一种应用于边缘计算的区块链分片方案

王珺，马建炜，罗金喜
（南京邮电大学通信与信息工程学院，江苏 南京 210003）

摘　要：边缘计算的数据安全性低和隐私性差等问题制约了边缘计算的发展，区块链可以利用自身的难以篡改性为边缘计算场景中的数据提供安全保障，同时利用可追溯性保护隐私，但是区块链的扩展性瓶颈成为其应用于边缘计算领域的障碍。针对区块链应用于边缘计算时无法满足大量节点同时处理数据的需求的问题，提出了一种符合边缘计算场景需求的双层分片方案，用改进的 *K*-means 算法实现节点基于地理位置的分片，并结合权益委托证明（DPoS, delegated proof of stake）与实用拜占庭容错（PBFT, practical Byzantine fault tolerance）的思想设计了一种局部区块链网络共识（LBNC, local blockchain network consensus）算法达成片内共识，通过多分片并行处理交易的方式提高系统能容纳的节点数量。仿真结果表明，所提方案比 PBFT 有更低的时延和更高的吞吐量，并且总吞吐量随分片数量增加。

关键词：区块链；分片；边缘计算；共识
中图分类号：TP309
文献标志码：A
doi: 10.11959/j.issn.2096−3750.2023.00333

A blockchain sharding scheme in edge computing

WANG Jun, MA Jianwei, LUO Jinxi
School of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Abstract: The low security and poor privacy of the data in edge computing restrict the development of edge computing. Block chains can provide security for data in edge computing using their own tamper resistance, while protecting privacy by use of traceability. But the bottleneck of blockchain's scalability has become a barrier to their application in the field of edge computing. To solve the problem that blockchain can not meet the needs of a large number of nodes to process data at the same time when applied to edge computing, a two-layer sharding scheme was presented, which meets the needs of edge computing scenarios. Geographic location-based partitioning of nodes was implemented using the improved *K*-means algorithm, and a local blockchain network consensus (LBNC) algorithm was designed based on the idea of delegated proof of stake (DPoS) and practical Byzantine fault tolerance (PBFT). Simulation results show that the proposed scheme has less delay and higher throughput than those of PBFT, and the total throughput increases with the number of shards.

Key words: blockchain, sharding, edge computing, consensus

0　引言

随着物联网[1-3]、大数据[4]、云计算[5]、机器学习[6]、人工智能等技术的进步，越来越多的应用将数据放到云服务器上进行计算或存储，以解决智能终端等设备存储容量有限、计算速度不足的问题[7]。因为边缘服务器更接近用户，将智能终端等设备的数据放入边缘服务器以进行计算或存储能获得比

收稿日期：2022−10−25；**修回日期：**2023−03−12
通信作者：王珺，wang_jun@njupt.edu.cn
基金项目：江苏省重点研发计划（No.BE2020084-5）；江苏省研究生科研与实践创新计划（No.46006CX21732）
Foundation Items: The Key Research and Development Program of Jiangsu Province (No.BE2020084-5), The Postgraduate Research and Practice Innovation Program of Jiangsu Province (No.46006CX21732)

云计算更低的时延，因此边缘计算已经成为云计算框架的有力补充。但边缘数据中心的数据容易丢失和泄露，数据的保密性和完整性无法保证[8]。区块链是基于对等网络的去中心化分布式账本数据库，其账本开放和难以篡改的特点可以解决边缘计算面临的安全问题[9]。

虽然区块链的优点使得它拥有广泛的应用前景，但它存在扩展性较差的问题。因此，面对不同的场景，设计对应的区块链扩容方案，提高区块链的性能，是区块链的热门研究方向[10-12]。针对目前区块链无法满足边缘计算场景下众多节点的吞吐量和时延需求的情况，本文提出了一种双层分片方案，该方案将节点按地理位置进行分片后以分片为单位处理交易，分片内的节点只需要验证所在分片的交易，存储属于该分片的局部区块链，以此减小节点运行区块链的计算和存储压力，提高区块链的吞吐量，在云端对局部区块链进行整合再验证，形成全局区块链，保证全局统一和安全。本文的主要贡献如下。

1) 提出了对应边缘计算 3 层框架的区块链双层分片方案，提高区块链吞吐量，降低节点计算和存储压力，使更多资源有限的边缘节点能加入区块链网络。

2) 改进 *K*-means 算法的初始簇心选择和节点划分，提高算法的收敛速度并保证各簇节点数量均匀，将节点按照地理位置进行划分，保留边缘计算低传输时延的优势。

3) 结合权益委托证明（DPoS, delegated proof of stake）与实用拜占庭容错（PBFT, practical Byzantine fault tolerance）的思想提出一种局部区块链网络共识（LBNC, local blockchain network consensus）算法以加快共识达成速度。

1　相关工作

区块链受到了产业界和学术界的广泛关注，目前区块链在医疗[13]、工业物联网[14]、智能电网[15]等领域发挥着重要作用。文献[16]设计了云边场景下的数据缓存方案，利用区块链防止数据被篡改，并且使用智能合约进行数据交易。文献[17]采用区块链保证边缘卸载过程中的数据完整性，然后采用遗传算法生成资源分配策略。这些研究都成功应用了区块链技术，提高了系统安全性，但是没有考虑如何优化区块链的性能和扩展性。

限制区块链广泛应用的一大问题是去中心化、安全性、可扩展性三者的均衡问题，即目前的区块链系统最多只能实现去中心化、安全性、可扩展性中的两点，难以满足全部。分片技术能在不牺牲中心化程度的同时提高区块链的性能，可以克服区块链的可拓展性问题，因此，分片技术逐渐成为区块链扩容的主流方法之一。文献[18]提出一种区块链分片的方法，将整个区块链网络的节点随机分配到各个分片中，各个分片可在各自的内部达成共识。虽然提高了整体的吞吐量，但是为了保证区块链的全局一致性，需要引入“状态块”存储交易摘要，并且需要处理跨片交易并不停地对验证节点进行随机分配。文献[19]将分布式数据库的分片方案引入区块链，设计了一个高效和安全的分片协议。但是其协议的实现依赖于如 Intel SGX 的可信执行环境[20]，这提高了网络的硬件成本。虽然以上分片方法能提高区块链的扩展性，但是这类方案从整个区块链网络角度考虑，将全球的区块链节点进行分片和调度，并且需要周期性地进行分片重配置等操作防止恶意节点的攻击，产生的数据迁移将消耗大量的带宽，不能满足能量和带宽资源有限的边缘节点的需要。

在边缘计算与区块链的融合框架中，一些研究者提出了与分片类似的扩容方案。文献[21]提出了一个基于区块链和边缘计算的分布式边缘数据共享框架，利用代理节点形成两层拓扑，使用权限表进行访问控制，实现了可接受开销情况下较高的安全性和可靠性。文献[22]在利用代理策略的同时优化了共识机制，并利用堆栈机制对任务进行排序，然后进行任务卸载，提高了区块链共识效率和边缘计算的任务卸载效率。以上代理策略的核心思想是将整个区块链网络进行划分，由代理节点领导局部区块链网络达成共识，再在代理节点之间达成全局的共识。但是这样的方案中代理节点趋于固定，权力较大，存在中心化的趋势，并且较少的研究说明全局一致性的保持、存储问题。

边缘计算的优势在于靠近用户节点，拥有更快的响应速度，在应用区块链时应当尽量保留这样的优势[23]。因此如何针对边缘计算场景应用区块链技术，保留边缘计算低时延、低能耗的优势，同时享受区块链提供的安全保障，是近年来的研究热点和本文着重研究的内容。本文根据边缘计算场景设计了一种双层分片方案，保证资源有限的边缘节点能加入区块链网络，设计了 LBNC 算法保证分片后的安全性。

