

# 北京理工大学

## 本科生毕业设计（论文）

### 基于区块链的出租车调度系统的完善

The perfection of the taxi dispatching system based on  
blockchain

学    院：	计算机学院
专    业：	计算机科学与技术
学生姓名：	万琦玲
学    号：	1120180744
指导教师：	陆慧梅

2022 年 5 月 29 日

# 原创性声明

本人郑重声明：所呈交的毕业设计（论文），是本人在指导老师的指导下独立进行研究所取得的成果。除文中已经注明引用的内容外，本文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

特此申明。

本人签名: \_\_\_\_\_ 日期: \_\_\_\_\_ 年 \_\_\_\_\_ 月 \_\_\_\_\_ 日

## 关于使用授权的声明

本人完全了解北京理工大学有关保管、使用毕业设计(论文)的规定,其中包括:①学校有权保管、并向有关部门送交本毕业设计(论文)的原件与复印件;②学校可以采用影印、缩印或其它复制手段复制并保存本毕业设计(论文);③学校可允许本毕业设计(论文)被查阅或借阅;④学校可以学术交流为目的,复制赠送和交换本毕业设计(论文);⑤学校可以公布本毕业设计(论文)的全部或部分内容。

本人签名: \_\_\_\_\_ 日期: \_\_\_\_\_ 年 月 日

指导老师签名: \_\_\_\_\_ 日期: \_\_\_\_\_ 年 \_\_\_\_\_ 月 \_\_\_\_\_ 日

## 基于区块链的出租车调度系统的完善

### 摘 要

随着车联网的快速发展，信息交互更加频繁多样，为智能交通提供了强有力的数据支持。作为智能交通的重要一环，在线车辆调度相比于传统线下出租车大大提升了用户出行的便捷程度，减少乘客出行时间的同时也能够提升司机工作效率。区块链技术的发展为这一领域提供了新的改革机会，利用区块链去中心化的特点，解决传统中心调度存在的效率及安全问题。本文在实验室前期工作的基础上，设计完成了司机与乘客客户端界面，并将其与现有车辆调度和路径规划算法整合，完善了车辆调度系统的设计实现，使其能够尽量贴近真实场景下的系统使用情况；同时本文提出了一套基于车辆位置验证和交易评价的信誉评估算法，并通过实验验证了该算法的正确性。相比于实验室已有信誉值评估算法工作，本文丰富了信誉评估衡量因子，改进了算法，并将该评估模型整合至车辆调度系统，为乘客对于司机可信度的评估提供一定参考。

**关键词：区块链；出租车调度系统；信誉管理；车联网**

## **The perfection of the taxi dispatching system based on blockchain**

### **Abstract**

With the rapid development of the Internet of Vehicles, information exchanges are more frequent and diverse, providing strong data support for intelligent transportation. As an important part of intelligent transportation, online vehicle scheduling greatly improves the convenience of users' travel compared with traditional offline taxis, reduces passenger travel time and can also improve driver work efficiency. The development of blockchain technology provides new opportunities for reform in this field, taking advantage of the decentralized characteristics of blockchain to solve the efficiency and security problems of traditional central scheduling. Based on the preliminary work in the laboratory, this thesis designs and completes the driver and passenger client interface, integrates it with the existing vehicle scheduling and path planning algorithms, and improves the design and implementation of the vehicle scheduling system, so that it can be as close as possible to the system usage in the real scene; At the same time, a set of reputation evaluation algorithms based on vehicle position verification and transaction evaluation are proposed, and the correctness of the algorithm is verified by experiments. Compared with the existing reputation evaluation algorithm in the laboratory, this thesis enriches the reputation assessment measurement factor, improves the algorithm, and integrates the evaluation model into the vehicle scheduling system to provide a certain reference for passengers to assess the credibility of the driver.

**Key Words: Blockchain; Taxi dispatch system; Reputation management; Internet of Vehicles**

## 目 录

摘 要 .....	I
Abstract .....	II
第 1 章 绪论 .....	1
1.1 研究背景 .....	1
1.2 国内外研究现状及发展趋势 .....	2
1.2.1 车辆调度 .....	2
1.2.2 信誉值管理 .....	3
1.3 相关工作 .....	4
1.3.1 区块链和智能合约 .....	4
1.3.2 Geohash 编码 .....	5
1.3.3 基于 Geohash 的车辆调度系统 .....	6
1.3.4 车载网信誉值评估 .....	6
1.4 论文的研究内容及贡献 .....	7
1.5 论文组织架构 .....	7
1.6 本章小结 .....	8
第 2 章 区块链出租车调度系统架构 .....	9
2.1 系统完善目标 .....	9
2.2 可行性分析 .....	9
2.3 调度系统结构 .....	9
2.3.1 区块链 .....	11
2.3.2 客户端 .....	12
2.4 本章小结 .....	13
第 3 章 客户端系统设计与实现 .....	14
3.1 用户行为分析 .....	14
3.2 前端模块设计 .....	16
3.3 前端实现 .....	17
3.3.1 开发相关工具选用及介绍 .....	17
3.3.2 界面设计 .....	18
3.3.3 leaflet 地图渲染 .....	21
3.4 本章小结 .....	22
第 4 章 信誉值评估系统设计与实现 .....	23
4.1 信誉值影响因子 .....	23

4.2	算法设计 .....	24
4.3	科学性分析 .....	26
4.4	信誉值评估合约 .....	27
4.5	本章小结 .....	28
第5章	测试与实验 .....	29
5.1	区块链平台部署 .....	29
5.2	客户端测试工具 .....	30
5.2.1	脚本测试 .....	30
5.2.2	浏览器模拟测试 .....	31
5.3	前端使用时延测试 .....	32
5.4	信誉值模块测试 .....	33
5.4.1	位置验证模块测试 .....	33
5.4.2	主观评价模块测试 .....	33
5.5	信誉值综合计算参数选择 .....	35
5.6	性能测试 .....	37
5.6.1	测试服务器环境 .....	37
5.6.2	实验设计及结果 .....	37
5.7	综合实验 .....	38
5.7.1	模拟数据运行 .....	38
5.7.2	真实数据运行 .....	39
5.8	本章小结 .....	41
结 论	.....	41
参考文献	.....	42
致 谢	.....	44

## 第1章 绪论

### 1.1 研究背景

2021年10月12日公安部发布最新统计数据:截至2021年9月,全国汽车保有量有2.97亿辆;全国汽车驾驶人有4.39亿人。随着城市中流通车辆数量不断增加,道路承载能力有限,城市交通压力不断增加,“出行难”已经成为人们生活中的一大问题。在面对停车难、道路拥堵、限号出行等多种问题时,乘坐公共交通、打车等已经成为人们出行的重要选择,而相比于定时定点的公共交通,打车更为灵活方便。随着交通基础设施的不断完善,人们愈发依赖便捷的交通方式,但同时对于出行的需求也更多,传统的出租车<sup>[1]</sup>已经无法满足人们对于便捷度的更高要求,由此催生出了网约车这一事物。

随着车联网(Internet of Vehicle, IoV)的快速发展,一种新的商业模式随之出现——在线车辆调度<sup>[2]</sup>,更为通俗的名称是网约车。互联网平台充当交易中介,对于乘客提交的出行请求匹配合适的司机。相比于传统线下打车服务<sup>[3]</sup>来说,这毫无疑问大大节省了乘客等待时间成本,并在很大程度上实现了决策自主权,整体而言也能提升市场车辆运行效能,减少资源浪费。然而目前车辆调度系统等IoV应用中仍旧存在以下挑战:

1. 集中调度系统的效率问题。当前成熟网约车应用,毫无疑问均是通过互联网第三方平台控制。由于数据处理和传输依赖中心服务器,在网络规模较大时,该种调度方式对于时间和能源的消耗都比较大。
2. 统一数据存储方式的安全性问题。目前系统中用户相关数据均存储在第三方平台服务器上,一方面,若该平台遭受攻击,则其上存储数据的安全将受到很大威胁;另一方面,由于平台对数据的保护,用户无法完全控制自己的数据安全,这也导致了一定的信息安全隐患。
3. 统一调度的效益问题。在通过第三方平台完成交易的情况下,最终受益者包含司机、乘客、和平台三方,然而由于调度规则的不透明,对于司机和乘客而言无法保证利益的最大化,容易出现争议。

近年来,区块链技术<sup>[4]</sup>不断发展,由于其去中心化、开放性、独立性、安全性、

匿名性等特点保证了相关技术的“诚实”与“透明”，越来越受到研究者们的青睐，研究者在尝试将区块链与车联网工作相结合。由于区块链技术<sup>[5]</sup>数据存储与流动不依赖额外第三方，去除了中心管制，为打破数据垄断提供了必要的技术基础。利用区块链技术建立可信任、去中心化的数据库，创造真正的共享式经济<sup>[6]</sup>。

在打车中应用区块链技术能够彻底消除中介影响，允许乘客与司机的直接交流与交易。同时由于区块链平台数据的不可篡改性，能够为双方提供更为可信的验证，降低了信任成本。在网络出租车服务中使用区块链有助于参与的所有利益相关方关系更加紧密，这将为安全高效的打车服务铺平道路<sup>[7]</sup>。

考虑到网约车交易过程中信息不透明、交易对象陌生等特点，交易双方身份不一定完全真实可信、交易数据也并非完全透明公开，当存在交易过程中存在恶意参与者时，很可能会导致严重的安全问题。如何对交通网络的车辆可信度进行评估是一个重要问题。在车辆调度交易过程中，还存在司机与乘客的直接交互，所以司机行为对于车辆信誉值也有较大影响，通过合理评估该车辆信誉值，能够为乘客提供有效参考，同时也能对存在恶意行为的车辆做到有效标识，提升系统的实用性和安全性。

本文将在已有的出租车调度系统的基础上，完善用户接口设计，并结合区块链技术与车联网，完善实现了用户信誉值评估模块设计，并与现有调度系统结合，可视化展示车辆的行驶状态、信誉评估。

## 1.2 国内外研究现状及发展趋势

### 1.2.1 车辆调度

目前针对区块链的研究主要分为两个方向：一个方向是对于区块链底层共识机制的研究与探讨，另一个方向是针对区块链技术应用领域的尝试与研究。本文主要探讨应用方面。

杨柳青在 2018 年探讨了区块链技术在车联网中的应用研究前景<sup>[8]</sup>，区块链网络与车辆网络有着天然的相似性，都存在节点、节点间通过广播通信等特点。她提出主要应用方向有两个方向，一个方向是车辆与车辆间的直接通信，在车联网中，车辆可以模仿区块链节点组织方式，通过共识机制达到消息同步和更新的目的，同时通过智能合约约束通信车辆双方行为，达到分布式控制的目的；另一个是区块链数据记录的真实有效性，可以作为出现交通事故时责任认定的有效证据，通过智能合



约约束车辆行为，规避套牌等行为，同时管理者能够通过车辆唯一快速标识有效追踪联网的所有车辆，提供监管便利。

目前在车辆调度方面国内存在一些研究，但都处于设计层面，未出现能够完全应用于实际场景的方案。文献<sup>[9]</sup>中利用联盟区块链构建去中心化的交易系统<sup>[9]</sup>，重点研究了车辆与调度需求之间的数据交互，探讨最优调度策略；文献<sup>[10]</sup>提出了一种适用于车联云任务调度的轻量区块链架构<sup>[10]</sup>，提出利用路边单元构建区块链网络，由此构建了一种低延时、容错率高、不消耗额外算力的调度模型<sup>[10]</sup>。国内外相关研究均说明了区块链技术在车辆调度方面广阔的发展前景。同时也可以看到现有工作侧重于调度算法与模型的研究以及安全性、性能消耗等方面的研究与优化，未针对实际调度过程进行较为全面的分析描述。

### 1.2.2 信誉值管理

第五代通信技术为车联网奠定了通信基础，随着车载设备的不断发展，车与车、车与路边单元互联互通。在车联网应用中，为了保证服务质量，采用信誉评估对网络中<sup>[11]</sup>各节点行为进行约束管理。常见信誉值管理可分为集中式管理和分布式两类。

集中式的信誉值管理<sup>[4]</sup>通过中心化服务器评估车辆自组网（Vehicular ad hoc networks, VANETs）中车辆消息的可靠性。服务器负责收集消息反馈，汇总反馈并计算出信誉值，从而维护各车辆的信誉，由此对车辆行为进行约束管理，监督车辆的行为，通过信誉惩罚故意传播虚假消息等恶意攻击手段。

