|  |  |
| --- | --- |
| 分 类 号 | 密 级 |
| U D C | 单位代码 10151 |

大 连 海 事 大 学

**硕士学位论文**

**比特币矿工节点的区块链存储分析**

**与优化研究**

**李博**

|  |  |
| --- | --- |
| 学位授予单位 | 大 连 海 事 大 学 |

|  |  |  |  |
| --- | --- | --- | --- |
| 指 导 教 师 | 李志淮 | 职 称 | 教授 |

|  |  |  |  |
| --- | --- | --- | --- |
| 申请学位级别 | 工学硕士 | 学科（专业） | 计算机科学与技术 |

|  |  |  |  |
| --- | --- | --- | --- |
| 论文完成日期 | 2017年12月 | 答辩日期 | 2018年3月 |

|  |  |  |
| --- | --- | --- |
|  | 答辩委员会主席 |  |

**Analysis and Optimization of Block Chain Storage for Bitcoin Miner Node**

**A thesis Submitted to**

**Dalian Maritime University**

**In partial fulfillment of the requirements for the degree of**

**Master of academic**

**by**

**Li Bo**

**(Computer Science and Technology)**

**Thesis Supervisor: Professor Li Zhihuai**

**March 2018**

**大连海事大学学位论文原创性声明和使用授权说明**

**原创性声明**

本人郑重声明：本论文是在导师的指导下,独立进行研究工作所取得的成果，撰写成博/硕士学位论文 “比特币矿工节点的区块链存储分析与优化研究”。除论文中已经注明引用的内容外，对论文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本论文中不包含任何未加明确注明的其他个人或集体已经公开发表或未公开发表的成果。本声明的法律责任由本人承担。

学位论文作者签名：

**学位论文版权使用授权书**

本学位论文作者及指导教师完全了解大连海事大学有关保留、使用研究生学位论文的规定，即：大连海事大学有权保留并向国家有关部门或机构送交学位论文的复印件和电子版，允许论文被查阅和借阅。本人授权大连海事大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，也可采用影印、缩印或扫描等复制手段保存和汇编学位论文。同意将本学位论文收录到《中国优秀博硕士学位论文全文数据库》（中国学术期刊（光盘版）电子杂志社）、《中国学位论文全文数据库》（中国科学技术信息研究所）等数据库中，并以电子出版物形式出版发行和提供信息服务。保密的论文在解密后遵守此规定。

本学位论文属于： 保 密□ 在 年解密后适用本授权书。

不保密□ （请在以上方框内打“√” ）

**论文作者签名： 导师签名：**

**日期： 年 月 日**

摘 要

目前，区块链技术发展迅猛，实际应用涉及多个行业。但是该技术依然存在诸多问题，例如区块链数据在P2P网络节点中的存储问题。以比特币为例，目前比特币系统已经不间断运行八年，产生区块接近五十万块，单个节点占用磁盘空间超过166Gb。根据比特币系统原理，P2P网络中每个节点都需要保存全部的历史区块信息。随着时间的推移区块链在节点中占用的空间会越来越多，节点的运行成本也会越来越大。因此比特币区块链的存储问题不容忽视。

目前针对区块链的存储空间优化有迷你区块链、分段存储等几种方案。本文通过分析现有区块链空间优化方案存在的问题，设计了一种基于星际文件系统IPFS的区块链空间存储优化方案。本方案旨在以经典的比特币分布式架构为基础，利用星际文件系统的文件存储转发功能以及迷你区块链思想设计一种针对矿工节点的区块链空间优化方案。该方案通过定期对历史区块进行总结，剔除对验证交易没有作用的过期交易输出来达到空间优化的目的。在优化过程中被保留下来的未被花费的交易输出UTXO将会以文件的形式完整的保存在IPFS网络中，方便新加入的矿工节点读取。最终达到优化节点区块链存储空间以及提高新加入矿工节点区块同步效率的目的。

论文对提出的优化方案进行了模拟实验。实验结果表明，本优化方案能够定期对历史交易数据进行总结，剔除对验证交易没有作用的过期交易输出，以达到有效减少节点存储空间的目的。

关键词：比特币；区块链；存储优化；节点分类

**ABSTRACT**

At present, blockchain technology is developing rapidly and its application involves many industries. However, there are still many problems, such as the storage problem of blockchain data in P2P network nodes. Take bitcoin for example, the Bitcoin system has been in operation for eight years, generating a total of about 500,000 blocks and single node takes up 166Gb of disk space. According to the principle of bitcoin system, every node in P2P network needs to save all the history information of the block. Over time, the blockchain will occupy more space in the node, and the running cost of the node will also be larger and larger. Therefore, the storage problem of bitcoin blockchain can not be ignored.

Currently, there are several solutions for reducing the storage of blockchain, such as mini-blockchain and segmented storage. In this paper, by analyzing the problems of these schemes, an improved scheme which can reduce the storage space of blockchain is designed. Based on the classic Bitcoin distributed architecture, this paper aims to design a block-space optimization scheme for miner nodes. The file storage and forwarding capabilities of the Inter Planetary File System and the idea of mini-blockchain scheme are all involved. The scheme mainly achieves the purpose of space optimization by summarizing the blocks regularly, excluding the expired transaction output that has no effect on verification. The Unspent Transaction Output will be saved as a file on the IPFS network during optimization. In this way, the miner node can quickly get the file and start mining. The ultimate goal is to optimize the storage space for the node and to increase the synchronization efficiency of the newly added miners.

The paper has carried on the simulation experiment to the proposed optimization program. The results show that this program can regularly summarize historical transaction data, and eliminate the expired transaction output that has no effect on the verification transaction in order to reduce the node storage space effectively.

Key Words：Bitcoin; Blockchain; Storage optimization; Node classification

目 录

[第1章 绪论 1](#_Toc503623846)

[1.1 研究背景 1](#_Toc503623847)

[1.2 国内外研究现状 2](#_Toc503623848)

[1.3 主要研究的内容 3](#_Toc503623849)

[1.4 论文的章节安排 4](#_Toc503623850)

[第2章 相关技术简介 5](#_Toc503623851)

[2.1 区块链技术 5](#_Toc503623852)

[2.1.1 区块结构 5](#_Toc503623853)

[2.1.2 区块链原理 9](#_Toc503623854)

[2.1.3 区块的产生 10](#_Toc503623855)

[2.2 比特币交易原理 11](#_Toc503623856)

[2.2.1 比特币地址 11](#_Toc503623857)

[2.2.2 比特币交易的结构 13](#_Toc503623858)

[2.2.3 输入与输出 14](#_Toc503623859)

[2.2.4 交易的本质 15](#_Toc503623860)

[2.3星际文件传输系统IPFS 16](#_Toc503623861)

[第3章 现有区块链存储空间优化方案 18](#_Toc503623862)

[3.1 区块链的存储容量可扩展模型 18](#_Toc503623863)

[3.2 迷你区块链 19](#_Toc503623864)

[3.2.1 账户树 19](#_Toc503623865)

[3.2.2 迷你区块链 20](#_Toc503623866)

[3.2.3 证明链 21](#_Toc503623867)

[3.3 问题分析 21](#_Toc503623868)

[第4章 比特币矿工节点的区块链存储优化方案 23](#_Toc503623869)

[4.1 矿工节点工作流程分析 23](#_Toc503623870)

[4.1.1 加入网络流程 23](#_Toc503623871)

[4.1.2 参与挖矿流程 23](#_Toc503623872)

[4.2 空间优化方案设计 24](#_Toc503623873)

[4.2.1 整体设计 24](#_Toc503623874)

[4.2.2 总结区块的生成及验证 26](#_Toc503623875)

[4.2.3 UTXO文件存储 31](#_Toc503623876)

[4.2.4 证明链 32](#_Toc503623877)

[4.3 UTXO文件健康情况检查机制 33](#_Toc503623878)

[4.3.1 节点角色设置 33](#_Toc503623879)

[4.3.2 UTXO文件健康检查模块 33](#_Toc503623880)

[第5章 区块空间优化模型实验及结果分析 35](#_Toc503623881)

[5.1 实验环境介绍 35](#_Toc503623882)

[5.1.1 硬件介绍 35](#_Toc503623883)

[5.1.2 软件介绍 36](#_Toc503623884)

[5.2 比特币交易数据统计分析实验 36](#_Toc503623885)

[5.2.1 比特币交易输出统计实验 38](#_Toc503623886)

[5.2.2 已被花费交易输出比例统计实验 39](#_Toc503623887)

[5.3 存储优化方案实验 40](#_Toc503623888)

[5.3.1 容量优化实验 40](#_Toc503623889)

[5.3.2 双花攻击实验 42](#_Toc503623890)

[5.4 实验分析与小结 42](#_Toc503623891)

[第6章 总结与展望 44](#_Toc503623892)

[6.1 总结 44](#_Toc503623893)

[6.2 展望 45](#_Toc503623894)

[参 考 文 献 46](#_Toc503623895)

[攻读学位期间公开发表论文 50](#_Toc503623896)

[致 谢 51](#_Toc503623897)

第1章 绪论

1.1 研究背景

比特币最早出现于2009年[1]，在此后的很长一段时间里人们只知道比特币，而对维系比特币安全稳定运行至关重要的区块链技术[2]了解甚少。公认的最早关于区块链的文献记载是中本聪所撰写的《比特币：一种点对点的电子现金系统》，但该文献重点在于讨论比特币的整个交易系统，实际上并没有明确提出区块链的定义和概念。在文中区块链的概念被描述为用于记录比特币交易的账目历史[3]。时间的车轮转动到2015年，区块链技术逐渐走进人们的视野，尤其是引起科技以及金融领域专家学者的注意。

区块链技术涉及密码学[4]、分布式共识、P2P网络[5]等多个领域，可以在去中心化、去信任的前提下实现点对点的价值传递。作为多个领域融合的产物，运行在虚拟网络中的区块链可以有效的降低真实世界中的信任成本，重新为世人诠释价值的传输方式。

区块链行业逐渐火热主要源于一种可能，一种为真实世界各个行业带来创新的可能[6]。例如在防伪溯源领域，世界性连锁企业沃尔玛试图利用区块链保障我国市场猪肉供应链的安全；在医疗领域，多家医疗机构建立互联的、不可篡改的电子病历和检验报告体系，用于保存证据，避免医疗纠纷。

著名的管理咨询公司麦肯锡在2016年发布报告，认为区块链技术将在未来五年内颠覆众多行业，尤其以银行业和保险业为主。全球最大的管理咨询和信息技术跨国企业埃森哲预测，区块链技术发展到2025年可以为全球八大投资银行每年节省80至120亿美元的基础设施成本。世界金融巨头如IBM、摩根大通、花旗银行等也纷纷宣布涉及区块链领域。自2014年以来，以ETH以太坊、QTUM量子链、EOS等为首的区块链平台不断涌现，这意味着越来越多的专家学者在区块链技术改进优化的道路上不断前行。

基于以上内容可以推测，区块链将为世界带来巨大的变革。世界四大会计师事务所之一的德勤公司有这样的预言：“区块链是一场改变信任的革命，将重塑金融行业。”作为一项伟大的技术，区块链将影响的不仅仅是金融行业，可能是相关的各行各业[7]。

1.2 国内外研究现状

目前区块链技术已经被应用到多个领域。在国内，基于其数字货币的本身属性，金融领域首先引入区块链技术，如文献[8]所述。文献[9]尝试将区块链技术应用到数据保护方面。文献[10]构建了一个医疗数据安全存储模型，用于对医疗数据进行存储和分享。文献[11]对区块链在保险行业的应用进行了探讨与研究。在国外，引入区块链技术的领域同样很多。文献[12]尝试将区块链不可篡改的特性应用到产品溯源方面。文献[13]尝试在教育评价体系中引入区块链技术。文献[14]的主要内容依然与金融领域相关。

