(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 110741372 A
(43)申请公布日 2020.01.31

(21)申请号 201880038016.8

(22)申请日 2018.06.05

(30)优先权数据
1709098.6 2017.06.07 GB
1709099.4 2017.06.07 GB

(85)PCT国际申请进入国家阶段日
2019.12.07

(86)PCT国际申请的申请数据
PCT/IB2018/054006 2018.06.05

(87)PCT国际申请的公布数据
WO2018/224954 EN 2018.12.13

(71)申请人 区块链控股有限公司
地址 安提瓜和巴布达圣约翰

(72)发明人 G·德斯蒂法尼斯 S·马蒂奥
P·莫蒂林斯基 S·文森特

(74)专利代理机构 上海晨皓知识产权代理事务所(普通合伙) 31260
代理人 成丽杰

(51)Int.Cl.
G06F 21/64(2006.01)

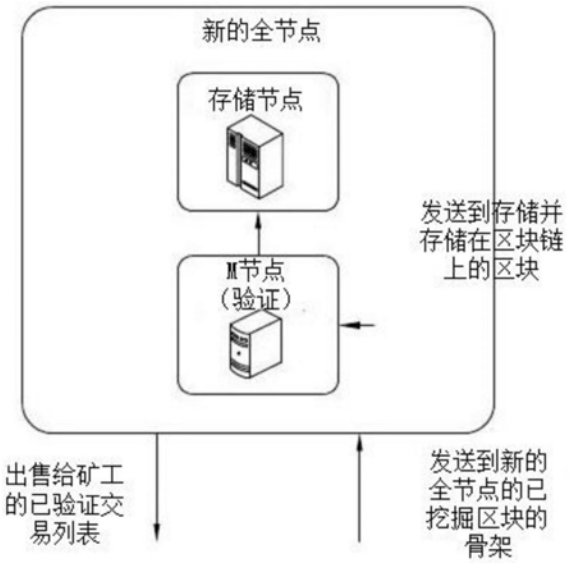
权利要求书2页 说明书21页 附图6页

(54)发明名称

用于管理区块链网络上的交易的计算机实现的系统和方法

(57)摘要

提供了适合在区块链网络的交易验证节点中实现的计算机实现的方法和系统。描述了用于处理大量交易和大交易区块的修改后的区块链节点结构、网络架构和协议。本发明特别适于与比特币区块链一起使用,但不限于此。提供了一种计算机实现的方法,包括:(i)从所述区块链网络接收交易;(ii)验证从所述区块链网络接收的所述交易;(i)与区块链网络中其他交易验证节点一起维护已验证交易的分布式、去中心化存储;和(iv)将对应于所述已验证交易的数据分发到所述区块链网络用于挖掘。



CN 110741372 A

1.一种用于区块链网络的交易验证节点的计算机实现的方法,所述计算机实现的方法包括:

(i)从所述区块链网络接收交易;

(ii)验证从所述区块链网络接收的所述交易;

(i)与所述区块链网络中的其他交易验证节点一起维护已验证交易的分布式、去中心化存储;和

(iv)将对应于所述已验证交易的数据分发到所述区块链网络用于挖掘。

2.根据权利要求1所述的计算机实现的方法,其中,

与所述区块链网络中的其他交易验证节点一起维护已验证交易的分布式、去中心化存储的步骤包括:同步所述区块链网络上的交易验证节点,以去中心化和分布式的方式维护所述已验证交易的最新列表。

3.根据权利要求2所述的计算机实现的方法,

其中,通过交换可逆布隆过滤器查找表来同步所述验证节点。

4.根据前述权利要求中任一项所述的计算机实现的方法,其中,

所述已验证交易按定义的顺序排序,以便在所述区块链网络中的交易验证节点上使用通用排序系统,以维护所述已验证交易的分布式、去中心化存储。

5.根据权利要求4所述的计算机实现的方法,其中,

使用规范的排序系统将所述已验证交易按所述定义的顺序排序,以维护所述已验证交易的分布式、去中心化存储。

6.根据前述权利要求中任一项所述的计算机实现的方法,其中,

将对应于所述已验证交易的数据分发到所述区块链网络以用于挖掘的步骤包括:

准备与所述已验证交易的列表对应的数据。

7.根据前述权利要求中任一项所述的计算机实现的方法,

其中,将对应于所述已验证交易的数据分发到所述区块链网络以用于挖掘的步骤包括:

创建用于数字资产换取向矿工提供与所述已验证交易的列表对应的所述数据的承诺交易。

8.根据权利要求7所述的计算机实现的方法,

其中,在所述承诺交易包括在内的情况下,计算散列树、帕特里夏树、或另一类型的基数树。

9.根据前述权利要求中任一项所述的计算机实现的方法,其中,

对应于所述已验证交易的所述数据以可逆布隆查找表和任何附带数据的形式分发到所述区块链网络。

10.根据前述权利要求中任一项所述的计算机实现的方法,还包括:

(v)从所述区块链网络接收对应于所述已验证交易的已挖掘数据;

(vi)基于所述已挖掘数据组装区块;和

(vii)将已组装区块发送到存储实体以存储在区块链上。

11.根据权利要求10所述的计算机实现的方法,其中,

从所述区块链网络接收的所述已挖掘数据包括对应于所述已验证交易的区块头。

12.根据权利要求10或11所述的计算机实现的方法,其中,

所述已挖掘数据包括用于数字资产换取基于所述已挖掘数据组装所述区块的交易。

13.根据权利要求10至12中任一项所述的计算机实现的方法,

其中,所述已挖掘数据包括用于数字资产换取所述已组装区块的存储的交易。

14.根据权利要求12或13所述的计算机实现的方法,还包括:

在接收所述数字资产之前,要求等待与最小数量的区块相关联的时间段t。

15.根据权利要求10至14中任一项所述的计算机实现的方法,其中,

基于所述已挖掘数据组装所述区块的步骤包括组装大区块,其中,每个该区块的大小至少为2兆字节。

16.根据权利要求10至15中任一项所述的计算机实现的方法,其中,

所述区块包括包含由矿工提供的随机数的区块头。

17.根据权利要求10至16中任一项所述的计算机实现的方法,其中,

所述存储实体在所述区块链网络上的多个节点之间共享,所述多个节点形成所述区块链网络上的超级节点,其中,共享的所述存储实体是公共存储节点、分布式存储设备或两者的组合。

18.一种包括计算机可执行指令的计算机可读存储介质,所述计算机可执行指令在被执行时使处理器执行权利要求1至17中任一项所述的方法。

19.一种电子设备,包括:

接口设备;

一个或多个处理器,耦接到所述接口设备;

存储器,连接到所述一个或多个处理器,所述存储器上存储有计算机可执行指令,所述计算机可执行指令在被执行时使所述一个或多个处理器执行权利要求1至17中任一项所述的方法。

20.一种区块链网络的交易验证节点,所述交易验证节点用于执行权利要求1至17中任一项所述的方法。

21.一种区块链网络的超级节点,所述超级节点包括:

根据权利要求20所述的多个验证节点;和

共享存储实体,用于存储所述区块链,

其中,所述共享存储实体是公共存储节点、分布式存储设备或两者的组合,和

其中,由所述多个验证节点组装的区块被发送到并存储在所述共享存储实体上,由此所述共享存储实体维护所述区块链。

22.根据权利要求21所述的超级节点,

其中,所述共享存储实体包括至少100千兆字节的存储容量。

23.一种区块链网络,包括多个根据权利要求21或22所述的超级节点,

其中,所述超级节点连接至所述区块链网络,其中,每个所述超级节点的共享存储实体用于存储所述区块链的副本,并且其中,所述区块链网络包括至少10个所述超级节点。

用于管理区块链网络上的交易的计算机实现的系统和方法

技术领域

[0001] 本说明书主要涉及适用于在区块链网络的交易验证节点中实现的计算机实现的方法和系统。描述了用于处理大量交易和大交易区块的修改后的区块链节点结构、网络架构和协议。本发明特别适用于但不限于与比特币区块链一起使用。

背景技术

[0002] 在本文献中，我们使用术语“区块链”来包括所有形式的电子、基于计算机的分布式账本。它们包括但不限于区块链和交易链技术、许可的账本和未经许可的账本、共享账本及其变体。虽然提出并开发了其他区块链实施方式，但是区块链技术最广为人知的应用是比特币账本。虽然本文中出于方便和说明的目的可以引用比特币，但是应当注意，本发明不限于与比特币区块链一起使用，替代性的区块链实施方式和协议也落入本发明的范围内。

[0003] 区块链是一种基于共识的电子账本，它被实现为基于计算机的分散式、分布式系统，该系统由区块组成，而区块相应地由交易和其他信息组成。对于比特币而言，每个交易是一个数据结构，该数据结构对区块链系统中的参与者之间的数字资产的控制转移进行编码，并包括至少一个输入和至少一个输出。每个区块包含前一个区块的散列，以致于这些区块变为链接在一起，以创建自区块链开始以来就已经写入区块链的所有交易的永久、不可更改的记录。交易包含嵌入其输入和输出中称为脚本的小程序，它们指定如何以及通过谁来访问交易的输出。在比特币平台上，这些脚本是使用基于堆栈的脚本语言来编写的。

[0004] 为了将交易写入区块链，必须对其进行“验证”。某些网络节点充当矿工并进行工作以确保每次交易都有效，而无效交易从网络中被拒绝。例如，安装在节点上的软件客户端对引用未耗用交易输出(UTXO)的交易进行该验证工作。可通过执行其锁定和解锁脚本来进行验证。如果锁定和解锁脚本的执行评估为TRUE，并且如果满足某些其他条件，则交易有效并且可以被写入区块链。因此，为了将交易写入区块链，该交易必须i)由接收交易的节点来验证-如果交易被验证，则节点将其中继到网络中的其他节点；ii)被加入由矿工建造的新区块；iii)被开采，即加入过去交易的公共账本。当向区块链添加足够数量的区块以使交易实际上不可逆时，认为交易被确认。

[0005] 虽然区块链技术由于加密货币实现方式的使用而广为人知，但是数字企业家已经开始探索比特币所基于的加密安全系统以及可以存储在区块链上以实现新系统的数据这两者的使用。如果区块链可用于并非纯粹限于以加密货币计价的支付的自动化任务和过程，这将是非常有利的。这样的解决方案将能够利用区块链的好处(例如，事件的永久性、防篡改记录，分布式处理等)，同时在其应用中更通用。

[0006] 研究的领域之一是使用区块链来实现“智能合约(smart contract)”。这些智能合约是旨在自动执行机器可读合约或协议的条款的计算机程序。与用自然语言编写的传统合约不同，智能合约是一种机器可执行程序，其包括可处理输入以产生结果的规则，然后可以使得根据这些结果来执行操作。

[0007] 与区块链相关的另一领域是使用“代币(Token)”(或“彩色币(Coloured Coins)”)

通过区块链来表示和传输现实世界的实体。潜在的敏感或机密项可以由没有明显意义或价值的代币来表示。因此，代币充当允许从区块链引用现实世界的项目的标识符。

[0008] US2015/0287026公开了一种热钱包(Hot Wallet)服务系统。热钱包服务系统从一个或多个用户设备接收金融交易，使用实现多签名认证系统的多个认证服务器认证金融交易，聚合数字签名，并将认证的金融交易传播到虚拟货币网络中。一旦金融交易通过虚拟货币网络传播，金融交易就可以由虚拟货币网络的矿工实体以常规方式并入公共共享账本中。热钱包服务系统还包括分析系统，该分析系统监控虚拟货币网络并可以生成信用等级或分数，该信用等级或分数可以用于例如防止某些金融交易由多签名认证系统处理。

[0009] CN106548349涉及的问题是，如果区块链网络中的每个节点都需要验证每个交易的交易信息，那么这个过程产生相当大的工作量，给节点带来很大的压力。在CN106548349中描述的解决方案是将交易信息从第一节点传播到多个选定的第二节点，以便验证交易信息，而不是要求所有节点都需要验证交易信息。以随机、不规则的方式选择第二节点，以防止恶意欺诈的可能性，并且无需每个节点验证交易信息。

