Journal of Computer Applications
计算机应用, 2022, 42(12): 3785 - 3791
ISSN 1001-9081
CODEN JYIIDU
2022-12-10
http://www.joca.cn

文章编号:1001-9081(2022)12-3785-07
DOI:10.11772/j.issn.1001-9081.2021101711

基于BLS聚合签名技术的平行链共识算法优化方案

刘　琪[1],郭荣新[1*],蒋文贤[2],马登极[3]
(1.华侨大学 信息科学与工程学院,福建 厦门 361021; 2.华侨大学 计算机科学与技术学院,福建 厦门 361021;
3.杭州复杂美科技有限公司,杭州 310061)
(*通信作者电子邮箱grxeee@hqu.edu.cn)

摘　要:目前,平行链的每个共识节点均需发送各自的共识交易到主链上以参与共识,这导致大量的共识交易严重占用主链的区块容量,并且浪费手续费。针对上述问题,利用平行链上的共识交易具有共识数据相同签名不同的特点,结合双线性映射技术,提出一种基于BLS聚合签名技术的平行链共识算法优化方案。首先,用共识节点对交易数据进行签名;然后,用平行链各节点通过点对点(P2P)网络在内部广播共识交易并同步消息;最后,由Leader节点统计共识交易,且当共识交易的数量大于2/3时,将对应的BLS签名数据聚合并发送交易聚合签名到主链上进行验证。实验结果表明,所提方案与原始平行链共识算法相比能够有效解决平行链上共识节点重复发送共识交易到主链的问题,在减少对主链存储空间的占用的同时节省交易手续费,只占用主链存储空间4 KB并且只产生一笔0.01比特元(BTY)的交易手续费。

关键词:平行链;共识算法;BLS聚合签名;Leader节点;双线性映射
中图分类号:TP301.6　　**文献标志码**:A

Parallel chain consensus algorithm optimization scheme based on Boneh-Lynn-Shacham aggregate signature technology

LIU Qi[1], GUO Rongxin[1*], JIANG Wenxian[2], MA Dengji[3]
(1. *College of Information Science and Engineering, Huaqiao University, Xiamen Fujian* 361021, *China*;
2. *College of Computer Science and Technology, Huaqiao University, Xiamen Fujian* 361021, *China*;
3. *Hangzhou Fuzamei Technology Company Limited, Hangzhou Zhejiang* 310061, *China*)

Abstract: At present, each consensus node of the parallel chain needs to send its own consensus transaction to the main chain to participate in the consensus. As a result, a large number of consensus transactions occupy the block capacity of the main chain seriously and waste transaction fees. In order to solve the above problems, an optimization scheme of parallel chain consensus algorithm based on BLS (Boneh-Lynn-Shacham) aggregate signature technology was proposed by combining bilinear map technology with the characteristics of the same consensus data and different signatures of consensus trades on parallel chain. Firstly, the transaction data was signed by the consensus node. Then, the consensus transaction was broadcasted by each node of the parallel chain and the message was synchronized internally through P2P (Peer-to-Peer) network. Finally, the consensus transactions were counted by Leader node. When the number of consensus transactions was greater than 2/3, the corresponding BLS signature data was aggregated and the transaction aggregate signature was sent to the main chain for verification. Experimental results show that compared with the original parallel chain consensus algorithm, the proposed scheme can effectively solve the problem of consensus nodes on the parallel chain repeatedly sending consensus transactions to the main chain, save transaction fees with reducing the occupancy of the storage space of the main chain, only occupy 4 KB of the storage space of the main chain and only generate a transaction fee of 0.01 BiT Yuan (BTY).

Key words: parallel chain; consensus algorithm; BLS (Boneh-Lynn-Shacham) aggregate signature; Leader node; bilinear map

收稿日期:2021-10-08;**修回日期**:2022-01-12;**录用日期**:2022-01-24。
基金项目:国家自然科学基金资助项目(61901182);2020部省共建经费资助项目(605-52520005)。
作者简介:刘琪(1995—),女,湖北宜昌人,硕士研究生,主要研究方向:区块链; 郭荣新(1980—),男,福建泉州人,高级实验师,硕士,主要研究方向:区块链; 蒋文贤(1974—),男,福建漳州人,副教授,博士研究生,主要研究方向:区块链、信息网络安全; 马登极(1981—),男,浙江杭州人,硕士,主要研究方向:区块链。

0 引言

中本聪于2008年11月1日发表一篇关于阐述比特币原理的白皮书《比特币：一种点对点式(Peer-to-Peer，P2P)的电子现金系统》[1]，提出一种基于点对点技术的电子现金系统，使网上支付由一方发起，直接支付给另一方，这个过程没有通过任何的金融机构。如果在第三方的监督下才能解决双重支付的问题，那么中本聪提出的这个系统将没有意义。第三方或者中介机构的信任问题一直存在，区块链[2]的出现旨在解决信任问题，它被称作是“信任的机器”。

