

# 基于 Hyperledger Fabric 的汽车供应链系统的方案设计与性能分析

林旭丹<sup>1</sup> 鲍士兼<sup>2</sup> 赵立昕<sup>3</sup> 赵成林<sup>1</sup>

1 北京邮电大学信息与通信工程学院 北京 100876

2 中国国际工程咨询有限公司 北京 100044

3 中国移动通信集团河南有限公司 郑州 450008

**摘要** 当前的汽车供应链体系一般采用中心化的管理模式,供应链中上下游企业间数据交互困难,信息不对称,因此造成了供应链运行效率低下。另外,信息的不透明也导致了供应链中的信任问题。为此,文中引入区块链技术,采用超级账本(Hyperledger Fabric)作为开发框架,设计了一个基于区块链的汽车供应链原型系统,旨在为多方企业提供安全可信的交易服务。该系统具备访问控制和数据透明、可追溯且不可篡改等特点,同时通过设计的多通道架构,实现了企业合作中的隐私隔离。文中使用 docker 技术构建了实验环境,对功能接口进行了测试,通过对原型系统吞吐量性能进行实验分析,验证了该原型系统的可行性。文中引入“区块链+”的模式为传统汽车供应链的升级与转型提供了新的思路。

**关键词:** 区块链应用;供应链;Hyperledger fabric;隐私隔离

**中图法分类号** TP311

## Design and Performance Analysis of Automotive Supply Chain System Based on Hyperledger Fabric

LIN Xu-dan<sup>1</sup>, BAO Shi-jian<sup>2</sup>, ZHAO Li-xin<sup>3</sup> and ZHAO Chen-lin<sup>1</sup>

1 School of information and communication engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

2 China International Engineering Consulting Co., Ltd, Beijing 100044, China

3 Henan Co., Ltd, China Mobile Communications Group, Zhengzhou 450008, China

**Abstract** Hitherto, a centralized management mode is prevalent in Automobile Supply Chain System (ASCS). The difficulty in data exchange and the information asymmetry between enterprises, nevertheless, are leading to a low efficiency in such system. Furthermore, critical question of trust is also raised due to the opacity of information. For these reasons, inspired by the emerging blockchain technology, a novel distributed blockchain-based ASCS with Hyperledger Fabric as a development framework is designed to provide secure and trusted transaction services for multi-party enterprises. The proposed system is provided with a series of advantages, such as access control, Data transparency, traceable and non-modifiable. In addition, a multichannel architecture is devised to realize the privacy isolation in enterprise collaboration. In this paper, firstly, the experimental environment is constructed by utilizing the docker technology, and then the functional interface is tested. Finally, the feasibility of the proposed system is verified by analyzing the throughput performance. This paper introduces the model of “blockchain+”, which provides a new idea for the upgrading and transformation of traditional automobile supply chain.

**Keywords** Blockchain application, Supply chain, Hyperledger fabric, Privacy isolation

### 1 引言

随着全球化竞争日趋激烈,不同产业的各类企业间存在着日益密切的买卖双方供需关系<sup>[1]</sup>。供应链正是基于这种关系,将产品供应商、制造商、分销商、零售商和终端用户连成整体的功能网链结构<sup>[2]</sup>。

汽车供应链指汽车生产过程中的原材料采购、零配件生产、整车装配、代理销售和报废品回收的一整套汽车的供应链体系,其中包含复杂的物流、信息流和资金流的运转<sup>[3-5]</sup>。由于汽车供应链中的企业群体过于庞杂,传统汽车供应链中多存在牛鞭效应<sup>[6]</sup>、双重边际效应<sup>[7]</sup>和品牌效应等问题,在保证

服务质量的前提下,最大限度地实现信息共享,加强上游供应商和下游零售商之间的联系,提高企业间的运作效率,构建高效安全的汽车供应链管理系统是国内外研究的重点。然而,当前的管理系统通常采用中心化的体系结构和传统的数据库来记录交易和组织间的数据共享,这就导致了在管理和交易等方面上的一些问题。首先,中心化的系统有一个单一的可信中心进行管理(比如汽车制造企业),缺乏高可靠性,并且面对庞杂的多层级供应商时,单一的中心机构会承担很大的存储和管理负担;其次,各企业使用并维护自己独立的数据库,供应链中存在严重的信息孤岛问题,信息在系统中流通困难,导致信息核对繁琐,数据交互存在障碍,进而影响整个供货计

基金项目:工信部智能制造标准化和新模式应用

This work was supported by the Standardization of Intelligent Manufacturing and Application of New Model in Ministry of Industry and Information Technology.