集中式的信誉管理依赖中心平台存储处理数据，这无疑增加了数据泄露的风险，同时该种方式所得结果的正确性强烈依赖于中心平台的权威性，一旦中心服务器的可信程度降低，整个系统的可信赖程度也将不复存在。

分布式的信誉管理<sup>[12]</sup>基于车辆自组网发展，后期随着区块链的发展，提出了更加去中心化、分布式的管理方案。文献<sup>[13]</sup>针对车辆信誉值，提出了一种基于车辆消息计算车辆信誉值的方法<sup>[13]</sup>，通过设计基于该信誉值的共识机制，有效解决车联网中车辆消息的可信程度问题，该方案通过路边单元维护分布式账本，记录车辆间的交易请求。多节点的设计能够为车辆提供冗余的响应节点，由此可有效减小单点错误对整个系统产生的影响，同时透明的记账流程以及可回溯不可追踪的特点使得结果的正确性权威性得到保证。所以更多的研究<sup>[6][14]</sup>集中于去中心化的管理方案。

在实际情况中，作为消息的制造者，司机行为应与车辆信誉值直接相关。同时，在车辆调度系统，即在打车服务中，司机与乘客间的交互行为也应该纳入信誉值评

估体系中，由此能够更加全面合理的评估信誉。

## 1.3 相关工作

### 1.3.1 区块链和智能合约

当前互联网上绝大多数交易都依赖于可信第三方处理支付，虽然目前该方式已经发展成熟，但是不可否认的是其仍然存在一定的安全威胁。交易过程中的信任问题为交易过程带来了额外开销，调解纠纷等环节的成本毫无疑问也会增加交易成本，因此需要一个基于密码证明而非信任的电子支付系统，允许交易双方直接与对方进行交易，无需经过可信第三方的转发。区块链技术由此而生，通过将加密数据以区块形式按照时间顺序叠加成链，形成不可修改的分布式记录，且永久保存。

区块链技术起源于比特币，在 2008 年由中本聪提出，其本质<sup>[5]</sup>是利用块链式数据结构验证和存储数据<sup>[15]</sup>，利用分布式节点共识算法维护和更新各节点数据<sup>[15]</sup>，利用密码学的方式保证数据传输和存储安全<sup>[15]</sup>，利用脚本组成的智能合约编程和完成复杂数据处理的分布式基础架构与计算范式<sup>[5]</sup>。区块链主要有以下特征：

1. 去中心化，区块链最显著的特征。区块链技术能够摆脱对第三方的依赖，其数据存储和处理均由分布式节点负责，不存在中心化的硬件和管理机构。其运转能够摆脱中心化控制，各个节点权利与义务对等，均实现了消息认证、传播和存储功能。链上数据由各节点共同维护。
2. 开放性，区块链技术基础是开源的，同样区块链上数据对于所有用户也是公开的，除用户隐私数据会经过加密处理存储外，合法用户均能通过链上接口查看存储的数据，不需要额外指定权限。区块链系统拥有高透明性。
3. 独立性，基于共识机制，各节点间的数据传递认证均不通过其他第三方就能自动完成，保证了该平台运行的独立性。不需要任何第三方平台参与或者监控，整个运行流程均按照既定算法运行，并且该规则也是公开透明的。
4. 安全性，区块链上数据具有不可篡改性，当一个节点增加一条数据，其余节点会同步这一改变，同时由于数据呈链式存储，下一块区块会包含上一块区块的 hash 值，所以单独改变某一区块数据而不影响其他区块是不可实现的。
5. 匿名性，从技术上而言，节点间数据传递可以匿名进行，无需公开身份或验证。

智能合约的本质是一种计算机协议，允许在没有第三方的情况下进行可信交易，这些交易可追踪且不可篡改。智能合约，使得区块链除了简单的交易转账功能外，还能实现复杂的业务逻辑，推动了基于区块链的分布式应用（DApp）的发展。

### 1.3.2 Geohash 编码

常见的地理信息编码方式是经纬度编码，通过经度和纬度可以确定唯一的地理位置点。Geohash 也是一种地址编码方法，能够将二维的空间经纬度数据编码为一个一维的字符串，该字符串能够确定一个唯一的地理区域，Geohash 字符串长度越长，所能确定的地理区域范围越小。通过该转换，二维空间检索转换为一维字符串匹配，能够在一定程度上提升检索效率。该转换在编码长度恰当的情况下，其误差在可接受的范围内。

Geohash length	Latitude error	Longitude error	Km error
1	$\pm 3$	$\pm 23$	$\pm 2500$
2	$\pm 2.8$	$\pm 5.6$	$\pm 630$
3	$\pm 0.7$	$\pm 0.70$	$\pm 78$
4	$\pm 0.087$	$\pm 0.18$	$\pm 20$
5	$\pm 0.022$	$\pm 0.022$	$\pm 2.4$
6	$\pm 0.0027$	$\pm 0.0055$	$\pm 0.61$
7	$\pm 0.00068$	$\pm 0.00068$	$\pm 0.076$
8	$\pm 0.000085$	$\pm 0.00017$	$\pm 0.019$

表 1-1 不同精度的 Geohash 编码的误差

根据 Geohash 编码的特性，其在车辆调度相关应用存在很大的发展空间，文献<sup>[16]</sup>提出了一种基于 Geohash 编码的空载出租车巡游路线推荐模型<sup>[16]</sup>。该模型将城市区域划分为合适大小，通过 Geohash 编码天然的定位区域特性，可以通过控制编码长度轻松完成该工作。

在其他方面也有应用，文献<sup>[17]</sup>提出了一种基于 Geohash 编码的 K 匿名位置隐私保护方案<sup>[17]</sup>，能够满足正确性、可验证性和匿名性，相比于传统的经纬度编码，Geohash 编码方式将用户具体位置泛化为一个区域，该方案通过构造匿名位置集，在用户提出位置查询请求时增加 k-1 个伪查询混淆用户真实的查询内容，实现了 k 匿名位置隐私保护<sup>[18]</sup>，能够抵抗伪装、重放等攻击。

### 1.3.3 基于 Geohash 的车辆调度系统

成家壮完成了基于地理位置区块链的车辆调度系统设计与实现，该系统使用基于 Geohash 编码的地理信息实现了路径规划算法以及基于地理位置实现区域车乘匹配算法。该系统相比于传统调度模型，解除了对于中心化服务器的依赖，保证了交易与地图信息的安全存储。

其中路径规划算法依赖于静态路网拓扑信息，结合常规静态路网的 A\* 算法，对于拓补网络中的节点进行路径规划，其正确性已验证。文献<sup>[19]</sup>中也论证了 Geohash 索引对于 A\* 算法的优化<sup>[19]</sup>。A\* 算法在进行最短路径计算时，常用曼哈顿距离作为评估代价，在运算过程中需要计算所有的邻接点距离，迭代执行，在网络较为复杂的情况下，计算量显著增长，直接以 Geohash 索引值作为判断标准能够大大减小计算量，减轻计算压力。

Geohash 编码在匹配方面也有优势，文献<sup>[7]</sup>中提出了一种基于 Geohash 的行程匹配方法<sup>[7]</sup>，该文献同时证明了该方法在行程匹配上效率的提高。成家壮完成的区域车乘匹配算法，依赖于地理位置区块链，该区块链将用户地理信息作为区块属性存储，根据地理信息的层次性结构，形成树状区块结构。在乘客提交打车请求后，可针对请求中地理信息所在区域进行检索，相比于传统区块链该设计能够有效区域内匹配检索的效率。

该系统依赖于以太坊实现。以太坊是一个开源的有智能合约功能的公共区块链平台，提供去中心化的以太虚拟机处理运行节点上的合约。这也是本文中车辆调度系统使用的区块链平台。

### 1.3.4 车载网信誉值评估

吴玥仪分析讨论了车载网中车辆间交互对于交互参与者信誉的影响情况，并实现了具体交互结果对参与者可信度的量化评估，她设计的算法主要考虑影响因子包括车辆间的位置验证、消息传播。她所提出的信誉评估方案主要针对车载网内车辆交互产生的信息，筛选主要影响因素。<sup>[20]</sup>

其中位置验证是指证明该客体在某个时间处于某个确定的地理位置的一个数字签名<sup>[21]</sup>。车载网中为了有效防治恶意参与者的恶意攻击行为，要求网络中车辆提供具有时效性、准确性的位置验证，目前常见的验证方式仍为集中式验证，文献<sup>[2]</sup>提出了一种基于集中服务器存储提交用户验证请求的验证方式<sup>[3]</sup>以及相关信息，该种方案有着集中式服务难以规避的缺陷。基于分布式节点的信誉管理方案<sup>[12]</sup>能够克服

该缺陷，通过路边单边提交车辆间位置验证请求，能够大大满足车载网中流动性、安全性等需求。

车辆间除了位置信息验证，还能传播包括路况、事故警告等消息，此类消息能够更好的描述道路信息，相邻车辆可以利用这些信息，这能够提高道路安全和交通效率。但是，车辆生成的信息可能不可靠，所以针对这个问题，需要评估该消息的可靠性。文献<sup>[4]</sup>提出了一个有效的方案评估消息的可靠性，结合消息传播者的信誉，可以对该消息的可信程度进行明确的评估和判断。

### 1.4 论文的研究内容及贡献

本文采用基于 Geohash 的地图数据作为基础，以太坊作为服务器平台，使用成佳壮提出的路径规划及区域调度算法，实现车辆调度系统整合。本文完善了该系统用户接口，针对乘客与司机设计了各自的客户端，能够完成符合实际应用场景的完整调度流程。

在实验室工作——车载网中基于车辆消息的信誉值评估模型的基础上，结合实际交易场景，本文提出了一种针对车辆调度交易过程中司机提供消息的真实性、活跃度以及舒适性综合衡量的评估模型，并且验证了该模型的科学性以及合理性。并将该方案以智能合约的方式整合至车辆调度系统中。

该系统使用基于 Geohash 编码的地图数据，并通过 Leaflet 框架对区块链上地图数据渲染展示。相比于传统经纬度编码方式，该编码方式能够模糊泛化具体地理位置，一定程度上保证了用户数据的安全性。

基于区块链平台的后台设计实现了去中心化的出租车调度系统，有效解决了传统中心化服务器存在的数据安全隐患等问题。其中针对信誉值的设计完善了该系统在实际应用场景下的使用，基于车辆验证和交易评价综合衡量的设计更加贴合实际应用，能够更加全面的评估司机行为。

### 1.5 论文组织架构

本文首先对于该出租车调度系统进行了整体分析，从应用层角度以及数据处理角度分别分析了各模块的结构与功能。然后对于浏览器端完善的设计方案以及最终实现效果进行了展示，从用户行为入手，设计完成前端各模块职能划分，最后展示了各界面的样式，介绍了相应功能接口。在第四章，提出了一套基于车辆间位置交互以及交易过程中用户评价的信誉评估系统，具体阐述了算法相关设计以及科学性

验证。在第五章完成了该系统的模块化测试以及整体运行展示，实验证明了本文工作的合理性以及适用性。

## 1.6 本章小结

在这一章中对本文所选课题背景和发展趋势进行了介绍，目前针对区块链技术  
与车辆调度系统结合的研究仍旧是研究者们热衷的方向，这对于未来市场发展，技术改革都有着重大意义。介绍了本文所涉及到的相关工作，其中包括区块链这一主要技术，其次介绍了本文所用的地理编码方式——Geohash，然后对实验室前期工作进行了简单介绍，包括本文所用到的车乘匹配算法、路径规划算法以及基于车辆间交互的信誉值评估算法。最后介绍了本文的主要工作及贡献，本文主要完成了针对乘客和司机的前端界面设计与实现，并将其与车乘调度、路径规划算法相结合完成了出租车调度系统的整合，另外，本文还提出了一种基于位置验证和交易评价的信誉值评估模型，能够更加贴近实际应用场景，对车辆信誉值有更合理更全面的评估。最后通过相关实验论证了本文工作的合理性。

## 第 2 章 区块链出租车调度系统架构

本章首先介绍在实验室已有工作的基础上本文所做的完善工作目标，主要包括信誉评估模块的设计与实现以及对于前端开发需求的探讨；然后对系统的架构进行介绍，主要包括区块链后台的数据存储与处理、用户端的功能接口与交互。

### 2.1 系统完善目标

为了更加贴合实际应用场景，在实验室中已有调度算法和路径规划算法的基础上，针对用户使用前端接口的完善，能够加快该调度系统的实际应用；同时针对完整打车服务流程的研究与探讨，也能完善现有调度逻辑接口。通过搭建 Web 前端界面，实现司机与乘客的客户端应用，能够更加清楚的展示调度过程中的信息交互与事件响应。

