多的情况,本文提出了一种更为高效、安全的拜占庭容错算法 TRBFT. 该算法基于高效可信的 BLS 门限签名与分层式 Rbft 算法,并引入信誉值动态分组策略,当参与算法的共识节点总数为 N 时,算法所能容忍的最大拜占庭节点 f 不超过 N 的 1/3,即 $N>=3f+1$,且收到不少于 $2f+1$ 个一致信息,才能保证达成一致的非拜占庭节点数量大于拜占庭节点数量.

2.1 信誉值分组策略

分层结构共识算法之所以高效,大部分原因是在于将原本需要两两通信的节点进行分组,通过 Raft 选出主节点后由主节点作为小组代表参与 PBFT 共识,进而大大减少了共识过程的通信量. 在此之前已有不少关于节点分组的方法被提出,如 Luu 等[23]提出了一种基于 ELASTICO 协议的网络节点分组机制,但协议中的运算方法易导致分组不均;黄冬艳等[24]根据一种一致性 Hash 算法提出 Rbft 分组算法,但其监督策略在节点较多的情况下会产生较多的分组从而影响效率;王谨东等[25]根据 K-medoids 聚类算法提出了 K-RPBFT 分组策略,但分组过程较为繁琐且效率不高. 本文为节点引入信誉值属性来进行节点分组:

(1)初始分组:节点初始化时将会拥有最高信誉值,此时根据预设分组数进行随机分组,并设置预设值和及格线.

(2)分组检查:

① 根据 PBFT 共识 1/3 容错性,需要满足分组数 p 不小于 4;

② 根据 Raft 共识 1/2 容错性,需要满足每个分组节点数 r 不小于 3;

③ 在算法运行过程中出现拜占庭节点会导致节点信誉值降为预设值,网络波动导致的信息发送失败则会扣分,所以当一个分组中出现 1/3 以上的不及格节点时,为了预防信誉值较低用户联合进行恶意更换主节点的情况,会触发节点的动态分组.

(3)动态分组:将不及格节点与某个合理分组中的及格节点进行调换,对新加入的分组进行不及格节点占比检查,若仍然不合理则继续调换.

由于信誉值的引入,维护了单个分组内劣迹节点所占比例的稳定,且信誉值过低的节点将无法参与主节点选取,大大提高了方案的安全性. 相比其他分组策略,TRBFT 算法的分组方式更简单高效.

2.2　监督策略

Raft 并不具备抵抗拜占庭节点的功能,若出现主节点向从节点发送错误消息或者从节点恶意更换主节点的情况,就会导致该组复制的消息内容与其他组不同,进而影响共识结果的一致性. 因此,方案中设置监督节点参与组内共识,具体监督策略如下:

(1)主节点广播:主节点向分组内所有节点广播添加日志消息<AppendLog,i,d>,其中 AppendLog 为消息标识,i 为主节点编号,d 为当前区块交易信息摘要.

(2)从节点验证:Raft 算法不考虑拜占庭节点情况,从节点在收到主节点发送消息后会立即向主节点发送响应消息. 在 TRBFT 算法中,从节点接收到同步请求,之后会第一时间向监督节点发送验证请求,监督节点在收到从节点的消息后会对其中的消息摘要 d 进行比较,当有 2/3 个消息摘要与主节点消息摘要相同时,即进入主节点验证环节.

(3)主节点验证:监督节点会将主节点消息向其他监督节点进行广播,若所收到的超过半数的消息中的消息摘要与主节点相同,则证明主节点不是拜占庭节点,此时向从节点发送确认消息,否则进行视图切换. 从节点在收到监督节点确认消息之后,向主节点发送响应消息并进行日志复制,否则进行视图切换并重新开始 Raft 共识.

图 3 所示为组内共识信息交互过程.

