软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2019,30(6):1670−1680 [doi: 10.13328/j.cnki.jos.005743]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

响应式许可链基础组件——RepChain*

李春晓[1], 陈 胜[2], 郑龙帅[1], 左 春[1], 蒋步云[2], 梁 赓[1]

[1](中国科学院 软件研究所,北京 100190)

[2](北京连琪科技有限公司,北京 100086)

通讯作者: 梁赓, E-mail: lianggeng@iscas.ac.cn

摘 要: 区块链系统的核心价值是建立多方信任,在面对企业级应用场景时需要增强安全性、实时性、友好性设计;面对国内自主可控和技术发展需求,应积极倡导开源共赢.为此,提出了开源的响应式许可链基础组件RepChain(reactive permissioned chain),它通过全新设计系统架构,突出了响应式、松耦合、轻量级、协同性共识、合约分级部署、运行状态可视化等特点.通过在身份准入的基础上建立安全信道,采用协同性共识代替公有链的竞争性共识,提高了交易的实时性和交易通量.经实验证明,响应式许可链在交易通量、实时性和韧性方面有较大提升.

关键词: 许可链;响应式;Actor;协同共识;可视化

中图法分类号: TP309

中文引用格式: 李春晓,陈胜,郑龙帅,左春,蒋步云,梁赓.响应式许可链基础组件——RepChain.软件学报,2019,30(6):1670−1680. http://www.jos.org.cn/1000-9825/5743.htm

英文引用格式: Li CX, Chen S, Zheng LS, Zuo C, Jiang BY, Liang G. RepChain—A permissioned blockchain toolkit implemented by reactive programming. Ruan Jian Xue Bao/Journal of Software, 2019,30(6):1670−1680 (in Chinese). http://www.jos.org.cn/1000-9825/5743.htm

RepChain—A Permissioned Blockchain Toolkit Implemented by Reactive Programming

LI Chun-Xiao[1], CHEN Sheng[2], ZHENG Long-Shuai[1], ZUO Chun[1], JIANG Bu-Yun[2], LIANG Geng[1]

[1](Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

[2](Linkel Technology Co., Ltd, Beijing 100086, China)

Abstract: The core value of the blockchain system is to establish multi-party trust. In enterprise application scenarios, it is necessary to enhance the security, real time performance, and user friendliness. To meet the needs of the domestic independent control and technical development, open source and win-win cooperation should be advocated. This paper proposes a permissioned blockchain toolkit implemented by reactive programming named RepChain (reactive permissioned chain), which is the first open source reactive permissioned blockchain toolkit in China. It highlights responsive, loosely coupled, lightweight, collaborative consensus, hierarchical contract deployment and visualization of real-time status through a novel system architecture design. A secure channel is established based on access control, and a synergistic consensus is used to replace the competitive consensus of the public blockchain, therefore, the transaction real-time performance and throughput are improved. Experiments show that the reactive permissioned blockchain can significantly increase transaction throughput, real-time performance and resilience.

Key words: permissioned blockchain; reactive; Actor; collaborative consensus; visualization

区块链(blockchain)作为一种分布式可信账本[1],以其具备的分布式、去中心化、可追溯、不可篡改等特点,

* 基金项目: 广州市科技计划(201802020015)
Foundation item: Science and Technology Plan Project of Guangzhou (201802020015)
本文由区块链与数字货币技术专题特约编辑斯雪明教授和陈文光教授推荐.
收稿时间: 2018-06-25; 修改时间: 2018-10-12; 采用时间: 2018-12-18; jos 在线出版时间: 2019-03-27
CNKI 网络优先出版: 2019-03-27 16:40:31, http://kns.cnki.net/kcms/detail/11.2560.TP.20190327.1640.007.html

在全球金融和 IT 等领域掀起一波发展热潮,诞生了更灵活的商业模式[2],并推动了信任经济时代的快速发展.

然而迄今为止,区块链技术中最具影响力的领域仍然局限于以比特币实验和以太坊为代表的公有链和虚拟货币领域[3−6],对于区块链技术在企业级场景的应用仍然处于探索阶段[7,8].区块链的核心价值是通过可被多方验证的账本数据建立多方信任,公有链系统中固化的用户匿名、激励机制、完全去中心化、竞争性共识都是在其特定场景下采用的技术手段而非核心价值,在企业应用中,必须根据具体场景对其重新审视.现有的公有链开源程序难以为企业级场景所利用.

许可链对于参与组网的节点采取身份准入,并在此基础上建立安全的通信信道.相对于公有链提高了节点之间的信任程度,以此为基础,可以通过减少参与共识节点的数量、用协同性的共识代替公有链的竞争性共识,从而达到提高实时性和交易通量的企业应用目标.因此,企业应用场景通常选择许可链的组网方式.现阶段也出现了一些基于许可链面向企业应用的开源组件,但它们或多或少存在以下问题.

- 系统缺乏模块化设计,系统各组成部分耦合高,难以对模块进行替换或裁剪;
- 设计的目标场景过于多样化,导致系统臃肿,当面对单一场景时,其裁剪工作量巨大;
- 缺乏性能的可扩展性设计,在 BAAS(blockchain as a service)模式下,理想的超级节点应该能做到节点整体或节点内的某个处理环节自适应业务压力,算力可伸缩;
- 缺少直观的对区块如何形成的展示过程.

