计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202440367
61(10): 2554−2569, 2024

面向数据要素流通的区块链数据加速交换方法

李振宇[1,2] 丁 勇[1,2,3] 易 晨[3] 梁 海[1,2]

[1](桂林电子科技大学计算机与信息安全学院 广西桂林 541004)

[2](广西密码学与信息安全重点实验室(桂林电子科技大学) 广西桂林 541004)

[3](港专学院 香港 999077)

(lizhenyu.0@163.com)

A Blockchain Data Acceleration Exchange Method for Data Element Circulation

Li Zhenyu[1,2], Ding Yong[1,2,3], Yi Chen[3], and Liang Hai[1,2]

[1](*School of Computer Science and Information Security*, *Guilin University of Electronic Technology*, *Guilin*, *Guangxi* 541004)

[2](*Guangxi Key Laboratory of Cryptography and Information Security* (*Guilin University of Electronic Technology*), *Guilin*, *Guangxi* 541004)

[3](*HKCT Institute of Higher Education*, *Hong Kong* 999077)

Abstract With the advancement of the new technological revolution and industrial transformation, secure and reliable data circulation has become a pressing issue. To address this, data exchanges have begun leveraging blockchain technology to facilitate data circulation, thereby enhancing trust and regulatory capabilities during the process. However, blockchain technology still faces challenges in handling large-scale data element exchanges efficiently. To improve efficiency in the circulation of data elements via blockchain, we propose a blockchain data exchange acceleration method. Firstly, a dynamic location model for data circulation is introduced to aid data exchanges in managing the direction of data flows. Secondly, a data element aggregation upload method is proposed, enabling deep aggregation of data elements with specified circulation destinations, thus accelerating blockchain data exchange. Thirdly, an optional automatic deduplication download method is presented to address application issues encountered in blockchain-based data circulation. Finally, experimental results demonstrate that the proposed blockchain data exchange acceleration method, when combined with blockchain transaction packaging technologies, significantly enhances the efficiency of data element exchanges. Compared with the asynchronous data exchange method provided by existing blockchain frameworks, our method achieves approximately a 9.24 times improvement in data exchange performance.

Key words data element; blockchain; data circulation; data aggregation; acceleration exchange

摘 要 随着新一轮科技革命和产业变革的深入发展,安全可靠的数据要素流通已成为当前迫切需要解决的问题.从而,数据交易所已开始使用区块链技术进行数据要素流通,能一定程度提升数据要素流通过程中的信任和监管能力.但区块链技术在面对海量的数据要素交换时仍需进一步研究以提升效率.因此,针对当前基于区块链的数据要素流通过程中效率偏低问题,提出区块链数据加速交换方法.首先,提出数据要素流通动态位置模型,方便数据交易所进行数据流通方向管理.其次,提出数据要素聚合上传方法,实现针对数据要素附带流通目的地的深层次聚合,以进一步加速区块链数据交换.随后,提出可选自动去

收稿日期: 2024-05-31; **修回日期**: 2024-07-14

基金项目: 国家重点研发计划项目(2023YFB3107301); 广西科技计划项目(桂科 AA22068067); 广西研究生教育创新计划项目(YCBZ2024163)
This work was supported by the National Key Research and Development Program of China (2023YFB3107301), the Guangxi Science and Technology Major Program (AA22068067), and the Innovation Project of Guangxi Graduate Education (YCBZ2024163).

通信作者: 丁勇(stone_dingy@guet.edu.cn)

重下载方法，以解决数据经区块链流通面临的应用问题. 最后，通过一系列实验验证，提出的区块链数据加速交换方法能在已启用区块链交易打包等技术的条件下，进一步提升数据要素交换的效率，其相比于区块链框架所提供的异步数据交换方法，能获取约 9.24 倍的额外数据交换性能.

关键词 数据要素；区块链；数据流通；数据聚合；加速交换

中图法分类号 TP393.08

在新一轮科技革命和产业变革深入发展的背景下，数据要素正逐渐成为生产中不可或缺的宝贵资源，其价值日益凸显. 根据国家工业信息安全发展研究中心的预测，2025 年我国数据要素市场的规模预计将从 2021 年的 815 亿元激增至 1989 亿元[1]，其反映出数据要素的巨大经济潜力和对高质量社会经济发展的重要推动作用. 为把握数据要素发展机遇，国家数据局联合其他 16 个部委，共同印发了《“数据要素×”三年行动计划（2024—2026 年）》，覆盖工业制造、现代农业、商贸流通、交通运输、金融服务等 12 个行业，以全面促进数据要素在各领域的应用，推动经济高质量发展[2]. 鉴于数据要素的重要性日益增长，构建一个可信的数据交易和流通机制变得尤为迫切，其不仅能更高效地利用数据资源，也是解决数据流通中存在的各种问题的关键方法. 然而，当前数据流通面临多重挑战，包括数据供应方的权利受限、数据及其服务的质量问题，以及数据需求方的违约使用等[3]，这些问题在一定程度上制约了数据要素行业的快速发展.

为有效解决当前面临的挑战并优化数据流通机制，充分发挥数据要素的经济价值，推动未来社会经济发展，数据交易所已开始引入区块链技术作为加强数据要素流通安全性和可靠性的框架[4]. 区块链技术相比于传统的数据交易技术，可依托区块链技术的不可篡改性，保证数据要素流通记录不能被修改和删除，极大地提升数据的真实性与安全性，保障了数据需求方的权益[5]. 其次，区块链的可追溯特性允许所有参与方跟踪数据的来源和流向，能大幅度增加交易过程的透明度和信任度，并使数据交易所等监管方能有效对数据要素流通全流程监管. 此外，区块链的智能合约提供的可编程性能大幅度满足供需双方的要求[6]，有效缓解数据标准化程度低所带来的数据要素流通困难. 然而，区块链技术受限于共识机制的工作流程复杂，其长期面临着数据交换压力大、数据交换效率不高的问题. 如出现大量的数据要素流通的指令，基于区块链的数据要素流通系统将会面临不能及时处理情况[7].

因此针对区块链技术在数据交换效率方面遇到的挑战，业界正在积极探索一系列创新技术方案，这些方案主要聚焦于 4 个关键方向：首先，通过对区块链的共识机制进行改进，增强其处理交易的速度和能力[8]；其次，实施交易打包技术，在单个交易中打包多个交易请求，显著减少由碎片化请求导致的额外处理时间开销，从而提高整体交换效率[9]；再者，采用分片技术，通过将区块链网络分割成若干小的分片，每个分片独立处理一部分交易，实现交易的并行处理，以此大幅度提升系统的处理速度[10]；最后，利用多链技术，实现不同区块链之间的并行工作，也可进一步提升区块链系统的处理能力[11]. 然而，这些先进技术的应用虽然在一定程度上优化了区块链的交换性能，提升了数据处理的速度和效率，但在应对数据要素流通场景下的定向流通需求时，现有技术解决方案，例如物联网场景下的数据要素流通[12]，除了数据文件级的加速上传交换以及区块链的交易合并外，仍未能深度地对同目的地数据进行更高层次合并. 从而，基于区块链的数据要素交换效率仍存在较大的提升空间.

因此，本文针对基于区块链进行数据要素流通时面临目标动态变更、吞吐量不足以及应用开发难度偏高的问题，提出面向数据要素流通的区块链数据加速交换方法，以进一步提升当前数据要素使用区块链进行流通的性能. 本文的主要贡献包括 3 个方面：

1）提出了面向数据要素流通管理的动态位置方法，通过利用区块链智能合约进行数据要素传输位置管理，实现支持数据交易所对数据要素场内交易场景中动态数据要素流通目的地的管理.

2）提出了数据要素聚合上传流通方法，通过接收到需上传的数据要素后即刻附加数据流通位置标签，在实现数据要素聚合传输的同时，还可实现数据要素流通全过程的快速定位监管.

3）提出了不修改区块链内部结构的外挂式数据加速交换方法，通过设计周期同步区块链机制，有效提升区块链数据交换性能，其相比于区块链自带的

异步方法能额外获得约 9.24 倍数据交换速度.

1 相关工作

在使用区块链进行数据交易方面，Zhou 等人[12]提出了用于数据交易的区块链大数据交换框架，实现通过安全真实性及可靠性评级机制选择卖家，提供了基于区块链数据交易的可信交易方案. 其次，Sober 等人[13]调研基于区块链的物联网数据市场方案，依据调研结果分析可知，区块链技术实现数据市场的功能包括用户管理、设备管理、产品管理、结算和利用代理转发数据等，证明了区块链在数据交易管理上具有可行性. 此外，Su 等人[14]提出基于区块链的敏感数据交易系统，系统内包含电子合约层、软件即服务层，通过利用软件即服务层与交易方交换数据，并使用电子合约层存储交易信息，可实现基于区块链的敏感数据安全交易. 另外，季姝等人[15]提出区块链加隐私计算的数据要素跨域安全流通架构，主要从物理资源层、基础设施管理层、存储层、核心服务层、应用层和用户层总共 6 个层次进行设计，实现面向数据提供方、数据监管方以及数据使用方的区块链数据要素流通交易系统. 因此，当前使用区块链进行数据要素流通具有可行性，但如何通过区块链进行快速的数据要素流通交换仍需要进一步研究.