通信作者:林旭丹(linxd@bupt.edu.cn)

划;再次,各个企业之间信息不透明使得信息存在被篡改的风险,各企业间难以相互信任;最后,数据完全透明和隐私保护难以两全,一方面希望各方企业最大程度上传数据到系统中以便于核心企业的全局管控,另一方面各方企业不想将自己的商业机密信息暴露出去<sup>[8-10]</sup>。

针对上述问题,本文引入区块链技术,设计实现了基于区块链的汽车供应链原型系统<sup>[11]</sup>。该系统以 Hyperledger Fabric 作为区块链底层开发框架,采用自定义的、带有访问控制的智能契约,为链上的企业用户提供更可靠和安全的交易服务<sup>[12-14]</sup>。系统将所有的交易信息存储在区块链中,以不可变的、仅追加的方式持久化。在此基础上,本文还设计了一个多通道区块链网络结构,每个通道都拥有自己的账本,运行独立的智能合约,企业实体可以根据不同的业务需求订阅不同的通道服务,从而实现了对隐私数据的隔离和保护。此外,系统实现了基于 REST 的微服务体系结构,为企业用户访问提供了多种功能接口,用户可以通过 web 服务访问区块链网络进行查询、交易等操作。本文设计的原型系统是在 Hyperledger Fabric v1.1.0 基础上进行开发的,通过对不同操作和系统配置下的系统吞吐量性能进行测试分析,验证了该原型系统的可行性。

## 2 系统背景技术

### 2.1 区块链

2008 年中本聪在论文《比特币:一种点对点的电子现金系统》中首次提出了比特币的概念,而区块链正是比特币的底层技术<sup>[15]</sup>。区块链的本质是一种去中心化、去信任化、数据透明且不可篡改的分布式数据库,它按照时间顺序将数据打包成区块然后以顺序相连的方式组合成链,如图 1 所示。区块链的核心目标在于利用 P2P 网络协议、密码学技术和共识算法等关键技术的结合来解决数据传递和交换过程中的信任问题。采用区块链技术是解决传统供应链行业痛点最优雅的方式<sup>[16]</sup>。

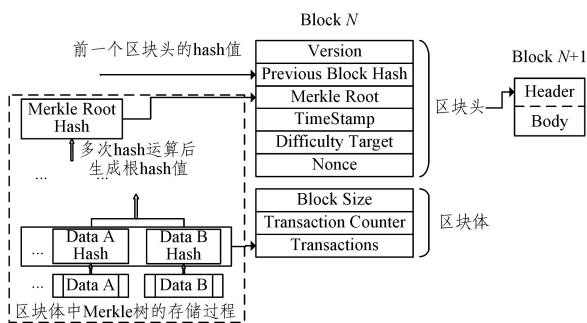


图 1 区块链结构图

Fig. 1 Structure of blockchain

### 2.2 智能合约

智能合约 (smart contract) 是一种旨在以信息化方式传播、验证或执行合同的计算机协议,它允许在没有第三方的情况下进行可信交易,这些交易可以追踪且不可逆转<sup>[17]</sup>。将智能合约部署在区块链节点上从而赋予区块链可编程的特性。业务开发者根据汽车供应链场景定义交易规则,并将该规则写入智能合约,外部应用通过调用智能合约并按照合同中规定的方式访问和操作区块链中的数据<sup>[18]</sup>。

### 2.3 Hyperledger Fabric

Hyperledger Fabric 是一个企业级的、开源的、许可型的

区块链平台,与比特币、以太坊不同,Hyperledger Fabric 网络中的节点必须经过授权认证后才能加入<sup>[19-21]</sup>。这样的设计避免了 POW 资源开销,大幅提高了交易处理效率,满足供应链场景下对处理性能的诉求,IBM 最新的试验结果表明,Hyperledger Fabric 的吞吐量最高可达 2000 TPS (transaction per second)<sup>[22]</sup>。同时,Hyperledger Fabric 拥有高度模块化和可配置的体系结构,支持可拔插的共识协议和 MSP 服务,并且它的运行不需要 token 支持。

