基于 K-medoids 的改进 PBFT 共识机制

陈子豪 李 强
(四川大学计算机学院 成都 610065)

摘 要 随着数字货币的普及与发展,区块链技术进入了大众的视野,并被誉为信用历史上第四个里程碑,是未来信用的基石[1]。但与此同时,区块链技术也面临着共识效率低、算力浪费等问题。文中利用 K-medoids 聚类算法对参与区块链共识的大规模网络节点根据特征进行聚类与层次划分,再将改进的多中心化实用拜占庭容错算法应用于这种聚类后的分层模型中。另外,为了提升聚类算法在多种场景下对区块链模型中共识节点进行聚类的可控性,对 K-medoids 算法进行了改进。网络拓扑仿真环境实验表明,当选择了适当的聚类特征评判节点间的相似度时,改进后的算法 K-PBFT 在 1000 个网络节点参与共识的场景中相较于传统实用拜占庭容错算法(Practical Byzantine Fault Tolerance,PBFT)算法,单次共识耗时缩短了 20%,共识过程的通信次数最佳能够降低 3 个数量级。结果证明 K-PBFT 算法优化了较大规模共识节点参与的共识过程,使区块链模型能够适用于更广泛的场景中。

关键词 实用拜占庭容错算法,K-medoids,区块链,聚类算法
中图法分类号 TP301 **文献标识码** A **DOI** 10.11896/jsjkx.181002014

Improved PBFT Consensus Mechanism Based on K-medoids

CHEN Zi-hao LI Qiang
(College of Computer Science,Sichuan University,Chengdu 610065,China)

Abstract With the popularization and development of digital currency,the blockchain technology enters the public's vision,and has been hailed as the fourth milestone in credit history,the cornerstone of future credit[1]. However,blockchain technology is also facing problems such as low efficiency of consensus and waste of computing power. The K-medoids clustering algorithm is used to cluster and hierarchically divide the large scale network nodes participating in the blockchain consensus based on features,and then the improved multi-centered PBFT (Practical Byzantine Fault Tolerance) consensus algorithm is applied to the clustered model. Moreover,in order to improve the controllability of clustering algorithm to cluster nodes in blockchain model under various scenarios,this paper improved K-medoids algorithm. Simulation results show that when appropriate clustering features are selected to evaluate the similarity between nodes,the improved algorithm K-PBFT reduces the single-consensus time consumption by 20%,and the consensus process communication times can be reduced by three orders of magnitude. The experimental results show that K-PBFT optimizes the consensus process involving large-scale nodes,so that the blockchain technology can be applied to a wider range of application scenarios.

Keywords Practical byzantine fault tolerance,K-medoids,Blockchain,Clustering algorithm

1 引言

近几年,随着数字货币的普及和快速发展,比特币已成为一种家喻户晓的数字货币,区块链作为其底层技术越来越受到关注。同时,银行和金融机构也意识到区块链技术对金融行业的意义,纷纷开始研究并着手布局区块链产业。2016 年 1 月,中国人民银行官方发布了一条题为《中国人民银行的数字货币研讨会在京召开》的新闻[2],这条新闻的发布,使得国内的媒体逐渐开始关注区块链技术;同时,该新闻也在区块链社区中激起千层浪,间接推动了国内区块链技术的研究热潮。区块链技术本质上是一种分布式的数据库技术[3],网络内的节点利用密码学、共识算法、点对点通信等技术,共同维护全网数据的一致性和有效性。

现代计算机技术中,去中心化和中心化各有适用的场景,并不存在绝对优劣之分。现实中,完全去中心化和完全中心化的场景并不多见,所以区块链完全去中心化的宗旨使其在一些领域难以落地。区块链技术的应用由核心共识算法驱动,共识算法本质上是为了解决拜占庭错误[4]。在比特币、以

到稿日期:2018-10-31 返修日期:2019-01-11 本文受基于云模式生态化大型软件平台关键技术研发及应用(2018GZ0105,2018GZ0104)资助。

陈子豪(1995—),男,硕士生,主要研究方向为区块链、机器学习,E-mail:chenzihao838@163.com;李 强(1963—),男,副教授,主要研究方向为嵌入式程序设计、移动云计算、大数据,E-mail:liq@scu.edu.cn(通信作者)。

太坊等公有链环境下，一般使用的是 POX 类型算法。POX 系列算法目前主要有工作量证明(Proof of Work，POW)、权益证明(Proof of Stack，POS)和股份授权证明(Delegate Proof of Stack，DPOS)[5]等。POW 算法实质上是一种概率性的拜占庭协议[6]，一般应用于公有链，需要较多节点和较大的算力来维护；POS 算法生成区块的过程取决于节点所持有的数字货币量；DPOS 算法需要数字货币持有者选出一定数量(一般为 101 个)的区块生成者，且每隔一段时间会重新选举区块生成者[5]。POX 系列算法通过向链上投入代币、增加服务请求提案成本、放宽最终一致性确认的需求来达成共识。但在某些联盟链或私有链环境下，并没有必要在链上加入代币激励层。联盟链首选的 PBFT[7]共识算法能够让区块链完全脱离链上代币的奖励机制，并且启动节点数量少(最少 4 个)，共识效率较高，不需要大量算力维护。然而，在联盟链的应用(如 IBM 的 HyperLeger[8])中，没有链上代币奖励“矿工”维护区块链的共识安全，完全由全体共识节点共同执行三阶段的共识过程来保障区块链的安全性，这也使得当共识节点数量增多时，为了保证分布式一致性，网络中的通信次数与数据传输量将大大增加。因此，PBFT 算法的去中心化其实是一把双刃剑，以牺牲性能作为代价，这也使得以此共识算法搭建的区块链系统如果不能保持小规模的共识集群，其系统的网络通信成本将远远高于传统中心化系统的通信成本。系统的效率与去中心化程度类似于一个跷跷板的两端，是无法兼得的。因此，我们认为使用区块链(特别是联盟链)解决问题时应当是多中心化协调管理，而不是完全的去中心化，若权力下放至每一个节点，反而会导致效率越来越低[9]。目前，共识算法的评价指标各异，但一般均侧重于社会学角度的公平性和去中心化程度，经济学角度的能耗、成本与参与者的激励相容性，以及计算机科学角度的可扩展性(交易吞吐量、节点可扩展等)、容错性和安全性等[10]。如何结合具体需求和应用场景，针对特定的性能评价目标自适应地设计共识机制以达到优化算法的目的[11-12]，并寻找去中心化与共识效率的平衡点，将是区块链共识算法未来研究的热点。

