DOI: 10.3969/j.issn.0255-8297.2021.01.007

基于信誉投票的 PBFT 改进方案

涂园超[1,2]， 陈玉玲[1,2]， 李 涛[1,2]， 任晓军[3]， 卿欣艺[1,2]
1. 贵州大学 计算机科学与技术学院，贵州 贵阳 550025
2. 贵州大学 公共大数据国家重点实验室，贵州 贵阳 550025
3. 潍坊科技学院 农蔬区块链实验室，山东 寿光 262700

摘 要： 区块链作为一种去中心化、防篡改的分布式账本，其性能从根本上受共识机制效率的影响。实用拜占庭容错算法以视图切换随机选取主节点的方式会导致安全问题，且在节点较多时共识效率变低。针对这两个问题，提出了基于信誉投票的 PBFT 改进方案。根据节点划分机制评估节点的可靠性，动态地选取高信誉值节点来参与共识，降低恶意节点成为共识节点的概率，增加系统的安全性；根据节点状态转移机制转换节点的角色，维持系统的正确运行，提高系统的稳定性。将所提方案与 PBFT 方案进行对比实验的结果表明：在系统长期运行中，所提方案能减少共识过程中的拜占庭节点和通信开销，提高容错率和吞吐量。

关键词： 区块链；实用拜占庭容错；共识机制；信誉积分；投票选举
中图分类号： TP309.2 **文章编号：** 0255-8297(2021)01-0079-11

Improved PBFT Scheme Based on Reputation Voting

TU Yuanchao[1,2], CHEN Yuling[1,2], LI Tao[1,2], REN Xiaojun[3], QING Xinyi[1,2]
1. College of Computer Science and Technology, Guizhou University, Guiyang 550025, Guizhou, China
2. State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, Guizhou, China
3. Blockchain Laboratory of Agricultural Vegetables, Weifang University of Science and Technology, Shouguang 262700, Shandong, China

Abstract: As a decentralized, tamper-proof distributed ledger, the performance of blockchain is fundamentally affected by the efficiency of consensus mechanisms. Practical Byzantine fault tolerance (PBFT) algorithm randomly selects master nodes through view-switching, leading to problems of security vulnerabilities and low consensus efficiency in the case of large number of nodes. In response to the two problems, a PBFT improvement scheme based on reputation voting is proposed. The reliability of nodes is evaluated according to node division mechanism, where high reputation nodes are dynamically selected to participate in the consensus, and a malicious node is assigned with lower probability of

收稿日期： 2020-11-12
基金项目： 国家自然科学基金（No.61962009）；贵州省科技重大专项计划基金（No.20183001）；贵州省公共大数据重点实验室开放课题基金（No.2018BDKFJJ003, No.2019BDKFJJ011）资助
通信作者： 陈玉玲，副教授，研究方向为大数据安全与隐私保护、区块链等。E-mail: ylchen3@gzu.edu.cn

becoming a consensus node, accordingly increasing the security of the system. By switching the role of nodes according to node state transfer mechanism, the scheme can maintain the correct operation of the system and improve the stability of the system. Experiments on the proposed and the traditional PBFT schemes show that the proposed one can reduce Byzantine nodes and communication overhead in long-term consensus processes, and improve the fault tolerance rate and the data throughput of transaction.

Keywords: blockchain, practical Byzantine fault tolerance (PBFT), consensus mechanism, credit score, vote by ballot

自 2008 年中本聪提出比特币 [1] 以来，数字货币不断发展，作为其底层技术的区块链 [2] 也备受关注。区块链本质上是一个去中心化、集体维护的分布式数据库，利用密码学、共识算法 [3]、点对点通信来实现数据防篡改和可溯源 [4]。共识算法的效率决定了区块链系统的性能，现有的经典共识算法包括工作量证明（proof of work, PoW）算法 [5]、实用拜占庭容错（practical Byzantine fault tolerance, PBFT）算法 [6]、权益证明（proof of stake, PoS）算法和委托权益证明（delegated proof of stack, DPoS）算法 [7] 等。PBFT 算法能够让区块链完全脱离链上代币的奖励机制，且不需要大量算力来维护，因此在分布式系统中得到了应用 [8-9]，但仍存在主节点选取的安全漏洞问题以及多节点时的通信开销过大等问题。