在数据交易场景下的区块链共享数据方面，Tao 等人[16]提出名为 B-FLACS 的数据共享方法，该方法首先进行预共享密钥，随后使用预共享密钥将数据加密后传输到区块链，以实现利用区块链完成数据共享. 其次，王栋等人[17]也提出基于区块链的数据访问控制方法，实现数据要素链上与链下结合方式共享. 此外，Chen 等人[18]提出了基于区块链的数据共享和流通方案，该方案主要通过区块链进行数据的密钥交换，并结合 Hadoop 分布式文件系统(Hadoop distributed file system, HDFS)实现数据加密流通. 另外，An 等人[19]提出了基于区块链的众包数据交易系统，名为 BCDT. BCDT 在工作时，首先利用区块链系统进行数据密钥传输；随后数据供应方将数据加密后，上传至星际文件系统(interplanetary file system, IPFS)；最后通过区块链对数据进行评分，从而使卖家获得相应报酬，完成数据交易. 此外，Han 等人[20]提出名为 FIBPRO 的基于区块链点对点数据管理和共享的云存储系统. 同样的，FIBPRO 也首先将数据加密，随后把密钥通过区块链传输，并使用 IPFS 共享加密数据，以减少区块链开销，提升整体系统性能. 因此，通过上述现状调研，可知目前在使用区块链进行数据共享方面，对于海量数据共享加速方法仍以外挂存储系统为主，对于数据要素快速上传至区块链的加速方法仍缺少研究.

此外，在非数据交易场景下的区块链加速传输相关研究中，主要技术方案为通过对区块链进行拓展，实现区块链压力卸载以及交易打包. 朱立等人[21]在研发上海证券交易所的交易系统过程中，提出了高性能联盟区块链技术，其通过共识分离、优化存储和数字签名优化等方法提升了交易系统的性能. 其次，Wang 等人[22]提出了基于大地址空间进行多级划分和应用布隆过滤器对地址检查过程进行优化的组合方法，可一定程度上减少磁盘操作. 此外，Qi 等人[23]提出了结合冗余拜占庭容错的两级共识机制，可提高吞吐量并减少共识所造成的延迟，从而加快区块链数据的交换速度. Lin 等人[24]提出面向异构区块链的数据实时交换方案，其通过设计基于抵押品和挑战机制的智能合约，将链上验证过程转移到链下执行，从而有效减少链上执行时间，提升区块链相关系统的效率. 另外，Kong 等人[25]提出名为 EVONChain 的双层公有区块链网络架构，其通过将验证交易打包和对工作量证明进行内外层分离，实现了提高吞吐量的目标. 因此，通过对上述现状调研，目前区块链的优化方法在拆分工作负载上具有可行性，但是仍然缺少面向数据要素流通的区块链外部加速交换方法的研究.

从而，针对当前数据要素流通面临的数据交换性能仍有待提高的挑战，本文将从面向数据要素交易的区块链加速交换、不改动现有区块链框架及适应更广泛场景上开展更深层次研究，以进一步提升基于区块链数据要素流通的效率，实现数据要素快速流通. 本文与其他文献的工作对比如表 1 所示.

2 数据要素流通架构

本节主要描述了基于区块链的数据要素加速交换架构. 相应地，介绍数据要素动态位置模型、数据要素聚合上传流通模型以及可选自动去重的数据要素下载模型. 关于本文所使用的主要符号定义如表 2 所示.

2.1 基于区块链的数据要素流通

本文假设数据要素在流通过程中，数据从提供设备开始发送，并利用数据要素流通节点将数据传输到区块链；随后使用设备访问相邻的数据要素流

Table 1 Comparison of Related Work

表 1 相关工作对比

方法	工作主要内容	加速数据交换	不改动区块链框架	适用场景
文献 [12]	区块链数据交换框架	IPFS 方案	√	数据交易
文献 [13]	数据市场		√	数据市场
文献 [14]	敏感数据交易系统		√	数据交易
文献 [15]	数据要素跨域流通架构		√	任意场景
文献 [16]	安全数据共享		√	数据共享
文献 [17]	数据访问控制		√	任意场景
文献 [18]	分布式数据流通	HDFS 方案	√	数据共享
文献 [19]	众包数据交易系统	IPFS 方案	√	传感器数据交易
文献 [20]	点对点数据共享系统	IPFS 方案		数据共享
文献 [21]	上海证券交易所交易系统	√	√	交易所
文献 [22]	面向大地址空间减少读写	√		物联网
文献 [23]	两级共识机制	√		物联网
文献 [24]	异构区块链数据实时交换	IPFS 方案	√	支付结算
文献 [25]	双层公有区块链网络架构	√		移动边缘计算
本文	区块链数据加速交换	√	√	任意场景

注:“√”表示文献对该项工作已进行研究或讨论.

Table 2 Symbols and Their Interpretations

表 2 符号及其释义

符号	定义
eid	位置记录标识符
loc	数据要素使用设备位置
lp	数据要素使用设备位置的网络区域信息
$udata$	上传至区块链的数据要素
uq	上传的数据元素
UQ	并行处理队列$UQ=\{uq_1,uq_2,\cdots,uq_n\}$
mid	聚合后依据处理时间生成的标识符
md_x^y	y设备编号的x批次的聚合上传元素条目
MUI	聚合上传条目集合
dle	可下载位置元素
DLE	可下载位置列表$DLE=\{dle_1,dle_2,\cdots,dle_n\}$
MID	最大的合并数据编号
di_x^y	聚合下载元素条目
dq	可下载数据元素
DQ	可下载队列$DQ=\{dq_1,dq_2,\cdots,dq_n\}$
L_x	装有x类型元素的列表

通节点,从区块链中获取数据.因此,在基于区块链的数据要素流通过程中将面临 3 个问题:1)如何解决使用设备动态变更的问题.针对此问题,常规的做法是对传输目标位置使用标识符进行标记.2)如何解决区块链上链缓慢问题.针对此问题,常规的做法是进行交易打包,但是单纯的交易打包对数据要素的条件下载情况并不友好,可能导致额外的资源开销.3)如何解决数据下载过程中面临的区块链智能合约超出资源限制的问题.针对此问题,常规的做法是按需返回数据,但仍然需要解决过多数据单次难以完成处理的问题.因此,当前仍需要对区块链数据加速交换中面临的问题进行研究.

2.2 区块链数据加速交换框架

为解决使用区块链进行数据要素流通所面临的数据交换效率核心问题,本文提出使用外部加速交换方法,其可加速基于区块链的数据要素交换速度,从而构建基于区块链的数据要素快速交换通道,关于区块链的数据加速交换架构如图 1 所示.

首先,数据要素提供设备结合流通动态位置信息,把需要传输的数据向使用设备发送.随后数据到达数据要素流通节点后,由数据要素流通节点,利用本文提出的数据要素聚合上传流程,直接对需要传输的数据进行聚合.之后再由区块链的智能合约写入到区块链中.在完成区块链上链后,使用设备可通

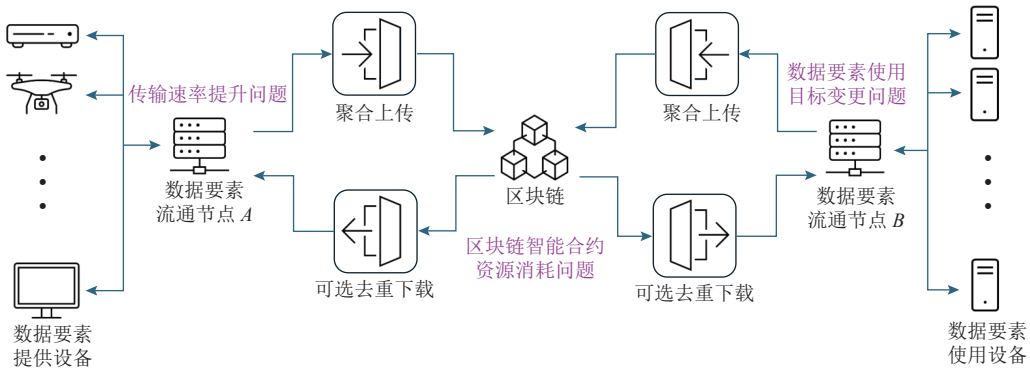


Fig. 1 Acceleration exchange framework of blockchain data

图 1 区块链数据加速交换框架

过对端数据要素流通节点，利用数据要素可选自动去重下载流程，对需要的数据进行可选择下载. 最后在区块链上标记已下载的数据信息.

2.3 数据要素流通动态位置模型

为实现面向区块链的数据要素快速流通以及支持动态选择使用设备，本文提出数据要素流通动态位置模型，其组件包含用于使用设备位置记录的智能合约、数据要素交易管理器以及数据要素流通节点. 其中，数据要素交易管理器用于对接数据要素交易系统，以实现基于数据要素交易结果进行动态调整数据传输路径. 关于数据要素流通动态位置选择工作流程图如图 2 所示. 在数据要素流通动态位置选择工作流程中，首先需要通过数据要素交易管理器与流通位置信息库关联的智能合约进行交互，更新对应的流通位置数据；随后数据要素发送端在发送数据要素以及接收端在接收数据要素前，先通过区块链的位置记录智能合约，进行流通位置查询；当获取目标流通位置信息后，进行后续的数据要素交换.

其中，在流通位置信息库内，位置记录智能合约包含n个位置信息元素le. 从而，位置记录的集合为：

$$LE=\{le_1,le_2,\cdots,le_n\}.$$

同时，位置信息元素le由多个包含使用设备的标记信息组成. 本文定义$le=\{eid,loc,lp\}$，其中eid为位置记录标识符，loc为使用设备位置，lp为使用设备位置的网络区域信息.

从而，动态选择使用设备模型首先需通过数据要素交易管理器设置位置记录LE. 随后数据要素流通节点通过智能合约读取所需的目标位置信息元素le，并利用对应的le附加标记后将数据上传到区块链. 最后，接收端可利用所在的位置信息le从区块链下载数据.