根据以上背景和研究，本文将利用聚类算法 K-medoids 改进 PBFT 共识，提出 K-PBFT 共识算法，使其成为一个多中心化、多层次的区块链共识机制模型，在不损失安全性的前提下提高共识效率，减少共识过程的通信次数，以使得区块链模型能够适用于更广泛的应用场景中。

2 准备知识

2.1 区块链技术

区块链技术是由多种已存在的计算机技术(P2P 网络、密码学等)组合形成的具有去中心化特征的记录技术。目前，区块链在学术上尚没有统一的定义，百度百科上对区块链的定义是：区块链是分布式数据存储、点对点传输、共识机制、加密算法等计算机技术的新型应用模式。

区块按照时间戳[13]的顺序链接形成区块链，可以将其看作记录整体状态变化的一种方式[14]。一个区块主要由区块头、区块主体及若干装有请求服务的交易单组成。区块链使用哈希算法[15]对区块头进行摘要计算，得到的哈希值作为区块之间的逻辑链接指针。区块链的链式结构如图 1 所示。

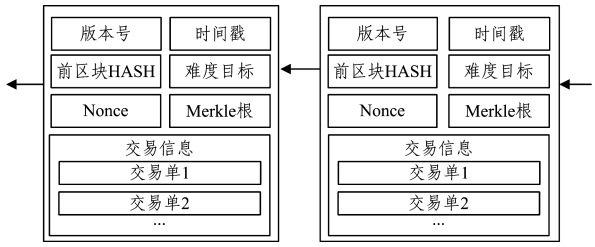

图 1 区块链结构

Fig. 1 Blockchain structure

2.2 密码学基础

密码学是区块链系统最重要的组成部分，是实现区块链的基础技术。

非对称加密是目前普遍采用的一种通信加密算法。区块链中的交易单需要依靠非对称加密来验证合法性。该类加密算法需要两个密钥：公钥和私钥。在这对密钥中，对外公开的密钥、被称为公钥，不对外公开的、仅自己持有的秘钥被称为私钥，且公钥与私钥是唯一配对的。如果用公钥对数据进行加密，只有用对应的私钥才能解密；如果用私钥对数据进行加密，那么只有用对应的公钥才能解密。利用非对称加密技术，可以保证区块链平台上通信各方安全、可信地传递信息。

数字签名技术是利用非对称加密技术保证数据的完整性、不可否认性和防伪性的一种密码学技术。在区块链系统中，通常使用数字签名技术对区块、交易单的相关信息进行签名，其是保障区块链数据安全性的基石之一[16]。

假设 A 网络节点与 B 网络节点之间进行数据传输，B 节点收到署名为 A 的数据消息，在这个过程中将存在两个问题：(1)B 节点如何确定收到的数据消息就是 A 节点发送来的；(2)B 节点收到的数据消息是否完整可信、未被其他恶意监听者在网络传输过程中篡改过。利用数字签名可以解决这两个问题，详细的解决过程如图 2 所示。

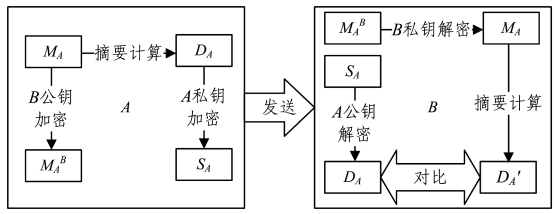

图 2 数字签名原理

Fig. 2 Principle of digital signature

A 对想要利用网络通信传输至 B 的数据消息内容 M_A 进行摘要计算，得到摘要 D_A；接着利用 A 的私钥加密 D_A 生成其数字签名 S_A；然后 A 利用 B 节点的公钥对 M_A 加密生成 M_A^B；最后将 M_A^B 和 S_A 打包后一起发送给节点 B。B 节点收到 M_A^B 和 S_A 之后，利用 A 的公钥对 S_A 进行解密，得到内容 M_A 的摘要 D_A；接着，B 利用自己的私钥对收到的 M_A^B 进行解密，得到数据消息的明文 M_A，再将 M_A 进行摘要计算得到摘要 D_A'；最后，验证 D_A' 与 D_A 的一致性来判断内容 M_A 是否由 A 节点发送且在网络传输中是否被篡改过。