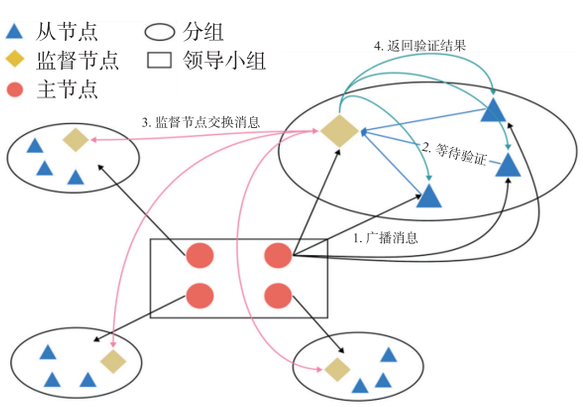

图 3　组内共识流程图

Fig. 3　Flow chart of consensus within the group

2.3　TRBFT 算法共识流程

客户端将请求发送给当前视图的主节点打包成区块后开始共识. TRBFT 算法首先将节点分成 n 组(实验后选出最优数)再结合 BLS 门限签名算法以此来减轻领导组共识过程中的通信压力,组内采用 Raft 共识算法通过信誉值及监督节点来降低拜占庭节点对共识网络的影响. 完整的共识过程如下:

(1)初始化:节点根据 BLS 算法生成初始密钥对并赋予初始值为 100 的信誉值,根据预设的分组数进行随机分组并确定监督节点,各个组内运行 Raft 算法选出主节点建立领导小组.

(2)请求阶段:客户端向当前领导小组主节点发送请求,该节点向其他领导小组成员广播请求消息,消息格式为<Request,v,i,d,m>,其中,Request 是消息标识,v 是当前视图编号,d 是消息 m 的摘要.

(3)请求验证:其他节点收到请求消息时会验证 m 的摘要信息与 d 是否相同,当前视图编号与 v 是否一致. 若一致则进行准备阶段,否则将节点 i 信誉值降为预设值并触发视图更换协议.

(4)准备阶段:节点在请求消息验证通过后生成部分签名,向领导小组内其他节点广播带有部分签名的准备消息,消息格式为<PartSign,v,i,d>.

(5)门限签名:节点在收到 $t(t=2f+1)$ 个及以上的部分签名消息后,首先验证部分签名消息的正确性及 v 与当前视图编号是否一致,验证通过后加上自身的部分签名一共 t 个通过 BLS 门限签名算法合成门限签名 ThresholdSig. 随后向其他组内节点广播签名消息,消息格式为<ThresholdSig,v,i,d>.

(6)签名验证:节点只需收到 1 条门限签名并验证签名正确且视图一致,即可证明各个节点状态达到一致,领导小组内共识完毕,开始各个小组的组内共识.

(7)组内共识:由 Raft 算法选出的主节点向各个从节点发送消息,具体过程详见 2.2 节监督策略.

(8)提交阶段:各个从节点向主节点发送正反馈后,由主节点向客户端发送完成响应,开始下一轮共识.

TRBFT 共识算法的完整流程如图 4 所示.

