华南农业大学

硕士研究生中期考核与学位论文开题报告

学号: 20173078011

姓名: 苏锐佳

学院: 数学与信息学院

专业(领域): 计算机技术

研究方向: 网络安全

导师姓名: 杜治国

攻读学位: 工程专业学位硕士

|  |  |  |  |
| --- | --- | --- | --- |
| 论文题目 | 基于Fabric框架的茶叶产业链溯源系统在大数据量录入时共识机制的优化 | | |
| 选题来源 | 其他 | 论文类型 | 基础研究 |
| 开题日期 | 2017-09-20 | 涉密 | 否 |
| **一、立题依据（包括研究目的、意义、国内外研究现状和发展趋势，需结合科学研究发展趋势来论述科学意义；或结合国民经济和社会发展中迫切需要解决的关键科技问题来论述其应用前景。附主要参考文献目录）**  1、研究目的和意义：  茶叶质量安全问题，如农残超标，如同其他食品行业的安全问题一样，一直是亟待解决的痛点。茶叶行业有其特殊性，许多地区都会生产属于其本区域的特色名茶，比如西湖龙井、凤凰单枞等，一旦发生质量安全问题，很多时候影响的将是整个区域大范围的茶行。茶叶溯源系统记录着茶叶从生产到销售整个流程的所有必要数据，使得发生质量安全问题时，能快速定位到问题的源头以及相关负责人，及时召回问题茶叶，遏制问题蔓延的势头，这是解决目前茶叶安全窘迫现状的有效方法之一。然而对于传统的溯源系统来说，数据都是由统一的中央数据库发出，每个流程的数据在存储过程中以及日后维护时都有被篡改的可能，造成数据的真实性存在质疑，即使数据上传到权威政府机构，也同样带来中心权利过大化的问题。区块链的出现则很好地解决了数据中心化和易篡改等问题，其去中心化、不可篡改性、开发透明、机器自治和匿名性的特性，弥补了传统溯源系统的缺点。  区块链系统首先是一个分布式系统，因此在区块链技术框架的工作流程中，共识机制是一个核心问题，即如何解决Leslie Lamport等人提出的拜占庭将军问题。随着技术的发展，区块链系统在解决拜占庭将军问题实现共识这方面，出现了概率性算法PoW、确定性算法RPCA和PBFT算法等。茶叶产业链溯源在与区块链技术相结合后，共识机制这一模块在区块链技术框架的工作流程中会成为最大的性能瓶颈，特别是前期企业刚入驻平台，往系统中大量录入以前的溯源数据时。PoW算法在电子加密货币系统中每10分钟左右生成一个区块，在以太坊中每10秒左右生成一个区块，而RPCA算法和PBFT算法为3-6秒生成一个共识区块。由于数字货币在溯源系统中用处不大，故选用RPCA、PBFT等算法。即便如此，在大量数据前面，这类算法依旧无法满足系统高性能的要求，所以根据实际的场景，对现有的高效算法诸如PBFT、dBFT进行改进就显得非常重要了。  茶叶产业链溯源系统中会涉及到权限角色地位不等的各机关和企业，由此选用联盟链。联盟链与公有链的区别不仅在于系统中心化和参与者的身份，还有共识机制与激励机制上的差异。选用联盟链，意味着区块链系统中的信任机制不再是PoW或PoS，可以通过降低去中心化长度的方法，减少参与共识机制的节点以克服挖矿过程的资源浪费问题；同时，维护茶叶产业链溯源系统是联盟链中各参与节点的责任，故区块链系统的运行无需激励机制，可以不依赖数字货币，简化记录账簿，降低系统运行所需带宽，进一步提高系统性能。  2、国内外研究现状  区块链从出现到现在经历了快速的发展，1.0阶段：区块链技术主要用于支持以比特币为代表的数字货币，通过支持账户间转账、付款等支付交易操作，使得交易双方无需借助第三方机构的保证即可实现数字货币安全可信的交易。2.0阶段：区块链技术的应用场景，从单一的数字货币交易延伸到图灵完备的通用计算领域。用户不再受限于仅能使用比特币脚本所支持的简单逻辑，而是可以自行设计任意复杂的合约逻辑。出现以以太坊为代表的，将数字货币与智能合约相结合的平台，利用区块链作为底层技术，在平台上运行任意复杂的分布式程序Dapp。3.0阶段：出现以超级账本为代表的项目，进一步引入权限控制和安全保障，将区块链技术从经济领域延伸到社会管理、慈善公益、文化娱乐、医疗健康等社会各个领域中。  在1.0阶段，应用于区块链系统中的共识算法是PoW这类概率性算法，共识结果是临时的，随着时间的推移或某种强化，共识结果被推翻的概率越来越小，成为事实上的最终结果。而且其高度的去中心化，并且平均每10分钟生产一个区块。到了3.0阶段，共识算法则是PBFT这类确定性算法，且不再高度去中心化，同时形成了联盟链。  目前，世界范围内对区块链技术的研究和应用主要集中在金融领域，很多商业银行、金融机构甚至政府都在大力支持区块链相关研究。世界经济论坛报告指出，目前有超过20个国家投资区块链相关的技术领域，80%的银行在2017年开始实施一些区块链分布式账本有关的项目。  2015年12月，由开源世界的旗舰组织Linux基金会牵头，30家初始企业成员（包括IBM、Accenture、Intel、J.P.Morgan、R3、DAH、DTCC、FUJITSU、HITACHI、SWIFT、Cisco等），共同宣布了Hyperledger联合项目成立。超级账本项目为透明、公开、去中心化的企业级分布式账本技术提供开源参考实现并推动区块链和分布式账本相关协议、规范和标准的发展。