2.3 实用拜占庭容错算法

实用拜占庭容错算法是在 1999 年由 Miguel Castro 和

Barbara Liskov 提出的一种共识算法。PBFT 是一种在分布式场景下解决状态机副本一致性的算法，可以在失效节点不超过$(n-1)/3$(n 为集群节点总数)的情况下保证共识集群的安全性和一致性[17]。在 PBFT 算法中，服务是具有确定性的，即相同状态下的操作在每个共识节点下执行所得到的结果都是相同的，并且每个共识节点在开始执行操作时，也必须拥有相同的状态[18]。共识集群中所有的操作都在某一视图状态(View)中进行，且每一个视图状态下只存在唯一主节点。副本节点由两种角色构成：主节点(leader)、从节点(follower)。

PBFT 算法的共识过程主要分为 3 个阶段：预准备、准备、确认。

(1)预准备阶段：主节点接收到请求服务消息并校验正确后，依据该请求服务消息生成预准备消息，并将其广播给各从节点。

(2)准备阶段：从节点收到主节点的预准备消息后，验证消息内容，确保消息内容在传输过程中没有遭到篡改。内容验证正确后，从节点会依据预准备消息生成准备消息并将其广播给所有副本节点。

(3)确认阶段：当节点收到至少$(2n+1)/3$ 个来自不同节点(包括自己)的准备消息且验证消息都真实有效时，该节点进入确认阶段，依据准备消息生成确认消息并将其广播给所有副本节点。同时，它也会陆续接收并验证来自其他节点的确认消息，当收到$(2n+1)/3$ 个有效确认消息(包括自己的)后，称该请求在这个节点上达到了 committed 状态。此时，只通过这一个节点，就能判断该请求服务已经被副本节点中的大部分节点验证通过。

当请求到达 commited 状态后，该请求就可以进入提交阶段，随后请求被所有副本节点执行。

PBFT 共识过程中主要的三阶段的数据传输流程(预准备、准备、确认)如图 3 所示。

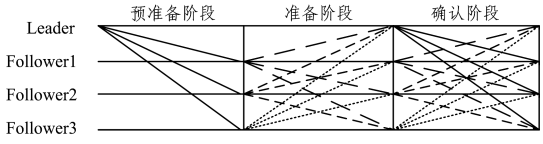

图 3　PBFT 算法的共识过程

Fig. 3　Consensus process of PBFT algorithm

根据上述过程，PBFT 算法虽然相较于其他共识算法不需要消耗大量算力资源，且共识速度较快，但也只适用于共识节点不多的情况。当大量节点加入区块链系统时，完全去中心化的特点使得所有共识节点需要共同进行三阶段共识，导致通信次数与数据传输量大大增加，很有可能造成网络堵塞或出现网络风暴。因此，下文将利用聚类算法 K-medoids 改进 PBFT 共识，提高 PBFT 算法在大量节点共同参与共识时的共识效率。

2.4　聚类算法

目前，聚类算法有许多种，最常见的是 K-means[19] 和 K-medoids[20] 算法。这两种算法的聚类过程比较相似，都是以待聚类集合中点在某些特征空间中的距离作为评判相似度的标准，来达到聚类的目的。其中，K-means 算法属于均值聚类算法，在聚类过程中选择的聚类中心是同一簇内一群真实节点的特征均值，但是这也使得该算法在没有定义平均值抽象点的情况下不能使用，且少量噪音或孤立点的加入会对簇内的聚类均值中心造成很大影响。而 K-medoids 算法放弃了 K-means 中的均值策略，采用实际的节点作为聚类中心，从而减小了 K-medoids 对孤立节点的敏感性，提高了聚类的准确性。本文将聚类算法与区块链相结合，把区块链共识过程的主节点当作各簇的聚类中心。因此，采用 K-medoids 聚类算法进行研究更加合理有效。

K-medoids 算法属于基于划分的聚类算法，其具体过程如下。

假设有 n 个 p 维特征的对象节点作为聚类数据集。用 $v_i=\{v_{i1},v_{i2},v_{i3},\cdots,v_{ip}\}$($i=\{1,2,\cdots,n\}$)表示第 i 个节点映射在 p 维特征空间 R^p 的一个坐标值。可将上述所有的节点数据集 $V=\{v_1,v_2,\cdots,v_n\}$表达为：

$$V=\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}=\begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1p} \\ v_{21} & v_{22} & \vdots & v_{2p} \\ \vdots & \vdots & & \vdots \\ v_{n1} & v_{n2} & \cdots & v_{np} \end{bmatrix} \tag{1}$$

K-medoids 算法使用欧氏距离来表达两个节点的相似度，利用式(2)来表示两个节点 v_i 和 v_k 的距离函数。

$$d(v_i,v_k)=\| v_i-v_k \|=\sqrt{\sum_{j=1}^{p}(v_{ij}-v_{kj})^2} \tag{2}$$

K-medoids 的目标是将数据集聚类为 K 个簇，每个簇用 V_i 表示。每个 V_i 有一个 c_i 代表聚类中心节点，中心节点集用 $c=\{c_1,c_2,\cdots,c_k\}$来表示。属于 V_i 簇内的节点 v_i 与同一簇内的其他节点的相似度较高，与其他簇中的节点的相似度较低。$\forall i\neq j,V_i\cap V_j=\varnothing$，有 $V=\bigcup\limits_{1\leqslant k\leqslant K}V_k$，用$|V|$表示整个数据集合的样本个数 n，$|V_k|$表示第 k 个聚类中的节点个数。为了方便表达这种算法策略，引入隶属函数：

$$u_{ik}=\begin{cases}1, & (d(v_i,c_k)=\min\{d(v_i,c_q)\}) \\ 0, & (d(v_i,c_k)\neq\min\{d(v_i,c_q)\})\end{cases},1\leqslant q\leqslant K,v_i\notin c \tag{3}$$

