

doi: 10.12068/j.issn.1005-3026.2021.02.003

# 基于区块链的电力数据资产化及交易系统设计

王柯元<sup>1,2</sup>, 于雷<sup>1</sup>, 颜拥<sup>3</sup>, 孙毅<sup>1,2</sup>

(1. 中国科学院计算技术研究所, 北京 100190; 2. 中国科学院大学 计算机科学与技术学院, 北京 100049;  
3. 国网浙江省电力有限公司电力科学研究院, 浙江 杭州 310014)

**摘要:** 在传统的中心化数据交易系统中, 电力数据类的敏感数据存在因可复制而易被滥用等问题. 为此, 设计了基于区块链技术的电力数据资产化及交易系统. 利用区块链技术的密码算法、多中心对等架构、分布式多方共识协议, 使参与记录数据交易信息的多方实体仅需通过对区块链技术的信赖建立多方信任关系. 该系统以超级账本项目的 Fabric 联盟链为框架, 借助区块链的技术特性实现了电力数据交易业务的安全、透明、可追溯、不可篡改, 并加入了监管机制. 该系统实现了应用层 Web 服务的开发, 便于使用. 对该系统在一台设备上模拟多节点部署进行测试, 结果表明, 该系统在前后端级联后运行正常, 满足设计需求, 维护成本低, 安全性提升. 相较于传统数据交易系统, 基于区块链的电力数据资产化及交易系统展现了综合的功能和安全优势.

**关键词:** 电力数据; 交易系统; 区块链; 超级账本; 系统设计与实现

中图分类号: TP 311.133.1

文献标志码: A

文章编号: 1005-3026(2021)02-0166-08

## Capitalization and Trading System Design of Power Data Based on Blockchain

WANG Ke-yuan<sup>1,2</sup>, YU Lei<sup>1</sup>, YAN Yong<sup>3</sup>, SUN Yi<sup>1,2</sup>

(1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China; 2. School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China; 3. State Grid Zhejiang Electric Power Research Institute, Hangzhou 310014, China. Corresponding author: YU Lei, E-mail: yulei2008@ict.ac.cn)

**Abstract:** In the traditional centralized data trading system, sensitive data such as electric power data are easily abused due to duplicability. To address this, the blockchain-based power data's capitalization and trading system was proposed. Considering the trust in blockchain technology, a multi-party trusty network can be established between multiple parties in distributed ledger with cryptographic algorithms, multi-centralized architecture and distributed multi-party consensus protocol. The system was developed with Hyperledger's consortium blockchain framework Fabric, which made power data trading safe, transparent, traceable and immutable by means of the technical characteristics of the blockchain. In addition, a supervision mechanism was introduced. The system also provided Web service as application layer for ease of use. Finally, the system was tested by simulating multi-node deployment on a single device. And the results show that the system runs well in integrated test, meets the design requirements, and has low maintenance costs and enhances security. Compared with the traditional data trading system, the blockchain-based power data's capitalization and trading system demonstrates comprehensive functionality and security advantages.

**Key words:** power data; trading system; blockchain; Hyperledger; system design and implementation

收稿日期: 2020-06-18

基金项目: 国家重点研发计划项目(2019YFB1404903); 国家自然科学基金资助项目(61972382, 61672499, 61772502); 北京市科技计划项目(Z181100003218018); 国家电网有限公司科技部项目(5100-201957475A-0-0-00).

作者简介: 王柯元(1997-), 男, 辽宁沈阳人, 中国科学院计算技术研究所博士研究生; 孙毅(1979-), 男, 山东青岛人, 中国科学院大学教授, 中国科学院计算技术研究所研究员, 博士生导师.

随着互联网与计算机技术的快速发展, 数据信息爆炸式增长, 对这些数据的储存、处理、分析带动了大数据的发展; 但伴随而来的是对数据的规范化管理问题, 这在数据隐私保护、数据确权、数据共享、数据交易方面体现得尤为突出. 由于数据的可复制性, 将数据交给其他人使用, 就相当于把数据的归属权交了出去. 在云计算时代来临之前, 大多数的商业模式倾向于自己管理自己的数据, 但仍然存在将数据与他人共享后会被二次滥用的问题. 云计算的大规模应用降低了维护数据的成本, 但依然难以掌握数据的流向.

电力大数据是推动电力行业发展的重要元素. 电力大数据涉及到用户隐私、商业机密等, 不应对社会公众开放, 因此需要设计一种仅限于电网公司、电力生产企业、科研实体间共享或交易电力数据的安全、隐私、互信的平台. 早期, 电力数据的管理平台是以中心化存储的方式管理数据. Rusitschka 等<sup>[1]</sup>提出利用云计算实现智慧电网中的信息共享, Bobba 等<sup>[2]</sup>提出在智慧电网中将敏感数据通过加密的方式在系统中流转, 这类方式天然地存在单点失效和数据易被滥用的问题.