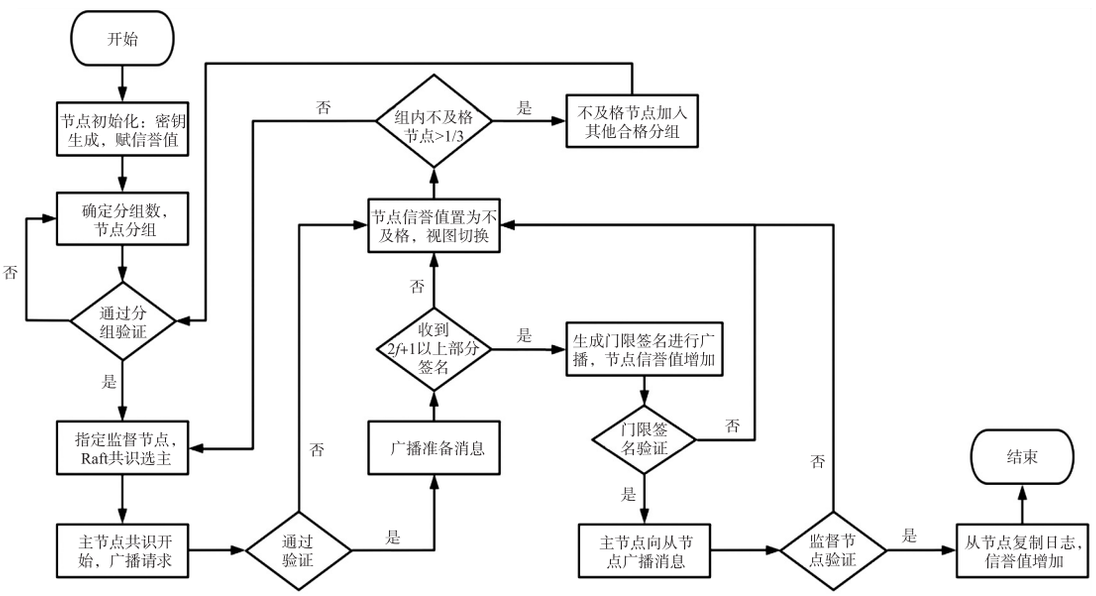

图 4　TRBFT 算法流程框图

Fig. 4　Flow diagram of TRBFT algorithm

3　安全分析与证明

TRBFT 算法在一定程度上继承了 PBFT 算法和 Raft 算法的优点. 组间共识阶段,在 PBFT 算法基础上,结合了更安全高效的 BLS 门限签名算法. 组内共识阶段,在 Raft 算法基础上,引入监督节点和信誉值动态分组,提高了算法的安全性和可行性.

3.1　安全性

在算法运行之前由 Raft 算法在合适信誉值范围内进行选主,各组领导者选择完毕后组成领导者小组开始 TRBFT 算法的第一阶段——组间共识,在该阶段由门限签名负责验证各个节点状态是否达成一致,若将门限值 t 设置为 $2f+1$,则在最终达成一致的结果中就能保证非拜占庭节点的数量一定大于拜占庭节点的数量,符合 PBFT 算法的容错性,且相比 PBFT 的双 $2f+1$ 安全性要求,使用门限签名只需一次即可.

共识的第二阶段即组内共识,由领导节点向从节点发送消息,此时从节点将消息发送给监督节点,由监督节点向其他监督节点互通消息来验证领导节点的正确性,使得 Raft 算法具备了抵抗恶意拜占庭节点的能力. 只有信誉值达到标准的节点才能对监督节点进行反馈,这样可防止出现多个恶意节点合谋恶意更换正常领导者节点的情况.

监督节点是由系统一开始认定的,因此只存在宕机情况. 令小组内各个从节点与监督节点保持心跳响应以确保监督节点存活,若从节点超时未收到响应则认为监督节点宕机,需重新指派监督节点. 节点间消息传递均采用密码学签名算法以确保安全性.

3.2　活性

原有的 PBFT 算法视图替换协议保证了 TRBFT 算法组间一致性的可行性. 在组间共识阶段,当主节点无法响应客户端请求时,将根据先前定义的编号顺序进行主节点更换,以确保共识能够继续. 系统正常运行的关键是保证诚实节点接收到状态一致的新视图,当视图替换协议被触发时,每个节点广播视图更改消息,该消息包含两个重要参数:部分签名和门限签名. 门限签名在视图中携带了法定人数的投票信息,以便可以将其传播到新视图. 若门限签名为空,则可由大于或等于 $t(t=2f+1)$ 的部分签名合成门限签名,从而在新视图中继续保持一致.

在组内共识阶段,领导者由 Raft 选主算法得出,同时也是组间共识的参与者. 组内共识的从节点通过接收来自领导者的心跳请求来确保领导者的正常运行. 若在超时后未能接收到心跳请求,则触发选主协议重新选择领导者. 新的领导者取代失效的领导者参与组间共识,降低了在组间共识阶段该节点出现错误的可能性,使得 TRBFT 算法的活性也有所增强. 且由于信誉值的存在,当小组内信誉值不达标的节点大于 1/3 时会触发动态分组,将多余不达标节点分摊给其他有接收能力的小组,保证了组内共识的正常运行.