为了解决上述问题,本文提出了响应式许可链的基础组件——RepChain.目前,国内并没有同类的开源基础组件,本文的原创性贡献如下.

(1) 采用无协商随机抽签算法(consultation-free random draw algorithm with distributed environment,简称 CFRD)协同性共识代替公有链的竞争性共识;

(2) 以异步消息交互代替传统的方法调用,实现了模块之间松耦合,方便根据不同场景对模块进行替换;

(3) 专注于提供必须的基础模块,用做加法的项目实施思路代替减法思路;

(4) 利用 Actor 集群化[9],可以实现节点或节点内部细粒度的自适应弹性计算,根据业务压力动态地申请或释放算力资源;

(5) 利用 Actor 的位置透明性,根据合约信任程度的不同采用不同的合约部署策略,从而合理解决了合约的安全隔离与执行性能之间存在的冲突;

(6) 采用图形化的实时状态显示,搜集和直观展示各节点入网与离网、数据同步、交易提交与传播、出块节点选举、候选块背书、正式出块的完整过程.

本文第 1 节介绍区块链的核心价值和响应式编程的特点,并对比公有链和许可链的组网方式.第 2 节介绍 RepChain 的系统结构,共分 6 个层次,对每一个层次的含义进行说明.第 3 节介绍 RepChain 的各个组成模块,对每个模块的设计方案进行说明.第 4 节介绍 RepChain 性能实验.最后总结全文并进行展望.

1 引 言

1.1 区块链技术

区块链系统的核心价值是建立信任,区块+链这种独特的数据结构[10]忠实、完整地记录了行为主体签名认可的授权行为[11].用户可以不信任区块链系统,但它出具的每条数据皆有其授权来源,是可追溯并可验证的.区块链的实施要避免类似传统系统越过行为主体的授权自行其是,并且好的区块链系统应该让行为主体在做出授权之前,清楚认识到其授权行为将可能面临的风险[12,13].

区块链的数据结构如图 1(a)所示,它是由一系列前后衔接的区块组成,每个区块里面包含按顺序排列的交易[14−16],每一个交易包含行为主体对授权行为的签名认可,以此来满足达共识[17]、可追溯[18]、防篡改[19]的目标,独特的链式结构确保了数据只能增加不能删除.

传统应用数据结构如图 1(b)所示,它是由各种表组成,数据本身缺乏可验证的授权来源,用户只能选择信任这个系统;而对于区块链来说,防篡改是指防止系统伪造或篡改数据,用户不需要信任系统,只需要信任系统出

具的可被验证的数据.

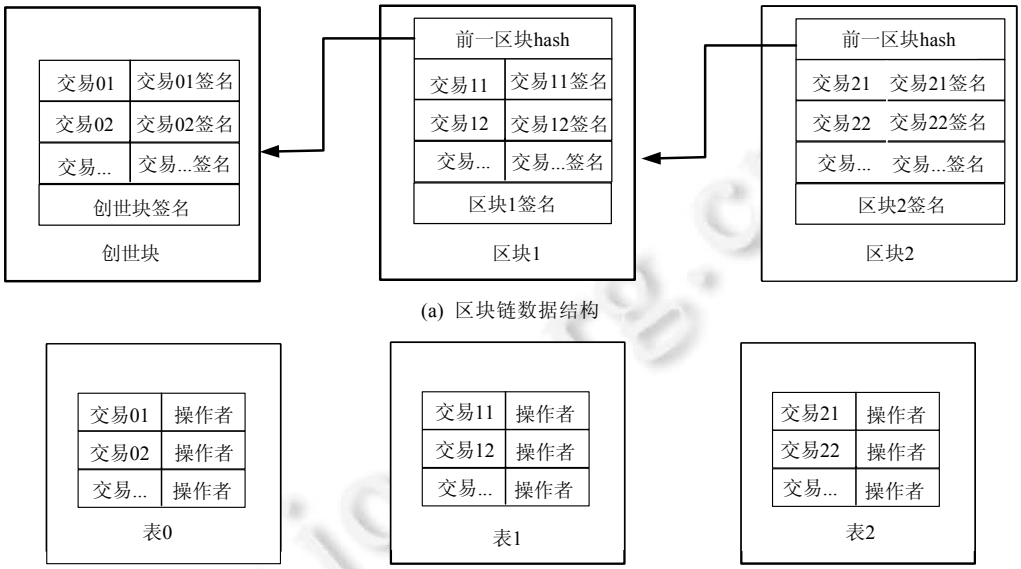

Fig.1 Data structure of blockchain and traditional application

图 1 区块链和传统应用数据结构

通过区块链和传统应用数据结构的对比可以发现:区块链最大的用处是通过对参与方授权行为完整的、忠实的记录来建立各个行为主体之间的信任,将对传统系统的信任切换到对系统出具的可验证数据的信任,这是区块链的本质特征.

