计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.201200170

基于区块链的企业联盟共享数字积分管理机制

凌 飞 陈世平
上海理工大学光电信息与计算机工程学院 上海 200093
(183800783@st.usst.edu.cn)

摘 要 传统的数字积分管理机制都具有分散管理、使用限制多、兑换流通不方便等问题,限制了积分的使用。针对传统数字积分机制存在的问题,提出企业联盟共享数字积分管理机制,利用区块链的去中心化机制、安全的身份认证管理、分布式数据库和智能合约自动化处理等优势,设计基于区块链的企业联盟共享数字积分管理机制,结合企业数字积分交易的应用场景,对区块链网络吞吐量和延迟等性能进行测试与研究,通过仿真实验验证企业联盟共享数字积分管理机制的可行性,为区块链企业级应用提供参考。

关键词:共享数字积分;区块链;分布式数据库;智能合约
中图法分类号 TP311

Shared Digital Credits Management Mechanism of Enterprise Alliance Based on Blockchain

LING Fei and CHEN Shi-ping
College of Optical-Electrical & Computer Engineering, University of Shanghai for Science & Technology, Shanghai 200093, China

Abstract Traditional digital credits management mechanisms have decentralized, utilization restrictions, inconvenient exchange circulation and other problems, which limit the use of credits. Aiming at the existing problems of traditional digital credits mechanism, this paper puts forward the management mechanism of enterprise alliance sharing digital credits. Based on the decentralized mechanism of blockchain, secure identity authentication management, distributed database and smart contract automatic processing and other advantages, a shared digital credits management mechanism of enterprise alliance based on blockchain is designed. Combined with the application scenario of enterprise digital credits transaction, the performance of block chain network throughput and delay is tested and studied. The feasibility of the shared digital credits management mechanism of enterprise alliance is verified through simulation experiment, which provides reference for the enterprise-level application of block chain.

Keywords Shared digital credits, Blockchain, Distributed database, Smart contract

1 引言

积分是各行各业普遍使用的一种营销方式,积分原先的设计思路是提高复购率,检验用户的忠诚度,能够反复使用。商城积分、银行积分等各种传统的积分模式下,积分系统为各个组织单独设计拥有,积分兑换繁琐、流通差,给客户带来了不好的用户体验;积分无法转让、赠送,导致积分发行商家的品牌传播有限。积分最大的问题就是很难流通,因为大量的积分是用户沉淀未使用的,最后造成积分对于用户可有可无,这是积分的一个困境[1]。当前部分企业组织个体与个体之间在策略目标的考虑下结成盟友,自主地进行互补性资源交换,各自达成目标产品阶段性的目标,最后获得长期的市场竞争优势,并形成一个持续而正式的企业联盟关系。

区块链技术设计的可转增的积分能够作为一个数字资产。这个积分可以打通多个商家,只要在同一个平台上入驻的商家都可以用积分去兑换商品。最终用户对积分有使用所有权,用户可以转赠积分,这样积分就变得灵活具有多样性。基于区块链的企业联盟共享数字积分保证用户有权自由支配自己的积分,最主要的是区块链解决了积分的利用率无法传播这一特点。积分是用户所有,用户来实现积分的价值[2]。

区块链是由众多节点共同组成的一个端到端的网络,不存在中心化的设备和管理机构。区块链数据的验证、记账、存储、维护和传输都不是基于中心机构,而是利用数学算法实现。去中心化使网络中的各节点之间能够自由连接,同时区块链中每一个区块都是与前续区块通过密码学证明的方式链接在一起的,当区块链达到一定的长度后,要修改某个历史区块中的交易内容就必须将该区块之前的所有区块的交易记录及密码学证明进行重构,有效实现了防篡改,保障了共享数字积分管理的可靠性和安全性。

从目前关于共享数字积分的区块链研究分析来看,Wang等基于区块链的商城积分系统方法进行研究[3],将区块链技术作为底层技术应用于商城的积分系统中,提出了基于区块链的商城积分系统架构,包含积分发行、积分升级、积分兑换、积分消费、积分互换、积分赠送和积分查询等内容,但只是从商城角度提出的积分系统研究方法适用的场景不够全面;Zhang提出了基于区块链的通用积分管理系统[4],使用了

基金项目:国家自然科学基金资助项目(61472256)
This work was supported by the National Natural Science Foundation of China(61472256).
通信作者:陈世平(chensp@usst.edu.cn)

Fabric 等技术开发设计，但未对区块链性能安全等方面进行测试与研究，随着区块链技术的不断发展，其存在有限的事务吞吐量和较高的响应延迟，特别是与传统的分布式数据库系统相比，可支持的并发事务处理量不高。

本文主要阐述了基于区块链的企业联盟共享数字积分机制，同时介绍了针对 Fabric 性能的研究，通过仿真实验分析了共享数字积分机制的可行性[5]。实验表明区块链共享数字积分机制不仅满足现实中的需求，保证了企业联盟的积分数据安全共享，在性能方面也达到了实际生产的要求。