[0010] WO2017/162904针对一种基于区块链的资源管理系统，在该系统中，通过在资源管理系统中建立新的区块之前验证交易，减少了交易延迟，减轻了资源双重花费。提供至少一个验证节点来验证交易，并向商户提供验证接受消息以将项目释放给顾客。这样，销售可以可靠地完成，而无需等待区块链的下一个区块建成。WO2017/162904还公开了验证节点可将验证请求分配到一组验证节点，并从该组验证节点接收响应。然后，当且仅当来自该组验证节点的响应均不拒绝交易时，验证节点可向商户发送验证接受消息。WO2017/162904还公开了当在区块链中建立区块的时间到来时，建立区块的节点可编译自最近建立的区块以来已经发生的一组交易，该下一个区块包括该组交易。

[0011] US2016/0259937、WO2017/004527和WO2017/011601描述了比特币区块链网络的配置和功能。特别地，各个交易通过网络传播，并通过挖掘并入区块链中，其中挖掘节点选择一组交易，将交易分组到原型区块(Prototype Block)中，并以传统的“工作证明”方式确定只被使用一次的非重复的随机数(Number Used Once，简称Nonce)的值。一旦矿工找到有效的只被使用一次的非重复的随机数，该区块就被其他节点验证，并合并到区块链中。

[0012] US2015/0310424公开了一种通过多模式密钥地址映射来生成用户目录数据的系统。加密货币网络包括加密节点，用于公开验证加密货币用户的一组交易。节点可以通过中继交易、维护验证账本和/或挖掘加密货币来参与加密货币网络。

发明内容

[0013] 在撰写本文时，比特币区块链网络基于包含大约2000个交易的区块大小，并且大约每10分钟挖掘区块一次(10分钟的区块时间设置为第一确认时间和在链拆分上浪费的时间之间的折衷)。这提供了大约每秒3.5个交易的交易处理速率。相比之下，VISA系统以大约每秒10000个交易、并且可能达到每秒50000多个交易的交易处理速率运行。

[0014] 显然，为了建立有竞争力的支付系统，有必要对区块链网络当前的限制条件进行某种规避。由于已经确定10分钟的区块时间，因此必须考虑区块大小的变化，进而考虑区块链本身的变化。在本说明书中，描述了一种可扩展的解决方案，其例如每秒大约能够处理50000个交易。重要的是，提供了一种可以支持高交易速率的解决方案，同时保留去中心化

的、分布式系统架构。

[0015] 从比特币网络的当前架构到可以处理大量同步交易的架构的过渡给整个网络带来了基础架构压力。已经提出了一种节点系统，其能够更有效且更快地验证和中继大量交易。由于交易量大，在存储池的设计中已经提出了分布式方法(尚未处理以及存储在区块链上的待处理交易)。交易量的大幅增加对区块大小产生了压力，并且当区块大小超过特定限制时，存储基础架构的问题就成为一个重要问题。在本说明书中，描述了处理和存储大的、千兆字节大小的区块问题的解决方案。

[0016] 本说明书描述了一种修改后的区块链网络架构，所述修改后的架构包括多个专用交易验证节点，所述多个专用交易验证节点用于验证交易并维护与区块链网络中的其他交易验证节点的已验证交易的分布式、去中心化存储。交易验证节点还用于为矿工准备已验证交易的列表，并且还创建用于数字资产换取向矿工提供已验证交易的列表的承诺交易。这样，专用交易验证节点在验证交易、构造已验证交易的列表、以及向矿工提供列表以换取费用的方面向矿工提供服务。一旦交易被挖掘，则已挖掘交易被提供给交易验证节点，所述交易验证节点构造已挖掘的交易的大区块并将该大区块存储在专用存储节点中。

[0017] 本发明的一个方面针对验证节点中用于接收、验证、存储和分发交易到区块链网络以进行挖掘的方法。提供了一种用于区块链网络的交易验证节点的计算机实现的方法，所述计算机实现的方法包括：

[0018] (i)从所述区块链网络接收交易；

[0019] (ii)验证从所述区块链网络接收的所述交易；

[0020] (iii)与所述区块链网络中其他交易验证节点一起维护已验证交易的分布式、去中心化存储；和

[0021] (iv)将对应于所述已验证交易的数据分发到所述区块链网络以用于挖掘。

[0022] 分发到区块链网络用于挖掘的、对应于已验证交易的数据可包括已验证交易的列表。每个列表可以提供一个完整的已验证交易列表，以便挖掘到区块中。

[0023] 这种方法有效地消除了矿工执行验证功能的要求，同时与区块链网络中其他交易验证节点一起保留了已验证交易的分布式、去中心化存储。此外，该方法使得交易验证节点能够通过准备对应于已验证交易的数据并将其分发到区块链网络用于挖掘来向矿工提供服务。例如，该方法使得能够准备和分发已验证交易的列表。

[0024] 所述计算机实现的方法还可包括：

[0025] (v)从对应于所述已验证交易的所述区块链网络接收已挖掘数据；

[0026] (vi)基于所述已挖掘数据组装区块；和

[0027] (vii)将已组装区块发送到存储实体以存储在区块链上。

[0028] 这些进一步的方法步骤使得验证节点能够构造要存储在存储实体上的大区块，无需矿工构造和存储大区块并通过区块链网络传输此类区块。此外，该架构允许使用专用于存储大的且不断增长的区块链的大型存储实体。

[0029] 与所述区块链网络中的其他交易验证节点一起维护所述已验证交易的分布式、去中心化存储的步骤可包括同步所述区块链网络上的交易验证节点，以去中心化和分布式的方式维护所述已验证交易的最新列表。例如，可以通过交换可逆布隆过滤器查找表来同步所述验证节点。所述已验证交易按定义的顺序排序，以便在所述区块链网络中的交易验证

节点上使用通用排序系统，以维护所述已验证交易的分布式、去中心化存储。例如，可使用规范的排序系统来维护所述已验证交易的分布式、去中心化存储。已得知这是对于保持去中心化、分布式存储的同时确保以一致的方式维护网络上的交易数据的一个特别有效的方法。

[0030] 将对应于所述验证交易的数据分发到所述区块链网络以用于挖掘的步骤可包括：准备与所述已验证交易的列表对应的数据(例如可逆布隆查找表和与所述已验证交易的列表对应的任何附带数据)。此外，将对应于所述已验证交易的数据分发到所述区块链网络以用于挖掘的步骤包括：创建用于数字资产换取向矿工提供与所述验证交易的列表对应的所述数据的承诺交易。例如，在所述承诺交易包括在内的情况下，计算散列树、帕特里夏(Patricia)树、或另一类型的基数树。

[0031] 在通过解决相关联的密码难题(例如散列难题)来分发和挖掘对应于验证交易的数据之后，已挖掘数据被发送回交易验证节点，而不是由矿工直接存储在区块链上。已挖掘数据可以组装成(大)区块，并存储在专门为存储大量数据而配置的存储实体上和/或分布式存储系统中。如前所述，这使得验证节点能够构造要存储在存储实体上的大区块，无需矿工构造和存储大区块并通过区块链网络传输此类区块。此外，该架构允许使用专用于存储大的且不断增长的区块链的大型存储实体。

[0032] 从所述区块链网络接收的所述已挖掘数据可包括对应于所述已验证交易的区块头。所述已挖掘数据还可包括用于数字资产换取基于所述已挖掘数据组装并存储区块的交易。此外，所述方法可包括在接收所述数字资产之前等待与最小数量的区块相关联的时间段t的要求。这提供了用于提供验证节点的激励方案，因为提供者将因提供已验证交易的列表(例如，以可逆布隆查找表的形式)用于挖掘和/或在区块链上存储已挖掘区块而获得奖励。在接收数字资产之前需要最短的时间段使得激励矿工将骨架区块(包括支付)传播到一系列节点，并且将激励节点将骨架区块传播到其他节点。

[0033] 基于已挖掘数据来组装区块的步骤可涉及组装大区块，每个区块具有例如至少2、4、6、8、10、50、100、500、1000或10000兆字节的大小。尽管上限会随着时间增加，但可以指定1PB的额定上限值。每个区块可包括例如至少5000、10000、500000、100000、500000或1000000个交易。尽管上限会随着时间增加，但可以指定每个区块10^{12}个交易的额定上限值。如前所述，本文所述的方法、节点和区块链网络架构使大区块能够被构造和存储在存储实体上，无需矿工构造和存储大量交易。这使得系统能够应对大幅提高的交易速率。

[0034] 在本文所述的方法中，可以将区块修改为包括包含由矿工提供的随机数的区块头。即，交易验证节点可以用于处理包括如下区块头的区块，所述区块头包含由矿工在从区块链网络接收已解决的交易时提供的随机数。这构成了对区块头的更改，因此矿工可以选择或随机生成插入到区块头中的数字。这有助于确保即使许多矿工选择了同一交易列表，所述矿工不会相互竞争尝试挖掘相同区块。

[0035] 用于存储前述大区块的数据的存储实体可以在所述区块链网络上的多个交易验证节点之间共享，所述多个交易验证节点形成所述区块链网络上的超级节点，其中，所述共享存储实体是公共存储节点、分布式存储或两者的组合。这种架构使得在区块链网络上形成一个超级节点，并允许提供专用存储设施来存储区块链并对区块链网络提供服务。

[0036] 鉴于上述内容，还提供了一种区块链网络的超级节点，所述超级节点包括：

[0037] 如前所述的多个验证节点;和

[0038] 用于存储所述区块链的共享存储实体,

[0039] 其中,所述共享存储实体是公共存储节点、分布式存储或两者的组合,和

[0040] 其中,由所述多个验证节点组装的区块被发送到并存储在所述共享存储实体,由此所述共享存储实体维护所述区块链。

[0041] 这种架构更适合于处理实现期望的交易速率增加所需的大区块尺寸,这是本文描述的方法和配置的目的。例如,共享存储实体可配置为具有至少100GB的存储容量,并且更优选地具有至少1、10、100或1000TB的存储容量。虽然上限会随着时间而增加,但可以指定10^6TB甚至10^6YB的额定上限值。

[0042] 就总体网络架构而言,可以提供包括多个这种超级节点的区块链网络。所述超级节点可以连接(但不重叠)在所述区块链网络上,每个所述超级节点的共享存储实体用于存储所述区块链的副本。超级节点实际上包括一组节点,这些节点形成了用作超级节点的池。为了保持区块链的分布式性质,有利地,应该有一定数量的这样的超级节点(例如,至少10、50、100或1000个,并且可选地小于100000000个)。

[0043] 本发明的实施例可以以多种形式提供。例如,可以提供一种包括计算机可执行指令的计算机可读存储介质,所述计算机可执行指令在被执行时使处理器执行本文所述的方法。还可以提供一种电子设备,所述电子设备包括:接口设备;一个或多个处理器,连接到所述接口设备;存储器,连接到所述一个或多个处理器,所述存储器上存储有计算机可执行指令,所述计算机可执行指令在被执行时使得所述一个或多个处理器执行本文所述的方法。此外,还可以提供区块链网络的验证交易节点,所述验证交易节点用于执行本文所述的方法。

[0044] 本文描述的本发明不同于下文中背景技术部分讨论的现有技术。

[0045] US2015/0287026的热钱包服务系统显然公开了接收交易、认证交易并将交易分发到区块链网络以进行挖掘。但是,US2015/0287026的热钱包服务系统不与区块链网络中其他交易验证节点一起维护已验证交易的分布式、去中心化存储。相反,US2015/0287026的热钱包服务系统利用多个服务器来发起多签名认证系统。然后,通过聚集认证因素进行验证的任何交易都被传播到虚拟货币网络中,而不是以分布式、去中心化的方式存储在热钱包服务系统中。另外,在US2015/0287026中未建议准备已验证交易的列表并创建用于数字资产换取向矿工提供已验证交易的列表的承诺交易。

