基于数据可用性采样与智能合约的去中心化存储方法

陈中[1,2],李晓风[1,2],赵赫[1],张凌浩[1,2]

(1.中国科学院合肥物质科学研究院,安徽 合肥 230031;2.中国科学技术大学,安徽 合肥 230026)

摘 要: 去中心化存储具有高可用性和强扩展性,但由于数据分散存储在多个节点上,存在数据存取速度慢、操作复杂等问题,用户体验相比中心化存储较差。为此,利用数据可用性采样技术,在保持方法去中心化特性的同时,融合中心化存储的优势。在数据可用性采样技术中,多个节点从单个数据拥有者处获取随机选取的规模较小的数据子集,其常与纠删码结合使用来提高数据可用性。基于数据可用性采样技术,引入去中心化的存储提供者,以单对单的形式为用户提供服务,同时利用数据保障者来监督存储提供者并为用户数据提供保障。设计一套较为完备的存储方法,来实现高可用的数据存储,并利用区块链与智能合约来增强其去中心化程度。通过支持再质押模式,并采用低计算资源消耗的存储证明算法,来提高节点的加入意愿。为解决较大的数据规模与数据保障者有限的带宽资源之间的矛盾,提出延迟确认机制。实验与分析结果表明,在该方法下,恶意节点共谋概率仅为 2.43×10^{-3},数据可用性采样结果不可信的概率仅为 2.93×10^{-4},在 300 万次模拟实验中发生数据不可用的次数为 0,中心化节点数为 0,为 1 MiB 大小的文件生成存储证明仅需 3.51 ms。该方法在提高用户友好性和节点友好性的同时,实现了高可用的数据存储,为优化去中心化存储提供了可行的技术路径。

关键词: 数据可用性采样;区块链;智能合约;去中心化存储;纠删码

中图分类号: TP311 **文献标志码:** A **DOI:** 10.19678/j.issn.1000-3428.0070289

Decentralized Storage Method Based on Data Availability Sampling and Smart Contracts

CHEN Zhong[1,2], LI Xiaofeng[1,2], ZHAO He[1], ZHANG Linghao[1,2]

(1. Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, Anhui, China;
2. University of Science and Technology of China, Hefei 230026, Anhui, China)

【Abstract】 Decentralized storage offers high availability and scalability. However, owing to the decentralized storage of data across multiple nodes, issues such as slow data access and complex operations arise, resulting in a poorer user experience compared to centralized storage. To address this, a data availability sampling technology is employed, which maintains the decentralized nature of the method while incorporating the advantages of centralized storage. In data availability sampling technology, multiple nodes obtain a smaller, randomly selected subset of data from a single data owner. This technology is often combined with erasure coding to enhance data availability. Based on data availability sampling technology, decentralized storage providers are introduced to serve users on a one-to-one basis, and data guarantors supervise storage providers and provide guarantees for user data. A comprehensive storage method is designed to achieve highly available data storage, and blockchain and smart contracts are employed to enhance decentralization. By supporting a repledging model and adopting a storage-proof algorithm with low computational resource consumption, the willingness of the nodes to join is increased. To resolve the contradiction between large data scales and the limited bandwidth resources of data guarantors, a delayed confirmation mechanism is proposed. Experimental and analytical results show that under this method, the probability of malicious node collusion is only 2.43×10^{-3}, the probability of untrustworthy data availability sampling results is only 2.93×10^{-4}, the number of data unavailability occurrences is 0 in 3 million simulation experiments, the number of centralized nodes is 0, and generating storage proofs for a 1 MiB file takes only 3.51 ms. This method achieves highly available data storage while improving user-friendliness and node-friendliness, providing a feasible technical path for optimizing decentralized storage.

【Key words】 data availability sampling; blockchain; smart contract; decentralized storage; erasure code

基金项目: 国家重点研发计划(2021YFB2700800)。

作者简介: 陈中,男,硕士研究生,主研方向为区块链存储技术;李晓风,研究员、博士、博士生导师;赵赫(通信作者),正高级工程师、博士;张凌浩,硕士研究生。

收稿日期: 2024-08-26 **修回日期:** 2024-11-14 **E-mail:** zhaoh@hfcas.ac.cn

0 引言

随着数字化进程的加快，各种形式的数据不断产生和积累，可靠的数据存储成为数字社会的基础。数据存储分为中心化存储和去中心化存储[1]。中心化存储因管理简单、用户体验感好[2]而被广泛应用，但其存在单点故障、扩展性差等问题。因此，近年来去中心化存储得到广泛研究。

部分研究团队基于区块链技术，提出 Filecoin[3]、Sia[4]、Storj[5] 等去中心化存储方法。Filecoin 和 Sia 通过强制节点质押通证与惩罚机制来督促节点诚实行事，且均实现了完全去中心化。相较之下，Storj 由于存在中心化节点，使其牺牲了部分去中心化特性。但是，Storj 中数据的存储期限完全由用户定义，而 Filecoin 和 Sia 在数据存储期限方面对用户有较多限制，使其用户体验较差。在上述 3 种方法的基础上，文献[6-7]进行了一些改进。文献[6]为了解决 Filecoin 的存储证明算法计算资源消耗较高的问题，提出一种改进的存储时间证明算法，但相较 Sia 和 Storj，其计算资源消耗依然较高，并且相较 Storj，文献[6]方法在数据存储期限方面对用户的限制依然较多。文献[7]采用了低计算资源消耗的存储证明算法，且通过设计结算机制引入对用户缩短存储期限的支持，但用户仍然无法便捷地延长存储期限。因此，相较用户体验好的中心化存储，现有的去中心化存储方法的用户友好性仍有较大的提升空间。

本文利用数据可用性采样技术[8]，将中心化存储与去中心化存储的优势相结合。在数据可用性采样技术中，多个节点从单个数据拥有者处获取随机选取的规模较小的数据子集，从而减少对节点的带宽和存储需求。本文设想由一个高带宽、大存储的存储提供者单对单为用户提供服务，并由多个低带宽、小存储的数据保障者来监督存储提供者，以保障数据可用性。数据可用性采样技术常与纠删码技术结合使用，纠删码技术将原始数据分割为片段并增加冗余，使得仅需部分编码后的数据即可恢复原始数据。本文基于上述技术，提出了一种去中心化存储方法，并利用区块链与智能合约来增强其去中心化程度。

此外，尽管部分去中心化存储方法（如 Filecoin[3]和 Sia[4]）通过强制节点质押通证与惩罚机制有效遏制了恶意行为，但其锁定的通证未得到充分利用，节点无法获得超额回报，间接降低了节点的加入意愿。为了提高节点的收益能力，EigenLayer[9]提出了再质押模式。在该模式中，节点仅需质押一份通证，便可在多个质押系统中履行职责，从而获得更多收益。这一模式不仅提升了节点的加入意愿，也推动了质押项目的发展。因此，本文方法采用再质押模式来吸引节点加入。本文的主要贡献如下：

1）在提高用户体验的同时，实现完全去中心化。本文方法使高性能节点直接对接用户，提高了用户体验。同时，本文基于数据可用性采样技术，激励普通节点来监督高性能节点，保障了存储的去中心化与数据可用性。

2）充分利用不同性能的节点，并增强各类节点的加入意愿。高性能节点在本文方法中通过承担更多职责来赚取更高收益，提高了其加入意愿。本文设计延迟确认机制，并采用低计算资源消耗的存储证明算法，来降低对普通节点的计算资源与带宽要求，使得普通节点也能充分利用其闲置资源来赚取收益。同时，本文方法对再质押模式的支持，增强了各类节点的加入意愿。

