华南理工大学

课程报告

课程名称：	企业软件项目实训
学生姓名：	卢越兴
学生学号：	201630665298
学生专业：	软件工程
开课学期：	2018-2019 第二学期

软件学院

2019 年 7 月

一， 区块链技术原理

区块链技术是以比特币为代表的数字加密货币体系的核心支撑技术，它最早出现在一个名为中本聪（Satoshi Nakamoto）的学者在密码学邮件组发表的论文《比特币：一种点对点的电子现金系统》[3]。区块链在文献中被描述成一种按照时间顺序将数据区块以类似链表的结构组合起来的特定数据结构，并以密码学方式保证的不可篡改和不可伪造的去中心化共享账本，能够安全地储存简单地、有先后关系的、能够在系统内进行验证的数据。

区块链具有去中心化、防篡改、匿名性、开放性、自治性等特点。首先是去中心化：由于采用的是分布式系统结构，在区块链中不存在起中心管理的硬件或机构，区块链上数据的验证、级长、储存、维护和传输都是在分布式节点之间进行，所有节点通过存数学的方式拥有平等的权力和义务；然后是防篡改：区块链采用非对称的密码学加密技术对数据进行加密，同时借助于分布式系统上的各个节点通过特定的共识算法来共同存储，一般来说，一旦数据通过共识算法验证并被添加到区块链中，就会永久的存在于链上，除非能够控制整个系统的 51%以上的节点，否则无法对区块链上的数据进行修改；然后是匿名性：由于节点之间的数据交换遵循的是固定的算法，节点之间的交互无需信任（不需要识别对方的身份是否有效），因此交易时不需要通过公开自己的身份来获取对方交易节点的信任；然后是开放性：抛开联盟链和私有链不谈，在公有链中，整个系统是公开的，除了交易双方的私有信息被加密以外，区块链上的数据对所有人开放，任何人都可以通过公开的接口查询区块链数据和开放相关的应用，因此区块链系统上的信息高度透明；还有就是自治性：区块链采用的是基于协商一致的规范和协议（比如一种公开的共识算法等），使得系统中的所有节点的可以在一个信任的环境中和对方交换数据。这样也将对某一个人或节点的信任转换为对系统或机器的信任，限制人对系统的干预程度。

区块链技术涉及到密码学、博弈论、分布式系统和数据库原理等领域，可以说区块链并不是什么新技术，也没有发明什么新技术，它只是以上领域中的成熟技术的组合。在文献[1]中，作者将区块链的基础模型与关键技术分为数据层、网络层、共识层、己理层、合约层和应用层。我认为其中主要的技术有：哈希函数、Merkle 树、非对称加密、链式存储、数据区块、P2P 网络、共识算法和智能合约。

哈希函数（也称散列函数）：一般来说、区块链不会直接保存原始数据，而

是保存原始数据的哈希函数值，也就是将原始数据通过一个哈希函数编码成特定长度的只由数字和字母组成的字符串，再将这个字符串储存在区块链中。哈希函数具有许多优良的特点，例如，想要通过哈希值来反推输入的数据基本不可能，输入的数据即使只是一个字符的变化也会导致输出的哈希值发生根本的改变，每次计算哈希值所需的时间基本一致，且输出的字符串的长度可以固定。比如常见的 SHA256 算法，就是将任意长度的数据经过运算转换成功长度为 256 位（即 32 字节）的二进制数字，SHA256 的巨大的散列空间（2^{256}）也可以避免不同的输入值产生相同的哈希值（碰撞），比特币区块链就使用了双 SHA256 哈希算法。

Merkle 树：默克尔树（又称哈希树）是一种典型的二叉树结构，也是区块链技术中的重要数据存储结构。Merkle 树最主要的特点就是只有叶子节点保存数据或数据的哈希值，而其他的非叶子节点则保存它的两个孩子节点的哈希值，通过这种组织结构，Merkle 树可以应用于：快速比较大量的数据，因为相同的数据在构建完成 Merkle 树以后，根节点一定会相同，因此只需要比较根节点即可；快速定位修改，根据其二叉树的特殊，在 O(log n)时间内就可以搜索到发生了改变的数据块；零知识证明，验证某一个数据块的正确性而不需要知道其他数据快，只需要将已有的数据块和非叶子节点的哈希值逐层向上进行运输，只要根节点相同即可。Merkle 树的诸多特性可以极大地提高区块链地运行效率。