区块链技术的出现, 为数据的规范化管理带来可行的解决方案, 在数字资产交易、数字版权保护、产品溯源等方面得到了大量应用. Pan 等<sup>[3]</sup>提出基于区块链的碳排放权交易应用, Mehta 等<sup>[4]</sup>提出一套基于区块链的去中心化图片共享及版权保护方案, Casino 等<sup>[5]</sup>提出利用区块链解决食物供应链溯源问题. 区块链技术在不同领域发挥了数据业务安全、透明、可溯源、不可篡改的特点.

然而, 目前采用区块链进行电力数据管理的研究相对较少, 且尚有不足. Liang 等<sup>[6]</sup>提出基于区块链的工业物联网数据传播, 利用区块链将多个实体相连, 分布式地交易与传输数据信息; 但其采用的是数据内容的直接共享, 对存储与通信造成了较大压力, 且缺少对数据交易的监管机制. 李旭等<sup>[7]</sup>提出基于区块链的电力数据共享机制, 描述了可能的系统架构, 但也仅限于理论上的愿景. 本文针对电力数据的资产化和交易详细设计了基于联盟链的系统架构并予以实现, 解决了数据规范化管理的问题, 有以下 3 个创新点:

1) 电力数据持有者可以自行定价发布数据资产, 数据资产可以从发布者处购买, 也可以来自其他购买者的转让 (而非复制), 避免数据被随意复制与滥用;

2) 引入管理员审查机制, 加强区块链系统的可监管性, 对于敏感的数据交易提供一层保护;

3) 电力数据的内容或访问权和交易记录存储在区块链上, 由多方实体共同维护, 保障了电力数据的安全性和数据流转的透明、可追溯及不可篡改.

本文采用 Linux 基金会超级账本项目的联盟链架构 Hyperledger Fabric<sup>[8-9]</sup>, 使用 fabric-sdk-go 接口, 结合 Go Web 的微框架 Gin-gonic, 访问底层区块链服务进行系统开发, 完成了电力数据资产化及交易系统的后端业务逻辑, 并提供简单的 Go Web 可视化界面, 方便用户通过 Go Web 进行注册、充值、销户、数据发布、数据转让、取消发布、查询记录等功能.

## 1 区块链技术及超级账本

### 1.1 区块链技术

区块链概念起源于比特币<sup>[10]</sup>. 区块链受到广泛认可的定义是去中心化的分布式账本系统, 系统中的各个节点无需相互信任就可以通过统一的共识机制共同维护一个账本.

区块链采用一种分布式、去中心化的计算与存储架构. 在宏观上, 网络上对等的节点分布式共享相同的数据, 即账本; 没有中心化机构控制, 而是由全网节点协作完成交易的验证和存储. 在微观上, 区块链是将交易等信息放入区块中, 有序地串联成链条, 并通过共识机制和数字签名等, 保证区块的合法性和其中交易的有效性. 在区块链系统运行过程中, 区块间按照密码学算法链接在一起, 这样的组织设计可以很容易地验证数据是否被篡改、追踪历史交易数据和保证整个区块链系统的安全<sup>[11-12]</sup>. 系统中各个节点共同组成一个 P2P 网络, 均要执行、验证和记录相同的交易, 没有一个中心化机构可以干预交易的执行顺序和结果. 因此, 该架构具有很强的鲁棒性.

区块链解决了如何在无可信的中心机构且多方信息不对称的情况下, 建立共识并形成多方参与的、去中心或多中心的价值和信任网络. 区块链的应用场景有很多, 凡是涉及金钱的金融领域, 涉及有不可篡改需求的物流溯源、版权保护等, 涉及信息透明的教育、慈善、医疗等, 都可以使用区块链技术解决问题<sup>[13]</sup>.

根据权限管理可以将区块链分为公链、私链和联盟链. 公链平台均被设计成去中心化的, 是因为其设计假设是以没有任何可以信任的节点作为前提. 去中心化在区块链中并不是目的, 而是达到可信这一目的的手段; 在实际应用中, 如果有一些

可信的元素,也是可以利用的.所以联盟链介于低信任级别的公链和高信任级别的私链之间,是多中心的模式.本文设计的基于区块链的电力数据资产化及交易系统,应用于电网公司、电力生产企业与科研实体等,且不向公众开放,因此联盟链的权限管理方式最合适.

