**课程报告**

|  |  |
| --- | --- |
| **课程名称：** | **企业软件项目实训** |
| **学生姓名：** | **李冠海** |
| **学生学号：** | **201630664703** |
| **学生专业：** | **软件工程** |
| **开课学期：** | **2018-2019第二学期** |

**软件学院**

**2019年6月**

**摘要：**区块链思想起源于数字货币的发展，它是分布式账本技术的一种形式，是分布式数据存储、点对点传输、共识机制、加密算法等计算机技术的新型应用模式。区块链根据去中心化程度可以分为公有链、联盟链和私有链，FISCO-BCOS是由国内企业主导研发、对外开源、安全可控的企业级金融开源联盟链底层平台。本文主要围绕实训的内容，对区块链相关技术及FISCO-BCOS的特性进行介绍。

**关键字：**区块链；FISCO BCOS；多群组；智能合约；联盟链

目录

[1. 区块链技术原理 3](#_Toc13402673)

[2. 联盟链与公有链的异同 4](#_Toc13402674)

[3. Gas与智能合约 4](#_Toc13402675)

[4. 群组架构及其好处 6](#_Toc13402676)

[5. 分布式存储的优势 8](#_Toc13402677)

[6. 链式存储和MPT存储 8](#_Toc13402678)

[7. 当前区块链实施的难度 9](#_Toc13402679)

[8. EVM及其数据存储结构 10](#_Toc13402680)

1. 区块链技术原理

一个区块链系统可以分为五个层次：数据层、网络层、共识层、激励层、合约层，每一层都由对应的技术来支撑。

数据层是对底层数据的封装，涉及数据加密等技术。在数据层，数据是分区块存储的，一个区块一般由区块头和区块体组成，前者通常记录有当前版本号、前一区块地址、当前区块的目标哈希值、默克尔树根等信息；后者则由当前区块的交易数量和交易记录组成。这些区块通过链式结构连接到一起，形成从创世区块到当前区块的一条最长主链, 从而记录了区块链数据的完整历史, 能够提供区块链数据的溯源和定位功能。在数据层，通过在每个区块的头部记录时间戳来证明数据的存在，即存在性证明；通过哈希函数对原始数据或者交易记录进行哈希并保存哈希后的值，比较常见的哈希算法有MD4、MD5和SHA系列算法；通过默克尔树来校验区块的完整性，默克尔树运算通过对区块体进行递归分组哈希，只到最终只剩一个根哈希值，这个根哈希值将被记录到区块头中；通过非对称加密来保证交易的安全性。

网络层是对节点通信的封装。在组网方式方面，区块链系统一般采用P2P网络来组织节点，每个节点的地位都是相同的，负责区块数据的传播、验证等功能，不存在中心化的节点；在数据传播协议方面，当生成数据区块数据后，生成数据的节点将区块数据广播给其他所有节点，当足够多的节点验证并通过后，区块就会被链接到主区块链上；在数据验证方面，节点会按照预先定义的标准清单，从数据结构、数字签名等方面来校验交易数据。

共识层是对共识机制的封装，常见的共识机制有PBFT，Raft，PoW，PoS等。PBFT和Raft属于合作型的算法，而PoW和PoS则属于竞争型的算法。PoW即工作量证明，各个节点利用自身的算力来竞争解决一个容易验证但求解困难的数字问题，解决问题最快的节点就可以得到记账权，将区块上链；PoS即权益证明，在PoS机制中，节点通过持有货币来获得币龄，币龄即货币数量与持有货币时间的乘积，节点在交易时会消耗币龄，消耗币龄越多的节点越容易获得记账权，币龄即所谓的权益，PoS提出的目的是为了解决PoW机制中的资源浪费和安全性问题；PBFT属于拜占庭容错类算法，它通过签名、签名验证、哈希密码学算法来保证消息传递的防篡改性、防伪造性和否可抵赖性，它可以容忍不超过三分之一的故障节点和作恶节点；Raft属于普通容错类算法，它可以容忍一半故障节点，不能防止节点作恶，可以达到一致性；FISCO-BCOS支持PBFT和Raft两种算法。

激励层的主要目标是吸引节点积极记账，并不是所有的区块链系统都有激励层，比如本次实训的联盟链则没有激励层，而像比特币等区块链货币系统则有激励层，在这一类系统中，节点为了获利而消耗一定的算法或者财力来竞争记账权。这一层主要考虑货币的发行机制和分配机制。