考虑到汽车供应链场景下,参与企业多、业务逻辑复杂、货物价值大等特点,供应链网络更应该采用联盟链的组织形式(即只有获得授权的企业才能加入网络)进行搭建;同时供应链管理系统需要可靠的交易安全保障、高效的交易性能和严格的权限管理,所以采用 Hyperledger Fabric 进行开发是非常合适的方式之一。

## 3 系统设计

### 3.1 系统整体架构

汽车供应链原型系统从架构设计上可以分为 3 个层次:数据层、服务层和应用层,如图 2 所示。数据层是一个提供分布式数据存储的区块链系统,服务层负责应用层与区块链系统之间的交互,应用层则为多方企业用户提供操作页面。

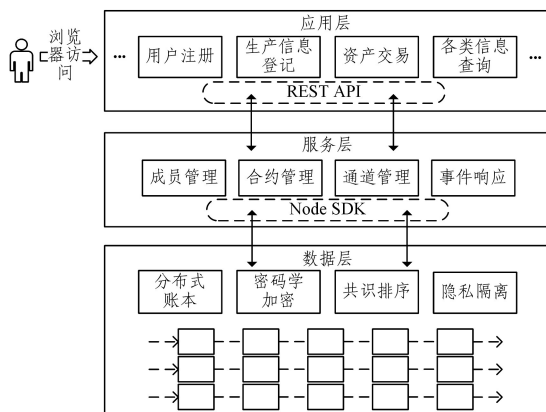


图 2 系统架构图

Fig. 2 System architecture

#### 3.1.1 数据层

数据层是基于 Hyperledger Fabric 开发的区块链系统,该区块链系统是由 1 个 Orderer 节点和 8 个 Peer 节点组成的 P2P 网络。Orderer 节点利用 SOLO 共识机制对网络中的交易进行全局排序,然后按一定数量规则将交易打包成区块发送给 peer 节点,由 peer 节点验证合法性后,将其不可变地写入区块链账本。所有的交易流程采用密码学算法进行加密以保证交易的安全可靠性。该网络中具备 3 个通道,每个通道都维护独立的账本,企业节点根据业务需求选择加入不同的通道中,以实现隐私数据的隔离。数据层的主要作用是存储汽车供应链中的交易数据,并确保数据的安全性和隐私性。

#### 3.1.2 服务层

服务层作为数据层和应用层之间的中间层,向上为应用层提供各种 REST API,向下利用 Node SDK 与区块链网络进行交互,将应用层的逻辑操作映射到区块链网络中。服务层还会对区块链网络进行管理,例如,成员管理功能对网络中的成员进行授权和验证;合约管理功能负责执行合约的安装、部署、更新和销毁;通道管理功能负责管理节点的加入和退出

以及执行通道的创建和关闭;事件响应功能会及时响应区块链网络中发生的事件然后反馈给应用层。

3.1.3 应用层

应用层为多方用户提供操作功能页面,用户通过浏览器或手机 APP 可以执行用户注册、生产信息登记、资产交易及各类信息查询等操作。系统的交互结构如图 3 所示,APP 通过 SDK 与区块链网络进行交互,进而访问安装在 Peer 节点上的智能合约(chaincode),并操作账本中的数据。链码在 Peer 节点上的隔离沙盒(docker 容器)中运行,通过调用 fabric 官方提供的 Shim API 实现各种操作逻辑,并使用 gRPC 协议与 Peer 节点交互。

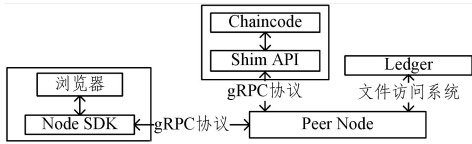


图 3 交互结构图

Fig. 3 Interaction structure

3.2 区块链多通道网络模型设计

汽车供应链一般包括 3—5 级各种类型的供应商企业,总数近千家。本文为测试系统模型,简化成 8 家企业组成的供应链,如图 4 所示。该模型包括 4 种类型的企业角色:汽车制造商、交易市场、一级供应商和二级供应商。本文在区块链网络中构建了 3 条通道(分别用绿色线、红色线和蓝色线表示),每个通道本质上都是由 orderer 节点划分和管理的私有原子广播通道,每个通道都可视为一条包含独立账本和智能合约的链。设计多个通道的目的是隔离通道信息,使通道外的企业实体无法访问通道内的数据,从而达到交易内容的私密性。