## 1.2 超级账本

超级账本(Hyperledger)是由Linux基金会牵头在2015年12月启动的一项企业级区块链开发及应用的开源项目.超级账本首次将区块链技术引入到联盟链的应用场景中,为未来基于区块链技术打造高效率的商业网络打下基础.相比于制定一个区块链标准,超级账本鼓励通过社区协作开发区块链技术,鼓励开源来研发具有知识产权的技术,并最终采用一套标准.Hyperledger Fabric是最早加入到超级账本项目中的顶级项目,是面向企业的分布式账本平台,引入了权限管理,具有账本、智能合约,支持可插拔、可扩展,由IBM等企业于2015年底提交到社区,是首个面向联盟链场景的开源项目.目前Hyperledger Fabric已更新至2.1版本.

Hyperledger Fabric从众多区块链系统中脱颖而出,源于它是私密的并且是权限化的.相比于允许以未知身份参与网络的开放式权限系统,Hyperledger Fabric网络的成员通过注册可信会员服务提供商(membership service provider, MSP)来保证系统的私密性.Hyperledger Fabric支持多种可热插拔选项,账本数据可以以多种格式来存储,共识机制可以随时切换开关,并支持多种MSP.Hyperledger Fabric还提供了创建通道(channel)的功能,允许一组参与者创建单独的交易账本;如果几个参与方单独创建一个通道,那么仅有这些参与者拥有该通道的账本副本.

在共识机制方面,Hyperledger Fabric允许网络初始者选择最能反映参与者之间存在关系的共识机制,目前包括solo, kafka<sup>[14]</sup>.Hyperledger Fabric中的节点与公链中节点的概念有所不同,Hyperledger Fabric中的节点有一定的分工,分别是:排序节点、背书节点和记账节点.同时各节点又有所在的组织,每个组织有一个主节点负责对外通信.当一个客户端发出一笔交易会先发向背书节点模拟运行一次,再将经过背书节点签名过的交易结果提交至排序节点,排序节点通过solo或kafka集群的方式得到共识结果,然后广播至各组织的主节点,主节点再通过gossip协议<sup>[15]</sup>发往该组织的各节点,验证并记录.

Hyperledger Fabric为其搭载的应用提供了与底层区块链交互的gRPC接口及封装接口的SDK.目前SDK支持Go, Java, Python和Node.js语言.上层应用可以通过SDK访问区块链网络,如访问账本、发布交易、执行链码、触发事件、权限管理等.Hyperledger Fabric的智能合约是用链码(chaincode)实现的,并且被区块链外部应用程序所调用,以此来与账本交互.链码目前支持的语言是Go和Java,未来将支持更多语言.

总的来说,用Hyperledger Fabric实现的联盟链,满足本文系统对底层区块链的要求,能够低成本地维护各实体间的节点网络,提供迅速的交易处理,具有良好的扩展性,保证了电力数据的安全性.

## 2 电力数据资产化及交易系统的设计与实现

### 2.1 系统需求

电网公司与电力生产企业按照不同地域可以划分为多个电网分公司和多个电力生产企业,数据每天增量更新,存储在各自的数据管理平台.科研实体从电网公司与电力生产企业获得电力数据,经过计算分析可生成加工后的电力数据或分析报告.基于区块链的电力数据资产化及交易平台的核心需求是在电网公司、电力生产企业、科研实体及其他互联网用户之间实现电力数据资源的共享及流通交易,在该系统中不同的实体均以用户身份参与交易.电力数据的持有用户可以将数据资源在链上定价发布或取消发布,关注资源的买方用户支付相应的价格后可以得到数据资源的所有权,且数据支持多次转卖.该系统需实现9项功能:用户开户、用户充值、用户销户(该用户发布的资源同时全部自动取消发布)、资源发布、资源取消发布(新买家不可再购买该资源,但旧买家依然持有该资源)、资源转让(资源所有权的变更:发布者仍然拥有资源,购买者只能转让资源)、用户查询、资源查询、资源历史记录查询.

### 2.2 系统设计

基于区块链的电力数据资产化及交易系统是在Hyperledger Fabric联盟链网络上开发的平台,具有多中心化、可扩展、低成本、电力数据安全可溯源等优点.它的整体架构如图1所示.

整体架构按照调用关系自上而下可分为4层.最上层是用户层,各组织的用户可以通过用户客户端将事务内容发送至负责用户管理和监管审

查的管理员客户端。第二层是 Web 服务层,选用 Go 语言的 Gin-gonic 微服务框架通过 fabric-sdk-go 实现与区块链链码交互的服务接口。第三层是电力数据资产化及交易系统的业务模型层,是开发过程中的核心环节,通过链码 (Hyperledger Fabric 中的智能合约) 定义区块链中存储的数据结构,实现相应的业务功能。账本记录了每一笔交易,所有的历史交易记录和用户与资源信息都可以通过访问账本来追溯。最底层是网络层,在将业务模型部署到区块链网络节点后,实现节点间的协同通信。