合约层则是对区块链系统的脚本代码、算法和智能合约的封装，它是建立在区块链虚拟机之上的业务逻辑，通过脚本或者智能合约来控制交易过程，使得系统能够支持更多的应用。FISCO-BCOS支持用solidity语言编写的智能合约，solidity是图灵完备语言，本次实训中我用solidity编写了两个智能合约，一个负责管理账户信息，一个负责管理市场信息，提供交易接口。Solidity与javascript和C++等语言类似，它支持结构体、数组、映射、函数、接口和继承等元素，在我所编写的智能合约中，定义了用户、交易等结构体，通过映射，即键-值对的方式来存储用户地址到用户信息的映射，通过数组的方式来存储交易结构体，在市场管理合约中，定义了用户、卡片和交易三个合约的接口，在交易的过程调用这三个接口来改变用户、卡片和交易的信息。

1. 联盟链与公有链的异同

联盟链和公有链的相同之处在于：它们都是区块链，都属于分布式系统，都通过高冗余和密码学算法来防篡改和伪造，都需要特定的共识算法来使节点达到共识，都有各自适合的应用场景。

联盟链和公有链的不同之处在于：联盟链是由多个机构组成的共同体，它是部分去中心化，只有特定的节点可以参与共识过程，它通过准入机制来限制节点的参与，公众可以查阅和交易，但不能验证交易和发布智能合约，如果要进行上述操作，需要获得联盟的许可，也就是说，这类区块链是受到一些预选节点的控制，它适用于商业联盟系统；而公有链则是完全去中心化的，任何节点都可以发送交易、验证交易，每个节点都可以参与到共识过程中，因此这类区块链很难被一些特定节点所控制，它适用于虚拟货币系统。

公有链的一个典型应用是以太坊，以太坊是一个开源的有智能合约功能的公共区块链平台，通过其专用加密货币以太币提供去中心化的以太虚拟机来处理点对点合约。以太坊协议将尽可能简单，即便以某些数据存储和时间上的低效为代价。一个普通的程序员也能够完美地去实现完整的开发说明。这将最终有助于降低任何特殊个人或精英团体可能对协议的影响并且推进以太坊作为对所有人开放的协议的应用前景。添加复杂性的优化将不会被接受，除非它们提供了非常根本性的益处。以太坊的不同部分应被设计为尽可能模块化的和可分的。开发过程中，应该能够容易地让在协议某处做一个小改动的同时应用层却可以不加改动地继续正常运行。以太坊开发应该最大程度地做好这些事情以助益于整个加密货币生态系统，而不仅是自身。

联盟链的一个典型应用是Hyperledger，也就是超级账本，它是一个旨在推动区块链跨行业应用的开源项目，是一个许可的区块链构架。它提供一个模块化的构架，把架构中的节点、智能合约的执行以及可配置的共识和成员服务。一个Fabric网络包含同伴节点执行chaincode合约，访问账本数据，背书交易并称为应用程序的接口。命令者节点负责确保此区块链的一致性并传达被背书的交易给网络中的同伴们；以及MSP服务，主要作为证书权威管理X.509证书用于验证成员身份以及角色。

1. Gas与智能合约

智能合约是一种由事件驱动的、具有状态的代码合约和算法合同，它利用协议和用户接口完成合约过程的所有步骤，允许用户在区块链上实现个性化的代码逻辑。类似于传统合约，智能合约全生命周期包括：合约生成、合约发布、合约执行３个部分：

合约生成，主要包含合约多方协商、制定合约规范、进行合约验证、获得合约代码４个环节。具体实现过程为：由合约参与方进行协商，明确各方的权利与义务，确定标准合约文本并将文本程序化，经验证后获得标准合约代码。其 中涉及２个重要环节：合约规范和合约验证。合约规范需要由具备相关领域专业知识的专家和合约方进行协商制定。合约验证在基于系统抽象模型的虚拟机上进行，它是关乎到合约执行过程安全性的重要环节，必须保证合约代码和合约文本的一致性。

合约发布与交易发布类似，经签名后的合约通过P2P的方式分发至每一个节点，每个节点会将收到的合约暂存在内存中并等待进行共识。共识过程的实现：每个节点会将最近一段时间内暂存的合约打包成一个合约集合，并计算出该集合的 Hash值，最后将这个合约集合的Hash值组装成一个区块并扩散至全网的其他节点；收到该区块的节点会将其中保存的 Hash值 与 自 己 保 存 的 合约集合的Hash值进行比较验证；通过多轮的发送与比较，所有节点最终会对新发布的合约达成共识，并且达成共识的合约集合以区块的形式扩散至全网各节点。