非对称加密：非对称加密是指在加密和解密的过程中使用不同（非对称）的密钥，分别为公钥和私钥，一般来说，私钥是通过随机数算法的来的，由个人持有，不向外界公开，公钥由私钥通过一定的算法生成，会向外公开，而且使用公钥加密的数据只能使用私钥进行正确的解密。在数据传输的过程中，发送方通过使用公钥对数据进行加密，接收方通过使用自己保存的私钥进行解密，可以防止数据在传输的过程被第三方获取。非对称加密还可以应用于身份验证，通过不对称的公钥和私钥来确保登录信息被正确接收和认证。常见的非对称加密算法有：RSA、ElGamal、椭圆曲线加密算法（ECC）等。

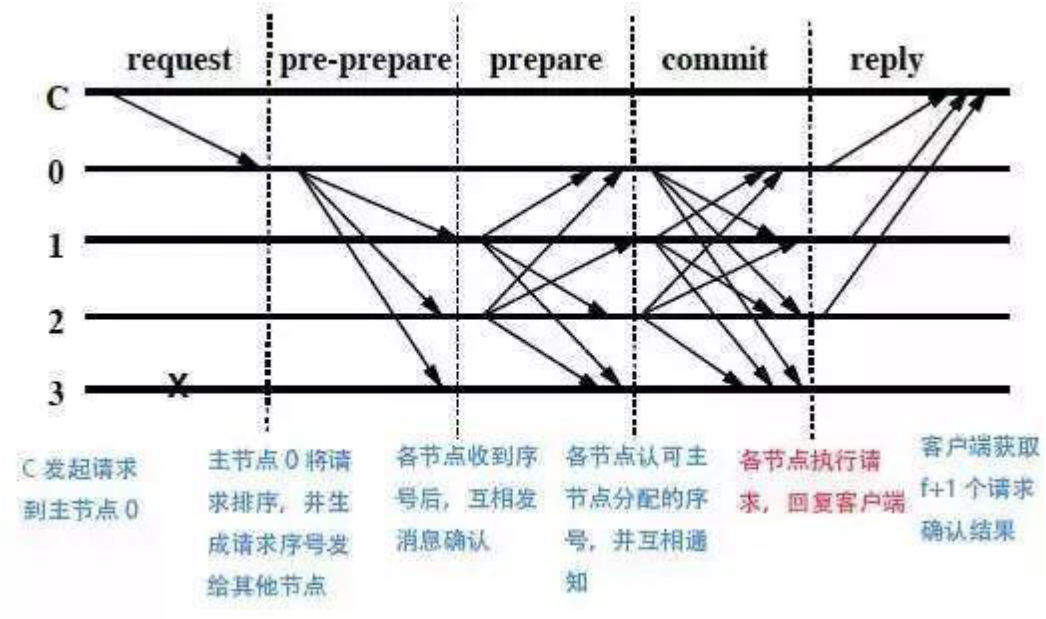
链式结构：获得记账权的节点可以将最新产生的区块连接到主链上，并在新区块中记录上一区块的哈希值。这样的链实结构可以使新的区块在主链上不断的增加，并且顺着记录向旧的区块查找，以达到追溯本源的目的。需要注意的是，当有多个新区块同时想要连接到主链上时，可以通过一些算法规则来选择哪一个新区块可以连接。例如比特币区块链中使用的就是工作量证明法，即新区块总会

连接到累计工作量证明最大的备选链上，从而形成更长的新主链。

数据区块：区块链中的每个数据区块都包含区块头(header)和区块体(body)两部分，其中区块头存储当前版本号、前一个区块的哈希值、当前区块的哈希值、当前区块在共识过程中产生的随机数和时间戳等信息。区块体这包含当前区块的数据，如交易数量和交易记录等。

P2P 网络：区块链系统采用的是一种分布式、自治性的对等网络(Peer to peer Network)，也就是 P2P 网络，来管理分布在世界各地的所有节点。在这种网络中，每个节点的地位都是相等的，不存在具有特殊权力的中心节点或者管理节点，并且每个节点都以拓扑网状结构相互连通和传输数据。P2P 网络中的每个节点都承担着验证和传播区块数据，网络路由和发现新的节点等功能。按照存储的数据量的大小划分，可以分为全节点和轻量级节点，全节点需要储存从创世区块到当前最新区块的所有完整数据，并需要实时的验证和更新主链；而轻量级节点只需要保存部分的区块链数据，并通过 Merkle 树的特点使用零知识证明来验证数据块的完整性。