2　边缘计算下的区块链的网络模型

本文考虑的是区块链在边缘计算框架下的应用场景。在该场景中，区块链网络节点包括云服务器、边缘服务器、终端设备和其他用户设备，终端设备利用区块链的智能合约上传收集的数据，边缘服务器利用区块链网络进行数据分发，用户可访问区块链查看数据传输记录，进行数据交易。

本文结合边缘计算 3 层框架[24]的特点和分片思想，设计了一个双层分片网络，用于解决传统区块链系统吞吐量偏低的问题。

分层区块链网络如图 1 所示。终端设备层与边缘服务器层对应于局部区块链网络，云服务器层对应全局区块链网络。终端设备层设备数量多，但大多资源有限，依靠边缘服务器提供服务，同时也容易产生数据篡改、丢失等安全问题，依靠区块链的透明性和可追溯性可以保障其安全。云服务器层资源丰富但时延较高，运行全局区块链为局部区块链提供资源和再次验证。

整个网络通过双层区块链实现共识的建立。局部区块链网络有多个，从而使得各网络能够并行地处理交易，并且只需要存储本地区块链网络产生的交易数据。全局区块链网络维护一条全局区块链为局部区块链提供保障。

2.1　局部区块链网络

边缘服务器与终端设备节点数量众多，因此本文采用改进的 *K*-means 算法将所有节点按照地理位置划分为多个局部区块链网络，即分片，原因如下。

1) 边缘计算场景中数据的使用价值存在地域局限性，如一条道路的交通数据主要用于该道路周围的交通管理和规划，地域临近的设备间数据重用的概率更高。

2) 边缘计算的目的是实现近端的卸载和存储而降低时延，因此终端设备几乎不会将数据托管到地域上遥远的边缘服务器，因此将远端的边缘服务器纳入该终端设备的区块链网络是没有价值的。

3) 通过根据地域分布划分出多个局部区块链网络而实现多分片的交易处理模式，可以解决系统的扩展性问题，即当有更多设备需要加入网络时，可以通过增多局部区块链网络的数目来避免网络中其他节点计算和存储压力的增加。

在区块链节点划分上，算力有限的终端设备作为轻节点，主要通过使用区块链平台完成合约调用。边缘服务器以及其他拥有算力和存储资源的节点以权重参数为依据选举成为委员会达成共识，产生的区块包含本地局部区块链网络的合约调用等记录，权重来源于节点诚实地参与区块转发、共识确认等工作。在局部区块链达成共识后将新产生的区块发给云服务器。

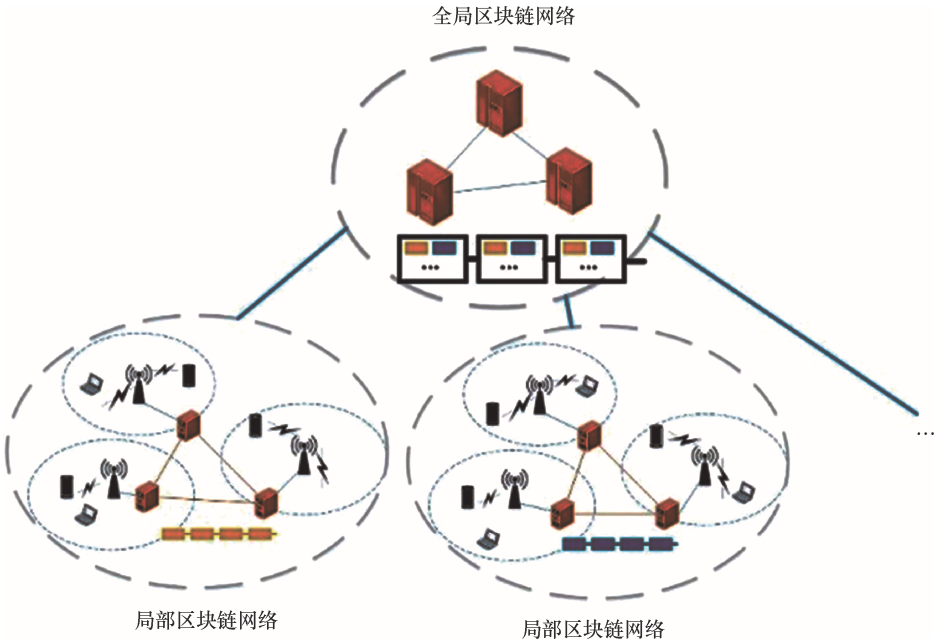


图 1　分层区块链网络

2.2　全局区块链网络

边缘计算与云计算相互补充，云服务器存储了整个区块链网络节点的注册等信息，这些信息通过云服务器间的区块链网络上链保证难以篡改性，各个云服务器都保留完整的全局区块链，以实现分布式存储保证安全。各个局部区块链达成共识产生新的区块后，云服务器将收到该区块并再次验证是否与全局区块链产生冲突。若产生冲突则对分片内恶意节点进行惩罚，若无冲突则将局部区块链的区块加入全局区块链。

3　一种双层分片共识协议

针对上述边缘场景下的双层区块链网络，本文设计了一种双层分片共识协议以达成共识，该协议分为 3 个阶段。

1) 初始化阶段

获取初始网络中所有节点的地理位置信息，根据 *K*-means[25]算法将初始网络划分为 k 个局部区块链网络，然后局部区块链网络为各个网络的节点设置初始权重。

2) 局部共识执行阶段

各局部区块链网络通过局部区块链网络共识处理交易生成区块。即先根据各节点的权重随机选取若干个节点组成委员会，委员会成员间运行 Tendermint[26]协议处理交易。

3) 全局共识执行阶段

在云服务器之间运行一个区块链网络，该网络将收集并打包所有局部区块链网络产生的区块，为最终全局安全性提供保障。

3.1　初始化阶段

在初始化阶段中，目的是将所有的边缘服务器以及其他参与共识的节点根据地理位置关系划分为 k 个局部网络，终端设备层的节点将被划分到其直接相连的边缘服务器所在的局部网络。而且局部区块链网络是半开放的，节点的相关交易都确认之后节点可以离开，当新的或离开的节点需要加入时，根据新的地理位置加入对应分片，其权重根据云服务器的全局区块链的信息进行转移。

在边缘计算网络中，可以通过无线网络定位技术[27]获得各节点地理位置信息，为了简化 *K*-means 算法的复杂度，该位置信息将被映射到二维平面中，节点 x 和节点 y 之间的距离为

$$d_{xy}=\|x-y\|_2^2 \tag{1}$$

K-means 算法会根据距离信息把给定的样本集合划分为 K 个簇。设样本集合被划分为 $\{C_1,C_2,\cdots,C_K\}$，*K*-means 算法的目标就是最小化平方误差之和 E。

$$E=\sum_{t=1}^{k}\sum_{x\in C_t}\|x-u_t\|_2^2 \tag{2}$$

其中，μ_t 为簇 C_t 的均值向量，由簇中所有节点的坐标计算获得，表示为

$$u_t=\frac{1}{|C_t|}\sum_{x\in C_t}x \tag{3}$$

为了让 *K*-means 算法适用于本文所提双层分片共识网络，需要对其进行一些改进。首先，初始簇心 μ_t 的选择对于 *K*-means 算法的收敛速度有着极大的影响，尤其是在节点数量庞大的情况下。而传统 *K*-means 算法通过随机的方式选取 K 个节点作为簇心 μ_t 不具备稳定性，因此本文在随机选择第一个簇心后选择与当前所有簇心的最小距离最大的点作为其他区域的簇心，以保证初始各个簇心尽量分散且均匀，初始簇心选择算法见算法 1。