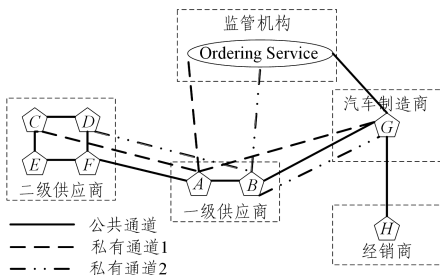


图 4 区块链网络模型

Fig. 4 Blockchain network model

汽车供应链一般包括 3—5 级各种类型的供应商企业,总数近千家。本文为测试系统模型,简化成 8 家企业组成的供应链,如图 4 所示,该模型包括 4 种类型的企业角色:汽车制造商、交易市场、一级供应商和二级供应商。

一般情况下,区块链网络内的交易信息全部存储在一条链上,链上参与节点共同维护这个唯一的账本,同时也将企业数据上传到这条链上,这意味着上传到区块链上的信息将会暴露给众参与方。在实际的汽车供应链场景中,存在竞争关系的企业并不想把全部信息暴露给竞争对手,比如生产同一零件的两个供应商,零件的销售策略信息对于他们来说十分珍贵,这涉及到能否获得零件订单。如果不能解决这些商业敏感信息的保密问题,将会影响企业上传数据的积极性。因此,本文设计了多通道的交易模型。

如图 4 所示,该模型中设计了 3 条通道:公共通道、私有通道 1、私有通道 2(图中分别用实线、虚线、点虚线表示)。所

有通道中的交易都由 orderer 节点进行统一处理。通道是 orderer 节点提供的基于发布订阅关系的一种通讯机制。将 orderer 节点和各个 peer 节点按照一定的业务需求连接在一起,形成一个具有保密性的虚拟通讯链路,利用通道对交易的隔离,保护企业的敏感信息。

在该模型中,8 个企业节点都加入到公共通道中,每个节点都要按照要求将指定的信息写入区块链账本中,并根据汽车供应链层级关系完成产品交易。此外,我们假设二级供应商 C 和 D 之间存在竞争关系,他们需要通过各自的营销策略为自己争取订单。假定营销策略属于商业机密,企业 C 和 D 都不想将自己的营销策略暴露出去。因此,C、D 分别订阅私有通道 1 和私有通道 2,且只在通道内上传营销策略信息。与 C、D 达成合作关系的一级供应商 A、B 也要分别订阅私有通道,对营销策略信息进行监督。企业在他们连接的私有通道里执行私有业务交易,不用担心通道外的企业会查看到隐私数据。

由 Orderer 节点提供的共识排序服务接受 3 条链(3 条通道)中的全部交易,因此数据的保密性仅与 peer 节点而不是 orderer 节点相关。当共识排序的功能角色由供应链体系中的可信方或核心企业担任时,这样是合理的,因为供应链中的交易作为业务需求仅对他们可见。如果 peer 节点不希望 orderer 节点知道交易内容,它可以通过哈希散列、零知识证明等密码学加密技术隐藏商业敏感数据,仅提供摘要信息给 orderer 节点。

3.3 数据存储设计

区块链系统中每个由 peer 节点维护的账本都由两部分组成:世界状态(world state)和区块链(blockchain),如图 5 所示。

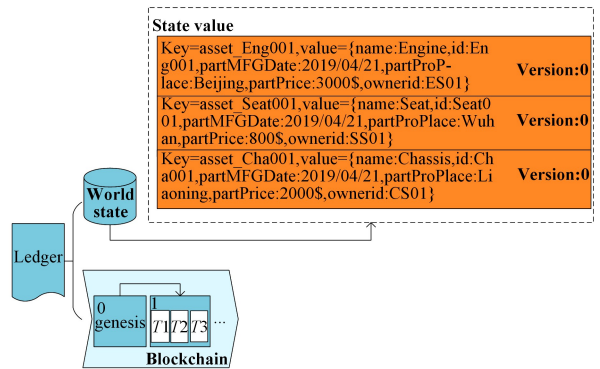


图 5 账本结构

Fig. 5 Ledger structure

世界状态(也称 stateDB)存储所有交易状态的最新值,图中世界状态包含 3 个状态值,version=0 表示该状态从被创建后还未修改过;区块链基于一个文件系统,以区块的形式存储数据,区块是一系列有序的、不可更改的状态传输日志,图中区块链中包含 0,1 两个区块,block0 代表创世区块,里面记录了区块链的配置信息;block1 包含了交易 T1—T3,对应着 asset\_Eng001 等 3 个初始化世界状态值。所有的状态数据都以键值对的形式存储,它们与 JavaScript 对象表示法(JSON)具有良好的交互性。