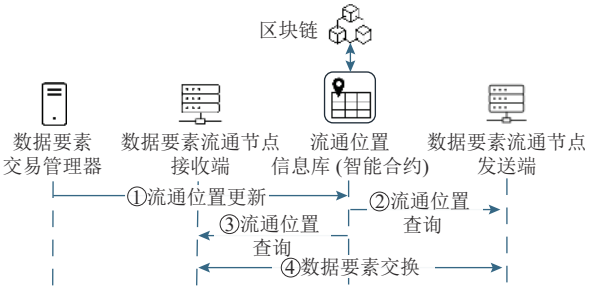


Fig. 2 Flow chart for dynamic selection of data element circulation locations

图 2 数据要素流通位置动态选择流程图

2.4 数据要素聚合上传流通模型

为了有效提升区块链上传性能，本文提出了依据目标聚合上传的模型，该模型由数据要素快速接收、聚合相同流通目的地的数据要素，以及聚合的数据要素周期性记录至区块链共 3 个部分组成. 关于数据要素聚合上传流通工作流程图如图 3 所示. 在数据要素聚合上传流通工作流程中，首先通过数据上传接口，结合区块链数据要素流通位置信息库，对数据要素快速附加流通位置标记，并上传数据并行处理队列完成数据要素的快速接收. 随后，对相同流通目的地的数据要素进行快速数据聚合，形成聚合数据集合. 最后，将聚合数据集合周期性地通过数据流通智能合约上传至区块链，从而完成数据要素的聚合上传流通工作流程.

2.4.1 数据要素快速接收

在依据目标聚合上传的模型设计中，针对上传速度可能会存在瞬时大于上链速度的情况，本文提

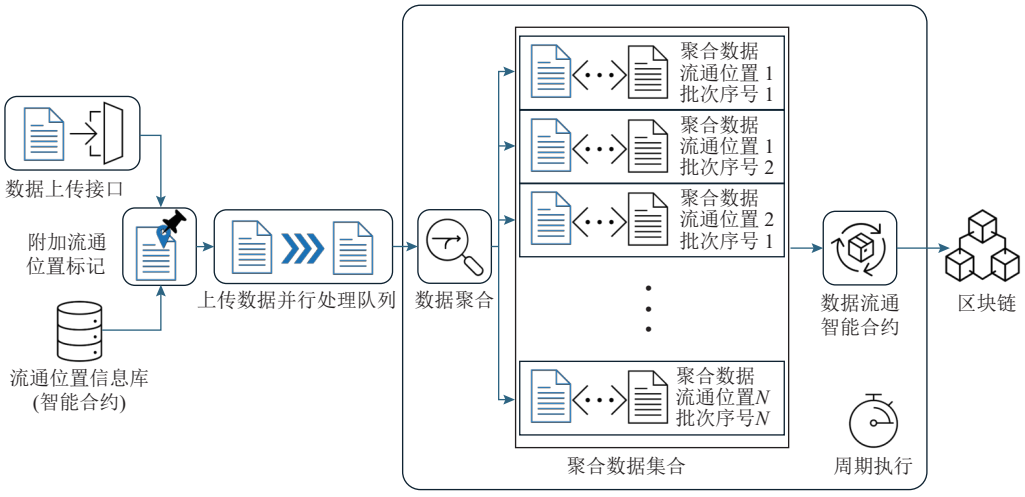


Fig. 3 Workflow chart of data element aggregation and uploading

图 3 数据要素聚合和上传工作流程图

出利用并行处理队列将传入的数据快速接收后再进行同目标地的数据聚合. 因此, 本文定义并行处理队列$UQ=\{uq_1,uq_2,\cdots,uq_n\}$, 其中$uq$为待处理上传元素. 此外, 每个待上传元素被定义为$uq=\{eid,udata\}$. 其中, eid为位置记录标识符, 可关联到使用设备信息; $udata$为待处理上传数据.

2.4.2 聚合相同流通目的地的数据要素

当完成数据快速接收后, 数据聚合器开始对上传数据并行处理队列UQ内的所有数据进行相同流通目的地的数据聚合, 生成聚合上传条目:

$$md=\{mid,eid,(udata_1,udata_2,\cdots,udata_n)\},$$

其中, mid为聚合后依据处理时间生成的标识符, $udata$为需上传的数据, n为需上传元素个数, eid为位置记录标识符, 当前记录为所属聚合上传条目所有$udata$的共同的位置记录标识符. 因此, 聚合上传条目集合为:

$$MUI=\left\{md_1^1,md_2^1,\cdots,md_x^1,md_1^2,md_2^2,\cdots,md_x^y\right\},$$

其中, x为批次号, y为使用设备编号. 同时, 由于区块链所接收的数据单次请求会有长度限制, 因此对于每个md_x^y需满足以下数据长度要求:

$$0<Len_{md_x^y}\leqslant Len_{\mathrm{c}},$$

其中, Len_{c}为智能合约单次接收支持的长度, 如果$Len_{md_x^y}$超过Len_{c},则需要把剩余的数据放入下一批次.

2.4.3 聚合的数据要素周期记录至区块链

完成数据聚合后, 需要定期地将需上传的数据通过智能合约传输到区块链. 此时上链过程需要消耗一定的时间, 时间消耗公式被定义为:

$$T_{\mathrm{u}}=\frac{Len_{MUI}}{v_{\mathrm{u}}},$$

其中, Len_{MUI}为上传数据总长度, v_{u}为上传 1 个长度单位数据的速度. 当数据量过大, 超出区块链吞吐能力时则会出现数据上传延时增大, 从而需要花费更多的时间对数据进行上链.

2.5 可选自动去重的数据要素下载模型

当数据完成上传后, 使用设备可以对区块链上记录的未处理数据进行下载. 从而, 为实现快速有效地下载数据, 本文提出可选自动去重下载模型, 其由预下载合并的数据要素、分离聚合的数据要素以及输出可下载的数据要素共 3 个部分组成. 关于可选自动去重数据要素下载的数据流程图如图 4 所示. 在可选自动去重数据要素下载的数据流程中, 首先流通节点需要获取受管理的流通位置标识集合, 随后结合已处理的最大聚合数据编号, 通过数据流通智能合约, 按设定周期定期从区块链上预下载合并的数据要素, 获得聚合数据集合. 其次对聚合数据集合内的各个聚合数据进行分离, 形成可下载数据要素. 最后将可下载数据要素传入下载数据待获取集合, 并依据请求下载信息, 经由数据下载接口, 输出与目标流通位置相关的数据要素.

2.5.1 预下载合并的数据要素

基于控制数据要素定向传输的需求, 本文提出通过使用数据要素流通动态位置模型所提供的位置信息, 并结合数据流通智能合约, 从区块链中周期下载聚合数据. 因此, 可下载位置列表定义为:

$$DLE=\{dle_1,dle_2,\cdots,dle_n\},dle_n\in LE,$$

其中, dle为可下载位置, n为可下载位置数量, 并且所有的dle_n都属于流通位置信息库的位置记录集合LE. 从而, 周期下载程序可以根据下载列表在区块链中获取预下载数据集合, 关于预下载的聚合数据集合可表示为:

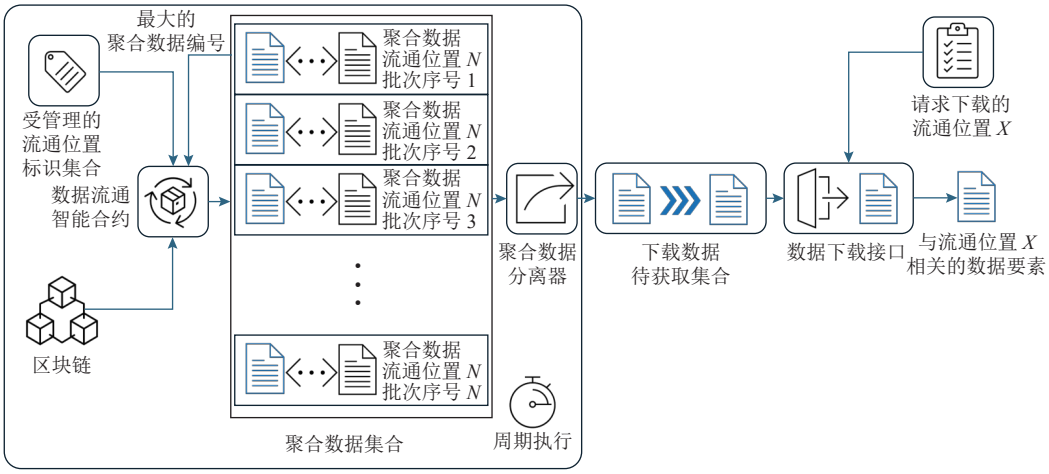


Fig. 4 Flow chart of optional automatic de-duplication download of data elements

图 4 数据要素可选自动去重下载流程图

$$DI=\left\{di_1^1,di_2^1,\cdots,di_x^1,di_1^2,di_2^2,\cdots,di_x^y\right\},$$
$$y\in DLE,ID\left(di_x^y\right)>MID,$$

其中，di为单个聚合数据条目，其结构与聚合上传条目md一致；x为批次号；y为使用设备序号. 同时，在聚合下载数据集合中，由于受智能合约的选择输出限制，y的集合必定属于DLE. 此外，为有效解决重复下载问题，预下载数据还需要计算当前数据集合DI的最大编号MID. 因此，本文定义最大的合并数据编号为：

$$MID=\max\left(ID\left(di_1^1\right),ID\left(di_2^1\right),\cdots,ID\left(di_x^1\right),ID\left(di_1^2\right),ID\left(di_2^2\right),\cdots,ID\left(di_x^y\right)\right),$$

其中，$ID(x)$为获取传入di的聚合后依据处理时间生成的标识符mid的函数.

2.5.2 分离聚合的数据要素

完成聚合数据预下载后，下载模型需要对聚合数据进行还原. 其中，数据还原器需对每一个di_x^y进行处理，根据位置记录集合LE所提供的数据，将di_x^y进行解析，生成下载元素. 因此，本文定义可下载数据元素为：

$$dq=\{eid,loc,lp,udata\},$$

其中，eid为位置记录标识符，loc为使用设备位置，lp为使用设备位置网络区域信息，$udata$为待处理下载数据. 此外，还原需要消耗一定的存储空间，本文定义存储消耗空间公式为：

$$S_{DQ}=\{s_{dq_1},s_{dq_2},\cdots,s_{dq_n}\},$$

其中，s_{dq_n}为dq_n所需的存储空间，与分离对象所申请的内存空间一致.