公有链通常采用匿名、开放的组网策略[20],采用激励机制吸引更多的算力加入组网;反过来,在庞大算力上形成的多数共识,又确保了共识结果的可信.但同时,公有链采用的这种大规模组网上的竞争性共识导致其交易实时性差,交易通量不高.许可链面向企业联盟式行业应用,在身份准入的基础上建立安全信道,降低参与共识节点的规模,用协同性的共识算法取代公有链竞争性的共识算法,这样能够较大地提高交易实时性和交易通量.

两者的组网方式如图 2 所示.

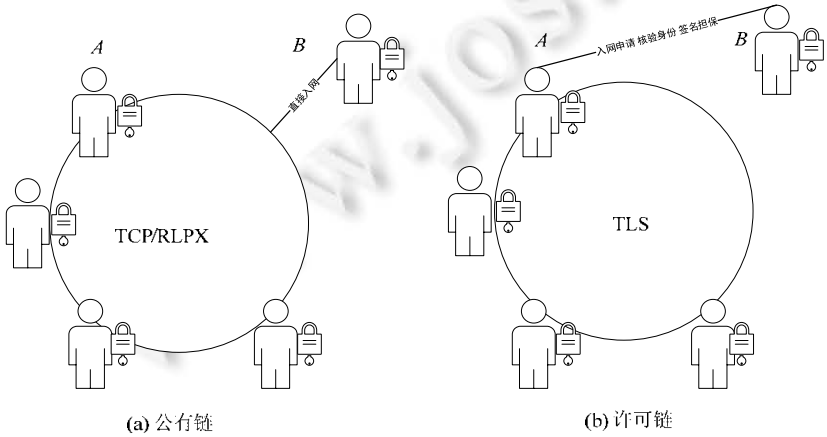

Fig.2 Access methods of public blockchain and permissioned blockchain

图 2 公有链和许可链入网方式

在图 2(a)的公有链中,用户节点 *B* 不需要其他节点(如在网节点 *A*)审核并验明身份;在图 2(b)的许可链中,

用户节点 *B* 入网申请需经过其他节点(如在网节点 *A*)进行核验身份,并签名担保.在公有链中,密钥是证明用户身份的唯一手段;而在许可链中,负责核验的节点后续可以协助用户恢复密钥.通过两者的比较可以发现:许可链在准入机制的基础上更容易支持密钥对的恢复,更好地保障用户的权益,同时还能更好地保护数据隐私.

1.2 响应式编程及实践

传统面向对象编程模型在应对高并发分布式系统时存在局限性.例如多个线程调用同一个方法时,需要用分布式锁保证共享变量的一致性,但加锁严重限制并发、多台机器通信的网络延迟过高且容易导致死锁.

Actor 模型[21]是响应式编程[22]的一种实现,在 Actor 系统中,每个实体都是一个响应消息的 Actor,Actor 的内部状态是私有的,以消息驱动代替传统的方法调用完成交互.经过验证,采用 Akka Actor 模型的 scala 语言显著降低了通信延迟,且在维持分布式系统中的休眠进程时表现最佳[23].

RepChain 的实现采用了 Actor 模型[24],表现出良好的容错性/韧性;同时,易于对系统的瓶颈环节进行算力调整.例如,在 RepChain 的性能优化过程中发现,请求背书环节存在比较严重的阻塞(耗时约 120ms),拖累了整个系统的每秒交易数(transactions per second,简称 TPS).因此,决定将背书请求的处理从原来的单 Actor 实例串行改为多 Actor 实例并行:将负责背书请求的模块调整为负责调度的 Actor 和负责背书处理的子 Actor;同时保持对外服务的背书请求消息格式不变,子 Actor 对背书请求消息的处理逻辑不变.改进后背书处理耗时降低了 40ms,系统的 TPS 指标提高了 60,改动的代码数不到 300 行[25].而同类系统采用 Kubernetes 等工具在容器级别进行算力调度,难以实现类似的模块/子模块级别的细粒度算力调整.

2 系统结构

参考工信部白皮书的分层模型[2],RepChain 针对企业级应用场景去掉了激励层,增加了 API 层和监控层.RepChain 系统共分为 6 层,从底层到上层分别是数据层、网络层、共识层、合约层、API 层、监控层,如图 3 所示.