区块链系统中使用 4 张数据表存储状态数据,如表 1—表 4 所列,表 1—表 3 属于公共通道中的数据表,表 4 属于私有通道中的数据表。表 1 是用户信息表,用户注册时录入此表,用户销户和发生资产交易时更新此表。

表 1 用户信息表

Table 1 User information

序号	字段名	描述
1	Name	用户注册时使用的名称
2	Id	用户的唯一标识
3	Asset	用户拥有的资产目录

表 2 是资产信息表,有新资产(如零件,车辆等)产生时录入此表。

表 2 资产信息表

Table 2 Asset information

序号	字段名	描述
1	Name	资产录入时使用的名称
2	Id	资产的唯一标识
3	PartMFGDate	资产生产日期
4	PartProPlace	资产生产地址
5	PartPrice	资产初始定价

表 3 是资产变更记录表,有新资产产生时录入此表,有资产交易时更新此表。

表 3 资产变更记录表

Table 3 Asset change record

序号	字段名	描述
1	AssetId	资产生产时的 Id
2	OriginOwnerId	资产初始拥有者
3	CurrentOwnerId	交易后的资产拥有者
4	TradeDate	资产交易日期

表 4 是营销信息表,有新的营销策略产生时录入此表。

表 4 营销信息表

Table 4 Marketing information

序号	字段名	描述
1	UserId	提供营销信息的用户 Id
2	Name	营销策略名称
3	Id	营销策略唯一标识
4	StrategyInfo	营销信息内容

此外,世界状态的底层默认使用 levelDB 存储状态数据。LevelDB 的特点是主键查询速度快,而非主键数据的查询速度慢,并且不支持更复杂的查询。然而,在汽车供应链业务管理中,需要对各种类型的数据进行非主键查询和模糊查询等复杂查询。因此,本文为每个 peer 节点配备了 couchDB 作为状态数据库,以弥补 levelDB 查询功能的不足。

### 3.4 智能合约设计

链码(Chaincode)即是 Hyperledger Fabric 中提供的智能合约,是上层应用程序和底层区块链系统之间的媒介。本文将复杂的业务逻辑写入链码,然后在 peer 节点上安装及运行链码。本文使用 Go 编程语言编写了两个链码文件:CC\_public.go 和 CC\_private.go。每个链码包含的功能接口如表 5 所列。

表 5 智能合约接口

Table 5 Interface of smart contract

合约名称	接口名称	接口描述
CC_public	userRegister	用户注册
	userDestroy	用户销户
	assetEnroll	资产登记
	assetTrade	资产交易
	queryUser	用户信息查询
	queryAsset	资产信息查询
CC_private	queryAssetHistory	资产交易历史记录查询
	strategyEnroll	营销策略信息登记
	queryStrategy	营销策略信息查询

CC\_private.go 运行在私有通道中,加入私有通道的 peer 节点可以上传折扣信息、附加服务以及是否支持分期付款等营销策略数据到账本中。CC\_public.go 运行在公共通道中,所有企业通过 CC\_public 将其身份信息和生产信息写入区块链,并完成资产交易。数据写入区块链后不能被篡改,添加到公共通道的 peer 节点可以查询通道内的所有交易信息。算法 1 和算法 2 详细描述了 CC\_public.go 中的生产信息注册和资产交易的过程。其中的 constructUserKey 和 constructAssetKey 函数的作用是将 id 转换成 user\_id 或 asset\_id,以方便链码访问和操作状态数据。

#### 算法 1 资产信息登记

输入:assetname,assetid,ownerid,Asset informations

输出;if success,return transaction data else throw exception

Begin:

1. if assetname or assetid or ownerid is empty then
  2. return Error("invald args");
  3. else:
  4. get userbytes by GetState and constructUserKey function;
  5. if userbytes is empty then
  6. return Error("owner not found");
  7. else:
  8. Get assetbytes by GetState and constructAssetKey function;
  9. if assetbytes is not empty then
  10. Return Error("asset already exist");
  11. else:
  12. Add asset information to constructAssetKey(assetid);
  13. Update user information,add new assets to constructUserKey(ownerid);
  14. Update asset change record;
- End

