计算机科学
COMPUTER SCIENCE

利用状态归约的分片负载均衡方法

陈静，李志淮，高冬雪，李敏

引用本文

陈静，李志淮，高冬雪，李敏. 利用状态归约的分片负载均衡方法[J]. 计算机科学, 2022, 49(11): 302-308.

CHEN Jing, LI Zhi-huai, GAO Dong-xue, LI Min. Shard Load Balancing Method Using State Reduction[J]. Computer Science, 2022, 49(11): 302-308.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210800109

利用状态归约的分片负载均衡方法

陈 静 李志淮 高冬雪 李 敏
大连海事大学信息科学技术学院 辽宁 大连 116026

摘 要 分片技术是解决区块链可扩展性问题的核心技术之一。当将P2P网络中的交易按规则汇集到既定分片,且验证节点随机均衡分配到各分片后,由于个别分片的交易验证负载可能远远超过平均负载,因此该分片内的交易可能会拥堵。为了解决分片间的负载不均衡,提出了利用状态归约的分片负载均衡方法。首先,给出了状态归约模型,允许性能高的节点存储更多的相邻状态,并据此将节点性能做出粗略分类;然后,根据每一时隙的交易验证情况将未经验证的交易作为剩余负载,并将其作为调整下一时隙分片内验证能力的依据;最后,对节点进行评分、等级划分,根据剩余负载、共识验证节点集合的平均评分,给出节点选取策略,合理且随机分配节点,并对高负载分片的剩余负载向上归约。实验结果表明,利用状态归约的分片负载均衡方法在不降低单个分片的交易验证率的基础上,有效处理了个别分片的异常过载。

关键词:区块链;分片;状态归约;负载均衡;状态同步;PBFT
中图法分类号 TP311

Shard Load Balancing Method Using State Reduction

CHEN Jing,LI Zhi-huai,GAO Dong-xue and LI Min
School of Information Science and Technology,Dalian Maritime University,Dalian,Liaoning 116026,China

Abstract Sharding technology is one of the core technologies to solve the scalability problem of blockchain. When transactions in the P2P network are aggregated into established shards according to the rules,and the verification nodes are randomly and evenly distributed to each shard,the transactions in this shard may be congested because the transaction verification load of individual shard may far exceed the average load. In order to solve the load imbalance between shards,a shard load balancing method using state reduction is proposed. Firstly,the state reduction model is given,which allows high-performance nodes to store more adjacent states,and the node performance is roughly classified according to this model. Secondly,according to the transaction verification situation in each timeslot,unverified transactions are taken as the remaining load,which is used as the basis for adjusting the verification ability in the next timeslot. Finally,the nodes are scored and graded,and the node selection strategy is given based on the remaining load and the average score of the consensus verification node set. Nodes are allocated reasonably and randomly,and the remaining loads of high load fragments are reduced upward. Experimental verification shows that the shard load balancing method using state reduction effectively balances the unusual load of one shard without reducing the transaction verification rate of a single shard.

Keywords Blockchain,Sharding,State reduction,Load balancing,State synchronization,PBFT

1 引言

2014年,去中心化模式下的比特币[1]已平稳运行5年,其底层的核心技术——区块链技术得到了越来越多的关注。区块链是一种分布式账本技术,它基于密码学原理,利用链式区块结构来存储数据,利用节点间共识算法来验证交易,具有去中心化、数据公开透明、不可篡改等特点。区块链技术可以转变当前人类社会的信任模式,不再依赖于中心化的信用机构,能够有效提高工作效率。随着区块链技术的发展,区块链的容量局限性限制了其大规模应用的落地,因此区块链可扩展性成为了目前亟待解决的核心问题,且对区块链可扩展性[2-3]问题的研究越来越多。

当前,对区块链可扩展性的研究从Layer 0,Layer 1,Layer 2展开。Layer 0扩容主要指修改和优化网络底层协议,加快区块在网络中的传输速度,主要的方案有区块链分发网络(Blockchain Distribution Network,BDN)、组播锁定组、快速UDP互联网连接(Quick UDP Internet Connections,QUIC)协议[4]等。Layer 1扩容是对区块链自身进行修改,主要的方案有隔离见证、分片技术[5-6]、DAG技术[7]等。Layer 2扩容指不改变区块链自身,将复杂的交易转移到链下,以此减轻区块链系统的负担,主要的方案有侧链[8]、状态通道[9]、十倍协议、Roll up等。Layer 1扩容方案和Layer 2扩容方案都有各自的优势与不足,本文选取有代表性的Layer 1扩容方案中的DAG(Directed Acyclic Graph)技术、分片技术和

到稿日期:2021-08-12 返修日期:2022-03-02
通信作者:李志淮(qhlee@dlmu.edu.cn)

Layer 2 扩容方案中的侧链、状态通道进行分析。

DAG 技术在理论上验证了节点越多，交易的验证速度越快，但无法保证其安全性和一致性。分片技术能够提高并行处理事务的能力，节省系统资源。侧链技术有较高的独立性和灵活性，创建和维护侧链需要的成本较高，且易遭受攻击。具有较好的隐私性和实时性的状态通道在理论上可以无限扩展，与此同时，它也存在一定的安全隐患。针对以上分析，分片是目前解决区块链扩容问题的核心和基础方案。