算法 1　初始簇心选择算法

输入　位置信息 $V=\{x_1,x_2,\cdots,x_N\}$，聚类簇数 k

输出　簇心 $U=\{u_1,u_2,\cdots,u_k\}$

$u_1\leftarrow \text{randchoose}(V)$

for i=1,2,$\cdots$,k　do

　for　$x_j\in V$, $D(x_j)\leftarrow\min(d_{x_ju_t})\,t=1,2,\cdots,k_{\text{selected}}+1$ 。

　　$u_{k_{\text{selected}}+1}\leftarrow\max(D(x_j))$

　end for

end for

传统 *K*-means 算法最后划分出的各个簇中节点数量是完全随机的，因此可能出现节点数量的不均衡。为了均衡各局部区块链的计算和存储资源，本文将在原 *K*-means 算法的收敛流程中加入簇节点的数量控制，簇形成算法见算法 2，最终网络将由算法 2 划分成 k 个均匀的局部区块链网络。

算法 2　簇形成算法

输入　数据集 $V=\{x_1,x_2,\cdots,x_N\}$，聚类数 k，最大迭代次数 M，初始簇 $U=\{u_1,u_2,\cdots,u_k\}$

输出　簇划分 $C=\{C_1,C_2,\cdots,C_k\}$

$C \leftarrow \varnothing,\ t=1,2,\cdots,k,\ V' \leftarrow V$

while $V' \neq \varnothing$ do

$x \leftarrow \text{randchoose}(V')$，计算 $d_{xu_t}, t=1,2,\cdots,k$

$d_{xu_\gamma} \leftarrow \min(d_{xu_t}),\ t=1,2...k$

if $\left|C_\gamma\right| < \dfrac{N}{k}$

$C_\gamma = C_\gamma \bigcup \{x\}, V'_{\text{remove}}(x)$

else $d_{yu_\gamma} \leftarrow \max(d_{lu_\gamma}), l \in C_\gamma$

if $d_{xu_\gamma} < d_{yu_\gamma}$

$C_\gamma = C_\gamma \cup \{x\}, V'_{\text{remove}}(x), V'_{\text{add}}(y)$

end if

end if

end while

新得到的 $C=\{C_1,C_2,\cdots,C_k\}$ 重新计算簇心 $\{u_1,u_2,\cdots,u_k\}$，若算法不收敛（相邻簇心变化大于阈值）且迭代次数小于 M，则重复算法；否则算法终止。

3.2 局部共识执行阶段

共识算法使分布式的区块链网络中所有的全节点的状态一致，传统的工作量证明（PoW, proof of work）算法需要大量的算力用于计算，而边缘层（局部区块链网络）中的算力资源紧张，PoW 无法采用。因此本文设计了 LBNC 算法用于处理局部区块链网络中的交易，LBNC 算法首先用 PoS 选取成员构成委员会，而委员会间运行 Tendermint 处理交易，其他节点只需要验证委员会产出的区块。

3.2.1 委员会的选取

任意用户节点 i 根据权重 w_i 参与委员会选举，局部区块链网络的权重总和为 $W=\sum_i w_i$，用户 i 被选取为委员会成员的概率与 $\dfrac{w_i}{W}$ 成正比，通过可验证的随机函数（VRF, verifiable random function）[28] 在节点间无交互且无信任的情况下随机选择出 K 个节点组成委员会。

委员会成员选取算法见算法 3，输入参数中的 role 表示该节点在共识算法中的角色，分为主节点和副本节点，因为 VRF 输出的哈希值具有随机性，可能有多个节点同时被选取成为主节点，这时可以按投票权轮流担任主节点。输入参数中的 τ 决定了委员会成员规模，τ 越大，委员会的成员数量和总票数也就越大。

算法 3　委员会成员选取算法

输入　节点私钥 sk，随机种子 seed，预期成员数 τ，角色 role，用户权重 w，网络总权重 W

输出　< hash, π, j>

$\langle \text{hash}, \pi \rangle \leftarrow \text{VRF}_{\text{sk}}(\text{seed} \,||\, \text{role})$

$p \leftarrow \tau / W$

$j \leftarrow 0$

While $\text{hash}/2^{\text{hashlen}} \notin \left[\sum_{k=0}^{j} B(k;w,p), \sum_{k=0}^{j+1} B(k;w,p)\right]$ & $j < \sigma$ do

j++

end while

节点对于共识的影响力由其权重所决定，因此不同于传统的一票制，权重高的委员会成员可能拥有多个投票权，但不高于上限 σ。算法 3 中的输出 j 为 0，代表该节点在本轮没有被选入委员会，j>0 代表该节点是委员会成员，且投票权为 j。为了按比例选择用户，LBNC 算法以最小权重节点的权重为单位，将每个节点的权重划为分子用户。如果用户 i 拥有的 w_i 的权重，拥有权重最小的用户有 $w_{\min}$ 的权重，那么 i 用户拥有 $w_i/w_{\min}$ 个子用户。(i, j)代表用户 i 拥有的第 j 个“子用户” j={1,⋯, $w_i/w_{\min}$}，每个“子用户”以相同的 $p=\tau/W$ 的概率被选中为委员会成员，其中 W 是局部区块链网络中权重的总和。

委员会成员验证算法见算法 4，用户通过计算⟨hash,π⟩←VRF_{sk}(seed||rol)进行筛选，其中 sk 是用户的私钥。伪随机哈希决定了有多少个子用户被选中，J 个子用户中正好有 k 个被选中的概率遵循二项分布

$$B(k;J,p)=\binom{J}{k}p^k(1-p)^{J-k} \tag{4}$$

$$\sum_{k=0}^{J} B(k;J,p)=1 \tag{5}$$

$$B(k_1;n_n,p)+B(k_2;n_2,p)=B(k_1+k_2;n_1+n_2,p) \tag{6}$$

由式(6)可知，分割一个用户的权重并不影响在他/她控制下选定子用户的数量。

为了确定一个用户有多少个子用户被选中，排序算法将区间[0,1]划分为

$$\begin{aligned} I^j &= \left[\sum_{k=0}^{j} B(k;J,p), \sum_{k=0}^{j+1} B(k;J,p)\right] \\ j &= \left\{1,\cdots,\left\lfloor \frac{w_i}{w_{\min}} \right\rfloor\right\} \end{aligned} \tag{7}$$

如果 hash / 2^{hashlen}（hashlen 是指 hash 的比特长度）落在区间 I^j 内，那么该用户正好有 j 个选定的子用户。所选子用户的数量可以通过证明 π（来自 VRF 输出）公开验证，以确保各个用户拥有合法且数量正确的票数。

算法 4　委员会成员验证算法

输入　节点公钥 pk，随机种子 seed，预期成员数 τ，角色 role，用户权重 w，网络总权重 W

输出　j

if !VerifyVPF$_{\text{pk}}$(seed || role)

　　return error

end if

$p \leftarrow \tau / W$

$j \leftarrow 0$

while $\text{hash}/2^{\text{hashlen}} \notin \left[\sum_{k=0}^{j} B(k;w,p), \sum_{k=0}^{j+1} B(k;w,p)\right)$ do

　　j++

end while

3.2.2　随机种子的产生

本文将随机种子 seed 的产生分为两个阶段。在第一阶段，总成员数为 τ 的委员会的每个成员选择一个 r 位的随机字符串 R_i，诚实用户会保持随机字符串 R_i 的隐私性的同时向委员会中的其他成员发送该字符串的哈希 $H(R_i)$。委员可以以此对一组字符串哈希值集合 S 达成共识，这个集合 S 至少包含 $2\tau/3$ 个哈希值。之后把集合 S 广播给该局部网络中的所有用户并将集合 S 的值上链。

