计算机研究与发展
Journal of Computer Research and Development
https://crad.ict.ac.cn
63(5): 1319−1336, 2026

兼容智能合约的区块链分片技术综述

张滋炯[1] 韩昊翔[1] 史晁铭[1] 闫 峥[1,2] 王海光[3] 李铁岩[3]
[1]（西安电子科技大学网络与信息安全学院 西安 710216）
[2]（西安电子科技大学杭州研究院 杭州 311231）
[3]（华为技术有限公司可信技术实验室谢尔德实验室 新加坡 238895）
（zhangzt@stu.xidian.edu.cn）

Survey on Blockchain Sharding Technologies with Smart Contract Compatibility

Zhang Zitong[1], Han Haoxiang[1], Shi Chaoming[1], Yan Zheng[1,2], Wang Haiguang[3], and Li Tieyan[3]
[1]（*School of Cyber Engineering*, *Xidian University*, *Xi'an* 710126）
[2]（*Hangzhou Institute of Technology*, *Xidian University* , *Hangzhou* 311231）
[3]（*Shield Lab*, *Trustworthiness Lab*, *Huawei Technologies Co. Ltd.*, *Singapore* 238895）

Abstract Blockchain sharding, which divides a blockchain network into multiple shards to process transactions in parallel, is widely considered to be one of the most effective techniques for enhancing blockchain scalability and system performance. However, existing sharding schemes primarily focus on optimizing point-to-point asset transfer transactions and exhibit significant limitations in supporting Turing-complete smart contracts. Since executing the business logic based on Turing-complete smart contracts typically involves initiating multiple internal transactions to synchronize state across shard, designing the schemes of blockchain sharding with smart contract compatibility (BS-SCC) presents significant and unique challenges. Although there are some studies in the literatures, there is still a lack of a systematic survey to review the academic progress and research landscape. This paper provides a comprehensive survey on existing BS-SCC schemes. First, we introduce the research motivation and basic concepts of BS-SCC. Next, we propose a taxonomy of BS-SCC schemes based on the dimensions of transaction execution and state storage, categorizing them into such classes as on-chain and off-chain execution models. Then, we propose a set of criteria for evaluating BS-SCC schemes that encompass reliability, effectiveness, and other performance properties. By employing the proposed criteria, we furthermore conduct a multi-dimensional comparative analysis of existing schemes. Finally, we identify a list of technical bottlenecks and suggest corresponding future research directions to offer theoretical insights and practical guidance for advancing BS-SCC.

Key words blockchain; blockchain sharding; smart contract; cross-shard transaction; off-chain scaling

摘 要 区块链分片技术通过网络分区实现交易的并行处理，被公认为是提升区块链可扩展性和系统性能的最有效解决方案之一。然而，现有分片方案主要针对用户间点对点的资金转移交易，在支持图灵完备智能合约方面存在显著局限性。由于执行基于图灵完备智能合约的业务逻辑通常需要发起多笔内部交易来同步状态，设计兼容智能合约的区块链分片（blockchain sharding with smart contract compatibility, BS-

收稿日期：2025-05-07；**修回日期**：2025-11-18
基金项目：国家自然科学基金项目 (U23A20300)；陕西省自然科学基金项目 (2023-JC-ZD-35)；西安电子科技大学杭州研究院项目 (GNYZ2024XX007)；高等学校学科创新引智计划项目 (B16037)
This work was supported by the National Natural Science Foundation of China (U23A20300), the Natural Science Foundation of Shaanxi Province (2023-JC-ZD-35), the Project of Hangzhou Institute of Technology of Xidian University (GNYZ2024XX007), and the Higher Education Discipline Innovation Introduction Plan Project (B16037).
通信作者：闫峥（zyan@xidian.edu.cn）

SCC)架构面临重大独特的技术挑战。尽管该领域已有大量研究成果,但目前仍缺乏系统性综述来梳理学术进展和研究脉络,为此对BS-SCC方案进行全面的文献综述。首先,深入剖析BS-SCC的研究背景与核心概念;其次,构建基于交易执行和状态存储维度的BS-SCC分类框架,将现有方案分类为链上和链下执行模型;随后,提出包含可靠性、有效性和其他关键性能指标的评估体系;进而,运用该评估体系对现有方案进行多维度比较分析;最后,识别当前研究中的技术瓶颈,并指出具有潜力的未来研究方向,旨在为学术界和工业界提供有价值的理论指导和技术参考。

关键词 区块链;区块链分片;智能合约;跨分片交易;链下扩容

中图法分类号 TP311

DOI: 10.7544/issn1000-1239.202550305 **CSTR:** 32373.14.issn1000-1239.202550305

区块链作为去中心化的分布式账本技术,凭借去中心化、不可篡改、透明和可追溯等核心优势,被广泛应用于金融、供应链、信任管理等多个领域。随着技术的发展,区块链2.0引入了图灵完备智能合约,能够处理复杂业务逻辑,进一步拓展了其在多领域的发展潜力。目前,区块链的发展已经历“加密货币”(区块链1.0)和“可编程金融”(区块链2.0)两个阶段,逐步迈向“可编程社会”(区块链3.0)[1]。然而,区块链的性能瓶颈限制了其在大规模场景中的应用。例如以太坊[2]的交易处理能力仅为每秒10~20笔,而VISA[3]等传统电子支付系统的吞吐量高达56 000 TPS(transactions per second)。这一性能差距源于区块链的共识机制,即每笔交易需由全网节点验证和存储,随着网络规模和交易量的增长,交易确认时延与通信开销会显著增加。

区块链分片技术作为提升区块链可扩展性和系统性能的关键方案,核心思想是将区块链网络划分为多个分片,每个分片并行处理交易并达成共识[4]。然而,当前分片方案专注于优化点对点的资金转移,在支持基于图灵完备智能合约的复杂业务逻辑执行方面存在显著不足。通常,同一智能合约的代码与状态部署于同一分片,不同智能合约的代码及其关联状态则分布在不同分片。当合约执行需调用其他分片的合约或访问其他分片状态时,则会触发跨分片交易。以电子商务为例,用户支付合约及余额状态部署于分片A,商品管理合约及状态位于分片B。要实现完整的订单处理逻辑,分片A需向分片B发起跨分片合约调用请求。本文提出的“兼容智能合约的区块链分片”(blockchain sharding with smart contract compatibility, BS-SCC)是能够支持图灵完备智能合约的业务逻辑完整执行的分片架构。即使涉及业务逻辑的智能合约分布在不同分片,BS-SCC也能确保系统状态的一致性。此外,考虑到图灵完备智能合约涉及复杂的计算逻辑和频繁的合约间数据传递,BS-SCC架构也需兼顾系统执行效率。

截至目前,已有研究分别对智能合约与区块链分片技术展开综述。关于智能合约,Wang等人[5]系统梳理了智能合约的概念、架构、应用及未来发展趋势。随后,Wang等人[6]介绍了智能合约的运行逻辑和主流平台,并提出了一种基于6层架构的智能合约研究框架。Hewa等人[7]关注智能合约在典型场景中的应用。Zheng等人[8]重点分析了智能合约的工作机制和当前面临的挑战。关于区块链分片技术,Liu等人[9]将现有分片系统拆解为不同的功能模块,并基于系统模型进行分类。Liu等人[10]根据区块链类型与分片策略对现有方案分类,并提出了一套包括可扩展性、适用性及可靠性的评估体系。Li等人[11]分析了分片系统的关键构建模块,同时提出一些抗攻击的安全策略,Yang等人[12]重点探讨了各分片方案的性能表现,并提出了对应的优化建议。唐海波等人[13]从节点分配安全性、链上数据分片高效性及跨片交易处理3个维度,对现有方案展开分类与评估。

表1总结和对比了本文与相关综述的贡献。结果显示:尽管已有研究对智能合约和区块链分片技术进行了独立综述,但目前仍缺乏对BS-SCC的系统性综述。此外,现有分片技术的分类和评估主要针对逻辑简单的点对点资金转移,未充分考虑图灵完备智能合约执行所需的合约调用和复杂状态管理特性,难以直接指导BS-SCC的后续研究。为填补这一研究空白,本文对BS-SCC技术进行了全面综述,主要贡献包括:

1)作为BS-SCC综述论文,系统介绍BS-SCC的研究背景和基本概念,以弥补当前该域缺乏系统性综述的研究现状。

2)提出BS-SCC的分类体系,基于执行方式和状态存储,将BS-SCC分为2个大类和6个小类,为系

Table 1 Comparison of Our Survey and Existing Related Surveys

表 1 本文综述与现存相关综述对比

类型	参考文献	主题	①	②	③	④
智能合约综述	文献 [5]	智能合约的概念、架构、应用和挑战	√	×	×	×
	文献 [6]	智能合约运行机制和主流平台	√	×	×	×
	文献 [7]	智能合约的重要应用	√	×	×	×
	文献 [8]	智能合约的工作机制和未来挑战	√	×	×	×
区块链分片综述	文献 [9]	区块链分片的功能组件	×	×	×	×
	文献 [10]	区块链分片的分类和评估	×	×	×	×
	文献 [11]	区块链分片的关键组件和攻防策略	×	×	×	×
	文献 [12]	区块链分片系统的性能	×	×	×	×
	文献 [13]	区块链分片的功能组件和评估	×	×	×	×
BS-SCC 技术综述	本文	兼容智能合约的区块链分片概念、分类和评估	√	√	√	√

注: “√”表示满足; “×”表示不满足; ①表示智能合约的基本概念; ②表示 BS-SCC 架构及概念; ③表示 BS-SCC 分类体系; ④表示 BS-SCC 多维度评估框架。

统性理解不同研究方案的设计思路与技术差异提供理论基础。

3)构建 BS-SCC 方案的多维度评估框架涵盖可靠性、有效性等多方面指标, 从定性与定量 2 个角度评估现有方案的优劣。

4)结合文献综述与评估标准, 明确现有 BS-SCC 研究的未解问题, 并指出未来研究方向, 为后续相关研究提供新视角与理论指引。

1 背景知识

本节介绍与 BS-SCC 密切相关的背景知识。这些基本概念为 BS-SCC 的构建提供了理论支持。此外, 提出一个新颖且全面的 BS-SCC 分类框架来指引本文的综述调研, 如表 1 所示。

1.1 区块链架构

区块链作为去中心化的分布式账本技术, 融合了分布式共识机制、密码学、智能合约等多项关键技术。该技术采用链式数据结构来验证和存储数据, 借助分布式共识机制生成并更新数据, 运用加密技术保障数据传输与访问的安全性, 还依靠智能合约实现数据操作的自动化与可编程性, 构建出全新的分布式基础设施[8]。

区块链的概念起源于 Bitcoin[14], 此名称来源它独特的数据结构: 每一个区块都依靠密码学算法与前一个区块相互连接, 形成链式结构。每个区块由块体与块头 2 部分构成, 其中块体记录着交易数量和具体的交易内容, 块头则存储元数据, 包含 Previous Hash, Merkle Root, Timestamp, Nonce 等信息, 具体字段组成及相应解释如表 2 所示。在 Bitcoin 共识机制中, 矿工节点通过调整 Nonce 参与计算, 尝试生成一个符合特定难度目标的哈希值。