共识算法：一般来说，一个系统中的决策权越分散，系统的个部分达成共识的效率就会也低，因此，由于区块链系统中的每一个节点都是平等的，当需要对一些事务进行决策的时候，就需要通过特定的共识算法来时分布式、去中心化系统的所有节点高效地达成共识，例如对区块数据的有效性达成共识。常见的共识算法有 POW、PBFT、Raft 等。其中最早的区块链（如比特币）使用的时 POW（工作量证明）算法来保证账本的一致性，这种共识算法高度依赖节点的算力，往往算力最大的节点会获得记账权，比特币系统中就是让每个节点计算一个十分复杂的 SHA256 数学问题，最早计算出结果并验证成功的节点就可以获得比特币奖励，往往算力最大的节点可以更快地算出结果；PBFT 共识算法则是降低了 BFT 的运行复杂度，如图所示，PBFT 共识算法的通信模式，每一个客户端的请求需要经过 5 个阶段，通过采用两次两两交互的方式在服务器达成一致之后再执行客户端的请求。由于客户端不能从服务器端获得任何服务器运行状态的信息，PBFT 中主节点是否发生错误只能由服务器监测。如果服务器在一段时间内都不能完成客户端的请求，则会触发视图更换协议。其中 C 为客户端，0~3 表示服务节点，特别的，0 为主节点，3 为故障节点。整个协议的基本过程如下



首先，客户端发送请求，激活主节点的服务操作。当主节点接收请求以后，开始启动三阶段的协议以向各从节点广播请求：1. 序号分配阶段，主节点给请求赋值一个序列号 n，广播序号分配消息和客户端的请求消息 m，并将构造 PRE-PREPARE 消息给各从节点；2. 交互阶段，从节点接收 PRE-PREPARE 消息，向其他服务节点广播 PREPARE 消息；3. 序号确认阶段，各节点对视图内的请求和次序进行验证后，广播 COMMIT 消息，执行收到的客户端的请求并给客户端以响应；最后，客户端等待来自不同节点的响应，若有 m+1 个响应相同，则该响应即为运算的结果。

智能合约：如果将前面的技术堪称时区块链底层基础的话，智能合约承担的就是更靠近应用的商业逻辑和功能实现，编写智能合约的也是一些具有图灵完备的脚本语言。以以太坊(Ethereum) 为例子，已经经过多个版本迭代的 Solidity 语言就被广泛地应用于实现在以太坊虚拟机上运行的智能合约，具体 Solidity 的语法等内容请参考官方的文档。通过编写智能合约，可以更好的监控和管理在区块链上发生的交易，并实现更具有商业价值的功能。

随着区块链技术的不断壮大、补充和完善，特别是智能合约的出现，越来越多的面向其他应用场景的区块链系统被开发出来，区块链技术也可以更好地被应用于金融交易、政府事务和资产管理等领域。

二， 联盟链和公有链的异同

区块链根据其去中心化的不同，还可以划分成私有链、联盟链和公有链。其

中，由于私有链仅仅使用了区块链相关的技术对账本进行记录，决策权往往在一个人、一个机构或公司中，与普通的分布式存储方案没有太大的区别，就不在这里讨论。这里主要分析联盟链和公有链的异同。

首先是它们的不同点，主要分为两个方面：开放程度和性能。

开放程度：联盟链只实现了部分的去中心化。在联盟链上，每一个节点往往都会对应某一个实体机构、组织或部门，节点不能随意的加入或退出区块链网络。新节点必须要经过授权才能加入到区块链网络中和参与日常的管理事务，而且不同的节点通常会被分配处理特定的事务或执行特定的功能。区块链上的数据也会有目的性地向部分节点隐藏，以保证高度隐私数据不会泄露。外界想要访问联盟链智能通过有限的查询接口，或者往往就不会有对外开放的接口。联盟链上节点的数量也是较少的。公有链的去中心化程度是最高的，它不受任何的第三方机构监管，每一个节点都是对等、公平的，而且节点可以随时，自由地加入和推出区块链网络，也可以按照节点的意愿和智能合约的限制对公有链进行操作。每一个节点都可以读取区块链上的数据和记录、参与交易以及竞争新区块的记账权等。也因为不会对新节点的加入进行限制，公有链上节点的数量往往会非常多。

性能：由于联盟链是部分去中心化，会对所有的节点进行授权管理，节点之间的合作也会较紧密，职能区分地也较为清晰，所以会表现出吞吐数量高，共识速度快，交易速度快等特点。而公有链因为节点地数量较多，达成共识所需要地时间往往较长，且因节点分布在世界各地，在考虑网络因素地情况下，完成交易的成本也会较高。一句话概括就是：联盟链的效率往往会比公有链高很多。