于是，我们可以定义 K-medoids 算法的目标函数为：

$$T(c)=\sum_{k=1}^{K}\sum_{i=1}^{n}u_{ik}d(v_i,c_k) \tag{4}$$

并对算法过程进行数学建模，具体如下：

$$\begin{aligned}&\min\ T(c)\\&\text{s. t. } C=\{V_1,V_2,\cdots,V_K\}\\&c=\{c_1,c_2,\cdots,c_k\}\\&\forall i\neq j,V_i\cap V_j=\varnothing\\&V=\bigcup_{1\leqslant k\leqslant K}V_k\end{aligned} \tag{5}$$

K-medoids 算法在递代时需要反复计算簇内所有节点到其他节点的欧氏距离，增加了算法的计算复杂度[21]。结合区块链的应用场景，参与区块链共识的节点集群数量不稳定，硬件条件参差不齐，且运行的环境条件不确定，因此在某些场景下，高层监督者选择出初始簇中心节点后，可能并不希望浪费共识集群的计算资源来频繁更换簇内中心。根据区块链的应用特点，下文将对 K-medoids 算法进行改进。

3 基于 K-medoids 算法的 PBFT 共识机制

3.1 改进的 K-medoids 算法

为了让 K-medoids 算法更加适用于区块链模型，本文对其进行改进。将 K-medoids 算法运用于区块链环境中，初始的 K 个中心节点并不是从所有样本中随机选择，而是通过上层监督者或者运营商进行严格审查，筛选出社会公信度较高的节点成为初始化的聚类中心节点。在这种情况下，我们不希望再对中心节点进行调整。改进的 K-medoids 算法更好地控制了簇中心节点的更换概率，其详细流程如下。

步骤 1 在数据集 V 中选取 K 个节点作为初始簇中心节点，并初始化惰性系数 $P(0<P<1)$。

步骤 2 根据式(2)、式(3)计算每一个节点距离 K 个簇中心的距离函数和隶属函数。

步骤 3 对每个簇中的所有节点，计算其到簇内所有其他点的距离和，将拥有距离和最小值的节点作为待定新聚类中心节点，得到待定新聚类中心节点集。

步骤 4 如果待定新聚类中心节点集与原中心节点集相同，算法终止并返回最终聚类结果；如果待定新聚类中心节点集与原中心节点集不完全相同，中心节点集将在 P 概率下替换为步骤 3 中的待定新聚类中心节点集并返回步骤 3，在 $(1-P)$ 概率下算法终止，返回最终聚类结果。

与传统 K-medoids 算法不同，改进 K-medoids 算法加入了惰性系数 P，当不希望初始化聚类中心被替换时，可在初始化时设置 P 为接近 0 的常数；当希望聚类算法能够真实反映集群的聚类情况时，可以设置 P 为接近 1 的常数。当 P 等于 1 时，改进 K-medoids 算法与传统 K-medoids 算法完全相同；当 P 等于 0 时，初始化节点直接作为聚类中心，不再进行聚类递代。这样可以更好地控制节点运行聚类算法时消耗的算力资源，并能够方便选择区块链环境下骨干共识集群的节点集，提高了区块链平台的可控性。

3.2 K-PBFT 共识机制

K-medoids 算法与 PBFT 相结合，利用聚类算法对原 PBFT 中无差别的所有共识节点进行划分聚类并分层，将聚类后的簇中心节点作为该簇内所有节点的主节点，每个簇内的非聚类中心节点作为该簇内的从节点。本文将每个簇称为一个子共识集群，非聚类中心节点构成了 K 个主节点领导下的 K 个子共识集群，再将所有 K 个主节点的集合构成一个骨干共识集群。区块链模型的结构如图 4 所示。

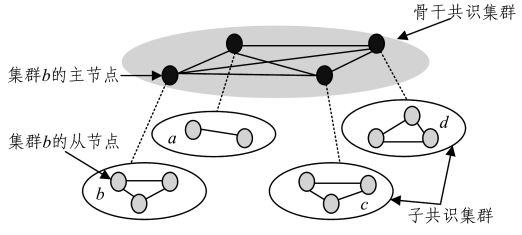

图 4 聚类后的区块链模型

Fig. 4 Clustered blockchain model

K-PBFT 共识算法的主要流程阶段如下。

(1)子共识集群共识阶段：每个节点将请求发送给它所属簇的主节点(簇中心节点)。主节点在接收一段时间的请求后，将若干请求打包成一个区块，然后将该区块广播给它所属的子共识集群进行一次 PBFT 共识。

(2)骨干集群共识阶段：区块通过子共识集群的共识验证过程后，将在骨干共识集群进行二次 PBFT 共识确认。骨干共识集群内的 K 个节点通过轮询的方式轮流广播已通过所属子共识集群共识验证的区块进行共识。轮询公式(6)为一个取模运算，其中 N 为下一个出块的节点编号，L 为现在区块链末尾区块的块编号，$|R|$ 为骨干共识集群中的节点个数。每个骨干集群中的节点都为某个子共识集群的主节点，根据式(6)可知当前轮询到本节点时，本节点会将这段时间内收集的通过了所属子共识集群共识的请求打包成一个区块广播给骨干集群的所有节点进行 PBFT 共识；若这段时间内子共识集群没有新的请求，那么节点将打包一个空区块发送给骨干集群共识上链，以推动轮询过程。

$$N=L \bmod |R| \tag{6}$$

(3)提交阶段：当区块经过骨干共识集群的共识后，所有主节点将对此区块进行数字签名并收集来自其他主节点的数字签名，表示认同此区块的真实性和有效性；然后，将此骨干共识集群的数字签名集合附带区块本身打包成提交消息广播至其所属的子共识集群内的所有从节点，表示可以将此区块进行上链操作。至此，任意子共识集群的节点都会收到此区块的提交消息。