除了将区块链技术应用到各个行业外，围绕区块链底层技术的研究也有很多。专家学者们的研究方向主要集中在系统整体架构的优化和节点间达成共识的效率上[15]。相关的例子有很多。

引入智能合约概念的以太坊Ethereum[16]。以太坊同样是一个分布式价值传输系统。与比特币不同的是，以太坊允许在区块链上运行智能合约，让开发人员建立并发布分布式应用（Decentralized Application）[17]。智能合约有自己的编程语言，可以处理复杂的业务逻辑。因此自2014年诞生以来，全球已有数百个以太坊应用诞生。

提出PoS（Proof of Stake）股权证明机制的点点币PeerCoin[18]。与基于PoW（Proof of Stake）工作量证明机制的比特币不同，点点币使用PoS股权证明机制。这样做的好处是降低成为矿工的成本，专业的硬件设备挖矿不再被需要，大量的电费成本被节约下来。

国内著名的区块链项目量子链Qtum也使用PoS股权证明机制[19]。它同时继承了比特币UTXO（Unspent Transaction Output）模型和以太坊智能合约的概念，并且对移动端有较好的兼容。量子链融合了各家的长处，目前发展十分迅速。

总体而言，虽然众多项目各具特色，但大部分的改进依然遵循比特币的区块链存储模式。目前比特币已经经历八年的风雨，随着区块链技术的火爆发展，比特币的用户数和交易量呈爆发式增长。快速增长的用户量意味着更加频繁的交易，同时也意味着区块链需要记载更多的交易信息。2014年7月单个节点中区块链的占用空间仅为19G。到2017年11月，比特币区块链空间占用已经超过166G[20]，区块链占用空间的增长速度可见一斑。目前比特币区块链的占用空间依旧可控，但这正在成为一个令人担忧的问题。

东北大学的贾大宇等人提出区块链的存储容量可扩展模型[21]。方案的核心思路是将区块的链式结构分成多段，分配给多个节点分别存储，从而达到缓解区块空间占用较大的问题。但是作为矿工节点，保存区块信息并不是其全部工作。在接收P2P网络中未被确认的交易时，矿工节点需要查询历史区块来验证交易的正确性。如果需要查询的区块不在本节点中保存，节点需要向其他节点发起网络请求。P2P网络中将会出现大量的区块请求广播，影响整体的运行效率。

另一个有前途的尝试是迷你区块链项目[22]，其目的是通过使用余额树的结构存储链上的账户信息，删减区块链上过期的交易信息，从而减少空间占用。由于节点只需要区块链中最新的部分完成与网络的同步，所以作者将链的这一部分称为“迷你块链”。这种修整过程引起的安全性损失可以通过一个小的“证明链”来解决，资产所有权数据的丢失可以用一个数据库来解决，数据库保存所有非空地址的余额，被称为“账户树”。证明链保证了迷你区块链的安全，迷你区块链保证了账户树的安全。这三种机制共同组建了一个高度完整和安全的系统，而且该系统比所有其他纯P2P类货币更为小巧。但是迷你区块链的改进并不针对比特币，而是重新开发一个独立区块链。新增加的账户树结构也增加了系统的复杂度。

1.3 主要研究的内容

针对现有区块链空间优化方案存在的各种问题，本论文旨在以经典的比特币分布式架构为基础，利用星际文件系统的文件存储转发功能和迷你区块链思想，设计一种针对矿工节点的区块链空间优化方案，并对其中的关键技术进行了研究和实现。

本文尝试通过定期对历史区块进行总结，剔除对验证交易没有作用的过期交易输出来达到空间优化的目的。在优化过程中被保留下来的未被花费的交易输出UTXO（Unspent Transaction Output）将会被完整的保存在IPFS网络中，方便新加入的矿工节点读取。最终达到减少区块链空间占用以及提高新加入矿工节点区块同步效率的目的。

1.4 论文的章节安排

针对上述提出的研究内容，本论文对文章的章节安排如下：

第1章 绪论 本章主要说明题目的背景以及选题的研究意义，阐述课题在国内外的研究现状。简要介绍本文的主要研究内容以及后续的章节安排。

第2章 相关技术简介 本章主要介绍本文涉及到的相关技术，包括区块链结构、比特币具体的交易细节以及星际文件系统IPFS的原理。

第3章 现有区块链存储空间优化方案的问题分析 本章主要阐述几种区块链空间优化方案的优缺点，分析现存的主要问题，提出改进方案。

第4章 针对矿工节点区块空间优化方案的关键技术设计 本章主要描述本文提出的针对矿工节点的区块存储空间优化的整体设计。主要内容包括：矿工节点的特点分析；系统整体结构；改进后的区块结构；总结后的UTXO在IPFS网络中的存储流程；

第5章 针对矿工节点区块空间优化方案的实现与测试 本章主要介绍本文设计的区块链空间优化方案实现与测试。在搭建的P2P网络中分别运行改进前后的比特币代码，验证本文设计方案对区块空间优化效果。同时测试节点能否正确完成交易验证及打包过程，观察改进方法对系统本身交易功能的影响。

最后一章对本文的主要工作做出总结，指出本文方案中尚且存在的不足之处，对未来的工作进行展望。

第2章 相关技术简介

2.1 区块链技术

区块链技术的主要目的是保证P2P网络中大部分节点保存账本的一致性。在Wikipedia网站中区块链的定义如下：区块链是用分布式数据库识别、传播和记载信息的智能化对等网络, 也称为价值互联网[24] [25]。区块链技术基于去中心化的对等网络，用开源软件把密码学原理、时序数据和共识机制相结合，来保障分布式数据库中各节点的连贯和持续，使信息能即时验证、可追溯、但难以篡改和无法屏蔽，从而创造了一套隐私、高效、安全的共享价值体系[26]。

与传统的客户机/服务器（Client/Server）或浏览器/服务器（Browser/Server）中心化的架构不同，区块链技术采用去中心化的分布式架构[27]。每个节点保存全部的交易信息。两者的体系结构如图2.1所示。



图2.1 中心化记账方式和去中心化记账方式对比

Fig. 2.1 Centralized accounting Vs Decentralized accounting

2.1.1 区块结构

在比特币网络中，交易信息会被定期打包，以文件的形式永久保存。这些被打包的文件就称为区块。一个区块中包含的主要内容是从当前区块产生时到上一区块产生的时间内整个比特币网络中产生的合法交易信息。一个区块的具体结构[28]如表2.1所示。

表2.1 区块结构

Tab. 2.1 Block structure

|  |  |  |
| --- | --- | --- |
| 数据项 | 描述 | 长度 |
| Magic No（魔法数） | 始终是0xD9B4BEF9 | 4字节 |
| Blocksize(区块大小) | 到区块结束的字节长度 | 4字节 |
| Blockheader（区块头） | 组成区块头部的字段 | 80字节 |
| Transaction counter（交易计数器） | 区块内包括的交易数量 | 1到9字节 |
| Transactions交易信息 | 记录在区块内的交易列表 | 可变大小 |

区块链中的区块结构与计算机网络中的IP数据包格式有些类似。IP数据包由报文头部和报文体两部分组成，区块也包含有区块头和区块体这两部分。报文体中包含该数据包需要传输的内容，区块体中记载的是最近一段时间比特币网络中的合法交易。报文头部包含数据包整体的情况描述，区块头的意义与之类似，是对本区块包含内容的总结。区块头具体信息如表2.2所示。

表2.2 区块头部结构

Tab. 2.2 Block header structure

|  |  |  |
| --- | --- | --- |
| 数据项 | 描述 | 大小 |
| Version（版本号） | 标识当前系统版本 | 4字节 |
| hashPrevBlock(前一区块的哈希值) | 引用上一个区块的哈希值 | 32字节 |
| hashMerkleRoot（梅克尔树根节点哈希值） | 该区块中交易的merkle树根的哈希值 | 32字节 |
| Time（时间戳） | 该区块生成的时间 | 4字节 |
| Bits（难度目标） | 当前挖矿的难度目标 | 4字节 |
| Nonce（随机数） | 0到32位的随机数 | 4字节 |

当软件更新时版本号字段将会改变，用于告知不同节点该区块遵循哪种版本规则[29]。hashPrevBlock字段十分重要，其意义在于标识前一个区块的哈希值。作为各个区块的唯一标识，通过哈希值可以定位到与之对应的区块上。每个区块都包含一个叫做hashPrevBlock的字段。用于指向其前一区块，这样就形成了一条以区块为节点的链式结构，区块链的名字就由此而来。这种结构与数据结构中链表的概念很类似，区块对应链表中的节点，而hashPrevBlock字段则类似链接链表节点的指针。

另一个比较重要的数据项是Nonce随机数字段。在比特币这样全网公开的P2P网络中，节点的加入和退出基本是没有成本的，不需要通知任何中心化的组织机构。这也是比特币的魅力所在。但是在没有统一安排的情况下，如此众多的节点向区块链中加入新区块必定是会出现混乱的。如何控制在一段时间内仅有少数几个节点拥有向区块链中写入区块的权利呢？比特币的解决方案是在每一个区块加入区块链后，给所有节点出一道难题。题目的难度非常大，在一段时间内只有少数拥有强大计算能力的节点才能计算出来。正确解决难题的节点拥有将区块加入区块链的权利。整个求解问题的过程叫做“挖矿”，尝试解决问题的这些节点我们称为矿工。

与挖矿相关联的是被称为难度目标的Bits字段，它是衡量矿工给出的解题答案是否正确的标准[30]。具体来说，只有小于难度目标的答案才是有效的。这样的设置方式是为了保证整个比特币网络平均每10分钟产生一个区块。每经过2016个区块，该数据项会进行一次调整，保证平均出块时间间隔为10分钟，既不会太快也不会太慢。很显然，这种以计算机性能瓶颈为约束的难度求解方案是非常有效的，但同时也是对资源的一种浪费。因为这些难题的设置本身并不解决任何实际问题，其本质只是一种选拔制度，但是浪费的电力以及计算能力是十分庞大而不必要的。

在每个区块中都包含一个HashMerkleRoot Merkle字段，表示通过该区块中所有交易信息生成的梅克尔树的根节点的值。梅克尔树[31]是一种哈希二叉树，本质上是该区块包含的交易集合的数字指纹。哈希函数是一种数据处理函数，函数输入可以为任意长度的数据，输出则为固定长度。这种特性可以应用与数据的完整性校验。在发布任意信息时，可以对整个数据进行哈希运算得到固定长度的哈希值，然后将得到的哈希值公布到网络中。当其他用户下载到数据时，可以对该数据再次进行哈希运算。如果计算结果与发布者公布的结果一致，证明用户下载到的数据与发布者发布的一致。否则说明下载者得到的数据是经过篡改的，因为对于哈希函数而言，输入数据只要有一点细微的改动都会导致输出结果有很大改变。

梅克尔树是由下至上构建的，构建的方法与哈希函数有很大关系。首先对区块包含的交易进行哈希运算，计算结果将作为梅克尔树的叶子节点。由于本质上梅克尔树是一棵二叉树，因此将相邻两个叶子节点的值拼接后再次进行哈希运算，最后将得到结果写入到两叶子节点的父节点中。其余的节点采用相同的计算方式，这样从下至上逐层构造。最终得到只有一个根节点的二叉树，根节点的值会被写入到HashMerkleRoot Merkle字段中。其结构示意图如图2.2所示。