区块链技术包括分布式存储技术、密码学技术、智能合约和共识机制等，具有数据不可篡改、信息可追溯、公开透明、集体维护、匿名性等特点。随着近几年区块链技术的快速发展，区块链的应用不再局限于金融领域，目前已经渗透于各行各业，比如存证溯源、政务、资产数字化、智慧监管等领域。尽管区块链技术迅猛发展，但是在去中心化的架构下仍然存在性能和交易吞吐量等问题，区块链只有解决此类问题才能有更大规模的落地。Chain33、Polkadot[3]、Cosmos[4]均使用类似的跨链技术[5]来解决区块链面临的性能问题。Cosmos的技术模式与Chain33和Polkadot不同，对于使用者来说门槛更高，每个使用者都要组共识、搭节点，并且链上数据维护成本高，而Chain33成本相对较低可快速部署。Chain33和Polkadot都是平行链模式均采用平行链架构，Polkadot目前在测试网阶段，从Polkadot发展规划来看，Chain33目前已经实现的资产跨链流通等功能，相对领先1~2年。

Polkadot最早提出平行链的概念是为了解决区块链在互操作性、可扩展性以及共享安全性方面存在的弊端，但到目前为止还没有实际的应用落地。平行链的概念最早由杭州复杂美科技有限公司提出，百度等也在白皮书中引入这个概念。Chain33是由杭州复杂美科技有限公司创建的区块链底层系统，Chain33首创的平行链架构中平行链依附于主链并且使用主链的共识网络，复杂的功能在平行链上完成，主链只做数据存储和共识，即使出现性能问题或者智能合约受到攻击，也仅破坏平行链不会影响主链的安全；由于主链的安全性极高，因此平行链的所有数据可以从主链同步回来。同时平行链也是在主链的基础上搭建的区块链，可以发展独立的区块链生态，拥有自己的节点和部分共识、能够编写多种智能合约、拥有独立的钱包、拥有独立的区块链浏览器等；基于主链统一的交易共识，不论是平行链和主链，还是不同的平行链之间的跨链资产都可以做到无缝转移，使之前的跨链时间从小时分钟级别缩小到几秒，跨链效率高。Polkadot与Chain33平行链对比分析如表1所示。

表1 Polkadot与Chain33平行链的对比分析

Tab. 1 Comparative analysis of Polkadot and Chain33 parallel chain

功能类型	Polkadot	Chain33平行链
出块机制	1)单独出块，2)从验证节点恢复	1)单独出块，2)从主链恢复
共识机制	1)双共识，优先出块共识，2)中继链共识最终化	1)先在主链达成共识，2)平行链出块并发送到主链共识，3)同步回平行链二次共识验证
治理	收集人，验证人，钓鱼人，提名人	超级节点，监督节点，普通节点
跨链机制	1)XCMP内部消息跨链，2)转接桥跨链	1)一笔跨链交易完成内部跨链，2)合约桥外部跨链
平行链接入	有限卡槽拍卖	注册制，数量不限

比特元(BiTYuan，BTY)是一种简单稳定、可扩展性强的公有链网络。比特元区块链的研发基于Chain33底层架构，是全球首个实现平行链架构的公有链网络。在比特元区块链上，以比特元为主链向外延伸多条平行链，将不同的交易分散到不同的平行链上执行，分担主链的运算负荷，同时提升整个系统的运行效率。比特元上的各条平行链既可独立开发去中心化应用(Decentralized Application，DApp)，建设多样化的应用生态，又可实现多链间的跨链交易和数据交换功能。在比特元中交易首先发送到主链被共识打包，随后同步到平行链上被执行，最后将执行结果的hash写回主链进行共识，实现交易并行执行提升系统吞吐量TPS(Transactions Per Second)，在此过程中BTY是交易必需的燃料。本文以比特元区块链网络为例介绍Chain33平行链架构以及结合BLS聚合签名技术对平行链共识算法做优化。

1 区块链中常用的共识算法

共识机制在去中心化的思想上解决了节点之间相互信任的问题，是区块链能够稳定持续运行下去的主要力量。目前区块链中常用的共识算法[6]可分为两大类：公有链和许可链。公有链的典型代表有工作量证明(Proof-of-Work，PoW)[7]、权益证明(Proof-of-Stake，PoS)[8]以及股份授权证明(Delegate Proof of Stake，DPoS)[9-10]；许可链的典型代表有实用拜占庭容错(Practical Byzantine Fault Tolerance，PBFT)[11]和(Replicated And Fault Tolerant)[12]。对上述共识算法从去中心化程度、容错率、资源消耗以及吞吐量等方面作简单对比分析，如表2所示。

表2 区块链常用共识算法的对比分析

Tab. 2 Comparative analysis of commonly used consensus algorithms in blockchain

共识机制	去中心化程度	容错率/%	资源消耗	吞吐量
PoW	去中心化	50	高	低
PoS	去中心化	50	较低	较高
DPoS	部分去中心化	50	低	高
PBFT	部分去中心化	33	低	高
RAFT	部分去中心化	非拜占庭容错	低	高

这些常用的共识算法都可应用于主链中，而平行链中每个超级节点都会向主链提供共识交易。n个超级节点的共识结果在主链达成共识，共识的规则是2/3的超级节点结果一致，第2章将详细介绍平行链的共识算法。

2　平行链架构

2.1　主链+平行链架构

图1展示了主链+平行链架构,该架构中一条主链附属多条平行链,每条平行链只与主链交互,各条平行链之间互相不干涉对方的交易。平行链的共识流程为:首先共识交易在主链上被打包;紧接着平行链上的超级节点从主链同步共识交易,并打包交易数据生成区块同时上传上链请求;然后主链上的节点广播共识交易并验证共识结果的正确性;最后平行链上的所有节点同步交易数据。