[0046] CN106548349显然公开了接收交易、验证交易以及将验证交易分发到区块链网络以进行挖掘。但是,CN106548349未公开与区块链网络中的其他交易验证节点一起维护已验证交易的分布式、去中心化存储。此外,在CN106548349中未建议准备已验证交易的列表并创建用于数字资产换取向矿工提供已验证交易的列表的承诺交易。

[0047] WO2017/162904显然公开了接收交易、验证交易和分发已验证交易以并入区块链。但是,WO2017/162904未公开与区块链网络中的其他交易验证节点一起维护已验证交易的分布式、去中心化存储。此外,在WO2017/162904中未建议准备已验证交易的列表并创建用于数字资产换取向矿工提供已验证交易的列表的承诺交易。在WO2017/162904中,矿工编制了一组用于挖掘的交易。

[0048] US2016/0259937、WO2017/004527和WO2017/011601公开了挖掘节点选择一组交

易，将交易分组为原型区块，并以常规的“工作证明”方式确定随机数的值。一旦矿工找到了有效的随机数，该区块便会被其他节点验证并合并到区块链中。US2016/0259937、WO2017/004527和WO2017/011601没有公开与区块链网络中的其他交易验证节点一起维护已验证交易的分布式、去中心化存储的步骤，该步骤提供了分布式池，该池具有提供给矿工用于挖掘的已验证交易列表。同样，US2015/0310424也没有公开一个验证节点，该节点接收交易、验证交易、与其他交易验证节点维护已验证交易的分布式、去中心化存储、准备已验证交易的列表以及创建用于数字资产换取向矿工提供已验证交易的列表的承诺交易。

[0049] 鉴于上述内容，相信本发明提供了一种独特的架构和方法，用于更有效地在区块链网络中处理和存储千兆字节大小的大区块。

附图说明

[0050] 参考本文所述的实施方案，本发明的这些和其他方案将变得显而易见并得以阐明。下面仅通过示例并参考附图来描述本发明的实施例，其中：

[0051] 图1示出了区块的整体结构；

[0052] 图2以操作图的形式示出了比特币网络的修改后的架构，所述操作图示出了从用户提交交易到交易在区块链结束的步骤；

[0053] 图3示出了在存储池中等待确认的交易的总大小的示例的曲线图；

[0054] 图4示出了链接到内部去中心化存储设施的多个节点。

[0055] 图5示出了一种配置，其中每个节点都是分布式存储池和分布式存储设施的一部分；

[0056] 图6示出了一种新的节点结构如何融入比特币网络，其示出了如下网络配置，其中验证节点是存储池的成员，这些池共同构成去中心化的分布式比特币网络；

[0057] 图7示出了新节点的功能；

[0058] 图8示出了新的默克尔树结构，所述结构构成了对当前协议的修改。

[0059] 图9示出了创建布隆过滤器的工作流程；和

[0060] 图10示出了说明交易如何在可逆布隆过滤器(Invertible Bloom Filters，IBF)和可逆布隆查找表(Invertible Bloom Lookup Tables，IBLT)中编码的工作流程。

具体实施方式

[0061] 在本说明书中，描述了处理和存储大的、千兆字节大小的区块问题的解决方案。

[0062] 区块链网络节点和验证节点的类型

[0063] 区块链网络可描述为点对点开放式成员网络，任何人都可以加入该网络，而无需邀请或无需经其他成员同意。运行区块链协议(区块链网络在区块链协议下运行)实例的分布式电子设备可以参与区块链网络中。这种分布式电子设备可以称为节点。例如，区块链协议可以是例如比特币协议或其他加密货币。

[0064] 运行区块链协议并形成区块链网络的节点的电子设备可以是各种类型的，包括例如计算机(如台式计算机、笔记本电脑、平板电脑、服务器、计算机群组)、移动设备(如智能手机)、可穿戴计算机(如智能手表)、或其他电子设备。

[0065] 区块链网络的节点使用合适的通信技术彼此连接，该通信技术可以包括有线和无

线通信技术。在许多情况下,区块链网络至少部分地在互联网上实现,并且一些节点可以位于地理上分散的位置。

[0066] 当前,节点维护区块链上所有交易的全局帐本,所有交易分组到多个区块中,其中每个区块包含区块链上前一个区块的散列。全局账本是分布式账本,每个节点可以存储全局账本的完整副本或部分副本。影响全局账本的节点的交易由其他节点验证,从而保持全局账本的有效性。本领域的普通技术人员将会理解实现和操作区块链网络(例如使用比特币协议的区块链网络)的细节。

[0067] 每个交易通常具有一个或多个输入和一个或多个输出。嵌入到输入和输出中的脚本指定了如何以及谁可以访问所述交易的输出。交易的输出可以是作为交易结果的值被转移到的地址。然后,该值与该输出的地址相关联,作为未花费的交易输出(Unspent Transaction Output,简称UTXO)。随后的交易可以将该地址作为输入参考以便花费或分散该值。

[0068] 节点可以根据其功能而具有不同的类型或类别。已经提出了与节点相关联的四个基本功能:钱包、挖掘、全区块链维护和网络路由。这些功能可能有所不同。节点可具有多个功能。例如,“全节点”(“full node”)提供了所有四种功能。轻量级节点例如可以在数字钱包中实现,并且可以仅具有钱包和网络路由功能。数字钱包可保持对区块头的跟踪(区块头在查询区块时用作索引),而不是存储全区块链。节点使用面向连接的协议(例如TCP/IP(传输控制协议))相互通信。

[0069] 可提供节点的附加类型或类别:商户节点(Merchant Node)(本文有时称为“M节点”)。M节点旨在专注于交易的快速传播。M节点可存储也可不存储全区块链,并且不执行挖掘功能。从这个意义上讲,M节点与轻量级节点或钱包类似。但是,M节点包括附加功能以实现交易的快速传播。M节点的操作重点是未确认交易的快速验证和传播,特别是传播到其他M节点,未确认交易被从这些M节点快速推动到区块链网络中的其他节点。为了便于实现此功能,允许M节点进行更多数量的传入连接以及特别是传出连接,否则,这些连接可能会被允许用于管理协议下的节点。

[0070] M节点可统称为商户网络(Merchant Network)(或“M网络”)。术语“商户”可解释为“专用的”。M节点可集成到区块链网络中。每个M节点都是区块链网络上的满足特定的硬件和性能能力、确保其能够执行M节点的功能的专用节点。也就是说,M网络可视为区块链网络内的一个子网,并通过区块链网络分布。M节点可布置并配置成执行一个或多个专用功能或服务。

[0071] 为了使M网络可靠地运行,能够在一定的安全级别上提供服务,M节点需要保持对整个M网络的良好概览,因此需要建立有效的路由协议。每当M节点接收到启动交易时,其需要将交易广播给其他几个M节点以及其他节点。在M网络环境下,这相当于寻找多旅行商问题(Multiple Traveling Salesman Problem,MTSP)的解决方案。有许多解决方案可以解决多这个问题,其中任何一个方案都可运用于M网络。每个M节点都以某种最新的形式运行路由优化。

[0072] 在一些实施方式中,M网络实现为去中心化的IP多播类型的网络。也就是说,为了使传入交易能够快速扩散到区块链网络,可以使用多播来确保交易在整个M网络中快速广播,从而使得所有M节点专注于将交易转发到区块链网络中的其他节点。

[0073] 多播网络架构允许向一组目的节点同时分发数据的可能性,无需为每个对接收信息感兴趣的节点复制数据。如果节点想要接收多播传输,那么节点加入多播组(注册阶段),此后该节点将能够接收通过多播组发送的所有数据。IP多播不需要具备有多少个接收器的先验知识就能扩展到更大的接收器群,且通过仅要求源发送一次数据包,就可以有效地利用网络基础结构。对于多播网络的性质,由于与大量其他节点同时通信,因此使用面向连接的协议(如TCP)是不切实际的。据此,使用无连接协议。

[0074] 一些区块链网络(例如比特币)使用TCP进行节点对节点的通信。使用TCP发送的数据包具有相关联的序列号,该序列号用于排序。除此之外,TCP协议在建立连接和终止连接时都涉及三次握手过程。通过TCP发送的数据包有相关的开销,所述数据包具有相关联的序列号并且有一个三次握手协议。建立连接时,传输了128-136字节,而关闭连接花费160字节。因此,包传输中的握手成本高达296字节。此外,当节点接收新交易时,该节点通过包含交易散列的库存(INV)消息通知其他节点。接收INV消息的节点检查该交易的散列是否已经被看到过;如果没有看到过,则该节点将通过发送获取数据(GETDATA)消息来请求交易。交易从节点A传输到节点B所需的时间为T1＝verification+TCP(inv+getdata+tx),其中,TCP()表示由TCP握手程序引入的时间开销。

[0075] M节点可被配置为在现有协议(如比特币)授权的情况下、使用TCP与其他节点通信。然而,M节点可使用无连接协议(如用户数据报协议(User Datagram Protocol,UDP))在多播情况下进行从M节点到M节点的通信,甚至更合适地从M节点到多个M节点的通信。与TCP不同,UDP不涉及握手协议,因此M节点能够更快地传播交易。这也可以避免恶意节点不发送实际交易而是通过发送重复的INV消息来捆绑其他节点。

[0076] UDP的轻量级特性与某些权衡相关联。错误检查较少,也没有错误恢复。在一些实施方式中,可以通过将错误恢复、排序和重新传输作为应用层的功能实现来在应用级别克服UDP的这些限制。把错误检查放在应用级别消除了网络的开销。

[0077] 在一个示例情况下,区块链网络上的常规节点生成其希望通过M网络处理的交易(例如基于商户的支付)。常规节点可以将交易发送到M节点,然后该M节点使用多播将交易广播到其他M节点,或者如果常规节点知道M节点的IP多播地址,则可以将交易直接发送到多个M节点。在一些示例中,M网络的所有M节点都是单个多播地址的成员,因此发送到该地址的所有交易都被所有M节点接收。然而,在一些情况下,可能有多于一个与M网络相关联的多播地址,并且接收M节点可以从路由信息中评估是否需要将交易进一步广播到其他多播地址,以将交易传播到全M网络。

[0078] 多播有助于确保新交易快速初始传播到所有M节点;但是,多播解决方案并不一定解决由增加的交易吞吐量引起的区块链网络的可扩展性问题。网络中的每个节点通常都维护一个存储池(Mempool),该存储池包含其已经看到的、且还没有被完成工作证明的矿工合并到区块链的未确认的交易。支付处理中使用的交易数量的显著增长会增加每个存储池中存储的交易数量。因此,尽管M网络中的节点能够几乎同时接收新的交易,但是所述节点对于大型且快速变化的存储池可能具有存储能力的限制。

[0079] 为了解决这个问题,M节点可以使用通过分布式散列表(Distributed Hash Table,简称DHT)实现的共享存储池,作为使用多播的替代。

[0080] 假设交易(TX)的平均大小为500字节,交易速率为大约10^4个交易/秒,则M网络可

以接收大约400GB的每日传入数据。所有这些数据都需要在未确认交易的存储池中存储不同的时间。因此，M网络需要大量存储空间和快速存储数据的能力。为了不对每个单独的M节点提出太多的要求，M节点实现一个依赖于DHT的共享存储池。代替让每个M节点将所有传入的交易保存在自己的存储池中，每个M节点只存储总数的某一部分，以及存储其余部分的散列和相关密钥值。

[0081] DHT是一类去中心化的分布式系统，其允许在节点之间对密钥集进行成员划分，能够以有效且优化的方式仅向给定密钥的所有者发送消息。网络的每个节点都可以看作是散列表阵列的一个单元。DHT旨在管理大量节点，允许新节点加入网络，旧节点离开网络或崩溃，而不损害共享数据的完整性。DHT确保去中心化(没有中心权力，也没有中心协调)、可扩展性(系统具有数百万节点的高效行为)和容错性(系统可靠，能够管理加入和离开网络或崩溃的节点)。网络中的每个节点可以只与少数其他节点保持联系，因此在出现变化或新数据时，网络不会过载。