图2.2 梅克尔树结构示意图

Fig. 2.2 Merkel tree structure diagram

可以看出，通过上述构造方法构建的梅克尔树的叶子节点必须是偶数，因此当区块链包含的交易数为单数时，算法将复制一份最后的交易进行补充。

梅克尔树除了作为整个交易集合的数字指纹外，还有另外一点好处。那就是快速校验某一条交易信息是否存在与特定区块中。由于一个区块中可以存在几千条交易记录，获取大量数据重新构建梅克尔树需要很大的工作量，因此快速校验的特性显得十分重要。以图2.2为例，为了证明HA对应的A交易是否存在，只需要知道HB、HCD、HEFGH、HIJKLMNOP四个哈希值即可。通过相互运算，如果最终计算结果与梅克尔树的根值相同，即可证明该区块包含交易A。

2.1.2 区块链原理

区块链是一条类似数据结构中链表的结构。链表中的节点对应区块链中的区块，指针对应区块哈希。区块链的具体结构如图2.3所示。



图2.3 区块链结构示例

Fig. 2.3 Example of Blockchain structure

每一个区块保存其前一区块的哈希值，如此环环相扣可以一直追溯到创世区块[32]。创世区块就是整个区块链的头节点，显然该区块是没有前一区块的。比特币的创世区块中包含了2009年1月3日泰晤士报头条新闻的标题，作为其在该日期之后出现的证据[33]。由于区块之间拥有紧密的联系，篡改中间的任何一点交易数据都会导致该区块哈希值的改变。这样将会导致其后续的一系列区块出现错误。如果想通过一并修改后续的所有区块基本是不可能的，因为以一个或几个节点的计算能力是无法与整个比特币网络相比的。经过篡改的区块链无法超越拥有更多算力支持的主链，也就没有办法说服其余节点接受篡改的部分。因此篡改的目的也就无法达成了。这就是区块链无法被篡改的特性。

虽然从任意区块向前遍历，都可以到达创世块。但是从创世块出发，却有可能出现分叉。可以想象在一个出块周期内，整个区块链网络中的两个或多个节点在短时间内相继算出答案的情况是存在的。假设同时存在两个正确答案在网络中传播，接受消息的矿工节点默认会将接受到的第一个正确区块加入到自己的区块链中。这样网络就被分成了两部分，一部分支持答案A，一部分支持答案B。这种情况就产生了分叉，不过没有关系。经过一段时间的运算，其中一组的矿工将会转而支持另外一组，因为另外一组所支持的区块链分叉拥有更多的工作量证明。被遗弃的分叉中包含的交易信息将会被重新加入到交易队列池中等待重新被写入到区块链的主链中。被遗弃的分叉所带来的矿工收益不会出现在主链中，因此这部分收益实际上是无效的。这就是比特币网络设置经过100个区块确认才算真正意义上写入区块链的原因。

2.1.3 区块的产生

在比特币网络中，有转账需求的节点会将验证正确的交易信息发布到P2P网络中[34]。其他矿工节点将会利用保存在本地的区块历史验证这些信息是否合法，合法的交易将会被整理到区块体中。矿工节点将会在区块头中加入一个随机调整数以及前一个区块的哈希值。所谓挖矿的过程实际上是矿工不断调整随机数，不断计算本区块哈希值，直到其哈希值符合要求的尝试过程。由于哈希运算是不可逆的，通过结果反向推导输入参数几乎是不可能的。矿工想要得到符合要求的运算结果，只有不断试错，通过不断调整输入参数碰运气。

找到符合要求的结果后，节点将会向全网广播该区块。其他节点在收到该区块后，将会验证区块中包含交易的正确性，以及区块整体的哈希值是否小于挖矿难度。如果验证通过，这些节点就会将该区块加入到本地的区块链当中，并加入到计算下一个区块合法结果的竞赛中。

如2.1.1节中区块头部结构表所示，区块中大部分数据项对所有用户相同，但时间戳可能有所不同。时间戳是否合法的标准为其取值应大于前11个区块的平均出块时间，同时小于网络调整时间向后推迟两个小时的时间。在P2P网络中，一个节点会与几个节点相互通讯，网络调整时间的取值为相连节点的平均时间。网络调整时间永远不会超过本地系统时间70分钟以上。

Nonce随机数通常也不会相同。默认情况下，它是以0开始的。矿工每进行一次区块的哈希运算，Nonce就会增加。因此每个节点计算出正确结果的试错次数很大概率是不同的。

即使两个矿工节点将要打包的交易完全相同，他们分别计算出的梅克尔树的根值也会不同[35]。因为默认情况下区块链中包含的第一笔交易是没有输入的，系统默认将一笔凭空出现的奖励发放给产生区块的矿工。这部分收益以及比特币被人们认可的价值保障了比特币网络始终保持大量矿工的积极参与。由于每个矿工的比特币地址是不同的，因此每个节点计算的区块中第一笔交易的信息也就不尽相同。因此，通过这些交易信息逐层计算得出梅克尔树的根值显然也是不同的。

总而言之，比特币网络区块的产生过程就是所有矿工在本地不断进行哈希运算，竞争谁可以最快算出符合要求的答案的过程。显然拥有更加强大的计算能力，单位时间内可以进行更多次试错尝试的矿工节点获得出块权的概率更大。

2.2 比特币交易原理

与现实中拥有唯一编号以及防伪标识的钞票不同，在只有0和1组成的数字世界中，文本或文件可以做到没有成本的复制，这样很容易带来双花问题。双花问题是指同一笔数字货币通过某些手段被多次花费。那么以比特币为代表的数字货币是如何解决双花攻击等安全问题的呢？首先从基于密码学的比特币地址说起。

2.2.1 比特币地址

比特币地址可以理解为拥有比特币资产的用户在网络中对外公开的一种标识。假设用户A想要给用户B转一笔钱，用户B只需要向用户A提供他的地址即可，不需要任何其他信息。比特币地址是由私钥经过多次单向的哈希计算而来的。

私钥[36]的本质是通往比特币世界大门的钥匙，它决定了对应比特币地址中比特币的归属。从结构上来说，私钥是32个byte组成的数组。1个byte等于8位二进制，而一个二进制只有0或者1两种可能，所以私钥的总数就是2^(8\*32)=2^256个。这个数是非常庞大的，甚至超过了宇宙中原子的总数，因此如果想要将所有存在的私钥遍历一遍几乎是不可能的。

获得私钥需要有一个随机性非常大的数据源。常用的方法包括用户鼠标的晃动、时间因素、终端设备的一些特征码。总之获得的结果越不确定，私钥被破解或产生碰撞的概率就越小。从本质上来说，获得一个私钥与在0到2^256之间选择一个数字是一样的。

通过椭圆曲线算法[37]可以从私钥得到公钥。椭圆曲线算法是一种不可逆的函数，通过输入私钥可以很快过的公钥结果。但即使同时拥有公钥和椭圆曲线算法，求解出私钥也是几乎不可能的。椭圆曲线算法的原理比较复杂，从本质上来说公钥就是私钥映射到平面中椭圆曲线上某个点的坐标。

比特币地址是由公钥经过单向加密哈希算法算出的。由公钥生成比特币地址时使用的算法是Secure Hash Algorithm (SHA) [38]和the RACE Integ rity Primitives Evaluation Message Digest (RIPEMD) [39]，具体来说是SHA256和RIPEMD160。如果将公钥设为K，公钥哈希设为H。那么K与H的关系为H= RIPEMD160（SHA256（K）），得到的是一个160比特的数字。用户比较常见的比特币地址是经过Base58Check编码过的。如果将比特币地址设为A，那么地址与公钥的关系为A=Base58Check(H)。Base58编码由字母数字组成，去除了一些容易让人混淆的字符，例如数字0、大写字母O等等。这样做不仅压缩了数据，也避免了一些人工抄写时产生的错误。从公钥到比特币地址的生成过程如图2.4所示。



图2.4 比特币地址的生成过程

Fig. 2.4 From the public key to the bitcoin address

综上所述，私钥与地址总是成对出现的。通过私钥可以生成地址，但通过地址不能逆向推出私钥。地址可以在比特币网络中接收转账。私钥则相当于钥匙，可以解开封存在区块链中的资产，随心所欲的将资产支付到其他用户手中。

2.2.2 比特币交易的结构

比特币交易可以称为比特币分布式系统中最关键的设计[40]。整个系统架构的设计实现都是为了确保比特币交易在生成、传播、验证、写入区块链等环节的安全性。

交易在创建的最初期会被一个或者多个签名加密。这个过程标志着将要被花费的比特币资产的拥有者许可了这次交易，这个行为类似现实生活中支票的所有者对一张支票进行签名。接下来交易将会被节点广播到比特币网络中，每个接收到交易的节点都会对其进行验证，如果验证通过则继续向自己的邻居节点广播此交易，直到整个网络中的大部分节点都接收到。最终再由成功挖矿的节点写入区块链中。经过足够多后续区块的确认，基本可以确认此交易已经被写入比特币主链中，这也就意味着交易成功了。

交易是一种数据结构，它包含输入值和输出值。具体结构如表2.3所示。

表2.3 交易结构

Tab. 2.3 Transaction structure

|  |  |  |
| --- | --- | --- |
| 数据项 | 描述 | 大小 |
| 版本 | 明确这笔交易参照的规则版本 | 4字节 |
| 输入数量 | 被包含的输入的数量 | 1-9字节 |
| 输入 | 一个或多个交易输入 | 不定 |
| 输出数量 | 被包含的输出的数量 | 1-9字节 |
| 输出 | 一个或多个交易输出 | 不定 |
| 时钟时间 | 一个UNIX时间戳或区块号 | 4字节 |

交易的输入可以理解为被封存在区块链账本历史中的一部分比特币资产。输出意味着需要将这笔资产转到谁的名下，即转给哪个地址。当然输入和输出的数量是不固定的，都可以为多个。

2.2.3 输入与输出

比特币交易的基本单位是未被花费的交易输出，英文简称为UTXO。当某个账户收到一笔比特币资产时，其实际上是以UTXO的形式存储在区块链上的。账户拥有的只是将这笔UTXO中的资产转移给一个或多个账户的权利。因此实际上在比特币系统中并不存在保存账户余额的地方，只有分散在区块当中被封存的UTXO。所谓的账户余额实际上是比特币钱包通过遍历区块链，将所有属于该用户的UTXO相加求和得出的。

从本质上来说，一次交易的过程就是一次消耗已有UTXO转而产生新的UTXO的过程。交易的输出分为两部分，一部分是输出的金额，另一部分是锁定脚本。输出金额可以是“聪”的任意倍数。“聪”是比特币可以分割的最小单位，就像人民币可以被分割成两位小数的“分”一样，比特币可以被分割成由八位小数表示的“聪”。锁定脚本是可以在比特币网络节点中运行的简易程序。它本质上是为想要花费该UTXO中资产的用户设置了一道题目，没有权利处理这笔资产的用户想要解答起来会非常困难，但是有权处理的用户却非常简单。而是否有权处理这笔资产的标准就是是否拥有锁定脚本中用户地址对应的私钥。

与交易输出相对应的是交易输入。交易输入可以通过交易的哈希值以及记录在区块链中的序列号定位对应的UTXO。交易的输入中也包含一个脚本，叫做解锁脚本。解锁脚本被用来解答上述交易输出中设置的那个难题。其本质是一种签名，用于证明锁定脚本对应的UTXO中比特币资产的所有权问题。

由于在比特币交易中未被花费的交易输出是作为一个整体被花费的，这就出现两种情况。举例来说，假设用户A想要给用户B转1比特币。这里基本可以分为两种情况。第一种情况，用户A拥有一个包含1.5比特币的UTXO。那么交易就将1.5比特币的UTXO做为输入。交易输出分为两个，一个是支付给用户B的1比特币，另一个是还给用户A的接近0.5比特币。送还给用户A的输出被称为找零输出。第二种情况，用户A只有0.7比特币和0.8比特币两个UTXO，那么两个UTXO都会作为交易的输入。交易的输出则为支付给用户B的1个比特币和给自己的找零接近0.5比特币。

