边缘计算场景下的多层区块链网络模型研究

殷昱煜 叶炳跃 梁婷婷 段宏岳 李尤慧子 万 健

(杭州电子科技大学 计算机科学与技术学院 杭州 310018)

(浙江科技学院 信息与电子工程学院 杭州 310023)

摘 要 近年来,越来越多的研究人员尝试将边缘计算场景中收集的数据存储在区块链上,以解决传统数据存储方案中数据安全性差、防篡改性弱的问题.然而在已有的设计方案中,设备往往需要保存完整的区块数据,并且在对链上的特定数据进行取回或验证时,需要遍历大量的区块以找到对应的数据,降低了边缘计算场景中用户侧的响应速度.此外,传统共识算法也不适用于资源受限的终端设备.针对上述问题,本文提出了一种面向边缘计算的多层区块链网络模型.具体地,该模型被分为了核心层、边缘层和终端层三层:终端层被划分为多个局部网络,每一个局部网络中都包含了一个终端侧区块链,链上存储了该局部网络中终端节点产生的数据;位于边缘层的边缘侧区块链对终端层中的每一个局部网络中的终端侧区块链定期进行备份;核心层中部署了一些核心设备,负责边缘层节点的审核、注册等工作.我们对模拟数据进行了仿真实验,结果表明在该模型中查找某一特定数据哈希的速度相比传统的有向无环图式区块链模型提高了 4-7 倍.我们还提出了一种自适应工作量证明算法,算法执行的难度可以根据终端节点的行为动态进行调整.实验结果表明,相比传统工作量证明算法,执行该自适应算法的正常节点的交易效率可以提升 4-5 倍.

关键词 边缘计算;区块链;智能合约;共识算法;数据存储

中图法分类号 TP393 **DOI 号** 10.11897/SP.J.1016.2022.00115

Research on Multi-layer Blockchain Network Model in Edge Computing

YIN Yu-Yu YE Bing-Yue LIANG Ting-Ting DUAN Hong-Yue LI You-Hui-Zi WAN Jian

Department of Computer Science and Technology, Hanghou Dianzi University, Hangzhou 310018)

(*School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou* 310023)

Abstract In recent years, more and more researchers have tried to store data collected in edge computing scenarios on the blockchain to solve the problems of poor data security and weak tamper resistance of traditional data storage solutions. However, in the existing schemes, the devices need to store complete block data and traverse a large number of blocks to find the corresponding data when retrieving or verifying specific data on the chain, which reduces the response speed of the user side in the edge computing scene. In addition, traditional consensus algorithms are not suitable for terminal devices with limited resources. To address the aforementioned problems, this paper proposes a multi-layer blockchain network model for edge computing. Specifically, the model is divided into three layers: terminal layer, edge layer, and core layer. The terminal layer is divided into multiple local networks, each of which contains a terminal side blockchain storing the data generated by the terminal node in the local network. Blockchain data between different local networks are isolated from each other, which improves the security and privacy of

收稿日期:2020-09-16;在线发布日期:2021-04-29. 本课题得到国家重点研发计划项目(No. 2020YFB2103805)、国家自然科学基金项目(No. 61802093,62002088)、浙江省属高校基本科研业务费专项资金(No. GK199900299012-025, GK209907299001-020)资助.**殷昱煜**,博士,教授,主要研究领域为服务计算、边缘计算、软件工程、区块链系统. E-mail: yinyuyu@hdu.edu.cn. **叶炳跃**,硕士研究生,主要研究领域为区块链系统、边缘计算. **梁婷婷(通信作者)**,博士,副研究员,主要研究领域为数据挖掘、机器学习、边缘计算. E-mail: liangtt@hdu.edu.cn. **段宏岳**,硕士研究生,主要研究领域为区块链系统. **李尤慧子**,博士,副教授,主要研究领域为边缘计算、计算机系统. **万 健**,博士,教授,主要研究领域为网格计算、服务计算、无线传感器网络.

data in local networks. The edge side blockchain located in the edge layer regularly backs up the terminal side blockchain in each local network of the terminal layer. Many edge nodes are deployed in the edge layer, which periodically repackage the block data on the terminal side blockchain and publish it on the edge blockchain. Some core devices are deployed in the core layer, which are responsible for the audit, registration and other work of the nodes in the edge layer. We conduct experiments on simulated data, and the results show that the speed of finding a specific data hash in the model is 4-7 times faster than the traditional directed acyclic graph blockchain model. We also propose an adaptive workload proof algorithm of which the execution difficulty can be dynamically adjusted according to the behaviors of the terminal node. The algorithm is improved based on the traditional proof-of-work algorithm, which introduces the concept of node contribution value in the multi-layer blockchain network model. The node contribution value is used to quantitatively analyze the historical behavior attributes of terminal nodes. The edge node will periodically check the block content published on the terminal side blockchain, and calculate the contribution value of the terminal node in the local network according to the predefined computational formula. Then the terminal node sets the difficulty of the workload proof algorithm it executes according to the contribution value. The experimental results show that, compared with the traditional proof-of-work algorithm, the transaction efficiency of normal nodes that execute the adaptive algorithm is improved by 4-5 times.

Keywords edge computing; blockchain; smart contract; consensus algorithm; data storage

1 引 言

边缘计算指的是在网络边缘执行计算的一种新型计算方式[1]. 区别于云计算的概念,边缘计算将计算、存储资源下放,为终端用户提供了更低时延、更高智能的服务. 在云计算场景中,终端用户的请求会被汇总到云服务器中心统一进行处理,这不可避免的会出现如网络不稳定、带宽不够用等问题. 而物联网技术中大量使用了边缘计算的概念,实现了物联网设备数据的低时延、高智能化处理[2]. 然而,边缘网络中的物联网设备产生的数据往往存在安全性差、防篡改性弱等问题.

区块链技术因其不可篡改和去中心化的特性受到学术界和工业界越来越多的关注,一批结合区块链与边缘计算特点的研究工作相继被提出用以解决上述问题[3-7]. Kang 等人[7] 基于联盟区块链搭建了名为 Vehicular Blockchain 的车联网数据存储与共享模型. 在该模型中,路边探测节点(Roadside Units, RSU)作为预选节点经过审核加入联盟网络,车辆收集的数据交由 RSU 进行公开审核、存储和共享. 数据根据规模大小被存储在 RSU 本地或是中心式服务器上,对应的索引数据则由 RSU 经过计算被存储在区块链上. 但这类存储方案存在如下问题:(1)区块链占用的存储空间过大. 边缘计算场景下,大量的终端设备生成了大量的数据,即使区块链上存储的只是数据对应的索引信息,一个完整的区块链仍然要占据大量的存储空间,而且区块链的数据规模正以一个惊人的速度不断增长. (2)边缘计算场景下,在区块链中对某一个特定区块进行查找需要消耗大量时间. 边缘计算场景下,常见的区块链结构是有向无环图结构的,在该结构区块链中,对某一特定区块进行查找的时间复杂度是 $O(n)$,这对于边缘计算场景中某些需要频繁查找区块内容的服务是难以接受的. (3)传统的区块链共识算法不适合动态性强、通信拓扑变化大的边缘计算场景. 大部分边缘计算场景下的应用、服务要求即时、低时延,这也正是边缘计算优于云计算的特点. 这就要求我们的区块链模型应当具有较高的交易效率,能及时提交当前收集到的终端数据. 同时,还需要考虑到单纯提高区块链的交易效率会降低区块链的安全性.

为了解决这些问题,本文提出了一种适用于边缘计算场景的多层区块链网络模型. 该模型根据网络的拓扑结构将网络划分为不同的局部网络,每个局部网络中包含了多个终端节点以及少量的边缘节

点.边缘节点负责对局部网络中的终端节点进行管理.在每个局部网络中,终端节点共同维护了终端侧区块链,该区块链上存储了终端节点生产的数据.而边缘节点负责对这些数据进行验证,确保终端侧区块链上的交易都是合法的.边缘节点还会定期将终端侧区块链中的数据打包上传到位于边缘层的边缘侧区块链上,边缘侧区块链是多链的结构,每一条链都是对应某一个局部网络终端侧区块链的信息链,信息链上存储了对应局部网络中终端侧区块链的快照信息.位于终端层的终端侧区块链存储在局部网络中的终端节点内,该数据信息不会被其他局部网络中的终端节点获取.而边缘侧区块链是所有局部网络中的边缘节点共同维护的.相对于传统的区块链存储模型,多层区块链网络模型使得边缘节点只需要存储当前局部网络对应的信息链,或是当前局部网络关心的其他信息链,大大减少了需要存储的区块链数据.同时,本文提出的多层区块链网络模型利用了分块查找的思路,即先获取在边缘侧区块链上的区块位置,再在终端侧区块链上找到对应的实际数据,大大提高了检索某一个特定数据的速度.在该模型基础上,我们还提出了一种自适应工作量证明(Adaptive Proof of Work, A-PoW)算法.该算法基于传统工作量证明(Proof of Work, PoW)算法,可以根据系统中节点的行为衡量节点对区块链网络的贡献程度,并根据节点的贡献值映射得到 PoW 算法的执行难度.

本文使用 Java 语言实现了该网络模型,利用模拟数据进行了相应实验.实验结果表明,本模型可以有效降低边缘节点需要存储的区块链模型数据量,可以有效加快对某一特定区块数据的检索速度.根据数据量不同,检索速度的提升大致在 4－7 倍之间.同时,相比传统工作量证明算法,本文提出的 A-PoW 算法可以使得正常终端节点的交易效率提升 4－5 倍.综上所述,本文的主要贡献如下:

(1)提出了一种适用于边缘计算场景的多层区块链网络模型.模型将网络划分为多个局部网络,分别在边缘层和终端层设置了边缘侧区块链和终端侧区块链,利用局部存储的思路,使得每个边缘节点只需要存储当前局部网络中终端侧区块链的数据快照,降低了每个边缘节点需要存储的区块链数据量.

(2)提出了一种可以对区块数据进行快速检索的方法.在多层区块链网络模型中,边缘节点会定期将终端侧区块链中的交易数据以默克尔树的形式,打包成区块发布到边缘侧区块链上.边缘节点利用时间戳信息,先在边缘侧区块链上定位到相应的区块,接着通过指针在终端侧区块链上找到数据所在的交易位置.

(3)设计了一种难度可以自适应变化的工作量证明算法,该算法根据节点的行为计算节点的贡献值,贡献值用以表示当前节点在局部网络中对终端侧区块链的贡献程度.该算法会根据贡献值调整终端节点执行的算法难度,提高了正常节点的交易效率,增加了恶意节点的作恶成本.

本文第 2 章介绍本模型涉及的区块链、共识算法等预备知识.第 3 章介绍边缘计算、边缘计算结合区块链、区块链中的共识算法等领域的相关工作.第 4 章介绍了本模型的系统架构.第 5 章介绍模型中各类方法的设计.第 6 章对模型进行了仿真实验,并对结果进行了分析.最后在第 7 章对全文进行了总结.

2　预备知识

本文提出的区块链网络模型的终端侧区块链是由基于有向无环图的区块链模型进一步改进而来的.同时,基于该模型,我们提出了一种新颖的共识算法,该共识算法基于传统 PoW 算法进行了改进,提高了交易效率,降低了系统的安全风险.本章将介绍区块链、有向无环图式区块链、共识算法以及 Hashgraph 的预备知识.

2.1　区块链

区块链是一种综合运用了分布式数据存储、点对点传输、共识算法、加密算法等多种计算机技术的新型应用程序模型,其本质是一个去中心化的数据库.区块链的底层数据结构是通过加密算法生成的一系列数据块,这些数据块被称作区块.区块中记录了区块链网络中关于交易的信息,同时还包含了前一个区块的哈希值.这个特性保证了这一系列区块是不可篡改的,并且是不能在密码学上伪造的.区块链使用区块存储数据,通过共识算法来将新的区块发布在区块链上,并使用加密算法对发布的区块进行签名,以实现传输、访问环节的身份识别[8].此外,节点通过使用由脚本语言编写的智能合约,可以对链上数据进行较复杂的处理.当前流行的区块链按照结构可以被分为两类,一类是基于链的区块链,另一类是基于有向无环图的区块链[9].链式区块链中,区块按照区块上记录的时间戳顺序写入到区块链网络中.由于链式区块链不支持并发的交易执行,因此

链式区块链的交易效率通常较低.当区块链上的交易数量增加时,区块链的性能就会线性下降[10].