2 区块链技术

2.1 概要

区块链是一个由分布式网络中的节点维护的不可篡改的账本[6]。这些节点通过执行被共识协议验证过的交易来各自维护一个账本的副本，账本以区块的形式存在，每个区块通过哈希值和前面的区块相连。第一个被广为人知的区块链应用是加密货币比特币[7]。图 1 为简化的比特币区块链示意图。

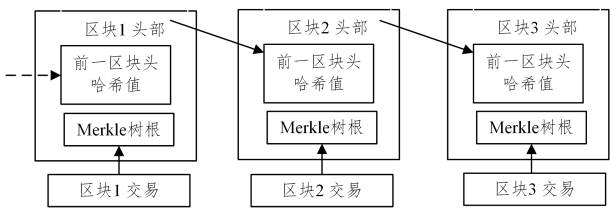


图 1 简化的比特币区块链示意图

Fig. 1 Simplified diagram of the Bitcoin blockchain

以太坊是另一种区块链加密货币技术，整合了许多类似比特币的特征，但是新增的智能合约为去中心化分布式应用创建了一个平台[8]。比特币和以太坊属于同一类区块链，将其归类为公共非许可区块链技术，其基本上都是公共网络，允许任何人在上面匿名互动。

随着比特币、以太坊和其他衍生技术的普及，越来越多的人想要将区块链基础技术、分布式账本和分布式应用平台用到企业业务中去[9]，这意味着区块链平台必须具备企业级属性。具体说来，对安全性的考虑会更为突出，在很多企业级应用场景需要有授权才能访问区块链，也就是权限控制链，一般来说企业级的区块链的部署模式是联盟链或私有链；同时对网络和共识算法的性能、每秒交易数都有比较高的要求。

2.2 Fabric 技术特点

Fabric 项目的目标是实现一个通用的权限区块链的底层基础框架。为了适用于不同的场合，其采用模块化架构，提供可插拔和可扩展的组件[10]，包括对交易顺序建立共识的排序共识算法、负责将网络中的实体与加密身份相关联的成员服务提供者 MSP、可选的 P2P gossip 通信服务协议、账本支持的多种 DBMS 和可插拔的背书和验证策略等。

2.2.1 节点类型

客户端发起读写事务请求到区块链中，该事务发送到区块链节点上。节点分为普通对等节点 peers 和排序节点 orderers，其中普通对等节点分为背书节点 endorers 和非背书节点。背书节点根据背书策略来接收客户端请求并处理，并将结果返回给客户端。客户端将收集背书节点的返回数据并发送给排序节点。排序节点组成排序服务，通过共识机制负责决定交易的顺序，并将交易打包成块，广播给对等节点。所有对等节点都将先校验区块并提交到区块链的本地副本，最终将相应的更改应用到维护当前世界状态快照的状态数据库[11]。

区块链网络中的不同参与者包括 Peer 节点、排序节点、客户端应用程序、管理员等。每一个参与者(网络内部或外部能够使用服务的活动元素)都具有封装在 X.509 数字证书中的数字身份[12]。这些身份确定了对资源的确切权限以及对参与者在区块链网络中拥有的信息的访问权限。

2.2.2 gossip 数据传输协议

Fabric 将工作负载拆分为交易执行节点和排序节点，优化了区块链网络的性能、安全性和可扩展性。这样同时需要一个安全、可靠和可扩展的数据传播协议来保证数据的完整性和一致性。为了满足该要求，Fabric 实现了 gossip 数据传播协议。

peer 节点通过 gossip 协议来传播账本和通道数据，gossip 消息是持续的，通道中的每一个 peer 节点不断地从多个节点接收当前一致的账本数据[13]。每一个 gossip 消息都是带有签名的，因此拜占庭成员发送的伪造消息很容易被识别，并且非目标节点也不会接收与其无关的消息。peer 节点会受到延迟、网络分区或者其他原因的影响而丢失区块，这时节点会从其他拥有这些丢失区块的节点处同步账本。

基于 gossip 的数据传播协议在 Fabric 网络中有 3 个主要功能：

(1)通过持续地识别可用的成员节点来管理节点和通道成员，并检测离线节点。

(2)向通道中的所有节点传播账本数据。所有没有与当前通道的数据同步的节点会识别丢失的区块，并将正确的数据复制过来以完成同步。

(3)通过点对点的数据传输方式，使新节点以最快速度连接到网络中并同步账本数据。

peer 节点基于 gossip 的数据广播操作接收通道中其他的节点的信息，然后将这些信息随机发送给通道上的一些其他节点，随机发送的节点数量是一个可配置的常量。peer 节点可以用“拉”的方式获取信息而不用一直等待。这是一个重复的过程，以使通道中的成员、账本和状态信息同步并保持最新。在分发新区块的时候，通道中主节点从排序服务拉取数据然后分发给它所在组织的节点[14]。