在第二阶段，委员会的每个成员向所有人广播一个包含随机字符串 R_i 本身的消息。这个阶段只有在商议集合 S 的协议完成后才开始，此时 S 已上链，无法被改变。

此时，系统中的每个用户都从最终委员会的成员那里收到了至少 $2\tau/3$，最多 τ 个 R_i，通过与上链的 $H(R_i)$比较，用户丢弃任何与承诺 $H(R_i)$不匹配的随机字符串 R_i。

每个用户对其收到的任何（$\tau/2+1$）个随机字符串 R_i 进行异或，用户可以选择不同的 R_i，然后附上用于生成种子的（$\tau/2+1$）个字符串的集合用于给其他节点进行验证。可以认为委员会发送的（$\tau/2+1$）个有效随机字符串的任何子集的异或具有有效的随机性。任何其他用户可以验证这些（$\tau/2+1$）的随机字符串是否与 S 中的承诺相匹配。这些随机字符串被用作下一次委员会选举时算法 3、算法 4 的种子，因此随机数种子仅在每次委员会轮换前生成。

3.2.3　Tendermint 算法

Tendermint 是一个安全的状态机复制算法，其借鉴于 PBFT，Tendermint 通过问责机制应对超过拜占庭假设上限的情况。Tendermint 共识过程如图 2 所示。

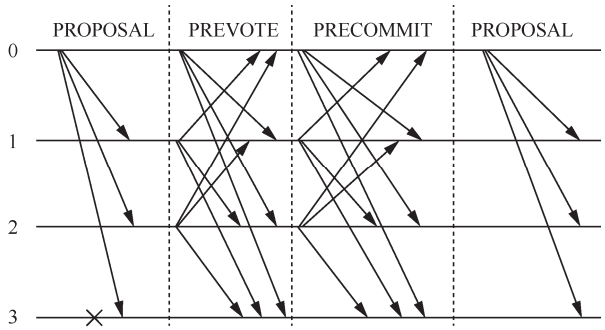


图 2　Tendermint 共识过程

总节点数量为 $3f$+1，主节点发起 PROPOSAL，其中包含当前区块链长度、共识轮数、本轮提议的区块。其他节点收到 PROPOSAL 后检查区块是否合法，如是则保存提议的区块，若自身未锁定区块则广播 PREVOTE，表示投票给该区块，其中包含提议区块的哈希以及对该哈希的数字签名，若有锁定的区块发送包含锁定区块哈希的 PREVOTE。所有节点收集保存发出 PREVOTE 消息的节点 ID 和数字签名，当收集到（$2f$+1）个对应于某个区块哈希的 PREVOTE 消息时保存（$2f$+1）个节点账号和数字签名且自身锁定该区块，并发送 PRECOMMIT，表示本轮已产生一个区块获得了多数投票。所有节点收到（$2f$+1）个 PRECOMMIT 时认为共识达成，将区块哈希对应的区块加入自身区块链，解锁锁定的区块，等待开始新一轮共识。此时主节点向云服务器提交新产生的区块以及附带收集到的数字签名，同时检查自己的任职轮数，如到达主节点更换轮数则按照委员会选取时的投票权轮换主节点。

Tendermint 算法通过各节点保存节点账号和锁机制来实现问责机制，当一个节点收到（$2f$+1）节点对一个提议区块的 PREVOTE 投票时就锁定该区块和当前轮，有锁定区块的节点只会对锁定的区块投票，只有收到（$2f$+1）个对锁定的区块的 PRECOMMIT 消息说明共识达成才会解锁区块。

这使整个区块链没有分叉，因为当锁定一个区块时说明已经有超过 2/3 的节点认可这个区块，随

着超过 2/3 节点相继锁定，在同一区块链长度上提议的其他区块最多只能收到 1/3 的投票，无法通过。由于节点保存了投票的节点账号和对应区块，可以通过检查预投票和预提交的投票检查拜占庭节点，除非拜占庭节点超过 2/3 才能完全控制共识。

3.3 全局共识执行阶段

云服务器负责对局部区块链网络产生的区块进行再验证，确保不与全局区块链存储的交易信息冲突。由于云服务器节点数量较少且信任基础较高，因此在云服务器间可以直接运行 PBFT。当局部区块链产生的交易被加入全局区块链，可以认为交易已被最终确认。

全局区块链需要保持稳定且快速的出块频率以确保局部区块链交易可以被快速确认，因此云服务器在收到局部区块且经过验证后，要尽快将局部区块打包成全局区块发送给其他云服务器，云服务器间通过 PBFT 共识保持一致。对于发送了有冲突内容的局部区块链网络，云服务器需要向这些局部区块链网络发送委员会重选消息并向局部区块链网络中所有的可通信节点请求运行 Tendermint 共识时保存的节点投票记录来搜查恶意节点。对多个时隙没有收到区块的局部区块链网络，云服务器也要检查是否有恶意节点干扰了共识达成。

4 安全性分析

本方案中区块链交易在局部区块链产生，由于局部区块链 LBNC 共识结合了 PoS 共识和 Tendermint 共识，因此可以抵御以下两种常见的攻击。

1) 双花攻击

双花攻击指攻击者使用资源 A 的一笔交易 O 被区块链确认后，在该区块之前的块上制造分叉，在分叉的区块中用 A 再创造一个交易 T，最终使得分叉链的高度高于主链，此时主链被替代，交易 O 被撤销，从而使一份资源被花费两次。在本方案中由于 Tendermint 共识不存在分叉，因此攻击者无法实施双花攻击。

2) 女巫攻击

女巫攻击通过制造多个无价值的节点参与到共识投票中，从而使得一个恶意攻击者就能影响整个网络的交易处理。本文则借用 PoS 的思想，只有节点账户中拥有权重才能有机会参与到共识的投票过程，因此能够抵御女巫攻击。

4.1 分片安全性

区块链分片的问题在于所有区块链节点被划分为数量较小的分组，虽然总体节点数量不变，但是有某些分片内恶意节点数量过多攻占分片的可能。

- 由于按用户权重选举委员会协商出块，诚实节点的权重占总分片权重数越高，攻击者控制分片成本越高。
- 假如攻击者选择攻击委员会成员控制分片，则系统可以通过减少单个委员会主持轮数，即提高委员会重选频率使攻击者无法控制足够多的当前委员会成员，一定程度上抵抗攻击。
- 云服务器在全局区块链会进行二次验证防止区块冲突，Tendermint 共识的责任制可以使恶意节点快速被云服务器检查出来，移出网络或受到惩罚。
- 在 PoS 中，存在出块权被大股东垄断的中心化风险，所提方案中即使一个节点拥有再多的权重也不能独立的控制记账权，只是有更大概率参与到处理交易的 Tendermint 过程中，即通过委员会限制单个拥有巨额货币节点的行为。
- 在分片中存在恶意节点被选择为主节点的可能，但是主节点只承担发起共识投票和向云服务器提交区块的责任，只要委员会的拜占庭节点不超过 2/3，伪造的区块将无法通过共识投票，委员会产生的区块无法通过，由于 Tendermint 问责机制的存在，主节点也无法在收到足够投票之后更换区块来欺骗其他节点。主节点向云服务器提交区块时，由于无法伪造分片内节点的数字签名，伪造的区块也无法得到云服务器认可。因此在拜占庭节点不超过总节点 2/3 的拜占庭假设下，该系统能抵御恶意主节点的共谋攻击。