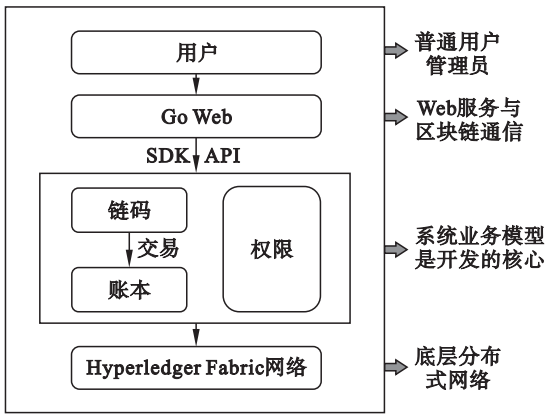


图 1 系统整体架构

Fig. 1 Overall system framework

该系统加入了监管机制:用户层存在普通用户和管理员两种身份。加入监管机制出于以下考量:与普通的商业交易平台不同,电力数据的敏感性使整个系统的用户接入和交易内容需要一个有最高权限的组织来监管,比如国家电网总部等。如果缺少监管,虽然区块链的不可篡改性保证了后续查询的操作可以追溯历史记录,但依然无法改变发生过的既定事实;因此这一层的特殊设计满足了开发区块链系统中的一大难点——可监管性,也一定程度上避免了人为操作失误带来的损失。有敏感数据交易的系统中,监管不可缺少。为避免监管机制成为该系统的性能瓶颈,应适当多设置几个管理员节点,尽力对每个事务做到立即响应,减少排队等待的情况发生。

系统业务模型层的设计避免了电力数据被滥用和随意复制的情况。在业务逻辑设计中,拥有电力数据资源的用户 A 可以将数据内容或数据访问权资产化,记为 S,并发布在链上,其他的购买用户可以支付相应的价格得到数据资源 S。若购买了数据资源 S 的用户想要二次转卖给用户,则链上的事务逻辑会根据售卖数据的用户是否为数据的原始发布者 A 来决定对数据资源 S

是进行复制操作还是转让操作。具体来说,如果用户 B 向数据资源 S 的原始发布者用户 A 购买数据资源 S,则数据资源 S 会被复制一份 S' 卖给用户 B,此时用户 B 拥有数据资源 S';如果用户 C 向用户 B (数据资源 S 的非原始发布者) 购买数据资源 S',则 S' 由用户 B 转让给当前购买用户 C,而用户 B 不再拥有数据资源 S'。倘若用户 B 出于某种目的留存数据资源 S' 并将其重复发布,系统的监管机制会驳回这一事务;若侥幸通过了监管机制,原始发布者 A 也可以以链上发布记录的时间戳为证据,“举报”用户 B 的这一行为。这种机制改善了传统数据管理系统中对数据缺少确权管理的情况,不会因为数字化数据的可复制性而造成交易过后数据泛滥不可控。图 2 所示为电力数据资源流转示意图。

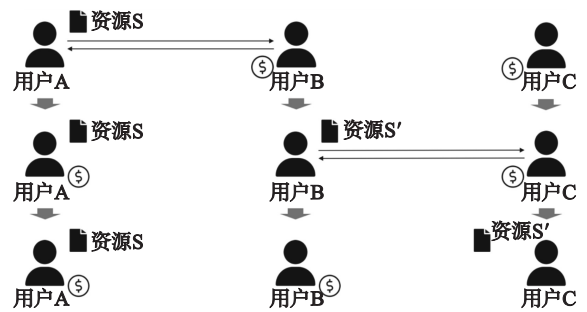


图 2 电力数据资源流转示意图

Fig. 2 Scheme of power data flow

链码是超级账本中智能合约的实现方式,用于实现系统业务的主要逻辑。链码被部署在 Hyperledger Fabric 的网络节点上,在 Docker 容器中运行,并通过 gRPC 协议与相应的 Peer 节点进行数据交互。当链码在网络节点上经过初始化、实例化和安装后,用户即可以通过 Hyperledger Fabric 中自带的客户端 (该客户端方便开发人员调试使用,因此还需上层可视化的 Web 客户端) 利用系统业务模型提供的 API 调用链码。

系统通过区块链的技术性质保障了电力数据的安全和交易记录的透明、可追溯、不可篡改。系统运行中,每一次的数据资源发布、交易等事务都会得到背书节点的背书、排序节点的共识和记账节点的验证与记录,在区块链的数据结构中形成了全局共识认证的持久化数据。因此系统对电力数据的所有相关操作都是透明、可追溯、不可篡改的,保障了电力数据的安全性。

以普通用户 X 发布电力数据资源 S 为例,介绍该系统中的事务流程:

1) 普通用户 X 登录普通用户客户端,将“普通用户 X 发布电力数据资源 S”这一事务发送至

管理员 A 的客户端;

2) 管理员 A 经过审查(用户认证、权限检查、内容审核)通过,创建一个交易提案给网络中的各个背书节点;

3) 背书节点模拟执行交易提案并签名,再将模拟执行的结果返回给管理员 A 的客户端;

4) 管理员 A 的客户端收集到足够的背书支持后,将模拟交易执行结果和背书节点签名提交

到排序节点;

5) 排序节点对交易进行排序并封装进区块,然后广播至所有主节点;

6) 主节点将区块同步至所有记账节点;

7) 记账节点对区块进行检查并保存区块,更新世界状态。

系统中其他事务类型的流程与上述并无差别,不再赘述。图 3 为具体事务流程图。

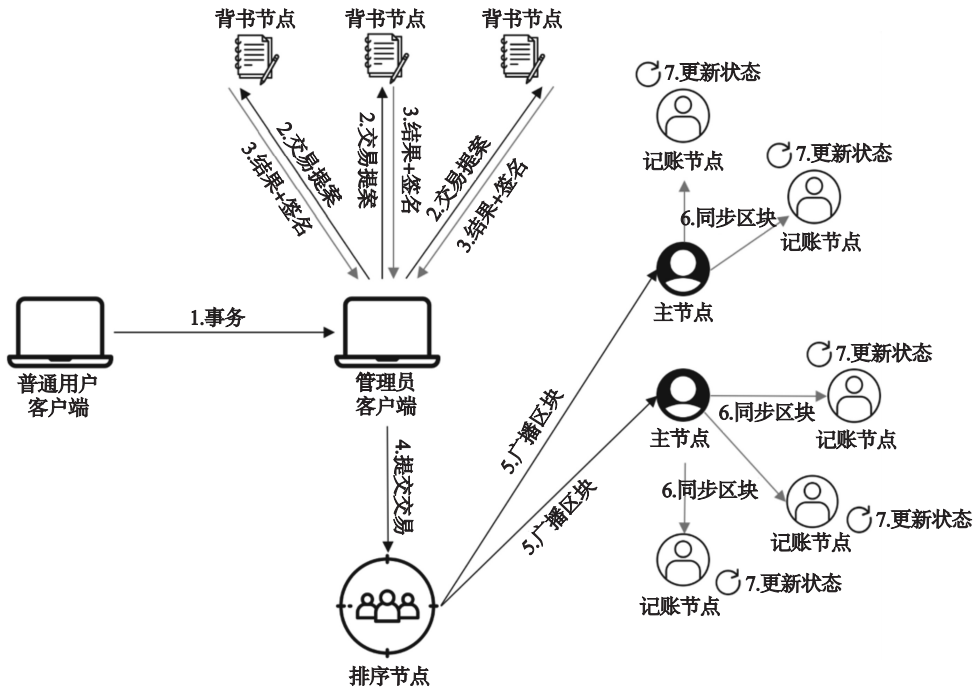


图 3 具体事务流程

Fig. 3 Specific transaction flow

相较于传统的数据资产交易系统,本文提出的基于区块链的电力数据资产化及交易系统有以下优点:

1) 电力数据持有者可以自行定价发布数据资产,数据资产可以从发布者处购买,也可以来自于其他购买者的转让(而非复制).这个过程是通过区块链上的链码逻辑实现的,可以避免数据被随意复制与滥用。

2) 引入管理员审查机制,加强了区块链系统的可监管性,为敏感的数据交易提供了一层保护。

3) 电力数据的内容或访问权和交易记录存储在区块链上,由多方实体共同维护,保障了电力数据的安全性和数据流转的透明、可追溯、不可篡改。

## 2.3 系统实现

### 2.3.1 编写链码实现业务逻辑

对应系统需求中的 9 项具体功能,首先建立以下 3 个业务实体:

1) 用户:  $User = \{Name, Id, Assets, Issues,$

$UnIssues, Token\}$ . 其中:  $Name$  表示用户名称;  $Id$  表示用户编号;  $Assets$  是一个列表,用于存储用户购买的资源;  $Issues$  是一个列表,用于存储用户发布的资源;  $UnIssues$  是一个列表,用于存储用户取消发布的资源;  $Token$  表示用户余额。

2) 资源:  $Asset = \{Name, Id, Metadata, IsValid, Value\}$ . 其中:  $Name$  表示资源名称;  $Id$  表示资源编号;  $Metadata$  用于抽象地表示电力数据的内容本身或访问权;  $IsValid$  表示该资源是否还在售卖,若发布者取消该资源的发布,则值为  $false$ ,其他用户不可以再购买该资源,但已购买的用户可以继续持有;  $Value$  表示资源价格。