在线的节点通过持续广播“存活”消息来表明其处于可用状态，每一条消息都包含了“公钥基础设施(PKI)”ID 和发送者的签名。

节点通过收集这些存活的消息来维护通道成员。如果没有节点收到某个节点的存活信息，这个“死亡”的节点会从通道成员关系中被剔除。因为“存活”的消息是经过签名的，恶意节点没有根 CA 签发的密钥，因而无法假冒其他节点。

除了自动转发接收到的消息之外，状态协调进程还会在每个通道上的 peer 节点之间同步世界状态。每个 peer 节点都持续从通道中的其他节点拉取区块，来修复他们缺失的状态。因为基于 gossip 的数据分发不需要固定的连接，所以该过程可以可靠地提供共享账本的一致性和完整性，包括对节点崩溃的容忍。

2.2.3 MSP 成员服务提供者

要使身份可以被验证，它必须来自可信任的权威机构。

成员服务提供者(Membership Service Provider，MSP)是 Fabric 中可以信任的权威机构。具体地说，一个 MSP 是定义管理该组织有效身份规则的组件[15]。Fabric 中默认的 MSP 实现使用 X.509 证书作为身份，采用传统的公钥基础结构分层模型。

X.509 身份认证过程如图 2 所示，发送方首先将相关身份信息通过 ecdsa sha256 算法生成 Hash 值，然后将该 Hash 值通过 CA 私钥生成数字签名，接收方使用 CA 公钥进行解密，产生另一个 Hash 值，通过与之前的 Hash 值比较来判断用户的身份信息。

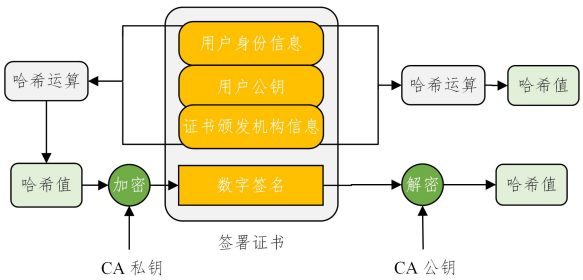


图 2　身份认证

Fig. 2　Identity authentication

2.2.4　事务流程

针对交易事务流，Fabric 引入新的架构(执行-排序-验证-提交)，解决了排序-执行-提交模型面临的弹性、灵活性、可伸缩性和机密性问题[16]，图 3 给出了交易的具体流程。

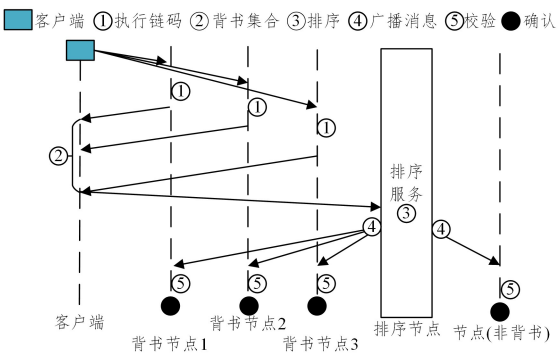


图 3　Fabric 事务流

Fig. 3　Fabric transaction flow

在第一阶段，客户端将交易请求发送给背书节点 endorsers；第二阶段每个背书节点在沙箱中执行交易，并计算相应的读写集，每个背书节点还使用业务逻辑来验证交易的正确性；第三阶段客户端等待足够数量的背书节点的响应，然后将这些响应发送给实现排序服务的排序节点 orderers；第四阶段排序节点首先对传入交易的顺序达成一致，然后将消息队列分割成块，再将块广播给 peers 节点；第五阶段由 peers 节点验证区块并将其提交到区块链中。

特定应用程序的背书策略可以指定需要哪些节点或多少节点来保证给定的智能合约正确执行。因此，每个交易只需要由满足交易的背书策略所必需的节点来执行(背书)。这样可以并行执行，从而提高系统的整体性能和规模。

排序服务为多个节点组成的集群，主要功能分为 3 部分。

(1)对交易事务的排序，通过可插拔式的共识机制，提供了排序服务的容错性，保证了区块链交易的顺序性，使各节点的账本达成共识。

(2)对交易的分块打包，排序服务分块规则默认由初始配置决定：

1)块的最大交易数量；

2)块的最大的 bytes 容量；

3)生成块的时间，即从该块的第一个交易到达时间后最迟多久生成区块。

(3)对块进行签名并广播给对等节点。

对等节点接收到数据区块后，首先会对广播区块的排序节点的签名进行验证，保证区块的合法性；然后会对区块中每笔交易按顺序执行 VSCC(validation System Chaincode)背书校验，即根据背书策略，校验交易的背书签名集合，若不满足背书策略则会将交易设置为无效交易；最后会进行 MVCC(Multiple Version Concurrency Control)多本版并发控制校验[17]。在事务流第二阶段时，背书节点执行完交易会保存读写集数据，其中读集合中包含所读数据的版本号，在进行提交数据前的验证阶段，需确保交易在背书阶段读取的版本号与提交时它们在本地账中的当前版本状态相同。这样避免了并发访问修改等带来的数据不一致问题。

