**课程报告**

|  |  |
| --- | --- |
| **课程名称：** | **企业软件项目实训** |
| **学生姓名：** | **莫泽威** |
| **学生学号：** | **201630845119** |
| **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**

1. **区块链技术原理**

    从数据的角度来看：区块链是一种分布式数据库（或称为分布式共享总账，DistributedShared Ledger），这里的“分布式”不仅体现为数据的分布式存储，也体现为数据的分布式记录（即由系统参与者来集体维护）。简单的说，区块链能实现全球数据信息的分布式记录（可以由系统参与者集体记录，而非由一个中心化的机构集中记录）与分布式存储（可以存储在所有参与记录数据的节点中，而非集中存储于中心化的机构节点中）。

    从效果的角度来看：区块链可以生成一套记录时间先后的、不可篡改的、可信任的数据库，这套数据库是去中心化存储且数据安全能够得到有效保证的。

    结论：区块链是一种把区块以链的方式组合在一起的数据结构，它适合存储简单的、有先后关系的、能在系统内验证的数据，用密码学保证了数据的不可篡改和不可伪造。它能够使参与者对全网交易记录的事件顺序和当前状态建立共识。

    如今的区块链技术概括起来是指通过去中心化和去信任的方式集体维护一个可靠数据库的技术。其实，区块链技术并不是一种单一的、全新的技术，而是多种现有技术（如加密算法、P2P文件传输等）整合的结果，这些技术与数据库巧妙地组合在一起，形成了一种新的数据记录、传递、存储与呈现的方式。简单的说，区块链技术就是一种大家共同参与记录信息、存储信息的技术。过去，人们将数据记录、存储的工作交给中心化的机构来完成，而区块链技术则让系统中的每一个人都可以参与数据的记录、存储。区块链技术在没有中央控制点的分布式对等网络下，使用分布式集体运作的方法，构建了一个P2P的自组织网络。通过复杂的校验机制，区块链数据库能够保持完整性、连续性和一致性，即使部分参与人作假也无法改变区块链的完整性，更无法篡改区块链中的数据。

    区块链技术重新定义了网络中信用的生成方式：在系统中，参与者无需了解其他人的背景资料，也不需要借助第三方机构的担保或保证，区块链技术保障了系统对价值转移的活动进行记录、传输、存储，其最后的结果一定是可信的。

    区块链技术原理的来源可归纳为一个数学问题：拜占庭将军问题。拜占庭将军问题延伸到互联网生活中来，其内涵可概括为：在互联网大背景下，当需要与不熟悉的对手方进行价值交换活动时，人们如何才能防止不会被其中的恶意破坏者欺骗、迷惑从而做出错误的决策。进一步将拜占庭将军问题延伸到技术领域中来，其内涵可概括为：在缺少可信任的中央节点和可信任的通道的情况下，分布在网络中的各个节点应如何达成共识。区块链技术解决了闻名已久的拜占庭将军问题——它提供了一种无需信任单个节点、还能创建共识网络的方法。

    区块链技术的本质是一种互联网协议。

    设想一下，如果现在我们想要在互联网世界中建立一套全球通用的数据库，那么我们会面临三个亟待解决的问题，这三个问题也是设计区块链技术的核心所在：

    问题一：如何建立一个严谨的数据库，使得该数据库能够存储下海量的信息，同时又能在没有中心化结构的体系下保证数据库的完整性？

    问题二：如何记录并存储下这个严谨的数据库，使得即便参与数据记录的某些节点崩溃，我们仍然能保证整个数据库系统的正常运行与信息完备？

    问题三：如何使这个严谨且完整存储下来的数据库变得可信赖，使得我们可以在互联网无实名背景下成功防止诈骗？

    针对这三个核心问题，区块链构建了一整套完整的、连贯的数据库技术来达成目的，解决这三个问题的技术也成为了区块链最核心的三大技术。此外，为了保证区块链技术的可进化性与可扩展性，区块链系统设计者还引入了“脚本”的概念来实现数据库的可编程性。我们认为，这四大技术构成了区块链的核心技术。

#### 核心技术1：区块+链

    关于如何建立一个严谨数据库的问题，区块链的办法是：将数据库的结构进行创新，把数据分成不同的区块，每个区块通过特定的信息链接到上一区块的后面，前后顺连来呈现一套完整的数据，这也是“区块链”这三个字的来源。