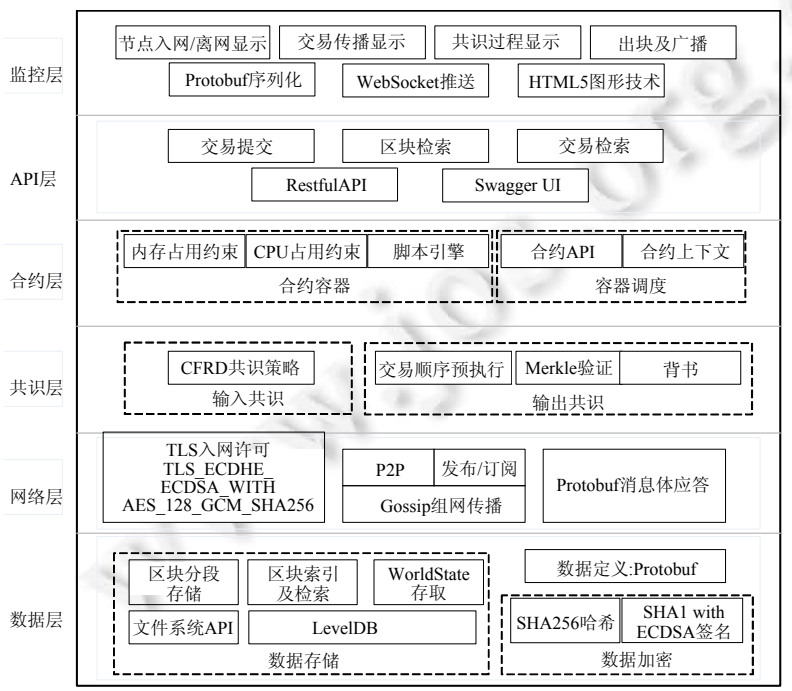

Fig.3 System structure of RepChain

图 3 RepChain 系统结构图

(1) 数据层:负责数据格式定义,数据结构采用 Protocol Buffers 定义文件,并以此为基础实现数据的交换、验证、存储、读取及检索;

(2) 网络层:采用 JDK 内置的 TLS 实现,支持入网许可验证,在此基础上进行去中心化的 gossip 组网,网络传播支持 P2P 和 Pub/Sub 两种方式;

(3) 共识层:完成区块的输入共识和输出共识.采用兼顾实时性和安全性的 CFRD 算法,既照顾到交易的实时性要求,又能在一定程度防止节点串通作弊;输入共识对入块的交易顺序达成一致,输出共识对交易顺序执行的结果达成一致;

(4) 合约层:为合约执行提供上下文环境,支持合约的动态部署、运行时加载和编译执行;

(5) API 层:提供外部接口,允许第三方应用以 Restful 的形式与系统交互,并允许开发者通过 Swagger UI 进行在线测试.API 层提供交易签名提交、区块和交易检索等基本功能;

(6) 监控层:在区块链网络中收集事件/日志,并将其以 Protocol Buffers 的格式推送至 Web 端,以 H5 图形技术进行实时状态的可视化展示和日志回放.

RepChain 采用轻量化和标准化设计,相比重新开发的非标准实现更稳定和易于升级(例如组网信道从 TLS1.2 升级到 TLS1.3).在目前已开源的功能完整的许可链基础组件中,RepChain 代码体量远小于同类实现,全部代码仅有 11 000 行左右,更适合在工程实践中根据业务场景进行定制修改.

3 模块设计

RepChain 模块化设计具备高内聚低耦合的特点,每个组网节点对应一个 ActorSystem,主要功能模块均封装为 Actor,包括 API 服务、合约执行、共识策略、事件订阅,如图 4 所示.去掉一个 Actor 不会影响其他 Actor 之间的消息交互,也不会产生编译依赖的问题,便于进行系统调整.

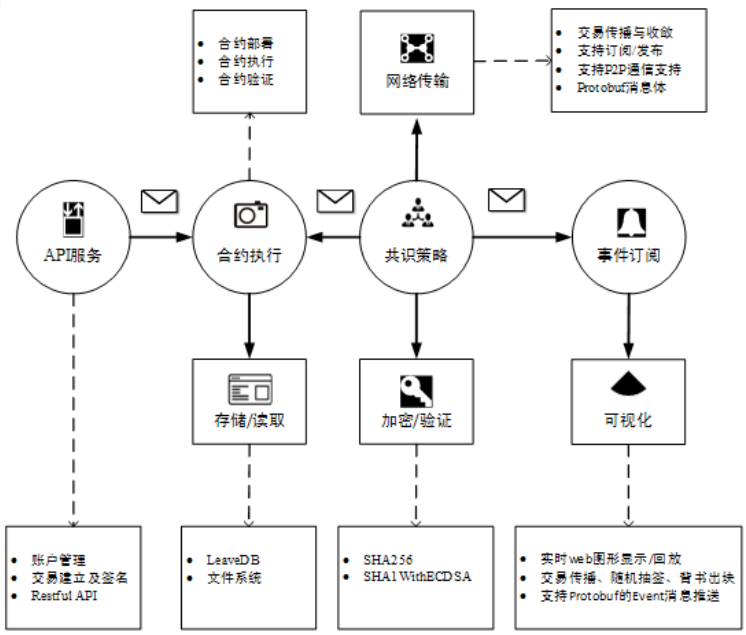

Fig.4 Interaction between modulars

图 4 模块化交互

(1) 合约执行

合约的安全隔离与执行性能是困扰开发者的普遍问题,为了避免非信任合约执行时对节点主机造成资源

(CPU、内存、存储)过度占用,通常都会对合约的执行采取进程隔离措施.但是由于合约执行仍然依赖节点主进程维护的 WorldState,因此不得不付出跨进程通信的性能代价.

RepChain 采用的 Actor 模型具备位置透明性,因此能够实现在合约执行时,按照合约的受信任程度,决定在节点进程内加载执行,还是独立的 JVM 进程执行,抑或在远程主机执行.根据合约信任程度的不同,采用不同的合约部署策略,从而合理解决了合约的安全隔离与执行性能之间存在的冲突.

另外,根据具体场景,Actor 模型能够对合约的执行方式进行调度:首先,执行请求被发送到根路由 Actor,根路由 Actor 将根据合约机制,决定哪些合约需要串行执行,哪些合约能够并发执行;然后,根路由 Actor 将串行执行请求发送给单例合约容器,将并发执行请求发送给下一子路由 Actor,以便调度更多 Actor 参与并发执行.如图 5 所示.