3.3　可用性

可用性是指在正常运行期间,产品、系统或设备所具有的功能满足用户需要的程度. 可用性是系统设计和实现中必须考虑的问题,是衡量系统质量标准的一个重要方面,直接关系到产品、系统或设备能否满足用户使用要求. 面对联盟链环境下的网络实时性差、节点众多等问题,TRBFT 算法在领导层共识中采用门限签名的单轮 $2f+1$ 来取代原 PBFT 的双轮 $2f+1$ 信息验证. 单个部分签名的不可伪造性保证了合成门限签名的 $2f+1$ 个成员都是非拜占庭节点,也就确保了共识的正确性,在效率更高的情况下满足了方案的容错性. 在组内共识阶段,Raft 选主机制确保了该部分具有 1/2 的容错性,只要从节点恶意占比不超过半数即可正常进行日志复制,完成共识.

3.4　通信次数

传统的 PBFT 算法的共识过程主要分为预准备、准备和确认 3 个阶段. 假设参与共识的节点一共有 N 个,在预准备阶段,主节点向从节点广播预准备消息,通信量为 $N-1$;准备阶段,各个节点互相广播确认消息,通信量为 $N*(N-1)$;确认阶段同样互相广播确认消息,通信量为 $N*(N-1)$;则共识过程的通信量为 $N-1+N*(N-1)+N*(N-1)=2*N*N-N-1$.

TRBFT 算法的共识过程中,假设节点个数 N 为 16,分为 4 个小组,每组 4 个节点. 在领导者共识阶段,广播请求为 4−1=3 次;广播准备消息为 4 * (3−1)= 8 次;确认消息为 4 次;一共是 3+8+4=15 次. 在组间共识阶段,领导者向从节点发送消息复制请求 4 * (4−1)= 12 次;从节点向监督节点发送验证请求 4 * (4−2)= 8 次;监督节点互相验证 4 * (4−1)= 12 次;验证通过,开始复制 4 * (4−2)= 8 次;一共是 12+8+12+8=40 次. 两个阶段的通信量一共是 55 次,相比于 PBFT 算法 16 个节点共计 2 * 16 * 16−16−1=495 次的情况,通信次数大大减少,且参与共识节点越多,效果越明显.

4　仿真实验

本文通过 IDEA 工具,采用 Java 语言对所提出的 TRBFT 算法进行仿真模拟,选用 PBFT 算法作为参照,在同等硬件环境及网络条件下进行测试. 以网络时延和平均吞吐量作为评价标准,测试在不同节点数量及不同分组情况下两种算法的性能差异.

4.1　网络时延

TRBFT 算法与经典 PBFT 算法的共识时延比较结果如图 5 所示. 从图中可知,TRBFT 算法和 PBFT 算法的共识时延均随着参与共识节点数量的增加而增加. 在满足分组条件前提下,TRBFT 算法共识所用时间远低于 PBFT 算法,且二者之间的差距随着节点数的增加而愈发明显. 对于 TRBFT 算法,共识节点数相同而分组数增加会导致共识时延的增加,这是因为越多的分组数会导致越多的节点参与到领导层共识,而领导层共识是影响效率的主要因素,从而导致了整体时延上升. 但组内节点数固定的 TRBFT 算法时延仍远小于 PBFT 算法,故在节点更多、分组更复杂的实际情况中,所提算法仍能保证共识的高效率性.

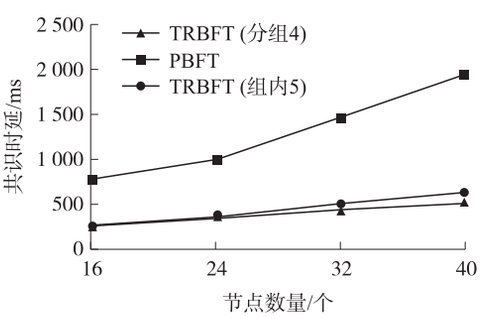

图 5　TRBFT 算法和 PBFT 算法时延对比

Fig. 5　Comparison of delay between TRBFT algorithm and PBFT algorithm

4.2　平均吞吐量

吞吐量是指对网络、设备、端口、虚电路或其他设施,单位时间内成功地传送数据的数量. 本文采用完成请求端发出的请求数量来衡量算法的吞吐量. 影响吞吐量的主要因素有两个:请求端发出的请求数量

和参与共识的节点数量. 本文通过两次不同的实验来比较算法功能.