针对 PBFT 存在的问题，涌现出了大量的研究成果。文献 [10] 设计了同步拜占庭法定人数系统，虽然可以得到正确的拜占庭法定人数，但是存在诚实节点被定义为恶意节点的可能性。文献 [11] 提出了蜜獾拜占庭容错协议，在没有设定时间的情况下可以达成异步网络中的分布式一致性共识，但是该协议使用了复杂门限加密方式和异步方式，使共识时间变得冗长而难以应用。文献 [12] 提出了交差容错方案，能够容忍敌手模型为 $n = 2f + 1$ 的宕机节点与拜占庭错误节点，但也只有在错误节点很少的情况下才能高效地达成共识，不太适用于节点众多的网络模型。文献 [13] 提出基于随机预言模型验证的异步拜占庭一致（validated asynchronous Byzantine agreement, VABA）算法，在每次共识过程中采用多个并行的节点进行提议，并从中随机选取一个作为最终结果；该模型采用门限签名等技术，能使每一轮共识的通信复杂度降为 $O(n^2)$，其缺点是随机选取的并行方案会导致安全性漏洞。文献 [14] 提出了一种随机游走算法，压缩了 PBFT 算法传输的通信数据，减少了通信成本，却选择过于随机而不能保证数据质量。文献 [15] 提出一种探测路径的攻击方法，提高了系统的安全性，但会把处于宕机状态的节点当成发动攻击的节点，且在安全性方面的花销过大。上述研究都在一定程度上改进并提升了拜占庭容错算法，但是并未对节点进行信誉评估，也就不能保障主节点选取的安全和共识过程的稳定；特别是当共识节点发生异常时，没有将其从共识节点集合中剔除的相应措施。此外，PBFT 始终受制于有限的扩展性和较低的交易处理能力，通常只在拥有少数节点的联盟链系统中使用 [16-17]。

PBFT 存在视图切换随机选取主节的安全漏洞以及共识节点较多时共识效率变低的问题。为了降低选出的主节点为拜占庭节点的概率，本文根据节点划分机制为节点匹配不同的信誉值，以信誉值区分信誉值高的节点和拜占庭节点；根据节点状态转移机制，在信誉值高的备选节点中选择新视图的主节点，降低了主节点为拜占庭节点的可能性。为解决系统在节点较多时共识效率较低的问题，用信誉投票模型选择信誉值高的诚实节点来完成共识过程，减少共识节点的数目，降低共识过程中视图切换的频率，从而提高了共识效率和吞吐量。

1 相关知识

区块链系统是否能高效地达成共识是决定其性能好坏的重要指标。随着区块链的发展，

出现了许多共识机制，其中主要分为证明类以及具有容错性的拜占庭容错（Byzantine fault tolerance, BFT）类。由于分布式系统中节点之间互不了解，受到利益驱使产生大量拜占庭节点。这些恶意节点会主动向其他节点发送错误信息，因此有必要使用具有拜占庭容错能力的共识机制。

1.1　证明类共识机制

1.1.1　PoW 机制

工作量证明可以通过计算一个随机数使得区块的哈希值满足当前的目标难度。由于哈希值在数学上主要采用穷举法碰撞所得，需要进行大量的计算。挖矿节点一旦计算出满足规则的随机数即被认定为付出了一定的工作量，从而获得记账权和奖励。工作量证明以巨大的算力维持系统的去中心化和安全性，但在挖矿过程中浪费大量能源以及硬件资源。在比特币中动态调整目标难度可以使每个区块的生成时间保持在 10 min 左右，可见吞吐量难以满足现实交易需求。

1.1.2　PoS 机制

PoS 机制引入币龄 (coin days) 的概念 —— 用户持有系统代币数量和持有时间的乘积，可以缓解 PoW 中算力资源浪费的问题。在 PoS 的共识过程中，系统可以根据矿工持有的币龄设置挖矿难度，矿工生成区块后会消耗相应的币龄。与 PoW 相比，PoS 降低了持有币龄矿工挖矿的难度，减少了挖矿算力资源的浪费和挖矿时间，但也会打击新节点参与共识的积极性, 也没有完全摆脱挖矿过程中算力资源的浪费问题。

1.1.3　DPoS 机制

DPoS 是在 PoS 的基础上改进而来的，它对中心化进行了适当妥协，让所有权益持有者选出 101 名委托节点来轮流生产区块。委托节点若诚实履行共识过程中的职责，就能从生成的区块中获得收益。当委托节点出现异常时，其他节点可以投票替换该异常节点。DPoS 能在秒级单位完成共识验证过程，其缺点是新加入节点参与共识的积极性不高，从而导致系统的中心化问题。