2.2 有向无环图式区块链

本文提出的区块链网络模型中,终端侧区块链是基于有向无环图式区块链进行设计的.有向无环图式区块链是在链式区块链基础上发展而来的一种新颖的区块链结构.链式区块链存在扩展性不足的问题.以比特币为例,比特币大约每 10 分钟产生一个区块,区块的大小被设定为 1 MB,仅仅能够包含 3000－4000 笔交易[11],即平均每秒只能处理 5－7 笔交易.随着比特币上需要处理的交易也越来越多,比特币网络的拥堵情况也越来越严重,交易费用也开始增加[12].针对这个问题,越来越多的人开始讨论区块链的扩展性问题[13-14].基于有向无环图的区块链带来了解决扩展性问题的新思路.有向无环图式区块链是由交易构成的,其中每个节点都可以是交易的发起方或交易的验证方.在节点向区块链网络广播交易前,它需要验证另外两个随机交易,并将当前发布的交易链接到这两个随机交易上,这种处理交易的方式被称为异步处理模式.Tangle 网络是有向无环图式区块链的一个实现.本文提出的终端侧区块链就是基于 Tangle 网络进行构建的,每一个新创建的交易都需要随机验证区块链中任意两个尚未得到验证的交易.

2.3 共识算法

共识算法是用于解决分布式系统中数据一致性问题的算法.常见的共识算法包括传统分布式系统领域中的 Paxos、Raft 以及加密货币领域中使用的工作量证明、权益证明、委托权益证明等.当前最大的区块链网络——比特币就使用了基于工作量证明的一致性共识算法.该算法要求网络中的节点去解决一个难度较高但答案易于验证的问题.第一个解决了该问题的节点就获得了当前区块的记账权.但是,基于工作量证明的共识算法存在交易效率低、消耗算力高等问题.如果只是单纯降低工作量证明算法的难度,会加快区块链的确认速度与交易效率,但是区块链会出现频繁的分叉现象.

权益证明一致性(Proof of Stake,PoS)算法采用了一种截然不同的思想.这种算法按照矿工持有的币或者股份给矿工分配打包权和投票权,币越多,投票权也就越高.在 PoS 算法中,打包权和投票权是分开的,即先根据所有参与者的持币量,分配打包权和投票权.接着从所有参与者中选出一个候选打包人,候选打包人打包出一个候选区块,并将该候选区块广播给其他参与者进行投票,如果得到多数票的支持,那么这个候选区块就会成为区块链上的正式区块.为了提高区块链的效率,PoS 算法也可以设计为从所有参与者中随机选出参与者,组成一个委员会,由这个委员会负责投票工作.

2.4 Hashgraph

本文提出的模型是一个多层区块链结构.其中,位于边缘层的边缘侧区块链采用的技术方案与 Hashgraph[15] 类似.Hashgraph 是 Swirds 公司提出的一种类似区块链的技术方案,它不是一个完整的区块链方案,也不是一种数字货币解决方案,它是一种数据结构和共识算法.Hashgraph 的特点是交易速度快,支持异步拜占庭容错,使用虚拟投票和 gossip 算法来解决区块链的分布式一致性问题.Hashgraph 还通过复杂的数学分析证明了其交易的不可篡改性.Hashgraph 是一种基于平行链的共识协议.Hashgraph 中的节点参与同步全网的状态以及数据.每个节点只能在自己的链上发布区块(也叫事件,event),在这样的一个区块中,包含了同步信息和交易信息.每个区块不仅仅包含了交易数据,还包含了在网络中传递的消息.每个节点发布新的区块之后都会使用 gossip 协议进行全网广播,当一个节点 N_A 收到别的节点 N_B 发来的区块之后,会先对区块的签名以及内容合法性进行校验,然后校验该区块是否比之前收到的区块的时间戳更晚.校验通过后,节点会创建一个新的区块,该区块的两个哈希指针分别指向自己链的上一个区块,以及节点 N_B 链上的最新区块.同时节点 N_A 可以在创建的区块中添加自己想要发布的交易数据.最后 N_A 将该区块进行全网广播.

通过这样的机制,Hashgraph 可以记录各个节点收到区块的顺序,最终能达成区块的全网顺序一致性.Hashgraph 在满足了异步拜占庭性质的同时,还通过虚拟投票机制减少了通信的开销.

3 相关工作

随着边缘计算技术场景的不断拓展,边缘计算场景下,网络的扩展性、功能整合、资源管理等问题受到了研究人员的广泛关注[16-21].Maglaras 等人[16]将传统的社交网络关系带入到使用了边缘计算技术的车联网场景中,提出了社交物联网 SIoT 的概念,该网络关注车辆与乘员之间的社交互动,并对该模

型可能的应用方向以及安全性进行了分析. Sun 等人[20]为了解决传统物联网架构中的扩展性问题,提出了一种采用移动边缘计算思想的物联网架构 edgeIoT. 其中,终端设备连接到雾节点,雾节点在本地提供计算资源. 基于 SDN 的蜂窝核心调度雾节点之间的数据包转发. 该架构可用于在移动边缘处理数据流. Zhou 等人[21]提出了一种双层车联网模型,通过维纳过程(Wiener Process)和贝叶斯非参数学习(Bayesian Nonparametric Learning)对车辆之间的共享内容进行建模,并提出了一种基于价格上涨的迭代匹配算法(Price-rising-based Iterative Matching Algorithm),实现了车辆到车辆的快速的内容共享.

但这些方案存在数据的共享安全性差、防篡改性弱、数据时效性差等边缘计算常见的问题. 随着区块链技术的发展,越来越多的研究人员开始利用区块链技术去解决边缘计算中数据安全存储的问题[4-5,7,22-24]. Ren 等人[25]提出了一种适用于边缘计算的混合存储架构和模型,该模型充分利用边缘网络设备和云存储服务器的数据存储优势,在云服务层构建全球区块链,在物联网终端上构建本地区块链,增强了终端数据存储的可靠性. Sharma 等人[26]提出了一种采用边缘计算技术的分布式区块链云架构,该架构是一个多层模型,能够实现低延迟的数据访问和低成本的高性能计算. Wang 等人[24]提出了一种基于区块链的分布式车联网认证机制,该认证机制引入受信任的云服务提供商完成对车辆的认证工作. Zhang 等人[27]提出了一种基于区块链的数据共享模型,该模型在实现安全性和隐私性的基础上,激励了车辆参与信息的共享.

除此之外,研究人员还对适用于不同场景的各类共识算法进行了研究[28-32]. Su 等人[28]提出了一种基于节点信誉值的授权拜占庭容错算法,并将其作为区块链网络的共识算法,该算法可以在区块链网络中的各个授权分区之间达成共识. Yu 等人[30]基于比特币中使用的工作量证明算法提出了一种可以在较长时间内抵挡 51%攻击的共识算法,根据该算法,矿工的实际算力是综合其历史工作量及当前持有的计算资源一起得出的. 该系统即使在攻击者控制了超过全网 50%算力的情况下也能够在一段时间内保证系统的安全性,同时还提供了每秒 10000 次交易(TPS)的高吞吐量. 但上述对共识算法的改进仍然不能同时满足物联网场景下高并发、低延时、强安全性的要求.

本文参考上述研究成果,提出了一种可以降低区块链数据存储量,并且适用于边缘计算场景的多层区块链网络模型,并在此基础上提出了自适应工作量算法,提高了局部网络中终端节点的交易效率,降低了系统的安全风险.

4　多层区块链网络模型架构

4.1　概述

本文提出的多层区块链网络模型主要由终端层、边缘层和核心层组成,如图 1 所示,层与层之间通过数据信息传输进行相互协作,以实现模型功能.

在终端层中,各类终端设备从环境中收集数据,通过与边缘设备通信来访问服务. 因为终端设备的计算力往往较低,因此终端设备上执行难度可以动态变化的 A-PoW 算法. 终端节点收集数据并将数据打包成交易上传至终端侧区块链. 边缘层中,算力较强的边缘设备作为局部网络的中心节点,负责处理来自局部网络中终端设备的数据请求,并对终端侧区块链中交易的合法性进行检查. 同样位于边缘层的边缘侧区块链是一个多链的区块链结构. 边缘设备会定期将终端侧区块链中的交易数据打包成区块,发布到边缘侧区块链中,这些区块被称为终端侧区块链的快照信息. 核心服务设备是核心层中的重要组件,负责定期对边缘侧区块链进行备份,并负责边缘设备的注册、授权等工作.

我们在终端层中设置了终端侧区块链,这降低了边缘层中边缘设备的计算压力,充分利用了终端层中终端节点的计算资源. 我们将部分只会发生在单一局部网络中的任务(比如在局部网络中的数据索引证明等)卸载到终端层,减轻了边缘层的压力. 这部分任务的执行不会在边缘层体现,也不需要边缘层的参与. 除此之外,我们还可以将部分具有数据隐私性的任务卸载到终端层,起到一个保护数据隐私性的作用. 因为边缘层中的边缘侧区块链上保存的只是终端层的一个数据快照信息,只对终端层数据起到数据验证和快照备份的作用. 某个局部网络中的节点无法直接读取其他局部网络中的完整数据内容,可以起到隔离局部网络数据的作用. 同时,多层区块链结构还可以减少边缘层中边缘节点需要存储的数据量. 在边缘计算场景中,数据具有地域性、时效性等特点. 我们按照通信拓扑结构划分了多个局部网络,确保大部分的数据交换发生在同一局部网络中,处于不同局部网络的节点之间在大多数情

况下没有兴趣也没有必要去进行频繁的数据交互.边缘节点中只需要存储其他局部网络的 快照信息即可.设置双层区块链结构减少了边缘节点需要存储的数据量.

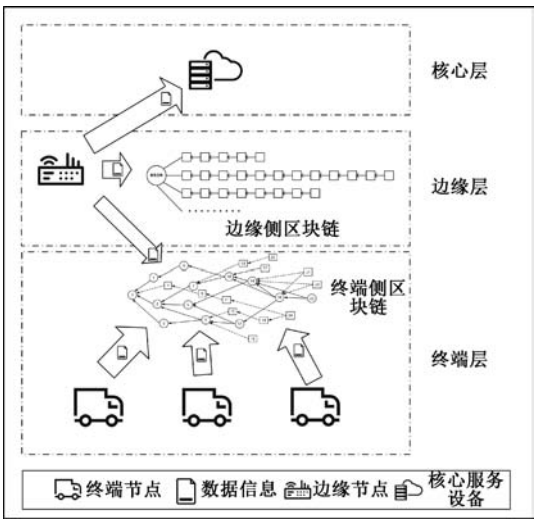

图 1 多层区块链网络模型架构图

4.2 终端侧区块链

终端节点的传感器从环境中收集数据,终端节点使用哈希函数对收集到的数据计算得到对应的字长固定的索引信息.收集到的数据被存入数据服务器中,而索引信息在被打包成交易后被上传至终端侧区块链网络.

为了提高终端侧区块链的交易效率,与比特币不同,我们的终端侧区块链使用了有向无环图(Directed Acyclic Graph, DAG)结构的区块链,该区块链对外提供分布式数据存储以及高效安全的数据查询服务.DAG 结构区块链是一种结构特殊的区块链,与链式区块链不同,DAG 结构区块链不是按出块时间组织的单向数据链,而是由交易单元构成的网络,支持异步并发地写入交易.