分片的主要思想是“分而治之”，它将系统中的节点分成多个独立的分片，通过各个分片并行处理事务来提高系统的吞吐量。分片通常被划分为网络分片、交易分片和状态分片。其中，网络分片是按照规则将节点划分到某个分片，网络分片是其他两种分片的基础。交易分片是将全网交易按照映射规则分配到不同分片进行验证和打包，状态分片是将完整的账本信息存储到各个分片，每个分片维护区块链的部分账本信息，每个节点不再存储完整的区块链状态信息。将分片技术作为区块链扩容方案的探索有很多，如 Elastico[10]、Rapidchain[11]、以太坊[12]等。

在区块链系统中，去中心化、安全性和可扩展性三者不可能同时得到满足，因此分片尽管可以有效提高区块链处理事务的能力，但在安全性上和去中心化程度上可能有所牺牲，同时带来了新的挑战，如分片内的 PoW[13](Proof of Work)共识中的 51%攻击问题[14]、PBFT(Byzantine Fault Tolerance)[15]共识的节点数量限制、女巫攻击问题、节点选择问题[16]、分片间的跨分片交易处理问题[17]、负载不均衡问题等。分片内验证和处理交易的速度与分片内节点的性能息息相关，当某个分片内节点的性能较弱时，性能差的节点可能无法在规定时间内完成对交易的验证或者验证时间增加，分配到该分片的交易可能得不到及时验证，造成该分片内的交易累积而出现拥堵，最终在分片间表现为负载不均衡，影响系统处理事务的能力。

目前基于分片的区块链协议在按既定规则将交易映射到分片时并未考虑分片内各节点的能力差异，节点依靠随机分配，因此可能会出现某个分片负载大但节点性能不一定好的情况；同时，某些负载小的分片的节点性能却较高。在各个分片性能差距过大的情况下处理交易，不仅会造成资源浪费，还不利于提升区块链的整体性能。本文主要针对采用分片技术扩容后产生的分片间负载不均衡问题进行描述和分析。

2　负载不均衡问题描述与分析

2.1　负载不均衡问题描述

负载指系统或网络中的工作任务量或请求数量，在区块链系统中，负载主要指系统内需要验证的交易。本文研究的负载不均衡问题是围绕分片内的剩余负载展开的，剩余负载指在某个时间段内提交到分片内，但未经验证的交易。区块链系统中的均衡指根据剩余负载大小对分片内的节点进行合理且随机的分配，使负载大的分片中的剩余负载得到及时处理的过程。分片技术将系统内的节点划分为多个节点集合，各个集合并行处理分配到的交易，分片技术在理论上可以提升系统的吞吐量，但每个分片内处理交易的情况不同。节点性能主要指节点的存储能力和处理能力，当分片内参与验证的节点性能较弱时，可能不足以处理目前分配到该分片内的交易，一部分交易得不到及时验证。这些未验证的交易成为分片内的剩余负载，每个分片内参与验证的节点性能不同，剩余负载也会不同，从而出现分片间负载不均衡的现象。

2.2　负载不均衡问题研究现状

Zilliqa 项目使用改进的 PBFT 作为共识算法，已经实现了网络分片和交易分片，没有实现状态分片。在对交易进行分片时，依据事务分类决定事务的处理过程，其中用户之间的交易为第一类事务，用户调用单一智能合约为第二类事务，除上述两类事务外的事务为第三类事务。第一类和第二类事务由普通用户负责处理，第三类事务由普通用户处理后还需交由 DS 委员会处理。具体的交易分配取决于发送者地址的最右边$\lfloor \log_2 l \rfloor+1$位，其中，$l$ 为分片数目，且满足$\lfloor \log_2 l \rfloor+1 \leqslant 160$。这种交易分配规则可能会引起负载不均衡，且随着用户的增加，DS 委员会需处理的事务增多，同样会出现负载过大的情况。QuarkChain[18]由一条根链和多条子链组成，现阶段一条子链对应一个分片，子链处理交易，根链负责确认。该方案实现了状态分片，设计了玻色子共识，支持多种共识算法。但对于单点过热问题而言，其主要靠市场调节，系统未做调整，因此也未在系统层面解决负载不均衡问题。由上述分析可知，区块链现有项目对于负载不均衡问题没有明确的解决方案。

《分片负载的多轮均衡化验证方案》[19]根据节点验证交易的能力对节点进行评级，计算式如式(1)所示：

$$P(i)=\frac{Count(TX_i)}{\sum_{j=1}^{N} T_j} \tag{1}$$

其中，$P(i)$表示节点 i 在生成微区块的过程中 1s 内可验证交易的数量，具体计算方法为计算所验证的全部交易数与验证交易花费的时间的比值。文中对于分片内未确认的交易，并不是将其丢弃，而是将其作为剩余负载分配新的节点进行验证，在新一轮验证中对负载过大的分片分配性能较高的节点，这样可以提升区块链系统的交易吞吐量。该方案进行负载均衡时是在网络分片和交易分片的基础上进行的，并未考虑状态分片。在这种情况下，节点可以任意选择，每个节点存储所有状态，节点压力较大。