(4)执行阶段：从节点在收到来自主节点的提交消息后，对区块附带的数字签名集合进行验证，可判断出该区块是否经过了骨干共识集群的共识验证。若验证失败，则可认为该从节点所属的主节点存在恶意行为，可向区块链运营者或监督者举报此次违法操作，达到从节点向上监督的作用；若验证成功，则可以执行此区块的请求内容，并将区块记录上链。

K-PBFT 算法共识过程的数据传输流程如图 5 所示(省略了骨干集群的轮询过程)。

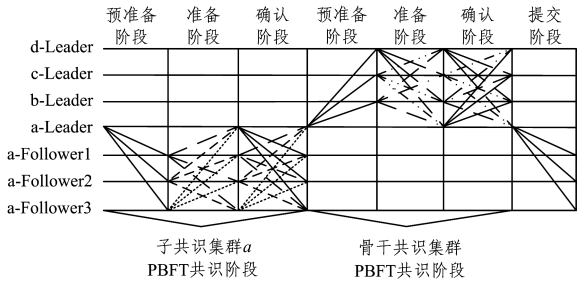

图 5 改进后的 K-PBFT 的共识过程

Fig. 5 Consensus process of Improved K-PBFT

K-PBFT 共识模型通过 K-medoids 聚类算法，将传统 PBFT 的全局去中心化共识改进为分层次的多中心化共识。在选择 K-medoids 聚类算法的样本特征空间坐标系来计算欧氏距离时，可以根据算法的应用场景考虑选择节点间网络延迟、地理空间位置、机器硬件条件等特征空间，使聚类算法更好地将相似度较高的节点划分为一个簇；或结合多种特征形成高维特征空间坐标系，使通过聚类算法选择出的簇中心节点是簇内硬件环境和网络环境等多种特征空间下评价均较良好的节点，以保证在同属一簇的从节点之间以及各个簇中心节点之间的各项特征差异性较小，进而获得更高的共识效率。根据区块链的实际应用场景，可以考虑使用 3.1 节的改进

K-medoids 算法，视情况初始化惰性系数 P，进而提高平台可控性，并降低 K-medoids 计算簇中心所消耗的算力资源。

3.3 K-PBFT 的实际应用场景

当前，在某些区块链场景中需要政府部门的服务器作为各簇中心初始节点，此时平台运营方可能并不想因在各方面特征评价优于政府节点的某一私人用户节点加入区块链平台而更换簇中心，使政府节点失去参与骨干共识集群共识的机会，进而导致政府丧失对多中心化区块链系统的宏观可控性。因此，在初始化时，可以将改进 K-medoids 算法中的惰性系数 P 设置成 0，以防止簇中心节点被更换。例如：在音视频类数字版权区块链场景中，需要加入具有上层监督作用（如版权局、网信办等）的节点才能更好地维护网络上的音视频创作内容的版权不被侵权，但将管理权限完全交给上述为数不多的节点进行中心化管理也存在着巨大的风险（如黑客网络攻击、管理人员泄露版权信息隐私等）。因此，可以设置这些具有较高社会公信度的节点担任 K-PBFT 中的骨干集群节点，将普通用户当作从节点，在聚类后进行多中心化共识，既能让从节点参与共识过程，从而监督骨干集群中主节点的行为，又能让具有社会公信度的骨干共识集群保留对区块链平台的宏观可控性。

同时，在某些无组织、用户自发维护的区块链应用场景中，如全球各高校的生物系想要联合组建一个专业论文数据区块链，由于链上不存在上层的监督者，各高校之间在区块链上应当拥有完全平等的地位。因此，可以将改进 K-medoids 算法中的惰性系数 P 设置为接近 1 的常数。这样做可以保证在选择了适当特征（如节点网络环境、硬件存储条件等）作为聚类评判标准的条件下，选择出具有一定代表性的节点作为各个簇的中心节点。在这种参与节点数量较多且不固定的场景下，相比传统 PBFT 的全局无差别共识，先对节点进行聚类与分层，然后应用 K-PBFT 共识的方法，能提高区块链平台处理请求的效率。

4 仿真实验

本文将分别从单次共识耗时与单次共识过程通信次数两个方面对改进后的区块链共识算法 K-PBFT 与传统 PBFT 共识算法进行对比实验。

4.1 共识耗时实验

实验采用 NS2[22] 配合 GT-ITM 拓扑生成器[23]，模拟底层的物理网络情况。图 6 为拓扑生成器生成的一个含有 9 个聚类中心节点且每个聚类含有 10 个节点的网络拓扑图，图 7 为拓扑生成器生成的一个含有 20 个聚类中心节点且每个聚类含有 10 个节点的网络拓扑图。