1 相关技术

1.1 数据可用性采样与纠删码

数据可用性采样[8]是一种检查数据可用性的方法，无须任何节点下载完整数据，即可确认数据可用与否。在数据可用性采样中，多个节点从单个数据拥有者处下载总数据中随机选择的一小部分数据[8]，并进行验证。对于单个节点，如果其能够获得采样的数据，且该数据与最初数据拥有者声明一致，则其认为数据可用。随着认为数据可用的节点变多，全网对该数据可用的置信度增加。数据可用性采样能够减少对节点的带宽和存储需求，使硬件资源受限的轻节点也能为网络的安全性和吞吐量作出贡献，有助于提高区块链的可扩展性。

数据可用性采样通常结合纠删码使用，如里德-所罗门码[10]、再生码[11]等，以便在数据拥有者出故障或作恶时能够恢复原始数据。纠删码是一种数据保护方法，其将原始数据拆分成多个分片，并用冗余数据片段对原始数据进行扩展和编码，创建可用于数据恢复的奇偶校验分片。换言之，纠删码将原始数据分为 d 个数据分片，并计算出 y 个奇偶校验分片，使用这 $(d+y)$ 个分片中的任意 d 个分片即可恢复原始数据[12]。

1.2 区块链与智能合约

区块链是一种分布式账本，包含不断增长的区块列表，这些区块通过哈希算法紧密连接在一

起[13]。主流的区块链平台包括比特币[14]、以太坊[15]、Solana[16]等。共识协议是区块链技术的基本组成部分，主流的共识协议通常基于工作量证明[17]、权益证明[18]等。工作量证明要求节点利用算力解决密码难题，由第一个解决难题的节点创建新的区块[17]。而在权益证明中，节点需要将通证质押到区块链上，奖励机制会奖励诚实的质押者，同时惩罚机制能够遏制恶意行为[18]。

智能合约是一种运行在区块链上、无法被第三方操控或更改的计算机程序[19]，主流的支持智能合约的区块链包括以太坊、Solana[16]、波场[20]等。区块链不仅负责存储智能合约的源代码，还负责智能合约的执行。智能合约明确定义了用户与其交互的方式与条件，其代码是公开且可验证的[21]。智能合约包含设有预定条件的代码，其主要优点是在满足特定条件时自动执行确定的代码，这消除了对可信中介的需求[22]。

1.3 去中心化存储方法

部分研究团队基于区块链技术，提出了 Filecoin[3]、Sia[4]、Storj[5]等去中心化存储方法。Filecoin 是一个允许任何人存储和检索数据的点对点网络，通过激励措施确保数据的可靠持续存储。Sia 基于密码学构建了一个无须信任的云存储市场，允许存储需求者与存储提供者直接进行交易。Storj 旨在提供安全且高性价比的存储服务，存储节点通过提供存储空间和带宽资源赚取收益。

然而，上述方法仍存在一定缺陷。Filecoin 使用了高计算资源消耗的存储证明算法，使得其对节点的硬件要求较高，限制了对普通主机闲置资源的利用。Sia 采用工作量证明[17]作为共识算法，能源消耗大，并且其用户在存储期限内无法根据自身需求缩短期限以获得退款，用户友好性较差。Storj 引入了卫星来负责存储计费、数据审计和数据恢复，但目前所有卫星均由 Storj 官方运营[23]，使该方法牺牲了部分去中心化特性。在上述 3 种方法的基础上，文献[6-7]进行了一些改进。文献[6]提出一种完全去中心化的基于智能合约的数据连续存储方案，并改进了存储时间证明算法，相较 Filecoin，该方法的计算资源消耗较低。然而，相较 Sia 和 Storj，改进存储时间证明算法的计算资源消耗依然较高，并且文献[6]未考虑用户对数据存储期限可能进行的更改，用户体验较差。文献[7]结合区块链、智能合约、纠删码等技术，提出一个应用于区块链网络环境的去中心化存储空间交易系统，为节点出租其闲置存储资源提供了平台。文献[7]方法采用了低计算资源消耗的存储证明算法，且通过设计租金交付机制，引入对用户缩短存储期限的支持。

2 系统框架

本文的研究目标是将中心化存储与去中心化存储的优势相结合，并充分利用不同性能的节点，增强其加入意愿。本文方法利用高性能节点直接服务用户来实现较好的用户体验，并基于数据可用性采样技术，充分利用普通节点来监督高性能节点，从而保障数据可用性。同时，本文方法基于区块链上的智能合约来实现完全去中心化。使用链接证明和延迟确认机制，使普通节点也能充分利用其闲置资源。此外，通过支持再质押模式，增强各类节点的加入意愿。本文方法的基本假设是普通节点中至少有 51%是诚实的。

本文方法系统框架如图 1 所示，涉及用户、存储提供者、数据保障者和状态更新者 4 种角色。存储提供者、数据保障者和状态更新者均通过在智能合约中质押指定数量的通证来参与系统，且本文方法利用再质押模式来提高其加入意愿。存储提供者负责向用户提供数据存储服务。用户的数据在存储前

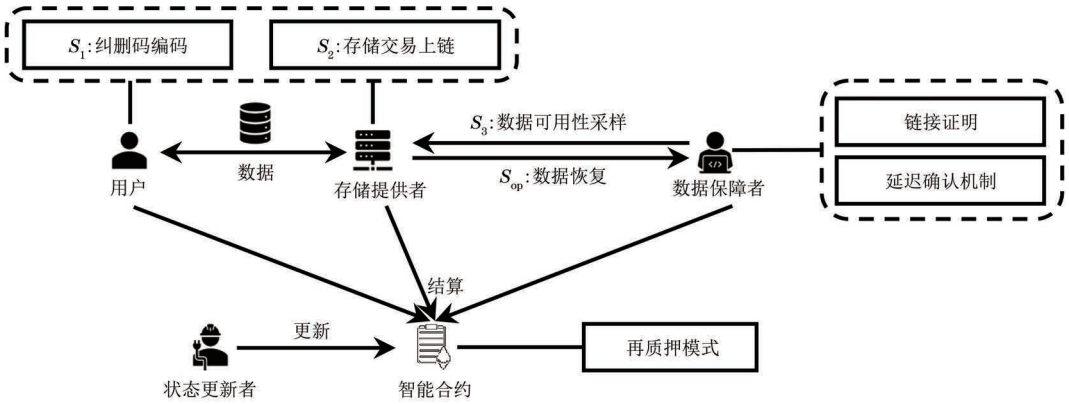

图 1 本文方法系统框架

Fig. 1 System framework of the method in this paper

会先经过纠删码编码处理，随后，用户与存储提供者之间的存储交易会由用户提交至区块链，并由存储提供者在智能合约上确认。数据保障者对存储提供者进行数据可用性采样，并保存采样的分片，以防止存储提供者作恶或发生故障导致数据丢失。本文方法使用链接证明和延迟确认机制，以降低对数据保障者的计算资源与带宽要求。用户、存储提供者和数据保障者的收入和支出皆在区块链上的智能合约中结算，并由状态更新者负责智能合约的更新。

2.1 用户

用户具有数据存储需求，其选择某个存储提供者为其提供服务，双方就存储期限和存储费用协商一致。存储交易和协商结果保留在区块链上，对所有人可见。设本文方法部署在区块链上的智能合约为 R，用户根据其存储需求向 R 发送相应数量的通证，存储提供者、数据保障者和状态更新者的收入皆来源于用户的支出，最终在 R 上结算。在数据存储期间，用户可以向数据保障者提出采样需求，即要求全体数据保障者对存储提供者进行数据可用性采样，以保障存储提供者诚实行事。用户可以提前终止存储需求，并获得相应退款。

2.2 存储提供者

存储提供者需要在智能合约 R 中质押指定数量的通证。存储提供者不仅需要满足用户的数据存储需求，还需响应数据保障者的采样要求，必要时还需进行数据恢复。存储提供者通过正确履行职责赚取收益，且其质押的通证支持再质押。