合约的执行是基于“事件触发”机制的。基于区块链的智能合约都包含事务处理和保存机制以及一个完备的状态机，用于接受和处理各种智能合约。智能合约会定期遍历每个合约的状态机和触发条件，将满足触发条件的合约推送至待验证队列。待验证的合约会扩散至每一个节点，与普通区块链交易一样，节点会首先进行签名验证，以确保合约的有效性，验证通过的合约经过共识后便会成功执行。整个合约的处理过程都由区块链底层内置的智能合约系统自动完成，公开透明，不可篡改。智能合约的实现，本质上是通过赋予对象（如资产、市场、系统、行为等）数字特性，即将对象程序化并部署在区块链上，成为全网共享的资源，再通过外部事件触发合约的自动生成与执行，进而改变区块链网络中数字对象的状态（如分配、转移）和数值。智能合约可以实现主动或被动的接受、存储、执行和发送数据，以及调用智能合约，以此实现控制和管理链上数字对象。目前已经出现的智能合约技术平台，如以太坊、Hyperledger等，具备图灵完备的开发脚本语言，使得区块链能够支持更多的金融和社会系统的智能合约应用。

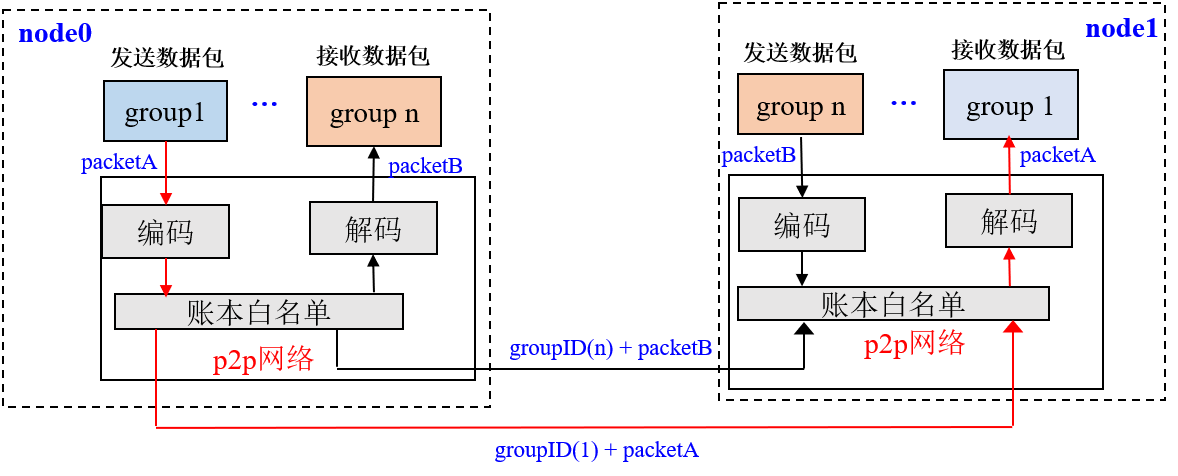
Gas是以太坊中的一个特殊单位，它相当于现实生活中的汽油，而智能合约则相当于汽车。Gas可以用来衡量一个操作或者一组操作要执行的工作量，比如：执行一次keccak256哈希函数需要消耗30个Gas。在以太坊平台上，每一个可以通过交易或合约执行的操作都需要一定数量的Gas，占用计算资源越多的操作消耗的Gas越多。Gas有助于确保向已经提交的交易支付合理的费用，通过要求一个交易或者合约为它执行的每一个操作支持费用，来保证区块链网络不会因为执行大量对任何人没有价值的计算密集型工作而陷入瘫痪。这是一种不同于比特币交易费的策略，比特币交易费仅基于交易的大小；由于以太坊允许任意复杂的计算机代码运行，即使一小段代码也可能会需要大量的计算工作，因此不能仅仅根据交易或合约的长度来确定费用，而要根据操作消耗的资源来衡量费用。所以，Gas是一笔交易费，它只存在于以太坊虚拟机内部，无法被节点所拥有，当要消耗Gas时，它会将一定量的以太币做为交易费，以太币是以太坊网络中的代币，当矿工产生区块后将获得以太币。不使用以太币来直接充当Gas的理由是：以太币和比特币一样，市场价格可能迅速的变化，但计算消耗的资源不会随着以太币的价格变化而变化，也就是说一个操作的成本（即消耗的Gas）不会随市场变化。换句话说，EVM中的操作会消耗Gas，而Gas本身又有价格，它相当于特定量的以太币。每一笔交易可以指定Gas的价格，即指定为每一单元Gas支付多少以太币。