固定参与共识的节点数量为 16 个,调整发出请求的数量,将成功完成的请求数量记录下来作为吞吐量,实验结果如图 6 所示. 从图中可以看到,请求数量在 300 之前,两个算法的吞吐量均稳定上升且 TRBFT 算法的上升幅度更大;在请求数量超过 300 之后,由于受实验环境的限制二者吞吐量开始下降,但 TRBFT 算法的吞吐量始终高于 PBFT 算法.

固定请求数量为 250,调整参与共识的节点个数,实验结果如图 7 所示. 从图中可以看出,无论节点数量多少,TRBFT 算法的吞吐量均高于 PBFT 算法,且在 32 个节点之后 PBFT 算法由于两两共识的高频信息交换,吞吐量的下降幅度也比 TRBFT 算法要快得多.

由此可知,TRBFT 算法的吞吐量要高于 PBFT 算法. 在网络环境更为复杂的联盟链中,TRBFT 算法能更优秀地处理更多事务.

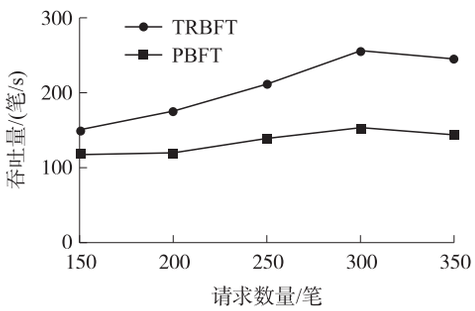

图 6　不同交易量下的吞吐量对比

Fig. 6　Comparison of throughput under different transaction volumes

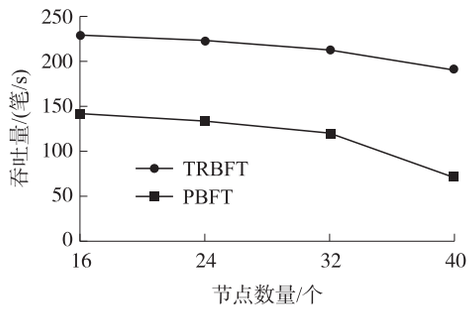

图 7　不同节点数下的吞吐量对比

Fig. 7　Comparison of throughput under different node numbers

5　结论

本文提出一种基于 Raft 算法及 BLS 门限签名的共识算法 TRBFT,该共识算法结合了分层算法和门限签名的优点,在降低通信复杂度的同时提高了用户对数据的信任度. 同时,在 Raft 算法中提出通过信誉值进行动态分组的概念,信誉值不达标节点将无法参与主节点选举,增加了拜占庭节点的作恶成本,动态地调整恶意节点的分组占比也增加了方案的活性及安全性. 实验结果表明,TRBFT 共识算法具有更高的共识效率及更优秀的吞吐能力,更适用于网络复杂环境中节点众多的联盟链环境. 后续将继续研究优化分组条件及取消监督节点,以进一步加强方案的高效性与安全性.

[参考文献](References)

[1] 邵奇峰,金澈清,张召,等. 区块链技术:架构及进展[J]. 计算机学报,2018,41(5):969-988.

[2] 刘明达,陈左宁,拾以娟,等. 区块链在数据安全领域的研究进展[J]. 计算机学报,2021,44(1):1-27.

[3] SWAN M. Blockchain thinking: the brain as a decentralized autonomous corporation[J]. IEEE Technology and Society Magazine,2015,34:41-52.