可以看到整个交易输入总量都是1.5比特币，而输出总量为接近1.5比特币。出现这种情况的原因是交易是需要交易费的，交易输入与输出的差值就作为交易费支付给矿工了。所以矿工的收益除了挖矿奖励以外，还包括产生区块中包含的所有交易的交易费。

2.2.4 交易的本质

比特币交易的验证主要依靠两个部分，一个是锁定脚本，另一个是解锁脚本。锁定脚本用于封存UTXO中的比特币资产，保证只有提供正确的解锁脚本的用户拥有处理资产的权利。相应的解锁脚本就是满足锁定脚本设定的花费条件的脚本。锁定脚本和解锁脚本的结构如图2.5所示。



图2.5 比特币脚本结构

Fig. 2.5 The structure of a bitcoin script

解锁脚本主要分Sig和PubKey两部分组成。PubKey表示用户完整的公钥信息，用于锁定脚本验证是否与保存在其中的公钥哈希相匹配。Sig是由用户私钥信息经过ECDSA 加密算法计算而来，用于在不暴露私钥明文的条件下验证解锁用户是否拥有生成PubKey的私钥。

锁定脚本主要由一些操作命令和公钥哈希组成。操作命令用于告知比特币客户端如何操作位于栈中的数据。公钥哈希为该锁定脚本希望支付账户的公钥哈希。解锁脚本与锁定脚本拼接后就形成了验证交易是否合法的完整程序。比特币的交易脚本分为很多种，其中最为常见的是针对公钥哈希的支付脚本，其具体运行过程如图2.6所示。



2.6 比特币脚本运行过程

Fig. 2.6 The running process of the bitcoin script

首先Sig与PubKey作为数据不做处理，按照顺序压入栈中。接下来压入的是OP\_DUP操作符，它的作用是将栈中下一层数据复制一份，替换掉自身的内容。OP\_HASH160操作符的作用是对下一层的完整公钥进行哈希运算，将运算结果公钥哈希压入栈中。接下来将锁定脚本中的公钥哈希压入栈中。OP\_EQUALVERIFY的功能分为判断是否相同和校验两部分。目的是检查由解锁脚本提供的完整公钥经过计算得到的公钥哈希与锁定脚本中保存的公钥哈希是否相同。如果相同继续运行，反之则直接停止整个脚本。最后一步OP\_CHECKISG的作用是验证由私钥经过加密的Sig与完整公钥是否有对应关系。如果有则返回true，证明整个脚本运行通过，解锁脚本的提供者拥有这笔资产的所有权。

2.3星际文件传输系统IPFS

星际文件系统IPFS（Interplanetary File System）是一个基于P2P网络的分布式文件系统，它的设计目的是希望利用相同的文件系统连接所有计算设备。从某种意义上来说IPFS与互联网有些类似，但它更应该看作像Git一样有版本管理功能的BitTorrent集群。IPFS提供了一个高吞吐量的基于内容寻址的区块存储模型。这形成了一个广义的梅克尔有向无环图Merkle DAG。梅克尔有向无环图是一种数据结构，在这个数据结构上可以建立有版本控制功能的文件系统、区块链、甚至永远不会停机Web网站。IPFS技术结合了分布式哈希表、有激励机制的块交换和自我认证的命名空间等技术，可以达到没有单点故障问题，节点之间不需要相互信任的特性。

目前在互联网中使用最频繁的是超文本传输协议HTTP[41]。HTTP协议是一种非常脆弱的、过度依赖骨干网络的协议。HTTP协议下，数据总是由一个或几个存储节点保存，再通过主干网络发送给接收者。在这种情况下，几个节点的异常或者主干网络出现问题都会导致无法服务的现象出现。

IPFS的出现就是为了解决现有中心化的文件存储带来的问题的。在IPFS网络中用户是基于内容寻址的。当文件被添加到IPFS节点后，它就会得到一个新的名字。这个名字实际上是一个[加密哈希](http://en.wikipedia.org/wiki/Cryptographic_hash_function)，它是从文件内容中被计算出来。通过加密保证该哈希始终只表示该文件的内容。哪怕只在文件中修改一个比特的数据，哈希都会完全不同。

在通过哈希值向IPFS分布式网络取文件的时候，它通过使用一个[分布式哈希表](https://en.wikipedia.org/wiki/Distributed_hash_table)数据结构，可以快速地找到拥有数据的节点，从而获取该数据，并使用哈希验证这是否是正确的数据。通常在一个拥有10,000,000个节点的网络中只需要20跳即可定位目标节点。

IPFS是通用的，并且存储限制很少。它服务的文件不限制大小，对于一些大的文件，它会自动将其切割为一些小块，使IPFS节点不仅仅可以像HTTP一样从一台服务器上下载文件，而且可以从数百台服务器上进行同步下载。IPFS网络是一个细粒度的、不可靠的、分布式的、易联合的内容分发网络（Content Delivery Network , CDN）。对于所有数据类型都很有用，包括图像、视频流、分布式数据库、操作系统、blockchains等。

第3章 现有区块链存储空间优化方案

2009年，中本聪首先将比特币公之于众，永远改变了许多人对金融和经济学的看法。在开始的几年中，比特币系统中交易数量较少，区块链长度也较短。因此区块链在网络节点中的存储问题并不明明显。现在是2017年的11月，比特币区块链的大小已经接近166Gb。虽然几百Gb的空间占用对于大多数矿工来说可以接受。但是随着时间的推移，区块链存储问题将会逐渐凸显出来。这正在成为一个令人担忧的严重问题。目前研究人员已经对该问题展开了研究。

3.1 区块链的存储容量可扩展模型

区块链的存储容量可扩展模型是由东北大学的贾大宇等人提出的。作者在设计模型时，提出了一种数据副本分配策略[42] [43]，将一条完整的区块链副本进行分片处理，并将分片数据保存在一定比例的P2P节点中。同时方案中还设置了验证节点，用于定期对节点进行数据的健康检查，方便及时发现问题。

该方案共建立两条新的区块链，分别为P（position）链和POR（Proofs of Reliability）链。其中Position链用于记录各个区块分段在P2P网络中的存储位置。而POR链则用于保存各个负责存储区块分段的节点的可靠性评价。网络中节点的身份包括三种，用户节点拥有原始的数据，存储节点保存着区块链的副本信息，验证节点的作用是衡量存储节点的稳定性。

在存储的过程中系统首先采用数据可检索性证明方法[44]对区块进行加密。生成的密钥将会在本地保存，同时发送给验证节点，写入POR链中。当有存储数据的需求时，用户节点会向验证节点发出请求。验证节点根据POR链中保存的评价信息给出合理的推荐节点。用户节点根据推荐情况选出一定数量的存储节点保存数据，最后将保存数据的分布状态保存在P链中。

在读取的过程中用户节点首先访问本地的P链，根据各区块的位置信息向存储节点发出请求。如果存储节点在线并且保存有请求数据，就会将请求数据返回给用户节点。最终用户节点利用保存在本地的密钥对数据进行解密，恢复出原始数据。

3.2 迷你区块链

迷你区块链是一种全新的提议，它的许多想法与比特币类似，但是也有其不同之处[45]。该方案通过取消彼此链接的交易来消除对完整块链的需求，从而允许所有交易在足够长时间后被丢弃。但是简单的丢弃区块链历史将会导致交易金额、验证脚本等信息的丢失。迷你区块链提出的解决方案是通过分析区块链的几种设计目的，将各个功能分离成各自独立的模块分别进行优化。总体而言，区块链有三个主要功能模块，分别为协调网络处理交易的方法、保护网络的工作量证明机制、管理账户余额以及记录资产的所有权。比特币区块链将这些功能组合在一起，这样做会导致系统不能很好的扩展。同时用户需要存储大量不需要永久存储的数据。 因此分离区块链的功能是改进的关键。

迷你块链方案摒弃了比特币脚本系统以及链接交易的整体思路，转而使用一个更简单的概念来代替它。具体来说，通过将所有非空地址的余额存储在被称为“帐户树”的结构中，该方案可以在不借助任何历史交易信息的情况下计算给定地址的余额。交易中的输入和输出不再指向其他交易，而是指向帐户树中的地址。因此交易并不是以比特币的方式链接在一起，节点可以在足够长的时间后丢弃所有交易，但账户余额这样的关键信息并不丢失。

3.2.1 账户树

迷你区块连方案的改进主要从账户树的概念开始。如果我们需要知道的只是所有非空地址中的余额，那么永久的记录和存储每一条交易就是非必需的。因此区块链管理账户余额以及记录资产的所有权的功能将被账户树取代。账户树可以被看作是一个分布式余额账本。它将包含所有的非空地址以及它们的余额以及一些其他可以限制提币的字段。

当账户余额改变时，节点不需要添加数据。相反只需要更新账户树当中的数字就可以完成记录操作。在现实中不会提供完全有限数量的数据来处理，因为新的非空地址将不断出现，但地址的总量将会接近有限。从某种意义上说总量是有限的，因为数字货币的分割是有限的，并且世界人口或互联网用户数量也不会永远持续增长。无论如何账户树是可扩展和可管理的。即使人口达到100亿，每个人都有十个不同的非空地址，账户树也只需要跟踪1000亿个地址。由于账户树会定期删除空的地址，并且交易只需要让节点调整数据，而不是添加新数据。所以帐户树的大小可以始终保持相当小。

保存账户信息的结构被命名为账户树是因为它应该有一个哈希树的结构。每一个哈希树中的“账户”都有一个相应的哈希值，并且作为树的叶子结点存在。作为一个哈希树，我们可以组合每个帐户的哈希构建一个散列的金字塔，并计算顶部的“主哈希”。但是这里的账户并不像比特币账户那样由一些地址组成。在迷你区块链中，每个帐户仅指一个地址或叶节点。

3.2.2 迷你区块链

迷你区块链提供了上文介绍的区块链的第一条功能，协调网络处理交易的方法。本质上来说，迷你区块链就是一条普通的区块链，不同的是迷你区块链方案不需要保存历史区块的副本。

比特币需要完整区块链的原因是获取所有地址余额的唯一方式就是查询区块链。但是迷你区块链方案使用账户树来完成管理账户余额和记录所有权的任务。因此保留完整区块链是非必需的，参与节点可以选择丢掉旧的区块来节省大量的磁盘空间。然而几百或几千个最新区块还是需要保留的，这些区块构成了迷你区块链。迷你区块链为整个系统提供了安全的保证。每个块具有嵌入在头部的根哈希，并且我们可以从头开始验证每个区块，确保每个块中的交易总是符合上一个块中的根哈希。由于迷你区块链使用POW工作量证明机制[46]，伪造一条迷你区块链对攻击者来说十分困难。

因为有完整的区块链，在比特币中我们可以从创世区块一路验证到最新区块来证明该区块链的正确性。但是在迷你区块链中如果攻击者从可用的最旧的块中创建一条新的区块链，新加入的节点将会无法将它与真正的区块链分开。因为在最旧的块之前的历史被删除了。攻击者可以花费尽可能多的时间来增加他们的迷你块链的累积难度。此后，攻击者可以广播伪造区块链，如果传播足够大，伪造区块链将有成为主链的风险。这时就需要证明链的帮助，通过提供一种可以存储长期工作证明历史的容器来解决这个问题，方便计算任何链条的总累积难度。因此迷你区块链并非完全删除旧的块，而是保留区块头。这样可以提供随时追踪任何给定链的历史，并比较每个链的总累积难度的可能。