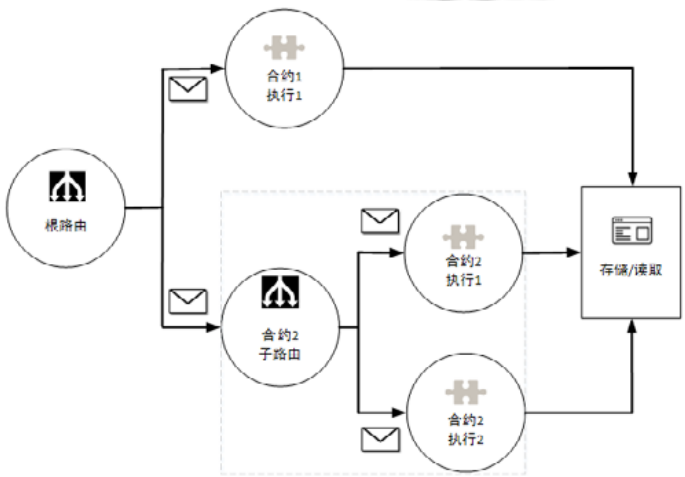

Fig.5 Execution of contract

图 5 合约执行

(2) 存储

RepChain 存储模块由文件存储和 LevelDB 构成,文件用来存放区块数据以及链信息,LevelDB 存放区块和交易的索引信息以及供合约读写的 WorldState 信息.存储模块提供快照以及状态回滚以支持交易预执行,对正式出块的交易执行结果进行数据验证之后持久化.

RepChain 存储结构设计如图 6 所示,功能包括:

- 支持区块的读写,对区块信息及交易信息的索引和检索;
- 支持在 WorldState 快照之上的读写以及交易预执行的状态回滚;
- 支持对 WorldState 进行 Merkle 计算.

在文件系统 API 之上,采用分段存储技术,避免单个大文件导致区块链数据的写入/读入性能下降;并且为了实现区块数据的快速检索,在 LevelDB 中建立对区块数据的内容索引,以支持高性能的检索.

(3) 通信

通信模块建立在 Akka Cluster 之上,组网节点之间以及节点与 Web 端之间消息的序列化均采用 Protocol Buffers(简称 Protobuf)协议.

Akka Cluster 采用去中心化的 gossip 协议组网,它具备以下功能.

- 节点之间采用 TLS1.2 作为安全通信信道;
- 通过订阅系统事件,每个节点可以维持一张全网在线节点视图;
- 支持节点之间采用发布/订阅方式通信;
- 支持节点之间采用 P2P 方式通信.

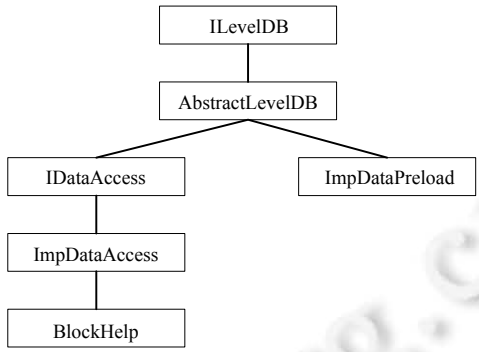

Fig.6 Design of classes structure for storage

图 6 存储类结构设计

(4) 共识

引入合约机制之后,不仅需要对打包入块的交易及其顺序进行输入共识,还需要对顺序执行交易对“账本”产生的影响进行输出共识.如图 7 所示.

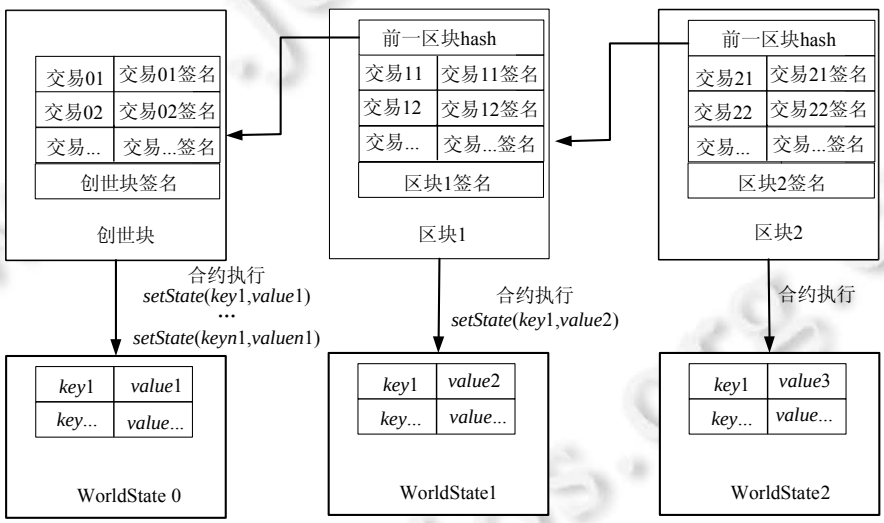

Fig.7 Consensus mechanism

图 7 共识机制

完成数据同步是参与共识的前提,只有数据达到最新区块高度的节点才有能力参与共识.RepChain 共识采用 CFRD 算法,各节点在无须协商的前提下,采用全网一致的伪随机种子抽签决定出块节点顺位序列.在每个时间段,只有唯一的一个节点拥有全网认可的出块资格.如果当前出块节点不能正常出块,随着时间的推移,出块权将顺位给下一个出块节点.