然后是相同点，联盟链和公有链都属于区块链的其中一种，都运用了区块链相关的技术，如分布式账本、P2P 网络、非对称加密和授权、共识算法和智能合约等，都存在明显的去中心化思想，但去中心化的程度不同。

与公有链相比，联盟链的部分去中心化的特点使它可以更好地应用到商业中来，比如金融交易和货币领域，大部分的商业领域为了监管或者管理的方便，都不可以采取完全的去中心化，但部分去中心化可以很大程度上降低人工干预的成本，提升系统运行的效率。

三， 信任链是如何建立的？

在区块链中，信任链的建立主要是通过两个方面：数字证书和共识机制。

首先是数字证书，数字证书也是基于非对称加密原理的，它是为了防止某一个数字签名（公钥）被人伪造，从而获取加密的信息。换句话说，数字证书可以证明某个公钥是某一个个人或实体机构的，并且确保一旦内容被篡改就会被识别出来，从而实现对用户公钥的安全分发和传输。数字证书根据所保护公钥的用途划分，可以分为加密数字证书和签名验证数字证书；其中，加密数字证书主要用来对加密信息的公钥进行加密，以保证传输信息的保密性的完整性；签名数字证书主要用于解密签名进行身份认证，以保障信息的有效性和不可否认性。数字证书一般由特定的证书认证机构颁发（CA），证书认证机构会给特定写层节点颁发特定的证书，以达到某种授权的目的。权威的证书认证机构包括 DigiCert、GlobalSign、VeriSign 等。但在联盟链中，为了提高隐私性，往往会搭建本地的证书认证机构系统，并且该机构只在本区块链中运行，不会被其他区块链系统认可。

在一个区块链系统中，特别是联盟链，通常会设立一个根证书认证机构（Root CA），它是整个系统信任链的基础，所有其下属机构或节点都会信任根证书认证机构颁发的证书。在此基础上，假设根证书认证机构下有两个节点 A 和 B，B 节点会先去根证书认证机构提交自己的公钥，证书认证机构会用自己的私钥加密 B 节点的公钥，生成一个证书签名，并将该证书签名和 B 节点的公钥合成一个数字证书，返回给 B 节点。当 A 想要确认 B 节点的身份时，B 节点会将 Root CA 颁发给自己的证书发送给 A 节点，证书上包含一个证书签名和 B 节点自己的公钥，A 节点在收到证书后使用根证书机构公布的公钥解密数字签名，并将解密出来的公钥与 B 节点发送过来的公钥进行比对，如果一致则说明发送证书过来的是真正的 B 节点，如果发现不一致的话，就是证书在传输的过程中被第三方篡改了。每一层的证书认证机构都可以向上一层的证书认证机构申请证书，这样，节点就可以通过更上层的证书认证机构来验证某一个机构或节点的有效性。通过这样一层层的颁发数字证书，就可以实现一条从根节点到中间节点到叶子节点的信任链。需要特别注意的是，当某一层的证书认证机构失效的时候，由于这种链式授权的关系，其下层的所有后续证书都会变成不可靠。

第二个建立信任链的基础就是共识机制，也称为共识算法。它在公有链系统中的作用会更加显著，因为在公有链系统中所有的节点都是平等的，当需要为某一件事情做出决定的时候，就需要一种多方协作的机制，用来协调多个参与方以达成共同接受的唯一结果，并且保证在做出决定的过程难以被欺骗或篡改，以维