1. 群组架构及其好处

群组架构是FISCO-BCOS的一个主要特性，它采用人们是日常生活中的群聊模式，由具有相同话题的多个节点来建立群组，群组中的每个节点就同一主题进行交流，一个节点可以参与到多个群组中，并行地收发信息。群组在建立之后可以继续增加成员节点。

群组架构自底向下主要划分为网络层、群组层，网络层主要负责区块链节点间通信，群组层主要负责处理群组内交易，每个群组均运行着一个独立的账本。

在网络层方面，所有群组共享P2P网络，群组层传递给网络层的数据包中含有群组ID信息，接收节点根据数据包中的群组ID，将收到的数据包传递给目标节点的相应群组。为了做到群组间通信数据隔离，群组架构引入了账本白名单机制，下图展示了群组架构下群组间收发消息的流程：



每个群组均持有一个账本白名单，用于维护该群组的节点列表。为了保证账本白名单群组内一致性，仅可通过发交易共识的方式修改账本白名单。

以node0的第一组向node1的第一组发送消息packetA为例，发包流程如下：

(1) group1将消息packetA传递到网络层；

(2) 网络层模块对packetA进行编码，在packetA包头加上本群组ID，记为{groupID(1) + packetA}；

(3) 网络层访问账本白名单，判断node0是否是group1的节点，若非group1节点，则拒绝数据包；若是group1节点，则将编码后的包发送给目标节点node1。

收包流程如下：

node1接收到node0 group1的数据包{groupID(1) + packetA}后：

(1) 网络层访问账本白名单，判断源节点node0是否是group1节点，若非group1节点，则拒绝数据包，否则将数据包传递给解码模块；

(2) 解码模块从数据包中解码出group ID=1和数据包packetA，将数据包packetA发送到group1。

通过账本白名单，可以有效地防止群组节点获取其他群组通信消息，保障了群组网络通信的隐私性。

群组层是群组架构的核心。为了实现群组间账本数据的隔离，每个群组持有单独的账本模块。群组层自下向上一次分为核心层、接口层和调度层：核心层提供底层存储和交易执行接口；接口层是访问核心层的接口；调度层包括同步和共识模块，负责处理交易、同步交易和区块。

核心层主要包括存储(AMDB/storage/state)和执行(EVM)两大模块。存储负责从底层数据库中存储或读取群组账本的区块数据、区块执行结果、区块信息以及系统表等。执行(EVM)模块主要负责执行交易。

接口层

接口层包括交易池(TxPool)、区块链(BlockChain)和区块执行器(BlockVerifier)三个模块。

模块1：交易池(TxPool)

交易池是客户端与调度层交互的接口，负责从客户端或者其他节点收到的新交易，共识模块会从中取出交易打包处理，同步模块从中取出新交易进行广播。

模块2：区块链(BlockChain)

区块链模块是核心层与调度层交互的接口，是调度层访问底层存储和执行模块的唯一入口，调度层可通过该模块提交新区块和区块执行结果，查询历史区块等信息。

区块链模块也是RPC模块与核心层的接口，RPC模块通过区块链模块可获取区块、块高以及交易执行结果等信息。

模块3：区块执行器(BlockVerifier)

与调度层交互，负责执行从调度层传入的区块，并将区块执行结果返回给调度层。

调度层包括共识模块(Consensus)和同步模块(Sync)。

共识模块主要负责执行客户端提交的交易，并对交易执行结果达成共识。共识模块包括打包(Sealer)线程和共识(Engine)线程，Sealer线程负责从交易池获取未执行交易，并将交易打包成区块；Engine线程负责对区块执行结果进行共识，目前主要支持PBFT和Raft共识算法。

同步模块主要包括交易同步和区块同步：

客户端通过RPC向指定群组交易池提交新交易时，会唤醒相应群组同步模块的交易同步线程，该线程将所有新收到的交易广播到其他群组节点，其他群组节点将最新交易插入到交易池，保证每个群组节点拥有全量的交易。