Fabric是最早加入到超级账本项目中的顶级项目，由IBM、DAH等企业于2015年底提交到社区。该项目的定位是面向企业的分布式账本平台，创新地引入了权限管理支持，设计上支持可插拔、可扩展，是首个面向联盟链场景的开源项目。  **本课题针对前人的研究进行以下两部分的探索：**   1. 本课题将使用超级账本Hyperledger中的顶级项目Fabric与茶叶产业链溯源系统相结合。Fabric具备完备的权限控制和安全保障、高性能、可扩展、较低信任要求的优势，最重要的一点是，其模块化设计，可插拔的架构设计，使得我们在共识机制这块可以根据实际情况选择替换，从而让我们可以对现有的各类主流优秀的拜占庭容错算法在茶叶产业链溯源系统中的性能进行分析对比。 2. 根据Fabric项目与茶叶产业链溯源系统结合后的批量数据的录入情况，我们可以对上一步在性能对比中处于优势的算法进行优化。得益于Fabric项目中的Orderer节点提供的排序服务，且这排序服务支持多通道，所以我们并不一开始就让所有节点参与到共识之中，而是按照一定策略和条件，选择相应数量的节点参与到共识，最后再广播通知剩余节点参与共识。   参考文献：  [1] Androulaki E, Barger A, Bortnikov V, et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains[J]. 2018.  [2] Sousa J, Bessani A, Vukolić M. A Byzantine Fault-Tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform[J]. 2017:1-2.  [3] 李亚楠. 基于区块链的数据存储应用研究[D].北京交通大学,2018.  [4] Schneider F B. Implementing fault-tolerant services using the state machine approach: a tutorial[J]. Acm Computing Surveys, 1990, 22(4):299-319.  [5] Castro M, Liskov B. Practical byzantine fault tolerance and proactive recovery[M]. ACM, 2002. [6] Camenisch J, Herreweghen E V. Design and implementation of the idemix anonymous credential system[C]// 2002:21-30.  [7] Kemme B, Alonso G. A new approach to developing and implementing eager database replication protocols[J]. Acm Transactions on Database Systems, 2000, 25(3):333-379.  [8] Kemme B, Alonso G. Alonso, G.: A New Approach to Developing and Implementing Eager Database Replication Protocols. ACM Transactions on Database Systems 25(3), 333-379[J]. Acm Transactions on Database Systems, 2000, 25(3).  [9] Yin J, Martin J P, Venkataramani A, et al. Separating Agreement from Execution for Byzantine Fault Tolerant Services[J]. Acm Sigops Operating Systems Review, 2003, 37(5):253-267.  [10] Castro M, Liskov B. Practical Byzantine fault tolerance[C]// Symposium on Operating Systems Design & Implementation. ACM, 1999:173-186.  [11] Dijkstra E W. On-the-fly garbage collection: an exercise in cooperation[J]. Communications of the Acm, 1978, 21(11):966-975.  [12] Fischer M J. Impossibility of distributed consensus with one faulty process[J]. Acm Tocs, 1985, 32(2):374-382.  [13] Schneider F B. Implementing fault-tolerant services using the state machine approach: a tutorial[J]. Acm Computing Surveys, 1990, 22(4):299-319.  [14] Li M, Tamir Y. Practical Byzantine Fault Tolerance Using Fewer than 3f+1 Active Replicas.[C]// ISCA, International Conference on Parallel and Distributed Computing Systems, September 15-17, 2004, the Canterbury Hotel, San Francisco, California, Usa. DBLP, 2004:241-247.  [15] Benini L, Bogliolo A, Micheli G D. A survey of design techniques for system-level dynamic power management. IEEE Trans Very Large Scale Integr (VLSI) Syst[J]. IEEE Transactions on Very Large Scale Integration Systems, 2000, 8(3):299-316.  [16] 汪登, 曾小珊, 白倩兰,等. 基于区块链的食品安全溯源技术[J]. 大数据时代, 2018(3).  [17] Swan M. Blockchain: Blueprint for a New Economy[M]// Blockchain : blueprint for a new economy. O'Reilly, 2015.  [18] Swan M. Blockchain Thinking : The Brain as a Decentralized Autonomous Corporation [Commentary][J]. IEEE Technology & Society Magazine, 2015, 34(4):41-52.  [19]Calcaterra C, Kaal W A, Andrei V. Semada Technical Whitepaper - Blockchain Infrastructure for Measuring Domain Specific Reputation in Autonomous Decentralized and Anonymous Systems[J]. Social Science Electronic Publishing, 2018.  [20]Yuan Yong, Fei-Yue Wang. 区块链技术发展现状与展望[J]. Zidonghua Xuebao/acta Automatica Sinica, 2016, 42(4):481-494.  [21] Peters G W, Panayi E. Understanding Modern Banking Ledgers Through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money[M]// Banking Beyond Banks and Money. Springer International Publishing, 2016.  [22]郭珊珊. 供应链的可信溯源查询在区块链上的实现[D]. 大连海事大学, 2017.  [23]吴霜, 喻朝新. 物联网和区块链技术在农产品溯源上的应用[J]. 电信工程技术与标准化, 2018(6).  [24]Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[J]. Consulted, 2008.  [25]Christidis K, Devetsikiotis M. Blockchains and Smart Contracts for the Internet of Things[J]. IEEE Access, 2016, 4:2292-2303.  [26]Tian F. An agri-food supply chain traceability system for China based on RFID & blockchain technology[C]// International Conference on Service Systems and Service Management. IEEE, 2016:1-6.  [27]Gandino F, Montrucchio B, Rebaudengo M, et al. Analysis of an RFID-based Information System for Tracking and Tracing in an Agri-Food chain[C]// Rfid Eurasia, 2007. IEEE, 2007:1-6. | | | |