存储提供者的生命周期包括加入、履行职责和退出。通过向 R 中质押指定数量的通证，任何人都可以成为存储提供者。存储提供者通过向用户提供数据存储服务来赚取通证。数据保障者从存储提供者获取分片以进行数据可用性采样，存储提供者向数据保障者收集分片以进行数据恢复，如图 2 所示。无论是意外还是恶意，未能正确履行其职责的存储提供者可能会失去其质押及赚取的部分或全部通证。存储提供者的质押和收益锁定在 R 中，存储提供者可以申请提取收益，也可同时申请退回质押并退出。为了提高存储提供者的加入意愿，其向 R 质押的通证可以选择再质押进流动性质押项目，以赚取额外收益。流动性质押类似活期存款，允许存入通证来赚取收益，但也存在一定的风险[24]。以太坊是全球最受欢迎的支持智能合约的区块链[25]，以太坊上的流动性质押项目包括 Lido[26]、Rocket Pool[27]等。

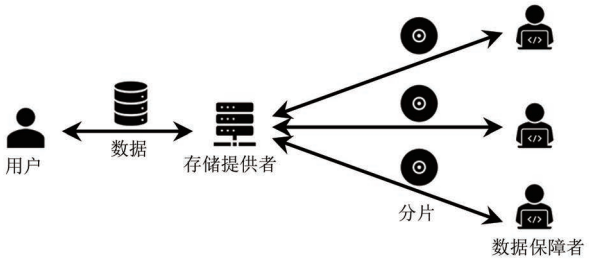

图 2 数据保障机制

Fig. 2 Data assurance mechanism

2.3 数据保障者

数据保障者需要在 R 中质押指定数量的通证，其不仅需要根据用户需求对存储提供者开展数据可用性采样，还需要将获取的分片存储指定期限，从而为用户存储的数据提供保障。

数据保障者的生命周期与存储提供者相同，数据保障者之间相互通信，通信内容包括用户需求信息、采样结果等，相互协作来履行职责。数据保障者需要定期接受存储检查，并提供存储证明。由于带宽资源受限，数据保障者依靠延迟确认机制，就某些事件结果达成一致。

每个数据保障者拥有身份私钥 c_{PRK} 和提款私钥 w_{PRK}。c_{PRK} 为数据保障者质押时的账户私钥，在其整个生命周期中不变。c_{PRK} 用于对其广播的消息签名，以便其他数据保障者确认消息来源。提款私钥可以控制该数据保障者质押及赚取的全部通证。提款公钥 w_{PUK} 和提款私钥 w_{PRK} 由数据保障者生成，在系统运行过程中会发生改变。

数据保障者之间通过广播消息进行通信，广播消息的结构如表 1 所示。数据保障者质押时的区块链地址及其对应的身份私钥是唯一的，且在其整个生命周期中不变。

表 1 数据保障者的广播消息结构

Table 1 The broadcast message structure of the data guarantor

字段	说明
address	数据保障者质押时的区块链地址
content	消息内容
signature	数据保障者的身份私钥对消息内容的签名

2.4 状态更新者

状态更新者需要在 R 中质押指定数量的通证，负责将本文方法产生的事件同步到区块链。状态更新者之间相互竞争，采用“先上链先得”的方式提议某次状态更新。

状态更新者的生命周期与存储提供者和数据保障者相同。从本文方法到区块链的状态更新包括某

些事件的结果、向区块链的某个账户发送通证以及更新角色列表。首先，由状态更新者向 R 提议状态更新；然后，数据保障者在 R 上对该更新投票，不投票则默认赞成；最后，由 R 根据投票结果自动判定是否执行该状态更新。提议的状态更新有效的状态更新者将获得通证；提议的状态更新无效的状态更新者将被罚没，即失去其质押及赚取的全部通证，其中部分会奖励给对不合法状态更新投反对票的数据保障者。

3 存储流程

数据存储流程包括纠删码编码、存储交易上链、数据可用性采样和数据恢复。为了保障数据可用性，首先由用户或存储提供者对数据进行纠删码编码；然后，用户将存储交易上链，存储提供者在链上确认这笔交易；接着，数据保障者对存储提供者开展数据可用性采样，必要时，由其他存储提供者进行数据恢复。

3.1 纠删码编码

为了防止存储提供者扣留一小部分数据使得完整数据不可用，用户或存储提供者需要对数据进行纠删码编码。

用户可以选择由自己在本地对数据进行纠删码编码，也可要求多个存储提供者为其数据提供编码服务。本文设计一对多纠删码编码机制，如图 3 所示，以督促存储提供者诚实编码。

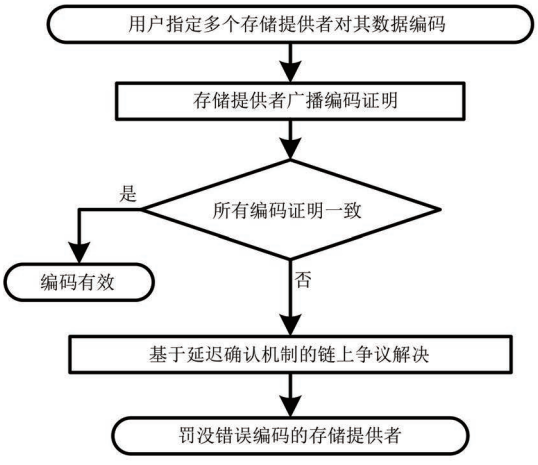

图 3 一对多纠删码编码流程

Fig.3 The process of one-to-many erasure coding

在一对多纠删码编码机制中，用户选择 k 个存储提供者为其数据进行纠删码编码，这些存储提供者各自独立编码，并相互监督。k 越大，用户的花费越高，但能够获得更高的编码正确性保障。每个被指定的存储提供者首先对数据进行纠删码编码，生成一定数量的分片，然后基于这些分片生成编码证明。编码证明生成算法描述如算法 1 所示。

算法 1 编码证明生成算法

输入 分片列表 shards

输出 编码证明 h_l

```
1. x←length of shards//获取列表长度
2. for i←1 to x
3.    l[i]←Hash(shards[i])//按序计算哈希值
4. hl←Hash(l)//对计算得到的列表进行哈希
5. return hl
```

如算法 1 所示，每个被指定的存储提供者对生成的所有分片按序计算哈希值，并将所有哈希值组合成一个列表 l，再对列表 l 进行哈希，生成编码证明 h_l。每个存储提供者需在规定期限内向所有数据保障者广播 h_l。假设生成的分片总数为 x，由于存在单层循环结构，因此算法 1 的时间复杂度为 $O(x)$。由于过程中存储了长度为 x 的列表 l，因此算法 1 的空间复杂度为 $O(x)$。

若所有存储提供者的编码证明不一致，则基于延迟确认机制，由状态更新者和数据保障者完成链上争议解决，并罚没错误编码的存储提供者，同时奖励正确编码的存储提供者。若所有的编码证明一致，则乐观地认为编码有效，但某存储提供者或数据保障者可以向此次编码发起挑战。基于延迟确认机制，由状态更新者和数据保障者在区块链上协作，确认挑战结果。若挑战成功，则罚没错误编码的存储提供者，并奖励挑战者；若挑战失败，挑战者将被罚没，即失去其质押及赚取的全部通证。

3.2 存储交易上链

数据存储交易流程如图 4 所示。在完成对数据的纠删码编码后，用户将存储交易提交至区块链上的智能合约，并向智能合约发送通证，通证的数量需与存储交易中声明的存储费用一致。存储提供者在链下接收用户的数据，并根据交易中的数据大小、存储期限、存储费用等信息，决定是否接受这笔交易。如果存储提供者选择接受这笔存储交易，则需要在规定时间内向智能合约确认这笔交易。如果存储提供者在规定时间内未确认这笔交易，或向智能合约表示拒绝，则这笔交易无效，由智能合约向用户退款。