[0082] 这种概念同样可应用于UTXO数据库，该数据库包含区块链上所有未花费输出的集合。可以使用DHT构建UTXO数据库，以便在一组节点之间共享内容。

[0083] 许多可能的DHT结构和协议可用于实现M网络的共享存储池。一个例子是Pastry™，此外还有许多其他的例子。Pastry™是一种旨在维护覆盖网络的协议，该网络能够在分布式系统上存储和传输信息。Pastry™网络中的每个节点都分配有128位标识符，该标识符用于指示节点在圆形节点标识符(ID)空间中的位置(范围从0到$2^{128}-1$)。当节点加入网络时，随机分配该ID。每个节点维护路由表、邻域集和叶集。

[0084] 在确定鲁棒的DHT的维数时，需要考虑的一个因素是确保整个网络的鲁棒性和可靠性所需的副本数量。如前所述，节点可以加入和离开网络，这不应影响数据的可用性。如果存储交易A的节点离开网络，则有必要在网络的另一部分中找到交易A。在现有的区块链网络中(例如比特币)，网络的区块链副本的数量等于网络中全部节点的数量(平均5000个副本)，但这影响了可扩展性。

[0085] 在一个M网络配置中，存储池不是在每个M节点上完全复制的，而是通过DHT来实现。为了提供可靠性，可将DHT实现为具有一些重叠；即每个交易数据项都在多于一个M节点中复制，尽管不是在每个M节点中都复制。例如，可将DHT实现为指定最少数量的2个副本。同时假设两个节点之间的完全独立性为$(\frac{1}{(24*365)})^2 = 1.30 * 10^{-8}$，这将可能导致两个节点在任何给定时间内同时停机。

[0086] 因此，用于在分布式存储池中存储新交易的过程可包括以下步骤，其中使用DHT实现分布式存储池。该过程包括：节点将交易发送到M节点。M节点根据所述实现来散列交易或交易ID以获得密钥值。密钥值指示该M节点或多个M节点(在复制数据的情况下)，交易存储在所述M节点或多个M节点。然后，M节点将交易存储在分布式存储池中，其可包括：基于密钥值和M网络中M节点的分配ID，将交易路由到要存储交易的正确的M节点。根据所涉及的DHT协议，M节点可能会接收到确认。当M节点从常规节点接收新交易时，M节点可以执行某些验证操作以验证交易的真实性。

[0087] 可以将交易散列以生成用于交易的密钥。密钥可以指示交易应存储在DHT中的哪个位置，其可以在当前M节点以外的其他节点上。然后，M节点评估交易是否已在运行的DHT

中。基于组成M网络的M节点之间的密钥空间划分，每个M节点具有一部分存储的交易。在一些配置中，密钥空间在参与的M节点之间划分。划分可能涉及重叠，以引起网络弹性的复制。在一些实施方式中(例如使用Pastry™)，每个M节点都被指派了唯一的密钥或ID号码，交易可以基于交易密钥值的接近度而存储在所述M节点或多个M节点(在需要复制的情况下)。M节点可以在本地具有交易的存储部分以及其余部分的散列或密钥值。因此，M节点能够基于运行中的本地数据来评估新交易是否在DHT中。

[0088] 如果交易不在DHT中，那么在运行中，M节点根据其密钥值将交易存储在DHT中。一般来说，这可以采取put(k,tx)操作的形式，其中k是密钥值，tx是交易。适用的DHT路由协议确保交易发送到并存储在适当的M节点。基于所选择的实施方式，DHT可以根据用于分布式散列表的各种协议来操作。使用DHT在M网络中存储交易避免了在M网络中使用库存/获取数据(INV/GETDATA)消息来将交易路由到每个M节点。

[0089] 在本示例的操作中，M节点可根据区块链网络的正常交易转发协议将交易发送到区块链网络中的常规节点。例如，到普通节点的通信可以使用TCP进行节点对节点的连接。

[0090] 在一种配置中，M节点包括处理器、网络接口、和存储器。可以使用任何合适的计算硬件来实现所述M节点，所述计算硬件具有网络连接性和足够的处理与存储资源来执行本文描述的功能。所述M节点可包括处理器可执行指令以实现本文描述的功能。在一些情况下，处理器可执行指令可称为区块链商户节点应用，但可以理解，基于硬件和操作系统，指令可以在一个或多个模块、应用程序、脚本、或其他编程结构中实现。处理器可包括多核处理器和/或多个处理器。

[0091] 存储器部分地基于其DHT密钥值(即M节点ID)存储数据，包括基于DHT的存储池的指派部分。在此示例实施方式中，存储器还存储路由表、邻域集和叶集。路由表包含M网络内特定路由目的地的列表，当节点接收到数据包时，会参考路由表以了解将数据发送到何处。路由表还可包含有关每个目的地与M节点间距离的信息。邻域集(例如基于邻近度量(ping延迟))包含关于接近的M节点的信息。叶集包含数字上接近的M节点。如果M节点的密钥值(节点ID)在数字上接近，那么所述节点在数字上接近。存储器还包括M节点信誉表，这将在下文中进一步说明。

[0092] 为了提供可扩展性，除了使用DHT实现存储池之外，M网络还允许节点加入M网络。新节点需要具有至少一个已经是M网络一部分的M节点的地址，以便该新节点可以将其加入请求定向到其中一个M节点。M节点可以执行某些验证动作，这可能涉及查询新节点。例如，M网络可以具有其指定给M节点的、与加入M网络相关联的一组最低标准。举例来说，该标准可包括可用的最小处理资源、可用的最小空闲内存或连接性要求。

[0093] 假设M节点完成为检验新节点而执行的任何验证操作，那么M节点根据控制DHT操作的任何DHT协议，将加入请求(Joinrequest())转发到DHT。然后，DHT与新节点通信，以向新节点提供路由表、密钥值(节点ID)和任何其他数据，以使新节点能够充当M网络上的新M节点。

[0094] 应当理解，节点能够轻易加入M网络造成了恶意节点也容易加入网络。为了识别和隔离潜在的恶意节点，一种配置提供M节点存储M节点信誉表，用于跟踪和更新节点行为排名。当新节点加入网络时，该节点可以被添加到M节点信誉表中，如节点ID字段所示。在一些实施方式中，该表还可以包括加入时间。该表还包括该M节点的分数或评级。

[0095] 分数可以根据某些行为指标向上或向下调整。例如，如果M节点未能转发交易，保持一段时间的静默，用被确定为非交易性的流量淹没M网络，或者以其他方式参与负面行为，那么该M节点的排名可能被降低或递减。如果节点的分数低于预设的最小值，那么可以将该节点从M网络中排除。

[0096] 在特定M节点维护的M节点信誉表可仅限于跟踪其邻居的分数，而不是全M网络。因此，当新M节点在t时刻加入网络时，其邻居的M节点信誉表不包含有关新节点的任何信息，但是从t时刻开始，他们开始建立新节点的信誉，将信息存储在节点注册表中。例如，如果新节点是静默节点，这意味着该节点不传输其通过网络接收到的信息，那么所有邻居都开始在其各自的M节点信誉表中记录此行为(例如给新节点的ID分配负值)。在一定时间t+n之后，如果知道新节点的所有节点的M节点信誉表都包含负值，那么节点可决定隔离新节点并将新节点从网络禁止。

[0097] M网络的分布式存储池中的交易可能需要等待相当长的时间才能被确认，即在被合并到添加至区块链并被确认的区块中之前。当足够数量的后续区块添加到其上方的区块链中时，该区块被视为“已确认”，这样，扭转链中的增长并删除区块以更改为不同的分支或分叉在计算上不可行。

[0098] 由于存储池的大小和灵活性以及交易量，给定交易未被确认的时间可能长于某些区块链实施方式(例如比特币)中的。在传统的比特币实施方式中，一旦将交易合并到区块中，该交易便从存储池中删除。这意味着，如果该区块最终成为孤立块，那么该区块中的所有交易都会在网络上重新传输。在快速交易网络的情况下，这可能不切实际，或者可能导致对某些交易的确认造成长时间延迟。

[0099] 因此，在一些实施方式中，存储池可以跟踪已并入交易的区块的确认数量，即在并入了交易的区块之后添加到区块链的区块的数量。只有在确认的预定数量发生后，交易才会从存储池中删除。预定数量可以是4、5、6、7或对于给定实施方式的任何合适的数量。存储池数据条目可构建为包括交易ID字段、交易字段和确认数量(NoC)字段。在另一实施方式中，存储池数据条目可以简单地记录区块数量，而不跟踪NoC。基于区块链的当前区块数量，该存储池数据条目可以根据区块数量评估发生了多少次确认。

[0100] 一旦发生必要数量的确认，交易就可以安全地从存储池中删除。这样，在孤立区块的情况下不会有交易损失，并且在必要数量的确认之后，交易将被永久删除。

[0101] 本说明书以下部分中描述的解决方案利用了如前所述的修改后类型的快速验证节点。描述了一种新的全节点配置，该配置实际上是以大规模存储能力和修改后的操作协议增强的M节点验证体系结构。M节点和存储节点共同构成了新的全节点的核心。详细描述了新的节点结构，包括必要的技术要求和技术解决方案，并提供了可持续激励模型。

[0102] 区块大小和存储要求

[0103] 当前区块大小为1Mb。当前，区块由包含所谓的幻数(总是相同的值)的字段、表明区块实际大小的值、所谓的区块头、区块中包含的交易数量、以及实际交易列表组成。实际交易列表总是从币库交易开始，该交易包含对于挖掘区块的奖励。在图1中，示出了区块的整体结构。

[0104] 区块头包含以下内容:

[0105] 1.版本号(4字节)

[0106] 2.前一个区块头的散列(32字节)

[0107] 3.默克尔根散列(Merkle root hash)(32字节)

[0108] 4.时间(4字节)

[0109] 5.目标阈值(编码为nBits-4字节)

[0110] 6.只被使用一次的非重复的随机数(4字节)

[0111] 当前,区块包含大约2000个交易,并且大约每10分钟挖掘一次(10分钟的区块时间被设置为第一确认时间和在链拆分上浪费的时间之间的折衷)。这提供了大约每秒3.5个交易的交易速率,理论上每秒最多7个交易。相比之下,VISA以大约每秒10000个交易的速率运行,并且能够达到每秒50000多个交易。

[0112] 显然,为了建立有竞争力的支付系统,有必要对当前的限制条件进行某种规避。由于已经确定10分钟的区块时间,因此必须考虑区块大小的变化,进而考虑区块链本身的变化。在本说明书中,描述了一种可扩展的解决方案,该解决方案能够处理每秒大约例如50000个交易。

[0113] 增加当前区块的大小、甚至完全取消限制,是一个备受讨论的话题,有时也是有争议的话题。双方似乎都有强有力的论据,因为保持现有大小和增加大小都有明显的好处和权衡。

[0114] 假设交易速率为r,我们可以计算出必要的区块大小。下面假设区块时间(平均)为10分钟。因此,假设每个区块的交易数量为T(r)。我们得到

[0115] $T(r) = r \cdot 6 \cdot 10^2 \ \text{block}^{-1}$

[0116] 如果s_{Tx}是平均交易大小(以字节为单位),则区块大小B(r,s_{Tx})可以表示为

[0117] $B(r, s_{Tx}) = s_{Tx} \cdot T(r) = s_{Tx} \cdot r \cdot 6 \cdot 10^2$

[0118] 因此,考虑到r=50000个交易/秒和s_{Tx}=500字节的情况,快速回溯计算得出:

[0119] $$T(50000) = 5 \cdot 10^4 \frac{\text{Tx}}{s} \cdot 6 \cdot 10^2 \frac{s}{\text{block}} = 3 \cdot 10^7 \frac{\text{Tx}}{\text{block}}$$

[0120] $$B(50000,500) = 3 \cdot 10^7 \frac{\text{Tx}}{\text{block}} \cdot 5 \cdot 10^2 \frac{\text{bytes}}{\text{Tx}} = 15\,\text{Gb/block}$$