Table 2 Block Header Field Composition and Their Interpretation

表 2 区块头字段组成及相应解释

字段名称	解释说明
Version	区块链协议的版本
Previous Hash	前一区块哈希值
Merkle Root	所有交易哈希值的根哈希
Timestamp	该区块的创建时间戳
Difficulty Target	该区块的工作量证明难度目标
Nonce	证明工作量的随机性整数

分布式共识机制是区块链系统运行的关键, 用于解决去中心化网络中各节点就交易记录和账本全局状态方面难以达成一致的问题。当前常见的分布式共识机制包括工作量证明(proof of work, PoW)[15]、权益证明(proof of stake, PoS)[16] 和拜占庭容错(Byzantine fault tolerance, BFT)[17]。这些机制依据预先设定的规则与协议, 使网络中各节点就新区块的生成与验证达成共识, 以此保证整个系统安全稳定运行。例如, PoW 要求节点通过解决复杂计算难题(如哈希计算)竞争新区块的生成权, 最先完成计算的节点获得记账权及相应奖励; PoS 则依据节点所持代币数量及其持有时长分配出块权, 持有代币数量越多和持有时间越长的节点被选中的概率越高; 而 BFT 凭借节点间的多轮投票与协商, 即使网络中存在恶意节点或者发生故障, 依然可就交易达成共识。

1.2 区块链交易

在区块链系统中,交易是指资金转移或信息记录等行为,例如代币转移、图灵完备智能合约的执行。所有交易数据被写入区块的块体部分,通过共识算法完成验证与确认。一旦交易被打包进区块并添加至区块链中,其记录将被永久保留,不可篡改。

区块链的交易模型主要分为2类:未使用交易输出(unspent transaction output, UTXO)[14]模型和账户/余额模型[2]。在UTXO模型中,每笔交易依赖于前一笔或多笔交易的输出,并生成新的输出供后续交易使用。每笔交易需明确指定输入来源和新生成的输出。而账户/余额模型则为每个用户分配固定的账户地址,通过更新账户余额来完成交易[2]。此外,账户/余额模型中的余额概念可扩展为复杂状态信息,以支持智能合约等可编程应用逻辑,如图1所示。

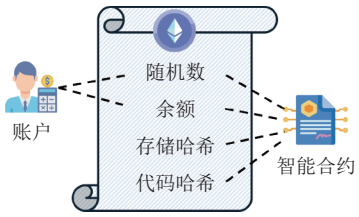


Fig. 1 Types of Ethereum transactions

图1 以太坊交易类型

基于账户/余额模型的交易可根据交易发送方的属性分为2类:

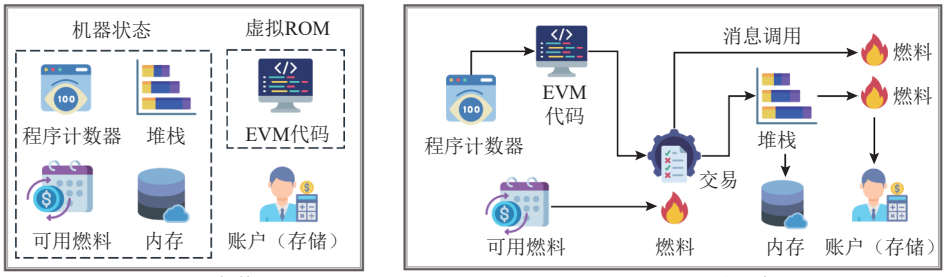
1)外部交易。用户与区块链平台之间的直接交互,包括资金转移、创建和调用智能合约。此类交易的发送方地址始终为用户账户,而接收方地址则根据交易类型不同有所区别:若为资金转移,则接收方地址是另一用户账户;若为智能合约的创建或调用,接收方地址是智能合约地址。

2)内部交易。智能合约之间的相互调用用于实现合约间的交互逻辑。内部交易的发送方和接收方都是合约地址,分别对应调用方和被调用方合约。内部交易通常在外部交易触发合约执行时自动生成,支持跨合约的逻辑传递,从而增强区块链系统的功能扩展性与逻辑性。

1.3 智能合约

智能合约是一种运行在区块链上的计算机程序,以数字化方式验证并自动执行多方之间的协议[18]。智能合约在区块链的首次应用出现于比特币,比特币系统采用了一种基于堆栈的脚本语言,其语法类似于逆波兰表示法,但由于该语言非图灵完备,不支持循环等控制结构,因而难以实现复杂的业务逻辑,仅能支持点对点的资金转移[5]。相比之下,以太坊等区块链平台所支持的图灵完备智能合约具备更强的功能,不仅可以存储合约状态信息,还嵌入了可执行代码。例如,在以太坊区块链上,去中心化金融(decentralized finance, DeFi)的智能合约可实现自动化的借贷协议:用户通过智能合约存入资金作为抵押,合约根据预定义规则自动执行借贷、利息计算及清算操作,确保交易透明且无需中介。若无特殊说明,后文提到的智能合约均指图灵完备智能合约。

在以太坊中,智能合约的运行机制如下:以太坊的智能合约子系统会定期遍历每个合约的状态机和触发条件。当合约满足触发条件时,它会被推到交易队列中等待验证。合约经验证并达成共识后执行,其状态更新会随即永久写入区块链。目前,智能合约的链上执行环境主要分为虚拟机和容器2类。两者均采用隔离的"沙盒"结构,通过限制和隔离资源来确保智能合约的安全执行。为了优化资源利用和提升性能,大多数区块链平台采用轻量级虚拟机架构,例如以太坊虚拟机(Ethereum virtual machine, EVM)。EVM是一种基于堆栈的虚拟机,为智能合约提供了灵活的执行环境,并支持处理复杂的逻辑运算,其架构与运行逻辑如图2所示。而容器则基于内核虚拟化技术,通过轻量级虚拟化实现进程与资源的隔离。例如,Hyperledger Fabric[19]利用Docker容器执行智能



(a) EVM架构 (b) EVM运行逻辑

Fig. 2 EVM architecture and execution logic

图2 EVM架构及运行逻辑

合约代码。由于容器直接运行于底层操作系统之上，因此相比 EVM，Docker 容器的执行效率通常更高。然而，Docker 容器存在安全漏洞，还需进一步强化安全防护措施，才能有效抵御潜在攻击。

区块链的去中心化特性要求智能合约在所有节点上重复验证，这会让交易数据对全网公开，进而泄露敏感信息。此外，计算复杂度高的智能合约会增加执行成本，例如以太坊中需支付高额的 Gas 费用。同时，智能合约需在多节点间重复执行，因此交易验证过程通常耗时较长。为解决这些挑战，研究者提出了链下执行策略。链下执行允许智能合约的计算逻辑在区块链外部环境中运行，仅将关键状态信息和最终执行结果提交至链上[20]。这种机制在确保交易隐私的同时也降低了计算成本。当链下执行的智能合约发生争议，还可以在区块链上重新运行链下逻辑，进而验证并解决争议。

目前，链下执行主要依赖可信执行环境（trusted execution environment, TEE）[21]。TEE 利用硬件安全特性，为在不可信平台上运行的可信代码提供安全的隔离执行环境，确保代码和数据不被未经授权访问或篡改，如 Intel SGX[22]。此外，TEE 的计算效率较高，支持复杂加密操作，能为智能合约执行提供强大的计算支持。

1.4 区块链分片

区块链分片技术分为网络分片、交易分片和状态分片 3 种类型[9]，如图 3 所示。这 3 种分片方式协同工作，共同提升区块链的可扩展性和并行处理能力。

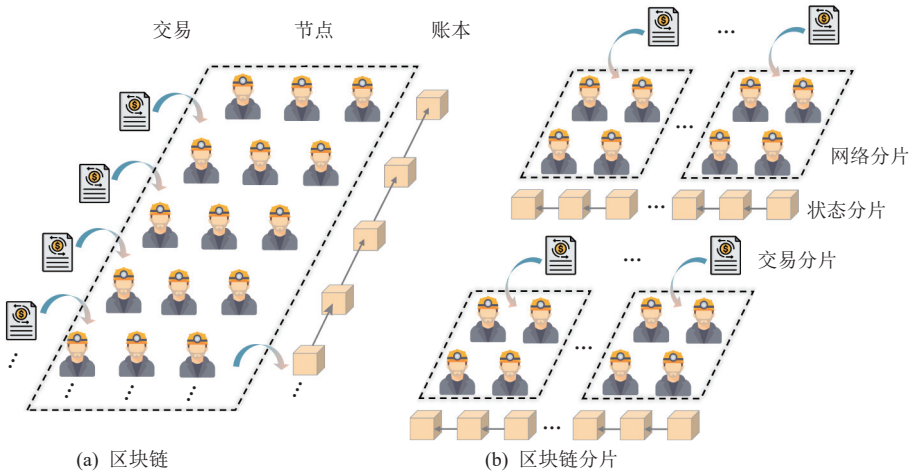


Fig. 3 Comparison of blockchain and sharded blockchain architecture

图 3 区块链与区块链分片架构对比

网络分片按照特定方式将整个区块链网络划分为多个较小的分片，每个分片由部分节点组成，且恶意节点数量要低于共识协议的安全阈值。这种方式显著降低了全网达成共识的成本，为交易分片和状态分片提供了必要的底层架构支持。

交易分片将账户、智能合约及相关交易分配至不同的分片，每个分片仅负责验证和执行部分交易。此外，分片间协作处理跨分片交易，以此确保区块链系统的一致性。

状态分片将区块链的全局状态分割为多个独立部分，每个分片仅维护其负责的状态数据。当交易需要访问其他分片的状态时，分片间会发起跨分片状态来同步数据。状态分片有效减轻了单个节点的存储和计算压力，提升了区块链的可扩展性。

尽管区块链分片技术能明显提升系统性能与扩展性，但跨分片交易的处理也带来了额外挑战。每个分片仅存储和维护全局状态的一部分，跨分片交易需要多个分片相互协调才能保证系统一致性，这个过程难免会增加通信开销与确认时延[23]。目前，处理用户间跨分片交易的常见方法是将交易拆分为若干子交易，相关分片独立执行各自的子交易以实现并行处理[24-25]。仅当所有子交易在相关分片中完成验证后，跨分片交易才会被最终提交。为确保账本一致性，交易执行期间，所有受影响的账户或状态将被锁定，直至跨分片交易完成并成功提交。

在 BS-SCC 架构中，跨分片交易的处理尤为复杂。智能合约交易通常包含明确的业务逻辑，要求分布在不同分片的合约按业务逻辑执行。然而，传统分片的处理方式要求各分片按预定义顺序处理子交易，并将执行结果传递至下一个分片，这不仅延长了交