2.3　负载不均衡问题分析

图 1 为对区块链系统网络分片的示意图。图 1 中，深色正方形表示自身性能相对较弱的节点，浅色圆形表示性能较强的节点。分片前，在进行 PBFT 共识时，整个区块链系统中的共识验证节点共同验证交易，根据 PBFT 共识算法的特点，一次共识验证中，当拜占庭节点少于验证节点总数的 1/3 时，不会对共识结果造成影响。性能较差的节点可能部分表现为拜占庭节点，如因为网络时延造成消息传递不及时、节点在验证过程中计算缓慢不能给出意见等。在采用分片技术前，假设区块链系统中性能较差的节点所占比重较小，远少于 1/3，在通信正常的情况下，不会影响系统对交易的验证。在分片后，性能较差的节点在分片 1 和分片 2 中的比重较小，即使全部表现为拜占庭节点，对共识结果的影响相对较小。但在分片 3 中，性能较差的节点数量较多，在验证交易时，性能较差

的验证节点在给定时间内无法完成交易验证，导致分片达成共识的速度较慢，甚至无法达成共识，最终该分片可能会出现交易拥堵的现象，使该分片的剩余负载过大，在此种分片方式下，分片间负载不均衡现象明显。

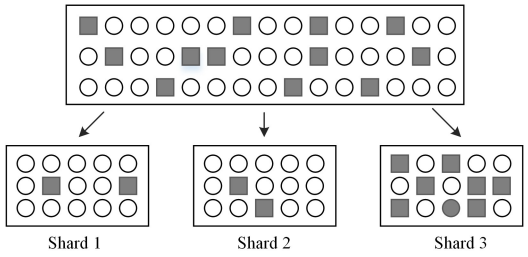


图 1 网络分片示意图

Fig. 1 Schematic diagram of network sharding

针对上述问题，假设系统内的节点数目为 N，其中在共识过程中表现为拜占庭节点的低性能节点数目占系统节点总数的比例为 f，分片内验证节点数目为 N_s，那么一笔交易能够共识失效的概率为：

$$P_{\text{consensus}}=1-\sum_{i=0}^{N_s/3}\frac{\binom{N\cdot f}{i}\binom{N-N\cdot f}{N_s-i}}{\binom{N}{N_s}} \tag{2}$$

其中，$N=1000$，$N_s=100$ 时，在共识过程中表现为拜占庭节点的低性能节点占系统节点总数的比例与分片内共识失效的概率关系如图 2 所示。

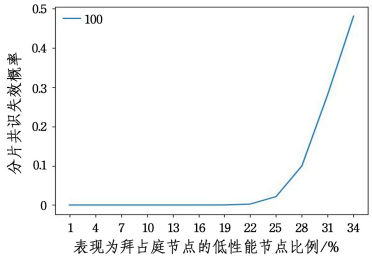


图 2 低性能节点对 PBFT 共识的影响

Fig. 2 Impact of low-performance nodes on PBFT consensus

从图 2 中可以看出，随着表现为拜占庭节点的低性能节点数目占系统节点总数的增加，共识失效的概率不断增加，当 $f=0.33$ 时，$P_{\text{consensus}}=0.48$，此时性能差的验证节点对共识结果的影响较大。

此外，交易分配不均也是造成分片内负载不均衡的原因。为了防止双花问题，交易按交易输入方地址映射到既定分片，分片内的交易数量不同。在进行负载均衡时，考虑到 P2P 环境没有中心化的服务器进行协调，很难预先计算下一时隙需要处理的交易数量，因此无法事先采取措施来验证更多交易。在考虑状态分片的情况下，对分片负载进行均衡时不能随意调度节点，否则共识验证节点在验证交易时，部分共识验证节点需要先获取相关状态才能进行交易验证，验证交易的时间会延长。但每一次验证过程需要的时间都比较短，如以太坊进行交易验证的每个时隙(slot)大约为 15 s，验证时间较短，不支持无状态的节点在获得状态后参与验证。根据上述分析，针对分片负载不均衡问题，只能在获得分片内剩余负载的情况下，针对剩余负载大的分片，采取提高有相关状态的共识验证节点集合的整体性能的方法来验证更多交易，以减少分片负载。

本文针对的主要问题是在状态分片下，共识验证节点集合不足以处理分配到既定分片的交易，分片内产生交易拥堵，造成的分片负载不均衡的现象。本文针对上述问题，提出了利用状态归约的分片负载均衡模型，尝试在状态归约的支持下，更好地处理分片内的剩余负载，实现分片负载均衡。

3 利用状态归约的分片负载均衡方法

在区块链系统进行分片后，各个分片位于状态归约模型的最底层，节点首先需要进行状态同步，完成状态归约模型的初始化过程。在节点根据自身性能动态调整一段时间后，状态归约模型基本形成。模型的各个分片内的集合中的节点数目达到最低要求后，分片开始交易验证。待一轮共识结束后，根据共识验证节点的表现对节点进行评分，并记入历史。同时按负载计算模型对剩余负载进行量化，根据剩余负载情况按照节点选取策略随机选取节点参与下一时隙的验证。当分片内出现高负载时，考虑选择性能更高的上层分片中节点进行验证，缓解分片内节点的压力。利用状态归约的负载均衡方法的流程如图 3 所示。