[0121] 这反过来又导致了$O(10^6)$Gb/year的存储需求。很明显,对于这种大小的区块,我们需要稍微不同的区块传播和存储方法。下表1示出了交易速率、平均交易大小、区块大小以及每月和每年所需存储空间量之间的关系。

[0122]

速率（交易/秒）	平均交易大小（字节）	交易/区块	区块大小 （Mb）	存储 （Gb/月）	存储 （Tb/年）
10	250	6000	1.50	6.6	0.1
10	500	6000	3.00	13.1	0.2
10	1000	6000	6.00	26.3	0.3
100	250	60000	15.00	65.7	0.8
100	500	60000	30.00	131.4	1.6
100	1000	60000	60.00	262.8	3.2
1000	250	600000	150.00	657.0	7.9
1000	500	600000	300.00	1314.0	15.8
1000	1000	600000	600.00	2628.0	31.5
10000	250	6000000	1500.00	6570.0	78.8
10000	500	6000000	3000.00	13140.0	157.7
10000	1000	6000000	6000.00	26280.0	315.4
100000	250	60000000	15000.00	65700.0	788.4
100000	500	60000000	30000.00	131400.0	1576.8
100000	1000	60000000	60000.00	262800.0	3153.6

[0123] 表1:交易速率、平均交易大小、区块大小以及每月和每年所需存储空间量之间的关系

[0124] 新的比特币网络

[0125] 我们为比特币网络提出的架构如图2所示,图2示出了从用户提交交易的时刻到交易在区块链结束的操作示意图。

[0126] 提供了一种系统,其中特殊验证节点(通过分布式散列表DHT维护特殊验证节点之间的共享存储池)接收交易,验证交易,并在存储池分配交易。验证节点然后向矿工提供服务,即提供有效交易散列的列表。矿工根据这些散列来组装预区块(区块骨架)(Block Skeletons),并尝试解决散列难题。在找到难题的解决方案时,获胜的矿工将区块骨架发送回验证节点。该验证节点验证区块并确保区块被存储。最初,验证节点存储区块本身是可能且可行的。当区块大小最终超过某个大小阈值时,验证节点将:a)扩展自身的存储能力;或b)将存储外包给专门的存储节点。本说明书下文将讨论这两种体系结构。

[0127] 新的全节点

[0128] 按照顺序$O(10)\ GB$的区块大小,依靠PC型节点来提供用于托管区块链全图像的存储能力似乎不再可行。相反,需要提供$O(1)\ PB$或更多存储的设备(见表1)。挑战就变成了创建一个既能容纳新区块又能保持网络的分布式、去中心化和不信任性的系统。

[0129] 设想了两种类型的全节点结构,以及这两种类型的组合:

[0130] 1.带有相关联的千兆字节存储机架的验证节点

[0131] 2.基于内部去中心化的分布式点对点(P2P)单节点网络(非常类似于当前的比特

币网络本身)的带有关联存储池验证节点

[0132] 3.1和2的组合

[0133] 所提出的解决方案试图通过引入类似于在当今比特币网络上运行的所谓全节点、但是相比之下具有随着区块的大小和交易数量的增加而扩展的能力的节点,来解决一直保持区块链的分布式和去中心化的记录的问题。

[0134] 区别不仅限于纯粹的结构和硬件相关问题。与撰写本文时运行的基于家用电脑的全节点相比,本文提出的新节点将是专用节点。所述新节点需要大量的投资,因此,激励措施将大不相同。在可扩展的范例中,M节点(验证节点)和新的全节点(验证和存储节点的组合)都将期望对其服务进行补偿。

[0135] 在谱的另一方面,我们拥有去中心化的分布式存储解决方案,这些解决方案主要由单个节点组成。Storj(Wilkinson et al.,2016)、Sia(NebulousLabs)和MaidSafe(maidsafe)是很好的例子。就Storj而言,其功能基于参与者因提供存储空间而获得奖励。

[0136] 如上所述,也可以想到由千兆兆字节(Pb)机架和点对点(P2P)存储系统组成的超级节点。

[0137] 因为比特币生态系统严重依赖于以去中心化方式分布的整个区块链的多个副本的存在,所以显然,对所有全节点进行补偿是很重要的。这与挖掘非常不同,挖掘本质上是一种由赢家获得全部奖金的游戏。因为矿工将依靠他们(赢得的)区块最终进入公共区块链,所以奖励存储全节点符合他们的利益。

[0138] 节点将组成为池,这些池充当超级节点。为了保持区块链的分布式性质,必须有一定数量的此类超级节点(≥100)。超级节点连接但不重叠。

[0139] 技术要求

[0140] 如上所述,在讨论新的全节点时,需要考虑两种完全不同的体系结构(见表2)。

[0141] 新的全节点需要维护两种类型的存储设备:

[0142] 1)存储池的类似随机存取存储器(RAM)的分布式散列表(DHT)存储器/存储设备。

[0143] 2)区块链的永久性磁带/磁盘式存储设备。

[0144] 如上所述,对于r=50000个交易/秒的交易速率,区块预计为$O(10)\ Gb$,这意味着年存储需求为~365×24×6×15Gb=7.9·10^5Gb=0.8Pb/yr(见表1)。

[0145] 表2示出了当前全节点和未来全节点之间的比较:

[0146]

特征	当前全节点	新的全节点
存储池	是	是
存储池尺寸	~10-100Mb	~0.1-1Tb
存储池类型	具有收费下限的RAM	具有“无限”存储的DHT
磁盘空间/年	~100Gb	~1Pb
交易验证	是	是
区块验证	是	是

[0147] 同时,机架/集群需要维护存储池。这样可以快速恢复区块。存储池的必要大小更难评估。当前,区块大小约为1兆字节($\backsim 1Mb$),每秒约有4个交易($\backsim 4\ Tx/s$),在存储池中等待的交易总大小在2兆字节到约70兆字节($\backsim 70Mb$)之间波动。图3示出了在存储池

中等待确认的交易的总大小的示例图。

[0148] 如上所述，我们设想了两种根本上不同的、能够存储大量数据的结构，以及两种结构的组合。图4和图5中示出了这两种结构。图4示出了包括能够访问内部去中心化存储设施的多个节点的配置。图5示出了一种配置，其中每个节点都是分布式存储池和分布式存储设施的一部分。图4所描述的架构构似乎适合拥有并维护多个验证节点的较大实体，这些节点都可以访问实体自身的存储设施。相比之下，图5中描述的架构是完全去中心化的。这一解决方案适用于希望加入共享的分布式存储池的单个节点(例如具有足够存储容量的家用PC)。已经存在用于此目的的底层存储技术(例如Storj、Sia、MaidSafe)。

[0149] 图6示出了一种将新的全节点如何适应比特币网络进行可视化的方法，图6显示了一种网络配置，其中验证节点是存储池的成员。这些池共同构成一个去中心化的分布式比特币网络。

[0150] 全节点操作

[0151] 在大区块场景中，我们将面临另一种情况，而不仅仅是由于空间需求。存储池应能够容纳相当于一个区块的量，即大约15千兆字节(～15Gb)，最好是与下一个要挖掘的区块等量。这也将与需要考虑的开销相结合。

[0152] 1)存储池需要与其他验证节点同步。这涉及交换可逆布隆过滤器查找表(Michael T.Goodrich，2011年)

[0153] 2)需要仔细检查IBLT，并检索丢失的交易(Tx)

[0154] 3)需要验证额外检索的交易

[0155] 4)基于从矿工或其他完整节点接收到的区块骨架来组装区块

[0156] 新的全节点保持最新的存储池。它通过与矿工交换的IBLT和其他验证以及新的全节点来实现。

[0157] 矿工发送包含以下内容的区块骨架(元组(Tuple))：

[0158] 1.只被使用一次的非重复的随机数，n

[0159] 2.IBLT

[0160] 3.币库交易

[0161] 基于此，新的全节点相应地对交易进行排序(根据一组特定的规则)，并组装新挖掘的区块。新的全节点继续将区块存储在自己的存储设备中，并将骨架传播到其他新的全节点。本说明书的下文将更详细地描述协议。

[0162] 激励

[0163] 特定配置的一个重要特征是将激励内置到系统中，以激励提供新的节点结构和服务。由于存储区块链需要大量成本，因此需要激励措施。图7示出了新的全节点的功能。新的全节点因两种类型的服务而获得奖励，主要是：

[0164] 1)编制已验证交易的列表，准备挖掘。将交易的散列值(默克尔根)发送给矿工，矿工选择列表并挖掘区块。

[0165] 2)获胜的矿工将已挖掘的区块的骨架发送给几个新的全节点。所述骨架包括币库交易，其中包含：

[0166] a.挖掘奖励。

[0167] b.作为承诺方案一部分的秘密，该方案用作提供经过验证列表的支付机制。

[0168] c.支付区块验证和/或在区块链上存储区块的费用。

[0169] [交易]验证节点将通过收费系统补偿交易验证。接收验证/新的全节点将因以下一项或多项而获得奖励：

[0170] 1)向矿工提供已验证交易(Tx)散列的列表(见上文b)

[0171] 2)从区块骨架重新组装区块(“固定”费用)。

[0172] 3)区块的大小(“每兆字节存储”付款)

[0173] 激励在于100个区块确认时间T_{100}。

[0174] 1)矿工需要等待时间t约为T_{100},以领取奖励。

[0175] 2)验证节点需要等待时间t约为T_{100},才能收到验证区块中交易的费用。

[0176] 3)新的全节点需要等待时间t约为T_{100},才能收到区块组装费和取决于大小的存储费用。

[0177] 因此,100个区块的确认时间将为矿工提供必要的激励,以将骨架区块(包括支付)传播到一系列新的全节点,并且激励新的全节点将骨架区块传播到其他新的全节点。

[0178] 还应指出,矿工可以自由选择他们希望包含在区块中的(交易)列表。因此,我们设想了一个由以下验证节点组成的市场,该验证节点通过编制矿工可以通过承诺交易从中选择和购买的、已验证交易的列表来进行竞争。

[0179] 重访挖掘

[0180] 比特币生态系统依赖于挖掘过程。矿工从存储池(或者如本文所设想的,从专门的验证节点)中收集交易(Tx),将交易组织为区块,并尝试找到解决散列难题的方案(只被使用一次的非重复的随机数)。区块头包含矿工包含的只被使用一次的非重复的随机数、区块链上前一个区块的散列、交易的默克尔树的根。该难题的解决包括计算与前一个区块散列和默克尔根串联的只被使用一次的非重复的随机数(迭代选择)的双SHA256散列,检查其是否小于所谓的难度目标。如果小于难度目标,则难题得以解决;如果大于难度目标,则只被使用一次的非重复的随机数的迭代继续。这在新的范例中保持不变。构成挑战的是极大地扩大的区块大小和已挖掘区块在整个网络中的分布。对于千兆字节大小的区块,通过网络广播整个区块不一定可行。

[0181] 相反,我们提出一种解决方案,包括以下步骤：

[0182] 1.矿工从验证/M节点和/或新的全节点接收已验证交易的列表。

[0183] 2.矿工本身可以或可以不操作自己的交易散列值存储池,该存储池遵循特定排序惯例。

[0184] [https://www.cryptocoinsnews.com/bitcoin-in-bloom-how-iblts-allow-bitcoin-scale/]中提供了此类排序的示例。

[0185] 3.矿工通过确定只被使用一次的非重复的随机数n来解决散列难题。

[0186] 4.接下来,计算散列树(默克尔树,下文称为HT),存储默克尔树的根(见下一节)。

[0187] 5.交易列表用于创建IBLT。IBLT可用于计算两个集合(例如存储池)之间的内容差异,以及协调两个集合。

[0188] 6.元组{n;IBLT;CoinBase Tx;HT root}被广播到验证/M节点。

[0189] 7.新的全节点为区块链的存储池和存储设备操作DHT。

[0190] 8.池基于元组{n;IBLT;CoinBase Tx;HT root}重新组装区块,并通过a)自身存储

区块或b)通过存储在专用存储节点上,将区块记录在区块链上。

[0191] 避免矿工之间的竞争

[0192] 矿工能够从由若干个验证节点组成的市场中选择已验证交易的列表。除非另有说明,否则假设矿工将选择最大化其潜在收入的列表是合理的。细心的读者可能会指出,这可能导致矿工主要从同一节点选择同一列表。反过来,这会导致一些矿工相互竞争,试图挖掘同一区块的情况。这有利于具有最大散列能力的矿工。