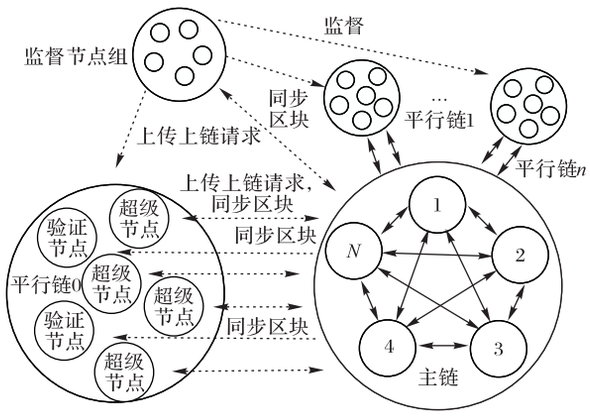

图1　主链+平行链架构

Fig. 1　Main chain + parallel chain architecture

主链+平行链架构具有如下几个特点:

1)高效性:平行链发送共识交易到主链上进行共识验证和数据存储,多条平行链交易并行处理,共识效率大幅提升。

2)稳定性:平行链上运行复杂的功能,主链只做存证等一些核心功能,整个架构简单稳定。

3)安全性:平行链的安全由主链保障,当智能合约或虚拟机被攻击时仅破坏平行链,平行链上的所有数据可以快速从主链同步,从而保证数据完整不被更改。

4)高扩展性:平行链拥有自己的生态,能在链上发行自己的通证(token),相当于一条独立自主的公链,并且支持主链和平行链跨链和平行链之间的跨链交易。

2.2　平行链的节点设计

Chain33平行链上的节点有授权节点、超级节点、监督节点和普通节点4种角色,下面对它们分别进行介绍。

1)授权节点。

授权节点是参与共识、发送共识交易的节点。不是所有的节点都有权限发送共识交易,只有授权节点可以发送共识交易,普通节点没有权限发送共识交易。

2)普通节点。

普通节点不发送共识交易,只接收共识交易并校验共识结果,以此体现区块链分布式一致性校验的设计思路,即授权节点负责共识安全,普通节点保持和授权节点一致,所有节点协同统一。

3)超级节点。

超级节点负责平行链的共识安全,并得到一定的挖矿奖励,每个超级节点都会向主链发送共识交易,n个超级节点达成共识的规则是超过2/3的超级节点共识结果一致;只有有了超级节点共识,平行链的跨链功能才可以启用。

4)监督节点。

监督超级节点的共识过程,防止超级节点联合作弊。当超级节点的共识结果和监督节点的结果不一致时,暂停共识,直到超级节点或监督节点修正共识结果达成共识为止。

2.3　平行链共识算法的过程分析

主链上的共识算法有多种,比如在公有链中使用PoS,在联盟链中使用Tendermint或PBFT,在私链中使用RAFT。平行链上的共识算法与主链使用的共识算法不同,主要的区别是平行链依赖于主链的共识网络,主要过程为平行链授权节点或超级节点发送共识交易到主链上进行共识,然后再同步共识交易到平行链进行自共识验证,共识规则为超过2/3节点共识结果一致则达成共识,自共识不通过的平行链节点将停止生成区块。

表3详细描述了平行链共识过程之前先设定共识过程中涉及的符号及其含义。

表3　符号及其含义

Tab. 3　Symbols and their meanings

符号	含义	符号	含义
x_i	平行链上超级节点	Num	达成共识的节点数
X_i	主链上节点	T_x	共识交易
B_i	待共识的区块	s_hash	状态哈希
S_i	签名信息	p_chain	平行链
H_i	待共识的区块高度	C_i	主链
MS_i	各交易的状态信息	u	共识标识
RM_i	区块信息	R	共识结果

目前平行链的共识算法如下所示。

1)各参数的设定。

① 设平行链上待共识的区块为B_1;

② 设第一平行链为$p_chain_1(x_1, x_2, \cdots, x_n)$,其中$x_1, x_2, \cdots, x_n$称为超级节点,并且满足$n \geqslant 3f + 1, f \in \mathbf{N}^+$;

③ 设$p_chain_1(x_1, x_2, \cdots, x_n)$对应的主链为$C_1(X_1, X_2, \cdots, X_n)$;

④ 待共识的区块B_1中包含的若干信息有:状态哈希s_hash、签名信息S、待共识的区块高度H和打包各交易的状态信息MS;

⑤ 各超级节点$(x_1, x_2, \cdots, x_n)$打包B_1的若干信息生成区块信息$BM_i = \{s_hash_i, H_i, S_i, MS_i\}$;

⑥ 达成共识的节点数设为Num。

2)共识过程。

①$x_1, x_2, \cdots, x_n$分别将$BM_1 = \{s_hash_1, H_1, S_1, MS_1\}, BM_2 = \{s_hash_2, H_2, S_2, MS_2\}, \cdots, BM_n = \{s_hash_n, H_n, S_n, MS_n\}$发送至主链$C_1$上对应节点$x_1, x_2, \cdots, x_n$处。由于发送1笔共识交易将产生0.001 BTY的手续费,因此n个共识节点发送n笔共识交易需产生$0.001n$ BTY的手续费。