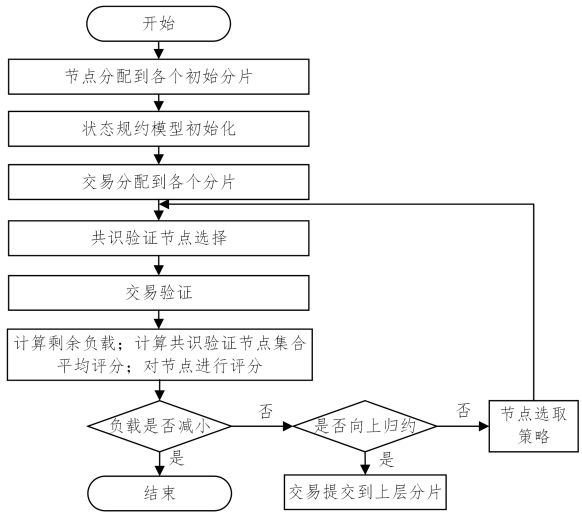


图 3 利用状态归约的负载均衡方法的流程图

Fig. 3 Flow chart of load balancing method using state reduction

(1)为了更精准地指明分片，将区块链系统进行网络分片后形成的分片称为初始分片。状态归约模型初始化指状态归约模型中的分片内节点在分片间或者集合之间进行相互转换。交易分片时按照交易的地址将交易映射到各个分片。

(2)在负载均衡的情况下，共识验证节点的选取是随机的，各个等级没有数目限制。

(3)判断负载是否减少，若负载减少，则结束，否则判断是否对交易进行向上归约。

(4)交易向上归约指将剩余负载提交到上层分片后与该分片下一时隙客户端提交的新交易一起验证。

(5)不向上归约的交易将根据不同的情况选择数量不同的各个等级的节点，并将其添加到共识验证节点集合以验证交易，共识验证节点集合验证的交易包括当前时隙提交的

交易和上一时隙的剩余负载。

3.1　状态归约模型

3.1.1　状态归约模型介绍

状态归约是依据节点在状态上的能力尝试用哈夫曼编码的思想来实现节点性能的划分,用二叉树结点到根结点的路径长度表示该结点中验证节点的性能,结点距离根结点的路径越短,节点的性能越好。

状态归约模型是在考虑状态分片的情况下提出的,状态分片主要是为了缓解节点的存储压力。该模型依据节点的性能差异,在状态条件的约束下,对节点进行动态调整,形成满二叉树结构。节点的性能主要指节点存储状态的能力和处理验证交易的能力。初始时,验证节点通过随机算法被随机分配到各个分片,这些分片被称为初始分片。之后依据节点自身性能和初始分片的约束,节点可以选择是否同步相邻分片的状态,决定是否向上归约,最终被划分为不同类别,每个类别节点的性能相近,最终形成一个分片,这种分片被称为合成分片。每个分片只存储部分状态信息,缓解了节点的存储压力,且性能相近的节点位于同一分片内,降低了因节点性能参差不齐导致的验证交易的时延。

状态归约模型采用的是满二叉树结构,每个结点表示一个分片,分片中的节点属于不同的集合,集合及其之间的转化关系将在 3.1.2 节详细介绍。在状态归约模型中,验证节点根据随机算法分配到各个分片后,每个初始分片作为一棵满二叉树的叶子结点。根据满二叉树的性质,分片数目设定为 2^k,$k=0,1,2,3\cdots$。首先,对满二叉树结点进行编号。

如图 4 所示,本文设定满二叉树的根结点编号为 1,按层次遍历的顺序依次编号,以分片数目为 2^k,$k=0,1,2,3,\cdots$进行分析。区块链中的节点随机分配后,初始分片为满二叉树的叶子结点,编号分别为 2^k,2^k+1,…,$2^{k+1}-1$,分片中的节点首先进入分片内待同步节点序列,进行状态同步,完成状态同步的节点进入分片内就绪节点队列,被随机算法选中后才能进入分片内共识节点验证集合,参与交易验证。

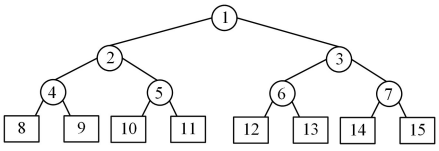


图 4　状态归约模型示意图

Fig. 4　Schematic diagram of state reduction model

在状态归约模型中,初始分片位于满二叉树的最底层,在完成状态同步后,节点可以根据自身性能决定是否向上归约,向上归约的节点将进入上层。节点向上归约指若节点有能力同步满二叉树中兄弟结点存储的状态,将进入对应的上层分片内待同步节点序列。即若节点在分片 s,那么该节点将归约到$\lfloor \log_2 s \rfloor$层的分片$\lfloor s/2 \rfloor$的分片内待同步节点序列,进行状态同步。因此,在状态归约模型中,从最底层的初始分片到最上层的合成分片,分片内节点的性能逐渐增强,且同一分片内节点的性能相近。下文中的分片是初始分片和合成分片的总称。3.1.2 节对状态归约模型中各个分片内状态节点集合及其变迁进行了详细介绍。

3.1.2　状态节点集合及其转化关系

在状态归约模型中,每个分片包括分片内待同步节点序列、分片内就绪节点队列、分片内共识验证节点集合、全局待选择节点队列、黑名单和局外节点集合。

(1)分片内待同步节点序列。该序列是待同步节点的集合,节点存储的验证交易需要的状态信息不完整,需要进行状态同步,在状态同步过程中,根据节点同步状态信息的时间和节点对自身性能的评价,给出了节点性能的合理评分。

(2)分片内就绪节点队列。该队列是完成状态同步的节点的集合,集合中的节点称为就绪节点,每个节点根据性能和验证交易的表现,都有相应的评分,队列按照评分由大到小排列。仅此集合内的节点有资格参与交易验证。