2.5.3 输出可下载的数据要素

完成数据还原后，数据要素下载模型将所有可下载元素dq放入可下载队列，在本文中可下载队列被定义为：

$$DQ=\{dq_1,dq_2,\cdots,dq_n\},$$

其中，n为可下载队列中元素的个数. 当DQ不为空时，请求下载使用设备或者数据要素中转设备可凭eid下载需要处理的数据. 此外，数据要素中转设备还可根据loc和lp将数据分发到指定的使用设备.

3 区块链数据加速交换方法

本节首先介绍使用区块链进行数据要素交换的应用样例，随后详细介绍用于区块链数据要素流通的外部加速交换方法. 关于区块链数据加速交换方法，其包含了2个核心方法：数据要素流通位置管理方法以及数据要素流通方法. 数据要素流通位置管理方法中包含流通位置更新方法和流通位置查询方法. 而数据要素流通方法中包含数据要素聚合上传方法和数据要素可选下载方法.

3.1 应用样例

本文根据数据使用区块链对数据要素流通的需求、提炼与数据要素交换的关键部分进行深度分析，以实现数据要素使用位置动态定位、数据要素快速上传和数据要素依据使用位置下载的功能. 从而，整个数据交换架构的核心是通过数据要素流通节点实现设备与区块链数据的交互. 因此，本文提出的基于区块链加速交换方法的数据要素流通流程图如图5所示. 在数据流通过程中，首先数据要素提供设备需要直接向数据要素流通节点发送数据，或由软件定义网络等技术实现透明的数据发送. 随后，当数据到达数据要素流通节点后，流通节点会直接对数据要素通过区块链的智能合约进行上链操作. 随后，在对端的数据要素流通节点中，会持续性地从区块链上通过数据交换区块链智能合约下载数据. 最后，对端

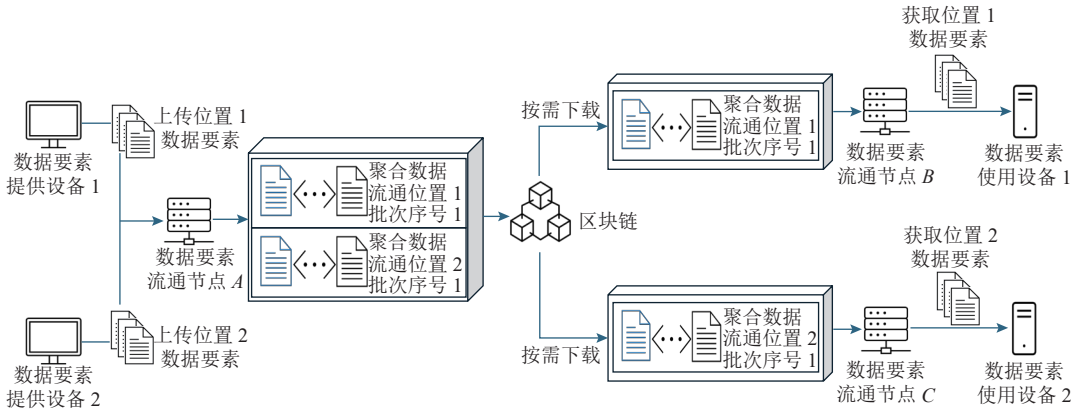


Fig. 5 Flow chart of data element circulation based on blockchain accelerated exchange methods

图5 基于区块链加速交换方法的数据要素流通流程图

的数据要素流通节点处理完数据后，发送数据至使用设备. 通过上述过程，可实现数据通过区块链进行数据流通.

3.2 数据要素流通位置管理方法

为有效对数据进行合并，减少区块链操作次数，并实现使用设备的动态选择，本文提出数据要素流通位置管理方法，其包含用于对区块链上定义最新生效位置信息的流通位置更新方法，以及提供给数据要素流通节点以确定工作范围的流通位置查询方法.

3.2.1 流通位置更新方法

流通位置更新方法如算法 1 所示. 首先位置更新方法需要输入区块链的位置服务智能合约地址，以实现区块链上的数据交换功能；同时还需提供位置记录标识符，以确定需要操作的对象；此外需提供使用设备位置以及使用设备位置网络区域信息，用于更新使用设备所在的位置信息. 从而，通过上述信息的整合使用，可对区块链上所记载的使用设备位置信息进行新增或更新.

算法 1. 流通位置更新.

输入：区块链的位置服务智能合约地址$addr_l$，位置记录标识符eid，使用设备位置loc，使用设备位置网络区域信息lp；

输出：区块链记录的新的流通位置信息.

① $item \leftarrow FindRecord\ (addr_l, eid)$;
② if $item$ is NULL then
③ $AddRecord\ (addr_l, eid, loc, lp)$;
④ else
⑤ $uf \leftarrow UpdateFields\ (loc, lp)$;
⑥ $cond \leftarrow SetCondition\ (eid)$;
⑦ $UpdateRecord\ (addr_l, cond, uf)$;
⑧ end if

在位置更新算法运行时，首先，使用指定的位置记录标识符在区块链上的位置服务智能合约中查找数据，以获取指定位置记录标识符的数据(行①). 当指定位置记录标识符的数据为空时，则说明区块链上没有与指定位置记录标识符相关的信息，进入新增位置记录模式. 在新增位置记录模式中，算法与位置服务智能合约的新增功能进行交互，传输位置记录标识符、使用设备位置及其网络区域信息至区块链. 当传输完成后，区块链新增指定位置记录标识符相关的位置信息(行②③). 如果查询的指定位置记录标识符的数据不为空，则说明区块链上有指定位置记录标识符相关的信息，此时进入更新位置记录模式(行④). 在更新位置记录模式中，首先设置需更新的使用设备位置以及网络区域信息，通过对数据进行绑定，生成更新元素组(行⑤). 随后，将位置记录标识符传入设置函数，生成智能合约需要更新数据内容的条件集合(行⑥). 最后，与位置服务智能合约的更新功能进行交互，向其传输需要更新数据内容的条件集合和更新元素组，以实现指定位置记录标识符的使用设备位置和网络区域信息更新(行⑦).

3.2.2 基于流通标识的位置查询方法

位置查询方法能够为数据要素流通节点提供依据位置进行分类聚合数据的服务，关于位置查询方法的主要流程如算法 2 所示. 当数据要素流通节点查询使用设备位置列表和使用设备位置的网络区域信息列表时，需要准备当前数据要素流通节点所管理的位置记录标识符列表，以支撑智能合约确认所要读取的数据范围. 同时还需提供智能合约中读取列表的起始位置以及连续读取个数，以实现限制批量读取数据，解决消耗过多资源导致智能合约执行失败的问题.

算法 2. 流通位置查询.

输入：区块链提供位置服务的智能合约地址$addr_l$，位置记录标识符列表L_{eid}，起始位置s，读取个数c；

输出：使用设备位置信息列表L_{srv}.

① $items \leftarrow FindAll\ (addr_l, L_{eid}, s, c)$;
② foreach i in $items$
③ $L_{\text{reid}} \leftarrow FetchEID\ (i)$;
④ $L_{loc} \leftarrow FetchLocation\ (i)$;
⑤ $L_{lp} \leftarrow FetchPrefix\ (i)$;
⑥ end foreach
⑦ $L_{\text{srv}} \leftarrow (L_{\text{reid}}, L_{loc}, L_{lp})$;
⑧ return L_{srv}.

在位置查询方法开始执行时，首先与区块链上的位置服务智能合约交互，依据位置记录标识符列表、起始位置以及读取个数，获取指定位置记录标识符的数据(行①). 随后，开始对获取的指定位置记录标识符数据进行解析，由于智能合约的限制，需要逐个对位置以及网络区域信息进行拉取(行②). 对于指定位置记录标识符数据中的每个数据，首先读取当前处理数据的位置记录标识符，放置入待输出的位置记录标识符列表(行③)；随后读取当前处理数据的使用设备位置，放置入待输出的使用设备位置列表(行④)；最后读取当前处理数据的使用设备位置的网络区域信息，放置入待输出的使用设备位置的网络区域信息列表(行⑤)，完成操作后，将位置记录标识符列表、使用设备位置列表以及使用设备位

置网络区域信息列表进行封装,以生成使用设备位置信息列表并进行返回(行⑥~⑧).

3.3 数据要素流通方法

数据要素流通方法包含聚合上传方法以及可选择下载方法.通过2个方法的结合,可实现基于区块链的数据要素快速流通.

3.3.1 数据要素聚合上传方法

聚合上传方法的主要流程如算法3所示.在执行聚合上传算法前,需要在区块链上部署完成数据交换智能合约,并获取数据交换智能合约地址.同时还需定义周期上传的间隔时间,以减少区块链交易次数.此外,需提供使用设备位置信息列表以实现数据包传输位置标注.最后,还需准备待上传数据,以提供数据进行上传.

算法3. 数据要素聚合上传.

输入:数据交换的智能合约地址$addr_{\mathrm{e}}$,周期上传时间间隔ts,使用设备位置信息列表L_{srv},待上传数据$udata$;

输出:数据记录至区块链.