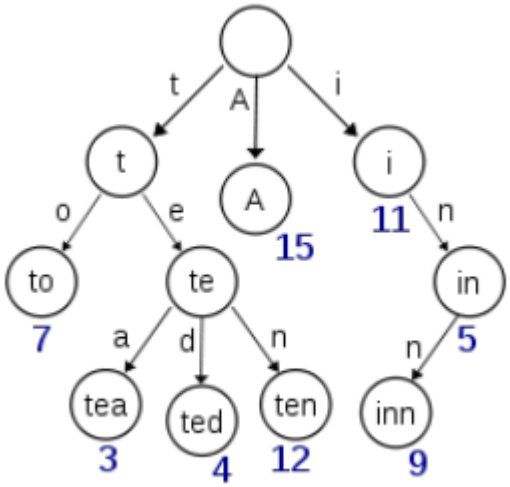
持系统稳定的运行。常见的共识算法有 PBFT、Raft、POW 等。

数字证书和共识机制，将在区块链系统中的对“人”的信任转换成对系统，或者是对算法，的信任，从而一步步建立起系统中的信任链。

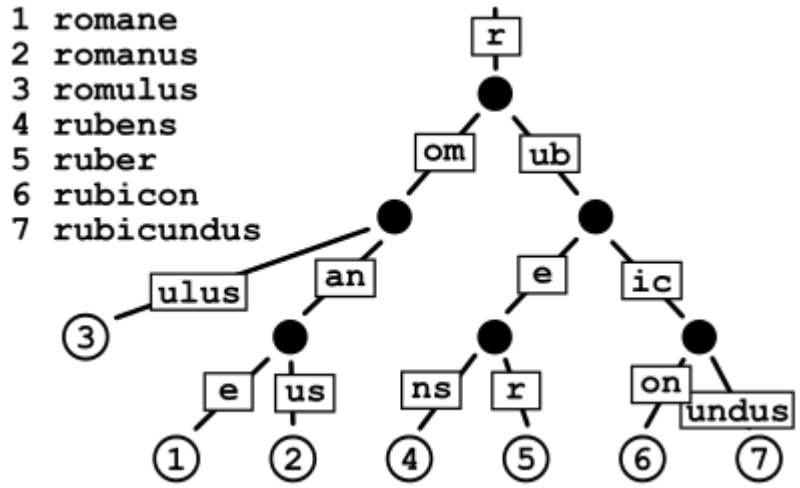
四， 链式存储和 MPT 存储

链式存储：就是采用一种类似链表的结构老保存数据。每一个新的数据块会被添加在链式结构的最尾端，并保存上一个数据块的哈希值。这种数据结构可以帮助通过每个数据块中的记录来查找旧的数据块，以达到溯源的目的。当任意一个数据区块被篡改，都会引发其后搜友区块链哈希只在的连锁改变，所以当从不可信的节点下载某些数据块的时候，可以使用基于块哈希验证各个数据块是否被修改过

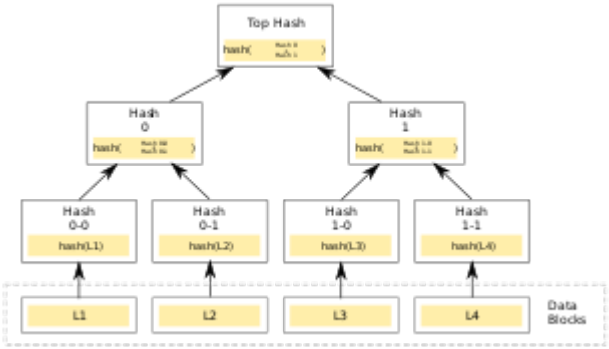
在了解 MPT 存储之前，必须要先了解几个概念，分别是 Trie 树、Patricia 树、Merkle 树。



首先是 Trie 树，如上图所示。Trie 树又被称为前缀树或者字典树，它是一种搜索树，用于存储动态的数据集或关联数组的有序树的数据结构。其中的键通常是字符串。与二叉搜索树不同，树中没有节点存储与该节点相关的密钥；相反，它在树中的位置定义了与之关联的键。节点的所有后代都具有与该节点关联的字符串的公共前缀，并且根与该节点相关联空字符串。密钥倾向于与叶子相关联，但是一些内部节点可以对应于感兴趣的密钥。因此，密钥不一定与每个节点相关联。Trie 树最大的特点就是：树的最大深度就是 key 的最打长度；key 离得越远，value 离得也越远；不是一种平衡树，如果没有相同得前缀得话需要储存更多的节点。