RepChain 采用两阶段共识:第 1 阶段出块节点负责将收集到并通过签名验证的交易按顺序打包入块,附上预执行获得的输出结果之后,签名形成预出块,然后向参与共识的节点发送预出块并请求背书;第 2 阶段出块节点收集到足够的背书之后,将背书附加到预出块形成正式出块,向全网广播.两阶段机制能够避免恶意节点的非法出块.

(5) 事件订阅与推送

RepChain 事件模块通过 Akka Cluster 的 Sub/Pub,以 Event Topic 广播到提供事件服务的 Event Actor.Event

Actor 订阅 Event Topic 并接收事件消息,通过 EventServer 将消息序列化为 Protocal Buffers 字节流并推送到浏览器,浏览器通过 WebSocket Client 接收到推送的字节流后将其反序列化为 Event 对象,然后调用可视化模块进行实时状态展示.主要过程如图 8 所示.

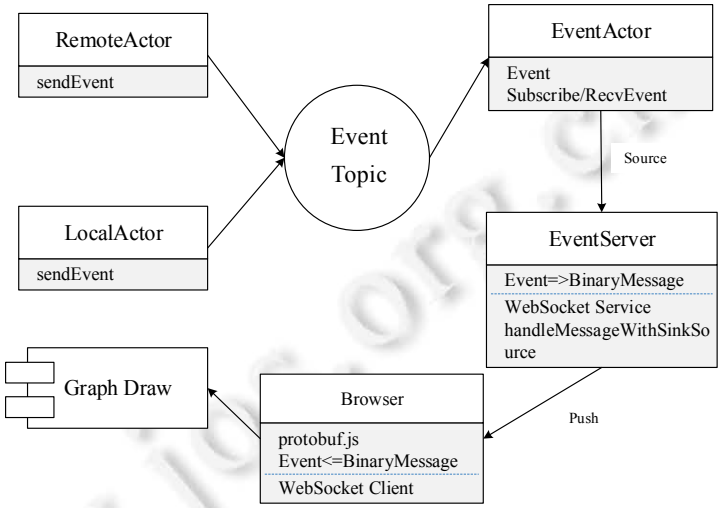

Fig.8 Subscription and push of event

图 8 事件订阅与推送

(6) 可视化

可视化模块将从系统中收集的实时状态,以图形化的方式呈现给用户.

RepChain 提供的实时状态图如图 9 所示:左边是图形区;右上部分是块堆栈区,显示不断生成的区块以及包含的交易内容;右下部分是日志区,以文本方式滚动显示接收到的事件.图形区圆周代表组网,圆周上的小圆代表当前参与组网的节点.小圆的不同颜色代表节点处于不同状态:亮绿色代表出块节点,深绿色代表背书节点,蓝色代表当前未完成数据同步的节点.

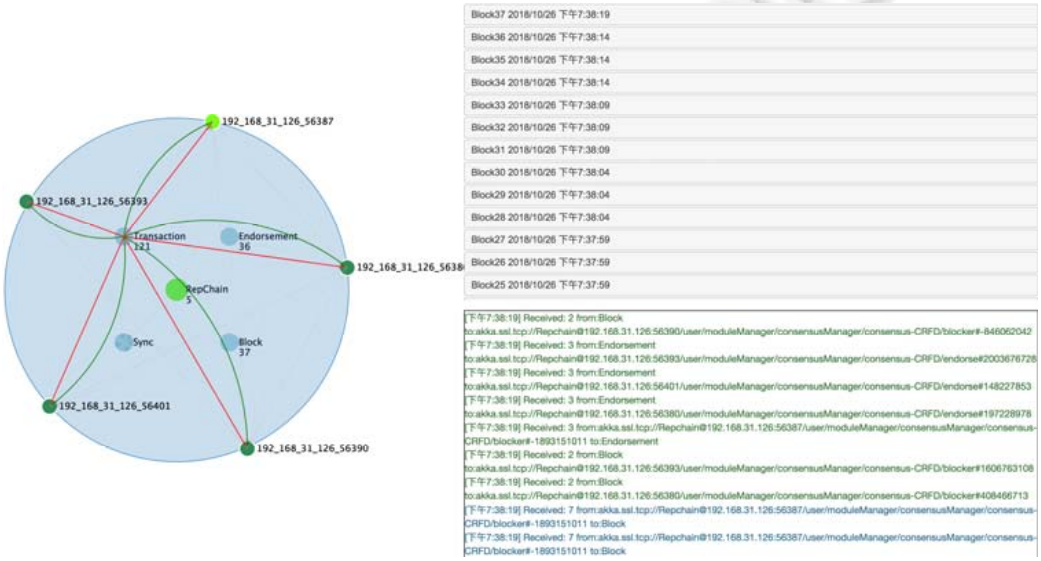

Fig.9 Real-time state diagram

图 9 实时状态图

图形区有以下几个 Topic.