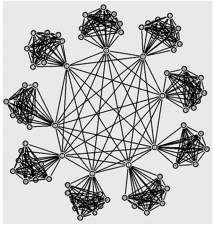

图 6　9 个聚类/10 从节点

Fig. 6　9 clusters/10 nodes

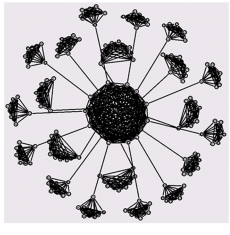

图 7　20 个聚类/10 从节点

Fig. 7　20 clusters/10 nodes

共识耗时实验中生成 1000 个节点，节点间的欧氏距离用网络延迟代表，共分成 10 个子共识集群，每个子共识集群有 99 个从节点。由于需要对比两种共识算法的共识耗时，因此需要设置每个节点与全网所有节点的延迟。本文根据骨干集群中的节点一般属于网络带宽条件较好的节点，以及簇内节点间的网络延迟比簇间节点的网络延迟低等聚类情况，分别设置如下模拟延迟。

(1) $delay(v_i,v_j)$：$\exists k,1\leqslant k\leqslant K,(v_i\cup v_j)\subset V_k,i\neq j$，属于同一子共识集群的从节点之间的延迟控制在 30～50 ms 之间。

(2) $delay(v_i,v_j)$：$\forall k,1\leqslant k\leqslant K,(v_i\cup v_j)\not\subset V_k,i\neq j$，不属于同一子共识集群的从节点之间的延迟控制在 200～250 ms 之间。

(3) $delay(c_i,c_j)$：$(c_i\cup c_j)\subset c,i\neq j$，骨干共识集群节点间的延迟控制在 50～100 ms 之间。

(4) $delay(v_i,c_j)$：$(v_i\cup c_j)\subset V_j$，骨干共识集群中的主节点与其同属于一个子共识集群的从节点间的延迟控制在 10～30 ms 之间。

(5) $delay(v_i,c_j)$：$c_j\subset V_j,v_i\not\subset V_j$，骨干共识集群中的主节点与其不同属于一个子共识集群的从节点间的延迟控制在 100～120 ms 之间。

实验中采用 K-medoids 聚类算法中的节点相似度来计算两个节点间的网络延迟，距离公式为：

$$d(v_i,v_j)=delay(v_i,v_j) \tag{7}$$

实验中不考虑单个节点在接收到信息后的处理耗时与单机上的资源利用、CPU 处理速度、磁盘读写速度、网络拥塞等因素，并假设同一时间点单机上所有消息的接收与发送都并行发生，只计算共识流程开始到共识流程结束的耗时。为了对比评价两种模型的共识耗时，进行了 20 次的独立对比实验（忽略骨干共识集群的轮询过程）。每次实验按照上述模拟控制条件随机生成节点之间的延迟，然后记录 K-PBFT 共识算法与普通 PBFT 共识算法进行单次共识的耗时，对比结果如图 8 所示。

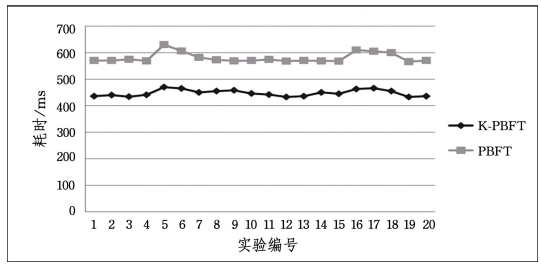

图 8　共识过程的耗时对比

Fig. 8　Consensus process time-consuming comparison

通过 20 次重复对比实验计算出 PBFT 单次共识的平均耗时为 580.7 ms，K-PBFT 单次共识的平均耗时为 447.7 ms。实验结果表明，K-PBFT 算法相较于 PBFT 算法，单次共识耗时缩短了 20%。

由上述实验分析可知，传统 PBFT 算法的每次共识都需要全局范围的所有共识节点进行一次三阶段共识过程，这会导致当大量网络节点都作为共识节点参与到 PBFT 共识过程中时，极易因节点间频繁广播共识消息造成网络堵塞。而利用 K-medoids 算法改进后的 K-PBFT 算法通过节点聚类的方

法，将网络延迟作为评价特征，对共识节点进行筛选分层与分簇，然后在簇间与簇内相对较优的网络环境下进行较小范围内的多中心化共识。实验结果表明，在共识节点数量较多的情况下，以共识节点间的网络延迟作为特征空间的 K-PBFT 算法相较于 PBFT 有效缩短了共识过程的耗时，提高了共识效率。

同时，K-PBFT 的分层次、多中心化结构，使得单个子共识集群之间以及子共识集群与骨干共识集群之间都可以并行地执行部分共识过程。因此，在真实网络环境下，K-PBFT 算法能更高效地处理多个子共识集群并发提出的请求。

4.2 通信次数的对比

为了验证改进后的共识算法模型在通信次数方面比传统的 PBFT 共识算法优秀，本文设置了共识过程通信次数对比实验。

(1)计算 PBFT 单次共识的通信次数

假设现在平台上有 $n(n>3)$个节点参与 PBFT 共识。可以利用共识算法计算出一次完整的 PBFT 共识需要的通信次数(这里只计算预准备阶段、准备阶段和确认阶段 3 个主要阶段的通信次数)。

在预准备阶段，主节点收到客户端请求后，处理请求，接着将预准备消息发送给所有从节点。本阶段，共识网络中的通信次数为$(n-1)$。在准备阶段，从节点收到主节点的预准备消息之后，对预准备消息进行验证，将准备消息发送给除自己之外的所有共识节点。本阶段，共识网络中的通信次数为$(n-1)\cdot(n-1)$。在确认阶段所有节点对收到的准备消息进行验证，当验证结果为真时发送确认消息给除自己以外的所有节点。本阶段，共识网络中的通信次数为 $n\cdot(n-1)$。