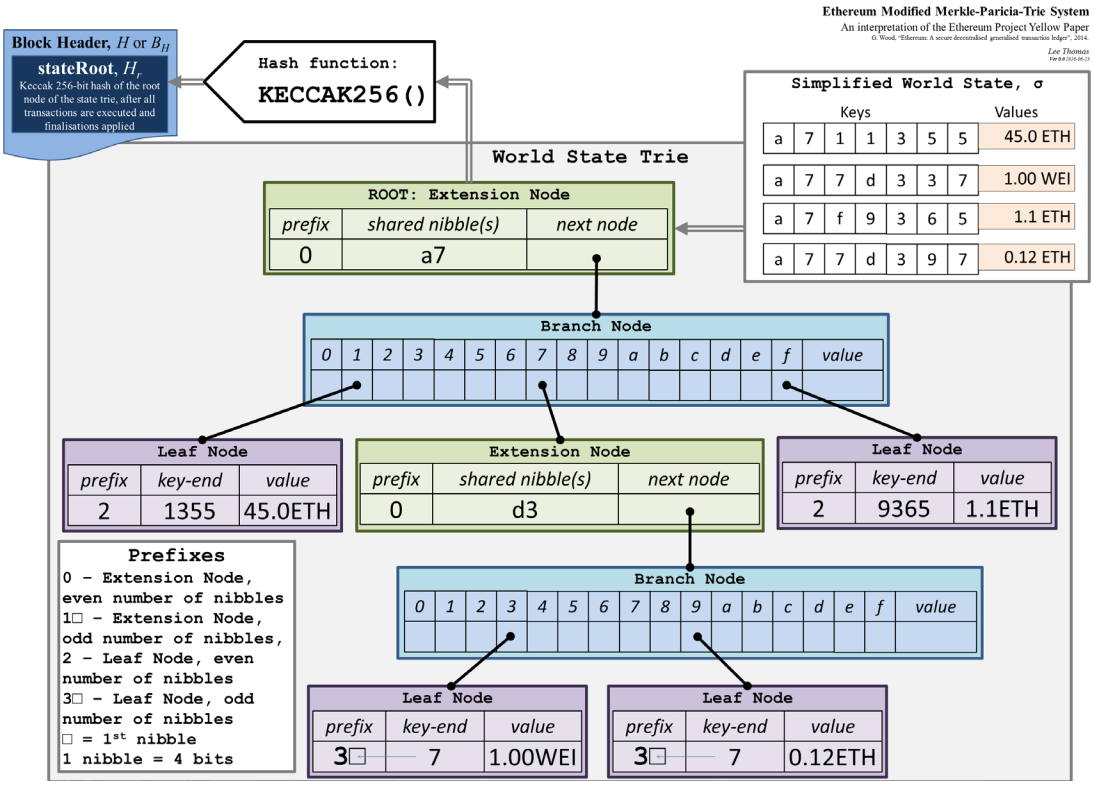
然后是 Patricia 树，如上图所示，它也被称为压缩前缀树，是一种更省空间的前缀树。对于压缩前缀树的每个节点，如果该节点是唯一的子树的话，就和父节点合并。



最后是 Merkle 树，如上图所示。它又称哈希树，是一种典型的二叉树结构，也是区块链技术中的重要数据存储结构。Merkle 树的每个叶子节点都是数据块的散列标记，并且每个非叶子节点用其子节点的标签的哈希值进行标记。Merkle 树是散列表(hash list)的泛化。关于散列表，其应用主要来源于在 P2P 网络中进行数据传输的时候，需要同时从多个节点上下载数据，而且许多节点在下载前是被认为不可靠或者不可信的。为了验证下载数据的完整性，我们可以把很大的数据文件切分成一个个小的数据块（每一个数据块的大小可以随意设定）。这样做带来的好处就是：当有一个小的数据块在传输的过程中损坏时，可以单独重新下载这一个小数据块，而不需要重新下载整一个大的数据文件。在 Merkle 树的最低层，就和散列表一样，把数据切分成小的数据块，而且每个数据块都会有与其对应的哈希值，上一层的父节点会将两个子节点的哈希值合并成一个，再上一次并重复这个步骤，最后可以得到一个根节点。我们可以再点对点网络下载文件之前，

从一些可信来源获取到根哈希，比如一些朋友推荐的大型网站。当根哈希可用时，就可以根据这一个根哈希，从任何不可信人的来源接收整个 Merkle 树，如果某一个源的哈希树被损坏或者篡改，就会尝试另一个源的 Merkle 树，直到找到与根哈希匹配的 Merkle 树。Merkle 树和散列表的主要区别在验证完整性的效率上。Merkle 树可以一次下载单独的某一个分支，然后就可以检查这一个分支的完整性，即使当前整个 Merkle 树还不可用；而散列表着需要将整个列表下载完成才能开始检查完整性，十分浪费时间。

现在就可以来讨论一下 MPT（Merkle Patricia Tree）了。顾名思义，MPT 就是 Merkle Tree 和 Patricia Tree 的混合产物。在以太坊中，使用一种特殊的十六进制前缀编码，所以在字母表中就有 16 个字符，其中的一个字符为一个 nibble。



如图所示，MPT 中的节点分为空节点、叶子节点(Leaf Node)、扩展节点(Extension Node)和分支节点(Branch Node)。其中的空节点就简单地表示为空，在代码中也是表现为一个空串；叶子节点表示一个从 key 到 value 映射地一个键值对，其中的 key 是一种特殊的十六进制编码，value 是 RLP 编码；扩展节点也是一个从 key 到 value 映射地一个键值对，但是这里的 value 是其他节点的哈希值，这个哈希值可以用来查询在数据库中的其他节点，也就是可以通过该哈希值连接到其他的节点；分支节点：因为 MPT 中的 key 使用一种特殊的十六进制编码，再