1) RepChain:组网节点的个数;
2) Transaction:节点间发生交易请求;
3) Endorsement:节点需要背书;
4) Block:交易达成出块;
5) Sync:当新加入的节点与已有节点区块高度不一致时,发出数据同步请求.

组网节点与 Topic 之间的连线代表消息通信,其中,红线代表发送 Topic 消息,绿线代表从订阅的 Topic 接收到消息.实时图能够直观展示节点入网与离网、数据同步、交易提交与传播、出块节点选举、候选块背书、正式出块的完整过程.

4 RepChain 性能评估

Pongnumkul[26]完成了以太坊(版本 geth 1.5.8)在局域网上的性能测试实验.作为对比,本文在局域网上设计了 RepChain(版本 improved_tps_20181026)的性能测试实验[24],实验结果见表 1.

Table 1 Contract of experimental result

表 1 实验结果对比

名称	测试环境	交易通量		时延	
		一次发送 100 笔交易	一次发送 1 000 笔交易	一次发送 100 笔交易	一次发送 1 000 笔交易
以太坊	5 台 8 核 3.20GHz、15G 内存、128G SSD 存储空间的虚拟机、千兆网络	38.93TPS	34.40TPS	2.15s	26s
RepChain	5 台 8 核 2.40GHz、15G 内存、100G 存储空间的虚拟机、千兆网络	44.7TPS	143.30TPS	1.75s	4.84s

注:在持续不间断发送交易的情况下,RepChain 的 TPS 交易通量稳定在 320TPS

实验结果显示:交易量从 100~1 000 时,以太坊的交易通量下降、时延显著增加,RepChain 的交易通量上升、时延有所增加;在上述实验条件下,RepChain 的交易通量均高于以太坊,时延均低于以太坊,且随着交易量的增大优势更加明显.

实验结果表明,CFRD 协同性共识算法比公有链的竞争性共识算法在交易通量 TPS 和时延方面有较大改善,从而提升了响应式许可链的整体性能.

另外,本文在 30 个节点组网的环境下,采用手动终止节点进程的方式模拟个别节点的机器故障,系统都能够在 90s 内重新同步全网节点状态并继续正确运转.实验结果表明:在分布式环境下,RepChain 具备良好的容错性/韧性.因为 Actor 模型的消息驱动只确保消息达到的顺序,不确保消息一定送达,所以程序逻辑不能假设发出的消息一定收到预期的回应.在这种编程约束下,系统能够应对分布式环境下的各类异常.

5 总结与展望

区块链通过忠实完整地记录行为主体的授权行为,在参与者之间共享可验证、防篡改、可追溯授权的记录以建立信任.传统应用中的数据存在着被非法篡改的风险,而区块链应用通过对交易签名,保证了只有行为主体可以增加数据;通过对链式区块的签名,保证了已生成数据的固定(即不可删除或修改),因而在行为主体之间建立起了基于可信数据的信任.当前,区块链以公有链方式实现的数字资产应用引人瞩目,但在企业场景的应用探索才刚刚起步.

RepChain 以企业应用为目标场景,提供了国内第一款响应式、轻量级的许可链基础组件,优化了交易的实时性和交易通量.考虑到应用场景的多样性,RepChain 专注于基础功能,并且尽量保持模块之间的松耦合,以方便在具体实施中进行调整.

目前,RepChain 已经应用到存证溯源、供应链升级、消费积分等领域.基于 RepChain 的服装供应链实现了

各参与方的权益保障,进一步地各参与方通过可信及时的销售数据反馈,能够有效提高其市场应变能力. RepChain 为传统集中化的电子证据存证服务注入防篡改和可追溯的特性,使其存证要素可被验证,从而保护了电子证据的完整性,并奠定了其可采用性的基础.

我们希望对 RepChain 的开源能够赋予它更多的实践机会,在历练中不断改进,在实体经济加区块链的创新发展中发挥应有的作用.

References:

[1] Zuo C, Liang G, Xu H, *et al*. Contrastive analysis of blockchain and traditional software technology. Information Technology and Standardization, 2017,(5):23−27. (in Chinese with English abstract).

[2] Ministry of Industry and Information Technology. White Paper on China's Blockchain Technology and Application Development (2018). 2018 (in Chinese).

[3] Shao QF, Jin CQ, Zhang Z, *et al*. Blockchain: Architecture and research progress. Chinese Journal of Computers, 2018,41(5):969−988 (in Chinese with English abstract).

[4] Xu XW, Weber I, Staples M, *et al*. A taxonomy of blockchain-based systems for architecture design. In: Proc. of the 2017 IEEE Int'l Conf. on Software Architecture (ICSA). IEEE, 2017. 243−252. [doi: 10.1109/ ICSA.2017.33]

[5] Cachin C. Architecture of the hyperledger blockchain fabric. In: Proc. of the Workshop on Distributed Cryptocurrencies and Consensus Ledgers. 2016.

[6] Vukolić M. Rethinking permissioned blockchains. In: Proc. of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts. ACM Press, 2017. 3−7. [doi: 10.1145/3055518.3055526]

[7] Sillaber C, Waltl B. Life cycle of smart contracts in blockchain ecosystems. Datenschutz und Datensicherheit—DuD, 2017,41(8): 497−500.