① $eid \leftarrow FindLoc\ (L_{\mathrm{srv}}, udata)$;
② $uq \leftarrow Attatch\ (eid, udata)$;
③ $UQ \leftarrow Enqueue\ (uq)$;
④ if $CheckInterval\ () > ts$ then
⑤ while $uq \leftarrow Dequeue\ (UQ)$ then
⑥ $md \leftarrow AttachTimestamp\ (uq)$;
⑦ $y \leftarrow eid \leftarrow GetEID\ (md)$;
⑧ $MUI \leftarrow MergeUploadItem\ (md^{y})$, satisfy $Len_{md_x^y} \leqslant Len_{\mathrm{c}}$;
⑨ end while
⑩ foreach md_x^y in MUI
⑪ $Upload\ (addr_{\mathrm{e}}, md_x^y)$ using T_{u};
⑫ end foreach
⑬ endif

在聚合上传方法开始运行时,首先利用使用设备位置信息列表提取并匹配待上传数据中的位置信息,从而获取位置标识符(行①).之后,将位置标识符附加到待上传数据,生成待上传数据元素(行②).随后,将待上传数据元素放入待上传的多线程处理队列(行③).完成后检查当前时间距离上次的上传时间是否超过合并上传时间间隔,如果超过则进入合并及上传区块链的流程(行④).当进入合并及上传区块链的流程后,算法首先提取待上传的多线程处理队列中的所有元素(行⑤).对于每个待上传元素,首先需要依据当前时间,附加聚合上传条目的编号,生成聚合数据(行⑥).然后提取当前处理的聚合上传数据的位置标识符作为使用设备序号使用(行⑦).随后将已标识使用设备序号的聚合数据,依据不超过区块链智能合约单次接收支持长度的规则,合并到聚合上传数据集合中,完成后再继续处理下一个上传元素(行⑧~⑨).处理合并完成后,开始对聚合上传数据集合中的每一个聚合数据,利用区块链的上传方法,将聚合数据传入区块链数据交换智能合约以进行上链操作,这个过程会花费聚合上传数据集合内所有聚合数据上链相加的时间.当上传完成所有的聚合数据后,整个聚合上传流程结束(行⑩~⑬).

3.3.2 数据要素可选下载方法

可选下载方法可用于对区块链上当前存在的未处理数据进行处理,其算法细节如算法4所示.通过使用可选下载方法,能够及时地将当前使用设备所需的数据通过数据要素流通节点进行下载,以实现完整的区块链数据流通流程.在使用可选下载方法时,仍然需要提供区块链的数据交换智能合约地址,以实现与区块链的数据交互;同时还需要定义周期下载时间间隔,以减少区块链服务的压力;并且需要提供请求下载的位置,以实现可选下载和减少获取不必要的共享数据.

算法4. 数据要素可选下载.

输入:数据交换的智能合约地址$addr_{\mathrm{e}}$,周期下载时间间隔ts,请求下载的位置rl;

输出:与请求下载位置rl相关的数据列表L_{rdata}.

① if $CheckInterval\ () > ts$ then
② $DLE \leftarrow GetManagedLoc\ ()$;
③ $MID \leftarrow GetLastMaxID\ ()$;
④ while $di \leftarrow PreDL\ (addr_{\mathrm{e}}, DLE, MID)$ then
⑤ $DI, MID \leftarrow AddExtract\ (di)$;
⑥ end while
⑦ $HideData\ (DLE, MID)$;
⑧ foreach di_x^y in DI
⑨ $eid, loc, lp \leftarrow GetInfo\ (DLE, md_x^y)$;
⑩ $L_{\mathrm{udata}} \leftarrow SplitData\ (md_x^y)$;
⑪ foreach $udata$ in L_{udata}
⑫ $dq \leftarrow Pack\ (eid, loc, lp, udata)$;
⑬ $DQ \leftarrow Enqueue\ (dq)$ using S_{DQ};
⑭ end foreach
⑮ end foreach
⑯ end if
⑰ while $rdq \leftarrow Dequeue\ (DQ, rl)$ then

⑱ $L_{\text{rdata}} \leftarrow rdq$;

⑲ end while

⑳ return L_{rdata}.

在可选下载方法运行时，首先需要检查上次下载时间是否超过批量下载时间间隔(行①)，如果超过时间间隔则继续执行预下载流程(行①~⑯)，否则直接进入读取数据流程(行⑰~⑳). 当进入预下载流程后，首先，获取当前流通节点所管理的可下载位置列表(行②). 随后，获取上次预下载流程执行后获得的最大下载的序号，以避免重复下载之前已经下载的元素(行③). 然后，为解决区块链智能合约面临的资源受限问题，使用循环方法，将可下载位置列表以及上次最大下载的序号输入到区块链数据交换智能合约中，逐条或按指定数据量处理预下载数据(行④). 随后，将预下载数据添加至预下载数据集合. 同时，利用获取的预下载数据集合中最大序号，将上次最大下载的序号进行更新(行⑤). 完成预下载后，依据下载位置列表以及最大下载序号，在区块链上标记已下载聚合数据，以便后续下载时跳过，从而提升处理效率(行⑦). 随后，开始对预下载数据集合的每个聚合数据进行处理，以便读取数据流程能够直接使用数据(行⑧). 在处理过程中，首先依据可下载位置列表，分析聚合数据，获取当前聚合数据的位置记录标识符、使用设备的位置及其网络区域信息(行⑨). 随后，还需要对聚合数据进行拆分，获取聚合数据列表(行⑩). 之后，循环封装聚合条目数据列表中的每个数据. 其中需要将每一个聚合数据的位置标识符、使用设备的位置及其网络区域信息一起进行封装，生成可下载元素. 之后依据可下载元素所需开销的空间，申请对应数量的内存空间. 最后将所有可下载元素放置入可下载队列(行⑪~⑯). 完成预下载流程后，进入读取数据流程. 其中需使用请求下载的位置，对可下载队列的相关元素进行出队，以获取出队可下载元素. 随后将所有的出队可下载元素放入与请求下载位置相关的数据列表进行返回(行⑰~⑳).

4 实验与评估

在本节，首先介绍本文实验所使用的环境，并介绍本文实验所使用的测试用例；然后介绍开展本文实验使用的相关实验配置；之后对本文提出方法进行性能测量；最后对所有实验进行总结.

4.1 实验环境及配置

本文选择了 VMWare ESXi 7.0[①]版本作为实验平台，它可以帮助我们在一台物理服务器上部署多个虚拟机. 在区块链部分的测试中，我们选择了 FISCO BCOS 3.5.0[②]构建区块链网络并进行测试. FISCO BCOS 是基于以太坊的开源联盟区块链平台，其智能合约开发工具包可支持多种常用编程语言. 因此，使用 FISCO BCOS 可以简便地将数据传输至区块链. 此外，FISCO BCOS 还拥有交易打包功能，能够很好地解决区块链自身快速上链所面临的性能问题. 同时，由于本文提出的方法是区块链外挂式加速方法，因此能为本文提出的加速交换方法提供区块链内部上链加速的基础支持. 另外，为了有效与主流的区块链加速方案[12,19-20]进行对比，在实验中添加了 IPFS 0.29.0[③]作为实验对照组展开相关实验.

在实验开始前，测试程序启动 2 个线程池，其中上传线程连接数据要素流通节点 A，下载线程连接数据要素流通节点 B. 而数据要素流通节点 A 和数据要素流通节点 B 均与区块链进行连接，从而完成数据流通全流程的测试. 关于本文的各测试对象的拓扑连接如图 6 所示.

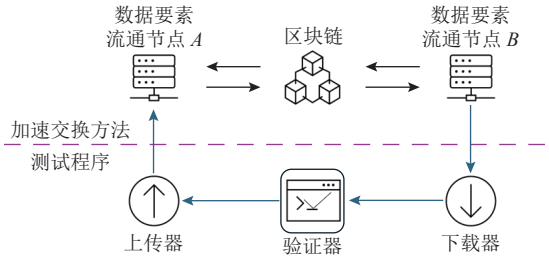


Fig. 6 Experimental topology

图 6 实验拓扑

此外，为了减少网络传输导致的测量误差，实验中将数据要素流通节点 A、数据要素流通节点 B 以及区块链放置到主机 M_A，并且对主机 M_A 同时设置在 2 个网络段的分别监听数据，以实现数据传输到不同的数据要素流通节点上的环境. 此外，测试程序单独放到主机 M_B. 同样地，主机 M_B 也支持在 2 个网络段发送和接收数据，以支持测试开展. 另外，主机 M_A 和主机 M_B 的配置如表 3 所示，以满足开展实验的需求.

① https://www.vmware.com/products/esxi-and-esx.html

② https://github.com/FISCO-BCOS/FISCO-BCOS

③ https://github.com/ipfs/kubo

Table 3 Virtual Machine Configuration for Experiments

表 3 实验虚拟机配置

虚拟机	虚拟 CPU 个数	内存大小/GB	操作系统
主机 M_A	8	16	Ubuntu 22
主机 M_B	8	32	Windows Server 2022

4.2 测试用例

在实验中, 由于测试程序需要同时发送和接收工作以及确认数据是否完成流通, 因此本文设计了专用测试程序用于本文的实验. 测试程序可以自动连接本文方法所部署的 2 个数据要素流通节点, 并向接口依次发送指定数据量的数据, 以及接收数据进行验证. 从而, 在数据量测试中, 本文实验选取了 64~1 500 B 的数据, 以适应面向透明网络数据要素交换的场景.

同时, 为有效评估本文方法在相邻 2 次同步数据至区块链中的时间间隔设置对传输速度的影响, 本文选取设置同步时间间隔为 0 ms、10 ms、50 ms、100 ms 以及 200 ms 开展对比实验. 其中, 设置同步间隔为 0 ms 的实验, 其目标是评估本文方法在立即上传数据至区块链的模式中, 与直接使用区块链内置的数据交换方法之间的性能差异. 此外, 本文还根据小规模测试的实验结果, 选取可明显观察到本文方法在同步间隔为 10 ms、50 ms、100 ms 以及 200 ms 实验设置的性能变化. 另外, 因同步时间过长, 同步间隔为 200 ms 时, 已可明显观察到性能下降的趋势, 因此不开展同步间隔 200 ms 以上的实验.

4.3 测量方法

在实验中, 所有数据的每秒交换数量的测量均通过测试程序连续进行 20 s 发送, 并设置超时时间, 以支持超出区块链吞吐量情况的测试需求. 随后利用测试程序接收端对数据进行统计, 并依据计数结果, 对 20 s 内所有交换数量求和后, 再计算平均值, 从而获得每秒平均交换数量. 关于数据每秒平均交换数量的计算公式被定义为:

$$E(t)=\frac{\sum_{i=1}^{t}\left(e_i^{s}-e_i^{o}\right)}{t},$$

其中, t代表测试时间, 其在实验中被定义为 20 s; e_i^s代表当前 1 s 内发送的数据量; e_i^o代表当前 1 s 内发送后超时的仍未完成处理的数据量, 其中超时时间在实验中被定义为 10 s. 从而, 通过连续的发送数据并扣除超时处理数据, 能获得较稳定的测量数据.

