|  |
| --- |
| 1. **本课题主要研究内容分为两部分：**   1）茶叶产业链溯源系统与区块链系统结合。  茶叶产业链溯源系统划分为五大管理模块，分别为种植管理、采收加工管理、销售管理、仓库物流管理和系统管理。各大模块下均负责其他详细的小模块管理功能，如下图1所示。    图1：茶叶溯源模块图  在茶叶产业链溯源系统中，每一样商品都需要被数字化，包括身份标识及参数信息，信息维护者方可全面具体的维护产品信息。通过商品唯一的身份标识ID，随着业务流程的进展，使得商品能在不同的管理模块间保持溯源信息不断裂。如下图2 为系统的主要业务流程，其中涉及到企业用户、系统管理者、消费者等参与角色。    图2：业务流程  Fabric架构设计中的区块链网络中有两种类型的节点：Peer节点和Orderer节点，如下图3所示：  图片1  图3：Fabric架构  Peer节点：Chaincode部署在Peer节点上，它对账本进行读写操作。一个Peer节点可以充当多种角色，如背书者Endorser，提交者Committer。一个区块链网络中会有多个Peer节点。  Orderer节点：对交易进行共识排序，按照区块生成策略，将交易打包到一起生成区块，发送给Peer节点。  一个Fabric交易流程大致如下：客户端应用程序利用任意SDK构造交易提案propose，该提案是一个调用智能合约功能函数的请求，用来确认哪些数据可以读取或写入账本。客户端把交易提案发送给一个或多个Peer节点。背书节点endorser收到交易提案后，验证签名并确定提交者是否有权执行操作（此时还不会更新账本），然后返回给客户端模拟交易结果。客户端验证背书节点签名并比较各节点返回的提案结果，判断提案结果是否一致以及是否参照指定的背书策略执行。而后把收到的各个背书节点的应答打包到一起组成一个交易并签名、发送给Orderers节点。Orderers对接收到的交易进行共识排序，然后按照区块生成策略将一批交易打包到一起生成新的区块。  茶叶产业链溯源系统与Fabric框架相结合，如下图4所示：  C:\Users\Administrator\Desktop\图片2.png图片2  图4：结合图  企业对溯源数据的增删改操作（未生成区块前）依旧在溯源平台上进行。假设目前商品处于种植管理模块，则Fabric通过智能合约的方式对产品当前担责节点开放权限，保证数据维护的有序性和可靠性，防止非相关节点违规操作。被授予权限的参与者（企业）维护信息时需以私钥连接到网络中，而一旦参与者发起产品转移，比如茶叶从存储仓库转移到加工场，系统通过内嵌的智能合约进行相关的审核，通过后将参与者之前提供的信息打包录入区块链中。由此可见，数据打包成区块的时间节点是可控的，从而保证内部溯源数据不会与外部溯源数据被封装进同一个区块，造成区块信息访问权限发生矛盾（为保护企业隐私，内部溯源信息的访问权限必然高于外部溯源信息）。溯源数据在商品最终生成可开放给消费者的二维码或RFID之前，属于内部溯源数据，只能由企业内部进行访问。  2）拜占庭容错共识算法在系统中的性能分析对比与大量数据录入时系统生成区块的性能优化  一般情况下，在系统中一旦参与者发起产品转移，将会有区块产生，而区块加入到区块链之前，则需要经由框架选用的共识机制达成共识才行。Fabric在共识机制上可拔插的架构设计，让我们可以将目前主流的共识算法分别应用其中，并进行性能分析和对比。Fabric框架的共识机制默认为PBFT算法，因此在把Fabric框架和茶叶产业链溯源系统结合在一起之后，我们无需做任何修改即可使用PBFT算法。Tenermint算法是PBFT算法的简化版，PBFT为3阶段共识过程的异步协议，Tenermint则是2阶段共识过程的弱同步协议。RPCA算法使用一种子网络内部相互信任并构成大网络的方案。本课题打算使用这三种算法进行性能分析与对比，不过我们无需分开进行测试，Fabric框架中的Orderers节点提供的排序服务支持多通道功能（如下图5所示），这意味着我们可以同时使用三种共识算法在所有节点里面生成三条区块链，并且彼此之间的消息沟通互不干涉，少了每次测试都要重置系统的麻烦，这有助于提高测试的效率。  图片3  图5：Ordering提供的多通道功能  有些企业在刚入驻系统时，需要录入大量的历史数据，如果按一般情况的做法去生成区块，那么最终完成所有的区块生成将要花费大量时间。在此，我们可以参考一下RPCA算法的思路：RPCA算法为了降低节点间同步沟通的成本，使用了一种子网络内部相互信任，由这些内部信任的子网络构成大网络的方案，只要子网络之间的连接度不低于一个阈值即可。由于Fabric具备成员管理服务，成员必须被许可才能加入网络，因此在本系统中我们无需像RPCA那样长期维护一个可信任的子网络。参考RPCA算法的设计思路，并结合茶叶产业链溯源系统数据的实际录入情况，运用Fabric框架的设计，本课题在此提出如下BFT算法：每个Peer节点维护一张状态表，标明自己是处于无请求状态、已发送请求状态（等待生成区块）或是大数据量录入请求状态这三个状态中的一个。