易确认时间，还增加了通信开销。此外，智能合约交易中常见的嵌套调用和循环调用问题，可能导致状态解锁困难以及回滚频率增加，进一步降低系统的性能和可靠性[26]。因此，如何优化智能合约在分片架构下的执行效率，已成为区块链分片研究的关键方向之一。

1.5 BS-SCC 分类

基于智能合约执行方式和状态存储方式的差异，提出一个 BS-SCC 分类框架，如图 4 所示。为深入分析不同执行模式的影响，现有的研究工作可分为链上执行模型和链下执行模型两大类：

1)链上执行模型。智能合约的计算与验证完全由链上节点负责，链下执行器不参与任何计算。这种方式确保了合约执行的透明性和可验证性，但存在隐私泄露的风险。此外，由于所有计算任务集中在链上节点，在交易量较大或合约逻辑复杂的情况下，可能导致节点计算负载过重，增加用户成本并降低系统性能。

2)链下执行模型。智能合约的计算任务由链下执行器处理，链上节点仅负责验证关键状态信息和最终执行结果。通过将大量计算工作从链上转移到链下，该方案显著降低了链上节点的计算负担，进而减少交易成本并提高交易执行效率。然而，该模式引入了链上与链下之间的复杂交互，可能引发潜在的信任风险。

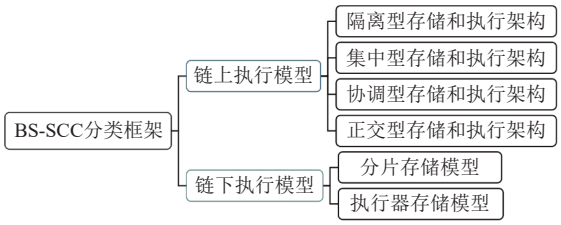


Fig. 4 The taxonomy of BS-SCC

图 4 BS-SCC 的分类

根据合约状态信息存储、合约逻辑代码部署和交易执行方式的差异，链上执行模型可细分为隔离型存储和执行架构、集中型存储和执行架构、协调型存储和执行架构，以及正交型存储和执行架构。

1)隔离型存储和执行架构。每个分片独立存储不同智能合约的状态与逻辑，并独立执行相应的智能合约。这种架构显著增强存储和计算资源分配的灵活性与可扩展性，支持高效工作负载分配。然而，处理涉及多分片智能合约的跨分片调用时，需进行多轮分片间共识与状态同步，导致系统吞吐量降低及交易确认延迟增加。

2)集中型存储和执行架构。每个分片独立存储不同智能合约的状态与逻辑，交易的执行则依赖于特定的执行分片。如果发生跨分片调用，执行分片可在过程前访问相关分片的合约状态与逻辑，执行完成后返回状态更新结果。尽管该架构可在一定程度上降低跨分片通信开销，但其交易处理能力高度绑定特定执行分片，难以随分片数量增加而线性扩展，限制系统的可扩展性。

3)协调型存储和执行架构。该架构允许部分节点存储多个分片的状态信息和逻辑代码，形成协调方。交易能在掌握全局状态的协调方节点直接完成验证与执行。根据具体需求，此类架构可实现降低跨分片交易频率、减少消息交换和有效处理冲突交易等不同功能。然而，协调节点需维护多个分片的状态数据，增加了节点的存储负担。

4)正交型存储和执行架构。正交架构的核心思想是灵活编排合约的状态存储和执行逻辑[27]，打破分片间的存储隔离。在此架构中，每个节点同时隶属于 2 种类型的分片：存储所有合约逻辑代码的执行分片，以及存储不同合约状态信息的状态分片。因此，执行分片无需进行跨分片共识，即可通过重叠子集同步状态分片的相关状态，在处理复杂合约时表现出较高效率。然而，其实现难度较高，需确保状态分片与执行分片间的高效交互，增加了状态管理的难度。

根据合约状态存储方式的差异，链下执行模型可进一步细分为分片存储模型和执行器存储模型。

1)分片存储模型。每个链上分片独立存储不同智能合约的状态信息，而智能合约的执行则由链下执行器负责。该模型通过将合约的执行与共识过程解耦，有效提升了合约执行效率。但链上与链下之间的频繁交互可能引入延迟，从而限制状态更新的响应速度，影响合约执行的实时性。

2)执行器存储模型。链下执行器不仅负责智能合约的执行，还维护合约的部分状态，而链上节点仅存储最终执行结果。通过解耦复杂逻辑合约的执行与共识，该模型提高了执行效率。然而，状态的链下维护可能无法保证更新操作的原子性与隔离性，链下执行器的错误或故障可能导致状态不一致，进而影响系统的可靠性和稳定性。

2 评估标准

本节将从可靠性、有效性及其他方面探讨用于

评估现有 BS-SCC 方案的关键指标，具体评估标准如图 5 所示。该评估框架主要借鉴并融合了以下 3 个方面的研究成果：

1）分布式系统的事务属性，包括原子性、活性、隔离性。这些属性借鉴了分布式数据库中的 AICD 基本要素，并根据 BS-SCC 特征进行了适配，从而满足基于智能合约的业务逻辑的执行要求。

2）区块链系统的通用评估准则，包括容错、吞吐量、确认时延、通信开销、存储开销等核心指标，构成了衡量区块链系统可靠性和有效性的重要基础。

3）面向区块链可扩展性与安全性的综合优化需求，包括负载均衡、隐私保护、链下计算与链下存储。该部分借鉴了链下执行以及隐私增强方案的众多研究成果，解决区块链分片在大规模应用中存在的性能与安全瓶颈。

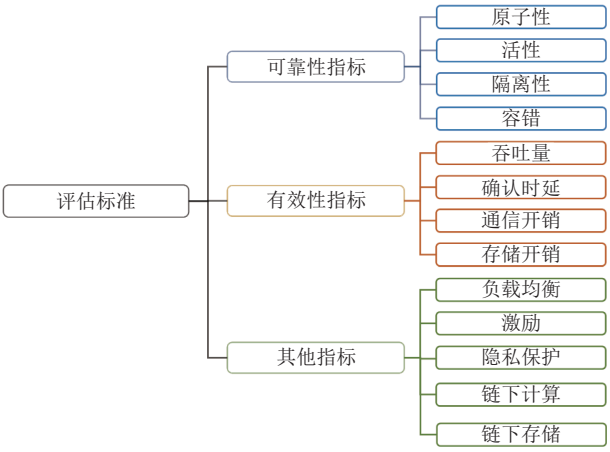


Fig. 5 Evaluation criteria of BS-SCC

图 5 BS-SCC 的评估标准

2.1 可靠性指标

智能合约分片方案必须确保交易执行的一致性，尤其在面对交易冲突、网络延迟以及潜在的攻击等情境。为了实现交易可靠且高效的执行，智能合约分片方案需具备原子性、活性、隔离性和容错等核心特性。

2.1.1 原子性

原子性要求涉及智能合约调用的交易要么全部成功执行，要么全部失败，不允许出现部分操作成功或者部分操作失败的中间状态[28]。这一特性借助原子提交机制得以实现，确保在涉及多个合约的复杂交易中，所有相关合约均能完成执行，从而维持逻辑上的一致性。即便合约状态分布于不同分片，原子性依然能有效防止因部分操作失败引发的状态不一致问题，进而保障系统的整体一致性和可靠性。

2.1.2 活 性

活性要求智能合约一旦被调用，必须在有限时间内完成执行，确保交易不会因网络延迟、共识争议或节点的恶意行为而陷入无限等待或停滞状态[28]。活性机制通过保证交易执行的持续进展，避免因系统故障或网络阻塞引发的“死锁”现象。即使在高负载或遭受恶意攻击的情况下，活性机制依然能够确保交易得以及时处理并更新状态，从而提升系统的稳定性与用户体验。

2.1.3 隔离性

隔离性是指即使系统同时执行多个交易，每个交易都应如同在独立环境中执行一样，不受其他并发交易的干扰[29]。该特性在高并发场景下尤为关键，能够有效防止交易冲突导致的交易视图不一致问题。即使多笔交易更新同一账户或合约状态，分片系统的隔离性也能保证每个交易完整运行，防止不同分片间状态的相互影响，进而维护系统的稳定性。

2.1.4 容 错

容错指的是每个分片在拜占庭节点存在的情况下稳定运行的能力。在区块链环境下，容错通常通过容错率 f/n 来量化，其中 f 是分片中拜占庭节点的最大数量，n 是分片内节点的总数。拜占庭节点可能通过广播错误信息或拒绝服务干扰交易共识[30]。因此，系统必须具备足够的容错能力，以确保在拜占庭情况下仍能维持交易执行的可靠性和网络的可用性。

2.2 有效性指标

有效性是衡量智能合约分片方案处理交易能力的核心指标，涵盖计算、通信、存储 3 个方面。分片方案的主要目标包括提高系统的吞吐量，降低交易确认时延、通信开销和存储开销。为了准确评估兼容智能合约分片方案是否达到了预期的性能，多维度的有效性测量至关重要。

2.2.1 吞吐量

吞吐量是指区块链系统在单位时间内可处理的交易数量，是衡量系统处理能力的关键指标。在分片区块链系统中，吞吐量尤为关键。通过合理分配计算和网络资源，分片方案能够在相同时间内处理更多的交易，从而实现更高的交易吞吐量，有效缓解单一区块链的性能瓶颈。

2.2.2 确认时延

确认时延是指交易从被添加至交易池到最终记录于参与分片的账本中所经历的时间[9]。较低的确认时延意味着交易能够更快地被确认和执行，这对于需要快速响应的应用场景尤为关键。通过缩短确

认时延, 系统可大幅提升跨分片交易的实时性和响应能力, 从而优化用户体验。

2.2.3 通信开销

通信开销是指交易处理过程中传输的数据量, 通常由通信复杂度来量化, 且受分片的大小(n)和跨分片交易中涉及的分片总数(N)的影响。通信开销直接影响网络的性能, 过高的通信开销可能导致网络拥塞和延迟, 进而降低系统整体性能。因此, 优化通信开销对于提升分片方案的运行效率和减轻系统负担具有重要意义。在本文中, 我们默认 BFT 共识机制导致的分片内通信开销为$O\left(n^2\right)$, PoW 共识机制导致的分片内通信开销为$O(n)$。

2.2.4 存储开销

存储开销是指共识节点在维护区块链网络时, 为存储交易副本所消耗的资源。这种开销占用大量存储资源, 限制了系统的可扩展性。存储开销通常通过空间复杂度x/N来量化, 其中x是区块链全账本的大小, N是分片数目。通过优化存储结构和减少冗余数据, 可以有效降低存储开销, 从而增强系统的可扩展性。

2.3 其他指标

其他指标涵盖了智能合约分片为进一步提升系统效率与安全性所采用的技术特性。这些特性旨在优化分片架构的性能, 同时增强其在复杂应用场景下的可靠性。

2.3.1 负载均衡

负载均衡是指将交易负载在多个分片之间进行合理分配, 避免部分分片因过载而成为“热分片”, 同时防止其他分片因负载不足而处于闲置状态。通过实现负载均衡, 区块链分片方案能够有效缓解交易拥塞导致的性能下降, 显著提升单个分片的处理能力, 优化整体系统的资源利用率。