4.2.1 局部网络

终端侧区块链由局部网络中的多个终端节点共同持有、维护.按照通信拓扑上的结构,网络被分为多个局部网络,每一个局部网络都有相应的边缘节点对其进行维护.边缘节点是那些算力更强的设备,如中心路由、路边边缘单元等.边缘节点除了负责对局部网络中的终端节点进行管理,如终端节点的注册、局部网络的信息配置等工作之外,还需要定期对终端侧区块链中的交易进行检查,以防止恶意终端节点对区块链网络的破坏.边缘节点持有局部网络中所有终端节点的识别码.识别码是由终端节点本地生成的一串用于身份识别的字符串.终端节点生成识别码的过程如下:终端节点本地随机生成一个 32 字节的二进制数,作为其私钥(SEC-KEY),接着利用 ECDSA 椭圆曲线算法对其私钥进行计算得到对应公钥(PUB-KEY).最后,终端节点对 PUB-KEY 依次使用 SHA－256 和 RIPEMD－160 算法进行计算,计算结果就作为该终端节点的识别码对外公布.

4.2.2 终端侧区块链的结构

终端侧区块链采用了 DAG 结构,而不使用传统区块链解决方案中的链式结构.这是因为链式结构适用于那些强调强一致性的区块链金融体系.而在边缘计算场景下,终端节点之间发布交易是高并发的.如果使用传统的链式区块链,多个节点争夺区块链的记账权,会产生大量的无效区块,这对于系统效率的影响是显而易见的.对比之下,有向无环图式区块链对微交易、高并发以及异步操作的支持,能够很好地适用于我们的边缘计算场景.

区块链中使用交易的数据结构代替了链式区块链中区块的数据结构.而且,每一笔上链的交易都需要指向区块链中尚未完全得到确认的交易,即新发布的交易在上链前,需要对终端侧区块链中至少两个交易进行验证.与比特币中区块的 6 次确认类似,当终端侧区块链中的一笔交易被除发布者以外的终端节点验证超过 6 次后,我们就称这个交易是“已确认”的.终端侧区块链鼓励终端节点选取“待确认”的交易进行验证,并且终端节点相应的行为会被其所属的局部网络感知,并由对应的边缘节点进行检查.终端侧区块链中交易的结构包括了交易头和交易内容两部分.交易头中包含了由当前交易计算得到的哈希值、当前交易发布时验证的交易的哈希值、发布当前交易的终端节点的识别码.交易内容中包含了终端节点计算得到的数据索引等信息.

我们通过自适应工作量证明算法以及检查点交易的方法控制了有向无环图式区块链中常见的分叉现象.在有向无环图式区块链中,影响区块链分叉拓扑结构的主要有两个要素,一个是当前网络中的并发程度,一个是网络的延迟时间.网络的并发程度越高、网络的延迟水平越高,分叉现象也就越明显.我们通过后文提出的自适应工作量证明算法,对网络的并发程度进行了控制.同时,我们还仿照 IOTA 提出了检查点交易的概念.位于边缘层的边缘节点每隔一定的时间间隔就会发布一笔可信的检查点交易,表示上一段时间间隔内的所有交易都已经被确

认,同时对所有终端节点的工作量证明算法难度进行调整.这样一笔检查点交易会验证当前所有的待确认交易.在检查点交易发布之后,区块链中就只有检查点交易一笔待确认交易,新发布的交易会验证并链接检查点交易.这样的机制会对终端层的有向无环图式区块链的拓扑结构产生一定的收束作用.同时通过调节终端节点工作量证明算法的难度,来对网络中的并发度进行控制,也能够起到抑制大量分叉现象的作用.网络延迟则不是论文讨论的重点.在边缘计算的真实应用场景中,同一个局部网络中的终端节点通常处于相同或者相近的网络环境下,实现全局的数据一致性并不需要太长的时间.

4.2.3　边缘节点的检查过程

类似于 IOTA 中协调节点对 Tangle 网络中交易的检查,边缘节点也会定期对终端侧区块链上的交易进行检查,检查内容包括了交易本身的合法性、前置交易的属性等.边缘节点会根据检查结果,计算得到对应的终端节点的贡献值.终端节点的贡献值用来表示该终端节点对终端侧区块链的贡献程度.除了终端节点发布的交易内容会影响该节点的贡献值外,影响因素还包括终端节点在交易上链过程中的其他行为,具体内容会在 5.3 节进行详细描述.边缘节点将局部网络中所有终端节点的检查结果打包进一个新的交易,发布到终端侧区块链中.该交易也被称作"检查交易".后续终端节点发布的交易必须直接或间接地对该"检查交易"进行验证和确认.终端节点需要根据"检查交易"中的内容设定自身的贡献值,并调整 A-PoW 算法的难度.

4.3　边缘侧区块链

边缘侧区块链位于边缘层,其中存储了每个局部网络中终端侧区块链中交易的数据快照.

4.3.1　边缘侧区块链结构

参考 Hashgraph,边缘侧区块链采用是多条单向链相互交织的结构.关于 Hashgraph 的原理与结构我们在第二章进行了详细的描述.边缘侧区块链以创世区块为起点,引出多条单向链,每一条单向链中都存储了一个局部网络中终端侧区块链中存储的交易信息.单向链上的数据存储结构与比特币相同,都是区块.每一个区块同样被分为了区块头和区块体两部分.区块头中存储了指向前置区块的哈希指针以及创建区块的时间戳信息.区块体中包含了交易信息,交易以默克尔树的形式存储.其中,树的叶子节点存储的是一个时间段内终端侧区块链中新发布的交易的哈希值.这棵默克尔树可被用于在当前网络模型中对交易数据进行快速检索,关于快速检索的机制会在 5.3 节中进行描述.除此之外,区块体中还包含了数据负载信息.数据负载信息中包含了终端节点对检查结果的的签名.与比特币中全节点和轻节点的作用类似,为了降低边缘节点需要存储的区块链数据,终端侧区块链上的链被分为了"全链"与"轻链"两种类型.全链上的区块存储结构是完整的,轻链上的区块中只保存了区块头的内容.

边缘侧区块链上每一条链的第一个节点被称作"起始区块",该区块记录了当前链对应的局部网络信息.

4.3.2　智能合约

智能合约是一种用代码进行有效力约束的自动执行合约.即使在没有中央参与者的情况下也可以有效地完成对合约的正确执行.

本文提出的多层区块链网络模型充分利用了区块链以及智能合约的特点.终端节点通过执行位于终端侧区块链的数据存储智能合约实现原始数据以及对应数据索引信息的安全存储.原始数据即由终端设备的传感器从外界收集的第一手信息,例如温度、湿度、当前移动设备的坐标等.数据索引信息指的是由终端设备对原始数据进行哈希计算后得到的字长固定的哈希值.数据量较大的原始数据被存储在私有的存储服务器中,而占用空间较小并且保密性较好的数据索引信息被存储在公开的边缘侧区块链上.在本文提出的模型中,智能合约的执行被分为了四个阶段,分别是部署阶段、存储阶段、调用阶段和执行阶段.

在部署阶段,边缘层首先将要部署的智能合约发布到核心层,核心层完成备份之后再由边缘层将智能合约部署到终端层局部网络中.智能合约由边缘节点通过交易的形式完成部署.当边缘节点需要在局部网络中发布智能合约时,需要调用检查点交易智能合约,对当前的终端侧区块链进行检查.然后将待发布的智能合约编译后的字节码文件放入发布的检查点交易的数据负载中,并且用标志位显式地标记此处有新的智能合约发布.由于每个终端层的节点都需要访问并验证最新的检查点交易,此时终端节点就会在本地更新智能合约池.每个终端层节点都完成了智能合约池的更新之后就代表当前新的智能合约部署完成了.

在存储阶段,智能合约是在每个终端节点以及边缘节点的本地进行存储.同时,发布智能合约的交易也会在区块链中公开透明地保存(即检查点交

易).当节点调用智能合约时只需要找到在本地存储的合约字节码文件运行即可.

最后是智能合约的调用与执行阶段.在以太坊中,智能合约的调用是通过一笔交易的形式显式地存储在区块链中.交易中包含了想要调用的合约的索引以及与合约相关的输入数据.接着,矿工在验证交易的时候,就会去执行相应的智能合约,完成对以太坊状态机状态的更改.但是在我们的终端侧区块链中并没有账户的概念,智能合约的执行者也不再是矿工,而是合约的调用者.所以我们将合约的调用与执行进行了合并.智能合约的调用与执行过程都记录在一笔交易中,并发布到区块链上.这笔交易中总共包含了:

(1)合约调用者调用的合约的索引以及输入的参数;

(2)合约的调用者在本地执行一次相应的智能合约,所产生的输出;

(3)合约调用者的哈希与签名.

智能合约是一段可运行程序,区块链为其提供了安全可靠、公开透明的执行环境.每一笔交易的发布都需要选择两个交易进行验证,验证交易的过程是系统安全可靠的表现.节点通过复现一遍合约的运行过程,并检查产生的结果是否一致来检验智能合约的运行流程是否安全可靠.同时系统中的边缘节点也会通过检查点机制来检查合约的调用是否正确.

4.4 自适应 PoW 算法

终端节点需要执行共识算法将打包好的交易发布到终端侧区块链,以保证局部网络中的所有终端节点都对该交易达成共识.常用的共识算法有工作量证明算法、权益证明算法等.位于终端层的终端节点往往是那些算力较弱的终端设备,因此,在终端侧区块链上需要执行的共识算法对算力的要求就应该尽可能地低.但是,过低的算力要求又会降低终端侧区块链的安全性.在本文中,终端侧区块链采用的共识算法是难度可动态调整的工作量证明算法.工作量证明算法的难度会根据终端节点的贡献值动态进行调整.关于终端节点的贡献值计算公式会在5.4节中给出.

5 数据的安全存储与快速检索

基于第4章中描述的多层区块链网络模型,我们在各层上提供了多种与数据存储、数据检索相关的服务,供网络中的各类节点进行调用,其中对外提供的基础服务是数据的安全存储以及检索.为了实现这两个功能,需要我们借助区块链中的智能合约来完成.我们还会介绍边缘节点对终端侧区块链的检查机制以及自适应工作量证明算法的详细计算公式.

5.1 数据存储智能合约

数据存储智能合约向终端层的终端设备提供了安全的数据存储服务.数据存储智能合约被部署在终端侧区块链上,终端设备在当前局部网络中设置的边缘节点上注册后,就可以调用该智能合约完成数据的安全存储.终端设备收集的原始数据以及对应的索引数据信息分别被存储在数据服务器和终端侧区块链上.在终端节点这一侧还可以对原始数据预先进行数据处理、数据切分等操作,以实现特殊的应用.存储在终端侧区块链的索引数据信息中包含了原始数据的存储地址,用于原始数据的取回.除此之外,索引信息中还包含了时间戳、终端节点的识别码等信息.

数据存储智能合约的执行分为三个阶段,第一个阶段是数据索引信息的准备阶段,终端节点的传感器收集数据后,通过 MD5(Message-Digest Algorithm 5)信息摘要算法计算得到对应的 MD5 值.终端节点将 MD5 值以及其他必要信息(原始数据的哈希值、存储地址等)进行打包,生成数据索引信息 RawJson.RawJson 中包含了原始数据的 MD5 值、原始数据的版本号、多个原始数据存储源的存储地址、存储时间戳信息.RawJson 是以 JSON 格式序列化存储的摘要信息.多个原始数据存储源的存储地址表示原始数据可以生成多个数据副本,保存在多个可信的存储节点中,防止某个存储节点因为故障导致原始数据丢失.接着,终端节点对 RawJson 进行签名,打包成交易,并将交易作为输入参数调用终端侧区块链上的数据存储智能合约.

第二阶段是数据存储智能合约的执行阶段.在该阶段,数据存储智能合约会对输入的交易内容进行检查,确认交易的签名节点在当前局部网络中的边缘节点中进行过注册.接着,智能合约会返回一个签名消息给终端节点,消息中包含执行合约的交易地址.当前的合约执行过程会进入等待状态,等待数据服务器的唤醒消息.