叫上最后面的 value，分支节点就是一个长度为 17 的 list，前 16 位表示对应 key 中的 16 个可能的十六进制字符。如果有一个键值对在这个分支节点终止，最后一个元素表示一个值，即分支节点既可以搜索接下来的路径也可以是路径的终止。

MPT 中还有一个很重要的概念就是特殊的十六进制前缀编码，用来对 key 进行编码。因为字母表是 16 进制的，所以每个节点可能有 16 个孩子。因为有两种键值对节点(叶节点和扩展节点)，引进一种特殊的终止符标识来标识 key 所对应的是值是真实的值，还是其他节点的 hash。如果终止符标记被打开，那么 key 对应的是叶节点，对应的值是真实的 value。如果终止符标记被关闭，那么值就是用于在数据块中查询对应的节点的 hash。无论 key 奇数长度还是偶数长度，HP 都可以对其进行编码。最后我们注意到一个单独的 hex 字符或者 4bit 二进制数字，即一个 nibble。因为 MPT 融合了 Merkle 树和前缀树的特点，因此既有查找能力，可以以一个以太坊账户地址为查找路径，能够快速地从 Merkle Patricia 树根向下查找到叶子节点中账户地状态数据，这种查找能力是二叉 Merkle 树所不具备的[4]。

五， Gas 在智能合约中的作用

首先先要回答一个问题：什么是 Gas？这就要设计到以太坊中的货币——以太币了，在以太坊生态系统中，以太币是其主要的代币，用来已理参与者执行以太坊相关的智能合约项目。Gas 是满足特定智能合约所有需求所需的“燃料”量，它被用来衡量智能合约中的一个行为或一系列行为有多少工作量。不过，Gas 只是计算费用的一种方式，最终费用的计算还是要以以太币来计算。

同时，以太坊十分依赖矿工的哈希效率，更多的矿工意味着更快的哈希效率，和更快速的系统。为了吸引更多多的矿工进入该系统，系统设计者需要让矿工有利可图。为了将一个交易放入区块中，矿工需要使用他们的算力来验证智能合约，Gas 系统可以允许他们在验证的过程中收取一定的费用。

在以太坊中，每一笔交易都会包含量参数：gasLimit 和 gasPrice。其中 gasLimit 指 Gas 的限额，用以防止代码的指数型爆炸和无线循环；gasPrice 是每一计算步骤需要支付矿工的费用[6]。如果 gasLimit 设置的太小的话，当用尽所有的 Gas 都不能完成交易操作是，它就会恢复到原来的状态，就好像这个交易从来没有发生

过一样，但是，发起交易者或者智能合约创建者仍需要向矿工支付计算验证的费用，即使交易并没有被成功的记录下来。如果将 gasLimit 设置地很高呢？因为以太坊中每个数据块会有最大六百七十万 gas limit 的限制。而且矿工只能添加加起来小于或者等于六百七十万的操作，所以如果 gasLimit 设置得很高的话，矿工可能并不会从操作中获取到太多的费用。

而且，目前的智能合约都是使用类似 Solidity 等具有图灵完备的脚本语言编写的，可能会产生一些例如死循环的恶意代码，在智能合约中设置 Gas 的消耗，可以计算在执行时所占用的 CPU 和内存资源消耗，一旦 Gas 被消耗尽，智能合约就会自动停止执行，从而有效的避免合约死循环和无效交易的发生。

但并不是所以的调用合约操作都会消耗 Gas，一般来说，只有发生对数据进行写操作的调用才会消耗 Gas，而读数据往往不会消耗 Gas，一些智能合约脚本语言中也可以通过添加特殊的标识来保证不消耗 Gas。

六， EVM 中的数据存储结构

以太坊为用户提供了一整套完整的合约运行环境，这包括完整的智能合约脚本语言以及图灵完备虚拟机。以太坊虚拟机本质上还是一个堆栈机器，设计上除了满足基本的执行需求外，还要考虑空间节省、安全保证核优化等方面的问题。

以太坊虚拟机将其的存取系统固定为 256 位的机器位宽，这表明了设计者在设计 EVM 的时候需要考虑一套自己的关于操作、数据和逻辑控制的指令编码。256 位的宽度虽然并不会是计算速度加快，但是这样设计更适合进行密码学的计算。同时，固定的位宽是需要支持的操作树减少，简单可控；另一方面，从以太坊上的经济学来讲，操作数的减少意味着以太坊中智能合约执行所需要消耗的 Gas 的数量就会减少，也可以一定程度上降低成本。