在将生产信息写入账本之前,需要验证生产用户(owner)是否已注册。如果 owner 已经在账本中,我们将资产信息写入账本,然后更新生产者的资产列表,最后更新资产变更记录;若在账本记录中找不到 owner,则返回“owner not found”的错误信息。

#### 算法 2 资产交易

输入:ownerid,assetid,currentownerid

输出;if success,return transaction data else throw exception

Begin:

1. if ownerid or assetid or currentownerid is empty then
2. return Error("invald args");
3. Else:
4. get originownerbytes by GetState and constructUserKey function;
5. if originownerbytes is empty then
6. return Error("user not found");
7. Get currentownerbytes by Getstate and constructUserKey function;
8. Else if currentownerbytes is empty then
9. Return Error("user not found");
10. Get assetbytes by Getstate and constructAssetKey function;
11. Else if assetbytes is empty then
12. Return Error("asset not found");
13. Else:
14. Instantiate and serialize orginownerbytes object;
15. Aidexist←false;

```

16. For (i=0;i<len(originowner.assets);i++):
17.     If originowner.assets[i]==assetid then
18.         Aidexist←true;
19.         Break;
20. If aidexist is not true then
21.     Return Error("asset not match");
22. Else:
23.     constructUserKey(Originowner) delete assets of this trans-
        action;
24.     Add assets of this transaction to constructUserKey(cur-
        rentowner);
25.     Update asset change record.
End

```

在进行交易之前,应首先确认原始资产所有者是否真正拥有本次交易包含的资产。如果验证通过,我们将更改交易双方的资产列表,然后更新资产变更记录;若验证失败则返回“asset not match”的错误信息。

此外,每个链码中都具备查询功能,通过输入相关信息(如 userid, assetid, strategyid 等),用户可以获得对应的内容。

### 3.5 Web 服务设计

本文使用 Express 实现了基本的 HTTP 服务器功能,并提供了一组 RESTful API。HTTP 服务器通过 Node-SDK 与区块链网络进行交互。首先,服务器必须充当客户机访问 Fabric CA,以注册管理员 ID 并设置用户上下文。其次,通过使用 Node-SDK 提供的 API,服务器可以通过为客户端请求分配指定的通道 ID 和函数参数来调用目标通道上的特定链码,目的是查询和更新指定通道中的账本数据。由服务器设置的特定用户上下文可用于对 API 调用的所有请求进行签名。此外,从帐本中检索到的所有数据都可以转换成 JSON 格式。最后,本文利用 HTTP 调试工具 Postman 来调试 HTTP 服务器的各种接口。

## 4 性能评估与分析

本文在一台使用 Intel(R) Xeon(R) Platinum 8163 CPU @2.5 GHz 处理器、4 GB RAM、操作系统为 Ubuntu16.04(64 位)的计算机上进行了一系列的实验测试。由于实验条件的限制,本文将系统中涉及的节点(orderer, peer 等)和所使用的服务(couchDB, cli 等)放入 docker 中运行,并使用 docker 技术进行编排组织。

本文使用开源负载测试框架 Locust 通过分析 HTTP 请求的系统吞吐量来评估原型系统的性能。使用 Locust 模拟多个用户与区块链网络交互,不断地向系统添加记录。在每次测试开始之前,设置模拟用户的总数、每秒启动的虚拟用户数量以及模拟负载的任务执行之间的等待时间。在测试过程中,记录当前仿真所模拟的用户数、响应时间的中值以及不同负载下系统的 RPS 值。本文选择两个特定的 HTTP 接口在 peer 节点上执行 query 和 invoke 操作。

### 4.1 query 操作的系统性能

query 操作不需要与 orderer 节点建立连接,并且不会生成新的区块数据。通过 Locust 模拟 6000 个用户查询区块链中的数据,实验结果如图 6 所示。当请求到达率在 0~250 之间时,RPS 会稳步上升。当请求到达率在 250~350 之间时,RPS 缓慢增加并逐渐稳定。当请求到达率超过 350 时,RPS

有所下降并在 240 RPS 左右波动,说明此时服务器磁盘 I/O 达到峰值,成为系统瓶颈。通过对实验结果的分析,我们得到查询接口的系统吞吐量约为 245 RPS,基本可以满足企业数量较少的供应链系统。由于本次实验环境硬件限制较大,原型系统的性能仍有非常大的提升空间。