[0193] 我们提出在区块头中添加一个额外字段。该字段包含每个矿工选择的随机数。这保证了每个矿工从不同的起点开始,因此防止将解决区块仅归结为散列能力。这又将模拟现在的情况,即矿工倾向于挖掘相似但独立选择且略有不同的区块。

[0194] 协议

[0195] 在此,我们对操作新的全节点所必需的协议进行说明。

[0196] 为了使所提出的系统发挥作用,所涉及的节点(验证者、矿工、新的全节点……)的存储池应遵循交易的排序惯例。在此,我们提出使用加文·安德烈森(GavinAndresen)提出的规范顺序。在该排序中,排序与区块中的交易列表有关,但是在此,我们提出了一个想法:所有验证节点和新的全节点都对其存储池使用相同的约定。

[0197] 约定可以总结如下:

[0198] 1)按照相对于前一个交易散列的升序对交易进行排序。

[0199] 2)从排序列表中添加不依赖于后续交易的第一个交易。

[0200] 如前所述,区块包含所谓的默克尔根散列。它是通过散列所有交易(包括币库交易),然后散列串接的散列,直到达到默克尔根散列而产生的。显然,如果不是针对矿工正在制造币库交易的事实,验证节点可以计算整个默克尔树,从而计算默克尔根和相应的散列。

[0201] 在此,我们提出通过以下方式的程序来计算默克尔树:

[0202] 验证者节点计算小默克尔根(Little Merkle Root)。该程序与计算标准默克尔根时的程序相同,但有几处例外:

[0203] 1)排出币库交易。

[0204] 2)包括所谓的承诺交易。

[0205] 3)矿工生成币库交易,并将其与小默克尔根散列串联起来,产生默克尔根散列。

[0206] 图8示出了新的默克尔树结构。注意,所述结构构成了对当前协议的修改。

[0207] 修改区块头

[0208] 如上所述,我们提出在区块头中添加一个额外字段,其中包含矿工选择的随机数。因此,散列难题的解决变化如下:

[0209] $$SHA256[SHA256[\text{Prev. Block Hash} \parallel \text{Merkle Root} \parallel \text{nonce}]]$$

$$SHA256[SHA256[\text{RND} \parallel \text{Prev. Block Hash} \parallel \text{Merkle Root} \parallel \text{nonce}]]$$

[0210] 因此,我们提出用包含随机数的额外字段来增强已挖掘的新区块的区块头。区块头将包含以下内容:

[0211] 1.版本号(4字节)

[0212] 2.前一个区块头的散列(32字节)

[0213] 3.默克尔根散列(Merkle Root Hash)(32字节)

[0214] 4.时间(4字节)

[0215] 5.目标阈值(编码为nBits-4字节)

[0216] 6.只被使用一次的非重复的随机数(4字节)

[0217] 7.随机数(4字节)

[0218] 验证→矿工

[0219] 验证节点：

[0220] o根据请求,验证节点(可能是也可能不是新的全节点)准备要挖掘的已验证交易的列表。

[0221] o验证者节点创建承诺交易。

[0222] o计算所谓的小默克尔根(Little Merkle Root)(见前面一部分),其中包括承诺交易。

[0223] o验证者节点为以下准备两个IBLT：

[0224] 1)为区块中的所有交易(IBLT1);和

[0225] 2)为区块中所有对应的交易标识符TxID(IBLT2)

[0226] o验证者节点向矿工发送：

[0227] 1)小默克尔根。

[0228] 2)IBLT1。

[0229] 3)IBLT2(可选——仅当矿工使用自己的TxID-/存储池运行时)。

[0230] 4)前一区块散列。

[0231] 5)上述散列校验和。

[0232] 矿工：

[0233] o从验证者节点接收数据后,矿工继续创建币库交易,其中包括挖掘奖励以及矿工希望将挖掘的区块发送到其的新的全节点的区块验证/存储的奖励。此外,币库交易包含带有秘密的输出字段,该秘密字段与承诺交易中的秘密匹配。

[0234] o矿工使用从验证者节点收到的小默克尔根,并将其与币库交易结合起来以创建默克尔根散列。

[0235] o矿工现在掌握了开始解决散列难题所需的所有信息。

[0236] o按照前述线进行挖掘。

[0237] 矿工→新的全节点

[0238] 矿工：

[0239] o挖掘一个区块后,矿工将以下内容发送到新的全节点列表：

[0240] o只被使用一次的非重复的随机数(难题的解决方案),n

[0241] o币库交易

[0242] o区块头

[0243] o默克尔根

[0244] o小默克尔根

[0245] oIBLT1

[0246] oIBLT2(可选)

[0247] o散列校验和。

[0248] 新的全节点：

[0249] o检查币库交易中是否包含适当的奖励。

[0250] o节点通过计算校验和(散列)来检查接收到的数据是否一致。

[0251] o节点使用IBLT1来确保区块中的交易存在于存储池中。

[0252] o通过使用IBLT1查询存储池，节点组装区块。然后存储该区块(见有关新的全节点和存储的部分)。

[0253] o从矿工收到的数据将广播到其他新的全节点。

[0254] 定制交易列表

[0255] 我们设想这样一种情况，即已验证交易的市场将适应矿工的需求。矿工倾向于选择最大化其潜力的列表，而验证M节点将顺应这类趋势。

[0256] 在某些情况下，矿工可能希望通过合并两个或多个列表中的交易来定制其区块。通过计算两个IBLT之间的差异，可以在两个集合之间进行集合调和(set reconciliation)。然后，矿工将包含差异的IBLT发送回其中一个提供节点，以此方式检索制作列表所需的信息，该列表包含两个列表中的所有交易。

[0257] 显然，如果矿工想要根据几份列表编制自己的列表，就会带来额外的挑战。本文简要地介绍一下各个要点。

[0258] 如果矿工要合并来自各个验证节点的列表，不清楚应该如何合并默克尔根。在此，我们提出以下建议：

[0259] 构建多个单个小默克尔根的大小默克尔根(Big Little Merkle Root)；和

[0260] 将大小默克尔根与币库交易相结合。

[0261] 额外费用与添加到列表/区块的额外交易的量不成比例。由于可以合理假设各种存储池大量重叠，因此合并列表等于从另一个列表中添加了少量(相对而言)交易。然而，为了合并列表，矿工必须从每个验证节点“购买”全列表(通过承诺交易)。这是否是一种对于矿工有利可图的方法还有待观察。

[0262] 合并来自多个验证节点的列表需要矿工和每个提供验证节点之间的承诺。可以想到，矿工可能会滥用该系统。目前，尚无规则/协议强制所有承诺交易都将在区块中结束。一种可能是验证节点可以检查每个区块并否决包含其交易的那些区块。

[0263] 总结

[0264] 就计算工作而言，如今的比特币网络主要集中在挖掘方面。随着交易量的大幅增加，集中在挖掘方面不一定可行。本说明书中描述的解决方案将各种任务降级到相应的专用节点，矿工本身也变得更加专用。编制已验证交易的列表、基于区块骨架重新构建区块、以及存储都是需要大量资源的功能。因此，预计比特币网络的结构将发生变化，激励机制也会随之改变。我们已经在本说明书中详细描述了这些问题。

[0265] 在本说明书介绍的新元素中，我们可以提及：

[0266] o新型的节点结构，本文称为新的全节点或超级节点，可能是也可能不是验证M节点的扩展。

[0267] o节点根据协议进行操作，该协议有效地允许从验证节点到矿工以及从矿工到新的全节点的千兆字节大小的区块的广播。

[0268] o用于存储区块链的两个整体存储结构，可能是也可能不是所提出的新的全节点

的一部分。

[0269] o激励模型，其允许建立已验证交易的预区块列表市场，在挖掘后进行区块组装和存储。

[0270] o新的默克尔树结构，使得矿工无需维护自己的存储池。

[0271] o在区块头中添加带有由矿工选择的随机数的额外字段，以避免挖掘行为变成纯粹基于散列能力的竞争。

[0272] o通过特殊承诺交易来奖励验证。

[0273] 布隆过滤器和可逆布隆查找表(IBLT)

[0274] 在本节中，我们总结了所谓的布隆过滤器的属性以及对这些过滤器的扩展，称为可逆布隆查找表。

[0275] 最简单的形式是布隆过滤器是一个阵列。该阵列有两个与其相关联的参数M和k。M是该阵列中的位数，而k是不同散列函数H_k的数量，使得

[0276] $$H_i:S_{16}\rightarrow\{0;M-1\},\forall\,1\leq i\leq k$$

[0277] 其中S_{16}是十六进制字符串的空间，散列函数作用于该空间。为了确定交易Tx_0是否属于布隆过滤器所针对的集合，我们需要计算$H_1(Tx_0)\cdots H_k(Tx_0)$，然后检查阵列中相应的位是否被设置为1。图9示出了创建布隆过滤器的工作流程。

[0278] 如果一个或多个为否，则Tx_0绝对不在查询的集合中。但是，布隆过滤器确实允许误报。这是因为散列函数将位更改为1的概率为p＝1/|size of array|＝1/M，其中，“size of array”为阵列大小。因此，没有使用所谓的如下给定散列函数将位设置为1：

[0279] $$1-p=1-\frac{1}{M}$$

[0280] 因此，如果有k个散列函数，则给定位不被设置为1的概率为

[0281] $$\bar{P}=(1-p)^k=(1-\frac{1}{M})^k$$

[0282] 如果需要插入n个元素，则所述概率变成

[0283] $$\bar{P}_n=(1-p)^{kn}=(1-\frac{1}{M})^{kn}$$

[0284] 布隆过滤器的一个明显缺点是它既不跟踪也不保持任何特定的顺序。很显然，如果我们希望维护要过滤的项目的索引，则需要扩展过滤器的功能。这就是可逆布隆过滤器(IBF)和可逆布隆查找表(IBLT)的作用。

[0285] 不同于只激活阵列中的位，密钥的异或和、散列值(像前面一样)和总计数器被存储在IBF的每个字段中。该过程如图10所示，图10示出了说明交易如何在IBF/IBLT中编码的工作流程。

[0286] 应用

[0287] 我们假设有两个节点N_1和N_2，分别维护存储池m_1和m_2。每个存储池包含来自整个十六进制字符串S_{16}的元素。我们进一步假设存储池遵循由安德烈森(Andresen)提出并且已在本说明书前文概述的排序惯例。

[0288] 现在，N_1发送m_1给N_2·，N_2·现在可以通过两种方式处理集合调和：

[0289] (1)通过$\Delta m=m_2-m_1$计算集合差(参见(David Eppstein,2011)，(Michael T.Goodrich,2011))

[0290] (2)迭代m_2中的交易,检查所述交易是否存在于N_1的存储池中

[0291] 我们看出IBLT至少可以用于两个目的:

[0292] 1)让节点根据其存储池中已经存在的交易组装已挖掘的区块,并帮助识别和检索其没有的区块。

[0293] 2)在属于不同节点的存储池之间保持一定级别的同步。

[0294] 应当理解,虽然交易可以转移比特币,但是用户可以改为使用本文所述的方法和系统来交换其他资源(例如信息、合同和代币)。代币根据与代币相关联的智能合约来表示资产或资源,使得代币的控制有能力控制资产或资源。智能合约本身可以存储在区块链外,也可以存储在一个或多个交易中。

[0295] 参考资料

[0296] An Integrated World.(n.d.).Retrieved from https://www.anintegratedworld.com/whats-in-a-block/

[0297] David Eppstein,M.T.(2011).What's the Difference?Efficient Set Reconciliation without Prior Context.ACM.