EVM中数据可以在三个地方进行存储，分别是栈、临时存储、永久存储。不同存储对应着不同的花费。由于EVM是基于栈的虚拟机，因此基本上所有操作都是在栈上进行，并且EVM 中没有寄存器的概念，这样EVM对栈的依赖就更大，虽然这样的设计使实现比较简单且易于理解，但带来的问题就是需要更多数据的栈操作。在EVM中栈是唯一的免费（几乎是）存放数据的地方，栈自然有深度的限制，目前的限制是1024，因为栈的限制，栈上的临时变量的使用

会受限制。临时内存存储在每个VM实例中，并在合约执行完后消失。永久内存存储在区块链的状态层。

七， 分布式存储有什么优势

分布式存储，顾名思义就是讲数据分散式地存储在多台独立的物理设备上面。传统的存储设备会将所有的数据存放在一台存储服务器中，这样的存储服务器很容易造成整体系统的性能瓶颈，也会大大降低数据的可靠性的安全性，并不能满足现代化的存储系统的需求。而分布式存储系统采用多台物理机的架构，将存储的任务平均地分配到多台物理设备上，可以很大程度地提高系统的可靠性和稳定性，而且分布式系统还具备很强的可扩展性，当存储设备不能满足需求的时候，可以随时的增加物理存储设备。

分布式存储的优势主要体现在五个方面：性能、可扩展性、可靠性、可维护性和较低的成本。

性能：对于绝大多数的传统存储系统来说，它们的性能一直是整个系统的最大短板，由于过度的将 I/O 处理集中到一台物理设备上，即使使用了很大个高速缓存，也很容易会因为 I/O 性能的不住而导致整个系统崩溃。而分布式存储系统可以将 I/O 负载均衡地分配多台物理设备上面，大大提高的整体系统的 I/O 能力。

可扩展性：分布式存储的最大优势就在于“分布式”，它就是将很多的物理存储设备统一整合起来，形成一个巨大的存储系统。如果将每一台存储物理设备看成一个节点的话，当节点的数量不足以承载当前的存储负载时，可以十分简单地通过增加节点地数量来提升系统地存储性能，这是传统的储存系统无法做到的，因为如果想要升级传统的储存系统，往往需要做如果数据迁移等十分复杂、繁琐的工作。

可靠性：也正因为分布式系统的“分布式”，数据被存分在分散的储存物理设备（节点）上面，而且往往会做大量的冗余节点，一保证当某一个节点失效的时候，可以有正常的节点顶替它的位置，以维持系统的稳定运行。传统的单机存储系统中的“单点故障”一直是一个大难题，而巨大的数据量使得准备多台存储设备变得不现实。现代的分布式存储系统甚至以及做到了跨地区的分布式节点，

可以达到预防重大自然灾害的级别。

可维护性：这里说的维护主要是指对硬件的维护。在传统的存储系统中，如果出现了故障，需要对整个系统进行详细、逐层地排查，而且这种单节点的存储系统往往十分的庞大和复杂，排查起来不仅需要大量的时间，而且需要熟悉系统的专业人士进行。而分布式存储系统可以通过管理软件轻松地定位到某一个出故障的节点，并且分布式节点往往是比较轻量级的，在节点上排查故障也相对简单地多，这就大大降低了维护地成本。

较低的成本：其实分布式储存系统的成本较低也是因为它的多种优势。相对较高的性能使用户不需要为了提高传统存储设备的一点性能就花费许多的资金；较高的可扩展性也可以使用户在初期不需要一次性投入大量的资金建设整体系统，可以在有需要的时候再对存储系统进行扩展。较高的可靠性也可以帮助用户减少因需要冗余备份而付出的大量金钱。

参考文献

[1] 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(04):481-494.

[2] 沈鑫,裴庆祺,刘雪峰.区块链技术综述[J].网络与信息安全学报,2016,2(11):11-20.

[3] Nakamoto S. Bitcoin: a peer-to-peer electronic cash system[Online], available: https://bitcoin.org/bitcoin.pdf, 2009

[4] 邵奇峰,金澈清,张召,钱卫宁,周傲英.区块链技术:架构及进展[J].计算机学报,2018,41(05):969-988.

[5] 骆慧勇.区块链技术原理与应用价值[J].金融纵横,2016(07):33-37+76.

[6] 曹迪迪,陈伟.基于智能合约的以太坊可信存证机制[J].计算机应用,2019,39(04):1073-1080.

[7] 孙炜.浅谈区块链虚拟机[J].信息通信技术与政策,2018(07):34-36.

[8] 杨永周.分布式存储关键技术及优势分析研究[J].网络安全技术与应用,2017(10):76+80.