② $x_1, x_2, \cdots, x_n$将$BM_1, BM_2, \cdots, BM_n$记录到主链上并互相广播。

③$X_1, X_2, \cdots, X_n$每个节点有$R_1, R_2, \cdots, R_n$这n笔共识交易的全部内容,由于1笔共识交易占用4 KB的存储空间,因此n笔共识交易共占用主链$4n$ KB的存储空间。

④ 此时进行共识验证,$x_1, x_2, \cdots, x_n$从C_1同步共识交易$T_x = \{BM_1, BM_2, \cdots, BM_n\}$,此时$p_chain_1(x_1, x_2, \cdots, x_n)$上每个节点都含有其他节点的共识交易信息。

⑤ p_chain_1进行自共识,若$Num \geqslant n*2/3$,即

$\{BM_1, BM_2, \cdots, BM_n\}$中至少有$n$*2/3个区块信息相同则达成共识。

⑥假设满足阈值，达成共识，共识结果为$R=\{s_hash, H, S, MS\}$生成的共识标识为u，共识高度为H。

2.4 平行链共识算法的缺陷

由2.3节可知平行链的共识过程为双层共识，首先平行链交易发送到主链上进行共识打包；然后平行链同步主链的区块并选取与本平行链有关的交易执行，再将区块哈希发送到主链上做共识；最后平行链上的节点同步主链的共识交易进行二次共识。平行链共识全过程的缺陷分析如下：

1）平行链上的超级节点越多，发送到主链上的共识交易就越多，多笔相同的共识交易将占据主链大量的存储空间；并且多条平行链依附于一条主链，主链上交易记录的越多，主链的交易处理能力[13]就越弱。

2）平行链上的超级节点每发送一次共识交易到主链上就会产生一笔手续费，当多个超级节点同时发送共识交易则会产生多笔手续费，同时消耗大量的资源。

3 平行链共识算法优化设计

3.1 Leader节点选取算法

在平行链聚合签名方案中，需要选取一个Leader节点发送整个共识交易，并且支持轮换操作。

当前的一些Leader节点选取算法比较复杂，比如有权重设计，或者不支持轮换操作，只让一个节点发送共识交易，可能只消耗一个节点的手续费，不具备公平性。此处通过对比RAFT[14]中的Leader节点选取方案可知，在RAFT共识机制中，Leader节点选举成功后会不断地向follower跟随节点发送心跳消息，选举周期将会一直持续到某个follower节点没有收到心跳消息并成为candidate候选节点为止，这时开启下一轮Leader节点的选举。由于RAFT中Leader节点任期时间无法确定并且不支持轮换操作，每个节点消耗的手续费差别大，因此不适用于平行链的共识算法。针对上述不足，设计较为轻量的Leader节点选举和轮换机制，保证系统稳定。

决定Leader节点的因素是高度（*height*）和偏移（*offset*），当*offset* = 0时，有关超级节点中Leader节点的选举和轮换机制如下。

1）平行链各个超级节点配置成功后，顺序固定，依次作为Leader节点。

2）假设现有超级节点A、B、C、D，对应的索引依次为0、1、2、3。

3）现设置初始Leader节点索引为0，即*base*=0，每隔一定共识高度轮换下一个节点为Leader节点，假设每隔100高度做一次轮换。

4）设置*base* = (*height*/100)% *nodes*，得到当前共识高度下Leader节点的索引，*height*依次增长，*base*也依次增长。

5）Leader节点需要每隔一段时间比如t = 5 s发送心跳消息，向其他节点声明自己是Leader节点；同时其他非Leader节点也在等待接收心跳消息。

6）Leader节点每隔5 s发送一次心跳消息，只要非Leader节点在1 min以内能够收到一次心跳消息就不用做*offset*偏移；如果非Leader节点在1 min之内没有收到来自Leader节点的心跳消息，那么设置一个*offset*偏移值，令*offset* = (*offset* + 1)% *nodes*，并且*Leader* = (*offset* + *base*)% *nodes*可以唯一确定一个新的Leader节点。

这种情况又分为以下两类。

1）Leader节点确实没有发送心跳消息，非Leader节点重新选取下一个Leader节点；若本节点发现自己是Leader节点，则广播心跳消息。

2）如果仅自己没有收到心跳消息，其他非Leader节点都收到心跳消息，那么仅该节点偏移Leader节点，直到自己是Leader节点为止，广播心跳消息。

但这样会出现两个Leader节点的情况，如果同时有多个不同Leader节点的心跳消息，则节点会和自己索引作比较。

①若两个*base*索引不同，则不处理，说明两个Leader节点不是同样的共识高度。

②若*base*相同，*offset*不同，如果自己的*offset*索引比较小，则接受大的Leader节点索引为Leader，具体为设置新的*offset*；如果自己*offset*比较大，则不处理。

这样在处理一些僵尸节点的时候，系统会自动切换到下一个节点为Leader节点继续服务，发送共识消息；因为共识*height*不变，所以此时*base*也不变，只是*offset*改变发生偏移。

当共识高度超过预先设置的间隔之后，*base*才会发生改变，从而更新下一个Leader节点，保持系统共识稳定。

3.2 BSL签名算法

BLS（Boneh-Lynn-Shacham）签名算法由Boneh等[15]提出，BLS签名无需随机数生成器即可实现聚合多个签名以及*m*-*n*多重签名，同时减少节点间的多余通信开销。该签名算法有两个基本概念：曲线哈希和曲线配对即双线性映射e函数[16-17]。