委员会中成员的数目多少也会影响局部区块链网络的安全性。设 U 是某局部区块链网络中存在的权重的总数，U 足够大（无穷大）。在委员会的每一轮筛选中，最后选择的子用户的数量是不确定的（由于随机性），但被选择的子用户的预期数量 τ 是固定的。一个子用户被选中的概率 p 为 τ/U。恰好有 K 个子用户被抽中的概率为

$$\binom{U}{K}p^K(1-p)^{U-K} \tag{8}$$

式(8)等于

$$\frac{U!}{K!(U-K)!}\left(\frac{\tau}{U}\right)^K\left(1-\frac{\tau}{U}\right)^{U-K}= \frac{U\cdots(U-K+1)\tau^K}{U^K}\left(1-\frac{\tau}{U}\right)^{U-K} \tag{9}$$

当 U 为无穷，而 K 为定值时

$$\frac{U\cdots(U-K+1)}{U^K}=1 \tag{10}$$

又有

$$\left(1-\frac{\tau}{U}\right)^{U-K}=\frac{\left(1-\frac{\tau}{U}\right)^U}{\left(1-\frac{\tau}{U}\right)^K}=\frac{\mathrm{e}^{-\tau}}{1}=\mathrm{e}^{-\tau} \tag{11}$$

综上，恰好有 K 个子用户被抽中的概率为

$$\frac{\tau^K}{K!}\mathrm{e}^{-\tau} \tag{12}$$

每轮至少需要有一个提议者被选中，但提议者数量也不能太大，否则将影响算法性能。设预期的提议者数量为 τ_p，没有提议者的概率为

$$\frac{(\tau_\mathrm{p})^0}{0!}\mathrm{e}^{-\tau_\mathrm{p}}=\mathrm{e}^{-\tau_\mathrm{p}} \tag{13}$$

例如，$\tau_\mathrm{p}=26$，委员会中成员的数目小于 70 的概率为

$$\sum_{K=1}^{70}\frac{26^K}{K}\mathrm{e}^{-26}>1-10^{-11} \tag{14}$$

因此，在合理设置 τ_p 的情况下，有极大概率选出适当数目的委员会成员数，使得共识算法安全且高效地执行，从而确保 LBNC 算法的安全性。

4.2　全局安全性

全局区块链在云服务器间运行且分布式存储。攻击修改单个云服务器中的数据并没有意义，只要攻击者无法攻占超过 1/3 的云服务器，就不能突破 PBFT 的容错上限，从而无法修改数据。

所提方案由于分片按照地理位置划分，且节点可以加入退出，可能存在恶意节点到达特定地理位置加入特定分片的情况，云服务器可以通过检查分片的节点进出记录预估风险，或设置退出/加入分片的等待时间提高攻击成本。当分片节点数量失衡时，可以局部运行 K-means 算法将分片拆分或合并，保证各分片的负载均衡。

5　性能仿真与分析

实验硬件条件为一台 64 位 Windows10 操作系统、内存大小为 16 GB、处理器为 8 核 16 线程 AMD Ryzen 7 5800H 的笔记本计算机。为了模拟网络中的多个节点而采用了消息传递接口（MPI, message passing interface），在本节的性能测试中，用到了 4 个线程，当每个线程模拟 k 个节点，那么全网络将存在 $4k$ 个节点，各节点间通过 MPI 协议定义的接口函数进行通信。为了让单线程能够模拟 k 个节点，本文将通过心跳的方式，让 k 个节点一直轮流地运行在同一线程上。

5.1　分片方案验证

K-means 算法仿真如图 3 所示，可以看出，改进的 K-means 算法能较均匀地将节点进行划分，并且随节点数量增多，迭代次数增加较少，可以满足目标需求。

5.2　网络性能验证

区块链网络环境为对等网络，因此在节点的初始化过程中，各节点将随机选取 4 个节点作为自己的邻接节点，在所提双层分片网络中，消息是通过洪泛的方式传播的，即当一个节点收到一个有效且此前未接收过消息时，它将向其除消息发送方外的所有邻接节点发送此消息。本节评估了 LBNC 算法在一个分片中的性能，验证不同节点数量和不同块大小下分片内共识的确认时间和吞吐量。再将 LBNC 算法与 PBFT 算法进行对比。

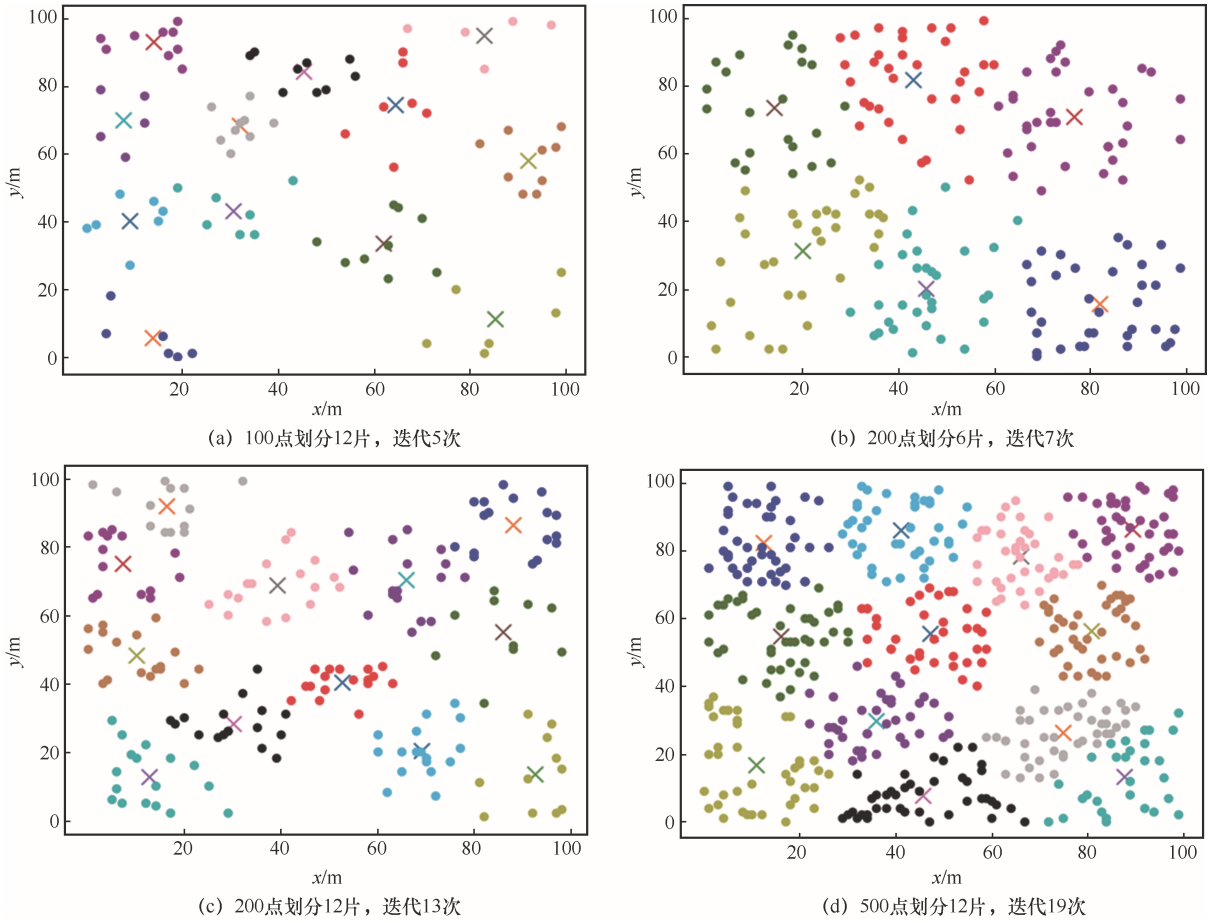
吞吐量和确认时间是描述系统性能的重要指标，吞吐量是单位时间内系统的交易处理数，确认时间是单位区块被共识确认需要的时间，表示为

$$\mathrm{TPS}=\frac{\sum_i \mathrm{transactions}_i\times\Delta_i}{\nabla_t} \tag{15}$$

$$\mathrm{Latency}=\frac{\nabla_t}{\sum_i\Delta_i} \tag{16}$$

其中，∇_t 表示系统运行时间，Δ 表示产生的区块，Transactions 表示各个区块包含的交易量，本文中交易量由区块大小除以 4 KB 的每笔交易大小[29]获得。在区块链网络中区块被共识确认才认为区块中的交易最终完成，因此区块确认时间即交易确认时延 Latency。