1.2　PBFT 共识机制

PBFT 由 Castro 和 Liskov 于 1999 年提出，是公认的解决拜占庭将军问题的最优协议。与 PoW 相比，PBFT 避免了算力资源的浪费，提高了区块链系统的出块速度。为达到共识状态，PBFT 共识机制运行以下 3 种协议：一致性协议、检查点协议和视图更换协议 [18-19]。

1.2.1　一致性协议

一致性协议将共识节点分为主节点和从节点，主节点只有一个，用于客户端的请求排序；从节点按照主节点排好的顺序执行，保证在各节点上执行请求的顺序一致，从而确保区块内容一致。一致性协议交互流程主要分为以下 5 个阶段：

1）客户端发起消息请求 $<\text{request}, o, t, c>$。

2）预准备阶段 $<<\text{pre-prepare}, v, n, d>, m>$。

3）准备阶段 $<\text{prepare}, v, n, d, i>$。

4）确认阶段 $<\text{comment}, v, n, D(m), i>$。

5）回复阶段 $<\text{reply}, v, t, c, i, r>$。

其中：request、pre-prepare、prepare、comment 和 reply 均为消息名称，o 为请求的具体操作，t 为时间戳，c 为客户端的标识，v 为视图编号，n 为消息编号，d 为客户端消息摘要，m

为客户端发送的消息内容，i 为当前副本节点编号，$D(m)$ 为副本节点签名的集合节点，r 代表请求结果。

一致性协议的交互流程如图 1 所示。

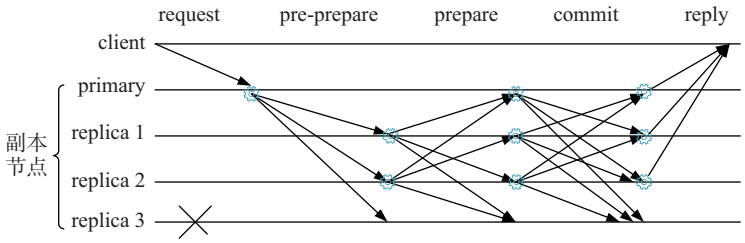

图 1 PBFT 算法一致性协议交互流程

Figure 1 Interaction process of consensus protocol for PBFT algorithm

1.2.2 视图切换协议

每个节点需在相同的配置信息下工作，该配置信息称为视图。视图切换协议是在主节点发生故障时维持系统正常运行的协议。当主节点发生故障时，需要更改视图，视图编号加 1 即 $v+1$，根据 $p=v \bmod |N|$ 选定新的主节点；从节点触发视图切换协议，从最近完成的一个区块的时间戳 t 开始，触发条件有两个：1）在限定时间 t_1 内没有收到主节点的 pre-prepare 广播；2）在限定时间 t_2 内没有生成新的区块，其中 $t_1 < t_2$ 。在上述两个触发条件中，只要满足一个条件就会触发视图切换协议。为了保证系统的正确性和一致性，视图切换需要进行以下 3 个交互通信步骤：

步骤 1 开启视图切换协议，从节点进入视图 $v+1$，向所有副本节点广播 view-change 消息。

步骤 2 副本节点收到包括自身的 $n=2f+1$ 条 view-change 消息后，向视图 $v+1$ 中的主节点发送 view-change-ack 消息，新的主节点收到 view-change-ack 消息后进入 new-view 阶段。

步骤 3 新的主节点选择检查点作为 new-view 请求的起始状态，然后根据本地块链接数据执行一致性协议。

1.2.3 检查点协议

在共识过程中，节点会生成大量的日志，导致存储开销过大。检查点协议可以减小节点数据存储规模，释放经过共识认证的日志消息，降低系统内存开销。当节点因自身故障或网络问题而不能与其他节点保持同步时，就会影响到系统的正常运行，因此有必要让检查点周期性工作，保持节点一致性。

PoW、PoS、DPoS 以及 PBFT 机制的执行速度、可扩展性、拜占庭容错、吞吐量（transaction per second, TPS）以及代表应用如表 1 所示。

由表 1 可知：PBFT 的执行速度和吞吐量相较于 PoW、PoS、DPoS 有明显的提高，但在扩展性和拜占庭容错能力方面存在不足之处。针对这两个问题的改进方案有 DPBFT[20]，但没有设计主节点的选取方案，也没剔除拜占庭节点的功能。本文提出基于信誉投票的 PBFT 改进方案，用以改进 PBFT 主节点选取方案以及解决节点较多时在扩展性方面的问题，并具有剔除拜占庭节点的性能。