    区块（block）：在区块链技术中，数据以电子记录的形式被永久储存下来，存放这些电子记录的文件我们就称之为“区块（block）”。区块是按时间顺序一个一个先后生成的，每一个区块记录下它在被创建期间发生的所有价值交换活动，所有区块汇总起来形成一个记录合集。

    区块结构（BlockStructure）：区块中会记录下区块生成时间段内的交易数据，区块主体实际上就是交易信息的合集。每一种区块链的结构设计可能不完全相同，但大结构上分为块头（header）和块身（body）两部分。块头用于链接到前面的块并且为区块链数据库提供完整性的保证，块身则包含了经过验证的、块创建过程中发生的价值交换的所有记录。



    区块结构有两个非常重要的特点：第一，每一个区块上记录的交易是上一个区块形成之后、该区块被创建前发生的所有价值交换活动，这个特点保证了数据库的完整性。第二，在绝大多数情况下，一旦新区块完成后被加入到区块链的最后，则此区块的数据记录就再也不能改变或删除。这个特点保证了数据库的严谨性，即无法被篡改。

    顾名思义，区块链就是区块以链的方式组合在一起，以这种方式形成的数据库我们称之为区块链数据库。区块链是系统内所有节点共享的交易数据库，这些节点基于价值交换协议参与到区块链的网络中来。

    区块链是如何做到的呢？由于每一个区块的块头都包含了前一个区块的交易信息压缩值，这就使得从创世块（第一个区块）到当前区块连接在一起形成了一条长链。由于如果不知道前一区块的“交易缩影”值，就没有办法生成当前区块，因此每个区块必定按时间顺序跟随在前一个区块之后。这种所有区块包含前一个区块引用的结构让现存的区块集合形成了一条数据长链。“区块+链”的数据存储结构如下图所示。



    我们引用《区块链：互联网金融的终局》（肖风）的一段话来总结区块链的基本结构：“人们把一段时间内生成的信息（包括数据或代码）打包成一个区块，盖上时间 戳，与上一个区块衔接在一起，每下一个区块的页首都包含了上一个区块的索引数据，然后再在本页中写入新的信息，从而形成新的区块，首尾相连，最终形成了区块链。”这个结构的神奇之处：区块（完整历史）+ 链（完全验证）= 时间戳

    “区块+链”的结构为我们提供了一个数据库的完整历史。从第一个区块开始，到最新产生的区块为止，区块链上存储了系统全部的历史数据。

    区块链为我们提供了数据库内每一笔数据的查找功能。区块链上的每一条交易数据，都可以通过“区块链”的结构追本溯源，一笔一笔进行验证。

    区块+链=时间戳，这是区块链数据库的最大创新点。区块链数据库让全网的记录者在每一个区块中都盖上一个时间戳来记账，表示这个信息是这个时间写入的，形成了一个不可篡改、不可伪造的数据库。我们认为，时间戳是区块链中一项伟大的技术创新，它可以证明什么呢？

#### 核心技术2：分布式结构——开源的、去中心化的协议

    我们有了区块+链的数据之后，接下来就要考虑记录和存储的问题了。我们应该让谁来参与数据的记录，又应该把这些盖了时间戳的数据存储在哪里呢？在现如今中心化的体系中，数据都是集中记录并存储于中央电脑上。但是区块链结构设计精妙的地方就在这里，它并不赞同把数据记录并存储在中心化的一台或几台电脑上，而是让每一个参与数据交易的节点都记录并存储下所有的数据。

    1.关于如何让所有节点都能参与记录的问题，区块链的办法是：构建一整套协议机制，让全网每一个节点在参与记录的同时也来验证其他节点记录结果的正确性。只有当全网大部分节点（或甚至所有节点）都同时认为这个记录正确时，或者所有参与记录的节点都比对结果一致通过后，记录的真实性才能得到全网认可，记录数据才允许被写入区块中。

    2.关于如何存储下“区块链”这套严谨数据库的问题，区块链的办法是：构建一个分布式结构的网络系统，让数据库中的所有数据都实时更新并存放于所有参与记录的网络节点中。这样即使部分节点损坏或被黑客攻击，也不会影响整个数据库的数据记录与信息更新。