第三阶段是原始数据的存储.终端节点收集到的数据会被存入数据服务器,数据服务器是终端节点自己选定的数据存储源,确保了隐私数据本身只有终端节点和终端节点信任的数据服务器所有.终端节点的数据存储成功后,数据服务器会发送一个

包含自己签名的存储凭证到第二阶段执行的智能合约处,唤醒智能合约.数据存储智能合约会继续执行,确认当前收到的存储凭证是否与之前接收的交易中的存储源地址一致.如果一致,表示原始数据存储成功.智能合约会通知终端节点执行自适应工作量证明算法,在局部网络中达成共识,将该交易发布到终端侧区块链中.

5.2　边缘节点检查机制

用于调节终端节点挖矿难度的贡献值并不由终端节点自己给出,而是由边缘节点对某一时间段内的交易进行检查后,显式地在终端侧区块链中给出.边缘节点的检查过程参考了 DAG 结构区块链项目 IOTA 中的协调器机制.

在 IOTA 的 Tangle 网络中,每一个节点都是一笔交易.每产生一笔交易,就会有一笔交易上链,由于没有传统区块链应用中的出块时间限制,IOTA 的数据吞吐量可以达到理论上的无上限.但是在网络初期,参与者比较少的情况下,整个网络不够健壮,安全性比较差.所以在 IOTA 的目前实现中,依靠一个中心化的协调器来确保系统的安全性,可以有效地防止双花攻击和寄生链攻击.协调器每两分钟发布一笔里程碑交易,验证当前所有的待验证交易.由于 DAG 的拓扑结构,每次里程碑交易都会直接或间接验证当前网络中的全部交易.于是只有得到了里程碑交易确认的交易才算是置信度为 100% 的交易,整个网络的共识也建立在协调器上.

边缘节点采取了相似的检查策略,即间隔一段时间 T ,发布一笔“检查交易”来协调局部网络内的边缘节点.边缘节点检查 T 时间内所有边缘节点发布的交易,并计算所有边缘节点的贡献值.以贡献值为指标评估局部网络内终端节点的行为,再通过发布“检查交易”的形式,将终端节点的贡献值公布在终端侧区块链中,供终端节点挖矿或者检验交易时查询,下面将介绍边缘节点对交易进行检查的工作细节.

终端节点是交易数据的生产者,在本地完成原始数据的索引信息计算以及原始数据上传.接着,通过智能合约的执行,终端节点将存储凭证以及索引信息打包成一笔交易,运行 PoW 算法,将这笔交易发布到终端侧区块链中.实际上整个终端侧区块链中保存的只是数据存储的索引,数据与数据之间的存储关系较独立,不存在因果关系.所以不必像比特币、以太坊那样,对每一笔交易都要进行严格的输入输出脚本检验.即区别于区块链金融应用,本系统对于传统的双花攻击、寄生链攻击等攻击手段都不用特别考虑.边缘节点主要考虑的终端节点恶意行为主要是终端节点的懒惰行为以及终端节点的大量发布无意义交易的行为.后者会增加网络的负载,最终造成系统的瘫痪.边缘节点对这些恶意行为进行检测,根据贡献值计算公式,计算出节点的贡献值,以“检查交易”的形式发布到终端侧区块链中.终端节点在执行 PoW 算法时需要向“检查交易”查询自己的贡献值,以得到本次执行 PoW 算法的难度值.这里难度值根据节点的贡献值动态变化的 PoW 算法被称为自适应 PoW 算法,相关的实现细节会在 5.4 节中描述.上文介绍了边缘节点检查机制的执行过程,其中有关区块链安全性分析的技术细节会在 5.5 节中进行描述.

边缘节点完成对终端侧区块链中某一时间段的交易的检查后,发布的“检查交易”会链接局部网络中所有的“待确认交易”,后续局部网络中新发布的交易都必须链接该“检查交易”.“检查交易”代表了终端侧区块链中定期进行的一次交易收束,可以降低终端侧区块链中的结构复杂性.

5.3　数据快速检索机制

边缘节点除了每间隔一段时间 T 在终端侧区块链上发布“检查交易”之外,边缘节点还会从本次检查中遍历到的交易中提取交易的信息打包成完整的区块发布到边缘侧区块链上.边缘侧区块链是由多条单链组成的存储结构,其中每条链都代表了一个局部网络中终端侧区块链的摘要信息,链上的每个区块代表该局部网络中某一个时间段内发布的交易的数据快照.

不同的局部网络之间借助边缘侧区块链实现跨区域的数据共享、数据检索功能.下面介绍边缘侧区块链的结构.

边缘侧区块链中将终端侧区块链的交易集合组织成一棵默克尔树,每个交易对应默克尔树的一个叶子节点.默克尔树的叶子节点中包含了对应交易的数据标签信息以及指向终端侧区块链中对应交易的哈希指针.注意,这个哈希指针只在对应的局部网络中有意义.默克尔树将交易从左到右分为两两一组,每一组中计算得出一个哈希值,作为该组的父节点信息.层层往上,最终得到的一棵强哈希关联的二叉树.区块头中会保存默克尔树的根节点哈希值.由于哈希算法的不可逆性,保证了默克尔树中的所有节点都是不可篡改的.默克尔树的模型结构如图 2 所示.

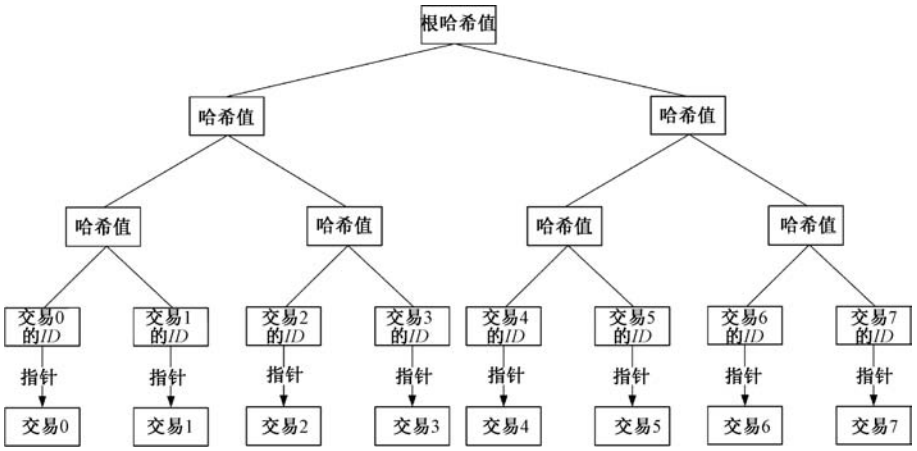

图 2 默克尔树的模型结构

在边缘计算场景中,数据具有地域性、时效性,因此我们将边缘计算网络根据通信拓扑结构划分为了多个局部网络,保证了大多数的数据交互请求都发生在同一个局部网络中,不同局部网络之间较少进行数据交互.由于局部网络之间存在一定的隔离性,所以单个边缘节点没有必要去记录边缘计算网络中所有局部网络的信息,它只需要去保存与之直接相连的局部网络的信息.参考比特币中全节点和轻节点存储的区块内容的区别,在多链结构的边缘侧区块链中,根据单链上区块的完整性区别,单链被分为了"全链"和"轻链"."全链"上的区块中包含了区块头和区块体结构,而"轻链"上的区块中只包含了区块头结构.某个边缘节点所处的局部网络以及与该边缘节点直接相连的局部网络在该边缘节点本地存储的边缘侧区块链中都以"全链"的形式进行存储,其他局部网络对应的单链以"轻链"的形式进行存储.不同层次的存储结构既降低了边缘节点的需要存储的数据内容,同时又保证了不同链之间可以对区块的内容进行验证,防止区块链数据被篡改.

多层区块链网络模型上的数据检索分为模糊查询和精确查询.模糊查询即事先不明确具体要找的交易实体,而是通过区块中包含的标签信息去判断是否需要该数据,如果需要该数据,就发起数据请求.精确查找指节点事先知道目标交易的哈希值,是通过该哈希值找到具体的交易实体.模糊查询和精确查询又被分为了跨局部网络的数据检索和局部网络内的数据检索.其中,如果是模糊查询中的局部网络内的数据检索,边缘节点首先遍历本地存储的局部网络对应的"全链".根据检索内容的一个大致时间范围,从后往前开始遍历(因为边缘侧区块链的也是向前验证),检查区块头中的时间戳信息,确认在时间范围内最后一个区块的位置.接着,从该区块开始,向前逐个遍历区块体中默克尔树的子节点,检索自己感兴趣的数据.检索到数据后,直接按照哈希指针找到对应的那个交易即可.

如果是模糊查询中跨局部网络的数据检索,此时情况会稍微复杂一些.假设边缘节点 N_R 需要在其他非直连的边缘节点 N_A 上检索某一份感兴趣的数据.此时,边缘节点 N_R 本地存储的边缘区块链中关于边缘节点 N_A 的链是一条"轻链",即链上的区块中只包含区块头信息.第一步,边缘节点 N_R 仍然需要根据一个大致的时间范围定位从后向前定位到靠近时间范围边缘的最后一个区块.确定区块范围后,边缘节点 N_R 需要更新本地的链数据,即向该局部网络的"全链"数据持有者同步这个时间范围内区块的完整信息.在这个同步过程中,边缘节点 N_R 会匹配本地区块头中包含的根哈希与"全链"数据持有者链上的区块的根哈希.接着,从该区块开始,向前遍历,直到找到所需的数据,接着向该链的数据所有者边缘节点发起数据共享的请求.关于模糊查找涉及到的跨局部网络的数据共享请求我们在 5.4 节中进行详细描述.

接着我们看精确查找中跨局部网络的数据检索是如何实现的.在边缘节点已知交易哈希值的情况下,边缘节点会先向交易所处的局部网络中的边缘节点递交查询请求.边缘节点根据存储时间范围在终端侧区块链中确定创建时间最晚的检查点交易,然后从该检查点交易开始,从后往前遍历,直到找到对应的交易.此时,被请求边缘节点需要在边缘侧区块链上找到该交易对应的区块中的默克尔路径.默克尔路径如图 3 所示.

这条默克尔路径用于被请求边缘节点向请求边

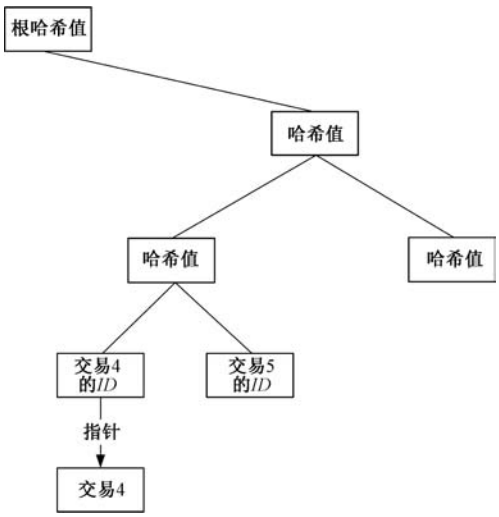

图 3　一条默克尔路径

缘节点证明交易 4 的数据内容不是杜撰的,而是真实存在的.当请求边缘节点收到这样一条默克尔路径之后,会进行一次哈希值验算,对比本地的"轻链"中对应区块的区块头中保存的根哈希值和得到的默克尔路径的根哈希值是否相同.若无误则说明在该默克尔路径中,没有数据被篡改,认为本次查询得到的默克尔路径以及对应数据真实存在且未被篡改.

我们在上面的描述中重点讨论了数据检索部分的内容,关于其中涉及的关于数据请求、数据共享部分的内容我们在 5.4 节中进行了详细描述.