此外，本文提出了一种对司机行为进行评估的信誉值管理方案，该方案基于车辆交互验证结果及交易评价，结合车辆间交互行为与乘客与司机的交互行为，综合衡量车辆信誉值，约束司机行为，为乘客对司机的可信程度评估提供一定参考。同时，调度系统信誉值管理模块的加入，能够一定程度上防范恶意参与者的恶意参与行为，该管理方案通过信誉值惩罚不规范行为，鼓励文明规范行为，营造良好的交易生态环境，提升车辆调度系统实际应用体验。

### 2.2 可行性分析

完善客户端设计，划分功能模块，界面更加友好，操作简单易懂，对用户来说操作流程清楚，结果直观，使用难度降低。能够有效降低提交打车订单、查看交易进度等各项功能的使用难度。完善相关功能的工作，提高了该系统的实用性，证明了该调度系统可用于真实使用场景中。针对信誉值的设计亦是直观评估了司机的驾驶行为，通过评估司机的行为对其进行有效监督，一方面能够监督司机的异常驾驶行为，另一方面也能鼓励司机提高自己的服务质量，为整个交易市场起正向激励作用。

### 2.3 调度系统结构

从应用层面看，如图 2-1 所示，该调度系统参与对象分为三部分：路边单元、车辆节点、乘客节点。各部分具体功能如下：

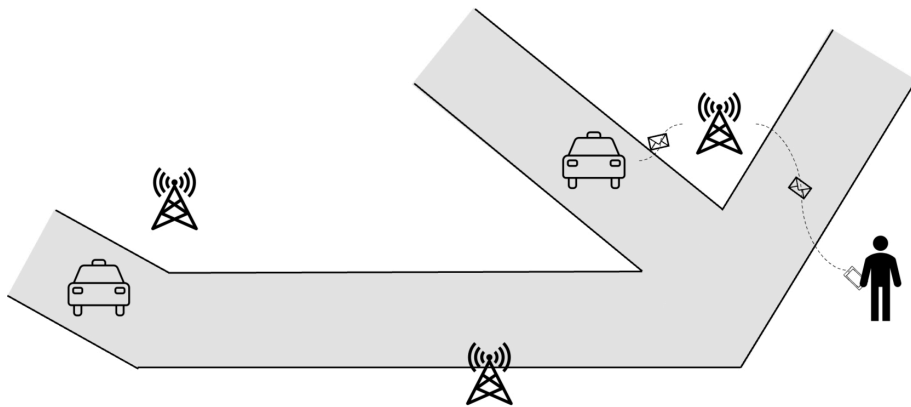


图 2-1 系统应用架构

- \* 路边单元：运行区块链节点，维护一个分布式账本，记录一定区域内车辆信息和交易记录。作为打车服务过程以及信誉值交互事件中的响应节点，记录处理相关数据。出于安全和可靠性的考虑，在一定区域内设置超出正常运行所需数量的节点，同时一辆车辆可在多个节点注册信息，在牺牲一定资源的情况下保证该系统的稳定运行，有效避免单节点故障对于整个系统运行的影响。
- \* 车辆节点：作为交易中信息产生和接收对象，其账户必须经过认证后统一指定。在该系统中，车辆节点主要作为车辆调度过程和信誉值评估过程中的被动节点，重在维护车辆节点信息的时效性。
- \* 乘客节点：乘客作为交易过程中提交交易的主要对象，其安全性同车辆节点一样，需要经过身份认证后由系统指定其账户。乘客节点发布交易请求后，该请求会提交至路边单元，路边单元处理后将结果返回至乘客节点，在必要时也通知负责区域内对应车辆节点。

分布响应节点的设计能够有效满足车载网中调度事件分散、区域性的特点，将全局事件转化为区域内事件响应能够有效提升事件响应速度，减小事件处理工作量，该改善效果是显然的，从响应速度、数据处理量、数据安全性等方面看，分布式设计能够更好的满足车辆调度系统的实际运行情况。

从数据处理层面看，该系统整体由两部分构成，区块链部分存储地图信息及其他交易相关数据，主要运行于城市交通系统的路边单元，且一辆车会在多个节点注册，保证在个别节点出现问题时整个系统也能不间断运行，节点间能够同步记录，保



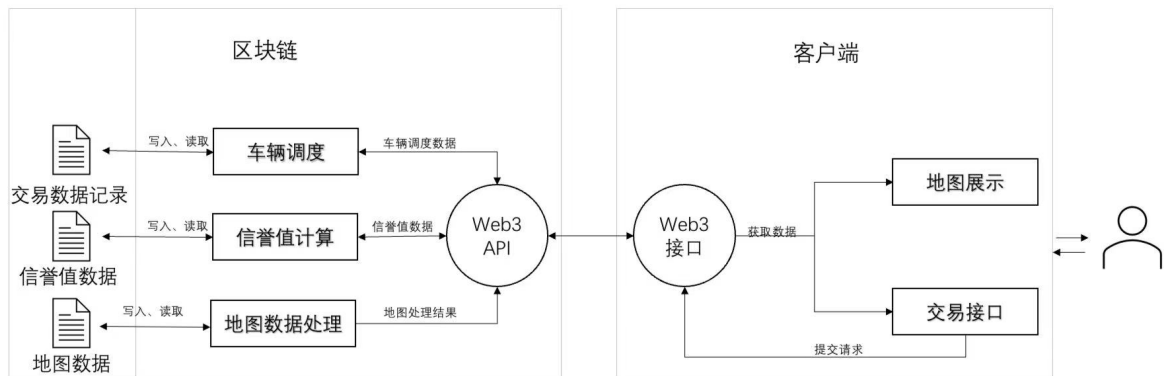


图 2-2 系统逻辑架构

证数据的安全性；智能合约部分负责相关数据的处理，该部分事实上也运行于区块链上，使区块链除简单的记账功能外还能完成复杂的数据处理和事件响应，在该系统中是后台数据的主要处理模块；浏览器端直接与用户交互，主要包含地图展示与交易接口两部分，用户通过操作交易接口触发调度事件，并通过地图与其他可视化信息得到事件反馈，完成整个打车过程，整体架构如图 2-2 所示。该调度系统的运行流程如下：

1. 系统初始化。浏览器端初始化三个合约，并与区块链建立通信
2. 浏览器端通过相关接口进行交易，并将交易数据发送至合约，等待响应，具体交易过程如 3.1 节所示
3. 上传交易结果。针对交易过程中的各步骤结果，上传至区块链，在区块链上负责重要数据的存储，利用区块链数据的不可篡改性，保证该次交易数据的安全有效，可在需要的时候有效溯源。

### 2.3.1 区块链

该系统使用地理区块链作为服务器。相比于传统区块链，地理位置区块链将区块与地理位置绑定，更加满足车载网中的交易环境。显然车载自组网与地理信息密不可分，传统区块链的交易记录并不重点关注地理位置这一信息，不支持通过地理位置进行位置定位或查询等操作，而地理位置区块链则对车载网有着良好的匹配度。根据 Geohash 地理位置编码与地理区域层次关系建立树状区块链，能够更好的支持区域内事件响应，提升区块链对于车载网中事件的响应性能。

智能合约作为运行于节点上的计算机协议，控制区块链与用户端的交互。根据该系统的用户角色，可分为区块链与乘客端的交互和区块链与司机端的交互。乘客端通过合约接口提出并生成打车请求，并保持联系。合约处理该请求，得到结果后通知乘客端，乘客端接收到车辆匹配结果后响应后续打车事件包括路径规划等，直至乘客支付完成，这个过程中将对关键交易信息进行记录；司机与区块链的交互相比乘客较为被动，在司机登录该系统后，会等待系统的匹配通知，若有乘客的打车请求匹配到该车辆，则司机端能够接收该消息，自主选择接受该订单或拒绝，若接受则继续事件响应，直至交易完成，在乘客支付完成后可重置为空车状态，具体过程见第三章。在交易过程中除了基本的交易事件外，还有信誉值相关事件，如司机端会周期性与邻近车辆产生位置验证，并将位置验证结果发送至区块链，同时司机接受或拒绝订单时，也会将司机选择结果上传，在司机接受订单并且交易结束时，乘客会为司机服务质量评分，该结果也会同步至区块链，在这个过程中，司机的信誉值不断更新，从而保证该系统中各用户的可信程度始终维持在较新的状态，更加有参考价值。

本系统中主要存储信息包括地图数据、交易记录、以及车辆信誉值信息，分别对应地图数据管理、打车交易、信誉值管理三个功能。三类数据存储于区块中，并通过区块链的共识机制同步至网络，保证了数据的不可篡改性和安全性，在必要时也能进行事件回溯，这是该系统区别于传统中心化调度服务器的显著特点。

### 2.3.2 客户端

根据该系统使用用户角色不同，分为乘客与司机。针对乘客而言，其主要功能包括查看地图、登录、选择上车点与目的地并提交请求、查看匹配司机信息，确认上车、查看规划路线、确认到达目的地并支付等；对于司机而言，主要功能包括查看地图、登录、查看匹配乘客信息，确认接受/拒绝订单，查看规划路线确认乘客上车等功能。

该模块是与用户直接交互的模块，需要能够提供足够清晰明白的操作界面，以及必要的信息提醒。该模块主要由两部分构成，主要是地图渲染模块以及事件响应模块。

1. 地图模块，针对从合约中获取的地图数据信息，需要解析渲染至地图中，并对不同信息分类展示，如静态地图数据绘画、司机乘客位置、规划路线等用户易于理解的展现方式，主要渲染过程见第三章。

2. 交易模块，该模块主要通过 Web3.js 的合约接口绑定，为用户提供多个的事件操作接口，决定交易执行状态。

## 2.4 本章小结

本章首先讨论了出租车调度系统的完善方向：一是整合该调度系统，实现针对乘客与司机的客户端设计，能够完成基本的打车流程；二是提出了基于车辆验证以及交易评价的信誉值管理方案，并将该方案整合至调度系统，为乘客评估司机信誉提供一定的参考。然后从两个层面描述了该系统架构，分别是系统应用层面和数据处理层面。从应用层面上分析，该系统中有三类参与节点：乘客节点提出交易请求；车辆节点负责接收请求并做出动作响应；路边单元则替代了传统调度系统的中心服务器，负责信息存储及处理。从数据处理层上看，该系统分为两个部分，区块链平台，区块上存储了地图信息及其他交易相关数据，智能合约接收用户请求，结合已有静态数据进行处理，最后将结果存储到区块上；客户端平台直接与用户交互，实现更好的信息展示效果，在一定程度上模拟该系统在真实应用场景下的交互情况。

## 第3章 客户端系统设计与实现

本章主要介绍了对于出租车调度系统中客户端部分的设计与实现。该部分与用户行为紧密相关，所以首先介绍了区块链车辆调度系统一次完整交易过程中，对用户行为的分析；然后针对用户行为对于前端模块进行分析设计，确定主要功能；最后介绍了实现该系统的具体过程以及完成效果。

### 3.1 用户行为分析

对于乘客而言，一次打车事件涉及主要功能包括登录、提交打车请求、确认上车、支付并评分、以及查看当前行驶状态五部分；对于司机端涉及主要功能包括登录、接单、查看规划路线三部分。针对完整的打车流程，抽象简化如图 3-1 所示。具体算法描述如下：

1. 乘客登录该系统；
2. 乘客提交打车请求，包含上车点以及目的地
3. 区块链平台接收到请求后，查找在线满足条件的车辆，若有结果则通知乘客以及司机，转 4。否则，转 5；
4. 司机可决定是否接受订单，若接受转 6，否则转 5；
5. 订单取消；
6. 区块链平台计算路线并通知乘客与用户，司机根据该路线行驶至乘客上车点；
7. 乘客确认上车；
8. 区块链平台计算路线并通知乘客与用户，司机根据该路线行驶至乘客目的地；
9. 乘客支付并对本次交易进行评价；
10. 订单完成，司机与乘客均重置状态；