[0298] maidsafe.(n.d.).Retrieved from github.com:https://github.com/maidsafe/Whitepapers

[0299] Michael T.Goodrich,M.M.(2011).Invertible Bloom Lookup Tables.Communication,Control,and Computing(Allerton),2011 49th Annual Allerton Conference on.

[0300] NebulousLabs.(n.d.).Retrieved from github.com:https://github.com/NebulousLabs/Sia

[0301] O(1)Block Propagation.(n.d.).Retrieved from github.com:https://gist.github.com/gavinandresen/e20c3b5a1d4b97f79ac2

[0302] Wikipedia.(n.d.).Retrieved from https://en.wikipedia.org/wiki/Distributed_hash_table

[0303] Wilkinson et al.(2016,December 15).Retrieved from https://storj.io/storj.pdf

[0304] 应当说明的是,上述实施例说明而非限制本发明,在不脱离本发明的由所附权利要求限定的范围的情况下,本领域技术人员将能够设计出许多替代性实施例。在权利要求中,括号中的任何附图标记不应解释为对权利要求的限制。词语“包括”等并非在整体上排除其他元件和步骤的存在,尽管这些元件和步骤并没有在任何权利要求或说明书中列出。在本说明书中,“包括”意指“包括或由.....组成”。元件的单数引用不意味着排除这些元件的复数引用,反之亦然。本发明可以借助包括若干不同元件的硬件,以及借助适当编程的计算机来实施。在列举了若干装置的设备权利要求中,这些装置中的若干个可以由硬件的同一个部件来体现。不争的事实是,在相互不同的从属权利要求中列举了某些方法,并不代表这些方法的结合不能获得有益效果。

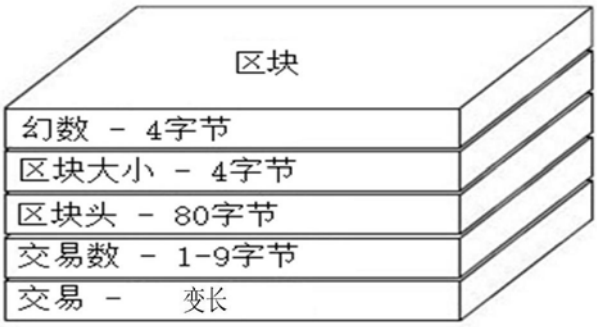

区块
幻数 - 4字节
区块大小 - 4字节
区块头 - 80字节
交易数 - 1-9字节
交易 - 变长

图1

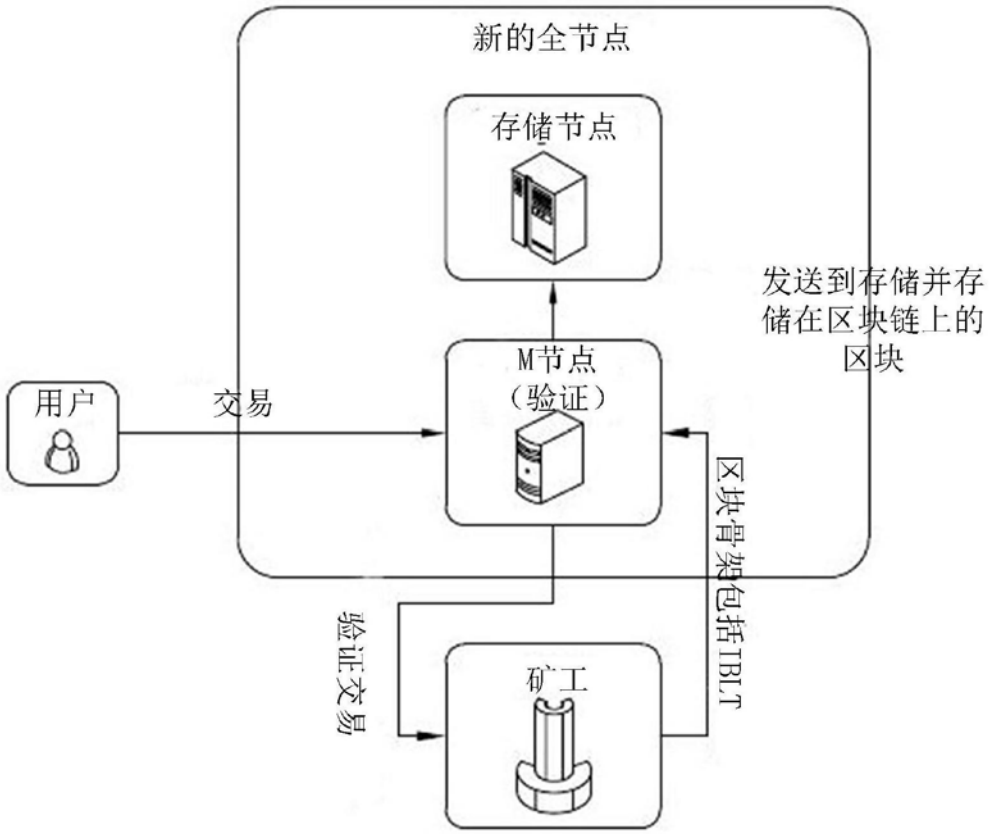

新的全节点
存储节点
发送到存储并存储在区块链上的区块
M节点
（验证）
用户
交易
区块骨架包括TBLT
验证交易
矿工

图2

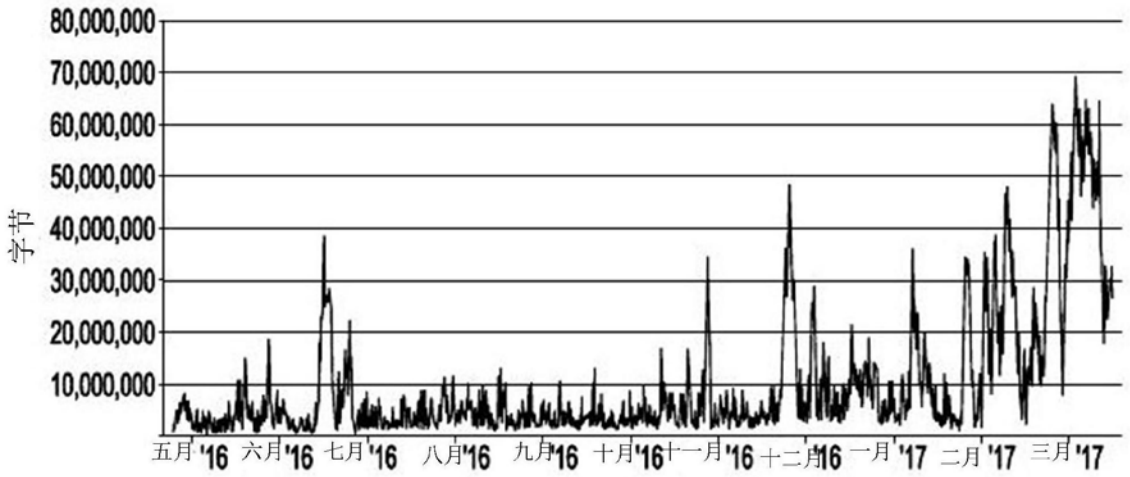

80,000,000
70,000,000
60,000,000
50,000,000
40,000,000
30,000,000
20,000,000
10,000,000
字节
五月'16
六月'16
七月'16
八月'16
九月'16
十月'16
十一月'16
十二月'16
一月'17
二月'17
三月'17