    区块链根据系统确定的开源的、去中心化的协议，构建了一个分布式的结构体系，让价值交换的信息通过分布式传播发送给全网，通过分布式记账确定信息数据内容，盖上时间戳后生成区块数据，再通过分布式传播发送给各个节点，实现分布式存储。

    分布式记账——会计责任的分散化（Distributedaccountability）

    从硬件的角度讲，区块链的背后是大量的信息记录储存器（如电脑等）组成的网络，这一网络如何记录发生在网络中的所有价值交换活动呢？区块链设计者没有为专业的会计记录者预留一个特定的位置，而是希望通过自愿原则来建立一套人人都可以参与记录信息的分布式记账体系，从而将会计责任分散化，由整个网络的所有参与者来共同记录。

    区块链中每一笔新交易的传播都采用分布式的结构，根据P2P网络层协议，消息由单个节点被直接发送给全网其他所有的节点。

    区块链技术让数据库中的所有数据均存储于系统所有的电脑节点中，并实时更新。完全去中心化的结构设置使数据能实时记录，并在每一个参与数据存储的网络节点中更新，这就极大的提高了数据库的安全性。

    通过**分布式记账、分布式传播、分布式存储**这三大“分布”我们可以发现，没有人、没有组织、甚至没有哪个国家能够控制这个系统，系统内的数据存储、交易验证、信息传输过程全部都是去中心化的。在没有中心的情况下，大规模的参与者达成共识，共同构建了区块链数据库。可以说，这是人类历史上第一次构建了一个真正意义上的去中心化体系。甚至可以说，区块链技术构建了一套永生不灭的系统——只要不是网络中的所有参与节点在同一时间集体崩溃，数据库系统就可以一直运转下去。

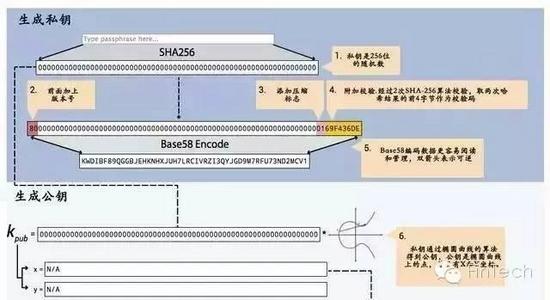
    我们现在已经有了一套严谨的数据库，也有了记录并存储这套数据库的可用协议，那么当我们将这套数据库运用于实际社会时，我们要解决最核心的一个问题（问题三）是：如何使这个严谨且完整存储下来的数据库变得可信赖，使得我们可以在互联网无实名背景下成功防止诈骗？

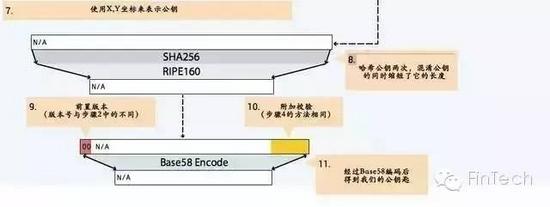
#### 核心技术3：非对称加密算法

    什么是非对称加密？简单来说，它让我们在“加密”和“解密”的过程中分别使用两个密码，两个密码具有非对称的特点：（1）加密时的密码（在区块链中被称为“公钥”）是公开全网可见的，所有人都可以用自己的公钥来加密一段信息（信息的真实性）；（2）解密时的密码（在区块链中被称为“私钥”）是只有信息拥有者才知道的，被加密过的信息只有拥有相应私钥的人才能够解密（信息的安全性）。

    简单的总结：区块链系统内，所有权验证机制的基础是非对称加密算法。常见的非对称加密算法包括RSA、Elgamal、D-H、ECC（椭圆曲线加密算法）等。在非对称加密算法中，如果一个“密钥对”中的两个密钥满足以下两个条件：1、对信息用其中一个密钥加密后，只有用另一个密钥才能解开；2、其中一个密钥公开后，根据公开的密钥别人也无法算出另一个，那么我们就称这个密钥对为非对称密钥对，公开的密钥称为公钥，不公开的密钥称为私钥。在区块链系统的交易中，非对称密钥的基本使用场景有两种：1、公钥对交易信息加密，私钥对交易信息解密。私钥持有人解密后，可以使用收到的价值。2、私钥对信息签名，公钥验证签名。通过公钥签名验证的信息确认为私钥持有人发出。