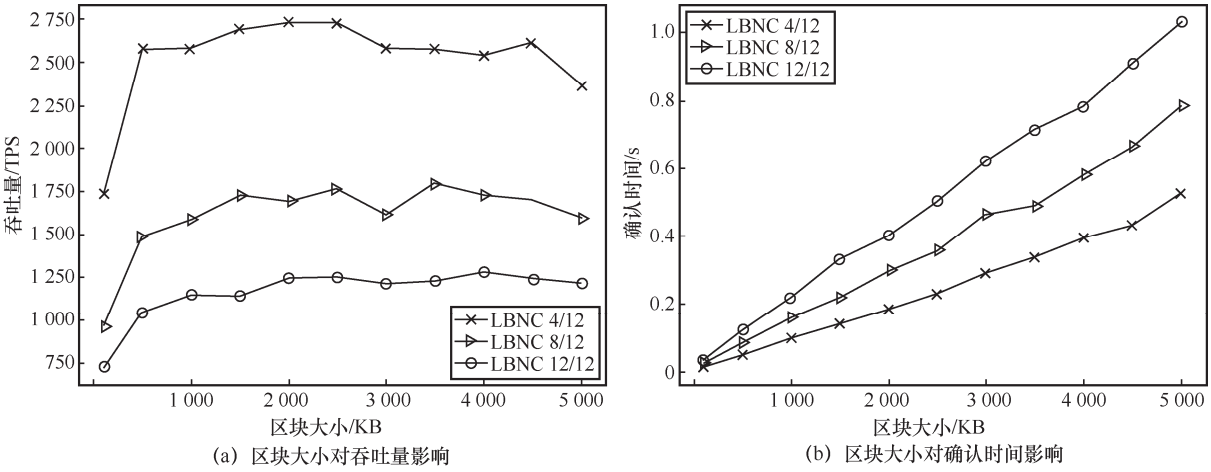
区块大小和委员会大小对吞吐量和确认时间的影响如图 4 所示，可以看出，委员会的数量对分



(a) 100点划分12片，迭代5次　(b) 200点划分6片，迭代7次

(c) 200点划分12片，迭代13次　(d) 500点划分12片，迭代19次

图 3　*K*-means 算法仿真



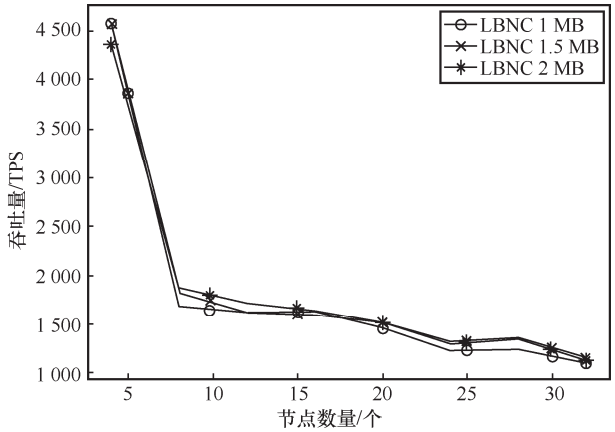
(a) 区块大小对吞吐量影响　(b) 区块大小对确认时间影响

图 4　区块大小和委员会大小对吞吐量和确认时间的影响

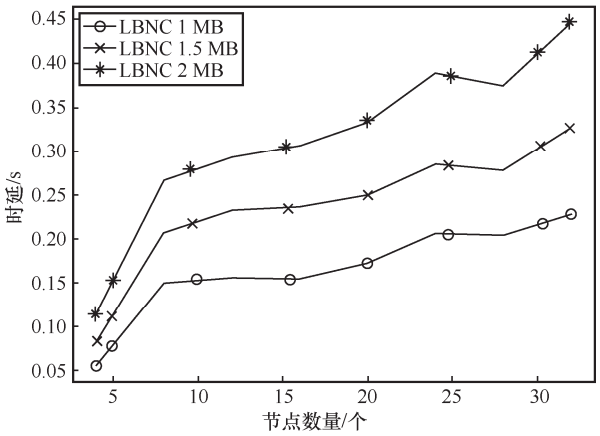
片吞吐量和时延影响明显，区块大小在0.1～1.5 MB时吞吐量明显增加，这是由于较大的区块能容纳更多的数据，每次共识可以确认更多的交易完成。但当区块大小高于 2 MB 时，吞吐量没有增加甚至开始下降，这是因为更大的区块需要更高的传输带宽，并且产生一个区块需要等待节点产生足够的交易数据，同时由于区块链网络的时延由区块产生速度决定，因此确认时间随区块大小增大而增大。因此，在实际应用区块链网络时需要根据区块交易大小和交易频率以及容忍的时延确认合适的区块大小。

委员会成员数量为 8 时，节点数量和区块大小对吞吐量和时延的影响如图 5 所示，总节点数量不

足 8 时，所有节点都是委员会成员。可以看到，区块的大小在 1～2 MB 时对吞吐量的影响不大，但是影响了确认时间。同时吞吐量随节点数量增加而降低，因为更多的节点意味着需要更多的通信消息数和同步时间。



(a) 节点数量对吞吐量影响



(b) 节点数量对时延影响

图 5　节点数量和区块大小对吞吐量和时延的影响

1.5 MB 的区块，分片内总节点数量为 24 下，委员会节点数量对吞吐量和时延的影响如图 6 所示，其中随着委员会成员的增加，吞吐量快速降低，时延增加，因此委员会成员的选取需要兼顾安全性和效率。不可否认的是委员会共识出块的方案不像全节点共识那样安全，但是性能和安全性本质上是一种权衡[30]。比特币牺牲吞吐量保证安全性，LNBC 算法根据边缘计算的场景做出牺牲，但是利用 VRF 的随机性，Tendermint 问责机制以及云服务器的再验证确保安全。