如果对数据进行纠删码编码的角色是用户而非存储提供者，则存储提供者在确认这笔交易前，还需额外检查用户提供的编码证明。存储提供者在链下接收用户要存储的数据，然后根据编码证明生成算法独立生成编码证明，并与用户提交至区块链的交易中声明的编码证明进行比对，如果不一致，则拒绝这笔交易。

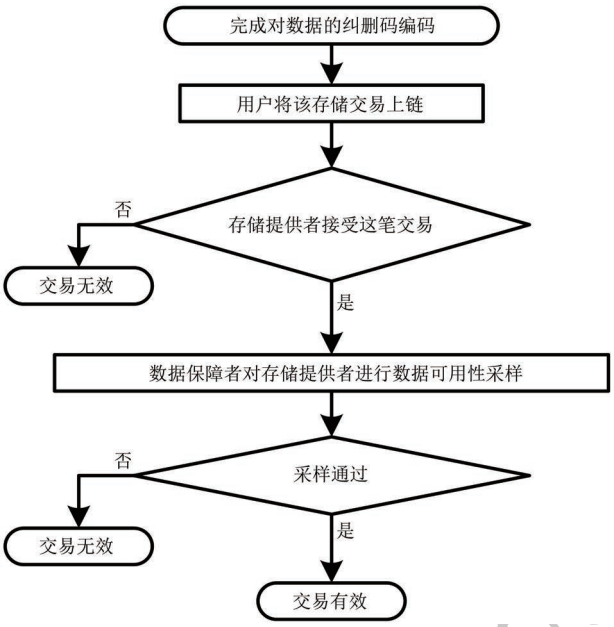

图 4 数据存储交易流程

Fig. 4 The process of data storage transaction

在存储提供者确认这笔交易后，数据保障者对该存储提供者进行数据可用性采样。如果采样通过，则该笔数据存储交易正式生效，数据保障者将采样的分片也存储相应期限，并定期接受存储检查；否则，交易无效，该存储提供者将被罚没，即失去其质押及赚取的全部通证，并由智能合约向用户退款。

用户在存储期间可以延长或缩短存储期限，延长存储期限需要支付额外费用，缩短存储期限会获得相应退款。

3.3 数据可用性采样

在数据可用性采样过程中，数据保障者使用自己的提款私钥作为种子，随机从存储提供者处获取分片并验证。根据验证结果，数据保障者广播由身份私钥签名的投票，投票结果分为通过和不通过。基于延迟确认机制，全体数据保障者的投票结果共同决定了此次数据可用性采样是否通过。数据可用性采样的算法描述如算法 2 所示。

算法 2 数据可用性采样算法

输入 提款私钥 w_{PRK}，交易信息中的编码证明 h_l

输出 true/false

```
1. l←get hash list from Storage Provider//获取哈希列
//表 l
2. if unable to get hash list or Hash(l)! =h_l then
3.     return false//判断是否可获取且与链上一致
4. seed←w_PRK//将自己的提款私钥作为随机数种子
5. sampleShardIdSet←GenRandomSet(seed)
6. for i in sampleShardIdSet//对于随机数集合中的每
//一项
7.     if unable to get shard_i from Storage Provider then
8.         return false//判断是否可获取
9.     else if Hash(shard_i)! =l[i] then
10.        return false//判断哈希值是否一致
11.    else
12.        continue
13. return true
```

如算法 2 所示，数据保障者首先从存储提供者处获取列表 l，并将 l 的哈希值与区块链上对应的存储交易中的 h_l 进行比对：如果相等，则继续从存储提供者处获取分片；若无法获取 l 或不相等，则投出不通过票。数据保障者根据自己的提款私钥 w_{PRK} 决定要获取的分片，若存在从存储提供者处无法获取的分片，则投出不通过票。对于每个分片，数据保障者计算其哈希值并与列表 l 中的对应哈希值进行比较。如果所有分片的哈希值均与列表 l 中对应的哈希值一致，则投出通过票；否则，投出不通过票。假设每个数据保障者采样 s 个分片，由于存在单层循环结构，因此算法 2 的时间复杂度为 $O(s)$。由于过程中存储了大小为 s 的随机数集合，因此算法 2 的空间复杂度为 $O(s)$。

每个数据保障者均根据数据可用性采样算法进行投票并广播，然后基于延迟确认机制，全体数据保障者的投票结果共同决定此次数据可用性采样是否通过。如果通过票数不少于 $n\times51\%$（n 为数据保障者总数），则此次数据可用性采样通过，投出通过票的数据保障者将获得一定奖励，并将采样的分片存储，定期接受存储检查；否则，此次数据可用性采样不通过。

在数据存储期间，用户可以要求数据保障者对存储提供者进行数据可用性采样，但有频率限制。如果存储提供者未能通过存储期间的数据可用性采样，则该存储提供者将被罚没，并进入数据恢复流程。

3.4 数据恢复

若某存储提供者 S_A 在数据存储期间未能通过数据可用性采样，将对其进行罚没，即失去质押及赚取的全部通证。对于 S_A 在存的每个数据，采取“先自愿后强制”的原则，由 S_B 来进行数据恢复。换言之，按照“先上链先得”的方式，其他存储提供者自发接管 S_A 负责的某个数据。如果在一定时间内没有存储提供者自愿，则利用可验证随机函数[28]选取一个存储提供者进行数据恢复，S_B 的信息会记录在区块链的智能合约中。

S_B 需要在规定期限内向数据保障者收集分片，利用纠删码进行数据恢复。然后，S_B 需要接受数据

可用性采样，如果通过，S_B 和数据保障者将会获得一定的奖励，并且 S_B 正式接管该数据的存储；如果 S_B 未通过数据可用性采样，则将其罚没，并重复上述过程。

尽管本文方法中数据无法恢复的概率较小，但若连续 3 个存储提供者都未能恢复数据，则终止数据恢复，并将部分罚没的通证补偿给用户。

4 系统机制

4.1 存储证明算法

数据保障者需要将采样的分片存储，用于在存储提供者作恶或发生故障时为数据恢复提供保障。为了对数据保障者进行存储检查，本节提出一种名为链接证明的存储证明算法。

每隔一定时间，基于可验证随机函数[28]选出一名数据保障者，由其余数据保障者对其进行存储检查。该数据保障者在接受存储检查时，需要提供自己所有采样的分片的链接证明。链接证明中的“链接”体现在以下 2 个方面：

1)将分片数据与数据保障者的提款私钥链接。由于提款私钥可以控制该数据保障者质押及赚取的全部通证，使得该证明只能由数据保障者自己生成。

2)将本次证明与上次证明链接，使得数据保障者无法预先生成后续多次存储证明，避免出现数据保障者只存储链接证明而未存储分片数据的情况。

对于每个分片，数据保障者对该分片的数据、自己的提款私钥、自己上次提供的证明材料一起进行哈希运算，从而生成链接证明。如果是首次接受存储检查，则上次的证明材料的哈希值为空。由于哈希函数具有单向性，因此数据保障者还需公布自己当前的提款私钥及新的提款公钥。

通过存储检查的数据保障者将获得通证奖励，而在规定时间内未提供存储证明的数据保障者的质押会被扣减，且其需要在一定期限内补交自己的存储证明，超时未补交则将其罚没。奖励、扣减和罚没信息均由各个数据保障者在本地维护，并最终依靠状态更新者在区块链智能合约上结算。

由于数据保障者以自己的提款私钥作为种子，随机从存储提供者处获取分片来进行数据可用性采样，因此不同数据保障者存储的分片数据大概率不同。由于哈希函数的单向性，没有某分片数据的数据保障者无法计算出该分片的链接证明，也就无法验证该证明，因此，本文采取“乐观加检举”机制，即乐观地相信某个数据保障者的存储证明，直至其被检举，该机制也降低了数据保障者的验证负担。数据保障者在接受存储检查时，可以检举其他数据保障者的链接证明。所有数据保障者依靠延迟确认机制来确认检举结果。若检举成功，被检举者将被罚没，即失去其质押及赚取的全部通证，同时被检举者的部分通证会奖励给检举者；若检举失败，检举者将被罚没。