通过这样的机制,边缘节点仅需要保存其他边缘节点的区块头链就可以对数据的正确性进行验证,同时边缘节点将本地的 DAG 网络(即终端侧区块链)中的交易按照时间顺序整理成单链区块链之后可以有效地降低遍历查找的时间.在实验 6.6 中有详细的实验验证.

在传统的区块链应用中,通常包含轻节点和全节点两个角色.全节点中保存全部的区块链数据,轻节点只保存区块链头信息.而在我们的系统中,每个边缘节点既是全节点又是轻节点.它既可以作为全节点向别的边缘节点提供交易数据查询服务.又像轻节点 1 样,仅需要保存别的边缘节点的区块链头信息就可以通过数据快速验证机制查询并验证得到别的边缘节点的交易数据.假定一个区块的全部数据量在 1M 左右,区块头数据大小为固定的 80 字节,那么对比传统区块链结构,每个边缘节点节省的存储空间是前者的 13000 倍左右.

下面给出边缘节点精确查找跨局部网络数据的完整流程代码:

算法 1　边缘节点检查算法.

输入:待检查交易列表

输出:节点贡献值

//遍历所有交易

For Transaction i in All Transactions

//获取当前交易的发布者 ID

ID_{Fulln} = GetID(i)

//根据交易的验证情况计算 C_i^P 的值

C_i^P ←validateAndCalculate($Transaction_{Verified}$)

//将计算好的 C_i^P 和对应的 ID_{Fulln} 保存在一个 Map 中

Map(ID_{Fulln}, C_i^P)

//累加 $N_{ID_{Fulln}}$

Map(ID_{Fulln}, $N_{ID_{Fulln}}$) ← $N_{ID_{Fulln}}$ +1

//计数器

Count++

//遍历所有节点

For FullNode ID in AllFullNode

//拿到上面计算的的 $N_{ID_{Fulln}}$

N = GetN (ID)

//计算 C_i^N

C_i^N ← checkBehaviors(N)

//计算得到节点 i 的贡献值 C_i

C_i ← calculateAll(C_i^P, C_i^N)

//将结果输出

PutCheckTransaction(ID, C_i, T_0)

5.4　边缘侧区块链上的数据共享

边缘侧区块链的结构类似于 Hashgraph,不同的链之间通过哈希指针完成区块验证以及数据同步的工作.关于 Hashgraph 的工作原理见第二章.

我们在边缘侧区块链上可以实现数据共享功能.我们将发生在多层区块链网络模型中的数据共享行为分为了两类,分别是"局部网络内的数据共享行为"和"跨局部网络的数据共享行为".其中,"局部网络内的数据共享行为"即在一个局部网络中,不同终端节点之间进行的数据共享行为,这种共享行为比较简单,相关数据只记录在终端层中,即共享的数据、数据请求者和数据所有者的信息以交易的形式被记录在终端侧区块链上."跨局部网络的数据共享行为"表示不同的局部网络之间发生的数据共享行为,这可能是某一个局部网络的边缘节点向其他局部网络中的数据所有者发起数据共享的请求行为,也可能是其他局部网络中的终端节点发起的请求.这一类数据共享行为较复杂,会同时与终端层和边缘层都发生交互.共享行为涉及到的关键数据会被记录在这两个局部网络对应的链上.下面我们描述

“跨局部网络的数据共享行为”的过程.

假设存在两个边缘节点 N_R 与 N_A,N_R 请求发送方,N_A 是请求接收方.在边缘节点 N_R 与边缘节点 N_A 的同步过程中(参照 Hashgraph 的同步方式),边缘节点 N_R 所处的局部网络中的某个终端节点 N_1 可能对边缘节点 N_A 所处局部网络上的某个数据感兴趣,或是边缘节点 N_R 本身对某个数据感兴趣.在第一种情况中,我们称边缘节点 N_R 是请求代理节点 N_{Proxy},而终端节点 N_1 则是真正的请求主体节点 N_{Main}.在第二种情况中,边缘节点 N_R 是请求主体节点 N_{Main},此时不存在请求代理节点 N_{Proxy}.

不管是上面哪种情况,边缘节点 N_R 首先都会打包创建一个数据请求区块,这个区块中包含了请求主体节点 N_{Main} 的公钥、当前时间戳、请求的内容(包含了区块的哈希、对应数据的哈希指针)、用途以及一个数据请求序号,上述信息会使用请求主体节点的私钥进行签名.接着,N_R 按照 Hashgraph 的同步方式同步边缘节点 N_A 所管理的链,即 N_R 将当前创建的数据请求区块通过哈希指针的方式指向自身链以及 N_A 所处的链.

接着,边缘节点 N_R 会通知边缘节点 N_A 有一个数据共享请求发生. N_A 此时会响应该请求,选择主动去同步 N_R 对应的链.同步到该数据请求区块后,边缘节点 N_A 会从中取出相应的信息,在自己的链上查找该数据内容,如果边缘节点 N_A 是该数据的所有者,我们称其为直接数据所有者 N_{Direct},此时 N_A 便可以直接根据情况决定是否要分享该数据内容;如果数据所有者是 N_A 所处局部网络中的某个终端节点 N_2,那么我们称 N_A 是间接数据所有者 $N_{Indirect}$,而 N_2 是直接数据所有者 N_{Direct}.在这种情况下,边缘节点 N_A 会与直接数据所有者即终端节点 N_2 通信,询问是否要分享该数据.终端节点 N_2 如果同意该数据请求,就会使用请求主体节点 N_{Main} 的公钥(从同步的数据请求区块中获得)对共享数据进行加密.不管是否接受该数据分享请求,边缘节点 N_A 都需要在自己的链上发布一个数据应答区块.如果同意该数据请求,加密后的数据内容就放在该数据应答区块中,数据应答区块中还包含一个状态码,用于表示是否接受数据请求,除此之外,区块中还包括时间戳、对应的数据请求序号以及附加信息.其中,数据应答区块中的数据请求序号与数据请求区块中的一致.附加信息可以用来记录拒绝数据请求的理由或是对该数据的一些额外说明.

最后,边缘节点 N_A 会通知 N_R 同步它的链,边缘节点 N_R 在接收到信号后就会主动同步边缘节点 N_A 所处的链.

5.5 自适应 PoW 算法

为了提高局部网络中终端节点的交易效率,终端节点使用一种叫做自适应工作量证明算法的共识协议.该算法会根据终端节点的行为动态调整 PoW 算法的执行难度.该算法在提高交易效率的同时,减少了局部网络中因为 PoW 算法难度降低而增加的安全风险.A-PoW 算法的正确运行依赖于局部网络中边缘节点检查机制的检查结果,在 5.2 节中详细描述了边缘节点检查机制的运行原理,我们下面描述边缘节点如何根据检查到的节点行为,计算该终端节点对应的贡献值.

我们定义系统中终端节点 i 具有贡献值 C_i 的属性.贡献值表示节点在其最近的生命周期中对区块链网络的贡献程度.影响贡献值的因素包括节点的懒惰行为、节点的恶意行为等.基于有向无环图的区块链需要解决懒惰问题.发生懒惰问题意味着在终端节点发布新交易时,会始终验证大多数终端节点都已经确认的“已确认”交易,而不是验证那些等待确认的交易.懒惰问题会降低终端侧区块链中的交易效率.不合法的交易也会对终端侧区块链造成破坏,并且破坏程度比懒惰问题更大.针对上面两类行为,由边缘节点定期对终端侧区块链中的交易进行检查,边缘节点会将检查结果打包进交易,并将所有待确认交易从结构上收束到“检查交易”上.接着,终端节点需要根据“检查交易”中的内容调整当前要执行的 PoW 算法的难度.

终端节点的 C_i 值越小,表示节点的贡献值越低,PoW 算法的执行难度也就越大,发布交易所需的时间也就越多.因此,自适应 PoW 算法是鼓励节点执行有益于终端侧区块链的操作,增加恶意节点发起攻击的成本,在提高交易效率的同时,防止系统安全性下降.

我们参考 Huang 等人[32]提出的基于节点荣誉值的一致性算法,设计了适用于本模型的节点贡献值计算公式.我们在其公式基础上进行了相应的改进,在计算节点贡献值时增加了时间衰减性的概念,即在公式中增加了时间衰减函数作为额外的权重参数,使得恶意行为对节点的影响效果更加持久,能够更加有效地惩罚恶意节点,增加其作恶成本;而贡献行为对节点贡献值影响的持续时间则要更短.我们在第 6 章中模拟了多种行为模式的节点,最终证明

了计算节点贡献值的公式是有效的,能够准确反映节点对区块链的贡献程度.

我们使用以下公式计算终端节点的贡献值:

$$C_i = \lambda_1 C_i^P + \lambda_2 C_i^N \tag{1}$$

其中 C_i^P 表示验证不同属性的交易对贡献值的影响,C_i^N 表示恶意行为对贡献值的影响,λ_1 和 λ_2 分别表示这两种影响的权重值.

C_i^P 的公式可被表示为

$$C_i^P = \sum_{k=1}^{n_i} T_1(t) * \alpha(b) \tag{2}$$

其中,n_i 表示全节点在一个单位时间内发布的交易数. $T_1(t)$ 代表时间衰减函数,该函数随着时间增加,值会逐渐减小.表示历史交间越近,重要性越大,$T_1(t)$ 可以被表示:

$$T_1(t) = N_0 e^{-\alpha(t+l)} \tag{3}$$

$$\alpha = \frac{1}{m}\ln\left(\frac{N_{init}}{N_{finish}}\right) \tag{4}$$

$$l = \frac{1}{\alpha}\ln\left(\frac{N_0}{N_{init}}\right) \tag{5}$$

公式(3)中的 N_0 表示函数的初始值,α 称为指数衰减变量,其中 l 表示向左的平移量,它可以让 $T_1(t)$ 的值不必从 N_0 开始衰减,而可以从任何位置处开始衰减.根据公式(4)和公式(5),衰减函数从 N_{init} 处开始衰减,经过 m 个时间单位衰减到 N_{finish} 处. $\alpha(b)$ 代表行为对贡献值影响的权重.当节点至少引用验证了一个待确认交易时,$\alpha(b)$ 的值为 N,否则,$\alpha(b)$ 的值为 $-N$.因此,贡献值计算公式会鼓励节点去主动验证那些"待确认"交易.

C_i^N 表示被检测出的恶意行为对贡献值的影响因子. C_i^N 可以被表示为

$$C_i^N = \sum_{k=1}^{n_i} T_2(i) * \beta(b) \tag{6}$$

系统对恶意行为的容忍度相对比较差,本次检查中发现的恶意行为不仅仅会影响到本次计算的贡献值,对以后的贡献值也会造成影响.如公式所示,$T_2(i)$ 是时间衰减函数,可以被表示为

$$T_2(i) = 0.8^i \tag{7}$$

C_i^N 计算公式中的$\beta(b)$是恶意行为的破坏程度的反映.假定当前系统,终端节点发布交易的阈值为 M,本次检查的 T 时间内发布的交易数量为 N.则 β(b) 可以表示为

$$\beta(b) = \begin{cases} 0, & N < M \\ k * (N - M), & N < M \end{cases} \tag{8}$$

在发布数量小于阈值时不做惩罚.当交易量超额达到 10%以上时,k 的值会比较大.当交易量超额不超过 10%的时候,边缘节点会认为是可以接受的误差范围.于是 k 的值会相对较小.

同时 $T_2(t)$ 是随着检查点交易而迭代衰减的函数.公式中的 i 表示恶意行为发生的检查点和当前正在进行的检查点之间间隔的检查点数目.即越久远的恶意行为对应的 $T_2(t)$ 会越小.

5.6　模型的安全性分析

本文提出的多层区块链网络模型是基于私链和联盟链技术构建的.其中,边缘侧区块链采用的是联盟链技术,终端侧区块链采用的是私链技术.根据边缘计算的常见场景,如车联网中的路边计算单元、物联网中的网关节点等往往是作为边缘计算场景中的边缘节点.这些边缘节点作为局部网络的一个计算中心默认是可信的.因此,我们结合边缘计算的场景特点,将边缘侧区块链设置为采用了联盟链技术,即每一个负责维护边缘侧区块链的边缘节点都是可信的,都需要在核心层的核心服务设备中进行身份识别和注册.这些边缘节点共同维护边缘层上的区块链.而位于终端层的终端侧区块链是一个私链,所有终端节点都需要在边缘节点上进行注册,并信任边缘节点后,才成为局部网络的成员.