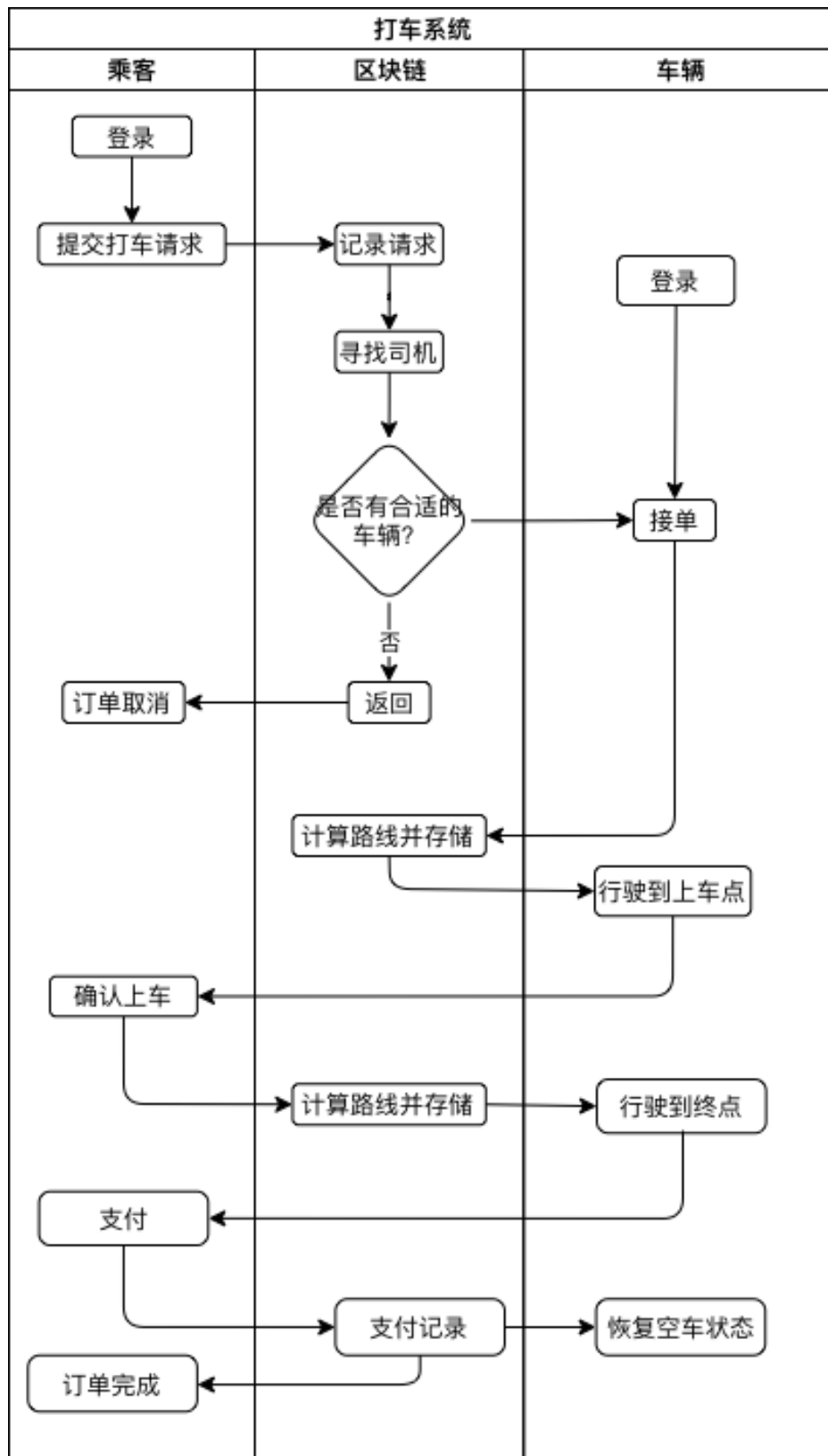


图 3-1 打车流程示意

### 3.2 前端模块设计

该系统存在两类用户，司机和乘客。两类对象的结构有一定差异：对于乘客而言，更为重要的是交易订单所需的上车点和目的地，内部均存储为 GeoHash 编码，便于与区块链上地图数据交互，实现更快的查询与处理速度；对于司机而言，还有状态和信誉值两类特殊信息，状态用于区分当前车辆是否空闲，信誉值则是用于评估司机行为，在交易过程的司机一系列行为都会对该数据产生一定的影响，也是作为乘客判断该司机可信度的重要参考。同时对于司机和乘客而言，都包含了 ID（用户与区块链交互的唯一账户），和定位（由各自客户端运行设备提供）。

乘客	司机
ID	ID
定位	定位
上车点	信誉值
目的地	状态（是否空车）

表 3-1 用户结构对比

从图 3-1 中流程分析可以看到，本系统针对乘客端可划分为四个模块，具体描述如下：

1. 登录模块：检查用户输入信息正误，完成用户登录。主要检查用户数据是否为区块链上可用账户。若乘客未拥有自己的唯一 ID，则需要区块链平台额外生成、指定，并通知该乘客。
2. 地图展示模块：静态地图数据展示，以及规划路线等动态数据展示。另外为了方便用户使用，在地图上展示附近预设上车点，用户可通过直接点击该点设置为上车点或目的地。为了配合区块链上地图数据存储格式，在此使用 GeoHash 作为地理位置的编码方式。
3. 事件响应模块：监听交易事件进行，为乘客提供事件处理接口，主要包括查看车辆调度结果、确认上车、确认支付、评价等响应功能。这些功能有严格的执行顺序，故设计为在功能按钮不可用时隐藏该按钮。该设计能够更好的避免用户操作失误，同时使用户专注于当前事件，符合简洁、清楚等设计要求。
4. 打车请求管理：添加、修改、删除上车点以及目的地，为了方便用户使用，在

用户输入地点时，通过模糊匹配展示相关地点信息，在用户输入部分关键字时，通过正则匹配到包含该关键字的匹配地点，返回给用户。该设计简单而又实用，使用户在花费较少操作的情况下得到理想结果，优化用户使用体验。

对于司机端，较为简单，可划分为三个模块，具体描述如下：

1. 登录模块：实现用户登录与注销，并在登录状态展示司机 ID、评分、位置信息。这些信息均为实时更新，随着车辆信誉值的变化动态改变，保证数据处于最新状态，借此让司机重新审视自己的行为规范程度。
2. 地图展示模块：地图信息展示，以及调度路线展示。在获取到订单时，将在地图上标注，便于司机直观查看乘客位置信息。对于司机而言，乘客的位置信息对于自己是否接受该订单有较大的影响力，由此，在地图上可视化展示乘客位置信息，能够让司机更为直观地得到自己想要的信息，同时通过定位附近建筑信息，能够为司机提供更为准确的乘客定位，减小误差。
3. 事件响应模块：监听交易事件进行，为乘客提供事件处理接口，主要包括确认接单、拒绝接单、确认乘客上车等响应功能。与乘客端一样，此类事件处理接口有着严格的执行顺序，同样采用不可用即隐藏的方式提升操作稳定性。

针对客户端的主要功能进行了模块划分，但是模块间也存在着数据通信，由此设计全局数据管理模块，作为模块间通信的桥梁。模块化的设计，能够提升开发效率，方便模块间组合分解以及代码复用，同时维护成本较低，也便于单个模块测试修改。

### 3.3 前端实现

#### 3.3.1 开发相关工具选用及介绍

Vue

Vue 是一套用于构建用户界面的渐进式框架<sup>[22]</sup>。与其他框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 只关心图层，易于上手，且便于与第三方库整合。Vue 支持组件化应用构建，允许开发者使用小型、独立和通常可复用的组件构建大型应用<sup>[22]</sup>；同时 Vue 最独特的特性之一，是其非侵入性的响应式系统<sup>[22]</sup>，数据模型的更改能够直接触发视图响应更新，这使得状态管理十分简单直接。

Web3.js API:

Web3.js 是与以太坊交互的一系列 API 的 javascript 实现版本。在本文中主要用于调用服务器上已经部署的合约函数，监听合约事件。

Leaflet:

Leaflet 是一个为构建移动端互动地图而开发的 Javascript 库，设计简单，易于上手。遵循简便、高性能和可用性好的设计原则，在桌面和移动端均能高效运行，并且由于其支持插件扩展，足以满足多种多样的功能需求，节约开发成本。

### 3.3.2 界面设计

主界面（图 3-2）

该界面是用户进入网站后的默认界面，其中司机端（左）只显示登录按钮，其余操作均在登录后可见；乘客端（右）包含登录按钮，以及上车点和目的地输入框，地图中展示了备选上车点。在打车服务中，显然对于用户来说重点应在地图上，所以界面主体为地图展示，登录模块固定在页面左上方，其余模块固定在界面下方，并且可以随时隐藏，达到充分查看地图的目的。

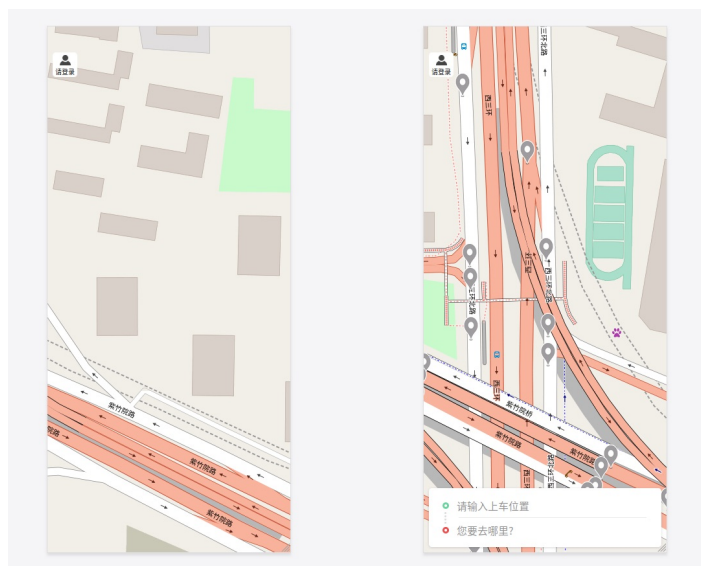


图 3-2 主界面

登录界面（图 3-3）

点击主界面的登录按钮后弹出登录界面，两类用户的登录界面均为图左所示。对于司机端，登录后可在该界面查看车辆信息，如图右所示，包括 ID，车辆位置，车辆评分。





图 3-3 登录界面

#### 地点管理界面（图 3-4）

通过主界面下方管理上车点和目的地，如图 1-1，点击输入框可进入地点选择窗口，如图 1-2，为了便捷查找，在输入时通过模糊匹配，筛选出待选的地点，如图 1-3 所示。

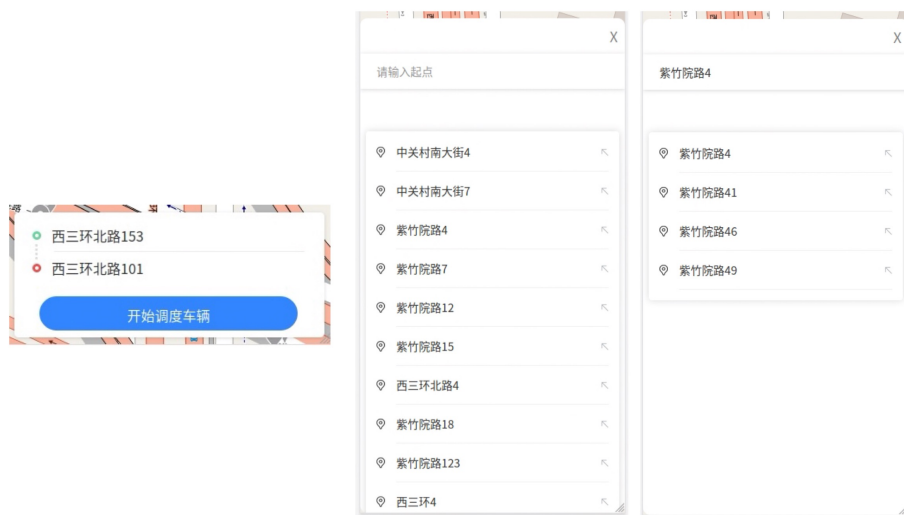


图 3-4 地点管理界面

#### 事件响应界面（图 3-5）

在该界面，监听交易中的进程，对乘客来说包括调度到车辆、等待车辆到达上车点、确认上车、等待车辆到达目的地、支付、评分等事件；对于司机来说包括接到

订单、前往乘客上车点、前往乘客目的地、等待乘客付款等事件。部分事件响应如图所示。



图 3-5 事件响应界面

地图展示界面（图 3-6）

该界面使用 OSM 提供的开源地图获取北京市的道路信息，并通过 leaflet 框架加载 OSM 数据。除了基本的地图信息外，乘客客户端中，额外显示内容主要包括以下三项：

1. 乘客周边预设上车点（图中灰色图标），可直接点击设置为上车点或者目的地。
2. 针对乘客输入的乘车点和目的地均在地图中显示（图中绿色和橙色图标）。
3. 对于交易过程中通过区块链计算得到的规划路线，也在地图中可视化展示（图中线段），其颜色则反映了调度的车辆实时信誉值变化，信誉值从 0 至 100 对应颜色红色至绿色。