BLS签名算法与椭圆曲线数字签名算法（Elliptic Curve Digital Signature Algorithm，ECDSA）和Schnorr签名算法不同：在ECDSA和Schnorr签名中对消息求哈希的结果是数值；而在BLS签名算法中用新型散列函数对消息求哈希，并将得到的结果映射到椭圆曲线的一个点上[18]。

双线性映射e函数由六元组$\eta=(G_1, G_2, G_T, n, e, g_1, g_2)$组成，$G_1$和$G_2$是阶为$p$的加法循环群，$G_T$是具有相同阶的乘法循环群。$g_1$是加法群$G_1$的生成元，$g_2$是加法群$G_2$的生成元。$e$是双线性映射，需要进行曲线配对，则有$G_1 \times G_2 \rightarrow G_T$。设$P, Q$为曲线上的两个点，$\forall a, b \in \mathbf{Z}_p$，则满足以下性质：

$$e(P, Q+R)=e(P, Q)e(P, R) \tag{1}$$

$$e(aP, bQ)=e(P, abQ)=e(abP, Q)=e(P, Q)^{ab} \tag{2}$$

可见，配对函数对曲线上的运算满足分配律、交换律和结合律。

BLS签名过程如下：假设待签名的消息为m，私钥为k，公钥$P = k*G$；求签名S时，先对消息m求哈希后映射到椭圆曲线上，得到曲线哈希$H(m)$，再乘以私钥k即得$S = k*H(m)$。公钥P验证签名S，则有：

$$e(G,S)=e(G,k*H(m))=e(k*G,H(m))=e(P,H(m)) \quad (3)$$

本文采用BLS聚合签名技术对平行链共识算法进行优化。假设Leader节点要聚合N个超级节点的签名,由于平行链共识的特点是签名不同共识数据相同,因此共识交易设为M,对共识交易求曲线哈希可得$H(M)$,k_i为私钥,可得公钥$P_i=k_i*G$,聚合签名$S_i=k_i*H(M)$,则需验证$e(G,S_i)=e(P_i,H(M))$成立。验证过程如下:

$$\begin{aligned}&e(G,S_i)=e(G,S_1+S_2+\cdots+S_n)=\\&\quad e(G,k_1*H(M)+k_2*H(M)+\cdots+k_n*H(M))=\\&\quad e(G,(k_1+k_2+\cdots+k_n)*H(M))=e(P_i,H(M))\end{aligned} \quad (4)$$

4　平行链共识算法的优化方案

传统的平行链共识算法是平行链的每个共识节点都发送共识交易。优化方案是平行链的共识节点先内部发送共识交易并附上BLS签名数据;再由Leader节点整合共识交易并聚合交易签名;最后将新的共识交易发送到主链上验证,同时结合BLS聚合签名技术防止篡改的发生。这种共识算法既节省区块空间、节约手续费,还能够保证系统的安全。

4.1　系统初始化设置

设平行链为p_chain_1,平行链上的共识节点n需满足$n\geqslant 3f+1, f\in \mathbf{N}^+$。预配置的聚合签名算法为BLS签名算法,BLS签名是利用双线性对构造的一种短签名方案;预配置授权账户组,根据$x_1,x_2,\cdots,x_n$顺序依次设置平行链上节点的索引为$0,1,\cdots,n-1(n\geqslant 4)$。

4.2　共识过程

共识算法流程如图2所示。

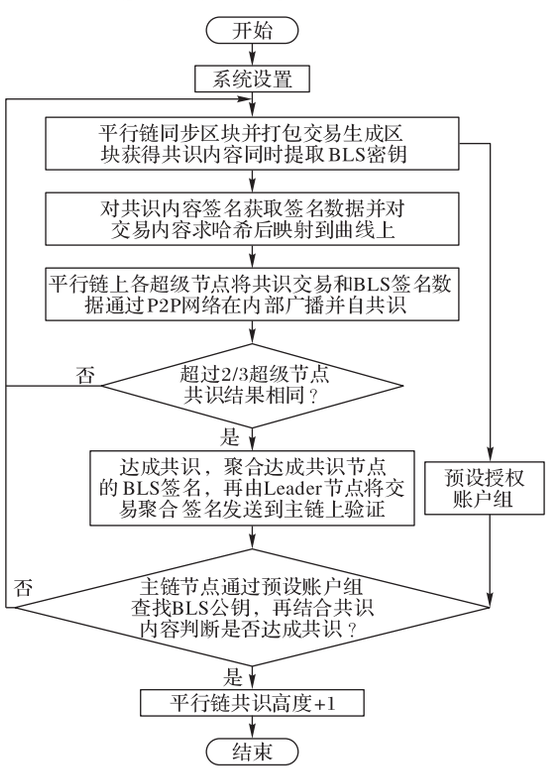

图2　优化后的平行链共识算法流程

Fig. 2　Flow chart of parallel chain consensus algorithm after optimization

1)系统设置。令q为大素数,加法循环群G_1和乘法循环群G_2为q阶,P是G_1的生成元,Q是G_2的生成元;双线性映射为$e=G_1\times G_2\rightarrow G_T$,定义散列函数$H_0:\{0,1\}^*\rightarrow G_1$。