3.2.3 证明链

证明链完成的是区块链的第二个功能，即保护网络的工作量证明机制。其本质只是一串包含区块头的链。所以迷你区块链本质上只保留最近的交易信息。这意味着所有节点仍然可以使用包含区块头的链来验证具有最高累积难度的最佳迷你块链。并且由于帐户树的存在，它们不需要旧的交易来计算地址余额。

迷你区块链本质上与普通区块链没有区别，所以挖矿的方式与比特币相同。节点需要不断计算区块头，直到找到低于目标数的结果。同时节点可以丢弃旧的交易信息，只存储区块头。因为证明解决方案不依赖于区块中的交易。与Bitcoin的安全程度相同，证据链中的每个证明环环相扣，使得几乎不可能生成假证明链。

证据链可以证明哪个迷你块链具有最长期的算力支持。这样攻击者就无法花费算力生产虚假区块链了，因为证明链必须可以接入迷你块状链。所以现在如果一个攻击者试图创建一个假链，他们也需要一个足够长的证明链来接入。这从本质上使迷你区块链方案回到了典型的完整块链方案提供的安全级别。

3.3 问题分析

区块链存储空间可扩展模型虽然能够减少网络中节点的区块空间占用，但是在效率上的付出也是很大的。因为整个比特币网络是非常忙碌的，矿工节点随时都在扮演着接收交易和验证交易的角色。由于区块链本身的设计结构，验证一笔交易的合法性需要查询区块历史。查询的内容主要包括两方面。首先需要查询该笔交易的输入对应的交易输出是否保存在区块链中。如果交易输出不存在，说明被验证的交易不合法。其次需要查询被验证交易花费的交易输出是否为未被花费状态。如果交易输出已被花费，同样说明被验证交易不合法。在分段存储区块链的情况下，必定会出现需要验证的交易历史不在本地保存的情况。由于每个矿工节点都会对接受到的交易进行验证，因此这将会导致网络中突然出现大量的集中针对区块链某一段的请求。显然在P2P网络实际运行中，这样做的效率是很低的。

迷你区块链方案能够在减少区块链空间占用的同时兼顾交易验证的效率问题。因此迷你区块链的优化思想更加可取。但是迷你区块链属于一条独立的区块链，与比特币有许多区别。在迷你区块链方案中，用户的状态信息保存在账户树中。而在比特币系统中，用户的状态信息并不是汇总到一处，而是分散在整个区块链的交易记录中。因此迷你区块链的优化思想值得借鉴，但并不直接适用于比特币的UTXO模型。

第4章 比特币矿工节点的区块链存储优化方案

通过分析第3章介绍的两种已有的区块链空间优化方案可知，迷你区块链的解决方案在处理矿工节点空间优化问题方面更加值得借鉴。但是迷你区块链属于独立的公有区块链，并不适用于比特币的UTXO模型。本文希望以经典的比特币分布式架构为基础，利用星际文件系统的文件存储转发能力和迷你区块链思想设计一种针对矿工节点的区块链空间优化方案。

矿工节点在比特币网络中充当的角色十分重要[47]。可以说比特币区块能够稳定产生、区块链中包含的数字资产能够安全的存储都与矿工节点息息相关。本文描述的区块空间优化方案主要针对的就是矿工节点角色。在比特币网络中，矿工节点主要任务是验证交易和产生区块，对已被花费的交易信息使用较少。因此这部分交易为节省区块链空间提供了可能。

4.1 矿工节点工作流程分析

4.1.1 加入网络流程

当新节点加入比特币网络后，首先需要完成的就是区块链的同步工作。节点首先与P2P网络中的邻居节点通讯，从邻居节点当中获取完整的区块链信息。由于目前区块链的体积已经达到166Gb，因此完整的同步过程是十分漫长的，通常需要几天的时间。但是对于矿工节点而言，其主要工作只是收集并验证交易以及不断寻求符合标准的区块头哈希。这也就意味着区块链中包含的许多已经被花费的历史交易在被节点下载到本地后并不会被使用。因此花费大量的时间下载完整的区块信息并不是一件必要的事。

4.1.2 参与挖矿流程

完成区块同步后，矿工节点就拥有了参与挖矿的资格。矿工节点首先从P2P网络中收集交易信息，根据交易中引用的未被花费的交易输出查找本地区块链中的数据，判断该笔交易是否合法。合法的交易数据将会被收集到矿工节点的交易池中，等待节点写入到最新区块中。因此每一笔合法的交易对应上一笔交易的一个可用的输出。一笔交易被确认并写入到区块链中的同时也意味着其引用的上一笔交易的输出变为不可用的状态，因为上一笔交易的输出已经被花费了。这其实形成了一个交易的链条，具体的结构如图4.1所示。



图4.1 比特币交易示例

Fig. 4.1 Example of Bitcoin transaction structure

图4.1中1号交易的输入是0号交易的第一个输出，2号交易的输入引用的是0号交易的第二个输出。去除第一笔给矿工的挖矿奖励外，区块链中每一笔交易都是如此相互引用而产生的。因此在区块链中类似0号、1号、2号、4号、5号这样所有的交易输出都已经被引用过的交易信息不在少数。从本质上来说，矿工节点在验证交易，计算区块哈希头的过程中是不需要这些信息的。因此这就为本文中区块链的存储空间优化提供了可能。

4.2 空间优化方案设计

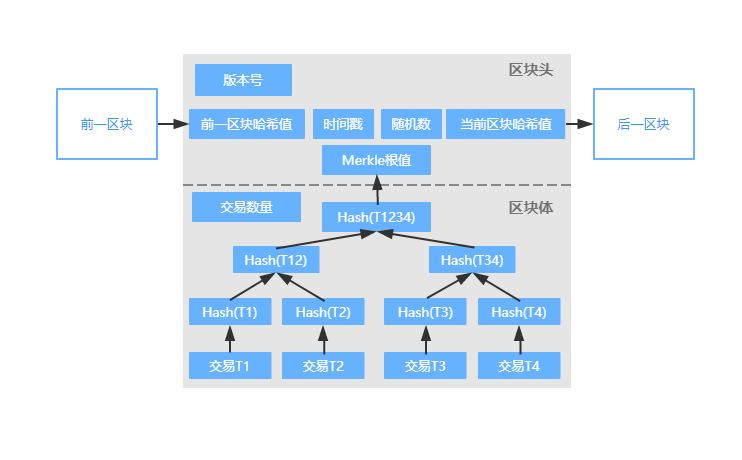
4.2.1 整体设计

通过分析矿工节点在网络中的工作流程可知，对于矿工节点的角色而言，区块链的存储是有一定优化空间的。在区块链中与未被花费的交易输出UTXO相对应的，那些已经被花费的交易输出就是区块链存储空间优化的突破口。因此本文希望以比特币分布式架构为基础，设计一种可以定期对历史区块进行总结，提取并整理其中包含的未被花费的交易输出，清理对验证无用的交易信息的机制。方案整体结构如图4.2所示。



图4.2 方案整体结构示意图

Fig. 4.2 The overall structure of the program



IPFS网络

存入

返回Hash值

全节点

定期检查

由图4.2可知，本文方案除了设置有普通区块外，还增加了特殊的总结区块。总结区块之间有一定跨度，即每隔n个普通区块出现一个总结区块。这种定期出现的总结区块承担了定期清理区块信息的任务。

总结区块的生成过程与普通区块有所不同，除了按照正常流程验证交易外，还包含整理当前可用的交易输出的功能。矿工节点将整理好的交易输出写成文件， 存储到IPFS网络中。存储成功后，将用于获取文件的哈希值写入到总结区块的区块头中。之后挖矿与广播总结区块的工作流程与普通区块一致。

当其他矿工节点接收到总结区块后，需要从IPFS网络中下载并验证UTXO文件的正确性。发现错误即丢弃区块，继续挖矿。如果验证通过，矿工节点继续验证总结区块中包含的交易信息。以后的工作流程与正常的区块验证流程一致。

当总结区块被网络中大部分节点接受并写入本地区块链后，即表示从创世块到最新区块的交易输出的整理工作已经完成。节点可以根据自身情况选择删除本地保存的区块链中的区块体，即存储交易信息的部分。由于矿工节点在后续验证交易的过程中需要使用的UTXO信息已经保存在IPFS网络中，并且在验证总结区块时下载到了本地，因此删除历史区块的区块体对矿工几乎是没有影响的。即使删除历史区块信息，矿工节点依然可以根据UTXO文件对交易信息进行验证。

删除区块体后，区块链依然可以通过区块头中的哈希彼此相连，这部分彼此相连的区块头就形成了一条证明链。本方案没有彻底删除历史区块的所有信息，而是选择保留证明链。这样做的目的是保障安全，即证明该条区块链是从创世块延伸而来，并且具有最多的工作量作为证明。

当然删除全部区块的区块体是非常激进的，根据迷你区块链的设计思想，本文方案中的节点依然保留最新的几千个区块。这样做的目的同样是为了安全，保留的最新区块可以与证明链链接一直延伸至创世块，大大增加攻击者秘密伪造一条假的区块链的成本。

4.2.2 总结区块的生成及验证

在比特币分布式系统中区块的出块间隔是根据挖矿的难度值确定的，目的是让每个区块的平均产生间隔维持在10分钟左右。系统每隔2016个区块就会去检测这些区块的平均出块时间，2016个区块的间隔大约是两周的时间。当发现平均出块时间低于10分钟时，系统就会调高这个难度值，这样在同等算力条件下矿工节点计算出符合要求的头哈希值的概率就会降低，所耗费的时间就会增加。相反，如果发现平均出块时间慢于10分钟，系统就会调低难度值。

结合比特币难度调整的设计思想，本文方案为总结区块之间设置了大小相同的跨度值，每经过一个跨度值的普通区块产生一个总结区块。总结区块之间以普通区块相连接，由于整个系统的出块时间基本恒定，因此以相同跨度产生总结区块满足定期对区块链交易历史进行总结的要求。生成区块的具体流程如图4.3所示。



图4.3 区块产生流程

Fig. 4.3 The generation process of the block

在挖矿过程中发现当前正在计算头哈希的区块为总结区块时，矿工节点首先对本地区块链进行遍历，从上一个总结区块开始，一直到最新区块结束。具体筛选过程以图4.3为例进行说明。



图4.4 区块链中的交易

Fig. 4.4 The trade in the block chain

为了方便说明UTXO的收集过程，图中示例设置的场景较为理想，并且没有考虑交易费等问题。在示例中，从编号0到编号2的三个区块中存在三笔相互关联的交易。其中1号区块，用户b向用户c进行转账的交易输入引用的是0号区块用户a向用户b转账的交易输出。同理2号区块用户c向用户d进行转账的交易输入来自1号区块用户b向用户c转账的交易输出。在不考虑其他交易信息的情况下，如果以2号区块作为总结区块，三笔交易中只有2号区块用户c向用户d发起的转账交易会得到保留。因为0号区块用户a给用户b的转账以及1号区块用户b向用户c的转账的交易输出都被使用过，对于用户a、b、c而言他们都曾经拥有5比特币，但是已经在交易中花费掉了。因此0号以及1号区块中的两笔交易是否保留并不影响后续矿工节点的交易验证过程。

区块具体的优化效率可以通过公式体现，公式如下。

(4.1)

公式4.1中E代表前n个区块优化效率。代表区块链中第i个区块占用的空间。代表第i个区块中包含UTXO交易占用的空间。公式中分母表示优化前n个区块占用空间的总和。分子表示优化后保留的交易信息占用的空间总和。两者相除即为优化的空间效率。