3　基于区块链的企业联盟共享数字积分管理机制

本节将会描述一个涉及多个组织的业务场景，这些组织使用基于 Fabric 构建的企业联盟共享数字积分平台来发行积分，根据不同的发行组织，积分会有不同的兑换比例，即会区分不同组织的积分；客户可以在平台内自由转让、兑换和消费数字积分。

3.1　共享数字积分网络结构

Fabric 是一个许可平台，通过其通道架构和私有数据特性实现保密。在通道方面，Fabric 网络中的成员组建了一个子网络，在子网络中的成员可以看到其所参与到的交易。因此，参与到通道的节点才有权访问智能合约(链码)和交易数据，以此保证了隐私性和保密性。私有数据通过在通道中的成员间使用集合，实现了与通道相同的隐私能力并且不用创建和维护独立的通道。

在创建共享数字积分网络时，首先需要定义一个排序服务，排序服务通常是多节点的，可配置在不同组织的不同节点上。该网络配置由各组织统一管理。接下来需要定义一个联盟，该联盟的定义存储在网络配置中，联盟共享彼此能够交易的需求。联盟间重要的一个部分就是通道，通道是一个联盟彼此进行通信的主要机制，使用通道将区块链网络以及企业的节点关联到一起，节点是存储区块链副本的网络组件。组织需将智能合约安装到节点上，通过智能合约访问区块链账本。

共享积分网络具体示例结构如图 4 所示，假设网络中存在 3 个组织，组成共享数字积分联盟，在区块链网络中每个组织都会使用成员服务提供者 MSP 来配置身份信息、控制和验证操作权限。每个组织对应一个节点，客户端通过通道访问对应节点的智能合约来与区块链账本通信，同一通道中的每个节点的智能合约和账本是相同的，即一个通道定义了一个联盟交互方式和数据结构，外部的节点无法访问到该通道上的数据，保证了联盟共享数据的安全性。同时需将排序服务连接到该通道中，为用户请求的交易进行排序来保证账本的一致性。

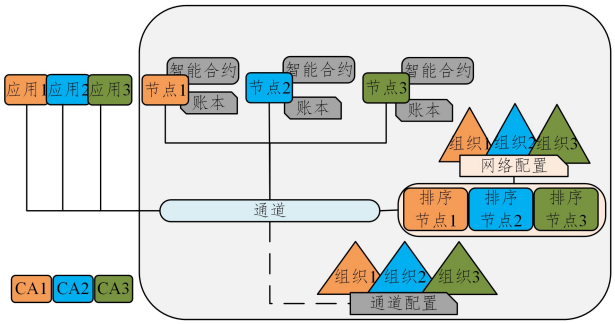


图 4 共享积分网络结构

Fig. 4 Sharing credits network structure

3.2 共享积分数据结构

本节主要讨论在企业联盟共享数字积分网络中如何设计共享数字积分的生命周期和与它相关的数据结构，以帮助随后智能合约的设计。

为满足实际的需求，从企业组织和用户两方面来讨论。

(1)组织可以向用户发行数字积分，组织可查询所发行积分的用户信息；用户的积分数据是属于该联盟的共享数据，每个组织都可以查询每个用户的积分数值，确认用户积分的发行出处；由于各个组织间的用户数据都是共享透明的，防止了组织间的相互欺骗行为。

(2)客户可以将不同组织的积分相互兑换，由于各个组织间的差异，所发行的积分价值会有所不同，可以将它们按一定实际比例统一兑换为一个超级积分，该超级积分可以理解为各个组织间公认的积分价值尺度，由联盟共同协定，该流程是由智能合约自动执行的过程，保证了组织不能擅自修改自己所发行积分的价值；用户可转让自己的积分给其他用户，同时可以消费自己的积分。

如表 1 和表 2 所列，组织发行积分和用户积分的结构包含但不限于以下几个属性。

表 1 共享数据积分发行

Table 1 The issuance of shared data credits

属性名	说明
Issuer	积分发行商
Owner	积分拥有者
Value	发行数值

表 2 用户超级积分

Table 2 User super credits

属性名	说明
Owner	积分拥有者
Token	超级积分数值

其中，用户超级积分数值由组织发行的积分值 Value，根据其价值比例计算得出。每当有另一个组织向该用户发行积分时，再次计算并与当前用户的 Token 相加，得出目前用户的超级积分数值。

同时用户每一笔的转让积分和消费积分的交易也会被保存到区块链账本中，具体结构如表 3 和表 4 所列。

表 3 积分转让

Table 3 Credits transfer

属性名	说明
Owner_out	积分转出者
Owner_in	积分转入者
Token_out	超级积分转让数值

表 4 积分消费

Table 4 Credits consume

属性名	说明
Owner	积分消费者
Token_ consume	超级积分消费数值

如图 5 所示，假设 Org_1 向 $Client_1$ 发行了 Value 为 50 的积分，Org_2 向 $Client_1$ 发行了 Value 为 20 的积分，Org_1 的积分价值与超级积分的兑换比例为1∶2，Org_2 的积分价值与超级积分的比例为 1∶1，此时 $Client_1$ 所持有的超级积分 Token 应为 120，$Client_1$ 又向 $Client_2$ 转让了 20 个超级积分，则最终 $Client_1$ 剩余的 Token 为 100。