表 1　共识算法对比

Table 1　Comparison among consensus algorithms

共识算法	执行时间/s	可扩展性	拜占庭容错	TPS	代表应用
PoW	>100	强	$<1/2$	<100	比特币
PoS	<100	强	$<1/2$	$<1\,000$	点点币
DPoS	<100	强	$<1/2$	$<1\,000$	比特股
PBFT	<10	弱	$<1/3$	$<2\,000$	Hyperledger

2　基于信誉投票的 PBFT 改进方案

PBFT 算法存在异常节点被选为主节点、共识过程中通信开销大、视图切换频率高等问题，于是本文提出基于信誉投票的 PBFT 改进方案。首先采用信誉投票模型并根据节点行为分配不同的信誉积分；然后按照节点信誉值将节点划分为不同角色来行使不同的功能，且角色间能动态转换；最后通过剔除机制使恶意节点无法参与共识。该方案解决了主节点选取方式过于简单而造成的安全问题，有效地减少了视图的切换频率和通信量，进而提高了共识效率。

2.1　方案整体流程

方案在原始 PBFT 算法的基础上进行改进，将信誉投票模型与 PBFT 相结合，筛选出在共识过程中快速而诚实的共识节点。依据信誉值将节点划分为普通节点、投票节点、候选节点和备选主节点，进而构成共识集合。此后的共识过程就由选出的共识集合完成，若接受交易信息则写入区块链；若不接受则剔除交易数据。方案整体流程如图 2 所示。

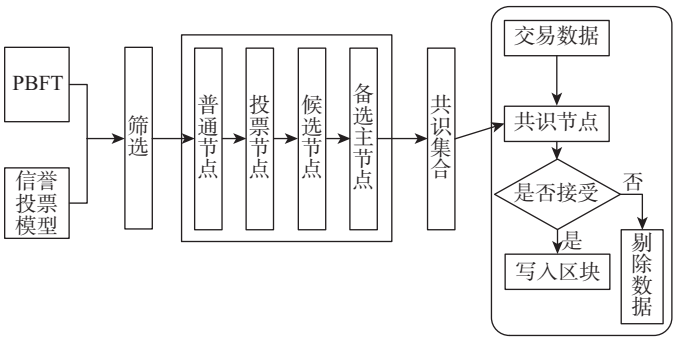

图 2　方案流程

Figure 2　Process of scheme

2.2　信誉投票模型

信誉投票模型的目的是为了选出可靠的共识节点，由节点划分机制、节点状态转移机制和恶意节点剔除机制构成。模型对参与共识节点的行为进行评估，区分高信誉值节点与有恶意行为的节点。该模型把节点间交互频率和节点的诚实度作为主要参考因素，分析节点之间的交易次数、达成交易时间以及能否正确完成共识过程，为节点分配相应的信誉值，然后在高信誉值节点中选择出普通共识节点、投票节点、候选节点和备选主节点，并通过投票节点监督打分的操作规范主节点的行为。

PBFT 方案无法对恶意节点虚假的评价和共谋行为进行有效的处理，而信誉投票模型能降低行为异常节点的信誉积分，使其无法加入到共识集合，从而避免干扰到正常的共识过程。

2.2.1 节点划分机制

经过随机 x（x 的值根据安全需要动态选择）轮 PBFT 共识后进行共识集合的选取。首先排除在前 x 轮共识过程中发生故障或发送与大多数节点消息不一致的节点，再实行如下方案：每次正确完成共识，节点的信誉值加 1；经过 Δ 轮 PBFT 共识后，诚实节点都会正确完成共识，节点的信誉值将为 Δ。原 PBFT 共识算法中共有 $3f+1$ 个节点，至少存在 $2f+1$ 个诚实节点才能正确完成共识。若有恶意节点想加入共识集合，则每次共识都必须诚实，此时会有大于 $2f+1$ 个节点的信誉值为 Δ，并被标记为诚实节点。将在规定时间内达到 Δ 分值的节点标为普通节点，且只能参与共识。经历多次共识再根据节点的信誉值可以把参与共识的节点依次划分为普通节点、投票节点、候选节点和备选主节点。这些节点在共识中行使不同的功能，维持系统的稳定运行，具体划分步骤如下：