|  |
| --- |
| **二、研究内容和目标（说明课题的具体研究内容，研究目标和效果，以及拟解决的关键科学问题。此部分为重点阐述内容）**   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交易流程导致如下：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。 |

|  |
| --- |
| **三、研究方案设计及可行性分析（包括：研究方法，技术路线，理论分析、计算、实验方法和步骤及其可行性）**  **1、研究方法：**  在课题的设计最初阶段通过文献研究法、调查法不断积累相关知识，拓展对各类共识算法的认知，进而粗略设计出整个方案的框架。然后继续通过文献研究法、实验法、经验总结法对方案进行具体化，优化各部分框架。再通过文献研究法和实验法、经验总结法完成各部分的设计，实现各模块的有机结合。  **2、技术路线：**  系统实现：由于主流的容错算法在网上都能找到现成实现，为了做性能分析和对比，故需先结合Fabric项目与茶叶产业链溯源系统  性能分析与对比：利用Fabric项目的模块化设计和可拔插架构，将RPCA、PBFT等主流的容错算法分别应用于系统中，在相同数据集上进行测试，并对结果进行分析和对比  系统性能优化：优化一类特殊的情况，即在大量数据录入而引起区块生成时间过长造成系统性能下降 |

|  |
| --- |
| **四、本研究课题的特色与新颖之处**  本课题是区块链与茶叶产业链溯源系统的结合以及共识算法在系统中的应用与优化，与现有的技术和研究成果相比，特色先进性体现在以下方面：   1. 将区块链技术有机地结合进茶叶产业链溯源系统中，解决传统溯源系统的缺点和弊端。 2. 解决在大量数据录入而引起区块生成时间过长造成系统性能低下的问题。 |
| **五、研究基础与工作条件（1.与本项目相关的研究工作积累基础 2.包括已具备的实验条件，尚缺少的实验条件和拟解决途径）**  **1、工作积累基础**  已实现了完整的茶叶产业链溯源系统，并有多家企业已入驻。  **2、实验条件**  1）开发工具：Laravel、NodeJS、VueJS、Fabric等   1. 开发环境：CentOS 7.5、windows   3）开发语言：NodeJS、PHP、JavaScript  4）服务器：CentOS 7.5  5）数据库：Mysql、Level DB  **3、尚欠缺的条件** |

**学位论文工作计划**

|  |  |  |
| --- | --- | --- |
| 时间 | 研究内容 | 预期效果 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**经费开支情况表**

|  |  |  |
| --- | --- | --- |
| **开支项目** | **数量** | **金额(万元)** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| **导师意见**（根据该生的政治表现、思想道德、学习成绩、学习态度、科研能力、遵纪守法和执行学校规章制度等提出是否同意继续攻读硕士学位）  导师签名： 日 期： 年 月 日 |
| **中期考核意见**（优秀、良好、及格、不及格）  **开题报告指导小组意见**（包括对以下各参考项目的评价、提出开题报告是否通过的意见。）  请在相应栏目内画“√”   1. 选题与导师研究方向符合度 符合( ) 一般( ) 不符合( ) 2. 选题与专业（领域）符合度 符合( ) 一般( ) 不符合( ) 3. 选题的新颖性 强( ) 一般( ) 不明显( ) 4. 研究的可行性 可行( ) 一般( ) 不可行( ) 5. 研究的工作量 充足( ) 一般( ) 不足( ) 6. 总体评价 优( ) 良( ) 中( ) 差( )   对开题报告是否通过的意见： |
| 小组成员签名：  日 期： 年 月 日 |
| **学科意见**（请对开题报告的选题与专业符合度作出评价，并提出是否同意开题的意见。）  学科带头人签名： 日 期： 年 月 日 |
| **学院意见：**  学院负责人签名： 学院公章： 日 期： 年 月 日 |

硕士研究生成绩单

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学号 |  | | 姓名 | | |  | | | | 性别 | 男 |
| 学院 | 数学与信息学院 | | | 专业 | | 计算机技术 | | | | 学制 | 2 |
| 导师 | 王美华 | | | | | | | | 入学日期 | | 2016-09-08 |
| 类别 | 课程名称 | | | | 学时 | | 学分 | 学期 | | 成绩 | 备注 |
| 学  位  课  程 |  | | | |  | |  |  | |  |  |
|  | | | |  | |  |  | |  |  |
|  | | | |  | |  |  | |  |  |
|  | | | |  | |  |  | |  |  |
|  | | | |  | |  |  | |  |  |
|  | | | |  | |  |  | |  |  |
| 非  学  位  课  程 |  | | | |  | |  |  | |  |  |
|  | | | |  | |  |  | |  |  |
|  | | | |  | |  |  | |  |  |
|  | | | |  | |  |  | |  |  |
|  | | | |  | |  |  | |  |  |
|  | | | |  | |  |  | |  |  |
| 总学分 | |  | | | | | 学位课学分 | | |  | |

打印日期：2017-09-22

、