考虑到区块链网络中机器性能不一致或者新节点加入都会导致部分节点区块高度落后于其他节点，同步模块提供了区块同步功能，该模块向其他节点发送自己节点的最新块高，其他节点发现块高落后于其他节点后，会主动下载最新区块。

总而言之，群组架构的好处主要有以下两点：

第一，各群组执行独立的共识流程，可以使用不同的共识算法，一个群组内的共识不受其他群组的影响，每个群组都有自己的账本来维护交易数据，因此各群组间是解耦的，每个群组都可以有自己的隐私。

第二，机构的节点只需要部署一次即可参与到不同的多方协作业务中，或将一个业务按用户、时间等维度分到各群组，群组架构可快速地平行扩展，在扩大了业务规模同时，极大简化了运维复杂度，降低管理成本。

1. 分布式存储的优势

分布式存储是一种与集中式存储不同的存储方式，在分布式存储之前，比较常见的存储方式是集中式。所谓集中式，顾名思义就是把数据集中存储在一个系统中，这个系统可以由多个设备组成，如控制器、磁盘阵列和交换机等。

分布式存储最早是由谷歌提出的，其目的是通过廉价的服务器来提供使用与大规模，高并发场景下的Web访问问题，是解决海量数据存储问题的有效手段之一,它利用冗余数据维护技术,通过分散在网络上大量存储节点之间的协作,能够实现长久可靠的数据存储服务;现有大规模数据中心、P2P网络存储和无线网络存储技术等均属于分布式存储系统的范畴。与传统的单体架构相比，分布式系统具有以下优势：

增大了系统的容量。我们的业务量越来越大，就需要多台机器来应对这种大规模的应用场景。因此我们可以使用分布式的架构，来垂直或是水平的拆分业务；

加强了系统的可用性。我们的业务越来越关键，需要提供整个系统架构的可用性，这样就不能存在单点故障。所以，通过分布式架构来冗余系统，提高系统的可用性；

使系统模块化，可以提高模块的重用度，同时系统的扩展性也更高了；

提高了开发和发布速度，因为软件服务模块被拆分，开发和发布都可以并行；

1. 链式存储和MPT存储

在不考虑区块链的情况下，计算机中的链式存储通常是指与数组这一存储结构相对的存储结构，即单链表、循环链表和双向链表等数组结构，这些存储结构的特点是每个节点持有下一个节点的的地址，因此可以通过一个节点得到下一个节点。而在区块链中，链式存储主要是指数据链这一存储方式：数据合入区块，区块构成数据链，每个区块包含一段时间（如10秒）内产生的交易数据，把相关的数据汇总计算摘要，进行汇总的完整性正确性证明，每个区块计算摘要时，把前一个区块的摘要做为一个数据计算在内，构成了数据链，最新区块包含了所有数据链的完整性证明，整个链条上的任何数据改动都会破坏数据链的相关性

MPT 是以太坊存储数据的核心数据结构，它是由 Merkle Tree 和 Patricia Tree 结合的一种树形结构。

Merkle树，也被称为 Hash Tree，主要用于数据集较大时的文件校验，它的主要特点有：

叶节点存储着数据块的 Hash（如：文件块、一段数据集）

非叶子节点 (包括中间节点和根节点) 存储着对应子节点 Hash 值串联字符串之后的 Hash 值。

在最底层，和哈希列表一样，数据被分成小的数据块，有相应的哈希和它对应；从最底层向上，并不是直接去运算根哈希，而是把相邻的两个哈希合并成一个字符串，然后计算这个字符串的哈希值，这样每两个哈希可以生成新的哈希。如果最底层的哈希总数是单数，那到最后必然出现一个没有相邻哈希的节点，这种情况就直接对它进行哈希运算，所以也能得到它的子哈希再往上推，依然是一样的方式，可以得到数目更少的新一级哈希；

最终必然形成一棵倒挂的树，到了树根的这个位置，这一代就剩下一个根哈希了，这就是Merkle Root。

Patricia树是建立在字典树的基础上的一种树结构。字典树，又称前缀树，是一种有序树，用于保存关联数组，其中的键通常是字符串。一个节点的所有子孙都有相同的前缀，也就是这个节点对应的字符串，而根节点对应空字符串，它有以下特点：