步骤 1 选取普通节点中先达到信誉值 A 的 $\lfloor 3(2f+1)/4 \rfloor$ 个节点，作为投票节点。

步骤 2 选取投票节点中先达到信誉值为 B 的 $\lfloor 3(2f+1)/20 \rfloor$ 个节点，作为候选节点。

步骤 3 选取候选节点中先达到信誉值为 C 的 $\lfloor 3(2f+1)/40 \rfloor$ 个节点，作为备选主节点。

选取信誉值最高的节点作为下个视图的主节点，在无单一最高信誉值的情况下选取信誉值最高节点中编号最小的节点作为主节点。其中 $C > B > A$, 备选主节点的数量是投票节点数量的 1/10，这样能防止投票节点进行恶意投票，使自己快速升级，减少了视图切换的次数。

PBFT 共识机制中的主节点由公式 $p = v \bmod n$ 随机依次选取，选出的主节点为拜占庭节点的概率为 1/3。本方案根据节点划分机制选取信誉值最高的节点依次作为新视图的主节点，选取的主节点为拜占庭节点的概率大大降低。在主节点生成区块过程中，投票节点对其进行监督评估，经过一段时间后对主节点进行投票并选择是否更换主节点。若超过一半的投票节点同意更换主节点，则按照规则选出新的主节点。投票方式既能避免单个节点始终作为主节点而导致系统过度中心化的问题，也能规范主节点的行为操作，有效降低了主节点出现异常行为的可能性，提高了系统的稳定性；同时选取出高信誉值的节点完成共识过程，减少了参与共识的节点数量，降低了通信开销，提高了共识效率。

2.2.2 节点状态转移机制

共识节点的角色是动态转移的，主要与节点在共识过程中的行为有关。当系统初始化时，节点处于正常状态。当节点多次达成有效共识且信誉值达到 Δ 时，可以转换到可信状态。处于可信状态的节点也可能会降级为无法参与共识的节点，状态变为检查状态。若发生恶意行为或停机等情况，处于检查状态的节点将转换为无效状态，无法参与共识过程。此外，新加入的节点诚实地记录每轮共识消息后，其信誉值相应地加 1。只有当节点信誉值达到 Δ 时才能成为共识集合中的节点。在加入新节点后的所有共识节点中，当信誉值积分值 $S_i < \Delta$ 时，节点无法参与到共识过程；当信誉积分值在 $\Delta < S_i < A$ 时, 节点为普通节点，只能参与共识过程而没有其他功能权利；当信誉积分值在 $A < S_i < B$ 时, 节点为投票节点，拥有投票替换主节点的权利，也可以升级为候选节点；当信誉积分值在 $B < S_i < C$ 时, 节点为候选节点，可以升级为备选主节点；当信誉积分在 $S_i > C$ 时，信誉状态极好，可作为备选主节点。当主节点出现错误时，在备选主节点集合内取信誉值最高的节点为主节点；若没有唯一信誉值最高的节点，则在并列信誉值最高的节点范围内选择节点编号最小的节点为主节点。主节点经过一段时间 T 后，可由投票节点进行投票降级为普通节点，并赋予其信誉积分值为 Δ，使其既能参与共识过程又能避免系统过度中心化的问题。角色状态转移流程如图 3 所示。

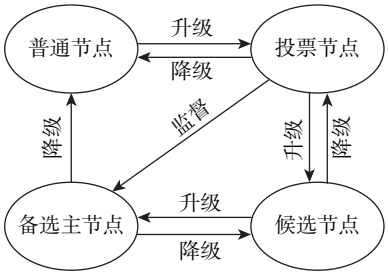

图 3　角色状态转移流程

Figure 3　Role state transition process

2.2.3　节点剔除机制

经过数轮共识后选取的高信誉节点在共识过程中仍然有一定概率碰到如主机故障、带宽时延或极端恶劣天象等无法避免的物理状况，以致无法参与共识。因此，可从投票节点中选出检查节点来降低故障节点信誉值。此外在共识过程中，检查节点首先能检测出在规定时间内没有向客户端发送反馈信息或者反馈的信息与大多数节点反馈的信息不一致的拜占庭节点，并将其从共识节点集合剔除，且添加标记使其无法再加入共识集合。然后选取上一个角色集合中信誉值最高的节点加入本角色集合，若存在多个信誉值相同的节点，则选取编号最小的节点加入本角色集合。剔除机制进一步降低了共识节点中拜占庭节点存在和作恶的可能性，规范了共识节点的诚实度，提高了系统的共识效率和安全性。