2.3.2 激励机制

激励机制通过奖励或惩罚机制引导系统参与者的行为实现设计者预设的目标[31]。完善的激励机制能够对诚实参与者给予适当奖励, 同时对恶意节点施加有效惩罚, 降低潜在攻击风险。激励机制对于强化智能合约分片方案的鲁棒性与安全性以及确保系统稳定运行至关重要。

2.3.3 隐私保护

隐私保护是指分片方案为用户数据提供的隐私保障策略。鉴于区块链的透明性特性, 智能合约的输入与输出数据面临着敏感信息泄露的潜在风险[32]。分片方案必须运用高效加密技术或隐私保护机制, 以此抵御恶意节点对用户数据及身份隐私的威胁, 确保用户合法权益不受侵害。因此, 隐私保护对于防范用户数据泄露至关重要。

2.3.4 链下计算

链下计算将智能合约的执行过程从链上转移至链下, 仅将关键信息和最终结果提交至链上验证。这种方式显著减少了链上资源的占用, 缩短了因全网共识过程产生的等待时间, 有效降低交易成本并提升合约执行效率。此外, 链下计算避免了交易细节在区块链上的公开披露, 进一步增强了隐私保护能力。

2.3.5 链下存储

链下存储是指将大部分交易数据存储在区块链之外, 仅在链上记录数据索引或哈希值。这种方式大幅减轻了区块链网络的存储负担, 同时降低了交易成本。链下存储还提升了数据隐私性, 允许用户将敏感信息存储于 TEE 中, 而非完全暴露于公共区块链。通过合理设计链下存储机制, 可在确保数据可用性的同时优化存储资源分配与利用, 从而提升系统的可扩展性和隐私保护能力。

3 BS-SCC 具体方案评估

本节对现有 BS-SCC 方案进行综述, 并根据第 2 节提出的评估标准对具体方案进行讨论和评估, 如表 3 所示。

3.1 链上执行模型

3.1.1 隔离型存储和执行架构

Bassam 等人[24]设计了一个基于分片架构的可扩展智能合约平台 Chainspace。该平台基于对象、合约和交易等核心概念构建数据模型, 能够处理复杂的智能合约逻辑。在 Chainspace 中, 对象是存储状态的原子单元; 合约定义了操作对象的可执行逻辑; 交易则依据合约中的逻辑实现状态更新。每个分片管理部分对象和合约, 并采用 33% 的 BFT 共识协议达成一致交易。为解决跨分片交易的一致性问题, Chainspace 引入 S-BAC 协议, 该协议结合 BFT 和 2 阶段提交(two phase commit, 2PC)[44]机制。执行过程中, 各分片相互同步状态达成共识, 从而保证原子性和活性。在各分片达成共识前, 相关对象会被锁定, 以防止并发访问, 进而确保交易的隔离性。此外, Chainspace 的智能合约执行过程集成了零知识证明技术。节点仅需运行纯函数型检查器来验证分片证明, 而无需暴露对象状态, 从而在一定程度上实现了隐私保护。用户还可以通过向 CSCoin 系统合约支付来激励节点

Table 3 Summary and Comparison of BS-SCC Schemes

表 3 BS-SCC 方案的总结与对比

类别	文献	可靠性指标				有效性指标			其他指标			
		原子性	活性	隔离性	容错 %	通信开销	存储开销	负载均衡	激励	隐私保护	链下计算	链下存储
隔离型架构	[24]	√	√	√	33	$O(n^2+N^2)$	$\approx x/N$	×	√	√	×	×
	[25]	√	√	√	50	$O(n+N^2)$	$\approx x/N$	×	√	×	×	×
	[33]	√	√	√	33	-	-	×	√	√	×	×
	[34]	√	√	√	33	-	$\approx x/N$	×	×	×	×	×
	[35]	√	√	√	33	$O(n^2+N^2)$	$\approx x/N$	×	√	×	×	×
集中型架构	[36]	-	-	-	50	$O(n+1)$	$\geqslant x/N$	×	√	×	×	×
	[37]	√	√	×	33	$O(n^2+N)$	$\approx x/N$	×	×	×	×	×
	[38]	√	√	√	33	$O(n^2+N)$	$\geqslant x/N$	×	√	×	×	×
协调型架构	[39]	√	√	√	33	$O(n^2+N^2)$	$\geqslant x/N$	×	×	×	×	×
	[40]	√	√	√	33	$O(1+N)$	$\geqslant x/N$	√	√	×	×	×
	[41]	√	√	√	33	$O(1+N^2)$	$\geqslant x/N$	×	√	×	×	×
正交型架构	[27]	√	√	×	33	$O(n^2+1)$	$\approx x/N$	×	√	×	×	×
分片存储	[26]	√	√	√	33	$O(e^2+E+ne)$	$\approx x/N$	×	×	√	√	×
	[42]	√	√	√	33	$O(ne+n^2+N)$	$\approx x/N$	√	√	√	√	×
执行器存储	[43]	√	√	√	33	$O(ne+n^2)$	$\ll x/N$	√	×	√	√	√

注：“√”表示满足；“×”表示不满足；“-”表示无法评估；“n”表示分片大小；“N”表示分片数目；“e”表示链下执行集大小（或链下节点数目）；“E”表示链下执行集数目；“x”表示账本大小。评价结果基于原始文献提供的实验数据或本文作者对方案设计逻辑的综合判断。由于实验环境难以统一，吞吐量与确认时延等计算效率的总结工作于正文中详细说明。

参与交易处理。然而，该方案通过确定性哈希映射方式将智能合约分配到不同的分片，这可能导致负载失衡。其次，Chainspace 仅支持链上智能合约处理，未涉及链下计算与存储。由于交易执行需锁定被使用对象以及分片间的相互确认依赖，Chainspace 的吞吐量和确认时延仍有进一步优化的空间。15 个分片的每个分片包含 4 个节点的配置下，Chainspace 实现了 350 TPS 的吞吐量。跨分片交易需分片交换可用性证书，执行跨分片交易的通信开销近似为 $O(n^2+N^2)$。每个分片的存储开销近似为 x/N。

Wang 等人[25]提出了一种支持交易异步并发执行的方案 Monoxide。该方案通过对交易地址进行哈希映射操作，将用户和负载分配到不同的异步共识区。每个共识区内采用容错率为 50% 的 PoW 共识机制验证交易。为确保跨区域交易的正确性，Monoxide 解耦交易执行流程，将交易拆解为多个仅涉及单个区域的子交易，并通过最终原子性激励矿工正确处理中继交易，从而确保活性和激励。此外，Wang 等人[25]将用户/余额模型扩展为可编程的通用模型，允许用户余额表示为包括各种数据结构的任意状态，取款和存款逻辑可拓展为具备复杂逻辑的形式。智能合约被触发时，系统执行所有操作，以异步流程更新用户余额或合约状态。在跨区域状态修改时，Monoxide 在交易提交前检测并中止冲突的中继交易，保证原子性和隔离性。然而，智能合约的处理仅限于链上操作，未涉及链下计算和存储的需求，且未对隐私保护予以考虑。并且，Monoxide 中的哈希映射分配机制可能导致负载失衡。总体而言，Monoxide 通过异步无锁的交易执行机制与最终原子性设计，规避了传统 2PC 协议的锁开销，一定程度上优化了系统吞吐量。然而，最终原子性机制缺乏对交易的严格时间限制，这也在一定程度上导致了确认时延的增加。Wang 等人[25]的测试结果显示，在 2 048 个分片的平台上，Monoxide 吞吐量达 11 694 TPS，确认时延为 13~21s。由于合约调用需要分片间同步状态，执行跨分片交易时的总通信开销近似为 $O(n+N^2)$。每个分片的存储开销近似为 x/N。

ZILLIQA 2.0[33]引入了一种主权化分片架构，将区块链全局状态细分为相互独立可定制的分片单元。各分片的区块大小、区块时间及共识机制可依据应用需求自定义。智能合约可部署于特定的分片，精准契合该分片交易属性。分片采用容错率为 33% 的 Pipelined HotStuff [45]协议达成共识。在跨分片合约调用场景中，源分片合约生成事件日志，目标分片的验证器监听日志并借助验证器中继机制来同步其他分片的状态信息，待共识确认后执行跨分片交易。

若交易执行失败则拒绝跨分片合约调用，以确保交易原子性。此外，ZILLIQA[33]为未完成执行的合约提供了预编译合约设定触发条件的功能，由后续区块提议者自动发起重试，确保交易活性。ZILLIQA[33]还运用乐观并发控制机制与冲突检测技术维护合约调用隔离性。在激励机制上，ZILLIQA[33]支持代币质押模型，依据区块空间利用率动态调整验证者奖励，同时对恶意节点予以代币扣除。隐私保护方面，私有分片可采用对称加密技术，仅分片成员持密钥，保障跨分片数据传输全程加密，其他分片验证交易合法性时无法获取交易具体内容，进而实现隐私保护。然而，ZILLIQA[33]的设计仅局限于链上，尚未涉及链下执行和存储，且难以达成负载均衡。为进一步提升系统吞吐量，采用 Scilla 语言和 Block-STM 多线程执行机制，Scilla 语言支持数据流编程与并行计算，Block-STM 则用于处理非冲突交易。在由 1 800 个节点构成的 AWS 虚拟测试网络环境下，网络吞吐量可达 1 218 TPS，确认延迟约 1.2s。然而，尽管跨分片合约调用需被调用方持续监控其他分片，但状态同步导致跨分片通信复杂度近似为 $O\left(N^2\right)$。此外，分片共识参数与区块大小存在差异，分片内的通信与存储开销难以评估。

Pirlea 等人[34]提出了一种基于状态所有权和操作可交换性的智能合约分片方案。状态所有权指智能合约的状态由不同的分片独立管理和更新，而操作可交换性则指特定合约操作的执行顺序不影响最终结果。基于上述理念，Pirlea 等人[34]开发了一种静态分析工具 CoSplit，通过分析智能合约源代码推断所有权关系与操作交换性特征，并生成分片签名以指导交易分配策略。该方案支持对涉及可交换状态的交易进行并行处理，通过状态合并策略保障全局一致性，满足原子性和活性要求。同时，CoSplit 能够识别并处理冲突交易，进一步确保隔离性。然而，该方案存在以下局限性：未探索激励和负载均衡设计，且交易执行仅聚焦于链上，未集成链下计算与链下存储，也缺乏足够的隐私保护。此外，该研究未深入探讨跨分片交易处理机制，难以量化分片数量对通信开销的具体影响。每个分片的存储开销近似为 x/N。CoSplit 通过静态分析识别可交换操作以支持并行执行，并仅在非交换操作时锁定状态，从而在一定程度上提高了交易并发性和系统吞吐量。最后，Pirlea 等人[34]将 CoSplit 与 Zilliqa 区块链[33]集成，固定分片规模为 5 个节点，并在分片内采用容错率为 33% 的实用拜占庭容错（practical Byzantine fault tolerance，PBFT）[17]协议。在 5 个分片的配置下，智能合约执行吞吐量可超过 400 TPS。