(3)分片内共识验证节点集合。该集合指通过 VRF 算法从分片内就绪节点队列中选中参与下一时隙交易验证的节点的集合,集合中的节点被称为共识验证节点。

(4)全局待选择节点队列。该队列是分片内就绪节点队列进行末位淘汰后,被替换出且想要继续参与交易验证的节点的集合,该集合中的节点存在静默期,静默期结束后节点将按 VRF 算法重新分配,参与验证。

(5)黑名单。黑名单是淘汰的节点中作恶节点的集合,作恶主要指节点在验证过程中被举报有不诚实的行为或者被证明参与合谋,该集合中的节点没有参与验证的资格。

(6)局外节点集合。该集合指不参与验证的节点集合。

节点在上述几种集合之间可以进行转换,如图 5 所示。

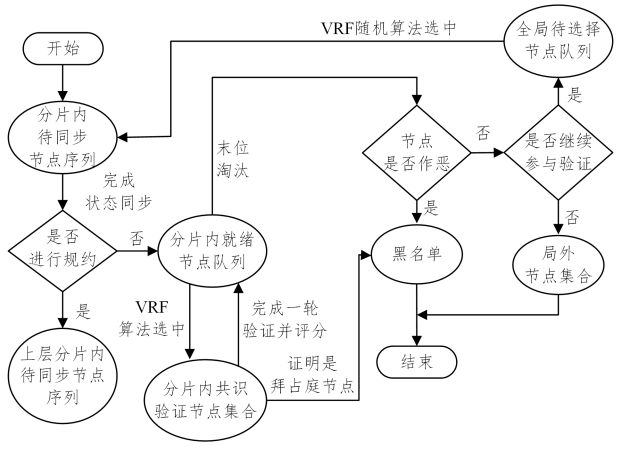


图 5　状态集合变迁图

Fig. 5　Diagram of state set transition

图 5 中,待同步节点在完成节点所在分片的状态同步后选择是否向上归约,若进行归约则进入上层分片内待同步节点序列,否则依据分片内就绪节点的情况等待或直接进入分片内就绪节点队列。就绪节点被随机函数选中后,进入分片内共识验证节点集合,参与下一时隙的交易验证。共识验证节点在完成一轮验证后,根据其在共识过程中的表现,若其行为被证明是拜占庭节点,则直接进入黑名单;否则节点的表现将计入历史数据,节点进入分片内就绪节点队列。

在对分片内就绪节点队列进行淘汰时,首先要保证有足够的节点用于处理交易,因此设定分片内就绪节点队列中的节点数至少为 80 个,至多为 150 个,当节点数少于 80 时,分片将不处理交易。当节点数少于 150 时,分片内待同步节点

序列中的节点在完成状态同步后直接进入分片内就绪节点队列中。若队列中的节点数为150，则实行末位淘汰，末位淘汰将替换分片内就绪节点队列中评分最低的节点。淘汰的节点若存在作恶行为，则将其加入黑名单，否则根据节点是否继续参与验证，将其放入全局待选择节点队列或局外节点集合。节点在集合间的变迁图如图4所示。

3.2 负载计算模型

3.2.1 负载来源分析

区块链系统在分片后，每个分片内的共识验证节点集合独立运行PBFT共识算法进行交易验证。认为剩余负载主要在以下3种情况下产生。

第一种情况是在交易传输的过程中，由于传输时延较大，交易得不到验证而成为分片内的剩余负载。在将交易传给共识验证节点的过程中，有些交易因网络原因未能在一个时隙的时间内到达，在这段时间内这些交易并没有得到验证，这些未被验证的交易成为了剩余负载。第二种情况是分片内共识验证节点集合的性能不佳，收到交易后未能在一个时隙内完成全部交易的验证，剩下的这些交易成为了该分片的剩余负载。第三种情况是在主节点打包过程中造成的，当交易得到超过2/3的共识验证节点确认后，由主节点打包，但PBFT没有机制确保主节点必须打包每个达成共识的交易，因此可能存在部分达成共识的交易因未被打包而成为剩余负载。

3.2.2 负载计算模型

对上述3种情况进行分析。第一种情况是每时每刻都存在的，由于区块链系统是在P2P环境下展开的，想要计算此情况下的全部未经验证的交易数是不现实的，因此本文不考虑这种情况下产生的剩余负载。在一个时隙内，由共识验证节点收集到的客户端发送的需要验证的交易的总数能够通过计算得出，主节点可以明确验证过的交易的数量，因此，在通信开销不需要额外增加的情况下，可以得到后两种情况下成为剩余负载的交易总数，本文基于后两种情况下计算得到的剩余负载进行研究。分片 s 在时隙 t 未经验证的交易数量 $Surplustr_t(s)$ 可以表示为：

$$Surplustr_t(s)=Sumtr_t(s)-Verificationtr_t(s) \tag{3}$$

其中，$Sumtr_t(s)$ 表示分片 s 的共识验证节点集合在时隙 t 收集到的交易总数，$Verificationtr_t(s)$ 表示分片 s 的共识验证节点集合在时隙 t 验证过的交易数。本文用未经验证的交易数表示分片剩余负载。

3.3 分片负载均衡模型

分片负载均衡模型的主要思想是根据分片的剩余负载情况，对负载进行处理。当分片内有剩余负载时，可以考虑将剩余负载提交给上层分片进行处理，即向上归约，或者由分片内性能更优的节点处理。为了避免上层分片出现因向上归约的交易过多而产生拥堵，本文方案不是直接将剩余负载向上归约，而是根据评分选取性能优的共识验证节点来验证交易，若在连续几个时隙内剩余负载依然没有减少，则认为该分片可能出现交易拥堵，考虑对交易向上归约，以减少该分片的负载。