2.3　基于信誉投票的 PBFT 共识过程

基于信誉投票模型，选取高信誉值节点参与共识。系统达到稳定状态后，在不考虑共识节点发生故障的情况下，本方案的共识一致性流程分为以下 5 个阶段：

1）请求阶段

客户端向主节点发送请求 $<$ request, o, t, $c>$。

2）预准备阶段

主节点接收到客户端请求后，给请求赋值一个序列号，并把交易产生的数据存入交易池。主节点将交易池中验证通过的交易数据赋予编号，打包后将数据块的信息广播 $<<$ pre-prepare, $v, n, d>, m>$ 发送给其他共识节点。

3）准备阶段

共识节点收到主节点发出数据块的信息 $<<$ pre-prepare, $v, n, d>, m>$ 后验证信息，若验证信息是正确的且没有被恶意篡改，则进行签名并加盖时间戳，接收此消息并广播 $<$ prepare, $v, n, d, i>$。

4）提交阶段

主节点收到共识节点中大于 $f+1$ 的确认消息 $<$ commit, v, n, d, $i>$ 后，将达成共识的结果发给其他副本节点。

5）响应阶段

客户端等待来自不同共识节点的响应 < reply, v, t, c, i, r >，若有 $f+1$ 个共识节点的响应相同，则该响应即为算法的一致结果。

改进方案的一致性协议交互流程如图 4 所示。

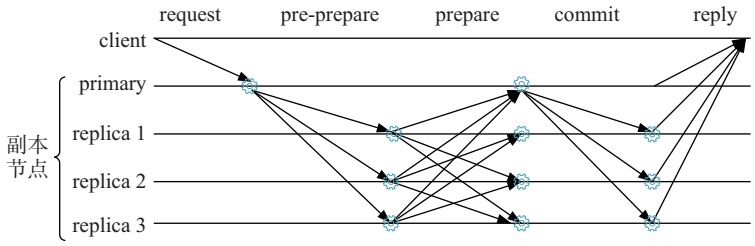

图 4 改进方案的一致性协议交互流程

Figure 4 Interaction process of consistency protocol in improved scheme

完成一致性协议交互过程后，主节点将一段时间内达成共识的数据打包成块写入区块链；若主节点在一定的时间内没有生成数据块，则被视为拜占庭节点，此时根据节点状态转移机制来选取新的主节点继续完成共识过程。

3 实验及分析

本节将主要从通信开销、吞吐量、容错性能和安全性四方面对基于信誉投票的 PBFT 方案进行实验分析，并对比 PBFT 和 DPBFT[20] 来论证本方案的优越性。

3.1 通信开销

在 PBFT 算法中，request 阶段的通信次数为 1，pre-prepare 阶段的通信次数为 $n-1$，prepare 阶段的通信次数为 n^2-n，commit 阶段的通信次数也为 n^2-n，reply 阶段的通信次数为 n，则总通信次数为

$$f(n)=1+(n-1)+(n^2-n)+(n^2-n)+n=2n^2 \tag{1}$$

因此，PBFT 算法复杂度为 $O(n^2)$.

基于信誉投票的 PBFT 改进方案的通信情况如下：request 阶段的通信次数为 m，pre-prepare 阶段的通信次数为 $m-1$，prepare 阶段的通信次数为 $(m-1)^2$，commit 阶段的通信次数为 $m-1$，reply 阶段的通信次数为 m，因此总通信次数为

$$f(m)=m+(m-1)+(m-1)^2+(m-1)+m=m^2+2m-1 \tag{2}$$

基于信誉投票的 PBFT 改进方案的算法复杂度为 $O(m^2)$。

生成单位区块的时间如图 5 所示。

从图 5 中可以看出：在 150 个节点的集合中，随着基于信誉投票模型的运转，相较于 PBFT 和 DPBFT，本文提出的基于信誉投票的 PBFT 改进方案在达到稳定状态的过程中减少了共识通信开销，其原因是本方案减少了共识节点的数量，并在 commit 阶段降低了共识通信级数。