    我们可以看出，从信任的角度来看，区块链实际上是数学方法解决信任问题的产物。过去，人们解决信任问题可能依靠熟人社会的“老乡”，政党社会的“同志”，传统互联网中的交易平台“支付宝”。而区块链技术中，所有的规则事先都以算法程序的形式表述出来，人们完全不需要知道交易的对手方是“君子”还是“小人”，更不需要求助中心化的第三方机构来进行交易背书，而只需要信任数学算法就可以建立互信。区块链技术的背后，实质上是算法在为人们创造信用，达成共识背书。





#### 核心技术4：脚本

    脚本可以理解为一种可编程的智能合约。如果区块链技术只是为了适应某种特定的交易，那脚本的嵌入就没有必要了，系统可以直接定义完成价值交换活动需要满足的条件。然而，在一个去中心化的环境下，所有的协议都需要提前取得共识，那脚本的引入就显得不可或缺了。有了脚本之后，区块链技术就会使系统有机会去处理一些无法预见到的交易模式，保证了这一技术在未来的应用中不会过时，增加了技术的实用性。

    一个脚本本质上是众多指令的列表，这些指令记录在每一次的价值交换活动中，价值交换活动的接收者（价值的持有人）如何获得这些价值，以及花费掉自己曾收到的留存价值需要满足哪些附加条件。通常，发送价值到目标地址的脚本，要求价值的持有人提供以下两个条件，才能使用自己之前收到的价值：一个公钥，以及一个签名（证明价值的持有者拥有与上述公钥相对应的私钥）。脚本的神奇之处在于，它具有可编程性：（1）它可以灵活改变花费掉留存价值的条件，例如脚本系统可能会同时要求两个私钥、或几个私钥、或无需任何私钥等；（2）它可以灵活的在发送价值时附加一些价值再转移的条件，例如脚本系统可以约定这一笔发送出去的价 值以后只能用于支付中信证券的手续费、或支付给政府等。

区块链的核心部分就是挖矿算法，接来下我们就以比特币的算法为例。它被称为SHA-256，也就是“256位安全散列算法”的缩写，它可接受任何内容的输入，包括文本、数字，或者任何长度的计算机文件。而产生输出称为“哈希”，且每次长度都相同为256位机器代码。同样的输入将得到同样的输出，并不会随机。但如果你对输入进行了微小的改变，则输出会完全不同。而且该算法也被称为单向函数，这就意味着你不能根据输出来计算出输入。你只能对输入进行猜测，然而你猜对的几率是2^256分之一，这几乎是不可能的，换句话该算法是安全的。现在我们已经知道什么是算法，那接下来我们就用一个简单的交易示例来演示区块链是如何运行的。现有Alice和Bob两个人，且我们知道他们俩的比特币余额。我们假设Alice欠Bob2个比特币。Alice如果想要向Bob发送那两枚比特币，则Alice需要向网络中的矿工们广播她想要进行交易的消息。在广播中，Alice将Bob的公共地址、她想要发送的比特币数量，以及数字签名和她公钥提供给矿工们。该签名是通过Alice的私钥生成的，由此矿工们就可以验证那些比特币是属于Alice的，且她也想要进行交易。一旦矿工确定该交易是有效的，则他们就会将该交易与其他交易一起纳入到区块中，并尝试对该区块进行挖矿。且区块中的这一输入是通过SHA-256算法来完成的。而输出则需要从具体的0's数值开始，以便才能被认为有效。具体0's的数值则将根据所谓的“difficulty（难度）”来决定，而“difficulty”也将根据网络中的计算能力进行更改。想要根据最开始要求的0's数值来生成输出哈希值，矿工则需要在区块中添加一个“noncee number（随机数）”。由于输入的很小改变会完全更改输出，所以矿工要不断随机nonces，直到找到一个有效的输出哈希值为止。如果该区块已经被挖掘，则该矿工就会向其他矿工们广播该新的已被挖掘的区块。之后矿工们就开始检查该区块以确保是有效的，之后再将该区块添加到他们的区块链副本中，至此交易就完成了。并且矿工也需要在已完成的区块中纳入前一区块的输出哈希值，以此才能使得所有的区块都相互连接，而区块相互链接的线性链也就被命名为区块链。因为系统是基于信任工作的，所以这一过程尤为重要。