图 3-6 地图展示界面

### 3.3.3 leaflet 地图渲染

针对区块链上存储的 GeoHash 编码格式的地图数据，周畅设计了一种矢量地理数据结构，GeohashTile<sup>[23]</sup>，实现了 GeoHash 坐标和屏幕像素点的转换，转换并显示的过程主要有以下两步：

1. 根据当前放缩程度瓦片的大小，计算 GeohashTile 的数量，再通过中心瓦片的 GeoHash 编码逐步确定周围所有邻近瓦片的编码，从而得到所有 GeohashTile 的编码。
2. 从服务器端得到所需地图的 GeoJson 数据后，通过相对位置投影转换为屏幕坐标。第一步是计算对应 GeohashTile 中心点到屏幕中心点的像素距离，后计算目标点相对于对应 GeohashTile 中心点的像素距离，二者叠加即为目标点到屏幕中心的像素距离，即可得到目标点的像素位置，后通过 leaflet 框架中绘制函数在图中标注。

上述情况均是在地图静止不动的情况下的渲染过程，在地图发生动态切换，如放缩、投影后，需要重新绘制画面，成佳壮对于该部分功能进行了设计与实现，在地图放缩或拖动后，重新渲染过程如下：

1. 放缩或者拖动后，需要重新确定中心像素点的 GeoHash 编码。首先确定当前操作点的瓦片中心点，计算操作点到该点的相对像素距离。
2. 根据当前缩放程度等信息计算出操作点的 GeoHash 编码，以该 GeoHash 作为地图展示的中心点，重新渲染。

通过对于渲染流程的修改，使得 leaflet 框架能够支持 GeoHash 编码数据的渲染与绘制，前后端编码一致对于数据存储以及数据传输带来了极大的简便，同时由于该系统中整体运用 GeoHash 编码格式，能够在不影响系统使用的情况下有效模糊用户定位，相比于经纬度编码直接确定唯一地理点的特性相比，GeoHash 所能确定位置为区域的特点能够更好的支持用户隐私保护。

### 3.4 本章小结

本章从用户行为着手，分析一次交易过程中司机和乘客的各个行为，并以此为基础划分出前端各个模块，明确模块功能与职责。该系统使用 Vue 框架进行前端开发，该框架支持组件化开发，符合该系统模块化结构。通过 Web3.js API 库与智能合约交互，其中路径规划与车乘匹配合约在实验室前期工作中实现，信誉值相关合约在本文第 4 章有具体介绍。然后基于一致性、可控、效率的前端设计原则，完成了各个界面及功能模块。

## 第4章 信誉值评估系统设计与实现

针对当前信誉值管理方案仅限于车辆间交互，消息验证内容较为单一的问题，本文提出了一种将车辆位置验证与交易评价两类因素相结合的评估方案，在车辆调度交易过程中，车辆信誉与司机行为紧密相关，该方案综合车辆间交互信息与司机行为评价，能够更加全面合理的评估车辆信誉。

### 4.1 信誉值影响因子

交易过程中无法避免恶意参与者的出现，为了尽可能减少恶意参与者对于正常用户造成的危害，对于已经认证的车辆发布虚假消息或篡改消息等行为可以使用基于信任的方法来解决<sup>[24]</sup>，因此采取一定的措施评估用户信誉值是有必要的。在整个交易环节中，产生了大量的信息，针对这些信息，本文从真实性、活跃度、舒适性三个角度进行筛选，各自选取显著的评价指标作为计算信誉值的参考因素。

本文提出的信誉值计算影响因子包括：

- 用户活跃度。主要通过司机上传有效验证信息的频次决定，司机上传、更新的频率越高，其对应的信誉值增益越明显。旨在鼓励司机积极参与验证行为，保证车辆在一定时间内的数据新鲜度。如果车辆长时间不参与交互，那么无法保证该车辆信誉值的时效性，所以应该对其信誉值有一定的抑制。
- 位置验证结果。这是针对车辆位置信息真实性评估的重要计算依据。车辆通过邻近车辆对其地理位置的重复确认，验证自己的位置信息的真实性。一次成功的位置验证能够提升其信誉值，相反，位置验证失败时可能意味着车辆定位不够准确或者故意捏造虚假位置信息等情况，信誉值都会有一定的减少，鼓励司机上传真实有效的车辆位置信息。
- 接单率。对于整个车辆调度系统而言，车辆的接单率低会直接影响乘客的使用体验，所以鼓励司机积极接单是有必要的。对于接单率较低的司机，其他指标所带来的增益结果将随之降低，为了排除这样的影响，司机需要积极接单提升自己的接单率。
- 主观评分。这主要来源于乘客对于本次交易的主观评分，评价因素更为综合，有利于司机从乘客体验角度改善提升自己的服务。同时也能一定程度上遏制部

分司机的不文明驾驶行为，因为来自乘客的频繁差评会降低他的信誉值评分。在一次打车服务中，乘客与司机存在较多的交互，在各方面均会产生一定的评价，例如是否准时、态度是否文明、车上环境是否干净等，但是为了算法的简洁性，在此抽象为一个统一的参数，即主观评分。

总体而言，信誉值评估系统设计如图 4-1 所示。

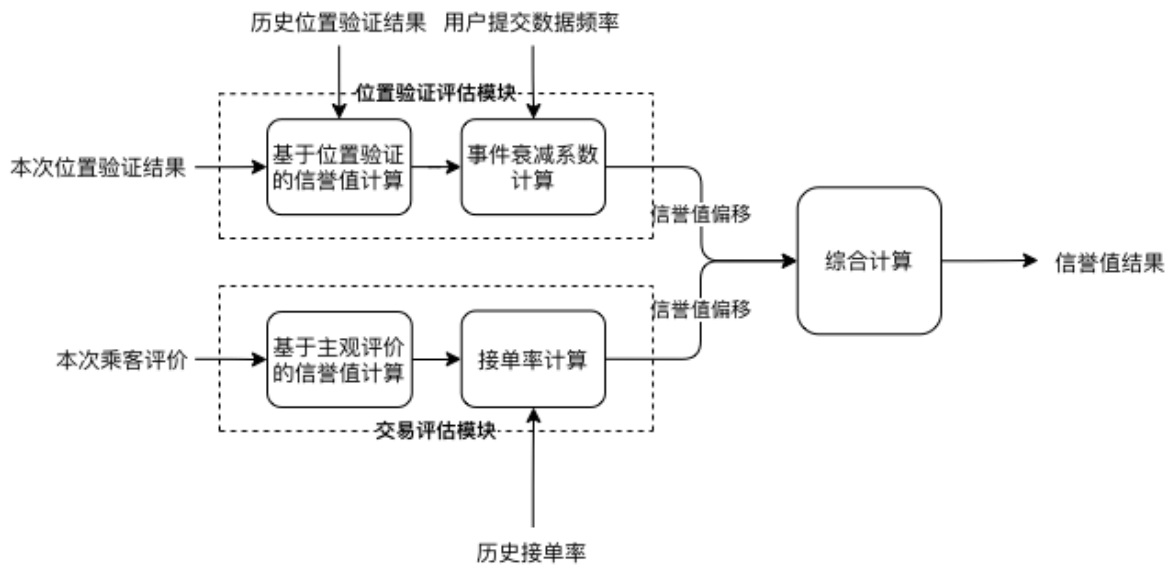


图 4-1 信誉值评估模型结构

在该模型中，目标车辆与邻近车辆不断发生位置消息交互，并验证位置信息真伪，将验证结果上传，通过历史位置验证结果平滑信誉变化情况。事件衰减系数的选择鼓励司机尽可能多的验证自己位置信息，从而保证较好的信誉提升。同时，为了切合车辆调度系统的应用场景，在交易结束后乘客提交对于本次服务的评价，通过历史接单率调整用户评价影响权重，该参数设计鼓励司机尽可能多的接单，从而提升系统活跃度。最后通过综合衡量两类信誉值偏移的影响权重，计算出最新信誉值结果。

## 4.2 算法设计

参考实验室前期工作，位置验证部分算法设计如下：

事件衰减系数：

$$T'_1 = \Delta T \times D(t) \quad (2-1)$$

其中  $T'_1$  为位置验证模块计算得出的信誉值偏移， $\Delta T$  是根据本次位置验证计算得出的偏移。 $D(t)$  是基于数据提交时间的事件衰减系数，具体计算如下：

$$D(t) = \frac{1}{\delta \times (t - t_1)} \quad (2-2)$$

其中  $\delta$  是参数， $t$  表示本次提交数据的时间， $t_1$  是上一次提交数据的时间。通过引入事件衰减系数，对车辆参与交互活跃度进行量化，使得参与活跃度能够调整交互结果在信誉值调整方面的影响大小。当间隔时间过长时，该参数会无限接近于 0，从而使得验证结果影响也显著减少；间隔时间较小时，用户活跃，应该对该种行为加以鼓励，信誉变化幅度增加，当验证结果较好时，能够有效提升车辆信誉值。

位置验证偏移：

$$\Delta T = \begin{cases} \max\{\theta_s \times (sRate + 1) + \theta_f \times fRate, 0\}, success \\ \min\{\theta_s \times sRate + \theta_f \times (fRate + 1), 0\}, fail \end{cases} \quad (2-3)$$

其中  $sRate$ ， $fRate$  分别表示目标车辆在近期一段时间内位置验证的成功率和失败率。 $\theta_s$  和  $\theta_f$  均是参数。其中引入了历史验证结果参数，能够平滑变化效果。

对于交易部分的评价，即乘客主观评价部分，本文设计如下：

$$T'_2 = I \times \theta_l \times \gamma \quad (2-4)$$

$$I = \begin{cases} -2, terrible \\ 2, normal \\ 4, comfortable \end{cases} \quad (2-5)$$

$$\theta_l = \begin{cases} 1, other \\ lRate \times \beta, lRate \geq 0.7 \& \& I < 0 \end{cases} \quad (2-6)$$

其中  $T'_2$  为交易模块计算得出的信誉值偏移， $\gamma$  是接单率， $I$  为乘客主观评价。 $\theta_l$  为检测乘客差评的敏感因子。

用户主观评价是复杂而又多样的，为了计算的简洁，在本文中抽象为三种评分来代表乘客对于本次交易的综合评价。同时将主观评分与接单率进行叠加，使得接

单率这一指标能够直接影响乘客评价对于信誉值的影响程度。好的评价会使信誉值增加，坏的评价会使信誉值减小，但是司机消极接单，接单率较低的情况下该种影响结果就会减弱，为了消除这样的影响，司机就需要积极接单增加乘客评价的影响效果。其中增加敏感因子  $\theta_l$  检测连续差评的情况，当乘客对于司机的差评率达到 70% 以上时，该算法将会增加对于差评的敏感度，有效督促司机改善服务质量。

最后，对着这两类信誉值偏移加权叠加，得到信誉值结果：

$$T = T_0 + \sigma_1 * T'_1 + \sigma_2 * T'_2 \quad (2-7)$$

其中  $T$  是计算得到的新的信誉值， $T_0$  是原始信誉值， $\sigma_1, \sigma_2$  为两部分信誉偏移所占权重，为常数因子。

考虑到各因素的影响力，其中位置验证部分信誉值变化体现出车辆提供的位置信息是否准确真实，这一点对于交易安全、防范恶意行为有很大的参考意义，故考虑为车辆位置验证信誉值偏移影响车辆信誉值变化方向，而乘客评价主要针对司机行为、交易服务质量进行评估，重要程度次之，故交易评价信誉值偏移影响车辆信誉值变化剧烈程度。

### 4.3 科学性分析

分析车辆的正常行驶状态，会不断与邻近车辆产生位置验证，并提交结果，同时在接收到路边单元匹配到的乘客打车订单后，可以选择接受订单或拒绝订单，若接受订单，在完成交易后会获得乘客评价。不断重复以上过程。在这个过程中正常情况下位置验证交互频率会远高于用户评价频率，所以信誉值变化会由位置验证结果主导，本文考虑到位置验证作为车辆信息真实性评估的重要指标，应该在信誉评估中占有更大权重，由此信誉值变化方向由位置验证结果主导的现象是合理的。分析以下场景车辆信誉值变化，探讨是否满足正常变化情况。

1. 司机正常驾驶，能够提供较为准确的位置信息。根据该算法设计，位置验证信誉值偏移与交易评价信誉值偏移均为正值，车辆信誉值逐步提升，符合设计要求。过程中存在少数因为客观原因造成的验证失败等，由于历史验证结果的平滑不会严重影响信誉值，即使存在下降，在司机检查并消除客观影响后能够很快回升。其中用户评价的算法设计，在位置验证效果良好时能够起增益效果，在位置验证效果较差时也能减缓下降幅度。