上述优化操作的最终目的是提取从创世区块开始，一直到当前总结区块中所有的有效UTXO信息。将这些信息以文件的形式保存，最后通过调用IPFS提供的开发接口存入星际文件系统IPFS网络中。成功存入网络后，矿工节点将会获得一个指向UTXO文件的哈希值，通过该哈希值可以从IPFS网络中下载到对应的UTXO文件。接下来矿工节点将与文件相对的哈希值写入总结区块的区块头当中。总结区块的区块头结构与普通区块的没有本质区别，只是在其基础上增加了UTXO文件哈希值字段。将文件哈希记录在区块头部，可以利用区块链不可篡改的特性保证UTXO文件的权威性。总结区块的区块头结构如表4.1所示。

表4.1 总结区块头部结构

Tab. 4.1 Block header structure

|  |  |  |
| --- | --- | --- |
| 数据项 | 描述 | 大小 |
| Version（版本号） | 标识当前系统版本 | 4字节 |
| hashPrevBlock(前一区块的哈希值) | 引用上一个区块的哈希值 | 32字节 |
| hashMerkleRoot（梅克尔树根节点哈希值） | 该区块中交易的merkle树根的哈希值 | 32字节 |
| Time（时间戳） | 该区块生成的时间 | 4字节 |
| Bits（难度目标） | 当前挖矿的难度目标 | 4字节 |
| UTXO Hash（UTXO文件哈希） | 存储在IPFS网络中的文件哈希 | 46字节 |
| Nonce（随机数） | 0到32位的随机数 | 4字节 |

完成区块头的填写后，接下来计算区块头哈希的挖矿过程与原比特币工作量证明机制的挖矿过程相同。当矿工通过大量计算得到合理的哈希头后，就会将总结区块向P2P网络中广播，宣布挖矿结果的同时也供其他矿工节点进行验证。节点验证区块的具体流程如图4.5所示。



图4.5 区块验证流程

Fig. 4.5 The verification process of the block

其他参与挖矿的矿工节点收到总结区块后，首先验证包含在总结区块中交易的正确性以及区块头哈希是否符合当前难度要求。如果验证通过，节点需要根据本地保存的区块信息收集创世块产生以来的UTXO信息，继续验证UTXO文件哈希的正确性。由于大部分矿工节点本地保存的区块链数据是相同的，因此使用相同方法收集到的UTXO信息也是一致的。在星际文件系统IPFS网络中，存储文件返回的哈希值与文件的内容直接相关。即将相同内容的文件存入IPFS网络中，返回的哈希值是相同的。因此收到总结区块的矿工节点可以根据自身保存的区块信息对总结区块中保存的文件哈希进行验证，判断文件哈希中保存的UTXO信息与自己收集的数据是否一致。如果数据信息一致，说明本阶段的总结工作没有出现问题，矿工节点可以将总结区块写入本地的区块链中。如果数据信息不一致，矿工节点会认为收到的总结区块不合法。因此矿工节点会丢弃收到的总结区块，转而继续计算自己打包的区块的哈希值，直到计算出符合要求的区块头哈希。如果在挖矿的过程中再次收到总结区块，矿工节点可以继续按照上述流程进行验证。

4.2.3 UTXO文件存储

星际文件系统IPFS是一种点对点的分布式超媒体分发协议[48]，其主要特点是基于内容的寻址方式，与传统基于域名的寻址方式有很大差别。用户在获取文件时不再需要考虑目标服务器的位置，以及目标文件的名称及存储路径。基于以上几点原因，本文方案的文件存储及读取模块使用星际文件系统IPFS完成。

当需要生成UTXO文件时，节点首先将前一个总结区块对应的UTXO文件读入内存，接着从总结区块的下一个区块开始读取区块中的交易记录。在读取交易记录的过程中不断检索内存中的UTXO信息，当内存中保留交易的所有输出都已被后续区块中的交易引用，说明该交易是否存在对后续流程没有影响，可以删除。删除后将新读取的交易加入内存，后续流程不断重复上述步骤，直到读取到最新区块。节点完成区块的读取工作后，将内存中整理的UTXO信息写入文件。接下来节点需要调用IPFS提供的应用开发接口存储整理的UTXO文件。当文件存入IPFS网络后，将会得到一个基于文件内容计算而来的唯一哈希值。哈希值的内容与文件内容密切相关，即使改变一个比特的数据都会导致生成哈希值的变化。

当其他节点需要根据哈希值下载文件时，可以通过本地保存的分布式哈希表向邻居节点发起询问请求，不断向存储有目标文件的目标节点逼近，最终获取UTXO文件信息。这种基于IPFS网络的文件存取模式可以大大加快新加入网络的节点的区块同步效率。

当节点加入网络后不必花费大量时间从创世区块一路下载到最新区块。在本文方案中新加入节点可以通过获取最新总结区块从IPFS网络下载UTXO文件。接下来通过获取后续普通区块，节点可以快速完成验证交易的准备，开始挖矿工作。

4.2.4 证明链

当总结区块在网络中达成共识后，矿工节点可以选择开始区块链历史交易的清理工作。清理的主要内容是本地保存的完整区块的区块体，即历史交易信息。由于可用的交易输出信息已经在生成总结区块环节整理到UTXO文件中，因此拥有较多区块确认的历史交易可以得到清理。清理后的具体结构如图4.6所示。



图4.6 清理后的区块链结构

Fig. 4.6 The block chain structure after cleaning

完成区块历史的清理后，矿工节点可以通过UTXO文件以及保留的最新区块部分验证接收的交易。如果交易输入可以在UTXO文件或保留的区块中找到说明引用的上一笔交易输出未被花费，可以运行验证脚本验证交易合法性。相反如果无法找到上一笔交易的输出说明该交易输出已经被花费或不存在，说明该笔交易无效。图中仅保留区块头的链式结构即为证明链部分，由于维持区块相互连接的哈希值保留在区块头中，删除区块体的区块依然可以与保留的最新区块相互连接，形成一条链式结构。首尾相接的链式结构可以从最新区块开始一直追溯到公认的创世区块，从而保证当前区块链中交易数据的真实性，抵抗网络中秘密挖矿等攻击行为。

由于比特币中有完整的区块链的支持，任意节点都可以从非常早的区块一路验证到最新区块来证明该区块链的正确性。如果选择删除区块全部内容，攻击者可以从可用的最早区块开始，创建一条新的区块链。这样新加入的节点将会无法将它与真正的区块链分开，因为保留的最早区块之前的历史被删除了。攻击者可以花费尽可能多的时间来增加他们的区块链的累积难度，然后开始广播假链。如果传播足够广，伪造链将会有成为主链的风险。证明链通过提供一种可以存储长期工作证明历史的容器的机制来解决这个问题，以便节点计算任何区块链的总累积难度。因此出于安全的考虑，本文方案选择保留最新的一部分区块，删除保存时间较长区块的区块体，同时保留这部分区块的区块头信息。以便新加入节点随时追踪任何给定区块链的历史，并比较每个链的总累积难度。

4.3 UTXO文件健康情况检查机制

在IPFS网络中存储的文件主要以密文的形式存储在多个节点当中。当文件较大时，系统将会对文件进行拆分，使大文件以多个文件块的形式存储。但是在P2P网络中，节点的加入和退出并不受到限制。因此当文件请求方发出下载请求时，存储该文件的节点是否处于在线状态是不确定的。缓存文件的某几个节点处于离线状态会导致文件请求方出现获取数据缓慢甚至长时间处于等待状态的情况出现。因此本文方案设置了一种检查UTXO文件在IPFS网络中的健康状况的机制。

4.3.1 节点角色设置

在本文方案中的点对点网络中主要包含两种节点角色，分别是普通矿工节点和全节点。

矿工的主要工作是通过挖矿的方式竞争向区块链写入最新区块的权利。从本质上说，网络中所有矿工节点的竞争行为维护了比特币网络安全性和稳定性。在不影响交易验证的前提下，矿工节点并不必须保留完成的区块链历史。

全节点与普通矿工节点不同，除了正常的交易验证工作外，还可以提供区块历史数据的查询服务。全节点主要由一些社区以及提供区块查询服务的网站维护，拥有存储以及维护全部区块信息的能力。提供查询的内容不仅包括当前可用的UTXO信息，还包含某账户当前可用交易输出、全部交易记录等信息。因此全节点并不删除区块历史信息，而是保存全部的历史记录。因此，全节点为UTXO文件在IPFS网络中健康情况的检查提供了可能。

4.3.2 UTXO文件健康检查模块

UTXO文件健康情况检查的任务主要由保存完整区块链信息的全节点完成。具体流程如图4.7所示。



图4.7 UTXO文件健康检查流程

Fig. 4.7 The health check process of the UTXO file

全节点每隔一定时间向IPFS网络发出UTXO文件的下载请求，根据请求的响应时间判断下一步任务。如果相应时间较短，且下载速度较快，说明网络中缓存文件的节点较多，文件健康度良好，新加入网络中的节点可以快速获取同步需要的UTXO文件。如果出现相应时间较长，文件下载缓慢的情况，说明节点保存的文件丢失或在线的节点较少。全节点可以通过自身保存的完整区块链总结UTXO文件，向IPFS重新上传UTXO文件，保障文件的健康状况。

第5章 区块空间优化方案实验及结果分析

本文实验包括两部分，分别为比特币交易数据统计分析实验和空间优化实验。比特币交易数据统计分析实验的目的是确定合适的总结区块跨度值。空间优化实验的目的是验证本文提的区块存储优化方案的可行性。

5.1 实验环境介绍

5.1.1 硬件介绍

本次实验主要使用实验室8台服务器组成P2P网络环境。各服务器的硬件参数及操作系统如表5.1所示。

表5.1 实验硬件环境说明

Tab. 5.1 Experiment hardware environment description

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 名称 | 内存 | CPU主频 | 内核数 | 数量 | 硬盘大小 | 操作系统 | 内网IP |
| 服务器1号 | 16GB | Intel Xeon  E5620/2.4GHz | 4 | 8 | 500GB | Ubuntu 16.01 | 192.168.10.2 |
| 服务器2号 | 16GB | Intel Xeon  E5620/2.4GHz | 4 | 16 | 500G | Ubuntu 16.01 | 192.168.10.3 |
| 服务器3号 | 16GB | Intel Xeon  E5620/2.4GHz | 4 | 8 | 1TB | Ubuntu 16.01 | 192.168.10.4 |
| 服务器4号 | 16GB | Intel Xeon  E5620/2.4GHz | 4 | 8 | 1TB | Ubuntu 16.01 | 192.168.10.5 |
| 服务器5号 | 16GB | Intel Xeon  E5620/2.4GHz | 4 | 8 | 500G | Ubuntu 16.01 | 192.168.10.6 |
| 服务器6号 | 16GB | Intel Xeon  E5620/2.4GHz | 4 | 8 | 300G | Ubuntu 16.01 | 192.168.10.7 |
| 服务器7号 | 16GB | Intel Xeon  E5620/2.4GHz | 4 | 8 | 350G | Ubuntu 16.01 | 192.168.10.8 |
| 服务器8号 | 16GB | Intel Xeon  E5620/2.4GHz | 4 | 8 | 500G | Ubuntu 16.01 | 192.168.10.9 |

5.1.2 软件介绍

比特币交易数据统计分析实验使用python语言编写，使用的数据库为Mysql数据库。由于python扩展包bitcoinrpc仅支持python2版本，本文实验使用的python版本为2.7.12。容量优化实验基于比特币核心源码修改，部分实验环境版本信息如图5.1所示。



图5.1 实验环境版本信息

Fig. 5.1 Partial version information of the experiment environment.