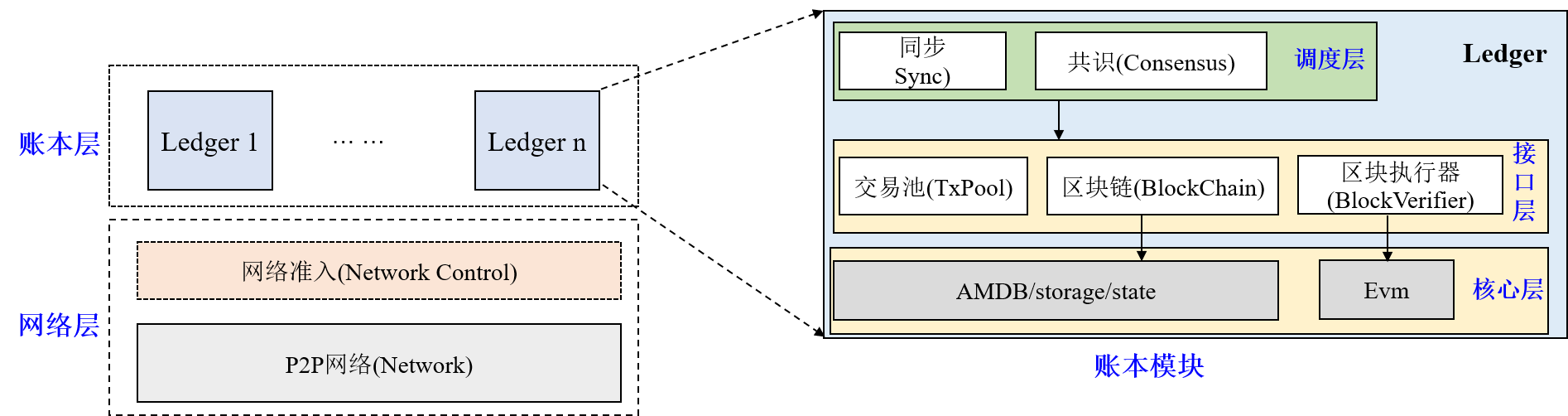
每一位矿工的电脑上都有他自己的区块链副本，且所有人都相信投入计算工作最多的区块链，就将成为最长的区块链。如果一位矿工更改了之前区块中交易，则该区块的输出哈希值也将改变，而这就导致了其后的所有哈希值都会改变（因为区块是通过哈希值链接的）。该矿工就不得不重新开始所有的工作，以便于其他人接收到正确的区块链。因此，如果一位矿工想要作弊，则他的算力就必须要超过网络计算能力的50%，这几乎是不可能的。而网络中类似的攻击也就被称为51%攻击。为了生成区块而使计算机运行的模式被称为工作量证明（PoW）。也存在其他模式列入权益证明（PoS），而该模式下则不需要太多的计算能力，这就意味着只需要相对较少的电力，并且该模式也将扩展到更多的用户。

1. **联盟链与私有链的异同**

联盟区块链是指其共识过程受到预选节点控制的区块链；例如，不妨想象一个有15个金融机构组成的共同体，每个机构都运行着一个节点，而且为了使每个区块生效需要获得其中10个机构的确认（2/3确认）。区块链或许允许每个人都可读取，或者只受限于参与者，或走混合型路线，例如区块的根哈希及其API（应用程序接口）对外公开，API可允许外界用来作有限次数的查询和获取区块链状态的信息。这些区块链可视为“部分去中心化”。  
完全私有区块链（Fully private blockchains）  
完全私有的区块链是指其写入权限仅在一个组织手里的区块链。读取权限或者对外开放，或者被任意程度地进行了限制。相关的应用囊括数据库管理、审计、甚至一个公司，尽管在有些情况下希望它能有公共的可审计性，但在很多的情形下，公共的可读性并非是必须的。