2)生成共识内容并提取密钥。假设平行链待共识的区块为$p_block(100)$,其对应的主链区块为$block(200)$。以平行链P_1上的节点x_1为例说明,x_1从$block(200)$同步p_chain_1的各笔交易$Tx=(tx_1,tx_2,\cdots,tx_n)$,$x_1$打包$Tx$并生成$p_block_x_1$,计算$p_block_x_1$获得共识内容$msg(100)_x_1=\{block_hash,height,bitmap\}$。BLS签名需要的私钥为Secp265k1的1/2,采用Secp265k1私钥不断取$hash$直到满足BLS范围为止作为BLS私钥,令节点x_1的私钥为x_a,则x_1的公钥为$Pub_bls_x_1=k_a*P$。

3)签名。x_1根据BLS聚合签名算法对$msg(100)_x_1$进行签名获得签名数据$bls_(msg(100)_x_1)=k_a*H(msg(100)_x_1)$。

4)对交易内容求哈希并映射到曲线上。$H(msg(100)\_x_1)=H\left(block\_hash\|height\|bitmap\right)$。

5)预设授权账户组。$x_1,x_2,\cdots,x_n$按序排列并设置索引为$0,1,\cdots,n-1$,一一对应于BLS公钥$Pub_bls_x_1,Pub_bls_x_2,\cdots,Pub_bls_x_n$。

6)广播。x_1将$msg(100)_x_1+bls_(msg(100)_x_1)$广播给其他节点$x_2,x_3,\cdots,x_n$;同理$x_2,x_3,\cdots,x_n$节点也作如上运算操作。在$x_1,x_2,\cdots,x_n$都正常运行的情况下,对$x_1$而言已收到其他节点的共识内容和签名信息$\{msg(100)_x_2+bls_(msg(100)_x_2),\ msg(100)_x_3+bls_(msg(100)_x_3),\cdots,msg(100)_x_n+bls_(msg(100)_x_n)\}$。

7)自共识。各个节点的共识内容$msg(100)_x_1,msg(100)_x_2,\cdots,msg(100)_x_n$在内部进行自共识。若$x_1$不同,其余都相同,则$Num=n-1\geqslant 2/3*n(n\geqslant 4)$,达成共识,共识内容为$msg(100)$。

8)聚合签名。聚合达成共识的节点,其签名数据为$bls_(msg(100)_x_2)+bls_(msg(100)_x_3)+\cdots+bls_(msg(100)_x_n)$,在之前设置好的授权账户组中,令达成共识为1,未达成共识为0,即$bitmap=(0,1,1,1,\cdots,1)$。

9)聚合后的共识交易。$Tx100=\{msg(100),bls(msg(100\_x_2)+bls(msg(100\_x_3)+\cdots+bls(msg(100\_x_n)\}$,$bitmap=(0,1,1,1,\cdots,1)$。

10)验证聚合签名正确性。由Leader节点将$Tx100$发送至主链上,主链节点根据$bitmap=(0,1,1,1,\cdots,1)$授权账户组查找对应的BLS公钥可得到$\{Pub_bls_x_2,Pub_bls_x_3,\cdots,Pub_bls_x_n\}$,聚合签名公钥得$\{Pub_bls_x_2+Pub_bls_x_3+\cdots+Pub_bls_x_n=k_{x_2}*P+k_{x3}*P+\cdots+k_{x_n}*P\}$,结合共识内容$msg(100)$判断:

$$\begin{aligned}&E\left(\sum_{i=1}^{n}pub\_bls\_x_i,H(msg(h))\right)=\\&\quad E\left(P,\sum_{i=1}^{n}bls\_(msg(h)\_x_i)\right)\end{aligned} \quad (5)$$

是否正确。

下面给出聚合签名验证过程的正确性证明。

证明

$$E\left(\sum_{i=1}^{n} pub\_bls\_x_i, H(msg(h))\right) = E\left(\sum_{i=1}^{n} k_{x_i} * P, H(msg(h))\right) = E\left(P, \sum_{i=1}^{n} k_{x_i} * H(msg(h))\right) = \prod_{i=1}^{n} E(P, k_{x_i} * H(msg(h))) = E\left(P, \sum_{i=1}^{n} bls\_(msg(h)\_x_i)\right) \quad (6)$$

证毕。

因为式(5)的正确性被成功验证,所以平行链上的各个共识节点可以先互相广播发送共识交易达成共识,再由Leader节点将达成的共识交易内容与各节点的交易签名聚合后发送到主链上;主链根据*bitmap*在预配置的授权账户组中查找到平行链的各共识节点所对应的BLS公钥,对聚合签名验证通过,且公钥账户超过2/3共识阈值,则共识通过,同时证明该优化算法的可行性。

5 实验与结果分析

5.1 Leader节点选取实验

Chain33是一套支持共识、数据库、执行器等可插拔、易升级的区块链架构,本文基于Chain33底层架构搭建平行链架构,通过Go语言实现平行链共识算法的优化。实验环境配置如表4所示。

表4 实验环境配置

Tab. 4 Experimental environment configuration

配置信息	版本号
操作系统	Ubuntu18. 04
Chain33	1. 65. 2
CPU	Intel i5 2. 5 GHz 四核
内存	16 GB
Go	Go 1. 15. 2 Linux/amd64