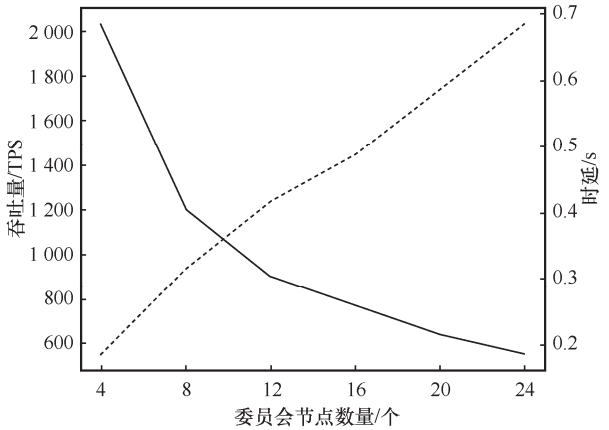
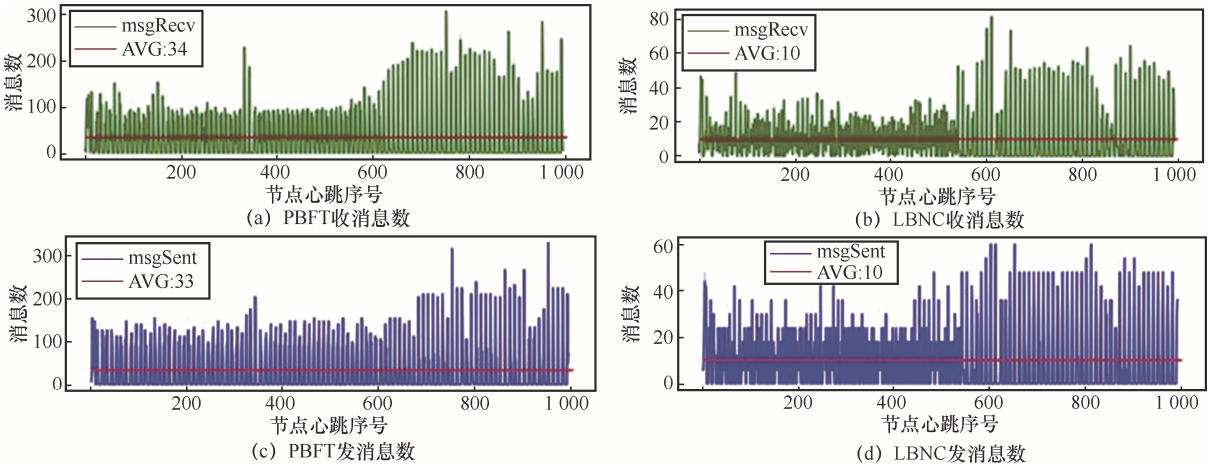


图 6　委员会节点数量对吞吐量和时延的影响

接下来将 LBNC 和 PBFT 在 24 个节点下进行对比，LBNC 中委员会节点数量为 8，8 个节点以下无委员会。PBFT 和 LBNC 收发消息数情况如图 7 所示，是 PBFT 算法和 LBNC 算法产生 102 个区块时，在同一个节点所产生的通信开销。可以看出，PBFT 算法单节点每次收到的平均消



(a) PBFT收消息数　(b) LBNC收消息数

(c) PBFT发消息数　(d) LBNC发消息数

图 7　PBFT 和 LBNC 收发消息数情况

息数为 34，每次心跳发出的平均消息数为 33，远高于相同节点在 LBNC 算法中每次心跳的平均收/发消息数(10,10)，即在处理相同数量交易的情况下，LBNC 算法中节点的通信开销远小于 PBFT 算法，符合上述分析。

不同节点数量情况下收发消息数对比如图 8 所示。可以看出，无论是 PBFT 算法还是 LBNC 算法，节点收到的消息数基本等于发出的消息数，这是由于本次仿真中没有设置恶意节点，因此节点收到的任何消息都是有效的，均需转发；在开始的时候，PBFT 算法和 LBNC 算法收发消息数基本相同，这是因为在无委员会情况下消息复杂度均为 $O(n^2)$，之后 PBFT 算法收发消息数均随节点数增加快速增长，LBNC 算法消息复杂度变为 $O(Mn)$，M 为委员会节点数量，因此消息数缓慢增加。因此，可以按需设置委员会成员数，使 LBNC 算法的通信开销小于 PBFT 算法。

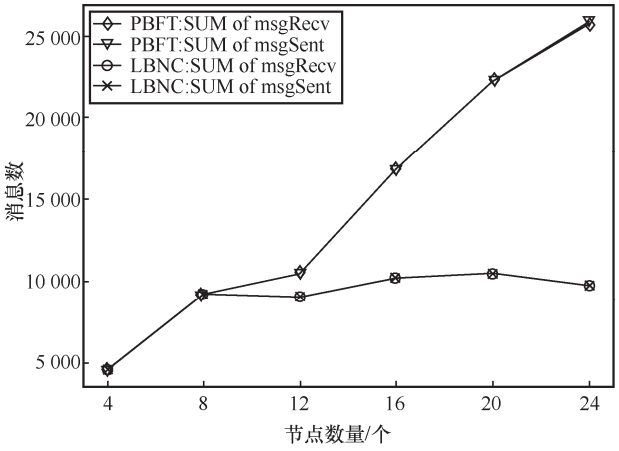


图 8 不同节点数量情况下收发消息数对比

不同节点数量情况下系统的吞吐量和时延对比如图 9 所示，实验结果表明，随着节点数量的不断增加，PBFT 算法和 LBNC 算法的吞吐量都呈下降趋势，因为 PBFT 算法下降速度极快，但 LBNC 算法复杂度为 $O(Mn)$的表现明显优于 PBFT 算法，且 LBNC 算法吞吐量曲线为平缓下降，而 PBFT 算法已接近为 0，即节点数量越大，LBNC 的优势越明显。

LBNC 算法吞吐量与分片数关系如图 10 所示，各分片分别包含 4、12、20 个节点。由于分片内部共识几乎不会受到其他分片的影响，因此吞吐量随分片数呈线性增长。

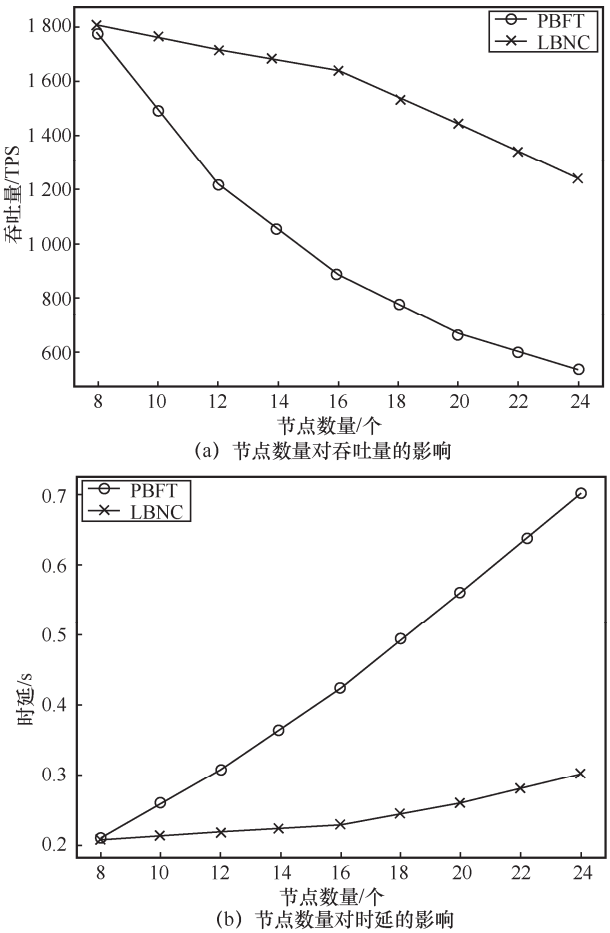


图 9 不同节点数量情况下系统的吞吐量和时延对比

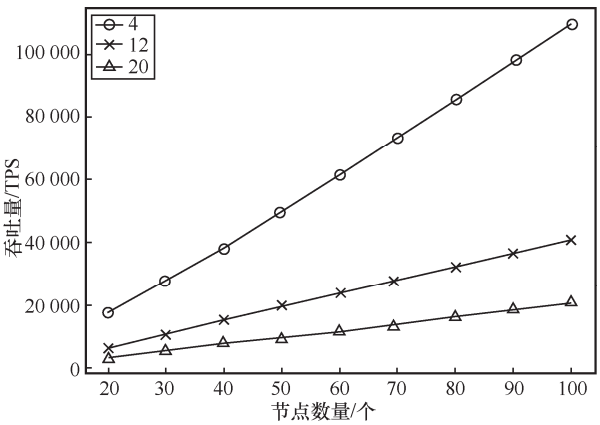


图 10 LBNC 算法吞吐量与分片数关系

6 结束语

本文提出了区块链与边缘计算的融合框架，借助区块链实现数据传输及交易的可追溯，保证数据安全。依据地理位置信息将网络分成多个局部区块链网络，在算力资源珍贵的局部区块链网络运行结合了 PoS 和 PBFT 思想的 LBNC 算法达成共识，再在云服务器间