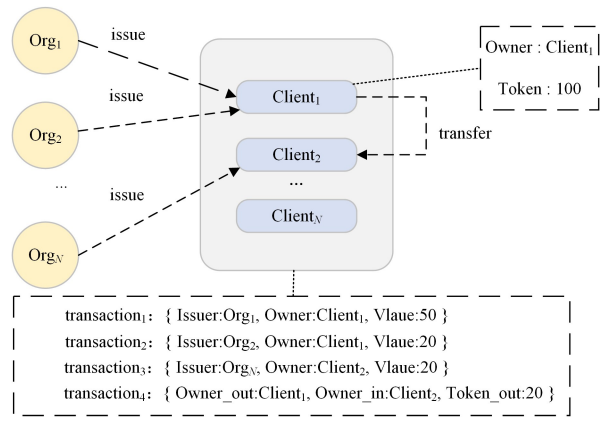


图 5 积分交易转移图

Fig. 5 Credits transaction transfer diagram

根据不同交易的属性，组织向用户发行积分、用户转让积分和用户消费积分的交易都将保存到区块链账本的交易集合列表中。由于组织在发行积分时都必须携带 MSP 管理的身份信息，并会校验组织身份的有效性，则可以通过查询检验组织的发行积分交易，确保企业之间用户积分数据的真实性、可靠性。

3.3 智能合约处理

区块链网络的核心是智能合约。在共享数字积分网络中，智能合约用可执行的代码定义了不同组织之间的规则。应用程序调用智能合约来生成被记录到账本上的交易。

根据共享数字积分的生命周期，智能合约规定了共享积分的处理逻辑，需要满足以下几个要求：

(1)调用智能合约前，判断调用者是否有权进行该操作，为此调用者需提供自己的身份标识发送给每个组织的背书节点进行身份的验证。

(2)当整个联盟验证完身份并同意该操作后，智能合约需要验证请求参数的有效性，只有满足条件的才可发起请求。在发行积分时，自动计算组织积分与超级积分的比例，转化到用户超级积分中。

(3)为保证组织对用户积分的每笔交易具有可追溯性，共享积分智能合约必须实现对指定用户每笔交易的查询，以保证用户积分来源的可靠性。

根据共享积分数据结构，使用 Go 语言编写的智能合约如图 6 所示，其分为发行交易、转让与消费积分和用户超级积分数据结构。

```
// Credits :  Define the Credits structure.
// Structure tags are used by encoding/json library
type Credits_issue struct {
    Issuer   string `json:"issuer"`
    Owner  string `json:"owner"`
    Value string `json:"value"`
}

type Credits_transfer struct {
    Owner_out   string `json:"owner_out"`
    Owner_in  string `json:"owner_in"`
    Token_out string `json:"token_out"`
}

type Credits_consume struct {
    Owner  string `json:"owner`
    Token_consume string `json:"token_consume"`
}

type creditsPrivateDetails struct {
    Owner string `json:"Owner"`
    Token string `json:"token"`
}
```

图 6　智能合约数据结构

Fig. 6　Smart contract data structure

根据实际处理流程，发行积分算法的步骤大致如算法 1 所示。

算法 1　发行积分

输入：(fcn，peers，chaincodeName，channelName，args)

输出：X

1. 根据 peers，确定需要进行背书的节点；
2. 根据通道名称 channelName、链码名称 chaincodeName 和方法名 fcn 定位到指定的发行积分方法；
3. 校验参数 args[Issuer，Owner，Value]；
4. 保存发行交易，根据发行组织和发行积分数值修改用户超级积分数值 Token；
5. 保存组合键 owner～key 和 issuer～key，该组合键可以被用来范围查询所有指定 Owner 的和所有指定发行商 Issuer 的发行交易集合；
6. 输出发行积分数据。