针对传统区块链分片在智能合约场景下暴露的流量集中化与跨分片动态性处理缺陷，吴恺东等人[35]提出了一种面向智能合约的联盟链分片系统 BETASCO。该系统通过构建“定位—定序—执行”三阶段执行模型，有效解决了智能合约调用交易的分片调度难题。BETASCO 将智能合约的代码与状态分别部署于独立分片，并基于分布式哈希表（distributed Hash table，DHT）[46]的空间邻近性原理，建立智能合约与节点的确定性映射关系。合约调用请求通过 DHT 路由机制精准定位至目标分片后，各分片采用容错 33% 拜占庭节点的 Kafka 共识算法[47]，实现交易序列的全局一致排序，形成可验证的执行序列，从而保证隔离性。当智能合约执行过程中触发跨分片调用时，系统采用 2 阶段补偿机制：调用合约在交易成功执行后生成异步调用交易，通过 DHT 路由机制转发至目标合约分片；若被调用合约执行失败，则触发调用合约的异常处理方法，确保跨分片调用的原子性与活性。然而，该系统仍存在多方面局限：首先，基于哈希邻近性分配智能合约和分片成员，导致分片节点负载失衡；其次，缺乏有效的激励机制，难以确保节点在跨分片交易处理中的持续参与度；最后，智能合约执行完全依赖链上资源，未集成链下计算与链下存储，且缺乏对敏感数据的隐私保护。由于合约调用需要分片间同步状态，通信复杂度为 $O(n^2+N^2)$。每个分片的存储开销近似为 x/N。值得注意的是，BETASCO 允许单个物理节点加入多个分片，通过合约进程隔离实现多个交易的并行处理，在一定程度上提高了系统的吞吐量上限。在由 30 个云节点组成的实验集群中，将节点划分为 6 个规模为 5 的分片，并在部署 5 个主业务合约及 1 个日志合约的场景中，系统整体吞吐量达到了 1 000 TPS。

3.1.2 集中型存储和执行架构

Tao 等人[36]指出仅参与单个智能合约的用户发送的交易可以被独立验证和确认，无需跨分片通信。基于此，他们提出将此类交易单独分配至一个分片，其他矿工可在不影响系统一致性的前提下，并行验证和确认剩余交易。这一设计实现了智能合约验证过程中的零跨分片通信。此外，Tao 等人[36]还提出了分片间合并算法以激励小分片合并，以及分片内最优交易集选择算法以优化资源分配。然而，由于分片的计算任务量取决于分片内矿工数量，该方案虽能满足激励要求，但未能实现有效的负载均衡。该方案主要聚焦于对用户调用的合约状态和执行过程

的部署优化，但未考虑链式调用和链下交易等场景，导致在链下存储与链下计算方面存在一定局限性，且缺乏隐私保护机制。由于该方案未明确提出具体的智能合约交易执行策略，其原子性、隔离性、活性等无法得以评估。最后，Tao 等人[36] 基于 Ethereum 平台实现了分片方案，分片内共识采用容错率为 50% 的 PoW 机制。实验结果显示，当分片数目为 9 时，与 Ethereum 系统相比，该方案吞吐量提高了 720%。同时，分片间合并和分片内交易选择方案分别为分片系统带来额外 11% 和 300% 的吞吐量增益。由于该方案避免了跨分片验证，消息传输主要来源于分片内共识，总通信开销近似为 $O(n+1)$。每个常规分片的存储开销近似为 x/N，而只涉及单个智能合约的用户交易的分片存储开销可能高于 x/N。

Zheng 等人[37] 提出了一种面向联盟链系统的分片方案 Meepo，旨在通过跨时代跨调用协议优化跨分片通信效率。为提升智能合约在分片环境下的灵活性，Meepo 引入了部分跨分片调用合并策略。该策略允许源分片在发送跨分片中继调用时附带部分参数，目标分片接收后对多个跨分片调用进行合并，重构完整的状态和执行逻辑，从而实现交易的高效执行。为确保交易原子性，Meepo 采用时代重放机制，当交易执行异常时，所有相关分片协同回滚异常交易，并通过区块更新机制传播修正后的状态，最终仅提交成功执行的交易。此外，Meepo 为每个主分片配置一个影子分片作为冗余备份。影子分片和主分片的分片内共识协议都采用容错率为 33% 的 PBFT 协议。影子分片虽不直接参与共识，但持续同步主分片的区块和状态信息，一旦主分片发生故障，影子分片将立即接管，从而确保系统的活性。然而，Meepo 方案存在以下局限性。首先，该方案缺乏严格的跨分片调用并发控制策略，使得跨分片并发交易可能发生视图冲突，因而不满足隔离性。其次，Meepo 未解决分片间的负载均衡。此外，Meepo 并未针对系统维护和交易执行提供相应的激励。最后，Meepo 未集成链下计算和存储，也无法增强分片的隐私保护。由于允许合约调用参数合并和交易批量执行，Meepo 能够显著提升系统吞吐量和降低通信开销。在由 128 个节点和 32 个分片组成的测试环境中，Meepo 的吞吐量可达 140 000 TPS，确认时延约为 1.2 s。Meepo 的通信复杂度近似为 $O(n^2+N)$。每个分片的存储开销近似为 x/N。

Liang 等人[38] 提出了一个基于跨分片缓存的区块链分片协议 Sparrow，该协议通过分片预取机制，将高频跨分片合约状态缓存至本地，从而将跨分片智能合约交易的多轮子交易压缩至单个区块时间内完成。具体而言，分片通过统计历史交易频率识别热点合约，并借助跨分片辅助节点构建的分布式缓存网络完成数据同步。分片内部运行容错率为 33% 的 BFT 协议，在共识达成阶段，通过 Merkle Root 和哈希验证保障缓存数据的完整性与时效性，确保只有通过校验的缓存更新才能提交至全局账本。此外，Sparrow 采用混合并发控制策略，结合乐观执行[48] 与悲观冲突管理解决并发执行中的交易冲突，因此满足隔离性。在乐观执行阶段，Sparrow 通过多分支探索机制处理子交易之间的依赖，并为账户访问创建多个逻辑版本，最终确定执行顺序。因此，Sparrow 满足原子性和活性。然而，由于交易涉及的所有账户被随机分配到不同的分片，可能导致负载失衡；而且 Sparrow 并未为跨分片辅助节点提供明确的激励机制，因此激励有限。Sparrow 尚未探索链下计算和存储且缺乏敏感数据的隐私保护。得益于本地预取和 IDA 编码机制，通信总复杂度为 $O(n^2+N)$。由于缓存机制，分片的存储开销高于 x/N。Liang 等人[38] 在包含 64 个分片的实验环境中进行了评估，每个分片的块更新时间控制在 1 s 左右，共识和网络传播的延迟控制在 100 ms。在此环境下，Sparrow 的吞吐量可达 654 TPS，交易确认延迟降低至 2 s。这表明 Sparrow 通过预执行压缩共识轮次、冲突管理、并行化子交易等策略能显著降低交易确认时延。

3.1.3 协调型存储和执行架构

Hong 等人[39] 提出了一种分层分片的区块链方案 Pyramid，其核心理念是允许部分节点存储多个分片的交易记录，形成重叠分片以处理和验证跨分片交易。为确保跨分片交易的一致性，Hong 等人[39] 设计了一种基于多分片协作的分层分片共识协议。该协议将节点分为内部分片和桥接分片，两者均采用容错率为 33% 的 BFT 协议作为分片内共识机制。桥接分片存储多个相关内部分片账本信息，在每一轮跨分片交易执行中，桥接分片生成一个跨分片块并发送至相关内部分片，只有在收到所有相关内部分片的确认后，桥接分片才会提交该区块。对于涉及多个分片的跨分片交易，Pyramid 借鉴 Monoxide 的中继交易机制，将交易分解为多个步骤处理，以适应桥接分片的重叠范围限制。这一设计保障了交易的原子性和活性。此外，当交易发生冲突时，系统通过随机接受一个块并拒绝其他块，确保了隔离性。然而，Pyramid 方案未深入探索激励和负载均衡的设计。

此外,其智能合约执行仅局限于链上,未集成链下计算和链下存储,也无法提供足够的隐私保护。由于引入 Monoxide 的异步调用和 BFT 类共识,通信复杂度近似为 $O(n^2+N^2)$。每个内部分片的存储开销近似为 x/N,而桥接分片负责处理跨分片交易,存储多个相关内部分片账本,存储开销高于 x/N,其中 x 是区块链全账本的大小。相较于传统完全分片方案中需将跨分片交易拆分为多个子交易并经过多轮共识提交的做法,分层分片共识能够在单轮共识内完成跨分片交易提交,显著提升效率。实验结果显示,在 3 500 个节点和 17 个分片的环境中,Pyramid 的最佳吞吐量能达到 3 800 TPS,交易确认延迟约为 3 s。

Elrond [40] 运用自适应状态分片技术实现了智能合约与分片架构的兼容性。在这种架构下,智能合约及其状态默认部署在同一分片内。各分片除维护自身状态外,还经由树状层级结构存储关联分片的历史状态。Elrond 鼓励分片拆分和节点重分配,当分片的交易量和节点数超过预设阈值时,系统将自动拆分分片,达成负载均衡。交易执行时,各分片借助伪随机算法,从分片验证者池里动态挑选固定数量的节点来构成临时共识组。该组运行改进的 PBFT 协议融合 2 轮 BLS[48] 多签流程以达成共识。对于跨分片合约调用场景,发送方分片先执行本地操作,再将跨分片交易打包为微块并提交至元链公证。接收方分片从元链获取该微块哈希值,验证无误后执行跨分片操作。若流程中环节出错,微块整体回滚,保障交易原子性[49]。元链负责记录各分片区块头与微块哈希,并依时间戳排序,确保交易序列化执行与隔离性。此外,分片运用冗余存储机制,分片故障时可快速重组,维护交易活性。同时,Elrond[40] 凭借节点评级机制激励诚实的区块提议者,降低恶意节点成为提议者的概率。然而,Elrond[40] 尚未探索链下计算和存储,也缺乏对敏感数据的隐私保护策略。因分片内共识融入 BLS 签名且经元链中继执行证明,分片内通信开销降低至常数级,通信总复杂度为 $O(1+N)$。鉴于存储冗余机制,每个分片的存储开销高于 x/N。在 16 分片测试网下,凭借动态负载均衡设计、轻量化共识协议及微块异步批量处理机制,Elrond[40] 实现了 12 500 TPS 和低于 10 s 的跨分片交易确认时延。