进行二次验证保证整体的安全。通过实验验证，该方案安全性较强，且 LBNC 算法相对于 PBFT 算法具有更好的性能表现。本文的双层分片方案按照边缘计算的场景需求设计，依据地理位置的分片可以充分发挥边缘计算的边缘优势，减少边缘节点数据传输压力，使资源有限的边缘节点也能加入区块链网络，LBNC 算法在提高共识达成速度的同时可以抵御常见的针对区块链的攻击，全局区块链发挥云边协同功能，为局部区块链提供保障。

对于未来的工作，按地理位置进行划分的分片办法在强调透明性的区块链网络中可能会增加隐私暴露的风险，考虑针对应用于边缘计算的区块链网络设计有针对性的隐私保护方案或者访问控制方案，同时研究协调局部区块链数据，实现高效的跨分片交易。

参考文献：

[1] DORRI A, KANHERE S S, JURDAK R. Towards an optimized blockchain for IoT[C]//Proceedings of 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI). Piscataway: IEEE Press, 2017: 173-178.

[2] CHRISTIDIS K, DEVETSIKIOTIS M. Blockchains and smart contracts for the internet of things[J]. IEEE Access, 2016, 4: 2292-2303.

[3] TAN L, SHI N, YU K P, et al. A blockchain-empowered access control framework for smart devices in green internet of things[J]. ACM Transactions on Internet Technology, 2021, 21(3): 1-20.

[4] CHEN Z Y, TIAN P, LIAO W X, et al. Zero knowledge clustering based adversarial mitigation in heterogeneous federated learning[J]. IEEE Transactions on Network Science and Engineering, 2021, 8(2): 1070-1083.

[5] ZHOU B W, DASTJERDI A V, CALHEIROS R N, et al. A context sensitive offloading scheme for mobile cloud computing service[C]//Proceedings of 2015 IEEE 8th International Conference on Cloud Computing. Piscataway: IEEE Press, 2015: 869-876.

[6] WANG H, XIE Q, ZHAO Q, et al. A model-driven deep neural network for single image rain removal[C]//Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 3100-3109.

[7] ZHANG Z H, FENG J, PEI Q Q, et al. Integration of communication and computing in blockchain-enabled multi-access edge computing systems[J]. China Communications, 2021, 18(12): 297-314.

[8] REN Y J, LENG Y, CHENG Y P, et al. Secure data storage based on blockchain and coding in edge computing[J]. Mathematical Biosciences and Engineering: MBE, 2019, 16(4): 1874-1892.

[9] FERNÁNDEZ-CARAMÉS T M, FRAGA-LAMASP. A review on the use of blockchain for the internet of things[J]. IEEE Access, 2018(6): 32979-33001.

[10] 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: ACM Press, 2016: 17-30.

[11] WANG J, WANG H. MONOXIDE. Scale out blockchains with asynchronous consensus zones[EB]. 2019.

[12] REN Z J, CONG K L, AERTS T, et al. A scale-out blockchain for value transfer with spontaneous sharding[C]//Proceedings of 2018 Crypto Valley Conference on Blockchain Technology (CVCBT). Piscataway: IEEE Press, 2018: 1-10.

[13] GUO H, LI W X, NEJAD M, et al. Access control for electronic health records with hybrid blockchain-edge architecture[C]//Proceedings of 2019 IEEE International Conference on Blockchain (Blockchain). Piscataway: IEEE Press, 2020: 44-51.

[14] GAI K K, WU Y L, ZHU L H, et al. Differential privacy-based blockchain for industrial internet of things[J]. IEEE Transactions on Industrial Informatics, 2020, 16(6): 4156-4165.

[15] ZHOU Z Y, WANG B C, DONG M X, et al. Secure and efficient vehicle-to-grid energy trading in cyber physical systems: integration of blockchain and edge computing[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, 50(1): 43-57.

[16] LI C, LIANG S Y, ZHANG J. Blockchain-based data trading in edge-cloud computing environment[J]. Information Processing & Management, 2022, 59(1): 102786.

[17] XU X L, ZHANG X Y, GAO H H, et al. BeCome: blockchain-enabled computation offloading for IoT in mobile edge computing[J]. IEEE Transactions on Industrial Informatics, 2020, 16(6): 4187-4195.

[18] KOKORIS-KOGIASE, JOVANOVIC P, GASSER L, et al. O mniLedger: asecure, scale-out, decentralized ledger via sharding[C]// Proceedings of 2018 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE Press, 2018: 583-598.

[19] DANG H, DINH T T A, LOGHIN D, et al. Towards scaling blockchain systems via sharding[C]//Proceedings of the 2019 International Conference on Management of Data. New York: ACM Press, 2019: 123-140.

[20] SABT M, ACHEMLAL M, BOUABDALLAH A. Trusted execution environment: what it is, and what it is not[C]//Proceedings of 2015 IEEE Trustcom/BigDataSE/ISPA. Piscataway: IEEE Press, 2015: 57-64.

[21] YANG L, ZHOU W, ZHANG W. EdgeShare: a blockchain-based edge data-sharing framework for industrial internet of things[J]. Neurocom puting, 2022, 485: 219-232.

[22] ZHANG L, ZHOU Y, WANG W. Resource allocation and trust computing for blockchain-enabled edge computing system[J]. Computers&Security, 2021, 105: 102249.

[23] GAO N J, HUO R, WANG S, et al. Sharding-hashgraph: a high-performance blockchain-based framework for industrial internet of things with hashgraph mechanism[J]. IEEE Internet of Things Journal, 2022, 9(18): 17070-17079.

[24] 刘炜, 阮敏捷, 佘维, 等. 面向物联网的PBFT优化共识算法[J]. 计

算机科学, 2021, 48(11): 151-158.
LIU W, RUAN M J, SHE W, et al. PBFT optimized consensus algorithm for internet of things[J]. Computer Science, 2021, 48(11): 151-158.

[25] SINAGA K P, YANG M S. Unsupervised *K*-means clustering algorithm[J]. IEEE Access, 2020(8): 80716-80727.

[26] BUCHMAN E, KWON J, MILOSEVIC Z. The latest gossip on BFT consensus[EB]. 2018.

[27] AVCI O, ABDELJABER O, KIRANYAZ S. Wireless and real-time structural damage detection: a novel decentralized method for wireless sensor networks[J]. Journal of Sound and Vibration, 2018, 424: 158-172.

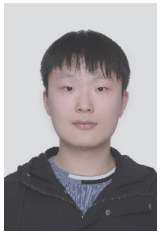
[28] MICALI S, RABIN M, VADHAN S. Verifiable random functions[C]//Proceedings of 40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039). Piscataway: IEEE Press, 2002: 120-130.

[29] ANDROULAKI E, BARGER A, BORTNIKOV V, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains[C]// Proceedings of the Thirteenth EuroSysConference. New York: ACM Press, 2018: 1-15.

[30] LUO C R, HU Y Y, ZHANG S, et al. Fission: autonomous, scalable sharding for IoT blockchain[C]//Proceedings of 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC). Piscataway: IEEE Press, 2022: 956-965.

[作者简介]

王珺（1975– ），女，博士，南京邮电大学副教授，主要研究方向为物联网、边缘计算、下一代网络等。



马建炜（1999– ），男，南京邮电大学通信与信息工程学院硕士生，主要研究方向为边缘计算和区块链。

罗金喜（1996– ），男，南京邮电大学通信与信息工程学院硕士生，主要研究方向为边缘计算和区块链。