3.3.1 节点评分策略

针对状态分片下分片负载不均衡问题，依据评分策略的评分，利用淘汰机制筛选分片内性能较强且表现较优的节点，提高共识验证节点集合处理交易的速度，减轻分片负载。

在对节点进行评价时，节点性能是评分的重要因素，但是由节点本身提供性能参数并不合理，节点可能提供假数据。因此，本文设定在节点提供性能评价后，根据节点同步状态的速度，对其给出的性能评分进行校正。假设节点 i 提供的性能评分为 s_{i01} ($0\leqslant s_{i01}\leqslant 100$)，根据节点状态同步速度评分为 s_{i02} ($0\leqslant s_{i02}\leqslant 100$)，节点的性能评分 s_{i0} 如式(4)所示：

$$s_{i0}=k\cdot s_{i01}+(1-k)s_{i02} \tag{4}$$

其中，k 为权重系数，如式(5)所示：

$$k=\begin{cases}0.8, & s_{i01}-s_{i02}\leqslant 10\\ 0.5, & \text{其他}\end{cases} \tag{5}$$

在节点完成状态同步后，节点性能的评分根据式(4)和式(5)就已经确定。

除节点性能外，节点的表现也是影响节点评分的重要因素。主要的节点评价指标和具体的分值如表1所列。

表1 评价指标表

Table 1 Evaluation index

序号	节点评价指标	分数
1	参与共识且达成共识	−2
2	参与共识不作为	−5
3	参与共识作恶	−30
4	证明参与合谋	−40
5	提供欺诈证明验证正确	20
6	提供欺诈证明	1

记表1中各项分数为 s_q，$q=1,2,\cdots,6$，r_{itq} 表示节点 i 在时隙 t 进行验证时是否表现出序号为 q 的行为，具体如式(6)所示：

$$r_{itq}=\begin{cases}1, & \text{节点 } i \text{ 在第 } t \text{ 轮进行序号为 } q \text{ 的活动}\\ 0, & \text{其他}\end{cases} \tag{6}$$

节点的所有行为都能在历史记录中查询到，则节点 i 在时隙 t 验证完成后的评分 $Score_i(t)$ 如式(7)所示：

$$Score_i(t)=s_{i0}+\sum_{j=0}^{t}\sum_{q=1}^{6}r_{ijq}s_q \tag{7}$$

根据式(7)的结果对节点进行等级划分，节点 i 在时隙 t 的评分 $Score_i(t)$ 与等级的对应关系如表2所列。

表2 节点等级划分表

Table 2 Node classification

分数范围	等级
$Score_i(t)>90$	1
$80<Score_i(t)\leqslant 90$	2
$70<Score_i(t)\leqslant 80$	3
$60\leqslant Score_i(t)\leqslant 70$	4
$Score_i(t)<60$	5

3.3.2 负载均衡策略

负载均衡策略主要是依据时隙 t 与时隙 $t-1$ 的剩余负载进行比较，当时隙 t 的剩余负载更多时，说明当前的分片内共识验证节点集合不足以及时验证该时隙的交易，此时需要适当提高分片内共识验证节点集合的性能，才可能在下一时隙验证由剩余负载和新提交交易组成的交易集合时实现更快的速度，以减少下一时隙的剩余负载，因此选取2/3节点评分不低于当前时隙平均评分的节点进行验证，以保证PBFT验证交易达成的速度。当分片内共识验证节点集合的性能已经很强但剩余负载仍然在增加时，这种情况很有可能已经出现交易拥堵，在这种情况下，为了缓解该分片的压力，可以考虑将交易向上归约，若上层分片产生交易拥堵或者已经在最高层，则将剩余负载单独验证。

假设分片内就绪节点队列中的节点超过限制的最低要

求，能够开始处理交易，根据负载计算模型可以确定每一轮各个分片的剩余负载。分片 s 在时隙 t 的剩余负载为 $Surplustr_t(s)$，本文将 $Surplustr_t(s)$ 与在 $t+1$ 时刻新提交的交易一起进行验证。分片内共识验证节点集合 $V_{sv}(t)[SS(t),TR(t)]$ 表示在时隙 t 状态集合为 $SS(t)$，共识验证节点集合对交易集合 $TR(t)$ 进行了一轮验证，该集合的节点平均评分 $Pervsv_t(s)$ 如式(8)所示：

$$Pervsv_t(s)=\frac{\sum_{i\in V_{sv}(t)[SS(t),TR(t)]}Score_i(t-1)}{N_{sv}} \tag{8}$$

验证结束后，计算剩余负载 $Surplustr_t(s)$，将其与上一时隙的剩余负载 $Surplustr_{t-1}(s)$ 进行比较，若 $Surplustr_t(s)>Surplustr_{t-1}(s)$，那么需要评分更高的节点进行验证。

本文假设各等级节点足够，根据共识验证节点集合中的节点平均评分 $Pervsv_t(s)$ 可以判断该评分所属的等级 sj，参与验证的共识验证节点的数量为 N_{sv}，各等级选择的节点数量可分情况讨论。