此外, 为了有效评估数据交换的数据量大小, 每秒平均交换数据量计算公式定义为:

$$EL=E(t)\times S_{\text{udata}},$$

其中, S_{udata}为当前测试批次单个交换的数据大小.

4.4 性能测量

最大化数据交换性能是本文的核心优化目标, 其可使区块链数据要素流通方法能够在更多的场合下应用. 因此, 为有效评估本文提出的区块链数据要素加速交换方法, 在区块链数据交换的性能测试中, 本节主要从区块链数据要素交换方法在单客户端的性能测试以及多客户端请求的性能压力测试, 共 2 个角度进行开展, 并对实验测量的数据进行分析.

4.4.1 区块链数据要素交换方法的性能测试

在区块链数据要素交换方法的性能测试中, 本节主要从本文方法在同步间隔设置的性能测试、主流的 IPFS 方案性能测试以及 FISCO BCOS 提供的同步及异步数据交换方法的性能测试等多个角度开展实验. 同时, 在测试过程中, 设置过多的测试线程会出现不同方法所支持的压力上限值不一致, 过大的压力会导致系统性能不足, 从而出现影响测量结果的问题. 因此在不同的交换方法测试中, 选择设置测试程序的线程数为 1, 以获取同等压力条件的实验数据. 关于区块链数据要素交换方法的单线程性能测试结果如表 4、图 7 以及图 8 所示.

如图 7 所示, 根据实验测试结果可知本文方法在同步间隔设置为 0 ms 时, 每秒交换个数最大平均值为 357; 本文方法在同步间隔设置为 10 ms 时, 每秒交换个数最大平均值为 1 334.1; 本文方法在同步间隔设置为 50 ms 时, 每秒交换个数最大平均值为 1 567.9;

Table 4 Performance Comparison of Single-Threaded Tests

表 4 单线程测试的性能对比

方法	每秒平均交换数据的个数		每秒平均交换数据量	
	最大平均个数	提升倍数	最大平均值/B	提升倍数
本文方法 同步间隔 0 ms	357.00	0.002 2	435 225	0.082 2
本文方法 同步间隔 10 ms	1 334.10	2.745 3	1 373 625	2.415 7
本文方法 同步间隔 50 ms	**1 567.90**	**3.401 7**	1 555 950	2.869 0
本文方法 同步间隔 100 ms	1 552.60	3.358 7	1 544 400	2.840 3
本文方法 同步间隔 200 ms	1 499.05	3.208 4	**1 602 525**	**2.984 8**
FISCO BCOS 的异步数据交换方法	356.20	0	402 150	0
	1.95	–0.994 5	2 850	–0.992 9
基于 FISCO BCOS 异步数据交换的 IPFS 方法	72.15	–0.797 4	97 275	–0.758 1

注: 黑体数值为最佳数值, 提升倍数表示相比 FISCO BCOS 的异步数据交换方法的性能.

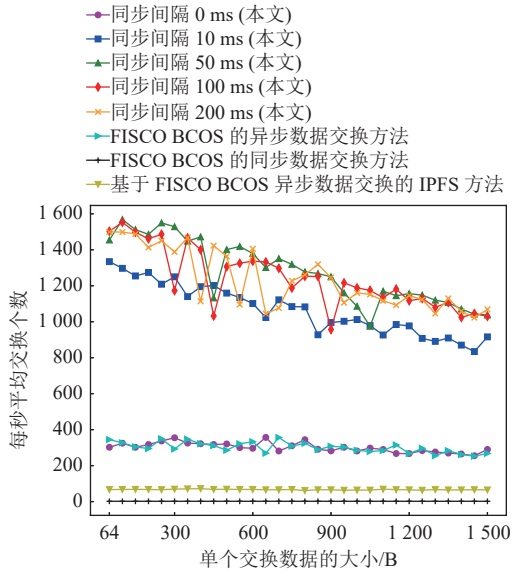


Fig. 7 Different data exchange methods in terms of the number of data exchange

图 7 不同数据交换方法的数据交换个数

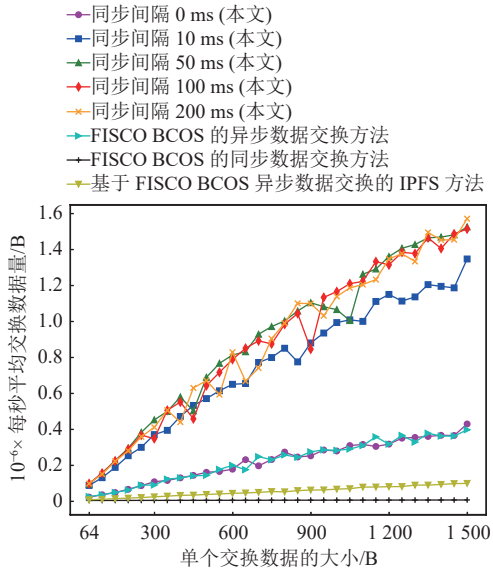


Fig. 8 Exchange data amounts of different data exchange methods

图 8 不同数据交换方法的交换数据量

本文方法在同步间隔设置为 100 ms 时，每秒交换个数最大平均值为 1 552.6；本文方法在同步间隔设置为 200 ms 时，每秒交换个数最大平均值为 1 499.05. 而仅使用 FISCO BCOS 所提供的异步数据交换方法，每秒交换个数最大平均值为 356.2；基于 FISCO BCOS 异步数据交换的 IPFS 方法，每秒交换个数最大平均值为 72.15；仅使用 FISCO BCOS 所提供的同步数据交换方法，每秒交换个数最大平均值为 1.95. 因此根据交换数据实验结果可知，本文方法在设置同步间隔为 0 ms 时，与 FISCO BCOS 所提供的异步数据交换方法性能差异不大，证明在不开启同步间隔机制下，本文方法不会对基础性能造成额外负面影响. 此外，在本文方案设置同步间隔为 50 ms 时，出现所测试结果的最大值，其与 FISCO BCOS 所提供的异步数据交换方法相比能获得约 3.4 倍的性能提升，与 FISCO BCOS 所提供的同步数据交换方法相比能获得约 803 倍的性能提升.

此外，在图 8 中，在单次请求交换的数据要素越大时，越能获得更高带宽的数据要素交换性能. 在单个数据交换大小为 1 500 B 的测试中，FISCO BCOS 的同步数据交换方法测试每秒平均能传输 2 850 B；FISCO BCOS 的异步数据交换方法测试每秒平均能传输约 0.402×10^6 B；基于 FISCO BCOS 异步数据交换的 IPFS 方法，每秒平均能传输约 0.97×10^5 B；而本文方法在同步间隔设置为 0 ms 时，能传输 0.435×10^6 B. 从而本文方法在时间间隔为 0 ms 时的性能相比于 FISCO BCOS 的异步数据交换方法的性能并不会降低. 此外，本文方法在设置同步间隔为 10 ms 时，每秒能传输 $1.373\,6\times10^6$ B；同步间隔为 50 ms 时，每秒能传输 $1.555\,9\times10^6$ B；同步间隔为 100 ms 时，每秒能传输 $1.544\,4\times10^6$ B；同步间隔为 200 ms 时，每秒能传输 $1.602\,5\times10^6$ B. 因此，本文方法在测试中，相比于 FISCO BCOS 的同步数据交换方法，可实现约 561 倍传输数据量的性能提升；同时本文方法相比于 FISCO BCOS 的异步数据交换方法，可实现约 2.98 倍传输数据量的性能提升.

4.4.2 区块链数据要素交换方法的性能压力测试

为有效评估本文提出的区块链加速交换方法所能支持的最大传输能力，同时结合实验中测量到 FISCO BCOS 的异步数据交换方法不能较好地处理大量数据交换请求的情况，因此在性能压力测试实验中，本文方法的同步间隔选取了有比较价值的 10 ms、50 ms、100 ms 以及 200 ms 共 4 个测试条件. 此外，为有效进行压力测试，并结合测试过程中对测试数据的观察，最后确定选取线程数 1~6 的测试方案进行测试，以获取本文方案在不同测试压力中的性能. 另外，在压力测试中，本文还选取了 FISCO BCOS 的同步数据交换方法及异步数据交换方法进行对比. 其中，在 FISCO BCOS 的同步数据交换方法中，直接选取与本文测试所用最大数量线程数为 6 时进行对比；而在与 FISCO BCOS 的异步数据交换方法对比中，由于 FISCO BCOS 的异步数据交换方法在接收大量的请求后会出现连接异常问题，因此在实验中仅获取了

测试线程数为 1 时的有效测试结果. 关于压力测试的测试结果如图 9 和图 10 所示, 本文方法在不同同步间隔的性能对比如表 5 所示, 不同线程的压力测试的性能对比如表 6 所示.