Hong 等人[41] 提出了一种确定性排序的分片架构 Prophet,通过最小化非确定性争用合约状态导致的交易中止来优化区块链系统的性能。该架构由侦察联盟、序列分片和执行分片 3 种类型的分片组成,每种分片均采用容错率为 33% 的拜占庭容错共识机制作为其分片内共识协议。侦察联盟作为 Prophet 架构的核心,由来自不同分片的节点组成,能够提供全局合约的视图并监控跨分片合约调用的整体状态。基于此,Prophet 实现了一个包含预执行、排序、执行和校正 4 个阶段的跨分片交易协作共识机制。在预执行阶段,侦察联盟内的节点协作预执行跨分片交易并收集交易的读写集。接着,序列分片根据预执行结果确定全局交易顺序,并生成每个分片的区块顺序。执行分片根据收到的区块顺序运行分片内共识机制,并按照既定顺序执行交易。最后,每个分片共享并接受执行证明,只有在收到所有相关分片的证明后,才能更新其存储的合约状态。通过侦察联盟和序列分片的协调,执行分片在调用合约时存在严格的顺序,因此 Prophet 能够满足隔离性要求。同时,分片间证明共享机制确保了交易的原子性和活性。此外,Prophet 会向成功提交预执行交易的侦察联盟支付一部分交易费用,激励节点积极参与预执行过程。然而,Prophet 架构也存在一些局限性:首先,它未提及负载均衡和隐私保护策略。其次,其交易执行和存储仍完全依赖于区块链,尚未涉及链下计算和存储。由于每个分片内共识引入了 BLS 签名且分片间共享接受证明,分片内通信开销降低至常数级,通信总复杂度为 $O(1+N^2)$。每个执行分片的存储开销近似为 x/N,而侦察联盟的存储开销高于 x/N。总体而言,Prophet 通过预执行协作优化与分片内高效共识机制显著提升了系统吞吐量。然而,其分片间证明共享机制也可能导致确认时延增加。在 64 个分片以及每个分片 50 个节点的实验设置中,恶意节点的比例固定为 12.5%。Prophet 达到了 1 203 TPS 的吞吐量,确认时延约为 2.5 s。

3.1.4 正交型存储和执行架构

Li 等人[27] 通过对智能合约的存储和执行逻辑进行编排,提出了名为 Jenga 的区块链分片方案。Jenga 要求每个分片存储完整的智能合约执行逻辑,使得任意分片在获取交易所需合约状态后即可独立完成智能合约的执行,无需多轮跨分片调用。此外,Li 等人[27] 在存储隔离状态的分片上“正交”地设置了多个执行通道,即每个节点同时属于一个状态分片和“正交”的执行通道。状态分片内部运行容错率为 33% 的 BFT 协议,且执行通道可以重叠所有状态分片,从而消除状态分片与执行通道之间的额外跨分片通信开销。基于上述架构,Li 等人[27] 提出了一种 3 阶段跨分片共识协议。在执行跨分片共识时,状态分片

确定交易所需的状态，并通过重叠子组广播到执行通道。执行通道接收状态后，启动状态计数器，并在所有状态信息到达后并行执行智能合约逻辑。执行通道完成任务后，通过重叠子组向状态分片返回执行结果，状态分片随后更新合约状态。因此，该方案满足原子性。然而，由于未提供严格的跨分片并发控制策略，该方案可能无法满足隔离性要求。系统要求客户端质押费用以参与跨分片共识，这一设计可视为一种激励机制的变体，能够有效防止交易阻塞，从而满足激励和活性要求。然而，Jenga 基于交易地址后缀决定执行通道，这导致其无法实现负载均衡。此外，该方案未集成链下计算和链下存储，在隐私保护方面存在一定局限性。由于该方案不涉及跨分片验证，其通信复杂度为 $O(n^2+1)$，每个分片的存储开销约为 x/N。总体而言，Jenga 通过合约逻辑全局共享、状态-执行正交分离以及重叠子组交互等设计，有效克服了传统分片方案中因合约隔离而导致的性能瓶颈问题，显著提高了吞吐量。Li 等人[27]基于 Amazon EC2 进行测试，在 12 个分片且每个分片包含 240 个节点的配置下，Jenga 实现了 4 400 TPS 的吞吐量，交易确认时延约为 11 s。

3.2 链下执行模型

3.2.1 分片存储模型

Zhang 等人[26]提出了一种基于链下执行模型的区块链分片系统优化方案 ShardCon，旨在解决传统基于锁的跨分片协议在处理复杂跨分片合约时效率低及死锁问题。ShardCon 的核心思想在于将跨分片合约交易的执行与共识过程解耦，将具体的合约执行任务转移至链下。具体而言，合约创建者可指定特定的链下执行器负责合约计算，而链上分片仅需验证合约状态更新的有效性，从而减轻链上计算负担并提升执行效率。在该方案中，每个链下执行节点独立完成完整合约的执行，并通过密码学证明和远程验证技术生成可追溯的执行结果，一定程度上增强了隐私保护。ShardCon 的链上部分建立在运行 BFT-SMR 共识的区块链分片协议之上。该协议具备 33% 的容错能力，并依赖部分同步网络的活性条件，即确保诚实节点发送的消息能在限定时间内传递至其他诚实节点，从而确保交易的活性。为确保跨分片合约执行结果的一致性，Zhang 等人[26]设计了一种链下状态原子提交协议。该协议通过确定性的执行顺序协调所有跨分片合约交易，并在提交状态更新前要求相关分片完成验证，从而保障交易的原子性和隔离性。然而，ShardCon 忽略了负载均衡和激励的要求。同时，尽管该方案强调链下执行，但其设计主要聚焦于链下计算，未深入解决链下存储的相关问题。在执行跨分片交易时，通信开销主要来自分片间的共识、执行器组内的共识、组间的集中式协调以及链上与链下之间的状态同步，因此通信复杂度近似为 $O(n^2+E+ne)$，每个分片的存储开销近似为 x/N。ShardCon 根据合约之间的依赖关系动态配置合约执行器，能够有效减少跨分片合约调用的频次，从而优化交易确认延迟。同时，ShardCon 融合了链下并行执行、TEE 硬件加速和链下状态原子提交协议等多种技术手段，保证系统的高吞吐量。在由 5 个分片，每个分片包含 4 个分片验证器和 8 个链下执行器的测试环境中，与 Chainspace 相比，ShardCon 吞吐量提升了 10 倍，延迟降低 50%。针对状态大小为 8 KB 的复杂跨分片合约，ShardCon 的吞吐量可以达到 1 000 TPS，确认延迟能够低至 160 ms。

Qi 等人[42]提出了一种基于链下 TEE 的区块链分片系统 LightCross，旨在提升复杂智能合约的执行效率。该系统包含客户端、执行器、共识分片和调度分片 4 类角色，其中共识分片和调度分片均采用容错率为 33% 的 BFT 共识机制。LightCross 的核心思想在于解耦智能合约的执行过程与共识机制，即将共识分片的执行任务委托给一组由 TEE 授权的链下执行器以降低链上计算负担。为提交复杂的跨分片智能合约交易，LightCross 采用轻量级跨分片提交协议。具体而言，执行器通过点对点通信同步必要的状态数据至相关共识分片。在交易执行完成后，执行器生成证明以验证执行过程的正确性，并将交易执行结果及相应证明发送至调度分片。调度分片将多个跨分片交易的执行结果打包为批次，单次共识完成批量验证。在最终提交交易前，会验证其序列化可行性，以确保交易不会因读取过时状态数据而提交失败。该一致性设计保障了交易的原子性、活性和隔离性。此外，LightCross 还能够根据交易历史动态调整智能合约的分布，以实现负载均衡。然而，该系统未提供任何激励机制。尽管 LightCross 采用链下执行模式，一定程度上提供了隐私保护，但其设计仅涉及链下计算，未考虑链下存储。在交易处理过程中，通信复杂度近似为 $O(ne+n^2+N)$，每个分片的存储开销约为 x/N。通过将跨分片交易卸载至链下执行，并集成动态智能合约迁移机制以降低跨分片交易频次，LightCross 能够有效保证系统的高吞吐量和低确认时延。Qi 等人[42]基于 FISCO-BCOS 平台实现了 LightCross 的原型。在分片数量为 16 且每

个分片包含 22 个节点的测试环境中，系统吞吐量可达到 15 800 TPS 且确认延迟不超过 4 s。

3.2.2 执行器存储模型

Xu 等人[43]提出了支持智能合约交易高效执行的无状态区块链系统 SlimChain，该系统将交易执行和账本状态存储转移至链下存储节点，仅在链上维护状态承诺的根哈希，从而实现共识节点的轻量化。系统执行流程如下：客户端向存储节点发送交易请求，存储节点通过 TEE 验证交易签名后，基于本地维护的状态快照执行智能合约。TEE 确保执行结果的完整性，生成包含交易输入、读写集及状态根的数字签名作为执行结果。随后，存储节点将执行结果及无状态根广播至共识节点。共识节点在验证交易有效性后，将其纳入新区块，并更新链上的状态承诺。交易被记录至区块链账本后，存储节点通过异步方式将交易的写集提交至本地状态数据库，实现链下状态的最终一致性更新。SlimChain 支持乐观并发控制和可序列化快照隔离[50] 2 种并发控制技术，用于检查交易的读写集冲突，以保证隔离性。此外，SlimChain 结合 TEE 进行链下计算和存储，有效防止隐私数据泄露，增强隐私保护。为了进一步提升系统的可扩展性，SlimChain 提出基于地址前缀的动态分片机制。存储节点可自主选择存储完整状态或特定分片数据，避免单个节点负载过重，实现负载均衡。对于跨分片交易，客户端可选择将交易直接发送至存储全量数据的节点执行，或由多个存储节点通过容错率为 33% 的 Raft 共识协议[51] 协同处理，以确保交易执行的原子性和活性。然而，SlimChain 并未针对系统维护和交易执行提供相应的激励机制，因此不满足激励要求。在交易处理过程中，由于涉及链下存储节点的全流程执行以及链上与链下之间的同步，SlimChain 的通信复杂度近似为 $O(ne+n^2)$。由于状态数据交由链下节点存储，每个分片的存储开销远低于 x/N。SlimChain 通过解耦交易执行与共识机制，并利用链下 TEE 进行多线程合约执行，显著提升了分片系统的吞吐量。同时，借助轻量化的 Merkle 压缩技术加速状态验证过程，在一定程度上降低了延迟。在包含 16 个共识节点和 4 个存储节点的网络中，SlimChain 的吞吐量可达 960 TPS，确认时延约为 2.7 s。

4 开放性问题和未来研究方向

基于上述 BS-SCC 方案的综述，本节归纳总结出现有 BS-SCC 技术的未解问题，并以此为指导提出未来研究方向。

4.1 通信开销

1)开放性问题。现有研究面临分片间通信开销较高的难题。在处理复杂逻辑时，智能合约的状态和代码通常存储于不同分片，使得跨分片合约调用频繁发生，导致通信成本显著增加。尤其在合约状态分布较为分散时，跨片通信的开销进一步加剧，严重影响系统的整体性能和效率。