根节点不包含字符，除根节点外的每一个子节点都包含一个字符。

从根节点到某一个节点，路径上经过的字符连接起来，为该节点对应的字符串。

每个节点的所有子节点包含的字符互不相同。

Patricia 树，是一种压缩前缀树，是一种更节省空间的字典树。对于每个节点，如果该节点是唯一的儿子的话，就和父节点合并。

MPT（Merkle Patricia Tree），顾名思义就是这两者的结合。在介绍MPT之前需要先介绍一个概念：Nibble，它是 key 的基本单元，是一个四元组。MPT树中的节点包含空节点、叶子节点、扩展节点和分支节点。

空节点：简单的表示空，在代码中是一个空串。

叶子节点：只有两个元素，分别为key和value，表示为（key, value）的一个键值对，其中key是key的一种特殊十六进制编码，value是value的RLP编码。

扩展节点：也是形如（key，value） 的键值对，但是这里的value是其他节点的hash值，这个hash可以被用来查询数据库中的节点。也就是说通过hash链接到其他节点。

分支节点：分支节点有17个元素，回到 Nibble，四元组是 key 的基本单元，四元组最多有16个值。所以前16个必将落入到在其遍历中的键的十六个可能的半字节值中的每一个。第17个是存储那些在当前结点结束了的节点(例如， 有三个 key,分别是 (abc, abd, ab) 第17个字段储存了ab节点的值)。

1. 当前区块链实施的难度

作为近年来兴起并快速发展的新技术，区块链必然会面临各种制约其发展的问题和障碍。在安全、 效率、 资源和博弈四方面，区块链技术都面临一些问题。

在安全方面，安全性威胁是区块链迄今为止所面临的最重要的问题。其中，基于 PoW 共识过程的区块链主要面临的是 51 % 攻击问题，即节点通过掌握全网超过51 % 的算力就有能力成功篡改和伪造区块链数据。以比特币为例，据统计中国大型矿池的算力已占全网总算力的 60 % 以上，理论上这些矿池可以通过合作实施 51 % 攻击，从而实现比特币的双重支付。虽然实际系统中为掌握全网 51 % 算力所需的成本投入远超成功实施攻击后的收益，但 51 % 攻击的安全性威胁始终存在。基于 PoS 共识过程在一定程度上解决了 51 % 攻击问题，但同时也引入了区块分叉时的 N@S (Nothing at stake) 攻击问题。研究者已经提出通过构造同时依赖高算力和高内存的 PoW共识算法来部分解决 51 % 攻击问题，更为安全和有效的共识机制尚有待于更加深入的研究和设计。区块链的非对称加密机制也将随着数学、 密码学和计算技术的发展而变的越来越脆弱。据估计，以目前天河二号的算力来说，产生比特币 SHA256 哈希算法的一个哈希碰撞大约需要 248 年，但随着量子计算机等新计算技术的发展，未来非对称加密算法具有一定的破解可能性，这也是区块链技术面临的潜在安全威胁。区块链的隐私保护也存在安全性风险。区块链系统内各节点并非完全匿名，而是通过类似电子邮件地址的地址标识 (例如比特币公钥地址) 来实现数据传输。虽然地址标识并未直接与真实世界的人物身份相关联，但区块链数据是完全公开透明的，随着各类反匿名身份甄别技术的发展，实现部分重点目标的定位和识别仍是有可能的。

在效率方面，区块链效率也是制约其应用的重要因素。首先是区块膨胀问题: 区块链要求系统内每个节点保存一份数据备份，这对于日益增长的海量数据存储来说是极为困难的。以比特币为例，完全同步自创世区块至今的区块数据需要约 60 GB 存储空间，虽然轻量级节点可部分解决此问题，但适用于更大规模

的工业级解决方案仍有待研发。其次是交易效率问题: 比特币区块链目前每秒仅能处理 7 笔交易，这极大地限制了区块链在大多数金融系统高频交易场景中的应用 (例如 VISA 信用卡每秒最多可处理10 000 笔交易); 最后是交易确认时间问题: 比特币区块生成时间为 10 分钟，因而交易确认时间一般为 10 分钟，这在一定程度上限制了比特币在小额交易和时间敏感交易中的应用。

在资源方面，PoW 共识过程高度依赖区块链网络节点贡献的算力，这些算力主要用于解决 SHA256 哈希和随机数搜索，除此之外并不产生任何实际社会价值，因而一般意义上认为这些算力资源是被“浪费”掉了，同时被浪费掉的还有大量的电力资源。随着比特币的日益普及和专业挖矿设备的出现，比特币生态圈