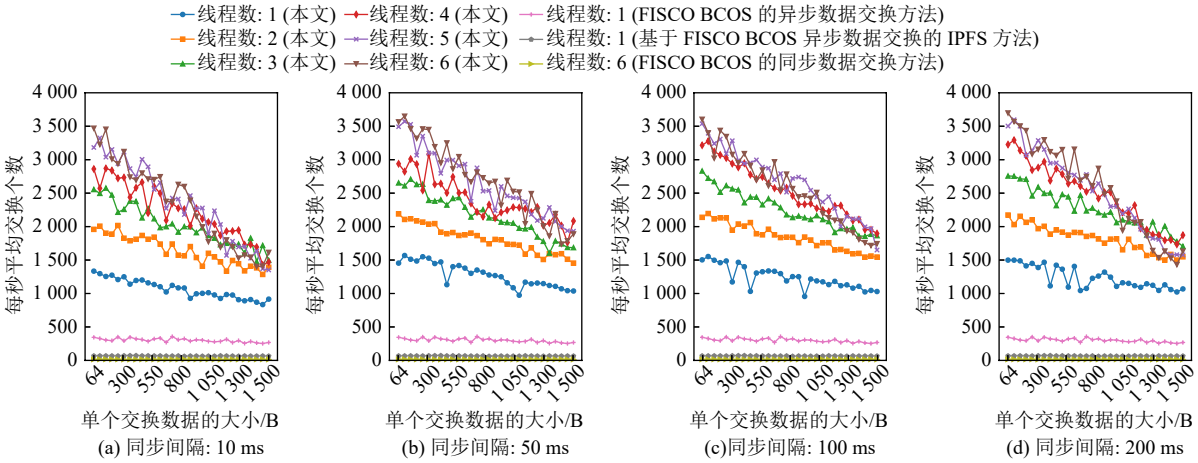


Fig. 9 The number of exchanging data in stress tests

图 9 压力测试中交换数据的个数

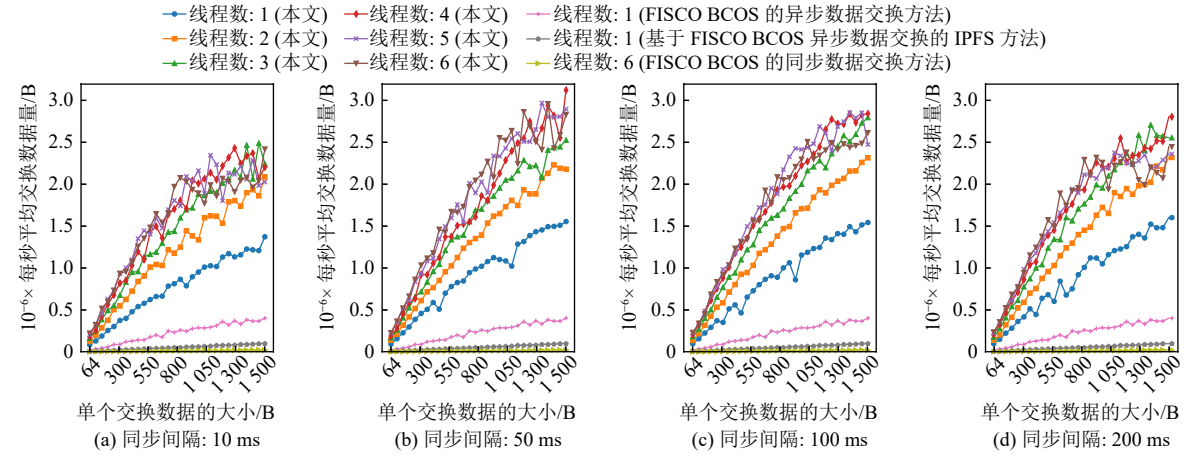


Fig. 10 Performance comparison of exchanging data amounts in stress tests

图 10 压力测试中交换数据量的性能对比

如表 5 和图 9 压力测试的交换数据个数性能测试结果所示, 在 6 个测试线程中, 同步间隔为 10 ms 时, 最大的每秒平均数据交换个数为 3 470.05; 同步间隔为 50 ms 时, 最大的每秒平均数据交换个数为 3 649.45 个; 同步间隔为 100 ms 时, 最大的每秒平均数据交换个数为 3 603.35; 同步间隔为 200 ms 时, 最大的每秒平均数据交换个数为 3 697.75. 从而可以得知在同步间隔时间过短时会出现性能下降.

此外, 当单个交换数据大小为 1 500 B 时, 在同步间隔为 10 ms 测试中, 最大的每秒平均数据交换个数为 1 614; 在同步间隔为 50 ms 测试中, 最大的每秒平均数据交换个数为 2 083.55; 在同步间隔为 100 ms 测试中, 最大的每秒平均数据交换个数为 1 896.4; 同步间隔为 200 ms 测试中, 最大的每秒平均数据交换个数为 1 871.8. 从而依据实验结果可知, 设置较大的同步间隔时, 不仅会出现数据由于同步时间过长而出现传输延时硬性增加, 同时也会导致传输带宽下降.

因此依据上述实验结果, 本节在后续实验对比中, 选取同步间隔为 50 ms 的实验数据进行对比, 关键实验结果如表 6 所示. 在测试线程数为 1~6 时, 最大平均每秒数据交换个数分别为 1 567.9, 2 189.2, 2 707.65, 3 082.65, 3 572 以及 3649.45. 另外, FISCO BCOS 的同步数据交换方法的最大每秒数据交换个数为 16.5; 基于 FISCO BCOS 异步数据交换的 IPFS 方法最大每秒数据交换个数为 72.15; FISCO BCOS 的异步数据交换方法的最大每秒数据交换个数为 356.2. 此外, 依据测试线程数为 5 和 6 的实验数据可知它们每秒平均交换数据个数差异不大, 因此在高

Table 5 Performance Comparison of Our Method for Different Synchronization Intervals

表 5 本文方法在不同同步间隔的性能对比

本文方法	最大的每秒平均交换数据的个数	单个交换数据为 1 500 B 时，最大的每秒平均交换数据数	最大的每秒平均交换数据量/B
同步间隔 0 ms（仅线程数为 1 时数据可用）	357.00	290.15	435 225
同步间隔 10 ms	3 470.05	1 614.00	2 491 680
同步间隔 50 ms	3 649.45	**2 083.55**	**3 125 325**
同步间隔 100 ms	3 603.35	1 896.40	2 857 072
同步间隔 200 ms	**3 697.75**	1 871.80	2 807 700

注：黑体为最佳数值.

Table 6 Performance Comparison of Stress Tests with Different Threads

表 6 不同线程的压力测试的性能对比

方法	每秒平均交换数据的个数		每秒平均交换数据量	
	最大平均个数	提升倍数	实验中最大平均值/B	提升倍数
本文方法 同步间隔 50 ms，线程数为 1	1 567.9	3.401 7	1 555 950	2.869 0
本文方法 同步间隔 50 ms，线程数为 2	2 189.2	5.145 9	2 232 650	4.551 7
本文方法 同步间隔 50 ms，线程数为 3	2 707.65	6.601 4	2 528 550	5.287 5
本文方法 同步间隔 50 ms，线程数为 4	3 082.65	7.654 2	**3 125 325**	**6.771 5**
本文方法 同步间隔 50 ms，线程数为 5	3 572	9.028 0	2 973 035	6.392 8
本文方法 同步间隔 50 ms，线程数为 6	**3 649.45**	**9.245 5**	2 961 967	6.365 3
FISCO BCOS 的异步数据交换方法（线程数为 1）	356.2	0	402 150	0
FISCO BCOS 的同步数据交换方法（线程数为 6）	16.5	–0.953 6	23 562	–0.941 4
基于 FISCO BCOS 异步数据交换的 IPFS 方法（线程数为 1）	72.15	–0.797 4	97 275	–0.758 1

注：黑体为最佳数值，提升倍数表示相比 FISCO BCOS 的异步数据交换方法的性能.

压力情况下，增加线程数也不会带来明显的性能提升. 关于数据加速交换提升方面，在测试线程数为 6 时，本文方法的数据交换个数相比于测试线程数为 1 时，能提高 1.32 倍的交换效率. 另外，本文方法在测试线程数量为 6 时，相比于 FISCO BCOS 的同步数据交换方法在测试线程数为 6 时，平均每秒交换个数能获得 220.17 倍提升；相比于基于 FISCO BCOS 异步数据交换的 IPFS 方法在测试线程数为 1 时，平均每秒交换个数能获得 49.58 倍提升；相比于 FISCO BCOS 的异步数据交换方法在测试线程数为 1 时，平均每秒交换个数能获得 9.24 倍提升.

此外每秒平均交换数据量是衡量使用区块链进行数据要素流通的一个重要指标，如表 5 和图 10 压力测试的交换数据容量性能测试结果所示，同步间隔为 10 ms 时，最大平均每秒交换数据量为 $2.491\,6\times10^6$ B；同步间隔为 50 ms 时，最大平均每秒交换数据量为 $3.125\,3\times10^6$ B；同步间隔为 100 ms 时，最大平均每秒交换数据量为 2.857×10^6 B；同步间隔为 200 ms 时，最大平均每秒交换数据量为 $2.807\,7\times10^6$ B. 因此，同步间隔为 50 ms 时，能获取最佳的性能. 同时，此结果也与交换数据个数性能测试结果一致. 从而设置一个恰当的同步间隔能使数据加速效率最大化.

在同步间隔为 50 ms 的测试中，其关键实验结果如表 6 所示，测试线程数从 1~6 的最大平均每秒交换数据量分别为 $1.555\,9\times10^6$ B、$2.232\,6\times10^6$ B、$2.528\,5\times10^6$ B、$3.125\,3\times10^6$ B、2.973×10^6 B 以及 $2.961\,9\times10^6$ B. 此外，FISCO BCOS 的同步数据交换方法，在测试线程数为 6 的情况下，最大平均每秒交换数据量约为 2.35×10^4 B；基于 FISCO BCOS 异步数据交换的 IPFS 方法，在测试线程数为 1 情况下，最大平均每秒交换数据量约为 9.72×10^4 B；FISCO BCOS 的异步数据交换方法，在测试线程数为 1 情况下，最大平均每秒交换数据量约为 4.021×10^5 B. 从而本文提出的方法能有效支持并发测试，在资源有限的情况下本文方法在测试线程数为 4 时，相比于测试线程数为 1 时的数据交换量能提升约 1.01 倍性能. 此外，本文方法相比于测试线程数为 1 的基于 FISCO BCOS 异步数据交换的 IPFS 方法，能获得约 31.12 倍性能提升；本文方法相比于测试线程数为 6 的 FISCO BCOS 的同步数据交换方法，能获得约 131.63 倍性能提升. 另外，本文方法相比于测试线程数为 1 的 FISCO BCOS 的异步数据交换方法，能获得约 6.77 倍性能提升.