3) 资源变更记录:  $AssetHistory = \{AssetId, OriginalOwnerId, CurrentOwnerId\}$ . 其中:  $AssetId$  表示资源编号;  $OriginalOwnerId$  表示资源的前一位持有人,若是首次发布,则值为  $firstIssueTag$ ;  $CurrentOwnerId$  表示资源的当前持有人。

基于这 3 个业务实体,针对 9 项具体功能设计了如下 9 种交互方法:

1) 用户开户: `function userRegister (userName, userId)`, 其中 2 个参数分别是用户名称、用户编号。

2) 用户充值: `function userTopup (userId, Token)`, 其中 2 个参数分别是用户编号、充值金额。

3) 用户销户: `function deleteUser (userId)`, 删除用户, 并将该用户所有发布中的资源取消发布。其中的参数是用户编号。

4) 资源发布: `function assetsEnroll (assetName, assetId, metadata, ownerId, value)`, 将资源加入发布者的发布资源列表, 添加一条该资源的历史记录。其中 5 个参数分别是资源名称、资源编号、资源内容本身或访问权、发布者编号、资源价格。

5) 资源转让: `function assetsExchange (assetId, originOwnerId, currentOwnerId)`, 若资源在源用户的发布列表里, 则在受让用户的购买列表里添加资源, 源用户的余额加上资源价格, 受让用户的余额减去资源价格; 若资源在源用户的购买列表里, 则在源用户的购买列表里删除资源, 在受让用户的购买列表里添加资源, 源用户的余额加上资源价格, 受让用户的余额减去资源价格。其中 3 个参数分别是资源编号、源用户编号、受让用户编号。通过这种方式使资源从发布者处购买和从其他购买者处购买区分开来, 从非资源发布者处购买时是真正的转让数据, 规避了数据资产被随意复制导致滥用的风险。

6) 资源取消发布: `function assetsCancel (assetId, issuerId)`, 将资源从发布列表中删除, 添加到取消发布列表中, 改变资源的有效状态为 `false`。其中 2 个参数分别是资源编号、发布者编号。

7) 查询用户: `function queryUser (userId)`, 其中的参数是用户编号。

8) 查询资源: `function queryAsset (assetId)`, 其中的参数是资源编号。

9) 查询资源历史记录: `function assetsExchangeHistory (assetId, queryType)`, 查询资源的所有变更历史记录, 通过指定查询类型 `queryType` 选择性地查询登记信息、转让信息, 或者全部信息。其中 2 个参数分别是资源编号、查询类型。

### 2.3.2 搭建区块链网络

配置 Hyperledger Fabric 区块链网络, 需要每个网络节点进行本地部署。Hyperledger Fabric 依

赖 `docker` 和 `docker-compose` 环境并需下载对应的镜像。

编译 `Cryptogen` (生成证书文件程序), 并按照预设的 `crypto.config.yaml` 文件中的联盟链组织关系生成各节点的公钥、私钥和证书。对 `Configtxgen` 进行编译, 然后通过它生成创世区块、通道配置和通道主节点间的交易。在 `docker-compose.yaml` 中配置网络节点后, 就可以通过 `docker-compose` 启动网络。在经过启动客户端、创建通道、加入通道、设置所在组织主节点的步骤后, 区块链的网络搭建成功。

链码编写完成后进入客户端, 安装链码并在通道中实例化, 此时就已经将链码安装到了已创建的通道上。电力实体间的电力数据发布及交易等记录都会被节点打包成区块记录在链上, 上链后的记录都无法被篡改, 便于追溯数据流向。

Hyperledger Fabric 中提供了 `fabric-sdk-go` 等 SDK, 供开发人员使用 API 与区块链服务通信。本文选择 Go 语言搭建 Web 服务层, 具体选用的是 `Gin-gonic` 微服务框架。Hyperledger Fabric 的 `fabric-sdk-go` 接口主要提供 4 种与区块链通信的方法: 区块链管理、数据查询、区块链交互、事件监听。其中, 区块链交互接口是本系统的开发重点, Web 后端应将各种前端提交的请求通过 SDK 与区块链服务连接起来。表 1 是 `Gin` 微服务框架下的 Web 层主函数代码。

表 1 Web 层主函数

Table 1 Web layer's main function

```
func main() {
    router := gin.Default()
    router.LoadHTMLGlob("./html/*")
    // 定义路由
    {
        router.GET("/index", console)
        router.POST("/users/register", userRegister)
        router.GET("/users/get", queryUser)
        router.POST("/users/topup", userTopup)
        router.POST("/users/delete", deleteUser)
        router.GET("/assets/get", queryAsset)
        router.POST("/assets/enroll", assetsEnroll)
        router.POST("/assets/exchange",
            assetsExchange)
        router.POST("/assets/cancel", assetsCancel)
        router.GET("/assets/exchange/history",
            assetsExchangeHistory)
    }
    router.Run() // 服务与监听于 0.0.0.0:8080
}
```