其大致代码如图 7 所示。

```
// issue credits to client
func (s *SmartContract) issueCredits(APIstub shim.ChaincodeStubInterface, args []string) sc.Response {
    //check args. Custom validations can be added
    if len(args) != 4 {
        return shim.Error("Incorrect number of arguments. Expecting 4")
    }
    // save credits issue transaction
    var credits_issue = Credits_issue{Issuer: args[1], Owner: args[2], Value: args[3]}
    issueAsBytes, _ := json.Marshal(credits_issue)
    APIstub.PutState(args[0], issueAsBytes)

    // save creditsPrivateDetails transaction
    var token_new =  getNewTokenByOwnerValue(args[2],args[3])
    var creditsPrivateDetails = creditsPrivateDetails{ Owner: args[2], Token: token_new}
    creditsPrivateDetailsAsBytes, _ := json.Marshal(creditsPrivateDetails)
    APIstub.PutState(args[2], creditsPrivateDetailsAsBytes)

    // create Composite Key of owner
    indexName_owner := "owner~key"
    colorNameIndexKey, err := APIstub.CreateCompositeKey(indexName_owner, []string{credits_issue.Owner, args[0]})
    if err != nil {
        return shim.Error(err.Error())
    }
    value := []byte{0x00}
    APIstub.PutState(colorNameIndexKey, value)

    // create Composite Key of issuer
    indexName_issuer := "issuer~key"
    colorNameIndexKey2, err := APIstub.CreateCompositeKey(indexName_issuer, []string{credits_issue.Issuer, args[0]})
    if err != nil {
        return shim.Error(err.Error())
    }
    APIstub.PutState(colorNameIndexKey2, value)

    return shim.Success(issueAsBytes)
}
```

图 7　智能合约积分发行

Fig. 7　Smart contract of issue credits

3.4　性能分析

区块链技术近年来受到了广泛的关注，但在性能和扩展性方面仍存在较大的技术挑战。随着网络使用量的稳步增长，为了在实践中可行，区块链必须支持与现有数据库管理所支持的相当的事务速率，超级账本结构的性能成为企业关注的一个重要问题。如共享积分管理，需要以高速的速度来记录和读取大量的用户数据，因此，有必要提高区块链的性能，以支持持续的增长。

近年来许多研究提出了各种优化来提高区块链的性能，从事务流程图可以看出，将共识节点和账本节点分离开来，有效地将工作负载分离，优化了区块链网络的性能、安全性和扩展性。但不可避免地，共识机制的处理效率会影响整个区块链事务的提交；同时在事务的背书和验证阶段，需要从区块链账本中读取相关数据，读取延迟历来是性能的瓶颈。

实验中发现，Fabric 由各种组件组成，如背书节点、排序服务和提交节点等。此外，它在处理交易中包含了不同的阶段，如背书阶段、排序阶段、验证和提交阶段。由于有许多组件和阶段，Fabric 提供了各种可配置参数，如区块大小、背书策略、通道、状态数据库等。因此，找到合适的配置参数值集可以有效地提高区块链的性能[18]。

4　仿真实验与结果分析

本节详细叙述了共享数字积分仿真实验的结果和性能的分析。通过使用 Hyperledger Fabric2.1 版本开发验证区块链共享数字积分管理机制的可行性和有效性；同时通过 Hyperledger Caliper 区块链性能基准框架，使用自定义用例测试共享积分区块链解决方案，获得性能测试结果[19]。

通过对实验结果的分析总结不同区块大小的配置和不同底层状态数据库下对区块链性能的影响。测试结果中包含的数据是在受控环境中测量的，在其他环境中得到的结果可能会有所不同。同时性能数据不能在不同版本的 Hyperledger Fabric 之间进行比较，因为测试软件和环境可能发生了变化。但根据实验数据分析，评判相关配置参数对区块链性能的影响是有意义的。

如图 8 所示的仿真实验测试的 Fabric 网络由 4 个组织组成，每个组织有 2 个对等节点，总共 8 个对等节点。有一个排序服务网络，由 3 个节点通过共识机制组成。

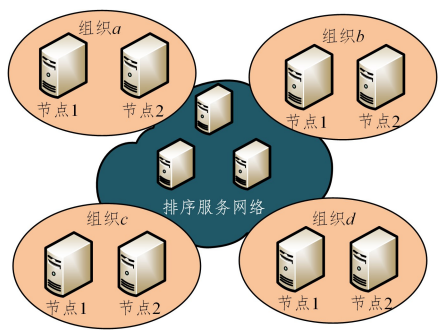


图 8　实验测试网络结构

Fig. 8　Experimental tested network structure

图 9 显示了随着事务发送率增加时，不同的区块大小(区块中所包含的交易数量不同)下的性能比较。

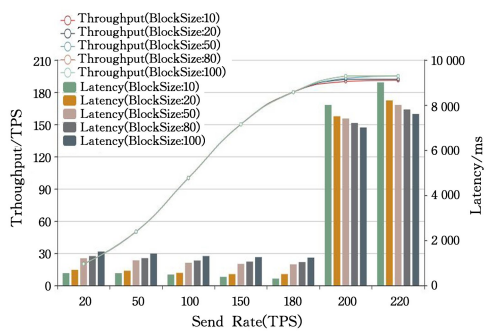


图 9　块大小和事务发送率对性能的影响

Fig. 9　Impact of the block size and transaction send rate on performance

从仿真实验测试的结果可以分析出：

(1)随着事务发送率的增加，吞吐量线性增长，直到在190 TPS左右持平。当发送率接近或超过饱和点时，延迟显著增加。这是因为在验证阶段，等待在VSCC队列中的有序事务数量增长，从而影响了提交延迟。