5.2 比特币交易数据统计分析实验

交易数据统计分析实验的目的主要包括两点。第一点是验证本文在第4章中对比特币区块链中包含大量已被花费的交易输出的推测。第二点是以真实的比特币历史交易记录为参考，通过分析确定合理的总结区块跨度值。

为了得到真实的比特币区块链数据，首先安装比特币核心客户端，下载比特币完整区块链数据。区块数据的读取操作使用python语言，通过bitcoinrpc扩展包调用比特币核心客户端PRC API端口获得数据[49]。与用户直接通过命令行向比特币核心客户端发送命令的形式类似，python的bitcoinrpc扩展包可以以脚本的形式向比特币客户端发送请求并接收数据。发送的请求主要包括getblockhash()、getrawtransaction()、decoderawtransaction()等方法，用于获取每个区块中各个交易的输入及输出数量。数据的保存模块使用Mysql数据库，使用python扩展包MySQLdb连接Mysql数据库保存读取的数据。数据库表中部分数据如图5.2所示。

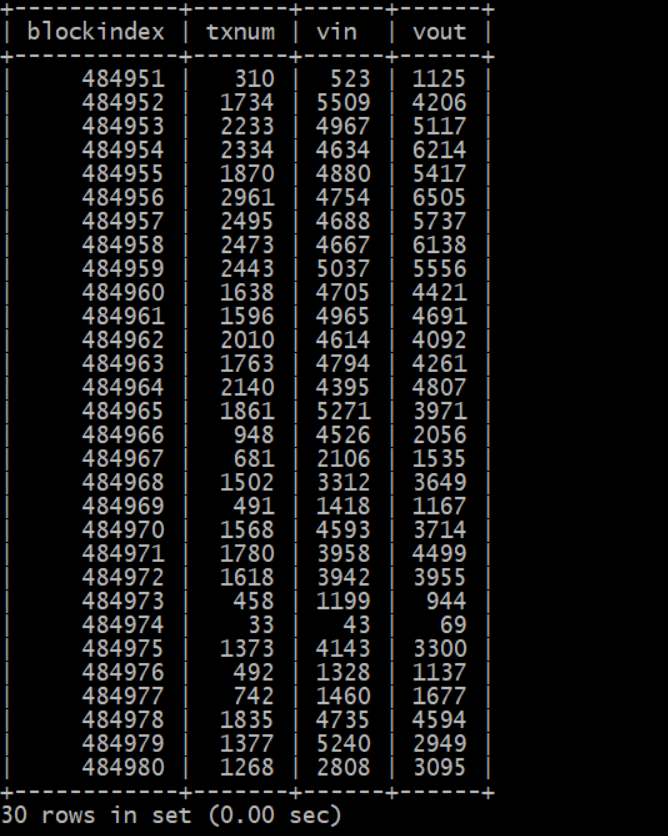


图5.2 数据库表结构

Fig. 5.2 Table Structure

数据库表中字段blockindex代表区块在区块链中的序号。字段txnum代表该区块包含的交易数。字段vin代表区块中包含的交易输入总数。字段vout代表区块中包含的交易输出总数。由于读取脚本包含较多循环遍历以及大量与比特币核心客户端的查询操作，因此整体数据的收集过程耗时较长。读取结束后数据库中共包含数据495000条，对应比特币区块链从创世块开始到第495000的区块。

5.2.1 比特币交易输出统计实验

比特币交易输出统计实验的统计内容包括比特币交易历史中的交易输出和已经被花费的交易输出。根据UTXO模型的定义，区块中的一笔交易的输入必然有上一笔交易的输出与之对应。当一笔交易被确认，该交易引用的未被花费的交易输出就会变为已花费状态。因此新区块中所有交易的总输入数量等于新区块加入后未被花费的交易输出的减少数量。例如新加入区块共包含一百个交易输入，那么当前所有可用的交易输出量应该减去输入的一百。同时新区块也包含交易的输出，由于是最新区块，因此交易输出全部处于未被花费状态。例如新加入区块共包含二百个交易输出，那么当前所有可用的交易输出量应该加上输出的二百。

基于以上原理本实验以一千个区块作为基本单位，统计从创世块开始比特币区块链不同阶段全部交易输出和所有可用交易输出的数量值。比特币历史交易输出统计情况如图5.3所示。

图5.3 比特币交易输出统计图

Fig. 5.3 The statistical graph of output

图5.3中横坐标代表区块序号，纵坐标代表交易输出的数量。虚线表示全部交易输出值，线段上任意一点代表从创世区块到当前区块出现的所有交易输出总量。实线代表未被花费的交易输出值，线段上某一点代表从创世块到当前区块所有未被花费的交易输出值。观察图中曲线可知，从创世块出现一直到第19万块左右，比特币交易量较少。从第19万块附近开始，交易输出逐渐增加，且呈现加速上涨的趋势。相比交易输出总量，未被花费的交易输出量增长较为缓慢。随着后续区块的不断产生，历史交易输出总量与可用的未被花费的交易输出之间的差距逐渐加大。因此本文在第4章中对比特币区块链包含大量可优化空间的推测成立。

5.2.2 已被花费交易输出比例统计实验

已被花费交易输出比例统计实验的目的是统计不同跨度下，已被花费的交易输出比例，最终选取合适的跨度值进行后续实验。实验以1000块为单位，设置不同大小的总结区块跨度值，计算对应跨度区块内已被花费的交易输出占保存的交易输出总量的比例。根据本文的理论推导，已被花费的交易输出即为可优化空间。最终的结果值通过记录每次总结后节省空间的比例相加后计算平均值得出。具体比例分布情况如图5.4所示。

图5.4 不同跨度值已花费交易输出占比

Fig. 5.4 Proportion of the UTXO in different span values

图中横坐标表示不同大小的总结区块跨度值，纵坐标代表被花费的交易输出占当前保留的交易输出比例的平均值。观察曲线可知，当跨度值小于20000时，曲线斜率较大。随着跨度值的增加，可优化空间比例增长较快。当跨度值大于20000时，曲线斜率逐渐减小，可优化空间比例增长速度趋于平缓。就曲线总体增长趋势而言，总结区块跨度越长，优化效率越高。但综合矿工节点实际空间占用情况分析，总结区块跨度值的设置并不是越大越好。由于在比特币系统中，区块产生的时间间隔大概在10分钟左右。当跨度值为1000对应时间为7天左右，跨度值为65000对应时间间隔为451天左右。因此总结区块跨度值越大，意味着每次清理数据的周期越长。根据比特币真实交易数据情况分析图中曲线可知，当总结区块间隔为两万五千块左右时，系统拥有较为合适的平均优化效率。以区块的产生间隔为十分钟计算，两万五千个区块对应173天左右，优化效率平均在百分之六十左右。对于运行八年，产生区块达到五十万左右的比特币系统而言，以半年左右的周期进行总结清理是可以接受的。

5.3 存储优化方案实验

本文中存储优化方案基于版本号v0.15.0的比特币客户端修改。实验硬件环境为八台服务器组成的小型P2P网络，占用的测试网络端口为18333。网络中模拟交易数据由1号服务器广播。其余节点负责接收广播并通过挖矿产生区块。

5.3.1 容量优化实验

容量优化实验的目的是验证改进方案的改进效果。实验使用相同数据集，在原比特币方案和改进方案上分别进行。为了减少出块时间，保证实验效率，本实验选择降低两组方案中的挖矿难度。根据上文5.2.2小结初步得出的结论，优化方案中总结区块跨度值选择为25000。实验中的交易数据根据比特币区块链中真实交易数据生成。交易信息依然由1号服务器以广播的形式向P2P网络发布。实验数据的收集主要通过定期记录两种方案客户端中区块文件在磁盘中的空间占用情况完成。最终根据统计数据，分析改进方案相比原比特币客户端的空间优化程度。

通过观察，各节点blocks文件夹中有区块文件产生，UTXO文件能够以平均15分钟的频率进行更新。利用IPFS客户端程序能够通过总结区块中保存的哈希值顺利获取UTXO文件。说明本文存储优化方案改进对比特币客户端验证交易及区块等基本功能影响较小。改进后矿工节点能够正确验证交易信息并稳定产生区块。

两组方案中区块占用存储空间数据随区块数增长的变化情况对比如图5.5所示。

图5.5 节点存储空间占用对比

Fig. 5.5 Comparison of storage conditions

图5.5中横坐标代表实验产生的区块序号，单位为块。本次实验共产生三十万块。纵坐标代表产生的区块信息占用磁盘的存储空间值，单位为Mb。虚线段代表原比特币方案保存全部区块信息情况下，被占用存储空间随区块增加的变化趋势。实线段代表加入空间优化机制的改进方案中，磁盘存储空间随区块增加的变化趋势。

观察虚线段增长趋势可知，随着区块不断产生，历史交易信息逐渐增加，并且在超过二十万个区块后呈现加速的趋势。通过观察图中实线段可以看出，由于前15万区块中交易数量较少，比特币方案与优化方案曲线变化相似，优化效果并不明显。当区块数大于15万后，区块中交易数量快速增加，比特币方案中区块占用空间呈现加速增长。优化方案磁盘空间占用值增长相对缓慢，线段每经过25000个区块就会产生一次下降，对应一个总结区块周期。本次实验共产生总结区块11个，每个总结周期删除数据的平均比例约为49.5%。最终区块链数据在比特币方案中占用空间约为19.6Gb，在优化方案中占用空间约为6.6Gb。

实验数据说明总结区块能够正常产生，节点能够随着总结区块的产生清理历史交易数据。对比原比特币方案与改进方案在区块链存储空间方面的表现可知，随着区块不断产生，节点在本地保存的交易数据也在不断增加。但是随着交易的输入输出之间不断引用，大量交易输出从未被花费状态转变为已被花费状态。因此随着区块不断增长，优化方案能够有效减少区块链的空间占用。

5.3.2 双花攻击实验

双花攻击实验的目的是检测优化方案对攻击者多次花费同一交易输出的抵抗能力。实验具体内容是模拟攻击者向网络发布花费同一笔交易输出的交易信息，观察最终是否有多笔交易被写入区块链中。利用1号服务器向P2P网络广播大量花费同一交易输出的交易数据，检查各节点对双重花费的交易数据的处理情况。

为了方便查询交易，各节点需要执行./bitcoind -reindex -txindex命令建立索引。通过在各节点执行getrawtransaction命令，利用交易哈希对非法交易进行查询。当交易输出变为已花费状态时，查询后续多个花费该交易输出的非法交易的返回结果值均为-5。该返回值为错误码，表示交易不存在。在查询正确交易的哈希时，客户端能够返回对应交易的十六进制信息。

实验说明本文方案具有抵抗双花攻击的能力。在删除部分历史交易数据的情况，交易的验证功能运行正常，节点能够通过UTXO文件中保存的未被花费的交易输出信息判断交易的正确性。

5.4 实验分析与小结

通过分析本文模拟实验可知，针对矿工节点的空间优化具有可行性。迷你区块链的空间优化思想在比特币UTXO模型中依然适用，系统基本功能正常。

在比特币交易数据统计分析实验中，通过分析比特币真实交易数据，证实了区块链中包含大量可优化数据的推论。

在容量优化实验中，本文方案能够明显减少区块链的占用空间。同时矿工节点能够完成正常区块以及总结区块的打包流程。其他矿工节点能够正确接收并验证区块合法性。通过验证的总结区块能够写入区块链并在网络中达成共识。在模拟环境中交易历史信息能够定期总结，正确生成UTXO文件并顺利上传至IPFS网络中。在正常挖矿环节，清理过历史交易数据的矿工节点依然能够验证交易的正确性。

在双花攻击实验中，本文方案能够识别无效的交易信息。当交易引用的上一笔交易输出已被花费时，节点能够正确识别。

因此分析整体实验数据可知，本文方案结合迷你区块链以及IPFS网络的设计思路较为合理。通过定期对历史区块进行总结，剔除对验证交易没有作用的过期交易输出来达到空间优化的思路具有可行性。矿工节点能够在不影响交易验证及记录的同时，通过定期总结的方式达到优化区块存储空间的目的。