路由定义很方便, 每一条用户请求对应系统中的访问主界面和其他系统功能。console,

userRegister, queryUser, userTopup, deleteUser, queryAsset, assetsEnroll, assetsExchange, assetsCancel, assetsExchangeHistory 是具体的功能函数名称,具体代码限于篇幅不再展开。

### 3 系统测试及分析

本地测试的开发环境是:操作系统 macOS 10.14.6,六核 CPU 3.0 GHz, RAM 16 GB. 单机配置多个区块链网络节点、一个排序节点 Orderer,以及两个组织 Org0 和 Org1,每个组织中有两个节点 Peer0 和 Peer1,采用 Solo 共识机制。

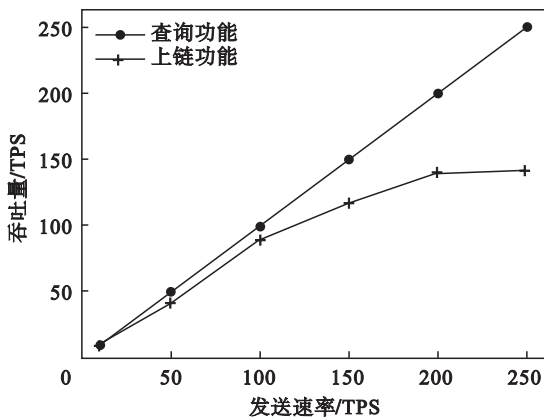
首先对本文设计的区块链网络的底层架构进行测试:

1) 测试区块链网络是否可以成功运行,即基本区块链功能(启动节点、生成区块、建立通道、安装链码等)的可用性;

2) 测试链码逻辑的正确性。

在生成证书文件、创建创世区块、生成通道的创世交易、生成组织关于通道主节点的交易后,启动网络并安装链码。

随后,测试各项功能,验证区块链的功能与链码逻辑的正确。确认底层的区块链网络和链码逻辑可以正常运行后,对系统前后端级联进行测试。



该部分测试一是为了测试 Go Web 部分的运行状态(前端是否能够向后端正常发送请求,是否能够与区块链正常级联以获取正确的响应);二是为了得到系统的整体运行效果,即支撑业务需求的完备性。同样在 docker 启动区块链服务,并安装实例化链码。编译运行网络服务,在本地 127.0.0.1:8080 端口运行。在浏览器中打开 127.0.0.1:8080/index,测试表明每项功能均能正常运行。

系统性能测试,采用 Hyperledger Caliper 性能测试框架,测试区块链网络在交易总数为 2000 笔时,上链功能(用户开户、用户充值、用户销户、资源发布、资源转让、资源取消发布)和查询功能(查询用户、查询资源、查询资源历史记录)的系统吞吐量和平均时延。性能测试结果如图 4 所示,可以看出,查询功能的吞吐量在测试范围内约等于发送速率,平均时延保持较小的数值,这是由于查询功能无需经过共识改变链上状态;而上链功能则随着发送速率的增加,吞吐量达到峰值(约为 140 TPS(transactions per second)后保持稳定,且平均时延相比于查询功能的平均时延增加得更快,这是由于上链功能涉及到交易的模拟执行、背书、排序、同步的过程。

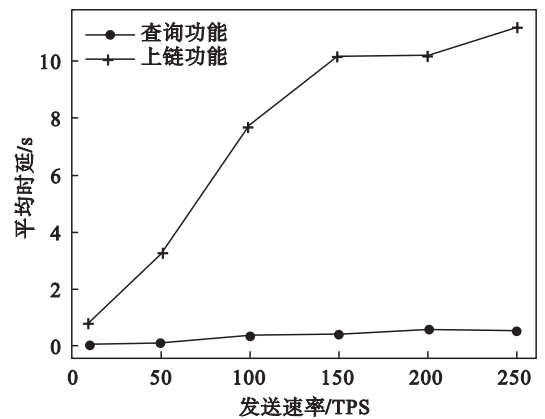


图 4 性能测试结果

Fig. 4 Performance test results

综上,系统的功能运行正常,满足电力数据资产化及交易需要的安全性、透明性、可追溯性、不可篡改性及可监管性的功能需求;系统的性能也可以满足正常的使用需求。通过 Web 服务层的封装,结果展示也更加直观易读。目前系统处于开发测试阶段,单个节点的部署步骤还较为繁琐;若投入企业级的应用,可以通过容器虚拟化技术(如 docker)加以改进,将节点部署所需的程序、配置文件和依赖项一起打包为单个容器镜像,利用容