(2)当事务发送率低于饱和点时，要为区块链网络实现较低的事务延迟，总是使用较小的块大小。在这种情况下，吞吐量将与发送率匹配。当预期事务发送率很高时，为了实现更高的吞吐量和更低的事务延迟，总是使用更大的区块。即在区块较大而请求压力小的情况下，客户端必须等待消息达到区块大小并将其打包成块，或者等待时间达到区块生成时间。但是对于高并发的情况，大区块会使更多的消息变成一个块，从而降低网络成本，提高网络效率。相比之下，对于区块较小、请求压力较小的情况，排序节点等待的消息更少，从而更容易、更快地达到区块大小要求来生成区块。在高并发且区块较小的情况下，事务会被分割成许多更小的块，网络传输则非常耗时。

图10显示了Fabric中，两个底层状态数据库LevelDB和CouchDB随着请求的数据量的增加在性能方面的比较。LevelDB被嵌入到peer节点进程中。它将世界状态存储为键值对。LevelDB是Fabric官方文档中的默认状态数据库。CouchDB是另一个可选状态数据库，它可以支持富数据查询功能。

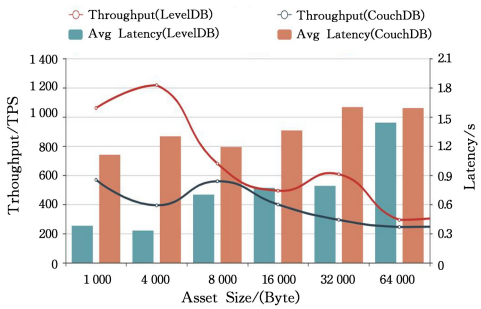


图10 CouchDB与LevelDB性能的比较

Fig. 10 CouchDB and LevelDB Performance comparison

从图10可以看出与LevelDB相比，CouchDB的缺点是数据处理效率较低。CouchDB作为独立的进程在peer节点进程之外运行，所以在设置、管理和操作中有更多的事情要做。如果有其他复杂的富查询需求，需要考虑从LevelDB迁移到CouchDB；否则，应该继续使用LevelDB来获得高性能。

结束语 区块链技术如今已越来越多地应用到企业中，本文提出了企业联盟共享数字积分管理机制，并结合超级账本的联盟链技术Fabric设计完成实验，同时通过实验数据对区块链吞吐量与延迟性能进行了研究。本文设计基于区块链的共享积分管理机制解决了当前传统企业积分管理机制的主要问题，采用分布式去中心化的管理机制，在保证用户积分数据安全、企业联盟数据共享的前提下，方便了用户的使用，降低了企业的管理成本。实验表明本系统实现了共享积分管理机制功能，保证了在多用户下系统可以拥有较高的吞吐量，但在处理积分数据的智能合约算法上还有待提高，对多种场景下的积分交易需求还未全部考虑到，后续研究中应对积分数据结构与智能合约处理进行改进，增强企业联盟用户积分体系。

参 考 文 献

[1] BEN E, LEO K, LEVARD H, et al. Blockchain for Enterprise: Overview, Opportunities and Challenges [C]// International Conference on Wireless & Mobile Communications. 2017: 1-6.

[2] CHIU J, KOEPPL T V. The economics of cryptocurrencies- bitcoin and beyond[J]. SSRN Electronic Journal, DOI: 10.2139/ssrn.3048124.

[3] WANG S G, LIU H F. Research on the point system method of mall based on blockchain[J]. Cyberspace Security, 2017, 8(Z5): 51-55.

[4] ZHANG M Y. Generic point management system based on blackchain[D]. Jinan: Shandong University, 2019.

[5] ALEKSIEVA V, VALCHANOV H, HULIYAN A. Implementation of Smart-Contract, Based on Hyperledger Fabric Blockchain[C]// 2020 21st International Symposium on Electrical Apparatus & Technologies (SIELA). 2020: 1-4.

[6] ANDROULAKI E, MANEVICH Y, MURALIDHARANS, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains [C]// The Thirteenth EuroSys Conference. 2018: 1-15.

[7] NAKAMOTO S. Bitcoin: A Peer-to-Peer Electronic Cash System [EB/OL]. https://bitcoin.org/bitcoin.pdf.

[8] ZHENG Z, XIE S, DAI H, et al. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends [C]// 2017 IEEE International Congress on Big Data (BigData Congress). IEEE, 2017: 557-564.

[9] TOYODA K, MACHI K, OHTAKE Y, et al. Function-level Bottleneck Analysis of Private Proof-of-Authority Ethereum Blockchain[J]. IEEE Access, 2020(99): 1-1.

[10] HUCKLE S, BHATTACHARYA R, WHITEM, et al. Internet of Things, Blockchain and Shared Economy Applications[J]. Procedia Computer Ence, 2016, 98: 461-466.

[11] BESSANI A, SOUSA J, VUKOLI M. A Byzantine Fault-Tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform [C]// The 1st Workshop. IEEE Computer Society, 2017: 1-2.