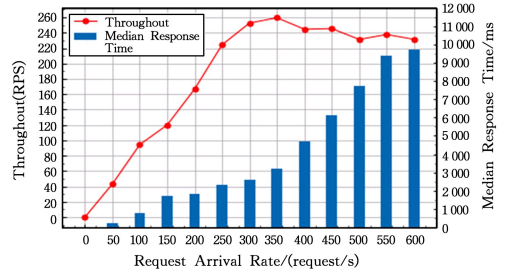


图 6 query 操作的吞吐量性能

Fig. 6 Throughput of query operation

### 4.2 invoke 操作的系统性能

与 query 操作相比,由于共识机制,invoke 操作更加复杂和耗时。当客户端发送 invoke 操作请求时,在向链中添加区块并将交易记录写入帐本之前,区块链系统需要经过 endorser 节点模拟运行和背书、orderer 节点排序和生成块、committer 节点验证等步骤。本文使用 solo 共识机制,通过改变 orderer 配置文件中的生成区块的策略和日志级别,进行了大量的实验测试,最后分析了不同参数对吞吐量的影响。通过控制变量,设置不同的参数如表 6 所列,实验结果如图 7 和图 8 所示。

表 6 实验参数设置

Table 6 Experimental parameters

number	Logging Level	Message Count	Block size/kB
1	DEBUG/INFO/WARN/ERROR	20	512
2	INFO	20	8/16/32/64/128
3	INFO	1/3/5/15	512

图 7 反映了日志级别对吞吐量的影响。可以看出,当日志级别为 DEBUG 时,吞吐量上限与其他 3 个级别有显著差异。这是因为在 Fabric 中,日志级别越低,系统日志的输出就越详细,在出现问题后调试就越容易。但输出太多日志会降低系统的吞吐量性能,在严重的情况下甚至会降低几个数量级。因此,在生产环境中需要仔细调整不同工作路径上的日志级别。

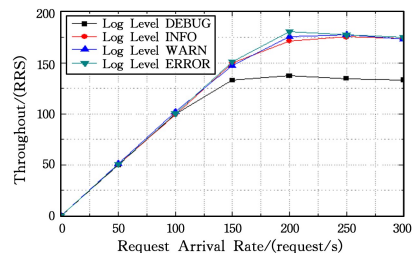


图 7 日志级别的影响

Fig. 7 Impact of Log Level

图 8(a)反映了区块大小对吞吐量的影响。可以看出,随着块大小的增加,吞吐量也会增加。当块大小超过 64 时,吞吐量不再上升并稳定在 175 RPS 左右。图 8(b)反映了交易数

量对吞吐量的影响。可以看出,消息数量的影响与块大小相似。当消息数量超过 10 时,继续增加不会改善系统性能。

从图 7 和图 8 可以看出,对 orderer 服务设置合适的参数对系统性能有很大的影响,参数具体的值取决于供应链场景下的实际生产环境。此外,还可以看出,本原型系统中 invoke 操作的系统吞吐量大约是 175 RPS。

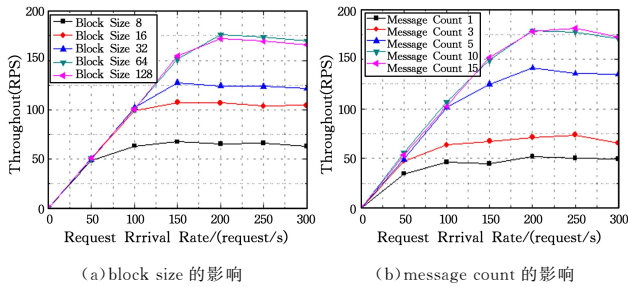


图 8 区块条件的影响

Fig. 8 Impact of blockchain conditions

**结束语** 本文结合现有汽车供应链管理中遇到的问题,设计并实现了基于区块链的汽车供应链原型系统,将供应链中的汽车制造商、零件供应商和交易市场等企业中的数据全部上链,利用智能合约确认不同企业间的每笔交易,最终形成透明有效的、不可篡改的区块数据记录在区块链中。在此基础上,本文设计了多通道结构,实现了对企业隐私数据的隔离。此外,该系统实现了基于 REST 的微服务结构,为多方用户访问提供丰富的功能接口。最后通过对系统吞吐量性能进行大量测试分析,验证了该方案系统的可行性。