器镜像可重用性提升节点的部署管理效率。

## 4 结 语

本文针对电力数据的特点,分析其资产化及交易平台的需求,提出了基于区块链中联盟链技术的解决方案。设计的基于区块链的电力数据资产化及交易系统适用于敏感数据的交易,实现了用户发布资源、交易资源、查询记录等 9 项基本

功能. 相较于传统的数据交易系统, 本系统能够保证用户、资源及历史记录信息的不可篡改, 保障了交易的安全和可溯源, 另外还加入了可监管机制. 本系统基于 Hyperledger Fabric 联盟链框架开发, 可以自行设计联盟间的组织结构, 通过编写链码 (智能合约) 实现对应的业务逻辑, 并提供了 Web 服务便于用户使用. 系统尚有不完善之处, 若在未来投入企业级的应用, 可用容器虚拟化技术将节点部署过程简化, 提高效率.

### 参考文献:

- [1] Rusitschka S, Eger K, Gerdes C. Smart grid data cloud: a model for utilizing cloud computing in the smart grid domain [C] // 2010 First IEEE International Conference on Smart Grid Communications. Gaithersburg, 2010: 483 - 488.
- [2] Bobba R, Khurana H, Alturki M, et al. PBES: a policy based encryption system with application to data sharing in the power grid [C] // ASIACCS'09: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security. Sydney, 2009: 262 - 275.
- [3] Pan Y, Zhang X, Wang Y, et al. Application of blockchain in carbon trading [J]. *Energy Procedia*, 2019, 158: 4286 - 4291.
- [4] Mehta R, Kapoor N, Sourav S, et al. Decentralised image sharing and copyright protection using blockchain and perceptual hashes [C] // 2019 11th International Conference on Communication Systems & Networks (COMSNETS). Bengaluru, 2019: 1 - 6.
- [5] Casino F, Kanakaris V, Dasaklis T K, et al. Modeling food supply chain traceability based on blockchain technology [J]. *International Federation of Automatic Control*, 2019, 52 (13): 2728 - 2733.
- [6] Liang W, Tang M, Long J, et al. A secure fabric blockchain-based data transmission technique for industrial Internet-of-things [J]. *IEEE Transactions on Industrial Informatics*, 2019, 15 (6): 3582 - 3592.
- [7] 李旭, 王合建. 基于区块链的电力数据共享机制 [C] // 生态互联数字电力——2019 电力行业信息化年会论文集. 无锡, 2019: 188 - 191.  
(Li Xu, Wang He-jian. The sharing mechanism of electric power data based on blockchain [C] // Eco-connected Digital Power: Proceedings of the Electric Power Industrial Informatization Annual Meeting. Wuxi, 2019: 188 - 191.)
- [8] Androulaki E, Barger A, Bortnikov V, et al. Hyperledger Fabric: a distributed operating system for permissioned blockchains [C/OL] // Proceedings of the Thirteenth EuroSys Conference. Porto, 2018: 1 - 15 (2018 - 04 - 17) [2020 - 04 - 25]. <https://arxiv.org/pdf/1801.10228.pdf>. DOI: 10.1145/3190508.3190538.
- [9] Cachin C. Architecture of the hyperledger blockchain fabric [C/OL] // Workshop on Distributed Cryptocurrencies and Consensus Ledgers. Chicago, 2016: 1 - 4 [2020 - 05 - 16]. <https://www.innovation4.cn/library/r44511>.
- [10] Nakamoto S. Bitcoin: a peer-to-peer electronic cash system [EB/OL]. (2008 - 10 - 31) [2020 - 05 - 05]. <https://bitcoin.org/bitcoin.pdf>.
- [11] Antonopoulos A M. Mastering Bitcoin: unlocking digital cryptocurrencies [M]. Sebastopol: O'Reilly Media, 2017.
- [12] 孙毅, 范灵俊, 洪学海. 区块链技术发展及应用: 现状与挑战 [J]. *中国工程科学*, 2018, 20 (2): 27 - 32.  
(Sun Yi, Fan Ling-jun, Hong Xue-hai. Technology development and application of blockchain: current status and challenges [J]. *Strategic Study of CAE*, 2018, 20 (2): 27 - 32.)
- [13] Swan M. Blockchain: blueprint for a new economy [M]. Sebastopol: O'Reilly Media, 2015.
- [14] Kreps J, Narkhede N, Rao J. Kafka: a distributed messaging system for log processing [C] // The 6th International Workshop on Networking Meets Databases (NetDB 2011). Athens, 2011: 1 - 7.
- [15] Demers A, Greene D, Hauser C, et al. Epidemic algorithms for replicated database maintenance [C] // Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing. Vancouver, 1987: 1 - 12.