4.2 延迟确认机制

为了解决较大的数据规模与数据保障者有限的带宽资源之间的矛盾，本文提出延迟确认机制，如图 5 所示。该机制使数据保障者能够有充分的时间进行独立操作，并确保最终结果在一定时间后在区块链上得到确认。

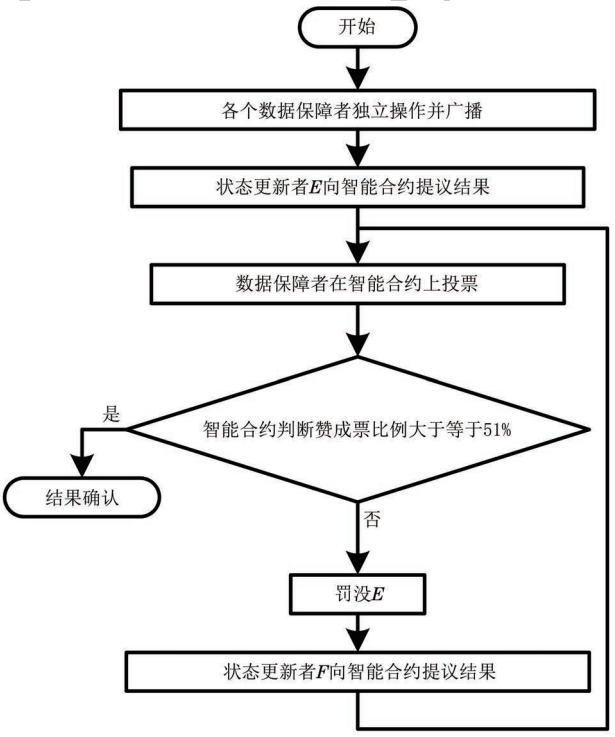

图 5 延迟确认机制流程

Fig.5 The process of the delayed confirmation mechanism

在延迟确认机制中，由状态更新者向智能合约提议结果，智能合约根据数据保障者在投票期内的投票情况，自动确认或否决提议。投票期的设置基于不同区块链平台的交易吞吐量与交易延迟，以确保所有诚实且活跃的数据保障者均有足够时间完成投票。投票期的时长固定在智能合约的代码中，一经设定将不可更改。在诚实且活跃的数据保障者占比不少于 51%的情况下，延迟确认机制可以有效运作。通过奖励正确投票的数据保障者并惩罚错误投票者，该机制激励数据保障者诚实且活跃行事。该机制的奖惩数据均在链下更新，由所有数据保障者共同维护，并在必要时由状态更新者在链上智能合约中完成结算。由于对错误提议结果的状态更新者

会有较重的惩罚，同时为了降低数据保障者的开销，因此在投票期内未进行投票的数据保障者将被视为默认赞成提议结果。

一对多纠删码编码机制、数据可用性采样和检举其他数据保障者的链接证明均采用延迟确认机制。

在一对多纠删码编码机制中，若所有存储提供者的编码证明不一致，则所有数据保障者依靠延迟确认机制来确认正确的编码证明。首先，每个数据保障者根据用户的数据，在本地独立生成编码证明；然后，状态更新者按照“先上链先得”的方式，向智能合约提议正确的编码证明。所有数据保障者在投票期内对该编码证明投票，不投票默认赞成。在投票期后，智能合约判断赞成票比例是否达到规定值，若达到，则正确的编码证明被确认，该状态更新者获得通证奖励，后续将罚没错误编码的存储提供者，并奖励正确编码的存储提供者；否则，罚没该状态更新者，并重复上述过程。

在数据可用性采样中，数据保障者各自进行数据可用性采样并广播签名后的投票，投票结果分为通过和不通过。在某数据的某次数据可用性采样期间，每个数据保障者只能提交一种投票，否则会被罚没，即失去其质押及赚取的全部通证。在规定时间后，状态更新者按照“先上链先得”的方式，依据自己本地的投票信息，向智能合约提议此次数据可用性采样的结果。所有数据保障者在投票期内依据自己本地的投票信息，在智能合约上对此次数据可用性采样的结果进行投票，不投票默认赞成。在投票期后，智能合约判断赞成票比例是否达到规定值，若达到，则此次数据可用性采样的结果被确认，该状态更新者将获得通证奖励；否则，罚没该状态更新者，并重复上述过程。

对于检举其他数据保障者的链接证明，检举者需要提供涉及的分片 ID、被检举者及相应的存储检查信息。然后，各个数据保障者从检举者或存储提供者处获取该分片，独立验证该检举。随后，状态更新者向智能合约提议检举结果。所有数据保障者在投票期内对该检举结果进行投票，不投票默认赞成。在投票期后，智能合约判断赞成票比例是否达到规定值，若达到，则此次检举结果得到确认，该状态更新者将获得通证奖励，后续将罚没检举者或被检举者；否则，罚没该状态更新者，并重复上述过程。

5 原型系统

基于上述方法设计，本节实现了原型系统，包括数据可用性采样模块和智能合约模块。数据可用性采样模块使用 Go 语言编写，开发工具为 GoLand；智能合约模块选用的区块链为以太坊，使用 Solidity 语言编写，开发工具为 Remix IDE。智能合约模块部署在以太坊的 Sepolia 测试网络上，地址为 0x00879C21Bb77ee42955a6dFf27E2bfBfd7ea14F2，通过 Sepolia 浏览器(https://www.oklink.com/zh-hans/sepolia-test)可以查看该智能合约。

数据可用性采样模块使用的纠删码为里德-所罗门码，基于由 Post[29] 编写的编解码库，使用的哈希算法为 Keccak-256，使用的数字签名算法为 ECDSA。该模块可以根据原始数据编码出指定数量的分片，用户需要指定数据分片数和奇偶校验分片数。依靠不少于数据分片数的分片，该模块可以恢复出原始数据。该模块采用 rand 包的 Seed 函数来设置随机数种子，使得数据保障者可以根据自己的提款私钥决定要获取的分片。

智能合约模块包含 25 个全局变量和 17 个函数。该模块使用地址到布尔值的映射，记录存储提供者、数据保障者和状态更新者。该模块利用关键字 require 实现了状态更新只能由状态更新者提议，基于全局变量 globalStateUpdateID 实现了对于某一状态更新，状态更新者之间按照“先上链先得”的方式相互竞争。对于状态更新者的某一提议，数据保障者调用 opposeProposal 函数来表示反对，若在投票期内某数据保障者未调用该函数，则由于 Solidity 语言会将布尔值自动初始化为 false，实现了数据保障者不投票则默认赞成的效果。在投票期后，智能合约遍历数据保障者列表，统计他们的投票，来判断该提议是否有效。例如，当某个存储提供者或数据保障者希望提取指定数量的通证时，状态更新者会向智能合约提交提案，提案内容包括所需提取的通证数量和接收账户信息。智能合约随后根据数据保障者的投票结果，自动处理并完成通证的发放。

智能合约模块实现了存储提供者、数据保障者和状态更新者的质押进入与再质押模式，为这 3 个角色各定义一个加入函数。以存储提供者为例，利用关键字 require 检查交易携带的通证(ETH)数，若其等于规定数量，则在相应的 map 中将该地址映射为 true。存储提供者、数据保障者和状态更新者加入时，可以选择是否再质押。以流动性质押项目 Lido[26] 为例，其上流通的通证为 stETH，这是一种变基的 ERC-20[30] 通证，账户中的 stETH 每天都会变多。若加入者选择再质押，则智能合约利用 Lido

的 fallback 函数，将加入者质押的 ETH 再质押进 Lido 中，换取 stETH 以赚取额外收益。当选择再质押的加入者申请退出时，依靠上述状态更新者提议、数据保障者投票的方式，智能合约调用 ERC-20 接口中的 transfer 函数，向选择再质押的加入者发送相应数量的 stETH。