当 $sj\neq 1$ 时，各等级的节点数量如式(9)所示：

$$Schedule_s(j)=\begin{cases}\frac{2}{3j}N_{sv}, & j\geqslant sj\\ \frac{1}{3(5-j)}N_{sv}, & \text{其他}\end{cases} \tag{9}$$

当 $sj=1$ 时，判断 $Surplustr_t(s)>Surplustr_{t-1}(s)$ 连续出现该情况的次数 r_l，若 $r_l>r_{l\max}$，则认为分片出现交易拥堵，此时若存在上层分片，且上层分片未出现交易拥堵，那么将剩余负载提交到上一层的相应分片中进行交易验证；否则，在本分片内随机选择一个分片内的共识验证节点集合对剩余负载进行单独验证。若下一时隙的共识结束后仍存在剩余负载，则将与下一时隙验证新提交交易的剩余负载合并，进行后续的负载均衡。若 $r_l\leqslant r_{l\min}$，则本轮剩余负载将与下一轮新提交的交易一起验证，各等级的节点数量如式(10)所示：

$$Schedule_s(j)=\begin{cases}\frac{2}{3}N_{sv}, & j=1\\ 0, & j=5\\ \frac{1}{3(4-j)}N_{sv}, & \text{其他}\end{cases} \tag{10}$$

在对节点进行选择时，需要满足以下要求。首先，选取规则具有随机性，每个节点被选中的机会均等。其次，选取规则具有不可预测性，不能推算出下一次使用该规则选中的节点，防止拜占庭节点据此对节点进行攻击以达到拜占庭节点当选的目的，从而引发安全问题。此外，选取规则应具有可验证性，以保证选中的验证节点确实有资格参与验证。基于以上3点考虑，本文选取 VRF(Verifiable Random Functions)[20] 算法进行节点选择。

4　模型验证

4.1　实验设置

本实验选用实验室局域网搭建的 P2P 测试网络作为实验的网络环境，实验数据采用 MATLAB，在进行多轮验证之前，需要对某些参数进行设定，同时需要设定下列实验条件。

(1)测试网络的情况会影响验证消息发送的效率，假定网络中各个节点之间的延迟在可控范围内。

(2)设定系统总结点数足够多，分片规模达到极限，分片内节点数与共识验证节点数目均匀恒定，且各个节点可以动态加入和退出。

(3)在不同轮的共识过程中，节点属性能够改变，即在本轮中表现为诚实的节点，在下一轮可能变为拜占庭节点。

(4)本实验采用节点收到交易信息后不进行转发来模拟恶意节点的不诚实行为。

实验中相关参数的设置如下：

(1)区块链系统节点数目为 $N=3000$；

(2)区块链系统内拜占庭节点所占比例 $P_s=0.33$；

(3)初始分片数目 $k_s=8$；

(4)分片内验证节点数目 $N_{sv}=16$；

(5)分片内节点数 $N_s=150$；

(6)负载均衡模型中的参数 $r_{l\max}=5$。

4.2　实验验证

为了验证本文方法可以有效实现分片负载均衡，将本文方法与多轮验证方案[19] 在交易验证率和单个分片交易能力进行对比实验。

4.2.1　交易验证率对比实验

交易验证率是区块链系统在一段时间内验证交易数占总交易数的比重，在相同时间内，系统内节点验证的交易数目越多，该系统的交易验证率越高，其性能就越好。实验的目的是验证本文方法可以有效提高系统的交易验证率。本实验设置每个分片投放 10000 笔交易，记录从交易投放时刻起每个分片的交易处理情况，并且分别计算每轮验证完成后各分片的交易验证率。重复 10 次实验，计算系统的平均交易验证率。

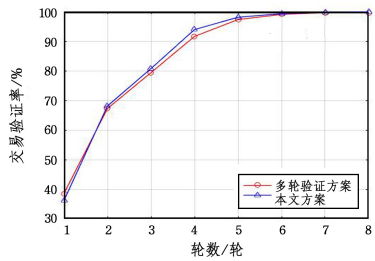


图 6　交易验证率对比图

Fig. 6　Comparison of transaction verification rate

从图 6 可以看出，本文方法与多轮验证方案在交易验证率方面差别不大，且与多轮验证方案相比，本文方法加入了状态归约模型，减轻了节点的存储压力。

4.2.2　单个分片交易处理能力对比实验

本文根据共识验证节点性能对分片间的负载进行均衡，实验目的是验证本文方法可以使各个分片间的负载相对均衡。在模拟系统中持续运行两个方案，每隔一分钟向两个模拟系统中投放 30000 笔交易，第十次投放交易时，模拟系统中的交易总数为 300000。重复实验，记录第十次投放交易后分片处理交易的数目，根据模拟系统中处理的交易数目计算单个分片的交易处理能力。

由图 7 可以看出，本文方案的单个分片处理能力与多轮验证方案有很大差异。主要原因是，本文模型是根据状态性能对节点进行区分，位于上层的分片性能优于下层分片，因此上层分片处理交易的能力相对较强，在对分片间负载均衡化处理方面，处于同一层的分片间的负载差异较小。因此，本文