由此可以计算得到：完成一次 PBFT 共识过程需要的总通信次数为$(n-1)+(n-1)\cdot(n-1)+n\cdot(n-1)$，化简得到 $2n\cdot(n-1)$。

总共的通信次数 Z_1 = PBFT 单次共识过程的通信次数，将上述推导过程中各个阶段的通信次数相加并化简可得：

$$Z_1=2n\cdot(n-1) \tag{8}$$

(2)计算 K-PBFT 单次共识的通信次数

同样假设现在平台上有 $n(n>3)$个节点参与 K-PBFT 共识，并设置聚类数目为 K，默认每个子共识集群的共识节点个数为 n/K，骨干共识集群的节点个数为 K。

下面利用 3.2 节中改进后的共识算法计算出一次完整的 K-PBFT 共识需要的通信次数。

区块被提交后，会先在所属的子共识集群进行一次 PBFT 共识过程。由上面的 PBFT 过程通信次数分析结果(式(8))可知，子共识集群中的通信次数为 $2n/K\cdot(n/K-1)$。子共识集群通过共识后，主节点会在骨干集群中对此区块再次进行 PBFT 共识，在骨干集群中共识的通信次数为 $2K\cdot(K-1)$。在骨干集群共识验证通过后，所有骨干集群中的节点都会将此区块广播至其所属的子共识集群中，以通知此区块已经通过共识过程，可以上链存储，此过程的通信次数为$(n/K-1)\cdot K$。

由此可以计算得到，完成一次 K-PBFT 共识过程需要的总通信次数为 $2n/K\cdot(n/K-1)+2K\cdot(K-1)+(n/K-1)\cdot K$。

总共的通信次数 Z_2 = K-PBFT 单次共识过程的通信次数，由上述推导过程可得：

$$Z_2=2\frac{n}{K}\cdot\left(\frac{n}{K}-1\right)+2K\cdot(K-1)+K\cdot\left(\frac{n}{K}-1\right) \tag{9}$$

令两种算法的通信次数之比为 Y，则由式(8)和式(9)可知 $Y=Z_1/Z_2$：

$$Y=\frac{2n\cdot(n-1)}{2\frac{n}{K}\cdot\left(\frac{n}{K}-1\right)+2K\cdot(K-1)+K\cdot\left(\frac{n}{K}-1\right)} \tag{10}$$

设 n 与 K 为自变量，Y 的曲面函数图如图 9 所示。

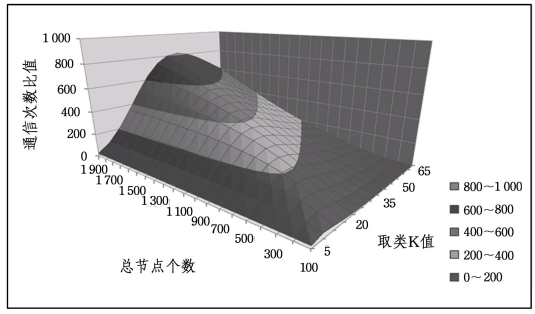

图 9 共识过程通信次数比的曲面图

Fig. 9 Consensus process communication times ratio surface map

由图 9 可知，比值的曲面函数呈拱形。将 n 看作常数，可以通过令 Y 的导数为 0 来求解一个四元一次方程，得到 Y 取最大值时 K 的取值为两个不等实根(一个正数、一个负数)和一对共轭虚根解。由于解此方程的过程太过冗杂，这里就不列出求解过程，只给出实验结果：由于式(10)中的 n 和 K 都为真实环境下的取值，因此不存在小数形式，用 int( )表示取整操作。当共识节点总数为 n 时，Y 大致在 $K=\mathrm{int}(\sqrt{n})$时取得最大值，此时 K-PBFT 相较于 PBFT 取得了最佳通信次数比。

n 等于 1000 时，PBFT 算法的三阶段共识总共需要发送 1998000 次消息，达到百万数量级。在 K 个聚类的情况下，K-PBFT 的通信次数如表 1 所列。

表 1 在 n 为 1000 时 K 个聚类下 K-PBFT 的通信次数

Table 1 Number of communications of K-PBFT under K clusters when n is equal to 1000

聚类 K 值	5	10	20	25	40	50
通信次数/次	80635	20970	6640	5295	5280	6610

由表 1 可知，n 等于 1000 时，K-PBFT 与 PBFT 相比，最佳能降低 3 个数量级的通信次数。

经过上述实验分析，K-PBFT 能够有效减少单次共识过程的通信次数。利用 K-medoids 算法进行聚类，每个聚类都是一个去中心化的子共识集群，再由各子共识集群的主节点构成一个上层的骨干集群。与传统 PBFT 每次共识都需要所有节点间进行数据传输推进共识过程不同，新共识算法将节点聚类分层后在较小范围内进行共识，有效减少了单次共识所需要的通信次数。同时，在区块链平台的共识节点数量庞大的情况下，也不再需要每个共识节点都与所有其余节点之间保持 P2P 网络连接，只需与同一子共识集群中的节点进行网络连接即可，减少了建立与保持网络连接在单个机器上的资源消耗。