[8] Min XP, Li QZ, Kong LJ, *et al*. Permissioned blockchain dynamic consensus mechanism based multi-centers. Chinese Journal of Computers, 2018,41(5):1005−1020 (in Chinese with English abstract).

[9] Hewitt C. Actor model of computation: Scalable robust information systems. arXiv preprint arXiv:1008.1459, 2010.

[10] Kosba A, Miller A, Shi E, *et al*. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In: Proc. of the 2016 IEEE Symp. on Security and Privacy (SP). IEEE, 2016. 839−858.

[11] Neisse R, Steri G, Nai-Fovino I. A blockchain-based approach for data accountability and provenance tracking. In: Proc. of the 12th Int'l Conf. on Availability, Reliability and Security. ACM Press, 2017. 14. [doi: 10.1145/3098954.3098958]

[12] Micheler E, Von Der Heyde L. Holding, clearing and settling securities through blockchain/distributed ledger technology: Creating an efficient system by empowering investors. Journal of International Banking & Financial Law, 2016,31(11).

[13] Jang H, Lee J. An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. IEEE Access, 2018,6:5427−5437. [doi: 10.1109/ACCESS.2017.2779181]

[14] Decker C, Wattenhofer R. Information propagation in the bitcoin network. In: Proc. of the 13th Int'l Conf. on Peer-to-Peer Computing (P2P). IEEE, 2013. 1−10.

[15] Xu XW, Pautasso C, Zhu LM, *et al*. The blockchain as a software connector. In: Proc. of the 13th Working IEEE/IFIP Conf. on Software Architecture (WICSA). IEEE, 2016. 182−191.

[16] Watanabe H, Fujimura S, Nakadaira A, *et al*. Blockchain contract: A complete consensus using blockchain. In: Proc. of the 4th Global Conf. on Consumer Electronics (GCCE). IEEE, 2015. 577−578.

[17] Gipp B, Meuschke N, Beel J, *et al*. Using the blockchain of cryptocurrencies for timestamping digital cultural heritage. In: Proc. of the Workshop on Web Archiving and Digital Libraries (WADL) Held in Conjunction with the 16th ACM/IEEE Joint Conf. on Digital Libraries (JCDL). 2016.

[18] Liang, XP, Shetty S, Tosh D, *et al*. Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In: Proc. of the 17th IEEE/ACM Int'l Symp. on Cluster, Cloud and Grid Computing. IEEE Press, 2017. 468−477. [doi: 10.1109/CCGRID.2017.8]

[19] Browning TR. Applying the design structure matrix to system decomposition and integration problems: A review and new directions. IEEE Trans. on Engineering Management, 2001,48.3:292−306. [doi: 10.1109/17.946528]

[20] He JH, Chen Y, Li BZ, *et al*. Research of financial security model based on compliance chain. In: Proc. of the 2017 Int'l Conf. on Arts and Design, Education and Social Sciences (ADESS 2017). Wuhan: Zhicheng Times Cultural Development Co., Ltd., 2017.

[21] Agha GA. Actors: A Model of Concurrent Computation in Distributed Systems. London: MIT Press Series in AI, 1987.

[22] The reactive manifesto. https://www.reactivemanifesto.org/

[23] Valkov I, Chechina N, Trinder P. Comparing languages for engineering server software: Erlang, go, and scala with akka. In: Proc. of the 33rd Annual ACM Symp. on Applied Computing. 2018. 218−225. [doi: 10.1145/3167132.3167144]

[24] RepChain: Reactive permission chain. 2017. https://gitee.com/BTAJL/repchain.git

[25] Concurrency. 2018. https://gitee.com/BTAJL/repchain/commit/949391c3f3b7d70469b6202192244843cd20d68a

[26] Pongnumkul S, Siripanpornchana C, Thajchayapong S. Performance analysis of private blockchain platforms in varying workloads. In: Proc. of the 26th Int'l Conf. on Computer Communication and Networks (ICCCN). IEEE, 2017. 1−6. [doi: 10.1109/ICCCN.2017.8038517]

附中文参考文献:

[1] 左春,梁赓,徐昊,等.区块链技术与传统软件技术对比分析.信息技术与标准化,2017,(5):23−27.

[2] 工业和信息化部信息中心.2018 年中国区块链产业白皮书.2018.

[3] 邵奇峰,金澈清,张召,等.区块链技术:架构及进展.计算机学报,2018,41(5):969−988.

[8] 闵新平,李庆忠,孔兰菊,等.许可链多中心动态共识机制.计算机学报,2018,41(5):1005−1020.

李春晓(1968－),女,河北深州人,高级工程师,CCF 专业会员,主要研究领域为区块链,数字博物馆,数字图书馆.

左春(1959－),男,研究员,主要研究领域为软件工程.

陈胜(1972－),男,高级工程师,主要研究领域为区块链核心组件架构,区块链技术的应用.

蒋步云(1972－),男,软件设计师,主要研究领域为区块链.

郑龙帅(1990－),男,助理工程师,主要研究领域为区块链,计算机应用.

梁赓(1962－),男,高级工程师,CCF 专业会员,主要研究领域为区块链,计算机应用.