1. **群组架构的好处**

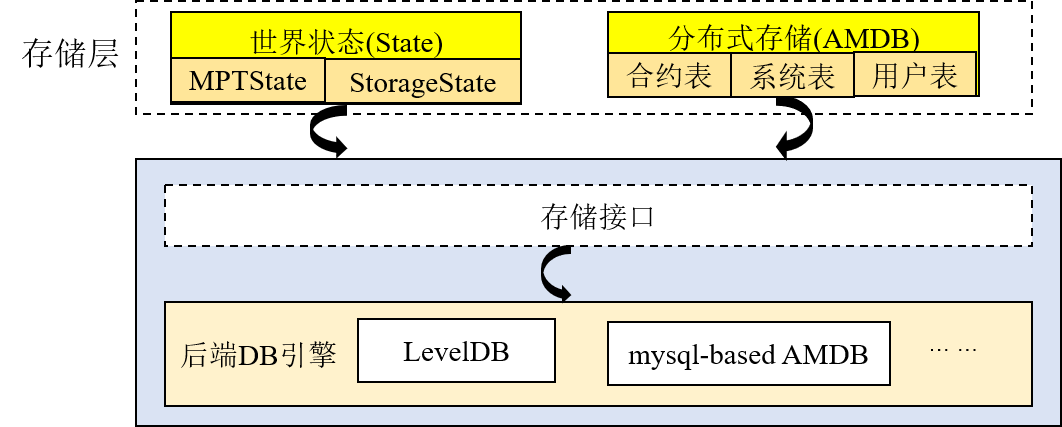
多群组架构中，群组间共享网络，通过[网络准入和账本白名单](https://fisco-bcos-documentation.readthedocs.io/zh_CN/latest/docs/design/security_control/node_management.html)实现各账本间网络消息隔离。



群组间数据隔离，每个群组独立运行各自的共识算法，不同群组可使用不同的共识算法。每个账本模块自底向上主要包括核心层、接口层和调度层三层，这三层相互协作，FISCO BCOS可保证单个群组独立健壮地运行。

核心层负责将群组的[区块](https://fisco-bcos-documentation.readthedocs.io/zh_CN/latest/docs/tutorial/key_concepts.html#id3)数据、区块信息、系统表以及区块执行结果写入底层数据库。

存储分为世界状态(State)和分布式存储(AMDB)两部分，世界状态包括MPTState和StorageState，负责存储交易执行的状态信息，StorageState性能高于MPTState，但不存储区块历史信息；AMDB则向外暴露简单的查询(select)、提交(commit)和更新(update)接口，负责操作合约表、系统表和用户表，具有可插拔特性，后端可支持多种数据库类型，目前支持[RocksDB数据库](https://github.com/facebook/rocksdb)和MySQL[storage](https://fisco-bcos-documentation.readthedocs.io/zh_CN/latest/docs/design/storage/storage.html)。



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

* **交易池(TxPool)**: 与网络层以及调度层交互，负责缓存客户端或者其他节点广播的交易，调度层(主要是同步和共识模块)从交易池中取出交易进行广播或者区块打包；
* **区块链(BlockChain)**: 与核心层和调度层交互，是调度层访问底层存储的唯一入口，调度层(同步、共识模块)可通过区块链接口查询块高、获取指定区块、提交区块；
* **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**：包括Sealer线程和Engine线程，分别负责打包交易、执行共识流程。Sealer线程从交易池(TxPool)取交易，并打包成新区块；Engine线程执行共识流程，共识过程会执行区块，共识成功后，将区块以及区块执行结果提交到区块链(BlockChain)，区块链统一将这些信息写入底层存储，并触发交易池删除上链区块中包含的所有交易、将交易执行结果以回调的形式通知客户端，目前FISCO BCOS主要支持[PBFT](https://fisco-bcos-documentation.readthedocs.io/zh_CN/latest/docs/design/consensus/pbft.html)和[Raft](https://fisco-bcos-documentation.readthedocs.io/zh_CN/latest/docs/design/storage/storage.html)共识算法；
* **同步模块**：负责广播交易和获取最新区块， 考虑到共识过程中，[leader](https://fisco-bcos-documentation.readthedocs.io/zh_CN/latest/docs/design/consensus/pbft.html#id1)负责打包区块，而leader随时有可能切换，因此必须保证客户端的交易尽可能发送到每个区块链节点，节点收到新交易后，同步模块将这些新交易广播给所有其他节点；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。

1. **分布式存储的优势**

**1. 高性能**

一个具有高性能的分布式存户通常能够高效地管理读缓存和写缓存，并且支持自动的分级存储。分布式存储通过将热点区域内数据映射到高速存储中，来提高系统响应速度;一旦这些区域不再是热点，那么存储系统会将它们移出高速存储。而写缓存技术则可使配合高速存储来明显改变整体存储的性能，按照一定的策略，先将数据写入高速存储，再在适当的时间进行同步落盘