智能合约模块实现了存储交易上链。用户通过调用智能合约的相应函数，向区块链提交存储期限、存储费用、编码证明、数据大小、为其服务的存储提供者的区块链地址，并附带指定数量的通证。智能合约会检查该交易携带的通证是否与用户声明的存储费用一致，若不一致则交易无效。然后，被用户指定的存储提供者需要在确认期限内决定是否接受这笔存储交易。如果存储提供者在规定时间内未确认这笔交易，或向智能合约表示拒绝，则这笔交易无效，由智能合约向用户退款。

智能合约模块实现了用户自定义存储期限。在数据存储期间，用户可以要求延长或缩短存储期限。当用户要求延长存储期限时，智能合约利用关键字 require 检查交易附带的 ETH 是否等于当初双方约定的单价乘以延长的期限，如果相等，则将存储期限延长。当用户要求缩短期限时，其缩短的期限不能超过剩余期限，用户可以指定该参数为 0 来提前终止这笔数据存储交易。智能合约按照当初约定的单价向用户退款，并缩短存储期限。

智能合约模块实现了延迟确认机制。智能合约利用关键字 require 限制结果只能由状态更新者提议，并记录投票期、提议的结果和该状态更新者。在投票期内，数据保障者调用 opposeProposal 函数来表示反对，不调用则默认赞成。投票期后，智能合约统计赞成票比例，如果该比例大于等于 51%，则将该结果记录在区块链上；否则，将该状态更新者的地址在相应 map 中的映射修改为 false，并重复上述过程。

6 分析与评估

6.1 一对多纠删码编码机制分析

在一对多纠删码编码机制中，k 个存储提供者为用户数据提供纠删码编码服务，这 k 个存储提供者相互监督。k 越大，用户的花费越高。在该机制中，仅当这 k 个存储提供者均作恶时，才可协商出一个错误但相同的编码证明，本文称该现象为共谋。设单个存储提供者作恶的概率为 q，则共谋概率为：

$$P_1=q^k \tag{1}$$

在式(1)中，k 增大会使得共谋概率降低。参照文献[31]中实验 4 的设置，分别取 q 为 5%、10%、20%、30%，利用式(1)计算不同 q 和 k 下的共谋概率，如表 2 所示，表中 $P_{5\%}$、$P_{10\%}$、$P_{20\%}$、$P_{30\%}$ 分别代表 q 为 5%、10%、20%、30%时的共谋概率。

表 2 恶意存储提供者共谋概率

Table 2 Collusion probability of malicious storage providers

k	$P_{5\%}$	$P_{10\%}$	$P_{20\%}$	$P_{30\%}$
2	2.50×10^{-3}	1.00×10^{-2}	4.00×10^{-2}	9.00×10^{-2}
3	1.25×10^{-4}	1.00×10^{-3}	8.00×10^{-3}	2.70×10^{-2}
4	6.25×10^{-6}	1.00×10^{-4}	1.60×10^{-3}	8.10×10^{-3}
5	3.13×10^{-7}	1.00×10^{-5}	3.20×10^{-4}	2.43×10^{-3}

当 $k=5$ 时，即使单个存储提供者作恶概率为 30%，共谋概率也仅为 2.43×10^{-3}，且由于挑战机制的存在，即使恶意存储提供者成功共谋，挑战者也可以向此次编码发起挑战，并赚取奖励。

6.2 数据可用性分析

6.2.1 存储提供者扣留数据分析

本节确定每个数据保障者采样的分片数，使得扣留数据的存储提供者以较大概率无法通过数据可用性采样。

设数据保障者总数为 n，每个数据保障者采样 s 个分片，则分片总数为 $n\times s$。数据保障者根据自己的提款私钥 w_{PRK} 决定要获取的分片，当且仅当这 s 个分片都可以从存储提供者处获取时投出通过票，且其哈希值均与用户声明的哈希值一致。设存储提供者扣留的分片比例为 $w(0<w<1)$，设 $p=1-w(0<p<1)$，则每个数据保障者投出通过票的概率为：

$$P_2=\frac{C_{pns}^{s}}{C_{ns}^{s}}=\prod_{i=1}^{s}\frac{pns-s+i}{ns-s+i}=\prod_{i=1}^{s}\left(\frac{pn-1}{n-1}+\frac{\frac{n-pn}{n-1}i}{(n-1)s+i}\right) \tag{2}$$

在式(2)中，s 变大时，$(n-1)s+i$ 变大，即分母变大，且 $\frac{n-pn}{n-1}i>0$，则 $\frac{\frac{n-pn}{n-1}i}{(n-1)s+i}$ 变小。又因为 $0<\frac{pns-s+i}{ns-s+i}<1$，则若 s 变大，连乘的每一项变小，且连乘的项数变多，导致 P_2 变小。换言之，如果存储提供者扣留数据，当每个数据保障者采样的分片数 s 增大时，每个数据保障者投出通过票的概率降低。

由于本文方法对数据进行纠删码编码，使得存储提供者需要扣留一定比例的分片才可使数据无法

恢复。假设存储提供者扣留的分片比例为 10%，即 $w=0.1$，则 $p=0.9$，仿照文献[32]中仿真实验 2 的设置，取数据保障者总数 n 为 50～1 500 间的特定值，利用式(2)计算在逐步增大的 s 下每个数据保障者投出通过票的概率。计算结果表明，当 s 增大到 44 时，概率首次低于 1%，如表 3 所示，表中，P_{43}、P_{44} 分别代表 s 为 43、44 时单个数据保障者投票结果错误的概率。

表 3 单个数据保障者投票结果错误的概率

Table 3 The probability of a single data guarantor making an incorrect voting decision

n	P_{43}	P_{44}
50	1.03×10^{-2}	9.24×10^{-3}
100	1.05×10^{-2}	9.47×10^{-3}
200	1.06×10^{-2}	9.58×10^{-3}
500	1.07×10^{-2}	9.65×10^{-3}
1 000	1.08×10^{-2}	9.67×10^{-3}
1 500	1.08×10^{-2}	9.68×10^{-3}

从表 3 可以看出，如果存储提供者扣留了 10% 的数据，当 s 增大到 44 时，每个数据保障者投出通过票的概率下降到 0.01 以下。因此，每个数据保障者采样 44 个分片。

依照文献[33]中实验 3 的设置，假设数据保障者总数 $n=500$，$p=0.9$，当且仅当通过票数不少于 255 时数据可用性通过，通过概率为：

$$P_3\approx\sum_{i=255}^{500}C_{500}^{i}0.01^{i}0.99^{500-i}\approx9.09\times10^{-363}\quad(3)$$

根据本文方法的基本假设，数据保障者中至少有 51%是诚实的，因此，恶意的存储提供者最多勾结 49%的数据保障者。假设存储提供者勾结了 245 个数据保障者，此时仅需 10 张通过票即可通过数据可用性采样，通过概率为：

$$P_4\approx\sum_{i=10}^{255}C_{255}^{i}0.01^{i}0.99^{255-i}\approx2.93\times10^{-4}\quad(4)$$

换言之，只要存储提供者扣留了 10%的分片，即使存储提供者勾结相当多的数据保障者，通过数据可用性采样的概率依然较小。因此，在后续数据恢复分析中假设存储提供者不会扣留数据。

6.2.2 数据恢复分析

本节确定数据分片与奇偶校验分片的比例，使得仅依靠诚实的数据保障者，就有较大概率可以恢复原始数据。

设数据保障者总数为 n，如第 6.2.1 节所述，每个数据保障者采样 44 个分片，因此取分片总数为 $44n$。根据本文方法的基本假设，数据保障者中至少有 51%是诚实的，因此通过模拟实验，对 $n\times51\%$个数据保障者采样的分片取并集，计算不同 n 下并集大小的最小值。仿照文献[32]中仿真实验 2 的设置，取 n 为 50～1 500 间的特定值，对于每个值模拟实验 50 万次，实验结果如表 4 所示。