终端节点在默认情况下是信任边缘节点的,但是在特殊情况下,边缘节点可能因为黑客入侵或者设备故障等原因导致在一段时间内产生的检查点交易中的内容出错.此时,终端节点仍有能力去发现这个错误.因为边缘节点在发布检查点交易前,会采用泛洪通信的方式将本次的检查结果以及"意见"(即是否认可该检查结果)发送到局部网络中的所有终端节点上.边缘节点使用自己的私钥对检查结果进行了签名,防止传输过程中被人修改.每个收到检查结果的终端节点都会对检查结果进行确认(确认检查结果是否由边缘节点发布以及检查结果是否计算正确),并在"意见"上使用自己的私钥进行签名."意见"可以是"认可"也可以是"不认可",终端节点根据检查结果对任一"意见"进行签名.签名后的"意见"以及检查结果被传递到下一个终端节点.同时,签名后的"意见"会与相应的终端节点 ID 进行绑定,边缘节点可以根据这个终端节点 ID 使用相应终端节点的公钥对这个"意见"进行解密,收集终端节点的意见.经过几轮消息传递后,边缘节点收集到所有终端节点签名后的"意见",如果有超过三分之二的终端节点对该检查结果"认可",边缘节点就可以将检查结果以及所有终端节点的"意见"一起打包进检查点交易,并发布到终端侧区块链上,完成对 DAG 区块

链的收束.此时,边缘节点才能打包相应的区块,将区块发布到边缘侧区块链上(区块中仍然包含当前局部网络中终端节点的待确认消息).

终端层的终端节点在检查点交易之后发布新交易时,可以对该检查点交易中包含的"意见"信息进行检查(即使用终端节点公钥对"意见"进行解密),判断局部网络中是否有超过三分之二的终端节点对该检查结果达成共识.如果没有达成共识,终端节点可以选择略过该检查点交易,继续在上一轮中的待确认交易中选择两个交易进行验证.边缘层上的其他边缘节点也可以通过区块中包含的"意见"去检查区块的发布是否合法,是否体现了其局部网络中的多数终端节点的意志.

除此之外,因为终端层侧区块链采用的是私链技术,即终端节点需要在边缘节点处进行注册才能在终端侧区块链上发布交易,这也在一定程度上防止了女巫攻击的发生.在 5.2 节中,我们也介绍了我们借鉴 IOTA 设计了符合本文模型的边缘节点检查机制,该检查点机制可以有效地防止双花攻击和寄生链攻击,可以维护网络的安全性和健壮性.

6 实 验

针对本文提出的多层区块链网络模型,我们进行了多组实验,从多个方面验证了该网络模型在数据检索效率和自适应工作量证明算法在提高正常节点交易效率、增加恶意节点作恶成本的有效性.具体地,我们使用模拟数据进行了数据检索的仿真实验,并与传统查找方案进行了对比;对自适应工作量证明算法的性能进行了评估,计算分析了不同节点行为对自适应工作量证明算法执行时间的影响,通过比较自适应工作量证明算法与传统工作量证明算法的交易效率,证明了自适应工作量证明算法的有效性.

我们使用 Java 语言搭建了仿真平台.我们的仿真平台按照我们提出的多层区块链网络模型结构从零开始完成了模型的搭建.模型主要分为了六个模块,分别是区块链通用模块、终端侧区块链模块、边缘侧区块链模块、自适应工作量证明算法模块、通用节点模块与核心层模块.区块链通用模块中包含了多个基础的区块链 Java 类,如区块类、默克尔树类等,其中主要实现了区块链中基础的操作,如打包、调用共识算法等操作.终端侧区块链模块与边缘侧区块链模块中包含了多个复杂的区块链 Java 类,通过类与类的组合以及类之间的继承实现了终端层的 DAG 式区块链、边缘层的多链平行结构区块链.自适应工作量证明算法模块中实现了传统的工作量证明算法,并在该算法基础上通过组合的方式,实现了自适应工作量证明算法 Java 类.该模块中还包含了对节点贡献值的计算 Java 工具类以及从节点贡献值到工作量证明算法难度设置的映射 Java 类.通用节点模块中实现了多个节点 Java 类,如边缘节点类、终端节点类、存储节点类等,这些节点类中通过继承并实现不同的接口类,来扩展该类节点能够进行的操作,同时增强了整个区块链系统的可维护性.核心层模块中实现了核心服务设备类,实现了对边缘节点的注册、智能合约的备案等操作.

以下五组实验都是在上述实现的区块链平台上进行的.该区块链仿真平台运行在的实验环境如下:处理器为 Intel(R)Core(TM)i5－6300HQ CPU@2.30 GHz,4 核,内存 8 GB,操作系统为 Windows 1064 位.

6.1 不同行为模式下节点的贡献值变化

终端节点的贡献值是一个历史值,边缘节点每次对局部网络中交易的检查得到的是终端节点在某一时间段内贡献值的变化值.边缘节点将变化值与终端节点的历史贡献值相加,就得到了当前终端节点的贡献值.理想的交易检测算法应该对恶意行为足够敏感.我们假设系统中终端节点的交易上限为 20.在该实验中,我们模拟了四种终端节点行为模式,其中设定节点 1 的行为是标准合法行为.实验设置了多个检查点,在每个检查点上分析不同行为模式的终端节点随着时间推移,其节点贡献值的变化.实验中,各个节点的行为模式设定如下:

(1)节点 1:模拟标准合法行为.在每个检查区间内都发布 20 笔完全不懒惰交易.

(2)节点 2:模拟持续少量的懒惰行为.在每个检查区间内发布的 20 笔交易中,混入一笔懒惰交易.

(3)节点 3:模拟低频次的恶意行为.在检查点三,发布 21 笔交易.发布交易数不超过规定交易上限的 10%,在其他检查点都为标准行为.

(4)节点 4:模拟高频次的恶意行为.在检查点三时,发布 23 笔交易,超过规定交易上限的 10%,在其他检查点都为标准行为.

如图 4,节点 1 和节点 2 的贡献值整体呈线性增长,节点 2 由于每个检查周期都有懒惰行为,所以增长速率比节点 1 慢.节点 2 模拟了在真实网络环境中的一种情况:终端节点在发布交易时,网络中只

有一个“待确认”交易,这意味着该终端节点必须验证一个“已确认”交易.这种懒惰并非出于终端节点的本意.所以边缘节点在计算贡献值的时候不会考虑终端节点的历史懒惰行为.对于节点 4,在第三次检查时发布的交易数量超过了其上限的 10%,不属于正常波动范围.可以看到节点 4 不仅在检查点 3 处受到了贡献值惩罚,在检查点 4、检查点 5,该节点的贡献值增速也比较缓慢,尽管节点 4 在后续阶段中发布的都是合法交易.该实验证明了边缘节点的检查机制对节点的历史恶意行为做出了比较好的反应.值得一提的是,持续做出好行为的终端节点的贡献值是呈线性增长的.对于本文的实时的连续系统而言,终端节点的贡献值很容易就累加到了一个非常高的程度.这样显然是不健康的,所以系统中设定贡献值的上限为 50.从上述实验中可以看到,一个完全诚实的节点,经过 5 次检查就能逼近贡献值的上限.

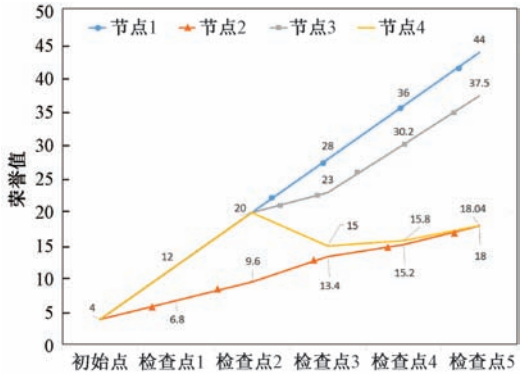

图 4　节点贡献值随行为变化情况

6.2　PoW 算法难度与执行时间的对应关系

边缘节点通过调整终端节点的贡献值来影响终端节点执行 PoW 算法的效率.最终达到降低诚实节点的挖矿开销、增加恶意节点的挖矿开销的目的.我们记录了 PoW 算法在不同难度下的执行时间,以制定最终的节点贡献值与 PoW 算法难度的映射关系.

图 5 中 x 轴表示 PoW 算法的执行难度,y 轴表示执行时间.最小难度为 1,最大值为哈希值的最长长度(这里取 24).传统工作量证明算法的执行过程是先把区块的各种信息(包括区块中的交易内容、前一个区块的哈希、当前区块高度以及时间戳)与一个随机值(nonce)进行拼接后,再进行哈希计算,并要求得到的哈希结果满足难度要求.假设难度值为 n ,即要求以二进制表示的哈希结果的前 n 位都是 0.设备通过不断调整随机值来得到符合难度要求的哈希结果.从上述描述中,我们可以发现,难度值每

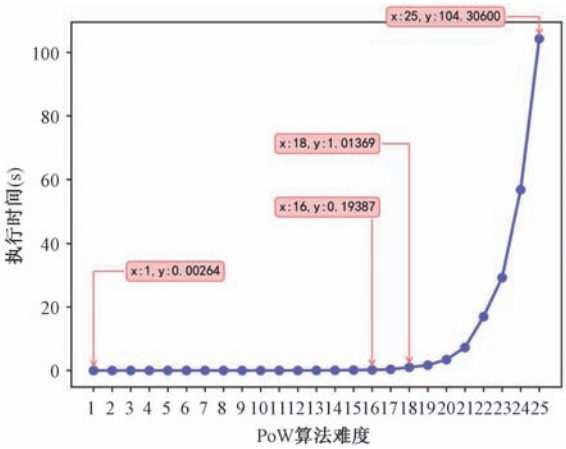

图 5　PoW 在不同难度的执行时间

增加 1,哈希结果前面需要为 0 的位数也需要相应增加一位,对应算出满足该难度要求的哈希结果的时间也需要增加一倍.图 5 的实验结果就是统计了在不同的 PoW 算法难度值情况下,需要多少时间才能算出符合难度要求的哈希结果(随机值 nonce 从 0 开始递增).每种难度值情况下,我们都进行了 50 组实验(区块的信息在每一组中均随机变化),并取平均值作为最终结果.我们可以从上图实验结果中看到执行时间随着难度上升呈指数级增长.同时,我们还可以看到,当难度小于 17 时,执行时间保持较低值.当难度大于 19 时,运行时间随难度增加而快速增加.因此,我们选择 18 作为 PoW 算法的基准难度值.

表 1　贡献值与 PoW 难度的映射关系

贡献值	难度	时间(秒)
$(-\infty,-15)$	25	104
$[15,0)$	24	56.7
$[0,1)$	23	29.2
$[1,2)$	22	16.9
$[2,3)$	21	7.25
$[3,4)$	20	3.43
$[4,5)$	19	1.68
$[5,6)$	18	1.01
$[6,10)$	17	0.36
$[10,15)$	16	0.19
$[15,20)$	15	0.12
$[20,25)$	14	0.06
$[25,30)$	13	0.03
$[30,35)$	12	0.01
$[35,40)$	11	0.006
$[40,45)$	10	0.003
$[45,50)$	9	0.001

表 1 给出了终端节点贡献值,PoW 难度以及相应耗时的映射关系.随着贡献值的升高,挖矿即发布交易的开销越来越小.而随着贡献值下降,执行

PoW 算法的耗时呈指数上升.这与系统的预期符合的很好.