2. 司机存在不规范行为时，直观表现为乘客评价将大幅度下降，这个时候敏感因子开始发挥作用。在近期差评率到达 70% 及以上时，会使偏移大幅度增加，借此对司机的不当行为进行惩罚。该影响效果不会被位置验证结果覆盖，能够直接反映到信誉值变化中。
3. 乘客恶意评价时，单次恶意评价对于信誉值变化影响并不大，除非恶意用户连续提交多次差评才能对于车辆信誉值有较大影响，但是该系统设置为必须完成交易才能提交评价，所以除非有足够多的乘客对于同一司机服务均为差评，会使车辆信誉值大大降低，否则单次或少数恶意评价不会有较大影响。同样，位置验证部分，部分邻近车辆的恶意诋毁也不会有较大影响，通过历史验证结果的平滑效果，使得该类恶意诋毁行为不会严重影响车辆信誉值。

#### 4.4 信誉值评估合约

solidity 语言是面向智能合约的高级编程语言，针对以太坊设计。本文使用 solidity 完成 4.3 节所写算法。针对每辆车辆，维护一个用户对象 `user`，其主要参数结构如下，还有部分计算中间结果未展示：

Type	Variable name
sting	uuid
mapping(int256=>int256)	timeList
int256	status
int256	successValid
int256	failValid
int256	validNum
int256	takeNum
int256	rejectNum
int256	tmpEvaluateCount
int256	lowEvaluateCount
int256	credit

表 4-1 user 结构展示

完整算法主要由三部分构成，其一是位置验证部分，其二是接单率更新，其三是乘客评价部分，其余还有若干获取信息的接口，包括获取用户信誉值、获取用户接单情况、获取用户位置验证历史结果等接口。该合约类如图所示

Contract
user delta_rev Theta_s Theta_f Sigma_valid Sigma_evaluate
Function initUser() Function revalueByvalidation() Function revalueByOrder() Function revalueByEvaluate() Function clear() Function getCredit() Function getOrder() Function getValid()

图 4-2 合约结构展示

## 4.5 本章小结

该章节主要介绍了信誉值评估模型的设计与实现。首先分析了车辆在交易过程的信誉值影响因素，对重要因素进行了筛选，分别从真实性、活跃度、舒适性三个层面考虑，更加贴近实际应用场景，最终选取了用户活跃度、位置验证结果、接单率、主观评分这四个影响因子。在第二节中对具体评估算法进行设计，总体上该算法分为两部分，一部分为位置验证结果对信誉值的影响，另一部分为乘客评价对于信誉值的影响。在位置验证方面，本文引用了吴月仪对于位置验证影响算法的设计，在主观评价方面，本文将用户评价抽象为三类评分，其中中评和好评对于信誉值影响均为正值，差评为负值，旨在鼓励司机规范行为、提升服务质量，从而得到较好的评价，过于严苛的评分制度会影响司机积极性，评分相对宽松的设计更加符合实际应用场景。同时在本文提出的评估方案中，通过引入相关参数，如接单率、历史验证结果，评估模型达到平滑信誉变化效果、应对特殊情况等效果。第三节则是对上述算法进行了科学性分析，从使用场景的角度分析论证了算法的适用性。最后描述了通过 solidity 实现的智能合约的结构与参数。

## 第5章 测试与实验

本章首先简要介绍了服务器部署平台以及部署流程，介绍了本文测试所用到的两种工具，JavaScript 脚本以及 JMeter 测试软件；然后基于单一变量原则，对于前端模块使用时延以及信誉值各影响因子分别设计了测试方案，对于测试结果进行分析；接着对于信誉值影响因子间权重参数设计实验调整优化；通过设计性能测试实验，验证了区块链作为后台在车联网应用中的可行性；最后将该区块链车辆调度系统中各模块整合，在不同场景下模拟运行，分析了运行结果，验证了该系统的适用性。

### 5.1 区块链平台部署

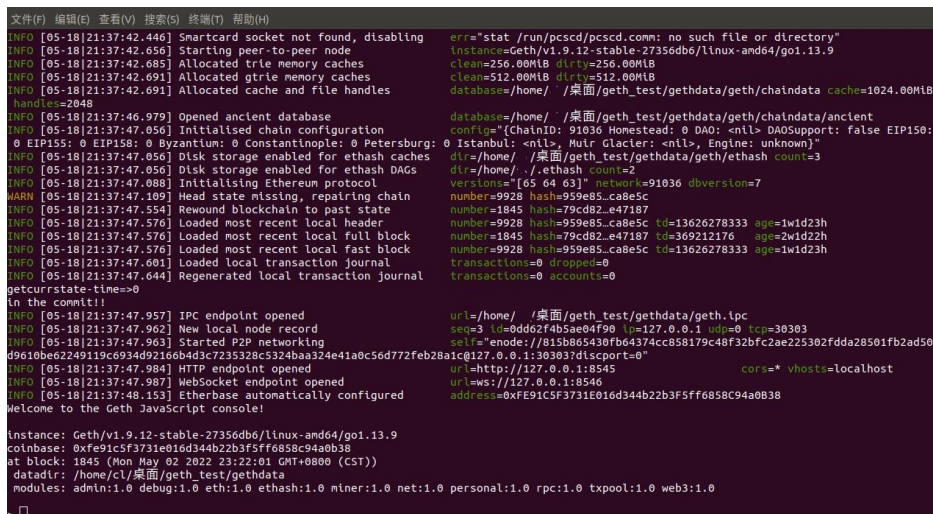
Geth 是由以太坊的官方平台。它提供了一个交互式 JavaScript 命令控制台，通过相关命令可以设置或查询区块链上相关信息。

在本文中主要涉及账户操作以及合约部署，具体流程如下：

1. 初始化区块链。区块链上除了第一个区块即创世区块外，其余区块均包含上一区块的 Hash 值以及时间戳，链接成链。由此，首先需要针对创世区块的相关信息配置，配置信息包括区块链 ID，挖矿难度，预设账户等信息，通过初始化该配置文件初始化区块链。
2. 启动区块链。通过启动命令，可启动上一步初始化的区块链，此时区块链已经正常启动，可以进行生成账户、挖矿等操作。
3. 部署合约。如第二章系统架构图所示，该系统所用合约有三个：地图合约，负责存储和读取地图数据；交通合约，负责车辆调度以及路径规划；信誉值合约，负责车辆信誉值管理和维护。将三者合约部署至区块链，并记录合约所在区块地址，该地址作为 web3.js 合约访问地址。

在本节的测试中，乘客与司机的账户均已提前生成并解锁，无需在测试环节负责该部分工作。注册新的账户时，区块链会返回给用户一对公钥和私钥，其中公钥后 20 个字节作为用户在区块链的账户地址，私钥作为操作该账户的唯一钥匙，且由创建账户时输入的密码加密生成。

值得说明的是，在一个节点上部署的合约，后续会根据区块链的共识机制在全网中扩展，智能合约会定期检查自动机状态，将未完成的事务推送至待验证的队列，



```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
INFO [05-18|21:37:42.446] Smartcard socket not found, disabling
INFO [05-18|21:37:42.656] Starting peer-to-peer node
INFO [05-18|21:37:42.685] Allocated trie memory caches
INFO [05-18|21:37:42.691] Allocated gtrie memory caches
INFO [05-18|21:37:42.691] Allocated cache and file handles
handbase=2048
INFO [05-18|21:37:46.979] Opened ancient database
INFO [05-18|21:37:47.056] Initialised chain configuration
0 EIP155: 0 EIP158: 0 Byzantium: 0 Constantinople: 0 Petersburg: 0 Istanbul: <nil>, Muir Glacier: <nil>, Engine: unknown}
INFO [05-18|21:37:47.056] Disk storage enabled for ethash caches
INFO [05-18|21:37:47.088] Initialising Ethereum protocol
WARN [05-18|21:37:47.109] Head state missing, repairing chain
INFO [05-18|21:37:47.554] Rewound blockchain to past state
INFO [05-18|21:37:47.576] Loaded most recent local header
INFO [05-18|21:37:47.576] Loaded most recent local full block
INFO [05-18|21:37:47.576] Loaded most recent local fast block
INFO [05-18|21:37:47.601] Loaded local transaction journal
INFO [05-18|21:37:47.644] Regenerated local transaction journal
getcurrstate-time=>0
in the commit!!
INFO [05-18|21:37:47.957] IPC endpoint opened
INFO [05-18|21:37:47.962] New local node record
INFO [05-18|21:37:47.963] Started P2P networking
d9610be624919c6934d92166b4d3c7235328c5324baa324e41a0c56d77272f2b28a1c@127.0.0.1:30303?discport=0
INFO [05-18|21:37:47.984] HTTP endpoint opened
INFO [05-18|21:37:47.987] WebSocket endpoint opened
INFO [05-18|21:37:48.153] Etherbase automatically configured
Welcome to the Geth JavaScript console!

Instance: Geth/v1.9.12-stable-27356db6/linux-amd64/go1.13.9
coinbase: 0xfe91c5f3731e016d344b22b3f5ff6858c94a0b38
at block: 1845 (Mon May 02 2022 23:22:01 GMT+0800 (CST))
datadir: /home/cj/桌面/gets_test/gethdata
modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

err="stat /run/pcscd/pcscd.comm: no such file or directory"
instance=Geth/v1.9.12-stable-27356db6/linux-amd64/go1.13.9
clean=256.00MiB dirty=256.00MiB
clean=512.00MiB dirty=512.00MiB
database=/home/ /桌面/gets_test/gethdata/geth/chaindata cache=1024.00MiB
database=/home/ /桌面/gets_test/gethdata/geth/chaindata/ancient
config="{ChainID: 91036 Homestead: 0 DAO: <nil> DAOsupport: false EIP150:
0 Istanbul: <nil>, Muir Glacier: <nil>, Engine: unknown}"
dir=/home/ /桌面/gets_test/gethdata/geth/ethash count=3
dir=/home/ /桌面/gets_test/gethdata/geth/ethash count=2
version="[65 64 63]" network=91036 dbversion=7
number=9928 hash=959e85.ca8e5c
number=1845 hash=79cd82.e47187
number=9928 hash=959e85.ca8e5c td=13626278333 age=1wid23h
number=1845 hash=79cd82.e47187 td=369212176 age=2wid22h
number=9928 hash=959e85.ca8e5c td=13626278333 age=1wid23h
transactions=0 dropped=0
transactions=0 accounts=0

url=/home/ /桌面/gets_test/gethdata/geth.ipc
seq=3 use=0dd6d2f4b5ae04f90 ls=127.0.0.1 udp=0 tcp=30303
self="enode://815b865430fb64374cc858179c48f32bfc2ae225302fdda28501fb2ad50
@127.0.0.1:30303?discport=0"
url=http://127.0.0.1:8545 cors=* vhosts=localhost
url=ws://127.0.0.1:8546
address=0xfE91c5f3731e016d344b22b3f5ff6858c94a0b38
```

图 5-1 区块链启动后界面

共识成功后，事务即可成功执行，循环该过程。所有事务处理和状态的切换均由区块链底层逻辑自动完成，不需要额外第三方参与，所有步骤全程透明，不可篡改。

## 5.2 客户端测试工具

### 5.2.1 脚本测试

Node 是一个开发工具，能够让 JS 运行在后端，不包含浏览器中 DOM 和 BOM，但是提供其他新模块，包括 Http，fs 等，为 JS 脚本提供相关接口。Node.js 是一个基于 Chrome V8 引擎的 JavaScript 运行环境（runtime），实质是对该引擎进行了封装，使得 V8 可以在非浏览器环境下运行。

JavaScript 本身是单线程语言，通过 Node 可以在不增加额外线程的情况下实现事务的并发处理，利用事件循环（Event loop）实现并发操作。所以可以通过非阻塞方式实现用脚本模拟乘客与司机的交易操作。

在本章中脚本测试环节即是通过编写 JavaScript 测试脚本，监控各环节执行情况。其运行流程如第三章图 3-1 所示。通过该脚本可以模拟用户行为，完成完整的打车交易流程，并且通过 Web.js 接口访问合约接口能够避免打开浏览器处理器性能消耗，减少响应时间，同时能够更好的记录步骤响应信息，方便各步骤单独测试。

### 5.2.2 浏览器模拟测试

JMeter 是一款压力测试工具，被设计为用于测试客户端/服务端结构的软件。它可以用来测试静态和动态资源的性能，同样也可以用来模拟运行负载，测试服务器整体性能。

selenium 是一个用于 Web 应用程序测试的工具，selenium 测试直接运行在浏览器中，模拟用户操作，比如用户输入、点击、滚动窗口等操作都能模拟运行，支持多种浏览器。