本文实验中设置4个超级节点进行Leader节点的选取,分别为A、B、C、D,对应的索引号分别为0、1、2、3,分别对Leader节点选取过程中出现的3种情况进行分析。

1)normal:正常情况。

2)case1-1:Leader节点因本身故障在规定时间内未发出心跳消息,进行*offset*偏移至下一个节点。

3)case1-2:Leader节点因本身故障未在规定的时间内发出心跳消息并且进行*offset*偏移至下一个节点处时,下一个节点仍发生故障再次进行*offset*偏移操作。

假设当Leader节点为D索引号为3时未正常发出心跳消息,Matlab仿真图如图3所示。

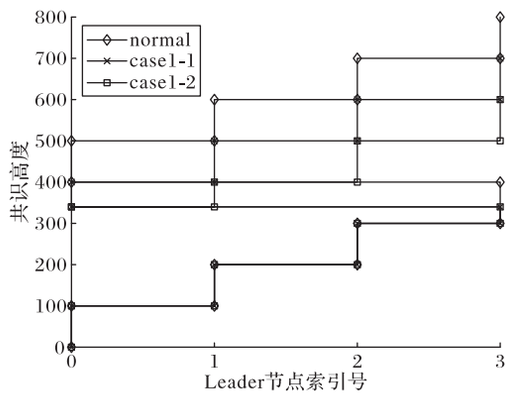

图3 Leader节点选取算法

Fig. 3 Leader node selection algorithm

分析图3可得,在normal正常情况下随着共识高度的增长每隔100共识高度发生一次Leader节点轮换操作,当Leader节点为D、索引号为3时,下一次轮换又回到节点为A、索引号为0处,一直重复这种循环;在case1-1情况下,Leader节点为D、索引号为3时发生故障,此时进行*offset*偏移至下一个节点为A、索引号为0处,此时A正常;在case1-2情况下,进行*offset*偏移至A,但此时A节点也出了故障,因此再次进行*offset*偏移至下个节点。由图3可知,无论发生哪种情况每个共识高度都只有唯一的一个Leader节点,保证每次只有一个Leader节点将共识交易发送到主链上且只收取一次交易手续费。

5.2 性能测试

在平行链上分别设置4、7、10、13、16、19、22个超级节点,从交易手续费、占用主链存储空间、交易吞吐量以及时延这4个方面进行测试,通过Matlab对优化前后的平行链共识算法仿真,如图4所示。

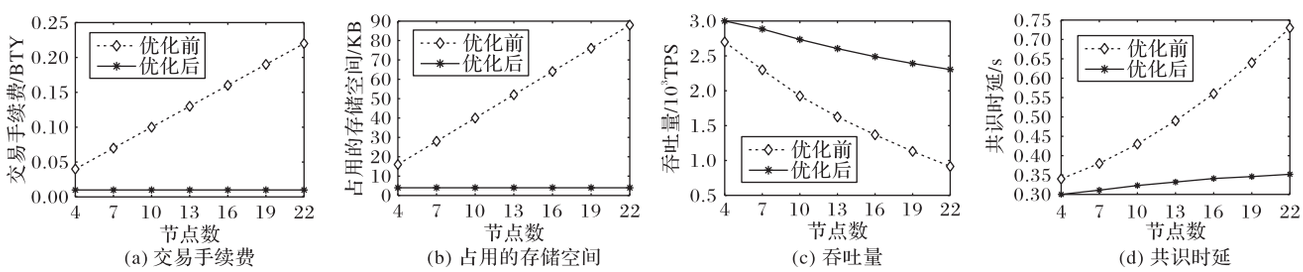

图4 优化前后不同指标对比

Fig. 4 Different indicators comparison before and after optimization

由图4(a)可知,优化前有n个超级节点发送n笔共识交易,收取n笔手续费0. 01n BTY;优化后仅由Leader节点发送一笔共识交易,因此无论有多少个超级节点,只收取一笔手续费0. 01 BTY。

由图4(b)可知,优化前n个共识节点会发送n笔共识交易到主链上参与共识,多笔共识交易会占用大量主链空间,n笔将占用4n KB的存储空间;优化后,最终由Leader节点发送一笔共识交易至主链,因此只占用主链区块空间4 KB。

由图4(c)可知,随着超级节点数量的增多吞吐量均呈下降趋势。优化前下降速度更快,当超级节点更多时对其影响更大;优化后吞吐量下降比较缓慢,当超级节点更多时,吞吐量可能仅有细微变化,最终曲线将接近水平状态,整体看来

TPS仍然会较高。

由图4(d)可知,随着超级节点数量的增多共识延时均呈增长趋势。优化前增长剧烈,超级节点数量越多发送到主链上的共识交易就越多,平行链上的节点进行二次共识的时间就越长,最终的共识时延也越长;而优化后虽然增加了BLS聚合签名的过程但该过程不到1 ms,对整个共识的过程影响不大,因此共识时延增长缓慢。

6　结语

共识算法是区块链的核心技术,本文首先介绍区块链主链的几种常见共识算法,包括PoW、PoS、DPoS、PBFT;然后介绍平行链+主链的架构,即平行链共享于主链的共识网络,一条主链可以附属多条平行链,能够实现交易并行执行,保障系统运行的稳定性和安全性;最后基于该架构对传统平行链共识算法的缺陷进行分析,并且针对该缺陷设计基于BLS聚合签名的平行链共识优化算法。该算法采用Leader节点选取算法和BLS聚合签名算法,在保障系统安全性的前提下能够有效解决多笔共识交易占用主链大量空间和浪费手续费的问题;同时在TPS吞吐量和共识时延方面优化后的平行链共识算法的性能也是优于优化前的平行链共识算法。这种优化只是平行链共识算法性能提升的一种尝试,未来还可以针对主链的共识算法作进一步改进,以此提升整个平行链架构的效率。

参考文献(References)

[1] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system[EB/OL]. [2021-05-13]. https://bitcoin. org/bitcoin. pdf.