**2. 支持分级存储**

由于通过网络进行松耦合链接，分布式存储允许高速存储和低速存储分开部署，或者任意比例混布。在不可预测的业务环境或者敏捷应用情况下，分层存储的优势可以发挥到最佳。解决了目前缓存分层存储最大的问题是当性能池读不命中后，从冷池提取数据的粒度太大，导致延迟高，从而给造成整体的性能的抖动的问题。

**3. 多副本的一致性**

与传统的存储架构使用RAID模式来保证数据的可靠性不同，分布式存储采用了多副本备份机制。在存储数据之前，分布式存储对数据进行了分片，分片后的数据按照一定的规则保存在集群节点上。为了保证多个数据副本之间的一致性，分布式存储通常采用的是一个副本写入，多个副本读取的强一致性技术，使用镜像、条带、分布式校验等方式满足租户对于可靠性不同的需求。在读取数据失败的时候，系统可以通过从其他副本读取数据，重新写入该副本进行恢复，从而保证副本的总数固定;当数据长时间处于不一致状态时，系统会自动数据重建恢复，同时租户可设定数据恢复的带宽规则，最小化对业务的影响。

**4. 容灾与备份**

在分布式存储的容灾中，一个重要的手段就是多时间点快照技术，使得用户生产系统能够实现一定时间间隔下的各版本数据的保存。特别值得一提的是，多时间点快照技术支持同时提取多个时间点样本同时恢复，这对于很多逻辑错误的灾难定位十分有用，如果用户有多台服务器或虚拟机可以用作系统恢复，通过比照和分析，可以快速找到哪个时间点才是需要回复的时间点，降低了故障定位的难度，缩短了定位时间。这个功能还非常有利于进行故障重现，从而进行分析和研究，避免灾难在未来再次发生。多副本技术，数据条带化放置，多时间点快照和周期增量复制等技术为分布式存储的高可靠性提供了保障。

**5. 弹性扩展**

得益于合理的分布式架构，分布式存储可预估并且弹性扩展计算、存储容量和性能。分布式存储的水平扩展有以下几个特性：

1) 节点扩展后，旧数据会自动迁移到新节点，实现负载均衡，避免单点过热的情况出现;

2) 水平扩展只需要将新节点和原有集群连接到同一网络，整个过程不会对业务造成影响;

3) 当节点被添加到集群，集群系统的整体容量和性能也随之线性扩展，此后新节点的资源就会被管理平台接管，被用于分配或者回收。

**6. 存储系统标准化**

随着分布式存储的发展，存储行业的标准化进程也不断推进，分布式存储优先采用行业标准接口(SMI-S或OpenStack Cinder)进行存储接入。在平台层面，通过将异构存储资源进行抽象化，将传统的存储设备级的操作封装成面向存储资源的操作，从而简化异构存储基础架构的操作，以实现存储资源的集中管理，并能够自动执行创建、变更、回收等整个存储生命周期流程。基于异构存储整合的功能，用户可以实现跨不同品牌、介质地实现容灾，如用中低端阵列为高端阵列容灾，用不同磁盘阵列为闪存阵列容灾等等，从侧面降低了存储采购和管理成本。

1. **并行计算**

问题是，更高的处理器性能意味着更大的功耗和更多发热量，即使你在芯片烧焦之前找到了一种方法来解决掉发热量过大的问题，延续这种趋势也会造成成本过高以及对环境的破坏过大。

另一种实现更快计算性能的方法也已经有一段时间的历史了。这种方法并不是让处理器突破速度极限，而是采用更多的处理器。大型计算机与服务器长期以来一直采用众多的处理器来处理繁重的工作。然而芯片技术的进步让人们能够在单一芯片中集成多个处理器，这种方法不仅效率更高而且还便宜很多。当今，高性能计算的方式是将计算任务进行分割，然后交由诸多处理器核心进行处理。假如是个人计算机，这意味着不仅CPU能够拥有大量核心，而且图形处理器(GPU)也能够包含数十个，甚至是数以百计的核心。