[4] JING N,LIU Q,SUGUMARAN V. A blockchain-based code copyright management system[J]. Information Processing & Management,2021,58(3):102518.

[5] JIA B,ZHOU T,LI W,et al. A blockchain-based location privacy protection incentive mechanism in crowd sensing networks[J]. Sensors(Basel),2018,18(11):3894.

[6] SHANG W Q,LIU M Y,LIN W G,et al. Tracing the source of news based on blockchain[C]//Proceedings of the 2018 IEEE/ACIS 17th International Conference on Computer and Information Science(ICIS). Singapore:IEEE,2018:377-381.

[7] 孟昊同,张大伟. Hyperledger Fabric 共识机制优化方案[J]. 自动化学报,2021,47(8):1885-1898.

[8] 冯了了,丁滟,刘坤林,等. 区块链 BFT 共识算法研究进展[J]. 计算机科学,2022,49(4):329-339.

[9] 巫史政. 基于拜占庭容错的区块链共识机制优化研究[D]. 昆明:云南大学,2021.

[10] CHUN B G,MANIATIS P,SHENKER S,et al. Attested append-only memory: making adversaries stick to their word[J].

ACM SIGOPs Operating Systems Review,2007,41(6):189-204.
[11] 王日宏,张立锋,徐泉清,等. 可应用于联盟链的拜占庭容错共识算法[J]. 计算机应用研究,2020,37(11):3382-3386.
[12] KAPITZA R,BEHL J,CACHIN C,et al. CheaPBFT:Resource-efficient Byzantine fault tolerance[C]//Proceedings of the 7th ACM European Conference on Computer Systems. Bern,Switzerland:ACM,2012:295-308.
[13] LIU J,LI W T,KARAME G O,et al. Scalable byzantine consensus via hardware-assisted secret sharing[J]. IEEE Transactions on Computers,2019,68(1):139-151.
[14] GUETA G G,ABRAHAM I,GROSSMAN S. et al. Sbft:a scalable and decentralized trust infrastructure[C]//Proceedings of the 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Net Works(DSN). Portland,USA:IEEE,2019.
[15] YIN M F,MALKHI D,REITER M K,et al. HotStuff:BFT consensus with linearity and responsiveness[C]//Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing. New York,USA:ACM,2019:347-356.
[16] CASTRO M,LISKOV B. Practical Byzantine fault tolerance[C]//Proceedings of the Third Symposium on Operating Systems Design and Implementation. Berkeley,USA:USENIX Association,1999:173-186.
[17] 储劲松,鲍可进,夏纯中. 基于改进的 PBFT 算法的性能模型研究[J]. 计算机与数字工程,2020,48(9):2225-2228.
[18] LAMPORT L. The part-time parliament[J]. ACM Transactions on Computer Systems,1998,16(2):133-169.
[19] ONGARO D,OUSTERHOUT J. In search of an understandable consensus algorithm[C]//Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference. Philadelphia,USA:USENIX,2014.
[20] BONEH D,LYNN B,SHACHAM H. Short signatures from the weil pairing[J]. Journal of Cryptology,2004,17(4):297-319.
[21] 刘琪,郭荣新,蒋文贤,等. 基于 BLS 聚合签名技术的平行链共识算法优化方案[J]. 计算机应用,2022,42(12):3785-3791.
[22] ABDULRAHMAN B J,MOHAMMED T H,SABEEH G M,et al. A secure and efficient public auditing system of cloud storage based on BLS signature and automatic blocker protocol[J]. Journal of King Saud University:Computer and Information Sciences,2022,34(7):4008-4021.
[23] LUU L,NARAYANAN V,ZHENG C D,et al. A secure sharding protocol for open blockchains[C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.New York,USA:ACM,2016.
[24] 黄冬艳,李浪,陈斌,等. RBFT:基于 Raft 集群的拜占庭容错共识机制[J]. 通信学报,2021,42(3):209-219.
[25] 王谨东,李强. 基于 Raft 算法改进的实用拜占庭容错共识算法[J]. 计算机应用,2023,43(1):122-129.

[责任编辑:严海琳]