目前,该原型系统只是在单机环境中进行实验验证,并未进行生产意义上的集群部署,还有很大的开发空间。在本文工作基础上,下一步工作将继续完善原型系统,例如开发一个功能完备的前端页面,对共识算法进行研究和改进等。

## 参考文献

- [1] SUNGBAE K, TAESOO M. [IEEE 2016 49th Hawaii International Conference on System Sciences (HICSS)-Koloa, HI, USA (2016. 1. 5-2016. 1. 8)] 2016 49th Hawaii International Conference on System Sciences (HICSS)-Supply Chain Integration and Collaboration for Improving Supply Chain Performance: A Dynamic Capability Theory Perspective[J]. 2016:307-316.
- [2] 郭珊珊. 供应链的可信溯源查询在区块链上的实现[D]. 大连:大连海事大学,2017.
- [3] MARODIN G A, FRANK, ALEJANDRO G, et al. Lean production and operational performance in the Brazilian automotive supply chain[J]. Total Quality Management & Business Excellence, 2017:1-16.
- [4] AZEVEDO S, CARVALHO H, VIRG'ILIO C M. Trade-offs among Lean, Agile, Resilient and Green Paradigms in Supply Chain Management: A Case Study Approach[M]// Proceedings of the Seventh International Conference on Management Science and Engineering Management Lecture Notes in Electrical Engineering. Berlin: Springer, 2014.
- [5] 李红峰. 供应链条件下的汽车零部件采购管理[J]. 现代经济信息, 2016(1):49-50.
- [6] 王敏. 华晨宝马供应链牛鞭效应研究[J]. 科技资讯, 2018, 509(8):251-253.
- [7] 马丽莎, 马燕. 分散型供应链中“双重边际效应”协调策略研究[J]. 现代商贸工业, 2016, 37(11):34-35.
- [8] 张瑾阳. LY 公司汽车零部件供应链管理研究[D]. 西安:西安科技大学, 2018.
- [9] 董晓磊. 汽车供销链体系结构研究与系统设计[D]. 重庆:重庆大学, 2011.
- [10] HARSASI M. The impact of supply chain management practices on competitive advantage[J]. International Journal of Economic Policy in Emerging Economies, 2017, 10(3):240.
- [11] KIM H M, LASKOWSKI M. Towards an Ontology-Driven Blockchain Design for Supply Chain Provenance[J]. Intelligent Systems in Accounting, Finance and Management, 2018, 25(1):18-27.
- [12] 杨慧琴, 孙磊, 赵西超. 基于区块链技术的互信共赢型供应链信息平台构建[J]. 科技进步与对策, 2018(5):21-31.
- [13] 史文斌. 基于 Hyperledger Fabric 的区块链应用系统云服务化[D]. 2018.
- [14] YANG Z, YANG K, LEI L, et al. Blockchain-based Decentralized Trust Management in Vehicular Networks[J]. IEEE Internet of Things Journal, 2018, 6(2):1495-1505.
- [15] NAKAMOTO S. Bitcoin: A peer-to-peer electronic cash system [OL]. <http://bitcoin.org/bitcoin.pdf>.
- [16] 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).
- [17] CHRISTIDIS K, DEVETSIKIOTIS M. Blockchains and Smart Contracts for the Internet of Things[J]. IEEE Access, 2016, 4:2292-2303.
- [18] WANG S, OUYANG L, YUAN Y, et al. Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019:1-12.
- [19] ANDROULAKI E, BARGER A, BORTNIKOV V, et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains[C]// Proceedings of the Thirteen EuroSys Conference, 2018:1-15.
- [20] TSAI W, BAI X, YU L. Design Issues in Permissioned Blockchains for Trusted Computing[C]// 2017 IEEE Symposium on Service-Oriented System Engineering (SOSE). San Francisco, CA, 2017:153-159.
- [21] SOUSA, JOÃO, BESSANI A, et al. A Byzantine Fault-Tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform[C]// 2018 48th Annual IEEE/IFIP International Conference on Dependable System and Networks(DSN). IEEE, 2018:51-58.
- [22] THAKKAR P, NATHAN S, VISHWANATHAN B. Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform[C]// 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems(MASCOTS). IEEE, 2018:264-276.



**LIN Xu-dan**, born in 1994, received the B. S. degree. His research interests include edge computing and blockchain.