已经在资本和设备方面呈现出明显的军备竞赛态势，逐渐成为高耗能的资本密集型行业，进一步凸显了资源消耗问题的重要性。因此，如何能有效汇集分布式节点的网络算力来解决实际问题，是区块链技术需要解决的重要问题。研究者目前已经开始尝试解决此问题，例如 Primecoin (质数币) 要求各节点在共识过程中找到素数的最长链条 (坎宁安链和双向双链) 而非无意义的 SHA256 哈希值。未来的潜在发展趋势是设计行之有效的交互机制来汇聚和利用分布式共识节点的群体智能，以辅助解决大规模的实际问题。

在博弈方面，区块链网络作为去中心化的分布式系统，其各节点在交互过程中不可避免地会存在相互竞争与合作的博弈关系，这在比特币挖矿过程中尤为明显。通常来说，比特币矿池间可以通过相互合作保持各自稳定的收益。然而，矿池可以通过称为区块截留攻击 (Block withholding attacks) 的方式、 通过伪装

为对手矿池的矿工、 享受对手矿池的收益但不实际贡献完整工作量证明来攻击其他矿池，从而降低对手矿池的收益。如果矿池相互攻击，则双方获得的收益均少于不攻击对方的收益。当矿池收益函数满足特定条件时，这种攻击和竞争将会造成“囚徒困境”博弈结局。如何设计合理的惩罚函数来抑制非理性竞争、 使得合作成为重复性矿池博弈的稳定均衡解，尚需进一步深入研究。此外，正如前文提到的，区块链共识过程本质上是众包过程，如何设计激励相容的共识机制，使得去中心化系统中的自利节点能够自发地实施区块数据的验证和记账工作，并提高系统内非理性行为的成本以抑制安全性攻击和威胁，是区块链有待解决的重要科学问题。

1. EVM及其数据存储结构

以太坊底层通过EVM模块支持合约的执行与调用，调用时根据合约地址获取到代码，生成环境后载入到EVM中运行。通常智能合约的开发流程是用solidlity编写逻辑代码，再通过编译器编译元数据，最后再发布到以太坊上。EVM的设计目标主要针对这些方面：简单性、确定性、节省空间的bytecode、专为区块链设计、更加简单的安全性保证、容易优化。

EVM中主要有3个用于存储数据的地方：栈、临时内存和永久内存。

每个账户有一块持久化内存区称为存储。存储是将256位字映射到256位字的键值存储区。在合约中读存储相对开销很高，修改存储的开销甚至更高，合约只能读写存储区内属于自己的部分。

第二个内存区域称为内存，合约会为每一次消息调用获取一块全新的被擦除过的内存实例。内存是线性的，可按字节寻址，但是读长度限制为256位宽，而写长度可为8位也可为256位宽。当读写之前从未读写过的内存字(word)时，比如偏移到该字内的任何位置时，内存都将按字进行扩展，这里每个字是256位。扩容也将消耗一定量的gas，随着内存使用量的增长，其费用也会程平方级增高。

由于EVM是基于栈的，因此所有的计算都是在一个被称为栈stack的区域执行的。栈的最大尺寸为1024个元素，每个包含256个位。栈访问只限制访问栈顶端，限制方式为：允许拷贝最顶端16个元素之一到栈顶或者交换栈顶元素和下面16个元素之一。所有其他操作都只能取得最顶端的两个元素，或一个或更多，这取决于具体操作，运算后，把结果压入栈顶。

**参考文献**

1. 柏乐飞.区块链的核心技术分析[J].现代商贸工业,2019,(17):84-85.
2. 杨灵运,张昌福,王飞飞.区块链技术研究与应用综述[J].当代经济,2018,(4):126-128.
3. 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.
4. 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.
5. 何蒲,于戈,张岩峰, 等.区块链技术与应用前瞻综述[J].计算机科学,2017,44(4):1-7,15.
6. 徐倩倩.区块链技术理论与实践阶段性研究及展望[J/OL].软件导刊:1-4[2019-07-01].http://kns.cnki.net/kcms/detail/42.1671.TP.20190624.1359.038.html.
7. 贺海武,延安,陈泽华.基于区块链的智能合约技术与应用综述[J].计算机研究与发展,2018,55(11):2452-2466.