结束语　本文提出了一种基于 K-medoids 改进的 PBFT 区块链模型，采用聚类算法将去中心化的 PBFT 共识算法改进为一个多中心化的共识算法，以使改进后的 K-PBFT 相较于 PBFT 更加适用于大规模共识节点共同参与且不需要代币激励的联盟式区块链应用场景。同时，根据不同的应用场景，使用者可以选择或者结合不同的特征，自定义 K-PBFT 算法中的特征空间坐标系，进而更好地聚类与划分不同场景下区块链系统中的共识节点，以达到更佳的共识效率。共识耗时与共识通信次数的实验结果表明，当选择了合适的聚类特征时，改进后的算法 K-PBFT 在共识效率与共识过程通信次数方面都优于传统 PBFT 算法。同时，本文还通过加入惰性系数 P 改进了 K-medoids 算法，使 K-PBFT 的共识节点聚类过程更加可控。

目前，K-PBFT 算法模型仍处于试验阶段。未来，我们会继续研究在共识节点总数不确定的情况下，K-PBFT 算法如何动态控制各个簇的节点数量大小，以达到更好的共识效率；并进一步研究不同场景下共识节点在哪些不同特征结合的高维特征空间坐标系下进行聚类后的共识效率更佳，为该模型能够在大规模共识节点参与的真实联盟式区块链中得到运用打下基础；也会考虑将两层次的 K-PBFT 衍生至多层次化，以优化联盟式区块链系统在“海量”网络节点参与共识的情况下的共识效率。

参 考 文 献

[1] SWAN M. Blockchain: Blueprint for a New Economy[M]. USA: O'Reilly Media Inc, 2015: 45-68.

[2] 中国人民银行. 中国人民银行数字货币研讨会召开[OL]. http://www.pbc.gov.cn/goutongjiaoliu/113456/113469/3008070/index.html.

[3] YUAN Y, WANG F Y. Blockchain: The State of the Art and Future Trends[J]. Acta Automatica Sinica, 2016, 42(4): 481-494. (in Chinese)
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.

[4] LAMPORT L, SHOSTAK R, PEASE M. The Byzantine generals problem [J]. ACM Transactions on Programming Languages and Systems, 1982, 4(3): 382-401.

[5] Delegated Proof-of-Stake Consensus [EB/OL]. [2018-07-10]. https://bitshares.org/technology/delegated-proof-of-stake-consensus/.

[6] 邹均. 区块链技术指南[M]. 北京：机械工业出版社，2016: 109-128.

[7] CASTRO M, LISKOV B. Practical byzantine fault tolerance and proactive recovery[J]. Acm Transactions on Computer Systems, 2002, 20(4): 398-461.

[8] ANDROULAKI E, BARGER A, BORTNIKOV V, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains[C]// Proceedings of the Thirteenth EuroSys Conference. ACM, 2018: 30.

[9] 李明轩. 区块链投资人李明轩：区块链通过多中心化机制解决传统互联网问题[EB/OL]. http://www.sohu.com/a/271184292_100133330.

[10] YUAN Y, NI X C, ZENG S, et al. Development Status and Prospect of Blockchain Consensus Algorithm[J]. Acta Automatica Sinica, 2018, 44(11): 2011-2022.

[11] NI X, ZENG S, HAN X, et al. Organizational management using software-defined robots based on smart contracts[C]// 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018: 274-279.

[12] WANG S, YUAN Y, WANG X, et al. An overview of smart contract: architecture, applications, and future trends[C]// 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018: 108-113.

[13] IRVING G, HOLDEN J. How blockchain-timestamped protocols could improve the trustworthiness of medical science[J]. F1000 Research, 2016, 5: 222.

[14] PAPADOPOULOS G. Blockchain and Digital Payments: An Institutionalist Analysis of Cryptocurrencies [M]. Academic Press, 2015: 153-172.

[15] DADDA L, MACCHETTI M, OWEN J. The design of a high speed ASIC unit for the hash function SHA-256 (384, 512)[C]// Proceedings of the conference on Design, Automation and Test in Europe. IEEE Computer Society, 2004.

[16] KATZ J, LINDELL Y. Introduction to Modern Cryptography: Principles and Protocols[M]. Chapman and Hall/CRC, 2014: 207.

[17] BRACHA G, TOUEG S. Asynchronous consensus and broadcast protocols[J]. Journal of the Acm, 1985, 32(4): 824-840.

[18] REITER M K. A Secure Group Membership Protocol[J]. IEEE Transactions on Software Engineering, 1996, 22(1): 31-42.

[19] MAO D H. Improved Canopy-Kmeans algorithm based on MapReduce [J]. Computer Engineering and Applications, 2012, 48(27): 22-26.

[20] PATTABIRAMAN V, PARVATHI R, NEDUNCHEZIAN R, et al. A Novel spatial clustering with obstacles and facilitators constraint based on edge detection and k-medoids[C]// International Conference on Computer Technology and Development, 2009(ICCTD'09). IEEE, 2009: 402-406.

[21] MA Q, XIE J Y. K-medoids clustering algorithm based on particle computing [J]. Computer Applications, 2012, 32(7): 1973-1977.

[22] ISSARIYAKUL T, HOSSAIN E. Introduction to Network Simulator 2 (NS2) [M] // Introduction to Network Simulator NS2. Springer, Boston, MA, 2012: 21-40.

[23] CALVERT K, EAGAN J, MERUGU S, et al. Extending and enhancing GT-ITM[C]// ACM SIGCOMM Workshop on Models, Methods and TOOLS for Reproducible Network Research. ACM, 2003: 23-27.