6.3 边缘节点检查机制的时间复杂度分析

边缘节点的交易检查机制是文中提出的自适应 PoW 算法的关键实施过程,对于边缘节点而言,它所做的工作就是使诚实的终端节点用更少的开销发布交易,同时使不诚实的终端节点用更高的开销发布交易.

边缘节点的协调模块需要遍历某个时间段内当前局部网络中发布的所有交易,协调模块检查得到的结果将用于其他终端节点挖矿以及验证交易的依据.因此协调模块的性能将成为整个系统吞吐量的瓶颈,检查速度一定要足够快.下面将分析边缘节点的交易检查机制的时间复杂度.

如图 6,假设在一段时间内终端侧区块链的拓扑结构如图所示.圆形节点是 22 号交易直接以及间接验证的交易.边缘节点的协调模块需要对每一个 tips 进行一次深度优先搜索(DFS)才能完成对整个图的遍历.DFS 的时间复杂度是 $O(n+e)$,n 为顶点数,e 为边数.假设网络中全部为诚实节点,那么每一笔交易都会选择两笔交易进行验证,则每个顶点都对应两个出度.可以近似地认为在网络中存在 n 个交易时,有 $2n$ 条边.则 DFS 的时间复杂度可以近似为 $O(3n)$.

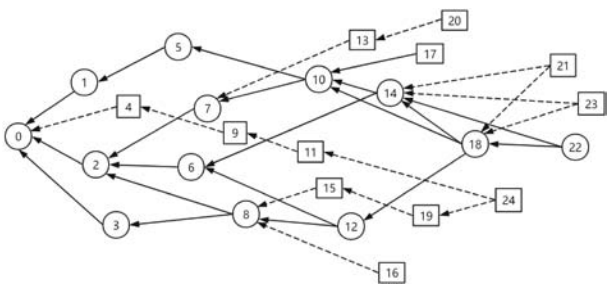

图 6 Tangle 网络拓扑结构

除了从网络中直接遍历,协调模块完全可以在本地维护一个下标值对应交易的线性表.这样通过线性遍历,时间复杂度将降低为 $O(n)$.而实际情况是协调模块确实需要在本地数据库中存储当前的区块链网络状态,所以采用这种线性表的方法是更好的方案.下文也将采用线性表遍历的办法进行实验.

对于协调模块而言,它是一种中心化的检查节点,可以说整个系统的共识都建立在边缘节点协调模块的完全诚实性上.所以在系统中对协调模块的挖矿难度设置的非常的低,以避免不必要的开销.

6.4 不同行为模式对应的时间开销对比

自适应 PoW 算法最终需要落实在边缘节点的挖矿耗时上的,我们设计了如下两组实验,用于测试终端节点在不同的行为模式下,它们的挖矿耗时的变化情况.

6.4.1 懒惰行为对节点发布交易的时间影响

实验设置了一组实验组和一组对照组.节点 1 为对照组,节点 2 为实验组.该组实验主要测试了系统对懒惰行为的惩罚机制以及对诚实行为的激励机制.

节点 1:在每个检查区间都发布 20 笔完全不懒惰交易.

节点 2:在每个检查区间发布的 20 笔交易中,混入一笔懒惰交易.

如图 7 所示,节点 1 和节点 2 的贡献值都是呈线性增长的.唯一的区别是节点 1 相对节点 2 的增速更快.如图所示,节点 1 每次发布 20 笔交易的速度越来越快,但是增速是越来越慢的,最后增速逐渐趋于 0.节点 2 每次发布 20 笔交易的速度也是越来越快的,但是增速低于节点 1.这样的实验结果表示该算法对懒惰行为有一定的抵抗力,在鼓励节点验证新交易这方面有不错的效果.诚实节点的整体交易效率是懒惰节点的 3～4 倍.

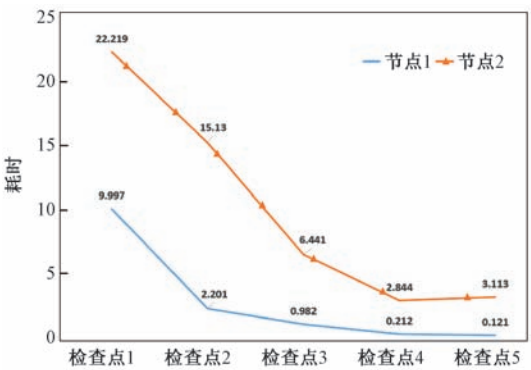

图 7 懒惰行为对照组实验

6.4.2 恶意行为对节点发布交易的时间影响

实验对节点恶意行为对挖矿耗时的影响进行了分析.该组实验同样设置了一个对照组和一个实验组,其中的恶意行为指的是终端节点发布过量无效信息.

对照组:节点 1 的行为模式是在检查点 3 处发布了 21 笔交易,该交易笔数不超过交易上限的 10%,其他时间都为标准行为.

实验组:节点 2 在检查点 3 处发布 23 笔交易,该交易笔数超过交易上限的 10%,其他时间都为标

准行为.

如图 8 所示,节点 1 与节点 2 在检查点 1 处与检查点 2 处,发布交易的耗时都差不多.此时它们都只做了标准合法行为,因此挖矿开销降低的很快.在检查点 3 时,两个节点都发布了过量的交易.系统认为超过量在 10%之内是可以接受的,属于误差范围,因此惩罚力度比较轻.而对于节点 2 而言,它发布的交易超过了上限的 10%,惩罚力度比较大,并且带有比较严厉的历史行为惩罚.尽管在检查点 4 和检查点 5,两个节点做的都是标准合法行为,但由于历史惩罚值的不同,导致了节点 1 和节点 2 的挖矿耗时出现了较大差距.最终两者的平均挖矿耗时相差了 2 倍.同时,在历史惩罚机制的作用下,在检查点 5 时节点 1 更快了节点 2 约 17 倍.

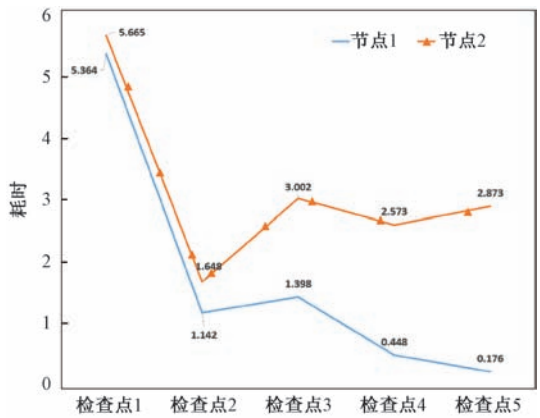

图 8　恶意行为对照组实验

6.5　A-PoW 算法与 PoW 算法在不同行为模式下的耗时对比

通过以上两组实验,我们可以认为边缘节点的交易检查机制对于懒惰行为以及以发布过量交易为例的恶意行为都有比较好的检测效果.本文自适应 PoW 算法在打击恶意行为的层面上有比较好的效果,同时该算法也能提高系统的吞吐量,为诚实节点降低挖矿的开销.下面给出两组实验,对比传统 PoW 和本文的 A-PoW 算法的耗时.

节点 1:执行 A-PoW 算法,在检查点 2 时发布了过量 10%的交易,其他时间都是标准合法行为.

节点 2:一直执行难度为 18 的传统 PoW 算法,属于对照组.

如图 9 所示,节点 2 的耗时主要在 20—30 秒之间波动,而节点 1 的耗时曲线变化比较明显.节点 1 在检查点二时的恶意行为被检测到,贡献值降低从而导致挖矿难度上升,进而导致耗时急剧增加.

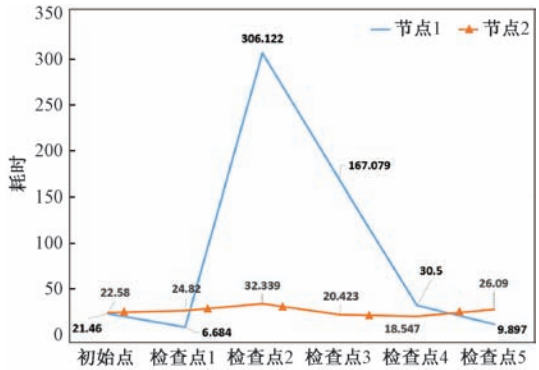

图 9　不诚实节点执行 A-PoW 算法与 PoW 算法耗时对比

节点 1:执行 A-PoW 算法,且一直执行合法标准行为;

节点 2:一直执行难度为 18 的传统 PoW 算法,属于对照组.

如图 10 所示,节点 1 的算法执行耗时降低的非常快,在检查点 5 时耗时降低到了 0.121s.

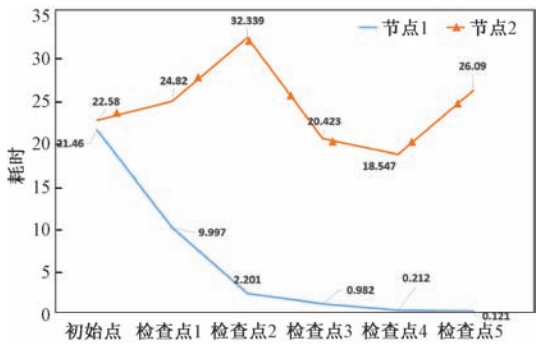

图 10　诚实节点执行 A-PoW 算法与 PoW 算法耗时对比

可以看到,文中提出的 A-PoW 算法对比传统 PoW 算法,在提高交易效率方面的效果比较明显,整体的交易效率提高了 4～5 倍.当节点 1 在检查点 5 时,贡献值接近最大值 50,此时交易效率提升了将近 100 倍 .综上所述,自适应 PoW 算法可以提高正常节点的交易效率,同时增加作恶节点的作恶成本,系统的安全性得到了保障.

6.6　数据快速检索机制

边缘侧区块链通过 MerkleProof 向别的边缘节点提供可靠的查询服务,MerkleProof 指的是默克尔树中从根节点到叶子节点的一条路径.

实验角色被分为了服务端和客户端,服务端是向其他节点提供 MerkleProof 查询的边缘节点,客户端指的是发起 MerkleProof 查询的节点.在服务端,查询主要分为两个步骤.首先通过客户端提供的交易时间戳确定这笔交易可能存在的区块范围,定位该区块

后,通过该区块的默克尔树生成一条 MerkleProof 路径,并将该路径返回给请求的客户端.

生成 MerkelProof 的时间主要取决于默克尔树包含的交易的数量.下图给出了默克尔树中交易数量与算法执行时间的对应关系.

如图 11 所示,纵坐标为算法的执行时间,横坐标为一棵默克尔树中包含的交易数量.虽然随着树中交易数量的增多,算法的执行时间也在增加.但是值得一提的是,尽管交易数量达到了 10000,整体的时间开销也在 166ms 左右.而对于系统设定的一个区块应该包含 4000-6000 笔交易的情况,时间开销大致在 20ms 以下.所以默克尔树的快速验证模块的时间开销并不是很大,算法效率比较高.

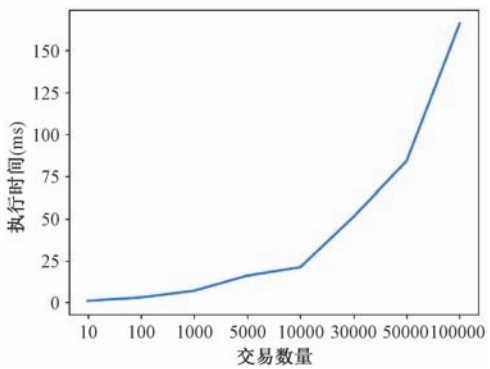

图 11 默克尔树中交易数量与算法执行时间对应关系

除了默克尔树部分,查询服务的时间开销还受到对应的终端侧区块链上的交易数量的影响.对于传统的纯 DAG 结构区块链,查询一般采用遍历查找的方式.在本文的系统中,边缘节点在完成对终端侧区块链中某一个时间段内发布的交易的检查工作后,会将这些被检查的交易打包成一个边缘侧区块链上的区块.双区块链的结构可以被看作是在原本的 DAG 结构区块链上按照时间对局部网络内的交易进行了时间上的分块排序.边缘节点首先通过时间戳定位到在边缘侧区块链上相应的区块区间,然后再在这个区间内进行遍历查找.