2)未来研究方向。未来研究应聚焦于优化分片间的通信效率。首先，分析智能合约之间的调用关系，将实现同一功能的合约部署在同一分片内以减少跨分片调用的次数[42]。其次，引入批处理机制，使单个分片签发的证书包含多个执行结果，从而提高数据传输效率[52]。此外，探索高效的编码技术以压缩分片间传输的数据量，例如采用数据压缩算法，最大限度减轻通信负担[53]。

4.2 负载均衡

1)开放性问题。当前研究普遍忽略分片间的负载均衡，导致分片过载和欠载。欠载分片指某些分片的负载不足，使得实际吞吐量低于理论峰值；而过载分片指另一些分片的工作负载超出其处理能力，导致交易拥塞和确认延迟较高。负载失衡问题严重影响了系统的整体性能和效率。

2)未来研究方向。为实现有效的负载均衡，交易分配策略需要从多个角度进行综合考量。首先，可以借鉴历史交易数据，在部署智能合约时，优先将其分配至当前负载较低的分片[54]。其次，策略应具备自适应能力，能够根据系统实际的交易负载情况灵活调整合约在不同分片中的分布。通过实时监测各分片负载状态，并将负载较高的合约迁移至较空闲的分片，可有效避免分片过载[55-56]。最后，交易分配策略应与前面提到的跨分片通信优化措施结合，在实现负载均衡的同时，兼顾跨分片通信的效率[57]。

4.3 隐私保护

1)开放性问题。当前智能合约分片在隐私保护方面存在显著不足，容易泄露用户敏感信息。由于区块链的透明性，链上所有数据全网可见。然而，在分布式公钥基础设施(decentralized public key infrastructure, DPKI)[58] 和车联网[59] 等场景中，多个实体交互时产生的数据往往不能完全公开。因此，如何在实现数据共享的同时有效保护用户隐私，成为一个亟待解决的关键问题。

2)未来研究方向。为应对隐私保护的挑战，未来研究可从以下 3 个方面展开研究：首先，智能合约

应嵌入细粒度的属性或角色访问规则，确保只有经过授权的实体才能访问特定数据[60]。其次，利用TEE将敏感数据的计算迁移至链下，以保护智能合约中的敏感数据隐私[61]。此外，在需要匿名的交易场景中，用户可以向证书颁发机构注册身份并申请证书，交易时利用零知识证明技术验证身份合法性，全程无需透露任何个人信息[62-63]。

4.4 链下存储

1）开放性问题。现有方案往往忽视了链下存储的可扩展性问题。在实际应用中，节点会产生大量认证数据，这些数据不仅占用大量存储空间，还需要支持高效的检索操作。若所有认证数据直接上链存储，容易引发存储瓶颈，显著影响系统的性能。

2）未来研究方向。首先，引入星际文件系统（interplanetary file system, IPFS）[64]等去中心化存储技术分散数据存储压力。这种技术能够有效减轻中心化存储的瓶颈问题，显著提升系统的存储能力。其次，对链下存储内容实施分片管理，将数据或合约分割为更小的单元进行分布式存储，提高数据检索效率的同时，减轻单个存储节点的负载。最后，考虑到链下存储需与链上分片频繁交互，应建立明确的映射机制，确保链下数据与链上状态能够高效同步。通过设计低延迟的一致性协议，可在维持数据完整性的同时，提升链上链下的协同效率。

4.5 链下执行

1）开放性问题。现有的链下执行仍存在显著的性能提升空间。智能合约的链下执行通常由TEE授权的链下执行器负责，而合约状态的维护可能由链上分片或链下执行器承担。在链上分片负责维护的合约状态的场景中，链下执行器需要频繁与链上分片交互以获取最新状态，导致较高的通信延迟。而在链下执行器负责维护合约状态时，需引入复杂的状态同步与协调机制来保障一致性，这限制了合约执行的并行程度，并可能成为性能瓶颈。

2）未来研究方向。未来研究应针对分片存储和执行器存储2种模式，分别设计高效的链下执行策略。首先，在链上分片负责合约状态的场景中，可通过分析合约间的关联性，将高度相关的合约状态部署于同一分片内[42]。这种集中部署策略能够减少跨分片状态同步的频率，从而降低链下执行器与链上分片间的交互延迟，提升整体执行效率。其次，在链下执行器负责合约状态的场景中，可引入流水线技术等并行化技术优化任务处理流程[65]。通过提升执行器内部及执行器之间的共识效率，这种方法不仅能够提高并行处理能力，还能减少状态同步延迟和冲突风险。

4.6 基准测试

1）开放性问题。相关研究缺乏统一的基准测试框架，导致不同BS-SCC方案间的性能对比缺乏依据，限制了研究人员对系统瓶颈的深入理解。目前的研究工作大多基于各自构建的定制化实验环境，由于负载配置及网络参数存在差异，实验不具备可比性和复现性。此外，现有测试环境忽略了合约嵌套调用、高并发冲突交易等真实工作负载特征，难以全面反映BS-SCC方案在实际应用中的表现。

2）未来研究方向。未来研究应高度重视BS-SCC方案的标准化测试与评估。首先，可以参考YCSB、DeFiBench等通用基准测试方法，设计覆盖智能合约部署和跨分片合约调用等关键环节的标准化测试流程[66]。其次，引入DeFi等现实区块链典型应用场景的交互模式作为工作负载基准。通过模拟真实应用场景，评估BS-SCC方案在复杂和高并发条件下的适用性。最后，构建支持多种BS-SCC方案集成与对比的自动化测试平台。该平台应具备灵活的配置能力，能够模拟不同网络环境和负载条件，便于研究人员快速部署和测试不同方案。

5 结　论

本文对BS-SCC技术进行了系统性综述。在介绍BS-SCC研究背景和基本原理的基础上，本文提出了一种新颖且全面的BS-SCC分类框架，为后续的综述工作提供了理论基础。进一步地，本文构建了一套BS-SCC的多维度评估框架，涵盖可靠性、有效性及其他指标。随后，本文对现有主流研究方案进行了全面综述，基于所构建的评估框架，深入分析各方案的优势与不足。最终，根据文献综述和评估结果，本文明确了当前研究中存在的技术瓶颈，并指出若干具有发展前景的研究方向，旨在为BS-SCC技术的后续研究提供新视角和理论指引。

作者贡献声明：张滋炯负责论文主要内容撰写及修改；韩昊翔和史晁铭补充文献资料并协助修改论文；闫峥负责提供研究方案和指导论文整体写作思路；王海光和李铁岩提出论文指导意见。

参　考　文　献

[1] Wu Canghai, Xiong Jie, Xiong Huanliang, et al. A review on recent

progress of smart contract in blockchain[J]. IEEE Access, 2022, 10: 50839−5086

[2] Buterin V. Ethereum: A next-generation smart contracts and decentralized application platform [EB/OL]. (2025-02-12)[2025-08-25]. https://ethereum. org/en/whitepaper/

[3] VISA. Benefits of accepting visa [EB/OL]. (2025-11-04)[2025-11-08].https://africa.visa.com/en_MW/content_library/benefits-accepting-visa.htm

[4] Luu L, Narayanan V, Zheng Chaodong, et al. A secure sharding protocol for open blockchains [C]//Proc of the 23rd ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2016: 17−30

[5] Wang Shuai, Yuan Yong, Wang Xiao, et al. An overview of smart contract: Architecture, applications, and future trends [C]//Proc of the 23rd IEEE Intelligent Vehicles Symp. Piscataway, NJ: IEEE, 2018: 108−113

[6] Wang Shuai, Ouyang Liwei, Yuan Yong, et al. Blockchain-enabled smart contracts: Architecture, applications, and future trends[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019, 49(11): 2266−2277

[7] Hewa T, Ylianttila M, Liyanage M. Survey on blockchain based smart contracts: Applications, opportunities and challenges[J]. Journal of Network and Computer Applications, 2021, 177: 102857

[8] Zheng Zibin, Xie Shaoan, Dai Hongning, et al. An overview on smart contracts: Challenges, advances and platforms[J]. Future Generation Computer Systems, 2020, 105: 475−491

[9] Liu Yizhong, Liu Jianwei, Salles M, et al. Building blocks of sharding blockchain systems: Concepts, approaches, and open problems[J]. Computer Science Review, 2022, 46: 100513

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

[11] Li Yi, Wang Jinsong, Zhang Hongwei, et al. A survey of state-of-the-art sharding blockchains: Models, components, and attack surfaces[J]. Journal of Network and Computer Applications, 2023, 217: 103686

[12] Yang Qinglin, Huang Huawei, Yin Zhaokang, et al. The state-of-the-art and promising future of blockchain sharding[J]. IEEE Communications Magazine, 2025, 63(4): 192−198

[13] Tang Haibo, Zhang Huan, Zhang Zhao, et al. Research progress on blockchain sharding technology[J]. Journal of Software, 2025, 36(3): 1040−1064 (in Chinese)
(唐海波, 张焕, 张召, 等. 区块链分片技术研究进展[J]. 软件学报, 2025, 36(3): 1040−1064)

[14] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system [EB/OL]. (2025-06-11)[2025-11-08]. https://bitcoin.org/bitcoin-paper

[15] Gervais A, Karame G, Wust K, et al. On the security and performance of proof of work blockchains [C]//Proc of the 23rd ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2016: 3−16

[16] Nguyen C T, Hoang D T, Nguyen D N, et al. Proof-of-stake consensus mechanisms for future blockchain networks: Fundamentals, applications and opportunities[J]. IEEE Access, 2019, 7: 85727−85745

[17] Castro M, Liskov B. Practical Byzantine fault tolerance and proactive recovery[J]. ACM Transactions on Computer Systems, 2002, 20(4): 398−461

[18] Szabo N. Formalizing and securing relationships on public networks [J/OL]. (1997-09-01)[2025-11-08]. https://myinstantid.com/szabo.pdf

[19] Elli A, Artem B, Bortnikov V, et al. Hyperledger Fabric: A distributed operating system for permissioned blockchains [C]//Proc of the 13th EuroSys Conf. New York: ACM, 2018: 15−30

[20] Frassetto T, Jauernig P, Koisser D, et al. POSE: Practical off-chain smart contract execution [C/OL] //Proc of the 30th Network and Distributed System Security Symp. San Diego, CA: Internet Society, 2023[2025-11-12].https://www.ndss-symposium.org/ndss-paper/pose-practical-off-chain-smart-contract-execution/

[21] Jauernig P, Sadeghi A, Stapf E, et al. Trusted execution environments: Properties, applications, and challenges[J]. IEEE Security & Privacy, 2020, 18(2): 56−60

[22] Victor C, Srinivas D. Intel SGX explained [J/OL]. IACR Cryptol ePrint Arch, 2016 [2025-11-08]. https://eprint.iacr.org/2016/086.pdf

[23] Zhang Zitong, Han Haoxiang, Yan Zheng, et al. A survey on cross-shard transaction allocation optimization and processing mechanisms [J]. IEEE Communications Surveys and Tutorials, 2025, 27(4): 1−27