图3

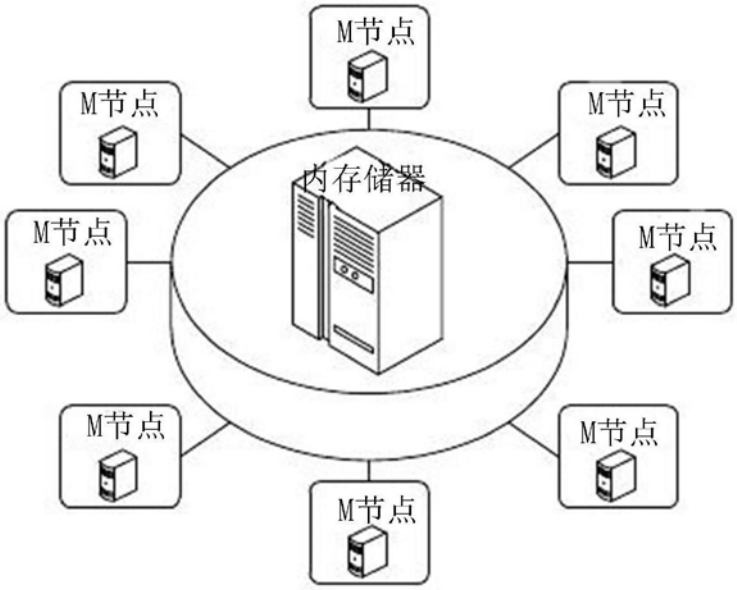

M节点
M节点
M节点
M节点
内存储器
M节点
M节点
M节点
M节点

图4

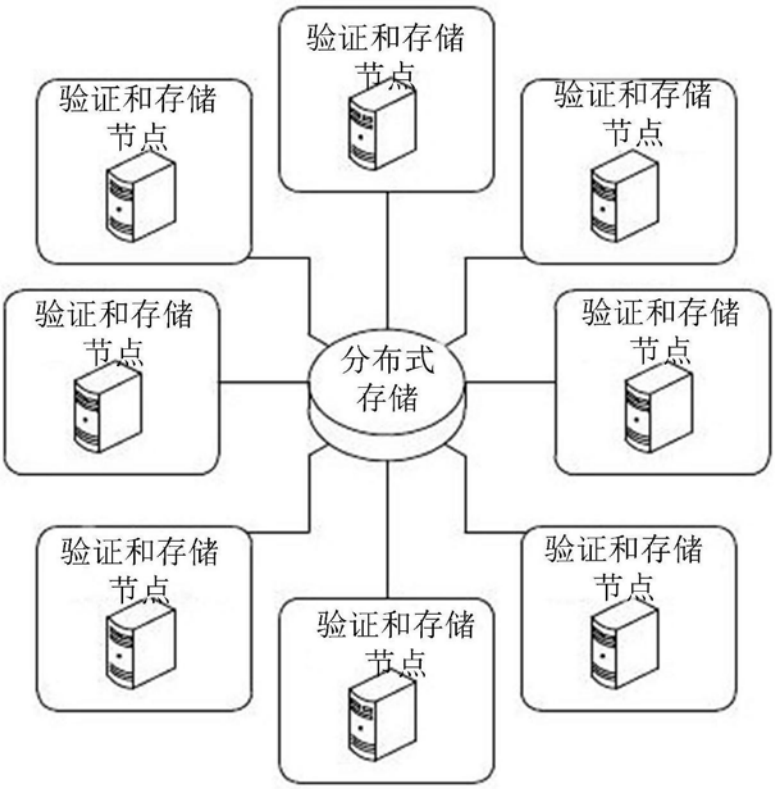

验证和存储节点
验证和存储节点
验证和存储节点
验证和存储节点
分布式存储
验证和存储节点
验证和存储节点
验证和存储节点
验证和存储节点

图5

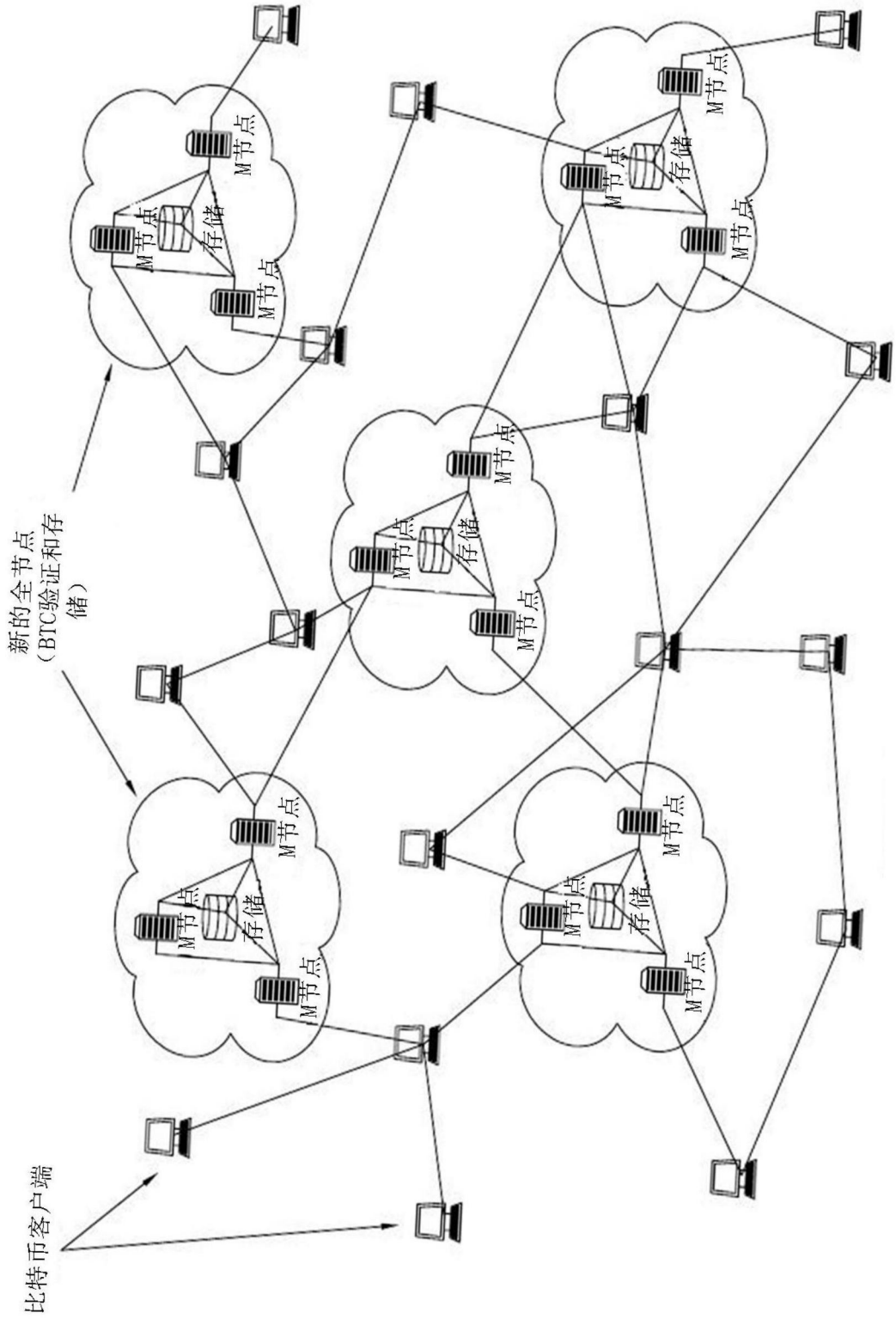


图6

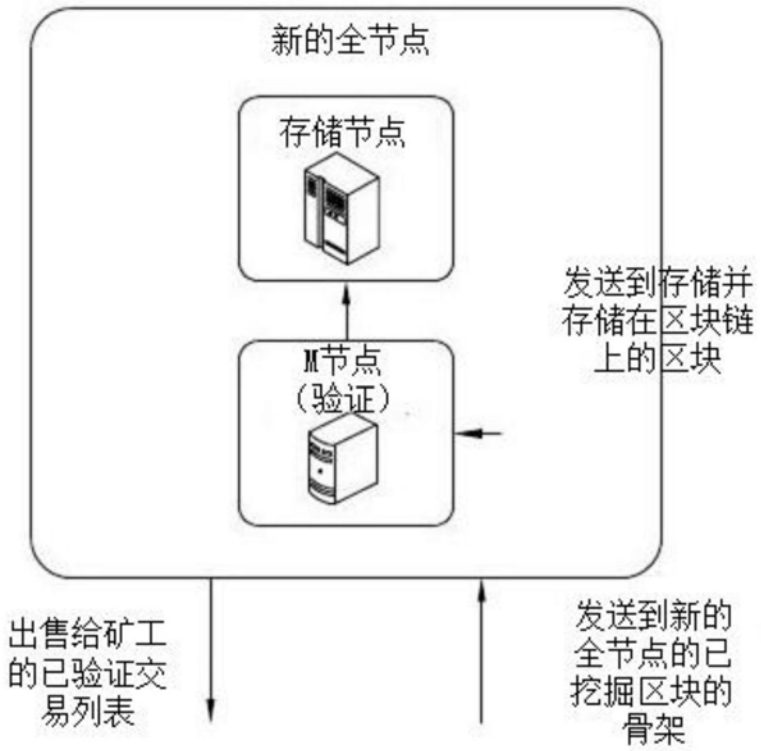

新的全节点
存储节点
M节点
（验证）
发送到存储并
存储在区块链
上的区块
出售给矿工
的已验证交
易列表
发送到新的
全节点的已
挖掘区块的
骨架

图7

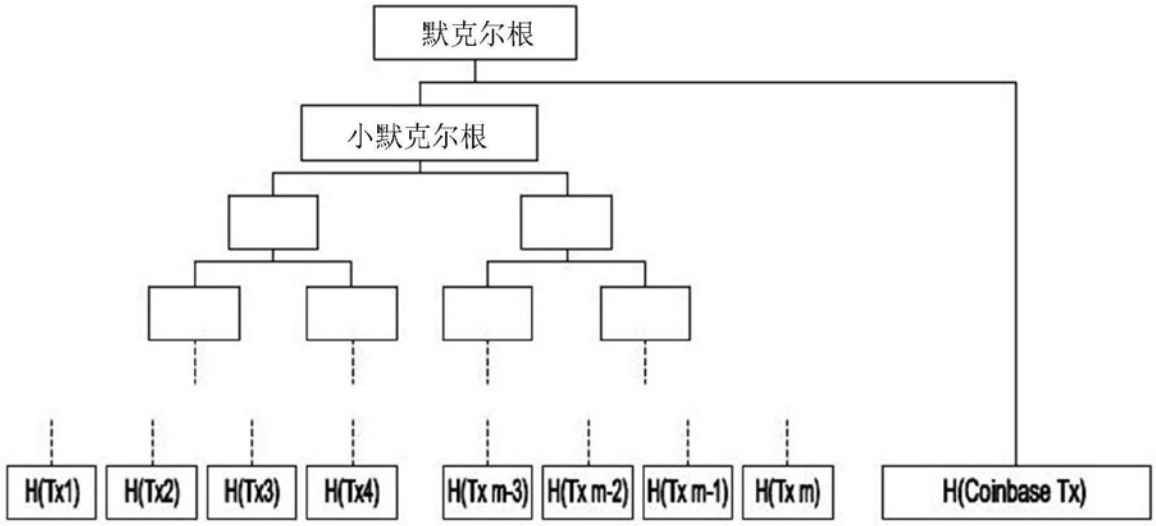

默克尔根
小默克尔根
H(Tx1)
H(Tx2)
H(Tx3)
H(Tx4)
H(Tx m-3)
H(Tx m-2)
H(Tx m-1)
H(Tx m)
H(Coinbase Tx)

图8

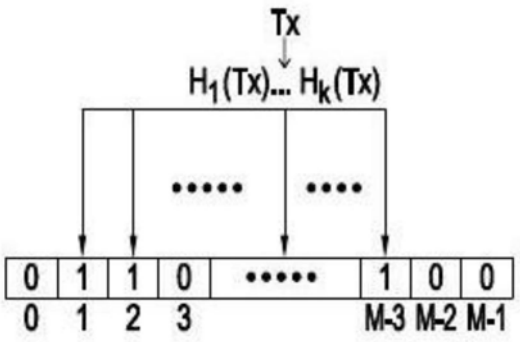

Tx
$H_1(Tx)... H_k(Tx)$
0 1 1 0 ••••• 1 0 0
0 1 2 3 M-3 M-2 M-1

图9

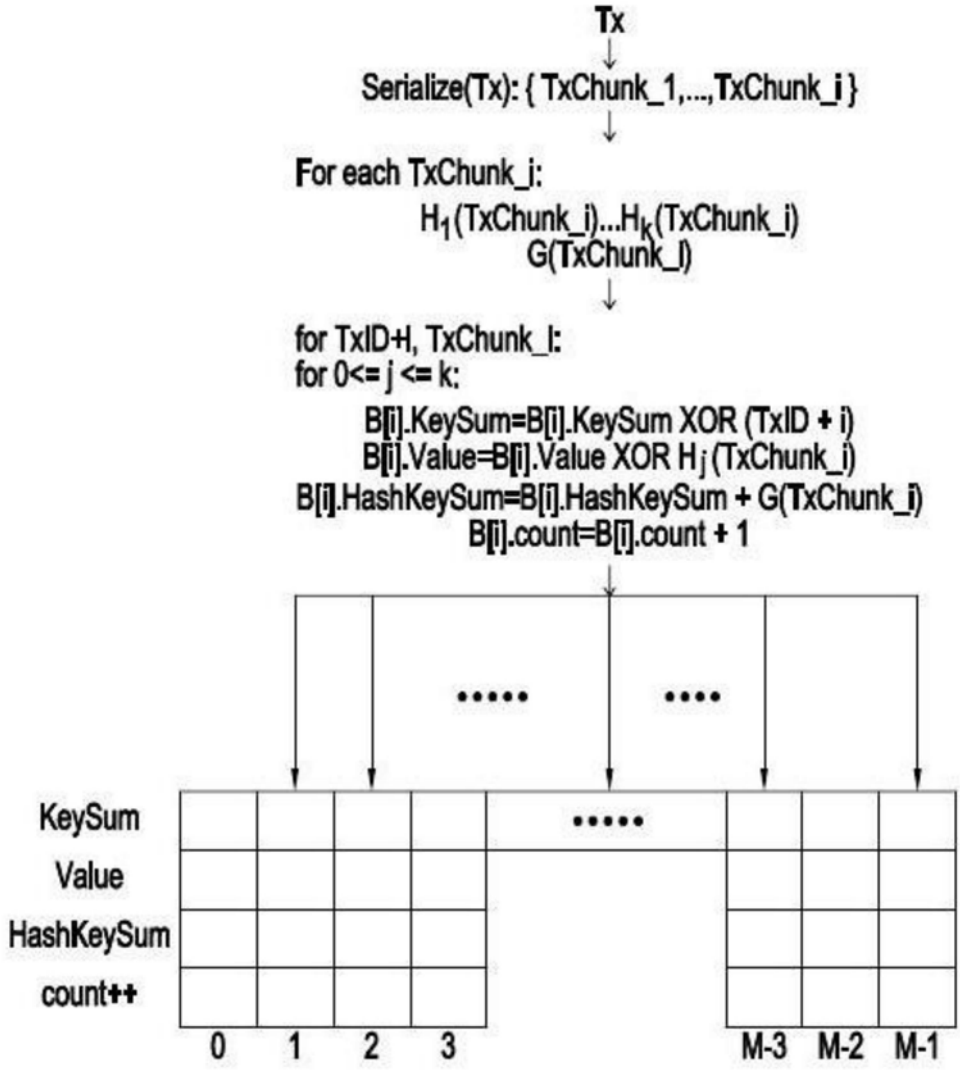

Tx
Serialize(Tx): { TxChunk_1,...,TxChunk_i }
For each TxChunk_j:
$H_1(TxChunk\_i)...H_k(TxChunk\_i)$
G(TxChunk_j)
for TxID+H, TxChunk_l:
for 0<= j <= k:
B[i].KeySum=B[i].KeySum XOR (TxID + i)
$B[i].Value=B[i].Value\ XOR\ H_j(TxChunk\_i)$
B[i].HashKeySum=B[i].HashKeySum + G(TxChunk_i)
B[i].count=B[i].count + 1
KeySum
Value
HashKeySum
count++
0 1 2 3 M-3 M-2 M-1

图10