企业用户使用的溯源平台和客户端节点是分开的，所有企业使用同一个溯源平台，只是通过平台后端的权限管理将不同企业区分开来。而客户端节点则是企业在自己服务器上搭建并接入Fabric系统的Peer节点。因此，企业在导入大量数据的时候，首先是通过溯源平台进行数据导入，再经由溯源平台将第一条溯源数据以及相关状态信息发送给企业的客户端节点（暂时称为节点A），表明本节点即将进行大量的生成区块的请求，如图4和图6所示。Fabric框架提供的身份管理服务保障网络中节点的可靠性，因此我们无需像RPCA算法一样让每个节点维护一个可信列表，然而相对的，在节点处于大数据量录入请求状态时，我们需要让每一个节点维护一个相同的状态列表，而这个列表中的每个节点的状态均为无请求状态（除了节点A）。RPCA算法的共识过程分为两步，先进行交易共识形成交易集，打包成区块后再共识。由于维护的状态列表中除了节点A外，其余节点都处于无请求状态，也就是说只有节点A在发生交易，因此我们可以省去交易共识这一步，直接进行区块共识。  图片4  图6：大数据量的第一次请求  状态列表的生成：节点A接收到第一个交易请求和状态数据后，将自身修改为大数据量录入请求状态，并将本节点加入状态列表中。之后节点A将交易和节点状态向全网进行广播，等待接收超过80%的状态为无请求的节点，并将这些节点加入状态列表中，如图7所示。在确定好了状态列表后，节点A通过Orderers节点将状态列表发送给列表中的每一个节点。区块打包，再共识：第一次交易请求生成状态列表后，每个节点开始打包新的区块，打包区块的过程如下：把当前区块号、交易的Hash、父区块Hash、当前时间戳等内容放到一起，计算一个区块哈希。每个节点广播自己得出的区块哈希到状态列表中的节点。节点收集到它所有状态列表中节点广播过来的区块哈希后，结合自己生成的区块哈希，对每个区块哈希计算一个比例，如果某一哈希的比例超过一个阈值（一般是80%），则认为这个哈希是共识通过的区块哈希。如果自己的哈希与之相同，则说明自己打包的区块得到了确认，是新的被共识过的区块，直接存到本地，并且更新状态。如果自己的哈希与共识通过的哈希不同，那么重新开始共识过程，直到满足条件。至此，一个区块的共识过程结束，开启下一轮共识过程。从第二轮共识开始，继续使用第一轮的状态列表，并直接进行区块打包共识。Fabric采用模块化架构把交易处理划分为3个阶段：通过Chaincode进行分布式业务逻辑处理和协商（endorsers）；交易排序(orderders)；交易的验证和提交(committers)。这样划分带来的好处：不同的阶段由不同的节点（角色endorsers, orderders, committers）参与，不需要全网的节点都参与。网络的性能和扩展性得到优化。  图片5  图7：状态列表  几种导致大数据量数据录入失败或暂缓的情况:  1.新的节点加入网络使得状态列表里面的节点数量不再超过全网络的80%。  2.状态列表中的某个或某些节点由无请求状态变成已请求状态  3.非状态列表中(可能是新添加的节点，也可能是先前状态为已发送请求状态转变成无请求状态的节点)的某个或某些节点由无请求状态变成已请求状态  4.状态列表中的某个或某些节点由无请求状态变成大数据量录入请求状态  5.非状态列表中(可能是新添加的节点，也可能是先前状态为已发送请求状态转变成无请求状态的节点)的某个或某些节点由无请求状态变成大数据量录入请求状态  6.状态列表中一定数量的节点发生故障使得节点数量不再超过全网络的80%。  针对上面提到的六种情况，对应的处理措施如下：  第1、6种情况使得状态列表失效从而导致区块共识失败，属于数据录入失败的情况，需要从新生成状态列表。  第2种情况，可让变成已请求状态的节点继续参与到状态列表构成的网络中。改变了状态的节点（假设只有一个，称为节点B，多个节点情况一样）向状态列表中的其他节点广播自己的交易，其他节点将收到的交易暂时缓存起来，等待这一轮节点A的区块共识结束。在下一轮节点A向状态列表中的节点发送自己的交易后，其余每个节点各自将节点A的交易和节点B的交易打包在一起。我们让节点B在广播了自己的交易后不进行任何等待反馈，而是在交易打包后直接进行区块打包并共识，如果成功，则进入下一轮共识，如果失败，则节点B继续向状态列表的其他节点发送自己的交易，直到成功为止。成功后，节点B的状态从已请求状态变成无请求状态。  第3种情况，在节点A进行完本轮的共识过程后，可将改变状态的节点加入状态列表中，形成新的状态网络，再按照上述第2种情况的解决方法去处理。  第4、5种情况，状态列表构成的网络中将有超过一个节点处于大数据量录入状态，因此交易共识这一步无法省略。在新一轮共识开始后，每个节点尽可能多的收集所能收集到的需要共识的交易，并放到“候选集”里面。每个节点对状态列表中的“候选集”做一个并集，并对每一个交易进行投票。所有投票超过80%的交易被放到共识过的交易集里面。之后就是区块打包，再共识。  **2.拟解决的关键问题**  1）不同拜占庭容错算法在结合了区块链技术的茶叶产业链溯源系统中的性能分析和对比。  2）企业在茶叶产业链溯源系统中录入大量数据时生成区块的性能优化。 |