表 4 将诚实数据保障者采样的分片取并集的模拟实验结果

Table 4 Simulation experiment results of taking the union of shards sampled by honest data protectors

n	分片总数	并集大小的最小值
50	2 200	844
100	4 400	1 691
200	8 800	3 413
500	22 000	8 634
1 000	44 000	17 358
1 500	66 000	26 097

从表 4 可以看出，当数据分片与总分片的比例为 840∶2 200(即 21∶55)时，本文方法仅依靠诚实的数据保障者，就有较大概率可以恢复原始数据。此时，数据分片与奇偶校验分片的比例为 21∶34。

6.3 数据存储的去中心化程度分析

本文方法基于智能合约，只要在区块链上质押指定数量的通证，任何人都可以成为存储提供者和数据保障者，从而实现了节点的完全去中心化。这些节点在各自的设备上存储数据，保障了数据存储的去中心化。同时，试图降低数据存储去中心化程度的做法收益较小而成本较高，因此可行性较小。

链接证明是基于数据保障者不会泄露自己的私钥，但数据保障者为了最大化收益，可能会将自己的私钥交给拥有广泛社会声誉的托管服务商，由其代为履行职责，并定期给数据保障者分红。由于被托管的不同数据保障者可能采样到相同的分片，对于该分片托管服务商只需存储一次，使得托管服务商的存储成本降低，如图 6 所示。

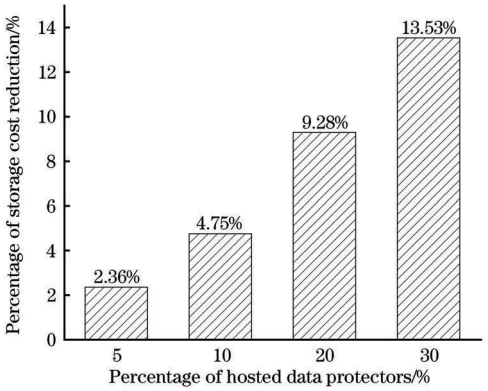

图 6 托管服务商存储成本降低比例

Fig.6 Reduction ratio of storage cost for hosting service providers

在图 6 中，依照文献[33]中实验 3 的设置，假设数据保障者总数 $n=500$。托管服务商托管的数据保障者越多，存储成本降低越明显。但如第 6.2 节所述，即使托管服务商托管了 30%的数据保障者，也对数据的可用性没有影响，并且，托管了 30%数据保障者的托管服务商的存储成本也仅下降 13.53%，而为了托管这 30%的数据保障者，托管服务商需要付出高昂的建立信任成本，来使得数据保障者愿意托管自己的私钥，托管服务商无利可图。

6.4 方法对比

部分研究团队基于区块链技术提出了一些去中心化存储方法，包括 Filecoin[3]、Sia[4]、Storj[5]、文献[6]方法和文献[7]方法，这些方法与本文方法的对比结果如表 5 所示。

表 5 本文方法与其他去中心化存储方法的对比

Table 5 Comparison of the method in this paper with other decentralized storage methods

方法	成为存储提供者需要质押	支持再质押	存储证明算法	存储证明算法计算资源消耗	完全去中心化	存储期限完全由用户定义
Filecoin	√	×	复制证明、时空证明	高	√	×
Sia	√	×	存储证明	低	√	×
Storj	×	—	扩展可检索性证明	低	×	√
文献[6]方法	√	×	改进存储时间证明	高	√	×
文献[7]方法	√	×	完整性证明	低	√	×
本文方法	√	√	链接证明	低	√	√

从表 5 可以看出：Filecoin 使用复制证明和时空证明作为其存储证明算法，存在计算资源消耗高的问题，同时，Filecoin 对用户数据的存储期限存在限制；Sia 的用户在存储期限内无法缩短期限以获得退款，用户友好性较差；Storj 中的卫星组件目前均由 Storj 官方运营[23]，牺牲了部分去中心化特性，且节点加入无需质押，作恶成本较低；文献[6]提出的改进存储时间证明的计算资源消耗依然较高，且在数据存储期限方面对用户有较多限制；文献[7]方法虽然支持用户缩短存储期限，但仍然不支持用户延长存储期限。此外，成为 Filecoin、Sia、文献[6]方法和文献[7]方法的存储提供者均需要质押，但这 4 种方法未支持再质押模式。

本文方法要求存储提供者和数据保障者在区块链上质押通证，以督促其诚实行事，并且本文方法支持再质押模式，而 Filecoin、Sia、文献[6]方法和文献[7]方法未支持再质押模式，因此本文方法的节点加入意愿相对较强。借助数据保障者的提款私钥，相比 Filecoin 和文献[6]方法，本文方法中的链接证明计算资源消耗较低。在本文方法中，只要在区块链上质押指定数量的通证，任何人都可以成为存储提供者、数据保障者和状态更新者，从而实现了完全去中心化。相比之下，Storj 牺牲了部分去中心化特性。此外，在本文方法中，用户在存储期间可以延长或缩短存储期限，延长存储期限需要支付额外费用，缩短存储期限会获得相应退款，因此，相较 Filecoin、Sia、文献[6]方法和文献[7]方法，本文方法实现了较高的用户友好性。

本文进行仿真实验来验证上述理论，使用的处理器为 Intel® Core™ i5-8250U CPU@1.60 GHz，实验采用 Go 语言和开发工具 GoLand。分别取大小为 1 MiB、8 MiB、64 MiB 和 512 MiB 的文件，进行 500 次实验取平均值，并以 1 MiB 为单位大小来处理实验数据，结果如表 6 所示。

表 6 数据对比结果

Table 6 Data comparison results

方法	存储证明生成时间/ms	中心化节点数	存储期限
Filecoin	44.99	0	至少 180 d
Sia	3.54	0	设定值
Storj	3.58	4	完全由用户定义
文献[6]方法	26.82	0	设定值
文献[7]方法	3.71	0	小于或等于设定值
本文方法	3.51	0	完全由用户定义

从表 6 可以看出：Filecoin 由于使用了零知识证明[34]，相较基于哈希函数的存储证明算法，其存储证明生成时间较长，且其存储期限至少 180 d；Sia 的用户在最初设定存储期限后，无法缩短期限以获得退款；Storj 包含 4 个中心化的卫星组件，削弱了该方法的去中心化特性；文献[6]方法的存储证明算法基于可验证延迟函数，其存储证明生成时间较长，且其用户在最初设定存储期限后无法更改；文献[7]方法的用户在最初设定存储期限后，仅支持缩短存储期限，不支持延长。相较上述 5 种方法，本文方法具有存储证明计算资源消耗较低、完全去中心化、用户体验较好的优势。

7 结束语

中心化存储中的存储节点通常具有较高的性能，能够提供较好的用户体验，但其存在单点故障、扩展性差等问题。现有的去中心化存储方法未能充分利用高性能节点，导致用户体验较差。本文方法通过直接将高性能节点与用户对接，提升了用户体验。同时，高性能节点也能够通过承担更多职责来赚取更高收益，从而增强了其加入意愿。基于数据可用性采样技术，本文方法激励普通节点来监督高性能节点，保障了存储的去中心化与数据可用性。此外，本文方法采用再质押模式进一步增强了各类节点的加入意愿。本文方法不仅实现了较高的用户友好性和节点友好性，还基于区块链与智能合约实现了完全去中心化，并通过原型系统进行了验证。

分析结果表明，本文提出的一对多纠删码编码机制能够有效防止节点共谋，当参数设置为 5 时，即使单个存储提供者作恶概率达到 30%，共谋概率也仅为 2.43×10^{-3}，并且本文方法通过挑战机制进一步遏制了节点的共谋行为。本文方法中的数据可用性采样结果具有较高的可信度，只要存储提供者扣留了 10%的分片，即使其勾结了全网 49%的数据保障者，数据可用性采样通过的概率也仅为 2.93×10^{-4}。本文方法通过设置数据分片与奇偶校验分片的比例为 21∶34，实现了较高的数据可用性，在 300 万次模拟实验中，发生数据不可用的次数为 0。本文设计了延迟确认机制，并支持再质押模式，能够有效吸引节点加入，充分利用闲置资源，为优化去中心化存储提供可行的技术路径。下一步工作将致力于增强检举机制的活性以及降低存储提供者的带宽压力，并结合可信硬件进一步提高数据存储的去中心化程度。此外，本文采用再质押模式来增强节点的加入意愿，后续工作将致力于持续保留高质量的节点。

参考文献

[1] KHAYRUTDINOV D D, VOLKOV D A, MUKHINA A G. The model and application of data storage in a decentralized network[C]//Proceedings of the 25th IEEE International Conference of Young Professionals in Electron Devices and Materials (EDM). Washington D. C., USA: IEEE Press, 2024: 1840-1843.