[2] 朱建明,张沁楠,高胜. 区块链关键技术及其应用研究进展[J]. 太原理工大学学报, 2020, 51(3):321-330. (ZHU J M, ZHANG Q N, GAO S. Research progress of blockchain key technologies and their application[J]. Journal of Taiyuan University of Technology, 2020, 51(3):321-330.)

[3] WOOD G. Polkadot: vision for a heterogeneous multi-chain framework [EB/OL]. [2021-03-09]. https://polkadot. network/PolkaDotPaper. pdf.

[4] BUCHMAN E, KWON J. Cosmos: a network of distributed ledgers [EB/OL]. [2021-04-17]. https://github. com/cosmos/cosmos/blob/master/WHITEPAPER. md.

[5] 路爱同,赵阔,杨晶莹,等. 区块链跨链技术研究[J]. 信息网络安全, 2019, 19(8):83-90. (LU A T, ZHAO K, YANG J Y, et al. Research on cross-chain technology of blockchain [J]. Netinfo Security, 2019, 19(8):83-90.)

[6] 郑敏,王虹,刘洪,等. 区块链共识算法研究综述[J]. 信息网络安全, 2019, 19(7):8-24. (ZHENG M, WANG H, LIU H, et al. Survey on consensus algorithms of blockchain[J]. Netinfo Security, 2019, 19(7):8-24.)

[7] JAKOBSSON M, JUELS A. Proofs of work and bread pudding protocols [M]// Secure Information Networks. Boston: Springer, 1999: 258-272.

[8] BUTERIN V. A next-generation smart contract and decentralized application platform[EB/OL]. [2021-10-20]. http://www. fintech. academy/wp-content/uploads/2016/06/EthereumWhitePaper. pdf.

[9] Bitshares. Delegated proof of stake [EB/OL]. [2020-10-20]. https://how. bitshares. works/en/master/technology/dpos. html.

[10] DANTHEMAN. DPOS consensus algorithm — the missing white paper [EB/OL]. [2020-09-10]. https://steemit. com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper.

[11] CASTRO M, LISKOV B. Practical Byzantine fault tolerance[C]// Proceedings of the 3rd Symposium on Operating Systems Design and Implementation. Berkeley: USENIX Association, 1999: 173-186.

[12] ONGARO D, OUSTERHOUTJ K. In search of an understandable consensus algorithm [C]// Proceedings of the 2014 USENIX Annual Technical Conference. Berkeley: USENIX Association, 2014: 305-319.

[13] 隋源,汪卫,邓雪. 一种面向区块链的链下数据库高吞吐量可验证查询方法[J]. 小型微型计算机系统, 2021, 42(6):1304-1312. (SUI Y, WANG W, DENG X. High throughput verifiable query method for blockchain-oriented off-chain database [J]. Journal of Chinese Computer Systems, 2021, 42(6):1304-1312.)

[14] 王日宏,张立锋,周航,等. 一种结合BLS签名的可拜占庭容错Raft算法[J]. 应用科学学报, 2020, 38(1):93-104. (WANG R H, ZHANG L F, ZHOU H, et al. A Byzantine fault tolerance Raft algorithm combines with BLS signature [J]. Journal of Applied Sciences, 2020, 38(1):93-104.)

[15] BONEH D, LYNN B, SHACHAM H. Short signatures from the Weil pairing [C]// Proceedings of the 2001 International Conference on the Theory and Application of Cryptology and Information Security, LNCS 2248. Berlin: Springer, 2001: 514-532.

[16] CHEN L Q. A DAA scheme using batch proof and verification [C]// Proceedings of the 2010 International Conference on Trust and Trustworthy Computing, LNCS 6101. Berlin: Springer, 2010: 166-180.

[17] OKAMOTO T. Cryptography based on bilinear maps [C]// Proceedings of the 2006 International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, LNCS 3857. Berlin: Springer, 2006: 35-50.

[18] GROTH J. Fully anonymous group signatures without random oracles [C]// Proceedings of the 2007 International Conference on the Theory and Application of Cryptology and Information Security. Cham: Springer, 2007: 164-180.

This work is partially supported by National Natural Science Foundation of China (61901182), 2020 Province Ministry Co-construction Fund (605-52520005).

LIU Qi, born in 1995, M. S. candidate. Her research interests include blockchain.

GUO Rongxin, born in 1980, M. S., senior experimentalist. His research interests include blockchain.

JIANG Wenxian, born in 1974, Ph. D. candidate, associate professor. His research interests include blockchain, information network security.

MA Dengji, born in 1981, M. S. His research interests include blockchain.