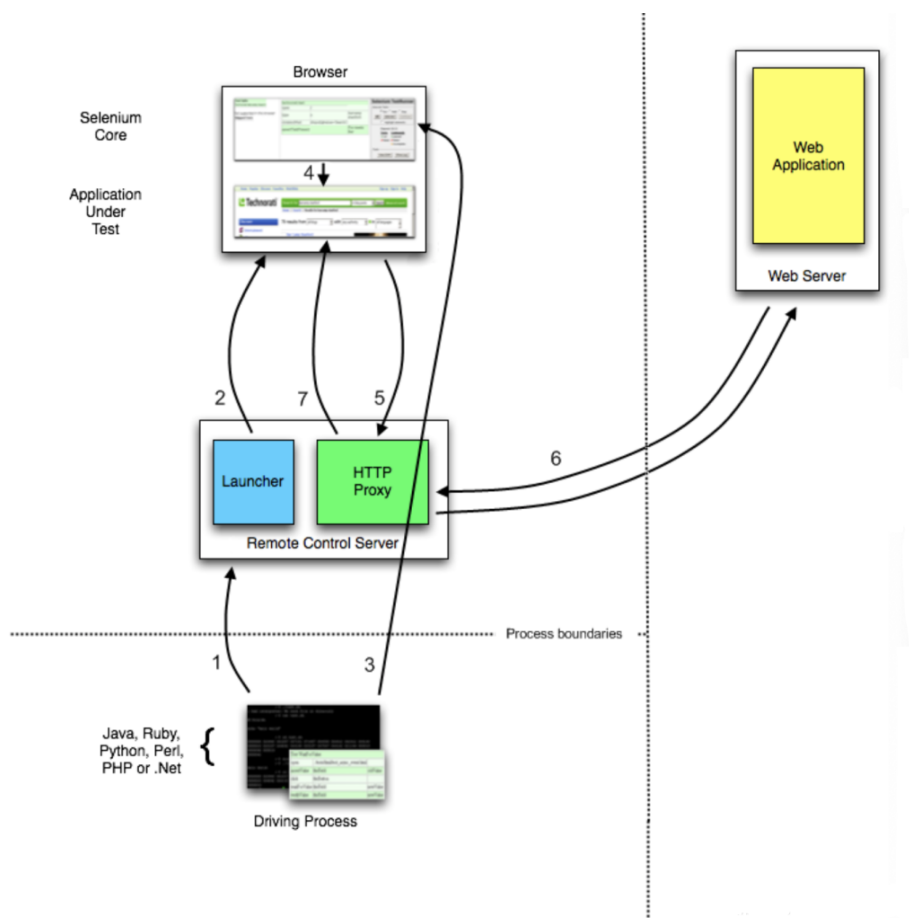


图 5-2 JMeter 测试原理结构图

本文中主要使用 JMeter+selenium+Webdriver 完成负载测试。大致流程如下：

1. JMeter 中根据设置创建测试用例，即创建不同线程。
2. 线程中首先发送 Http 请求，请求建立连接，成功后通过 Launcher 启动对应浏览器加载 Http 请求返回的页面，并将浏览器代理设置为 Http Proxy。

3. 根据 JMeter 脚本中的指令，自动触发事件或等待响应直至完成测试。整个过程中可添加 JMeter 中的监视器元件，记录运行情况。

### 5.3 前端使用时延测试

为测试整合后的前端接口时间消耗情况，设定在理想情况下，即并发请求数为 1，重复交易 50 次，在充分利用服务器与客户机处理性能的情况下，测试通过浏览器完成完整交易与通过 web3 接口直接与智能合约交互耗时对比，结果如图 5-3

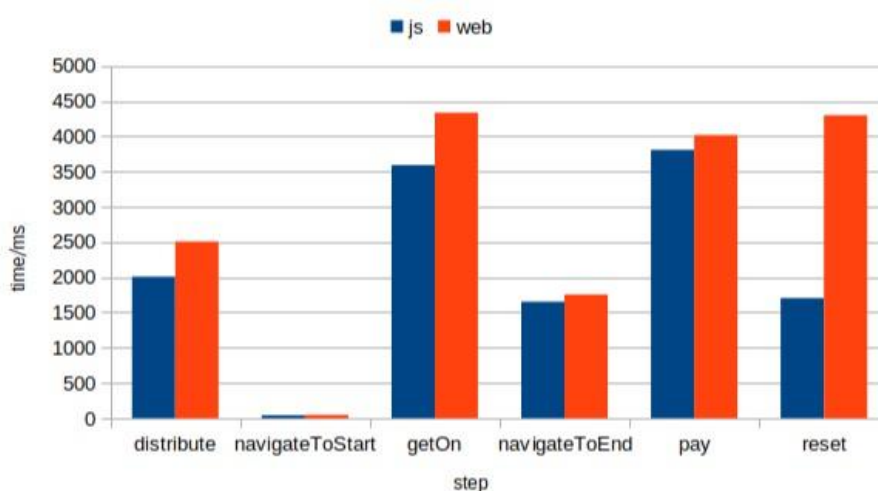


图 5-3 浏览器与脚本运行时间对比

该次测试中将完整的交易过程拆分为六步，分别是：匹配车辆（distribute）、计算出前往上车点的路径（navigateToStart）、乘客确认上车（getOn）、计算出前往目的地的路径（navigateToEnd）、乘客支付（pay）、车辆重置状态（reset）。可以看到，直接访问接口的方式所用时间均略高于通过浏览器访问的方式，但两者差距较小，除重置步骤外时间差距均在 1000ms 及以内，重置步骤对于浏览器接口方式包括浏览器窗口的关闭与打开，存在较大时延，对于用户体验来说，该差距是可以接受的，这说明了前端接口部分时间消耗在可接受范围之内。牺牲一定的时间，支持更加友好的使用接口是可以接受的且有必要的。

但是，可以知道的是当启动的客户端数量增加时，对于测试机的性能消耗也会相继增加，该差距会大幅度增加，同时不满足真实使用场景，故后续测试仍旧采用直接通过 web3 接口与智能合约交互的方式，不启动浏览器。

## 5.4 信誉值模块测试

### 5.4.1 位置验证模块测试

基于真实的交通状况，该实验模拟了一条道路上，有 20 台均匀分布的邻近车辆的情况。旨在观察其它因素不影响的情况下，车辆间的位置验证对于目标车辆的信誉值影响。

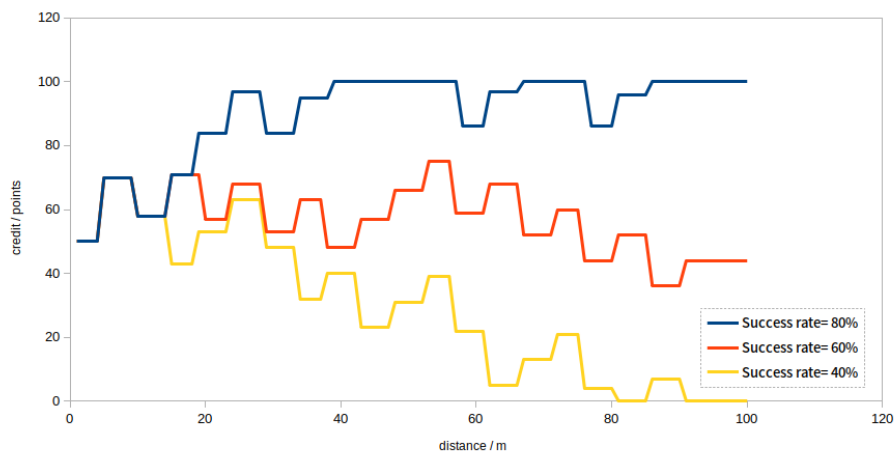


图 5-4 不同场景下位置验证对信誉值影响结果比较

可以看到，在位置验证成功率较高的情况下，目标车辆的信誉值曲线呈上升趋势，符合预期，数值波动也较为平缓；在验证成功率较低时，整体趋势呈下降状态，波动平缓。三种情况下信誉值曲线波动都符合预期结果，未出现急剧增加或急剧降低等异常情况，整体变化较为平稳，可以证明吴玥仪参数选择合理。

### 5.4.2 主观评价模块测试

根据真实评价状况，选择一个较为常见的评价分布模型测试该信誉值评价模块对于接单率的敏感率。本次试验设定在 20 公里内完成 20 次交易，交易评价中均匀分布 10% 的差评，20% 的好评，以及 70% 的中评，测试不同接单率情况下目标车辆信誉值变化情况，结果如图 5-5 所示

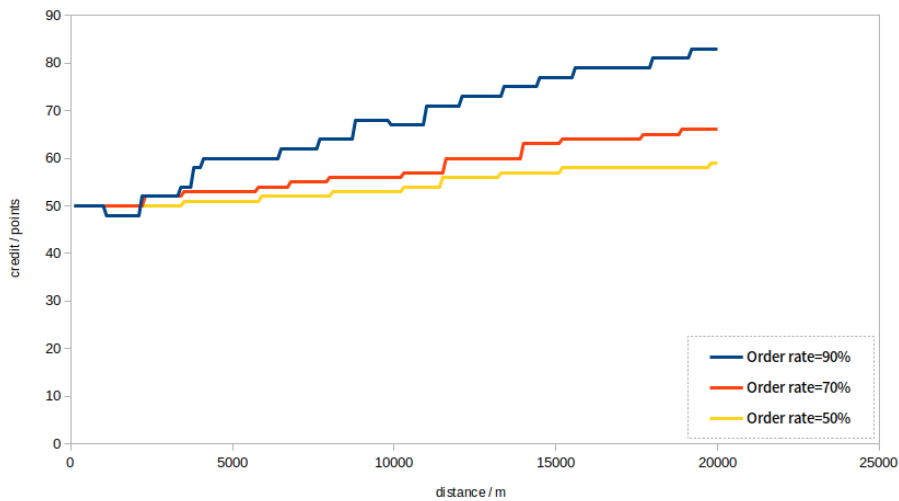


图 5-5 不同场景下接单率对信誉值影响结果比较

可以看到，在不同的接单率情况下，因为评价情况良好，总体上信誉值曲线均是上升状态，但是根据接单率的不同，信誉值上升幅度有显著差距。由此，当司机信誉值较低时，乘客好评带来的有效增益大幅度减少，为了解决该问题，司机需要积极接单提升自己的接单率。因此，该算法模型能够有效鼓励司机尽可能多的完成交易行为，由此创造积极的交易环境。

在交易主观评价模型中，本文将乘客评分抽象为-2，2，4 三类，对应差、良好、优秀三类评价。为了检测该算法对于不同评价分布的敏感度，设计三类评价分布场景：

1. 95% 好评，5% 中评。对应司机行为表现优秀，能够为绝大多数乘客带来舒适良好的乘车体验的情况。
2. 80% 好评，15% 中评，5% 差评。对应司机行为良好，能够满足大多数乘客的质量要求，存在极少数差评现象。
3. 60% 好评，20% 中评，20% 差评。对应司机行为一般，存在部分问题，差评数较多。
4. 40% 好评，20% 好评，40% 差评。对应司机行为较差，不满足多数乘客的乘车要求，存在较多的问题。



针对四类场景，保证目标车辆不受其他因素影响，观察乘客评价对于目标车辆信誉值的影响，结果如图 5-6 所示。

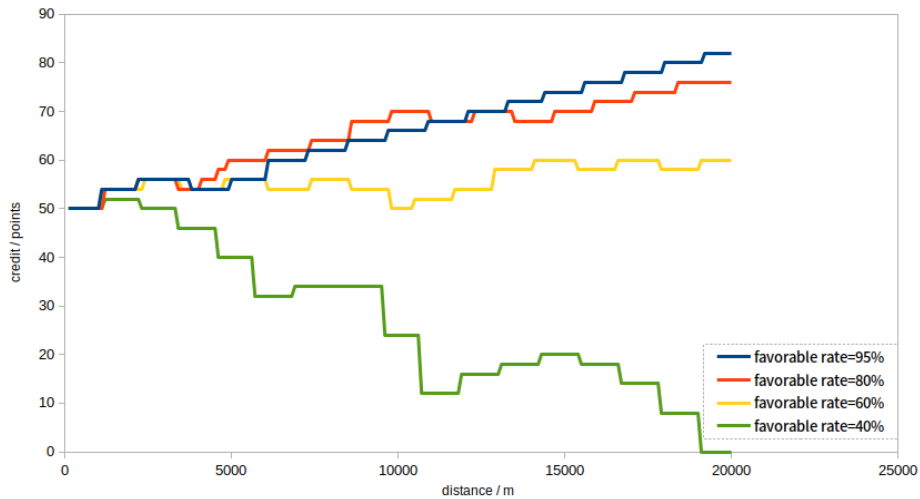


图 5-6 不同场景下乘客评价对信誉值影响结果比较

根据图中四条信誉值曲线变化可以看到该设计对于四类场景有较好的响应，同时其中差评敏感因子的设计针对差评较多的情况下，对信誉值有较大的惩罚力度。其中中评及好评对于信誉值的影响都是正向的，保证了司机在确保规范行为的情况下，信誉值都会有良性增加，总体而言该评价模型对司机行为起正向鼓励作用。