[2] SHARMA P, JINDAL R, BORAH M D. Blockchain technology for cloud storage: a systematic literature review[J]. ACM Computing Surveys, 2021, 53(4): 1-32.

[3] BENET J. Filecoin[EB/OL]. [2024-06-10]. https://filecoin.io/.

[4] VORICK D, CHAMPINE L. Sia[EB/OL]. [2024-06-10]. https://sia.tech/.

[5] WILKINSON S. Storj[EB/OL]. [2024-06-10]. https://www.storj.io/.

[6] WANG K, WU Q H, HAN T X, et al. DCSS: a smart contract-based data continuous storage scheme[C]// Proceedings of the 5th ACM International Symposium on Blockchain and Secure Critical Infrastructure. New York, USA: ACM Press, 2023: 53-63.

[7] 张水海, 孙昊驿, 孙逸伟, 等. 一种区块链网络下的去中心化存储空间交易系统[J]. 应用科学学报, 2022, 40(4): 583-599.
ZHANG S H, SUN H Y, SUN Y W, et al. A decentralized storage space trading system under blockchain network[J]. Journal of Applied Sciences, 2022, 40(4): 583-599. (in Chinese)

[8] AL-BASSAM M, SONNINO A, BUTERIN V. Fraud and data availability proofs: maximising light client security and scaling blockchains with dishonest majorities[EB/OL]. [2024-06-10]. https://arxiv.org/abs/1809.09044.

[9] KANNAN S. EigenLayer[EB/OL]. [2024-06-23]. https://www.eigenlayer.xyz/.

[10] REED I S, SOLOMON G. Polynomial codes over certain finite fields[J]. Journal of the Society for Industrial and Applied Mathematics, 1960, 8(2): 300-304.

[11] DIMAKIS A G, GODFREY P B, WU Y N, et al. Network coding for distributed storage systems[J]. IEEE Transactions on Information Theory, 2010, 56(9): 4539-4551.

[12] STEVAN A, LAVAUR T, LACAN J, et al. Performance evaluation of polynomial commitments for erasure code based information dispersal[C]// Proceedings of the 10th International Conference on Information Systems Security and Privacy. Washington D. C., USA: IEEE Press, 2024: 522-533.

[13] 王群, 李馥娟, 倪雪莉, 等. 区块链数据形成与隐私威胁[J]. 计算机工程, 2023, 49(8): 1-12.
WANG Q, LI F J, NI X L, et al. Data formation and privacy threat of blockchain[J]. Computer Engineering, 2023, 49(8): 1-12. (in Chinese)

[14] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system[EB/OL]. [2024-06-18]. https://bitcoin.org/bitcoin.pdf.

[15] BUTERIN V. A next-generation smart contract and decentralized application platform[EB/OL]. [2024-06-18]. https://ethereum.org/en/whitepaper/.

[16] YAKOVENKO A. Solana: a new architecture for a high performance blockchain v0. 8.13[EB/OL]. [2024-06-18]. https://solana.com/solana-whitepaper.pdf.

[17] 陈晶, 杨浩, 何琨, 等. 区块链扩展技术现状与展望[J]. 软件学报, 2024, 35(2): 828-851.
CHEN J, YANG H, HE K, et al. Current situation and prospect of blockchain scaling technology[J]. Journal of Software, 2024, 35(2): 828-851. (in Chinese)

[18] ZHANG M F, LI R J, DUAN S S. Max attestation matters: making honest parties lose their incentives in Ethereum PoS[C]// Proceedings of the 33rd USENIX Security Symposium. Berkeley, USA: USENIX Press, 2024: 6255-6272.

[19] CHU H T, ZHANG P C, DONG H, et al. A survey on smart contract vulnerabilities: data sources, detection and repair[J]. Information and Software Technology, 2023, 159: 107221.

[20] SUN Y C. TRON network[EB/OL]. [2024-06-18]. https://tron.network/.

[21] WANG H, GE C P, ZHOU L, et al. A publicly verifiable outsourcing matrix computation scheme based on smart contracts[J]. IEEE Transactions on Cloud Computing, 2024, 12(1): 70-83.

[22] LIU Y, HE J L, LI X Y, et al. An overview of blockchain

smart contract execution mechanism[J]. Journal of Industrial Information Integration, 2024, 41: 100674.
[23] LI H, MI X H, DOU Y Z, et al. An empirical study of Storj DCS: ecosystem, performance, and security [C]// Proceedings of the 31st IEEE/ACM International Symposium on Quality of Service (IWQoS). Washington D. C., USA: IEEE Press, 2023: 1-10.
[24] TZINAS A, ZINDROS D. The principal-agent problem in liquid staking [EB/OL]. [2024-06-18]. https://eprint.iacr.org/2023/605.
[25] LU P C, CAI L, YIN K T. SourceP: detecting ponzi schemes on Ethereum with source code[C]//Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Washington D. C., USA: IEEE Press, 2024: 4465-4469.
[26] LOMASHUK K, SHAPOVALOV V, RASMUSSEN K. Lido[EB/OL]. [2024-07-08]. https://lido.fi/.
[27] RUGENDYKE D, LANGLEY D, WALLMANN K. Rocket Pool[EB/OL]. [2024-07-08]. https://rocketpool.net/.
[28] MICALI S, RABIN M, VADHAN S. Verifiable random functions[C]//Proceedings of the 40th Annual Symposium on Foundations of Computer Science. Washington D. C., USA: IEEE Press, 2002: 120-130.
[29] POST K. Reed-Solomon erasure coding in Go [EB/OL]. [2024-07-08]. https://github.com/klauspost/reedsolomon.
[30] VOGELSTELLER F, BUTERIN V. ERC-20: token standard[EB/OL]. [2024-07-08]. https://eips.ethereum.org/EIPS/eip-20.
[31] ZHAI M Z, WU Q H, LIU Y Z, et al. Secret multiple leaders & committee election with application to sharding blockchain[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 5060-5074.
[32] ZHANG L, ZHANG B X, LI C. An efficient and reliable Byzantine fault tolerant blockchain consensus protocol for single-hop wireless networks [J]. IEEE Transactions on Wireless Communications, 2024, 23(3): 1974-1987.
[33] 旋逸昭, 赵红武, 金瑜. 一种基于双链的区块链共识机制[J]. 计算机工程, 2024, 50(5): 139-148.
XUAN Y Z, ZHAO H W, JIN Y. A dual-chain-based consensus mechanism for blockchain [J]. Computer Engineering, 2024, 50(5): 139-148. (in Chinese)
[34] NI N, ZHU Y X. Enabling zero knowledge proof by accelerating zk-SNARK kernels on GPU [J]. Journal of Parallel and Distributed Computing, 2023, 173: 20-31.

文字编辑 吴云芳
栏目编辑 赖玉玲