# 

第6章 总结与展望

从2009年创世块开始，比特币区块链经历多次起伏[50]，跨越五十万个区块，安全稳定地延伸到了今天。但是随着区块数量不断增加，区块链在磁盘占用的空间也越来越多，区块链的本地存储逐渐变成一个不容忽视的问题[51]。

6.1 总结

针对现有区块链空间优化方案存在的各种问题，本文以经典的比特币分布式架构为基础，利用星际文件系统的文件存储转发功能和迷你区块链思想设计了一种针对矿工节点的区块链空间优化方案。通过定期对历史区块进行总结，剔除对验证交易没有作用的过期交易输出来达到空间优化的目的。在优化过程中被保留下来的未被花费的交易输出UTXO将会以文件的形式完整的保存在IPFS网络中，方便新加入的矿工节点读取。本文研究的主要工作如下。

（1）通过阅读文献资料，掌握比特币运行原理。针对比特币矿工节点，提出区块链中存在可优化空间的推测。

（2）利用Python脚本对比特币区块链交易数据进行读取分析。统计不同阶段，区块链交易输出总量及未被花费的交易输出数量。对区块链中包含可优化空间的推测进行论证。

（3）设置大小不同的跨度值对比特币区块链数据进行拆分，统计不同跨度值中未被花费的交易输出比例。结合实际情况，确定合理的跨度取值。

（4）对优化方案进行设计与实现。在满足可用性和安全性的条件下，通过设置总结区块模块、UTXO文件存储模块、证明链等模块，达到定期清理历史交易数据的目的。

（5）对本文提出的优化方案进行实验。使用相同的交易数据分别对比特币方案和优化方案进行测试。记录实验过程中，两组方案生成区块链在节点中的空间占用情况。通过分析实验结果，得出结论。

实验结果表明，改进方案中历史区块空间优化效果明显。优化方案能够在不影响矿工节点正常工作流程的条件下，达到减少区块存储空间的目的。通过定期对历史交易数据进行总结，剔除对验证交易没有作用的过期交易输出的优化思路具有可行性。

6.2 展望

本文方案引入迷你区块链设计思想，能够在不影响矿工正常工作流程的条件下对区块的占用空间情况进行优化。但是节省空间的同时，系统整体安全性可能会受到一定的影响。由于本文实验环境资源有限，并且P2P网络环境下恶意攻击的方法多种多样。因此本文方案对于网络攻击的抵抗能力及表现可以作为下一步研究的内容。

另外在本文方案中总结区块跨度的取值方法是设置不同大小跨度结合比特币真实数据统计分析优化效率。在下一步研究中可以考虑根据节点当前的数据存储情况，在网路达成共识的情况下动态调节总结区块跨度值，使节点对区块历史数据的处理更加灵活。

参 考 文 献

[1] Böhme R, Christin N, Edelman B, et al. Bitcoin: Economics, Technology, and Governance[J]. Journal of Economic Perspectives, 2015, 29(2):213-238.

[2] Pilkington M. Blockchain Technology: Principles and Applications[J]. Social Science Electronic Publishing, 2016.

[3] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[J]. Consulted, 2008.

[4] Bonneau J, Miller A, Clark J, et al. Research Perspectives and Challenges for Bitcoin and Cryptocurrencies[J]. 2015, to appear:104-121.

[5] Donet J A D, Pérez-Solà C, Herrera-Joancomartí J. The Bitcoin P2P Network[C]// The Workshop on Bitcoin Research. 2014:87-102.

[6] 袁勇, 王飞跃. 区块链技术发展现状与展望[J]. 自动化学报, 2016, 42(4):481-494.

[7] 梅兰妮・斯万. 区块链:新经济蓝图及导读[M]. 新星出版社, 2016.

[8] 林晓轩. 区块链技术在金融业的应用[J]. 中国金融, 2016(8):17-18.

[9] 韩伟力, 梁蛟, 郭泽卿,等. 基于区块链的数据权益保护方法:, CN 106600405 A[P]. 2017.

[10] 张亚娇, 王枞. 区块链技术在医疗数据安全存储中的应用[J]. 2016.

[11] 张新宇. 区块链技术在保险行业的应用研究[J]. 中国金融电脑, 2017(1):48-50.

[12] Kim H M, Laskowski M. Towards an Ontology-Driven Blockchain Design for Supply Chain Provenance[C]// Submitted: Workshop on Information Technology and Systems. 2016.

[13] Sharples M, Domingue J. The Blockchain and Kudos: A Distributed System for Educational Record, Reputation and Reward[C]// European Conference on Technology Enhanced Learning. Springer, Cham, 2016:490-496.

[14] Jaag C, Bach C. Blockchain Technology and Cryptocurrencies: Opportunities for Postal Financial Services[J]. Working Papers, 2017.

[15] Swan M. Blockchain: Blueprint for a New Economy[M]. O'Reilly Media, Inc. 2015.

[16] Wood G. Ethereum: A secure decentralised generalised transaction ledger[J]. Ethereum Project Yellow Paper, 2014, 151.

[17] Buterin V. Ethereum white paper[J]. 2013.

[18] King S, Nadal S. Peercoin–Secure & Sustainable Cryptocoin[J]. Aug-2012 [Online]. Available: https://peercoin. net/whitepaper.

[19] King S, Nadal S. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake[J]. self-published paper, August, 2012, 19.

[20] Blockmeta, the blockchain data of Bitcion. [EB/OL]. [2017-11-07]. https://blockmeta.com//btc-stat.

[21] 贾大宇,信俊昌,王之琼,郭薇,王国仁.区块链的存储容量可扩展模型[J/OL].计算机科学与探索:1-10, 2017.

[22] Bruce J D. Purely P2P crypto-currency with finite mini-blockchain[J]. 2013.

[23] Benet J. IPFS - Content Addressed, Versioned, P2P File System[J]. Eprint Arxiv, 2014.

[24] 周平, 唐晓丹. 区块链与价值互联网建设[J]. 信息安全与通信保密, 2017(8):53-59.

[25] 赵刚. 区块链:价值互联网的基石[M]. 电子工业出版社, 2016.

[26] Wright A, De Filippi P. Decentralized Blockchain Technology and the Rise of Lex Cryptographia[J]. Social Science Electronic Publishing, 2015.

[27] 蒋润祥,魏长江. 区块链的应用进展与价值探讨[J]. 甘肃金融, 2016(2):19-21.

[28] Antonopoulos A M. Mastering Bitcoin: Unlocking Digital Crypto-Currencies[M]. O'Reilly Media, Inc. 2014.

[29] Okupski K. Bitcoin Developer Reference[J]. Availabl e at http://enetium. com/resources/Bitcoin. pdf, 2014.

[30] O'Dwyer K J, Malone D. Bitcoin mining and its energy footprint[C]// Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies. IET, 2014:280-285.

[31] Buchmann J, Dahmen E, Schneider M. Merkle Tree Traversal Revisited[M]// Post-Quantum Cryptography. Springer Berlin Heidelberg, 2008:63-78.

[32] Franco P. Understanding Bitcoin: Cryptography, engineering and economics[M]. John Wiley & Sons, 2014.

[33] blockchain.info, the blockchain data of Bitcion. [EB/OL]. [2017-12-07]. https://blockchain.info/block/000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

[34] Chaudhary K, Fehnker A, Pol J V D, et al. Modeling and Verification of the Bitcoin Protocol[J]. Computer Science, 2015, 196(Proc. MARS 2015):46-60.

[35] Szydlo M. Merkle tree traversal in log space and time[C]//Eurocrypt. 2004, 3027: 541-554.

[36] 唐文, 南相浩, 陈钟. 基于椭圆曲线密码系统的组合公钥技术[J]. 计算机工程与应用, 2003, 39(21):1-3.

[37] Miller V S. Use of elliptic curves in cryptography[C]//Conference on the Theory and Application of Cryptographic Techniques. Springer, Berlin, Heidelberg, 1985: 417-426.

[38] Stallings W. Secure Hash Algorithm[J]. Cryptography & Network Security Principles & Practice, 2007:1116-1116.

[39] Dobbertin H, Bosselaers A, Preneel B. RIPEMD-160: A strengthened version of RIPEMD[C]// Springer-Verlag, 1996:71-82.

[40] Decker C, Wattenhofer R. Bitcoin transaction malleability and MtGox[C]//European Symposium on Research in Computer Security. Springer, Cham, 2014: 313-326.

[41] Fielding R, Gettys J, Mogul J, et al. Hypertext transfer protocol--HTTP/1.1[R]. 1999.

[42] Tung Y C, Lin K C J, Chou C F. Bandwidth-aware replica placement for peer-to-peer storage systems[C]//Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE. IEEE, 2011: 1-5.

[43] Ng W S, Ooi B C, Tan K L, et al. PeerDB: A P2P-based System for Distributed Data Sharing[C]// International Conference on Data Engineering. IEEE Computer Society, 2003:633.

[44] Armknecht F, Bohli J M, Karame G O, et al. Outsourced proofs of retrievability[C]//Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2014: 831-843.

[45] Franca B F. Homomorphic Mini-blockchain Scheme[J]. 2015.

[46] Li J, Wolf T. A one-way proof-of-work protocol to protect controllers in software-defined networks[C]//Architectures for Networking and Communications Systems (ANCS), 2016 ACM/IEEE Symposium on. IEEE, 2016: 123-124.

[47] Kroll J A, Davey I C, Felten E W. The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries[J]. Proceedings of Weis, 2013.

[48] Curran T, de Graaff B. Analysing the Performance of IPFS During Flash Crowds[J]. 2016.

[49] Djilali S. P2P-RPC: Programming scientific applications on peer-to-peer systems with remote procedure call[C]//Cluster Computing and the Grid, 2003. Proceedings. CCGrid 2003. 3rd IEEE/ACM International Symposium on. IEEE, 2003: 406-413.

[50] Bonneau J, Miller A, Clark J, et al. SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies[C]// Security & Privacy. IEEE Computer Society, 2015:104-121.

[51] Alqassem I, Svetinovic D. Towards Reference Architecture for Cryptocurrencies: Bitcoin Architectural Analysis[C]// IEEE International Conference on Internet of Things. IEEE Computer Society, 2014:436-443.

攻读学位期间公开发表论文

[1] Bo Li, Yanyan Fu, Zhihuai Li. The research and improvement of distributee caching system Memcached. International Conference on Information, Cybernetics and Computational Social Systems (ICCSS), 2017, 460-463.

致 谢

首先，我要感谢我的指导老师李志淮教授，从本科的毕业设计到研究生的毕业论文一直都悉心指导。李老师不仅仅在学术上对我提供了帮助，指导我日常的学习。在平日的生活中，李老师对我的人生观和价值观也有积极的影响，传授了许多有益的观点。从论文的开题到论文的中期，以及最后的毕业答辩，李老师都一直在给我提供支持与帮助。对此本人表示衷心的感谢。

其次，我要感谢陈玉华老师在论文最后的定稿阶段对我提供的帮助。感谢陈老师能够在百忙之中抽出时间来审阅我的论文，并提出格式和论文内容方面的不足。

再者，在本篇论文的写作过程中，研究室中的其他成员也提供了许多帮助。从论文写作前期的思路理清，到论文写作过程中的讨论研究，以及最后帮助查看论文的格式等，大家都很积极地提供各种协助。在此我也表示对他们的感谢。

最后，感谢各位评审老师们抽出宝贵的时间对我的论文进行的评阅，并提出宝贵的修改意见。