图 12 中显示了在不同交易数量下,各种查询方案对应的执行时间.从图中可以看到,传统的有向无环图查找方案是从创世区块开始遍历的方法,其时间开销随着交易数量呈线性增长.而在本文的系统中,将 DAG 网络中的交易按照时间顺序打包分块.在给出了时间戳的场景下将能显著降低查找的时间开销,对比传统遍历查找的方法时间效率提高了 4-7 倍.

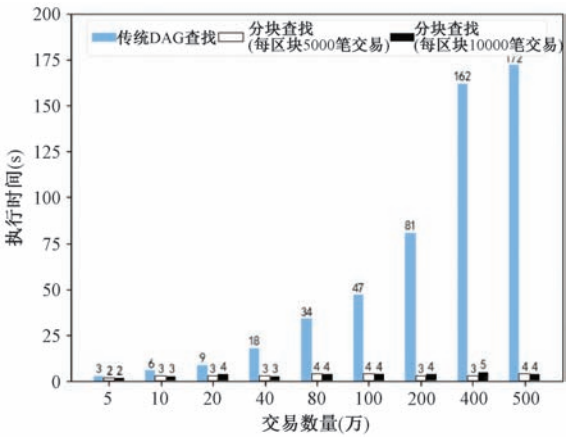

图 12 不同的查找方案在不同的交易数量下的时间开销

7 结 语

本文提出了一种适用于边缘计算场景的多层区块链网络模型.具体地,该模型利用智能合约技术并利用分块查找的思想,加快了在该模型中检索某一特定区块内容的速度.同时,利用双区块链的结构,减少了边缘节点需要存储的区块链的容量大小.同时,终端节点通过执行自适应工作量证明算法,提高了交易效率,降低了系统的安全风险.我们通过仿真实验说明了本文提出的多层区块链网络模型能有效加快边缘节点对某一特定数据的检索速度,并能有效提高正常节点的交易效率并惩罚作恶节点.我们还通过数据分析证明了该模型能有效节省边缘区块链需要占用的存储空间大小.

致 谢 在此,我们向对本文研究工作给予支持和建议的同行表示衷心的感谢!

参 考 文 献

[1] Shi Wei song, Zhang Xing zhou, Wang Yi fan, Zhang Qing yang. Edge computing: state-of-the-art and future directions. Journal of Computer Research and Development, 2019, 56(1):69

[2] Wei song Shi, Jie Cao, Quan Zhang, Youhxuizi Li, Lanyu Xu. Edge computing: Vision and challenges. Internet of Things Journal, 2016, 3(5):637-646

[3] Alexandru. Blockchain based distributed control system for edge computing//Proceedings of the 2017 21st International Conference on Control Systems and Computer Science (CSCS). Bucharest, Romania, 2017: 667-671

[4] Liu Hong, Zhang Yan, Yang Tao. Blockchain-enabled security in electric vehicles cloud and edge computing. IEEE Network, 2018, 32(3):78-83

[5] Zhi Li, W. M. Wang, Guo Liu, Layne Liu, G. Q. Huang. Toward open manufacturing: a cross-enterprises knowledge and services ex-change framework based on blockchain and edge computing. Industrial Management Data Systems, 2018, 118(9):303-320

[6] Claus Pahl, Nabil El Ioini, Sven Helmer. A decision framework for blockchain platforms for iot and edge computing// Proceedings of the IoTBDS. Funchal, Madeira, 2018: 105-113

[7] Jiawen Kang, Rong Yu, Xumin Huang, Maoqiang Wu, Sabita Mahar-jan, Shengli Xie, Yan Zhang. Blockchain for secure and efficient data sharing in vehicular edge computing and networks. IEEE Internet of Things Journal, 2018, 6(3):4660-4670

[8] Yuan-Yong, Wang Fei-Yue, Blockchain: The state of the art and future trends, 2016, 42(4):481-494(in Chinese)
(袁勇,王飞跃. 区块链技现状与展望. 自动化学报, 2016, 42(4):481-494)

[9] Huma Pervez, Muhammad Muneeb, Muhammad Usama Ir-fan, Ir-fan Ul Haq. A comparative analysis of dag-based blockchain architectures//Proceedings of the 2018 12th International Conference on Open Source Systems and Technologies (ICOSST). Lahore, Pakistan, 2018: 27-34

[10] Zhou, Qiheng, et al. Solutions to scalability of blockchain: a survey. IEEE Access,2020, 8 (2020): 16440-16455

[11] Tiwari A K, Jana R K, Das D, et al. Informational efficiency of bitcoin—an extension. Economics Letters, 2018, 163: 106-109

[12] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, Huaimin Wang. An overview of blockchain technology: architecture, consensus, and future trends//Proceedings of the 2017 IEEE international congress on big data(Big Data congress), Boston, USA,2017: 557-564

[13] Eyal, I., Gencer, A. E., Sirer, E. G., Van Renesse, R. Bitcoin-ng: a scalable blockchain protocol//Proceedings of the 13th {USENIX} symposium on networked systems design and implementation ({USENIX} NSDI 16), Boston, USA, 2017: 45-59

[14] Ehmke C, Wessling F, Friedrich C M. Proof-of-property: a lightweight and scalable blockchain protocol//Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain. Gothenburg, Sweden, 2018: 48-51

[15] The swirlds hashgraph consensus algorithm: fair, fast, byzantine fault tolerance, https://eclass. upatras. gr/modules/document/file. php/CEID1175/Pool-of-Research-Papers% 5B0%5D/31. HASH-GRAPH. pdf 2016,5,31

[16] Leandros A Maglaras, Ali H Al-Bayatti, Ying He, Isabel Wagner, Helge Janicke. Social internet of vehicles for smart cities. Journal of Sensor and Actuator Networks, 2016, 5(1):3

[17] Zhenyu Zhou, Caixia Gao, Chen Xu, Yan Zhang, Shahid Mumtaz, Jonathan Rodriguez. Social big-data-based content dissemination in internet of vehicles. IEEE Transactions on Industrial Informatics, 2017, 14(2):768-777

[18] Neeraj Kumar, Sudip Misra, Joel JPC Rodrigues, Mohammad S Obaidat. Coalition games for spatio-temporal big data in internet of vehicles environment: A comparative analysis. IEEE Internet of Things Journal, 2015, 2(4):310-320

[19] Chien-Ming Chen, Bin Xiang, Yining Liu, King-Hang Wang. A secure authentication protocol for internet of vehicles. IEEE Access, 2019, 7:12047-12057

[20] Xiang Sun, Nirwan Ansari. Edgeiot: Mobile edge computing for the internet of things. IEEE Communications Magazine, 2016,54(12):22-29

[21] Wenyu Zhang, Zhenjiang Zhang, Han-Chieh Chao. Cooperative fog computing for dealing with big data in the internet of vehicles: Architecture and hierarchical resource management. IEEE Communications Magazine, 2017, 55(12):60-67

[22] Ali Dorri, Marco Steger, Salil S Kanhere, Raja Jurdak. Blockchain: A distributed solution to automotive security and privacy. IEEE Communications Magazine, 2017, 55(12): 119-125

[23] Mohamed Amine Ferrag, Makhlouf Derdour, Mithun Mukherjee, Ab-delouahid Derhab, Leandros Maglaras, Helge Janicke. Blockchain technologies for the internet of things: Research issues and challenges. IEEE Internet of Things Journal, 2018, 6(2):2188-2204

[24] Xiaoliang Wang, Pengjie Zeng, Nick Patterson, Frank Jiang, Robin Doss. An improved authentication scheme for internet of vehicles based on blockchain technology. IEEE Access, 2019, 7:45061-45072

[25] Yongjun Ren, Yan Leng, Yaping Cheng, Jin Wang. Secure data storage based on blockchain and coding in edge computing. Mathematical Biosciences and Engineering, 2019, 16(4):1874-1892

[26] Pradip Kumar Sharma, Mu-Yen Chen, Jong Hyuk Park. A soft-ware defined fog node based distributed blockchain cloud architecture for iot. IEEE Access, 2017, 6:115-124

[27] Lei Zhang, Mingxing Luo, Jiangtao Li, Man Ho Au, Kim-Kwang Ray-mond Choo, Tong Chen, Shengwei Tian. Blockchain based secure data sharing system for internet of vehicles: A position paper. Vehicular Communications, 2019, 16:85-93

[28] Zhou Su, Yuntao Wang, Qichao Xu, Minrui Fei, Yu-Chu Tian, Ning Zhang. A secure charging scheme for electric vehicles with smart communities in energy blockchain. IEEE Internet of Things Journal, 2018, 6(3):4601-4613

[29] Fan Zhang, Ittay Eyal, Robert Escriva, Ari Juels, Robbert Van Re-nesse. {REM}: Resource-efficient mining for blockchains//Proceedings of the 26th {USENIX} Security Symposium ({USENIX} Security 17), Vancouver, Canada, 2017: 1427-1444

[30] Jiangshan Yu, David Kozhaya, Jeremie Decouchant, Paulo

Esteves-Verissimo. Repucoin: Your reputation is your power. IEEE Transactions on Computers, 2019, 68(8): 1225-1237

[31] Cao B, Li Y, Zhang L, et al. When Internet of Things meets blockchain: Challenges in distributed consensus. IEEE Network, 2019, 33(6): 133-139

[32] Huang J, Kong L, Chen G, et al. B-IoT: Blockchain driven Internet of Things with credit-based consensus mechanism// Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). Texas, USA, 2019: 1348-1357

YIN Yu-Yu, Ph. D., professor. His research interests include service computing, edge computing, software engineering and blockchain system.

YE Bing-Yue, master student. His research interests include blockchain system and edge computing.

LIANG Ting-Ting, Ph. D., associate professor. Her research interests include data mining, machine learning and edge computing.

DUAN Hong-Yue, master student. His research interest is blockchain.

LI You-Hui-Zi, Ph. D., associate professor. Her research interests include edge computing, computer system.

WAN Jian, Ph. D., professor. His research interests include grid computing, service computing and wireless sensor network.

Background

The data generated in edge computing scenarios has problems such as poor data security and weak anti-tampering modification. Therefore, more and more people use blockchain technology with immutable and decentralized characteristics to alleviate these problems in edge computing scenarios. In edge computing scenarios, there are usually a large number of terminal nodes producing various types of data, which requires the proposed blockchain storage scheme to have higher transaction efficiency to speed up the response speed of terminal-side devices. At the same time, with the development of edge computing technology, the large volume of generated data from terminal nodes causes the expansion of blockchain data.

This paper proposes a multi-layer blockchain network model for edge computing scenarios. This model divides the network into multiple partial networks, each of which contains multiple terminal nodes and a small number of edge nodes. The terminal nodes produce data, package the data into transactions, and upload it to the terminal side blockchain. Edge nodes regularly perform security verification on the terminal side blockchain, package and upload the verified data to the edge side blockchain at the edge layer. The edge-side blockchain is a multi-chain storage structure. This structure allows the edge node to only store the information chain corresponding to the current local network, reducing the amount of blockchain data that needs to be stored. We also proposed an adaptive PoW algorithm. The algorithm can calculate the contribution value of the nodes according to the their behavior in the system, and map the execution difficulty of the PoW algorithm according to the contribution values of the nodes. We have conducted experiments to verify the effectiveness of the blockchain network model. The experimental results show that compared to the directed acyclic graph structure blockchain, the proposed network model can effectively reduce the storage space occupation. Compared with the traditional PoW algorithm, the transaction efficiency of normal nodes that execute the adaptive algorithm is improved by 4-5 times. The speed of finding a specific data hash in this model is 4-7 times faster than the traditional direct acyclic graph blockchain model.