然而，多处理器硬件所带来的问题是软件上的巨大挑战。从20世纪40年代现代计算的初期开始，各种程序就一直是设计成顺序执行的方式。美国国防部高级研究计划署提供了大部分资金用来开发包含大量处理器的系统，并取得了一定的成功。在这类系统中，计算任务被分割成诸多细小部分，这些细小的任务可以同时运行，这些系统旨在通过这种方式来解决计算难题。然而，这些大规模并行系统没有能够在商业市场中生根发芽。

原因之一便是最常见的计算难题以及用于解决这些难题的算法不能很好地适应这种「分割」的处理方式。而顺序的想法似乎已经在我们的大脑中根深蒂固。神经系统科学家Jill Bolte Taylor指出，大脑的右半部分能够处理感官信号，它所做的是并行处理。而大脑的左半部分负责分析想法，「其运行方式就像是一个串行处理器。」不管是好还是坏，编程都是一项左脑的活动。

并行方法在数学上的最大障碍便是，许多进程是递归的：每一步都依赖上一步的结果。考虑一下找出两个整数的最大公约数这一简单问题。解决这一问题的标准方法是欧几里德算法，这种算法已经有2,000多年的历史，所使用的是重复减法。

例如，如果你想要找出2,987与1,751的最大公约数，那么可以先用2,987来减1,751。重复地减掉差数(视需要颠倒顺序，以防出现负数)，直到结果为零。在这一例子中，这两个数的最大公约数是103。这是一个完美而高效的方法，但是它是一个天生的串行式方法，因为每一次减法都依赖于上一次的结果。

串行思想虽然占据主导地位，但是也有例外。最好的例子便是图形。在图形中，一个非常简单、常见以及典型的需求就是旋转图像。如果你还记得一些三角学的话，你可能会想起一个简单的公式，将某个点逆时针旋转一个角度& Theta：

其重点在于，每个点的处理都可以独立于所有其它点之外。如果你的处理器数目与点数一样多的话，那么整个转换过程即可在一个大规模并行运算中完成计算。诸多更加复杂的图形任务也是如此。

图形任务的并行友好性特点导致了早期人们在图形处理器(GPU)中融入多处理器架构。NVIDIA®(英伟达™)顶级Tesla GPU目前包含240个处理器核心。虽然这些核心并不像CPU处理器那样灵活，但是它们在特定任务上却更胜一筹，例如诸多计算密集型难题当中重要的向量运算。

无论是针对CPU还是GPU来说，能够有效利用大量核心的软件仍然是个难题，但是情况已经变得越来越好。NVIDIA®(英伟达™)凭借着CUDA™并行编程模型以及C语言扩展充当开路先锋，该模型让通用计算能够在NVIDIA®(英伟达™)GPU上运行，而其C语言扩展则消除了对这种处理器进行编程的门槛。因此，开发人员能够分别通过CUDA™工具包以及PGI的CUDA™ Fortran编译器来利用C、C++以及Fortran语言对NVIDIA®(英伟达™)CUDA™ GPU进行编程。同时还能够利用诸多驱动程序级的API，例如OpenCL以及DirectCompute。

软件开发人员所面临的最大难题之一便是在现有应用程序上实现更高性能以及开发出更多全新的计算密集型应用程序。无论是选择多核CPU还是核群GPU，除了考虑将其应用程序实现并行化以外，他们别无选择。根据近几年的发展，CUDA™并行编程模型已经成为一款公认的「更简单的」并行编程方式(它仍然不简单，但是CUDA™的确使特定操作变得更加简单)。而且，与CPU相比，GPU还能够提供巨大的性能优势。因此这两大元素的有机结合为开发人员提供了一条开发更多创新应用程序的途径。

**参考文献：**

[1] 2013 BitCoin meets Google Trends and Wikipedia: Quantifying the relationship between phenomena of the Internet era  
[2] 2013 “When perhaps the real problem is money itself!“: the practical materiality of Bitcoin  
[3] 2015 Decentralizing Privacy: Using Blockchain to Protect Personal Data  
[4] 2014 Majority Is Not Enough: Bitcoin Mining Is Vulnerable  
[6] 2016 Blockchains and Smart Contracts for the Internet of Things  
[7] 2015 The economics of Bitcoin and similar private digital currencies  
[8] 2014 Mixcoin: Anonymity for Bitcoin with Accountable Mixes  
[9] 2017 A 2017 Horizon Scan of Emerging Issues for Global Conservation and Biological Diversity