4.5 实验总结

依据实验中设置不同参数的性能测试结果，均表明本文方法能在区块链自身已经支持交易打包的情况下，获取额外的数据交换性能. 在本文设计的所有有效测试中，本文方法相比于 FISCO BCOS 所提供的同步数据交换方法能获得最高约 803 倍的性能提升；同时，本文方法相比于 FISCO BCOS 所提供的异步数据交换方法能获得最高约 9.24 倍性能提升. 此外，本文方法也与当前主流的 IPFS 方案进行对比实验，由于 IPFS 方案主要针对大文件传输进行优化，其在较小单次测试数据中性能反而出现下降. 因此，本文方法能够一定程度上缓解当前基于区块链数据要素流通架构实际应用面临的挑战.

5 总结与下一步工作

为解决基于区块链的数据要素流通在生产过程中面临的交换速率问题,本文提出了区块链数据加速交换方法.其中动态位置模型能有效解决数据流通过程中的数据要素使用方出现变更的问题,同时聚合上传和下载模型也能有效对数据要素进行加速交换.最后,本文通过一系列的实验证明本文方法能进一步提升基于区块链的数据要素交换性能.依据实验测量结果,本文方法相比于区块链框架所提供的原生方法,最高可提升9.24倍传输性能,能更进一步满足生产过程中数据要素高性能交换的需求.

随着新型技术的不断发展,需要持续研究新的方法以适应更多场景.本文方法仅针对使用区块链进行数据要素交换开展研究,而对于数据要素提供设备以及使用设备,它们仍需要编写特定程序完成注册等工作.因此,在未来的工作中,拟进一步对数据要素透明流通方法开展研究,引入软件定义网络、人工智能等技术,实现终端简易配置的数据要素交换,完成数据要素流通过程中的各类型终端自动化接入数据要素传输系统.

作者贡献声明:李振宇提出了方法思路和实验方案,并撰写论文;丁勇针对论文提出了修改意见;易晨和梁海针对论文提出了修改意见.

参 考 文 献

[1] Fan Feifei. Moves on data set to drive digital fields[EB/OL]. (2022-12-21)[2024-07-10]. https://www.chinadaily.com.cn/a/202212/21/WS63a25d7da31057c47eba5687.html

[2] National Data Agency of the People's Republic of China. Circular of the seventeen ministries on the issuance of the three-year action plan on "data elements x" (2024−2026)[EB/OL]. (2024-01-05)[2024-07-10]. https://www.cac.gov.cn/2024-01/05/c_1706119078060945.htm (in Chinese)
(国家数据局. 十七部门关于印发《“数据要素×”三年行动计划(2024—2026年)》的通知[EB/OL]. (2024-01-05)[2024-07-10]. https://www.cac.gov.cn/2024-01/05/c_1706119078060945.htm.)

[3] Bao Xiaoli, Du Wanli. Institutional construction of trusted data trading system: Based on the perspective of on exchange trading[J]. E-Government, 2023, 6(6): 38−50 (in Chinese)
(包晓丽,杜万里. 数据可信交易体系的制度构建——基于场内交易视角[J]. 电子政务, 2023, 6(6): 38−50)

[4] Du Ziran, Dou Yue, Yi Chengqi, et al. TID-MOP: The comprehensive framework of security management and control in the scenario of data exchange[J]. Data Analysis and Knowledge Discovery, 2022, 6(1): 13−21 (in Chinese)
(杜自然,窦悦,易成岐,等. TID-MOP: 面向数据交易所场景下的安全管控综合框架[J]. 数据分析与知识发现, 2022, 6(1): 13−21)

[5] Song Rui, Xiao Bin, Song Yubo, et al. A survey of blockchain-based schemes for data sharing and exchange[J]. IEEE Transactions on Big Data, 2023, 9(6): 1477−1495

[6] Ma Yuhang, Zhang Liang, Wu Xingyu, et al. Multi-party cross-chain transaction scheme based on distributed key generation and attribute-based encryption[J]. Journal of Computer Research and Development, 2023, 60(11): 2534−2544 (in Chinese)
(马宇航,张亮,吴星雨,等. 基于分布式密钥生成和属性基密码的多方跨链交易方案[J]. 计算机研究与发展, 2023, 60(11): 2534−2544)

[7] Liu Tengfei, Guo Yang, Wang Jingjing. The blockchain technology empowers the path of data element value circulation[J]. Telecom Information, 2023, 11: 39−42 (in Chinese)
(刘腾飞,郭洋,王菁菁. 区块链技术赋能数据要素价值流通的路径[J]. 电信快报, 2023, 11: 39−42)

[8] Singh A, Kumar G, Saha R, et al. A survey and taxonomy of consensus protocols for blockchains[J]. Journal of Systems Architecture, 2022, 127(102503): 1−27

[9] Wang Wenbin, Liu Xiulong, Xu Hao, et al. An effective and robust transaction packaging approach for multi-leader BFT blockchain systems[C]//Proc of 2023 42nd Int Symp on Reliable Distributed Systems. Piscataway, NJ: IEEE, 2024: 14−24

[10] Liu Xinmeng, Xie Haomeng, Yan Zheng, et al. A survey on blockchain sharding[J]. ISA Transactions, 2023, 141: 30−43

[11] Dong Sihan, Xin Junchang, Hao Kun, et al. A join query optimization algorithm in multi-blockchain environment[J] Journal of Zhejiang University (Engineering Science), 2022, 56(2): 313−321 (in Chinese)
(董思含,信俊昌,郝琨,等. 多区块链环境下的连接查询优化算法[J]. 浙江大学学报: 工学版, 2022, 56(2): 313−321)

[12] Zhou Chuangming, Yang Zhou, Yue Shaohua, et al. Securing big data exchange: An integrated blockchain framework for full-lifecycle data trading with trust and dispute resolution[J]. Electronics, 2023, 12(4894): 1−24

[13] Sober M, Scaffino G, schulte S, et al. A blockchain-based IoT data marketplace[J]. Cluster Computing, 2023, 26(6): 3523−3545

[14] Su Xin, Ullah I, Wang Meiling, et al. Blockchain-based system and methods for sensitive data transactions[J]. IEEE Consumer Electronics Magazine, 2024, 13(2): 87−96

[15] Ji Shu, Zhang Jinlin. Cross-domain secure circulation system of data elements based on blockchain + privacy computing[J]. Network Security Technology & Application, 2023, 5: 65−67 (in Chinese)
(季姝,张金琳. 基于区块链+隐私计算的数据要素跨域安全流通体系[J]. 网络安全技术与应用, 2023, 5: 65−67)

[16] Tao Qi, Cui Xiaohui. B-FLACS: Blockchain-based flexible lightweight access control scheme for data sharing in cloud[J]. Cluster Computing, 2023, 26(6): 3931−3941

[17] Wang Dong, Li Da, Feng Jingli, et al. Blockchain-based data element resources sharing and access control scheme[J]. Journal of

Information Security Research, 2023, 9(2): 137−145 (in Chinese)
(王栋, 李达, 冯景丽, 等. 基于区块链的数据要素资源共享及访问控制方案[J]. 信息安全研究, 2023, 9(2): 137−145)

[18] Chen Jun, Wu Jiangyao, Qian Zhenghao, et al. Industrial chain data sharing and circulation of blockchain and big data technology[J]. Wireless Communications and Mobile Computing, 2022, 2022 (7719036): 1−11

[19] An Baoyi, Xiao Mingjun, Liu An, et al. Secure crowdsensed data trading based on blockchain[J]. IEEE Transactions on Mobile Computing, 2023, 22(3): 1763−1778

[20] Han Rui, Wang Yu, Wan Mingfa, et al. FIBPRO: Peer-to-peer data management and sharing cloud storage system based on blockchain[J]. Peer-to-Peer Networking and Applications, 2023, 16(6): 2850−2864

[21] Zhu Li, Yu Huan, Zhan Shixiao, et al. Research on high-performance consortium blockchain technology[J]. Journal of Software, 2019, 30(6): 1577−1593 (in Chinese).
(朱立, 俞欢, 詹士潇, 等. 高性能联盟区块链技术研究[J]. 软件学报, 2019, 30(6): 1577−1593)

[22] Wang Tianyu, Chen Zizhan, Zhu Wenbin, et al. A Bloom-filter-based unique address checking approach for DAG-based blockchain systems[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2023, 43(1): 85−98

[23] Qi Yuanyuan, Shao Sujie, Wu Shuang, et al. A distributed intelligent service trusted provision approach for IoT[J]. IEEE Internet of Things Journal, 2023, 10(24): 22341−22355

[24] Lin Zhikai, Wang Kexing, Wu Yongdong, et al. Fast payments across heterogeneous blockchains for Internet of things[J]. IEEE Access, 2024, 12(2): 33923−33937

[25] Kong Yihan, Li Jing, Xiong Ting, et al. EVONChain: A bi-tiered public blockchain network architecture[J]. Peer-to-Peer Networking and Applications, 2023, 16(6): 2892−2914



Li Zhenyu, born in 1992. PhD candidate. Student member of CCF. His main research interests include cyberspace security and network engineering.
李振宇, 1992 年生. 博士研究生. CCF 学生会员. 主要研究方向为网络空间安全、网络工程.



Ding Yong, born in 1975. PhD, professor. Senior member of CCF. His main research interests include cryptography and blockchain.
丁　勇, 1975 年生. 博士, 教授. CCF 高级会员. 主要研究方向为密码学、区块链.



Yi Chen, born in 1981. Master, professor, professor of engineering. His main research interest includes cyberspace security.
易　晨, 1981 年生. 硕士, 教授, 正高级工程师. 主要研究方向为网络空间安全.



Liang Hai, born in 1982. Master, associate professor. His main research interests include cyberspace security and blockchain.
梁　海, 1982 年生. 硕士, 副教授. 主要研究方向为网络安全、区块链.