方案可以缓解高负载分片的压力。

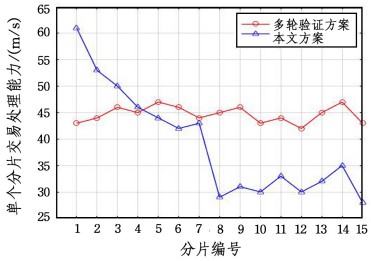


图 7 单个分片交易处理能力对比图

Fig. 7 Comparison chart of single shard transaction processing capacity

结束语 本文针对交易按照输入方地址映射到分片造成的分片负载不均衡问题，提出了利用状态归约的分片负载均衡方法。该方法应用状态归约模型弱化分片内节点间的能力差异，根据网络中的节点的性能和表现对其评分且划分等级，通过在剩余负载增大的分片随机分配性能和表现更优的节点，以求快速减少分片内的剩余负载。实验结果证明，本文方法可以在不降低单个分片的交易验证率的情况下，有效减轻分片间负载不均衡的情况。

本文粗略地提出了利用状态归约的分片负载均衡方法，该方法还存在需要进一步完善之处，如负载均衡模型中的参数、PBFT 节点减少而带来的安全性问题等。本文在考虑状态分片的基础上进行分片负载均衡，状态分片由状态归约模型实现，但关于分片间如何实现支持归约的状态同步并没有给出具体方案，这将是下一步的研究重点。

参 考 文 献

[1] NAKAMOTO S. Bitcoin: A peer-to-peer electronic cash system [J]. Journal for General Philosophy of Science, 2008, 39(1): 53-67.

[2] SUN Z X, ZHANG X, XIANG F, et al. Survey of Storage Scalability on Blockchain [J]. Journal of Software, 2021, 32(1): 1-20.

[3] LI Y, MEN J B, YU H, et al. Overview of Blockchain Capacity Expansion Technology [J]. Electric Power Information and Communication Technology, 2020, 18(6): 1-9.

[4] WILK A, IYENGAR J, SWETT I, et al. QUIC: A UDP-Based Secure and Reliable Transport for HTTP/2 [J/OL]. https://datatracker. ietf. org/doc/html/draft-tsvwg-quic-protocol-02.

[5] YU G, WANG X, YU K, et al. Survey: Sharding in Blockchains [J]. IEEE Access, 2020, 8: 14155-14181.

[6] DANG H, DINH T, LOGHIN D, et al. Towards Scaling Blockchain Systems via Sharding [C]// Proceedings of the 2019 International Conference on Management of Data. New York, Association for Computing Machinery. 2019: 123-140.

[7] GAI K K, HU Z Y, ZHU L H, et al. Blockchain Meets DAG: A BlockDAG Consensus Mechanism [C] // ICA3PP 2020. New York, Algorithms and Architectures for Parallel Processing, 2020: 110-125.

[8] AMRITRAJ S, KELLY C, REZA M. Sidechain technologies in blockchain networks: An examination and state-of-the-art review [J]. Journal of Network and Computer Applications, 2020, 149(102471): 1-16.

[9] LI G, HUANG Y M, ZHENG G P, et al. Technical Prospect of Raiden Network in Electric Vehicle Charging Transaction [J]. Electric Power Construction, 2018, 39(9): 78-86.

[10] LUU L, NARAYANAN V, ZHENG C, et al. A Secure Sharding Protocol For Open Blockchains [C]// The 2016 ACM SIGSAC Conference. ACM, 2016.

[11] ZAMANI M, MOVAHEDI M, RAYKOVA M. RapidChain: Scaling Blockchain via Full Sharding [C]// The 2018 ACM SIGSAC Conference. ACM, 2018.

[12] DANNEN C. Introducing Ethereum and solidity [M]. Berkeley: Apress, 2017.

[13] BACH L M, MIHALJEVIC B, ZAGAR M. Comparative analysis of blockchain consensus algorithms [C]// 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). 2018: 1545-1550.

[14] TIAN G H, HU Y H, CHEN X F. Research progress on attack and defense techniques in block-chain system [J]. Journal of Software, 2021, 32(5): 1495-1525.

[15] SUKHWANI H, MARTÍNEZ J M, CHANG X, et al. Performance Modeling of PBFT Consensus Process for Permissioned Blockchain Network (Hyperledger Fabric) [C]// 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS). 2017, 253-255.

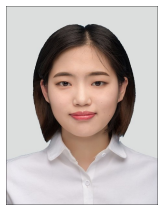
[16] LIU Y, LIU J, HEI Y, et al. A Secure Shard Reconfiguration Protocol for Sharding Blockchains Without a Randomness [C]// TrustCom 2020. Guangzhou, Security and Privacy in Computing and Communications, 2020: 1012-1019.

[17] YUN J, GOH Y, CHUNG J M. Trust-Based Shard Distribution Scheme for Fault-Tolerant Shard Blockchain Networks [J]. IEEE Access, 2019, 7: 135164-135175.

[18] ALDAKHEEL J S, ALAHMAD M A, FOUDERY A A. Comparison Between Bitcoin and Quarkchain [J]. Journal of Computational and Theoretical Nanoscience, 2019, 16(3): 818-822.

[19] LI M, LI Z H, BAI B, et al. Multi-Round Balancing Verification Scheme for Sharding Load [J]. Computer Science and Application, 2021, 11(1): 45-55.

[20] GOYAL R, HOHENBERGER S, KOPPULA V, et al. A Generic Approach to Constructing and Proving Verifiable Random Functions [C]// TCC 2017. Baltimore: Springer, 2017: 537-566.



CHEN Jing, born in 1995, postgraduate. Her main research interests include blockchain technology and so on.

LI Zhi-huai, born in 1964, professor. His main research interests include blockchain technology and network information security.

（责任编辑：喻藜）