[12] HAEUPLER B, PANDURANGAN G, PELEG D, et al. Discovery through Gossip[J]. Random Structures & Algorithms, 2016, 48(3): 565-587.

[13] EYAL I, SIRERE G. Majority is not Enough: Bitcoin Mining is Vulnerable [C]// International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2014: 1-18.

[14] DECKER C, WATTENHOFER R. Information propagation in the Bitcoin network [C]// IEEE P2P 2013 Proceedings. IEEE, 2013: 1-10.

[15] DELMOLINO K, ARNETT M, KOSBA A, et al. Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab [C]// International Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2016: 1-15.

[16] GORENFLO C, LEE S, GOLAB L, et al. FastFabric: Scaling Hyperledger Fabric to 20,000 Transactions per Second [C]// 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). John Wiley & Sons, Ltd, 2020: 1-9.

[17] THAKKAR P, NATHAN S, VISHWANATHAN B. Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform [C]// 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS). IEEE, 2018: 1-13.
[18] BENHAMOUDA F, HALEVI S, HALEVI T. Supporting Private Data on Hyperledger Fabric with Secure Multiparty Computation [C]// 2018 IEEE International Conference on Cloud Engineering (IC2E). IEEE, 2018: 1-7.
[19] AMPEL B, PATTON M, CHEN H. Performance Modeling of Hyperledger Sawtooth Blockchain[C]// 2019 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, 2019: 59-61.

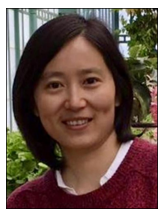
LING Fei, born in 1993, postgraduate. His main research interests include blockchian and big data.

CHEN Shi-ping, born in 1964, Ph. D, professor, Ph. D supervisor. His main research interests include cyber security and computer network communication.

（上接第 527 页）

[8] DOBRAUNIG C, EICHLSEDER M, GROβ H, et al. Statistical ineffective fault attacks on masked AES with fault countermeasures[C]// International Conference on the Theory and Application of Cryptology and Information Security. Cham: Springer, 2018: 315-342.
[9] ZHANG F, LOU X, ZHAO X, et al. Persistent fault analysis on block ciphers[J]. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2018(3): 150-172.
[10] ZHANG F, ZHANG Y, JIANG H, et al. Persistent fault attack in practice[J]. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2020(2): 172-195.
[11] BAR-EL H, CHOUKRI H, NACCACHE D, et al. The sorcerer's apprentice guide to fault attacks[J]. Proceedings of the IEEE, 2006, 94(2): 370-382.
[12] LOMNÉ V, ROCHE T, THILLARD A. On the need of randomness in fault attack countermeasures-application to AES[C]// 2012 Workshop on Fault Diagnosis and Tolerance in Cryptography. IEEE, 2012: 85-94.
[13] MESSERGES T S. Securing the AES finalists against power analysis attacks[C]// International Workshop on Fast Software Encryption. Berlin, Heidelberg: Springer, 2000: 150-164.
[14] PAN J, ZHANG F, REN K, et al. One fault is all it needs: breaking higher-order masking with persistent fault analysis [C]// 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2019: 1-6.
[15] BLOM G, HOLST L, SANDELL D. Problems and Snapshots from the World of Probability[M]. Springer Science & Business Media, 1993.
[16] CAFORIO A, BANIK S. A study of persistent fault analysis [C]// International Conference on Security, Privacy, and Applied Cryptography Engineering. Cham: Springer, 2019: 13-33.
[17] SELMKE B, BRUMMER S, HEYSZL J, et al. Precise laser fault injections into 90 nm and 45 nm sram-cells[C]// International Conference on Smart Card Research and Advanced Applications. Cham: Springer, 2015: 193-205.
[18] STALLINGS W. Cryptography and Network Security: Principles and Practice[M]. Beijing: Publishing House of Electronics Industry, 2017: 153-179.
[19] MANGARD S, OSWALD E, POPP T. Power Analysis Attacks [M]. Beijing: Science Press, 2010: 181-185.
[20] YAO Y, YANG M, PATRICK C, et al. Fault-assisted side-channel analysis of masked implementations[C]// 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST). IEEE, 2018: 57-64.
[21] DWORKIN M J. SHA-3 standard: Permutation-based hash and extendable-output functions: Federal Inf. Process. Stds. (NIST FIPS) - 202 [S]. NIST: 2015.
[22] MATSUDA K, FUJII T, SHOJI N, et al. A 286 f 2/cell distributed bulk-current sensor and secure flush code eraser against laser fault injection attack on cryptographic processor[J]. IEEE Journal of Solid-State Circuits, 2018, 53(11): 3174-3182.

WANG Jian, born in 1998, postgraduate. His main research interests include side-channel analysis and countermeasures.



CHEN Hua, born in 1976, Ph.D, senior engineer, Ph.D supervisor. Her main research interests include side-channel analysis and countermeasures.