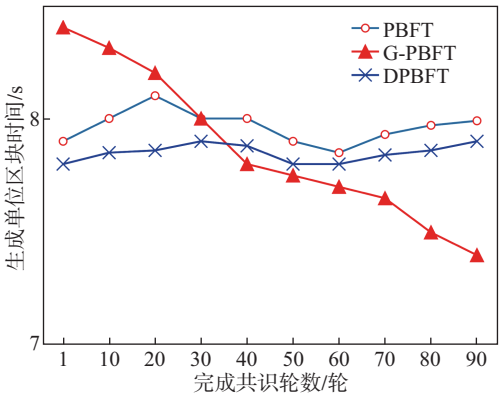

图 5　生成单位区块时间

Figure 5　Time for generating unit block

3.2　吞吐量

吞吐量是区块链系统运行效率高低的检测标准。吞吐量的定义是单位时间内打包进区块的交易数量的平均值，其计算公式为

$$\mathrm{TPS} = B_{\mathrm{transaction}}/\Delta t \tag{3}$$

式中，Δt 为出块时间，$B_{\mathrm{transaction}}$ 为 Δt 时间段内打包进区块的交易数。

对比系统的吞吐量如图 6 所示，当系统中的节点总数分别为 17、49、56、87、110、115 时，本方案相比于 PBFT、DPBFT 方案，具有更高的系统吞吐量；而且由图的趋势可以看出：当节点数持续增加时，本方案相比于 PBFT、DPBFT，在吞吐量方面的优势更明显。原因如下：1）有效减少了共识节点的数量，缩短了通信时间；2）主节点是可靠节点的概率更高，出错的几率更小，视图切换次数也随之减少，完成一次共识的时间相应缩短。

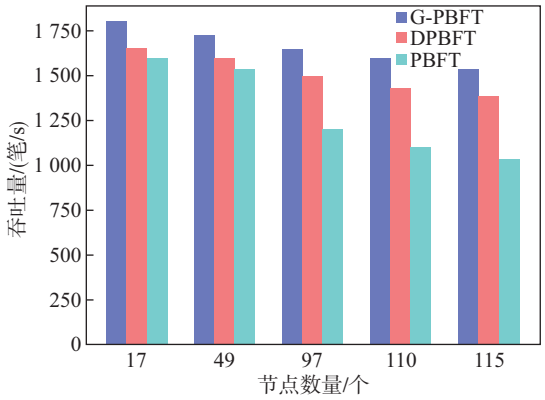

图 6　系统吞吐量的对比

Figure 6　Comparison of system throughput

3.3　容错性能和安全性分析

针对 PBFT 算法中容错性较低的问题，本方案加入了节点剔除机制，使系统具有更高的

容错性能。

如图7所示，在初始有1 000个节点的区块链系统中，有267个标记的拜占庭节点。当运行23次改进的共识方案后，共识集合中被标记的拜占庭节点降到43个。可以看出：一旦进入良性循环，本方案能将拜占庭节点逐渐剔除出共识集合，使系统更加安全可靠；选取出新的主节点为拜占庭节点的概率大幅下降，共识的数据被篡改和窃取的概率越小，系统也更可靠。随着系统的持续运行，新加入的拜占庭节点无法参与到共识过程，不会影响共识过程的正确运行。因此，本方案能使系统容忍新加入的节点中存在拜占庭节点，从而提高了容错性。

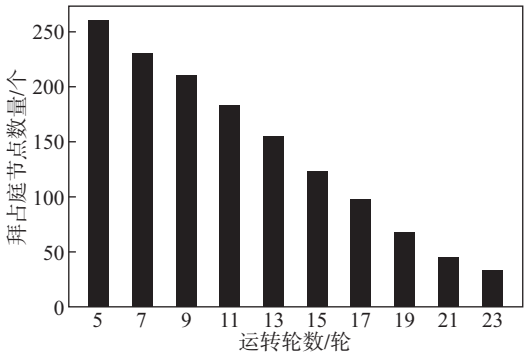

图7 改进方案的拜占庭节点数

Figure 7 Number of Byzantine nodes for improved scheme

4 结 语

本文对PBFT算法中主节点选取不安全和共识节点较多时共识效率较低等问题进行研究，提出了基于信誉投票的PBFT改进方案。以节点划分机制为节点匹配相应的信誉值来区分诚实节点和拜占庭节点，并把诚实节点划分为4个不同的角色参与共识；以节点状态转移机制使共识节点动态流转，维持系统的稳定，提高了新主节点的可靠性，降低了视图切换的频率；以节点剔除机制剔除共识过程中出现的拜占庭节点，减少了共识过程的通信开销，提高了共识效率和吞吐量。未来可以进一步研究如何更快更好地赋予每个节点更加精准的信誉值方法，缩短系统运行进入良性循环的时间，同时在节点划分机制的基础上提高新加入节点的存在认同感，让新加入的诚实节点更快地参与共识。

参考文献:

[1] Nakamoto S. Bitcoin: a peer-to-peer electronic cash system [EB/OL]. [2020-11-08]. https://bitcoin.org/bitcoin.pdf.