[24] Bassam M, Sonnino A, Bano S, et al. Chainspace: A sharded smart contracts platform [C/OL]//Proc of the 24th Network and Distributed System Security Symp. San Diego, CA: Internet Society, 2018[2025-11-08]. https://sonnino.com/papers/chainspace.pdf

[25] Wang Jiaping, Wang Hao. Monoxide: Scale out blockchains with asynchronous consensus zones [C]//Proc of the 16th USENIX Conf on Networked Systems Design and Implementation. Berkeley, CA: USENIX Association, 2019: 95−112

[26] Zhang Jianting, Chen Wuhui, Hong Zicong, et al. Efficient execution of arbitrarily complex cross-shard contracts for blockchain sharding [J]. IEEE Transactions on Computers, 2024, 73(5): 1190−1205

[27] Li Mingzhe, Lin You, Zhang Jin, et al. Jenga: Orchestrating smart contracts in sharding-based blockchain for efficient processing [C]//Proc of the 42nd IEEE Int Conf on Distributed Computing Systems. Piscataway, NJ: IEEE, 2022: 133−143

[28] Ozsu M, Valduriez P. Principles of Distributed Database Systems [M]. Berlin: Springer, 1999

[29] Bernstein P A, Goodman N. Concurrency control in distributed database systems[J]. ACM Computing Surveys, 1981, 13(2): 185−221

[30] Gramoli V. From blockchain consensus back to Byzantine consensus [J]. Future Generation Computer Systems, 2020, 107: 760−769

[31] Han Rong, Yan Zheng, Liang Xueqin, et al. How can incentive mechanisms and blockchain benefit with each other? A survey[J]. ACM Computing Surveys, 2022, 55(7): 1−36

[32] Liang Wei, Liu Yaqin, Yang Ce, et al. On identity, transaction, and smart contract privacy on permissioned and permissionless blockchain: A comprehensive survey[J]. ACM Computing Surveys, 2024, 56(12): 1−38

[33] Zilliqa. ZILLIQA 2.0 whitepaper [EB/OL]. (2024-06-24)[2025-08-25]. https:// www.zilliqa.com

[34] Pirlea G, Kumar A, Sergey I. Practical smart contract sharding with ownership and commutativity analysis [C]//Proc of the 42nd ACM SIGPLAN Int Conf on Programming Language Design and Implementation. New York: ACM, 2021: 1327−1341

[35] Wu Kaidong, Ma Yun, Cai Huaqian, et al. BETASCO: Consortium blockchain system based on smart contract-oriented sharding[J]. Journal of Software, 2023, 34(11): 5042−5057 (in Chinese)
(吴恺东, 马郓, 蔡华谦, 等. BETASCO: 面向智能合约分片的联盟

区块链系统[J]. 软件学报, 2023, 34(11): 5042−5057)

[36] Tao Yuechen, Li Bo, Jiang Jingjie, et al. On sharding open blockchains with smart contracts [C]//Proc of the 36th IEEE Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2020: 1357−1368

[37] Zheng Peilin, Xu Quanqing, Zheng Zibin, et al. Meepo: Sharded consortium blockchain [C]//Proc of the 37th IEEE Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2021: 1847−1852

[38] Liang Junyuan, Yao Peiyuan, Chen Wuhui, et al. Sparrow: Expediting smart contract execution for blockchain sharding via inter-shard caching[J]. IEEE Transactions on Parallel and Distributed Systems, 2025, 36(3): 377−390

[39] Hong Zicong, Guo Song, Li Peng, et al. Pyramid: A layered sharding blockchain system [C]//Proc of the 40th IEEE Int Conf on Computer Communications. Piscataway, NJ: IEEE, 2021: 1−10

[40] The MultiversX Team. Elrond: A highly scalable public blockchain via adaptive state sharding and secure proof of stake [EB/OL]. (2019-06-19)[2025-08-25].https://elrond.com/assets/files/elrond-whitepaper.pdf

[41] Hong Zicong, Guo Song, Zhou Enyuan, et al. Prophet: Conflict-free sharding blockchain via Byzantine-tolerant deterministic ordering [C]//Proc of the 42nd IEEE Int Conf on Computer Communications. Piscataway, NJ: IEEE, 2023: 1−10

[42] Qi Xiaodong, Li Yi. LightCross: Sharding with lightweight cross-shard execution for smart contracts [C]//Proc of the 43rd IEEE Int Conf on Computer Communications. Piscataway, NJ: IEEE, 2024: 1681−1690

[43] Xu Cheng, Zhang Ce, Xu Jianliang, et al. SlimChain: Scaling blockchain transactions through off-chain storage and parallel processing[J]. Proceedings of the VLDB Endowment, 2021, 14(11): 2314−2326

[44] Gray J N. Notes on data base operating systems [C]//Operating Systems: An Advanced Course. Berlin: Springer, 1978: 393−481

[45] Yin M, Malkhi D, Reiter K, et al. HotStuff: BFT consensus with linearity and responsiveness [C]//Proc of the 38th ACM Symp on Principles of Distributed Computing. New York: ACM, 2019: 347–356

[46] Urdaneta G, Pierre G, Steen M V. A survey of DHT security techniques[J]. ACM Computing Surveys, 2011, 43(2): 1−49

[47] Kreps J, Narkhede N, Rao Jun. Kafka: A distributed messaging system for log processing [C]//Proc of the NetDB. Athens: NetDB, 2011: 1−7

[48] Bacho R, Loss J. On the adaptive security of the threshold BLS signature scheme [C]//Proc of the 29th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2022: 193−207

[49] Kung H T, Robinson J T. On optimistic methods for concurrency control[J]. ACM Transactions on Database Systems, 1981, 6(2): 213−226

[50] Fekete A, Liarokapis D, Neil E, et al. Making snapshot isolation serializable[J]. ACM Transactions on Database Systems, 2005, 30(2): 492−528

[51] Ongaro D, Ousterhout J. In search of an understandable consensus algorithm [C]//Proc of the 23rd USENIX Conf on USENIX Annual Technical Conf. Berkeley, CA: USENIX Association, 2014: 305−320

[52] Liu Yizhong, Liu Jianwei, Wu Qianhong, et al. SSHC: A secure and scalable hybrid consensus protocol for sharding blockchains with a formal security framework[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(3): 2070−2088

[53] Li Songze, Yu Mingchao, Yang Chien-Sheng, et al. Polyshard: Coded sharding achieves linearly scaling efficiency[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 249−261

[54] Krol M, Ascigil O, Rene S, et al. Shard scheduler: Object placement and migration in sharded account-based blockchains [C]//Proc of the 3rd ACM Conf on Advances in Financial Technologies. New York: ACM, 2021: 43−56

[55] Li Mingzhe, Wang Wei, Zhang Jin. LB-Chain: Load-balanced and low-latency blockchain sharding via account migration[J]. IEEE Transactions on Parallel and Distributed Systems, 2023, 34(10): 2797−2810

[56] Wang Xin, Li Bo, Jia Linpeng, et al. Orbit: A dynamic account allocation mechanism in sharding blockchain system [C]//Proc of the 44th IEEE Int Conf on Distributed Computing Systems. Piscataway, NJ: IEEE, 2024: 333−344

[57] Zhang Yuanzhe, Pan Shirui, Yu Jiangshan. TxAllo: Dynamic transaction allocation in sharded blockchain systems [C]//Proc of the 39th IEEE Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2023: 721−733

[58] Khan S, Luo Fei, Zhang Zijian, et al. A survey on X.509 public-key infrastructure, certificate revocation, and their modern implementation on blockchain and ledger technologies[J]. IEEE Communications Surveys & Tutorials, 2023, 25(4): 2529−2568

[59] Chen Xiao, Huang Muhong, Tian Yifan, et al. Internet of vehicles data sharing scheme via blockchain sharding[J]. Journal of Computer Research and Development, 2024, 61(9): 2246−2260 (in Chinese)
(陈骁, 黄牧鸿, 田一凡, 等. 基于分片区块链的车联网数据共享方案[J]. 计算机研究与发展, 2024, 61(9): 2246−2260)

[60] Huang Junqin, Kong Linghe, Wang Jingwei, et al. Secure data sharing over vehicular networks based on multi-sharding blockchain[J]. ACM Transactions on Sensor Networks, 2024, 20(2): 1−23

[61] Makhdoom I, Zhou I, Abolhasan M, et al. PrivySharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities[J]. Computers & Security, 2020, 88: 101653

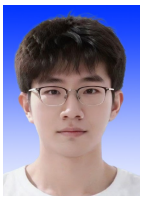
[62] Sun Xiaoqiang, Yu F R, Zhang Peng, et al A survey on zero-knowledge proof in blockchain[J]. IEEE Network, 2021, 35(4): 198−205

[63] Acar A, Aksu H, Uluagac A, et al. A survey on homomorphic encryption schemes: Theory and implementation[J]. ACM Computing Surveys, 2018, 51(4): 1−35

[64] Manpreet K, Shikha G, Deepak K, et al. IPFS: An off-chain storage solution for blockchain [C]//Proc of the 2nd Int Conf on Recent Innovations in Computing: ICRIC 2022, Volume 1. Berlin: Springer, 2023: 513−525

[65] Liu Andi, Liu Yizhong, Wu Qianhong, et al. CHERUBIM: A secure and highly parallel cross-shard consensus using quadruple pipelined two-phase commit for sharding blockchains[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 3178−3193

[66] Zhang Xiao, Qin Chunling, Wang Wenshou, et al. Survey on blockchain benchmarks[J]. Journal of Software, 2025, 36(7): 3151−3183 (in Chinese)
(张孝, 秦春玲, 王文收, 等. 区块链测试基准综述[J]. 软件学报, 2025, 36(7): 3151−3183)



Zhang Zitong, born in 2000. Master candidate. His main research interest includes blockchain scalability technology.

张滋炯, 2000 年生。硕士研究生。主要研究方向为区块链可扩展技术。



Han Haoxiang, born in 1998. PhD candidate. His main research interests include blockchain and blockchain sharding.

韩昊翔, 1998 年生。博士研究生。主要研究方向为区块链、区块链分片。



Shi Chaoming, born in 1999. PhD candidate. His main research interest includes blockchain and off-chain scaling.

史晁铭, 1999 年生。博士研究生，主要研究方向为区块链、链下扩容。



Yan Zheng, born in 1972. PhD, professor, PhD supervisor. Her main research interests include network security and trust management.

闫　峥, 1972 年生。博士，教授，博士生导师。主要研究方向为网络安全、信任管理。



Wang Haiguang, born in 1973. PhD, professor. His main research interests include 5G security and PKI.

王海光, 1973 年生。博士，研究员。主要研究方向为 5G 安全、公钥基础设施。



Li Tieyan, born in 1971. PhD, professor. His main research interests include network security and digital certificate.

李铁岩, 1971 年生。博士，研究员。主要研究方向为网络安全、数字证书。