[2] 袁勇, 王飞跃. 区块链技术发展现状与展望 [J]. 自动化学报, 2016, 42(4): 481-494.
Yuan Y, Wang F Y. Development status and prospects of block chain technology [J]. Acta Automatica Sinica, 2016, 42(4): 481-494. (in Chinese)

[3] Zheng Z, Xie S, Dai H, et al. An overview of blockchain technology: architecture, consensus, and future trends [C]//International Congress on Big Data, 2017: 557-564.

[4] 刘懿中, 刘建伟, 张宗洋, 等. 区块链共识机制研究综述 [J]. 密码学报, 2019, 6(4): 395-432.
Liu Y Z, Liu J W, Zhang Z Y, et al. Overview on blockchain consensus mechanisms [J]. Journal of Cryptologic Research, 2019, 6(4): 395-432. (in Chinese)

[5] Jakobsson M, Juels A. Proofs of work and bread pudding protocols [C]//Communications and Multimedia Security, 1999: 258-272.

[6] Castro M, Liskov B. Practical Byzantine fault tolerance [C]//Proceedings of the Third Symposium on Operating Systems Design and Implementation, 1999: 173-186.

[7] Larimer D. Delegated proof-of-stake consensus [EB/OL]. [2020-11-10]. https://bitshares.org/technology/delegated-proof-of-stake-consensus.

[8] Ongaro D, Ousterhout J. In search of an understandable consensus algorithm [C]//Usenix Annual Technical Conference, 2014: 305-320.

[9] 张伯阳, 张晓, 李阿妮, 等. 云存储系统可扩展性评测研究 [J]. 计算机应用研究, 2017, 34(7): 1957-1961.

Zhang B Y, Zhang X, Li A N, et al. Research on scalability evaluation in cloud storage system [J]. Application Research of Computers, 2017, 34(7): 1957-1961. (in Chinese)

[10] Bazzi R A. Synchronous Byzantine quorum systems [J]. Distributed Computing, 2000, 13(1): 45-52.

[11] Miller A, Xia Y, Croman K, et al. The honey badger of BFT protocols [C]//ACM SIGSAC on Computer and Communications Security, 2016: 31-42.

[12] Liu S, Viotti P, Cachin C, et al. XFT: practical fault tolerance beyond crashes [M]//Operating Systems Design and Implementation, 2016: 485-500.

[13] Abraham I, Malkhi D, Spiegelman A, et al. Validated asynchronous Byzantine agreement with optimal resilience and asymptotically optimal time and word communication [EB/OL]. [2020-11-10]. http://arxiv.org/abs/1811.01332.pdf.

[14] Zheng H, Guo W, Xiong N, et al. A kernel-based compressive sensing approach for mobile data gathering in wireless sensor network systems [J]. Systems Man and Cybernetics, 2018, 48(12): 2315-2327.

[15] Liu Y, Ma M, Liu X, et al. Design and analysis of probing route to defense sink-hole attacks for Internet of things security [J]. IEEE Transactions on Network Science and Engineering, 2020, 7(1): 356-372.

[16] Brewer E A. Towards robust distributed systems (abstract) [C]//Proceedings of the nineteenth annual ACM symposium on Principles of Distributed Computing. New York: Association for Computing Machinery, 2000.

[17] Gilbert S, Lynch N. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant Web services [J]. ACM SIGACT News, 2002, 33(2): 51-59.

[18] Reiter M K. A secure group membership protocol [C]//Proceedings of the IEEE Symposium on Research in Security and Privacy, 1994: 176-189.

[19] 甘俊, 李强, 陈子豪, 等. 区块链实用拜占庭容错共识算法的改进 [J]. 计算机应用, 2019, 39(7): 2148-2155.

Gan J, Li Q, Chen Z H, et al. Improvement of blockchain practical Byzantine fault tolerance consensus algorithm [J]. Journal of Computer Applications, 2019, 39(7): 2148-2155. (in Chinese)

[20] 刘肖飞. 基于动态授权的拜占庭容错共识算法的区块链性能改进研究 [D]. 杭州: 浙江大学, 2017.

(编辑：秦　巍)