## 5.5 信誉值综合计算参数选择

前文单独测试了位置验证评估模块与交易评估模块的信誉值敏感度，在这一部分将对二者权重分析探讨，即针对  $\sigma_1$ ,  $\sigma_2$  取值进行评估。根据真实交通数据，设计目标车辆在 20 公里内的信誉值交互事件，其中包括 200 次的位置验证事件以及 20 次乘客评价事件，为了充分观察乘客评价的影响效果，设置接单率保持 100%。

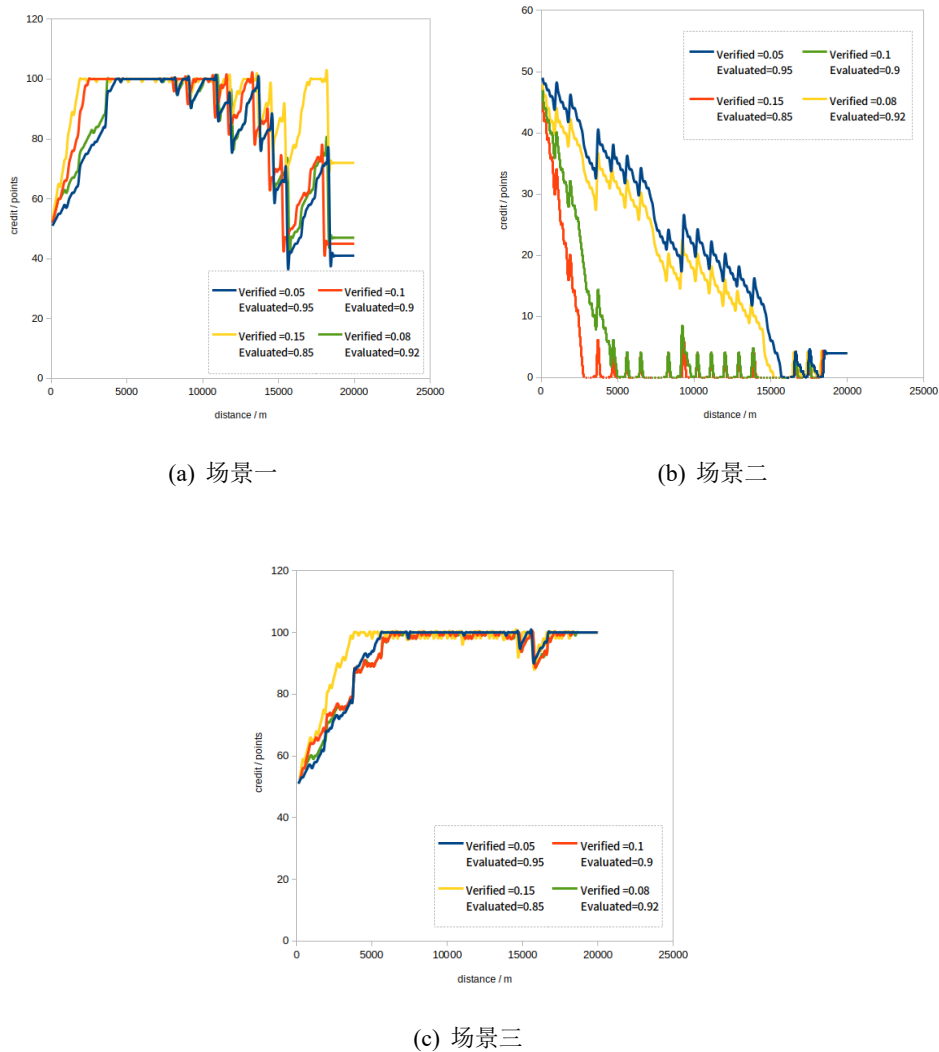


图 5-7 不同场景下不同参数设置信誉值变化情况

图 5-7 中分别为三个场景下信誉值变化情况：

1. 好评率 40%，位置验证成功率 90%。代表司机上传数据准确，真实性可以保证，但是服务质量较差，乘客评价较低的情况。
2. 好评率 80%，位置验证成功率 40%。代表司机服务质量高，但是存在 GPS 定位模糊、上传虚假位置信息等问题。
3. 好评率 70%。位置验证成功率 70%。代表一般情况，司机服务及提供信息真实性均相对良好。

如图中所示，场景一前半部分信誉值呈上升趋势，后期因为乘客评价过低，信誉值降低明显。四条信誉值曲线中，位置验证占比为 0.15 即黄色曲线走势与其余三条曲线相比有较大差距，原因是车辆位置验证对于信誉值影响权重过高，影响了交易评估模块的影响程度，没有对乘客评价状况作出有效响应；场景二中总体呈下降趋势，这体现了本模型对于数据真实性的重视程度。基于安全的角度，当车辆提供的信息不够真实时，应该对此作出严厉惩罚，场景二的四条曲线均符合该要求；场景三中，在司机数据真实性得到保证，且服务质量较好时，信誉值总体均呈上升趋势，中途对于乘客差评也能做出有效反应，四条曲线均符合要求。

综合考虑，从系统对交互时间、交互失败结果、差评的敏感度、信誉整体走势、数据波动性等方面进行评价，位置验证模块占比 0.08，交易评估模块占比 0.92 时变化曲线对两者响应度更好，因此参数选择结果为：

$$\sigma_1 = 0.08$$

$$\sigma_2 = 0.92$$

该组参数的选取，保证了信誉值变化对于位置验证以及用户评价均有较好的敏感度。针对合理频次的交互请求，系统能够有较为明显的反馈；同时，尽量保证了在交互频繁的情况下，数据波动保持在合理的范围之内，针对两类影响因子有较为合理的权重分配，避免了一类影响因子影响过大的情况。

## 5.6 性能测试

### 5.6.1 测试服务器环境

CPU 1.60GHz-1.8GHz, 内存 8192MB RAM

### 5.6.2 实验设计及结果

为测试一个区块链节点对于用户请求的处理能力，将用户请求分为 write 和 read 两类，其中 write 类代表需要向区块链写入数据的请求，该类请求需要通过挖矿将数据写入区块，永久保存，所以响应时间较长；read 类请求仅需要查询区块链上数据，返回结果，不对链上数据作任何更改，此类请求响应时间较快。测试相同环境下，对于不同规模的并发请求，区块链响应所需时间，测试结果如图 5-8 所示。

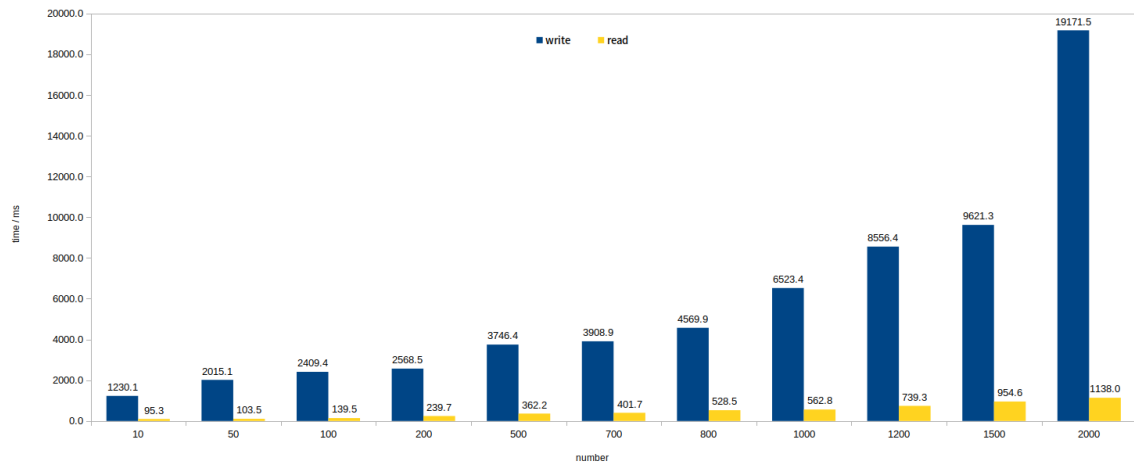


图 5-8 不同规模的读写请求的响应时间对比

在该实验中，用户在提交 **write** 类请求后不需要阻塞等待响应结果，可以继续执行后续事件，所以该类请求的响应时间对于用户规模影响较小；**read** 类请求延时较小，在规模为 1000 左右时延迟在 500ms 左右，对于用户使用不会产生较大的影响，满足实际使用。读写类请求实验的表现证明了区块链后台能够良好的应用于车联网应用中。

## 5.7 综合实验

### 5.7.1 模拟数据运行

在 6km\*8Km 的区域内，根据场景初始化不同数量的车辆，设定目标车辆不间断运行，在行驶过程中不断与周围车辆发生位置验证交互，同时在交易过程中接收乘客评价，观察目标车辆在 2 小时内信誉值变化情况。根据新京报 2019 年 12 月发布的道路拥堵数据，该实验设计了拥堵情况下和普通情况两类实验场景：

- 拥堵情况：在各个路口处初始化 30~50 辆车，模拟严重拥堵的情况；
- 普通情况：在各个路口初始化 0~10 辆车，模拟畅通运行时车辆周围情况。

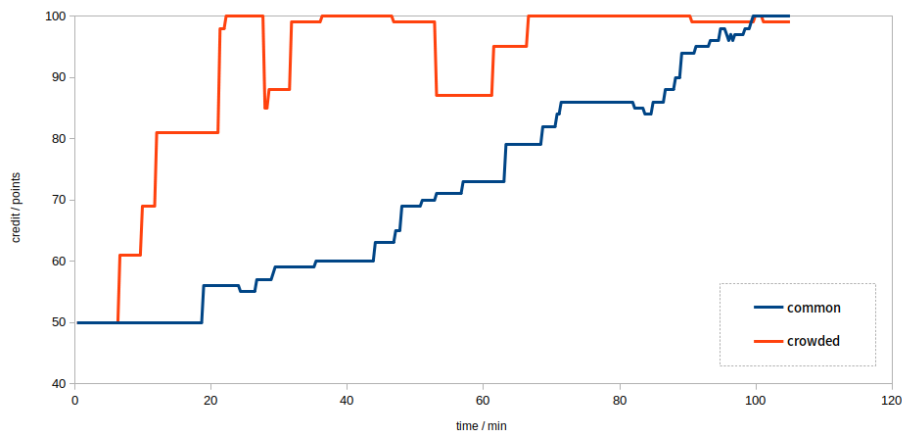


图 5-9 拥堵和普通情况下信誉值变化情况

从图 5-9 中可以看到，拥堵情况下车辆间交互更为频繁，信誉值变化也更为剧烈，目标车辆信誉值很快增加至 100，但在后续验证失败的情况下，也会带来较大影响。区块链在写入数据时，会将一定时间内提交的请求打包写入同一区块中，这造成了在交互密集的情况下信誉值变化剧烈的现象，但总体而言，信誉值在一个较为合理的范围内变化。普通情况下，信誉值变化较为平稳，变化情况也在预期内。

该实验将信誉值模块与路径规划算法相结合，目标车辆根据规划路线行程，图中不断与其他车辆发生交互，影响自身信誉值变化。实验结果证明了该信誉值评估方案应用于车辆调度系统的可行性和合理性。同时也可以看出在交互比较频繁时信誉值变化较为剧烈，很快到达了最高信誉值 100，使得目标车辆对于后续交互结果的响应不够敏感，这证明了目前算法仍有改进优化空间。

### 5.7.2 真实数据运行

在上述实验的基础上，根据真实运行数据，设计实验场景如下：目标车辆在两个小时内的不间断运行，不断完成分配到的乘客订单，共计完成五次交易，行驶过程中不断与邻居车辆完成车辆验证，共计完成 494 次交互验证。为观察车辆信誉值变化情况，不清除车辆历史行驶路径。运行结果如图 5-10.5-11 所示。

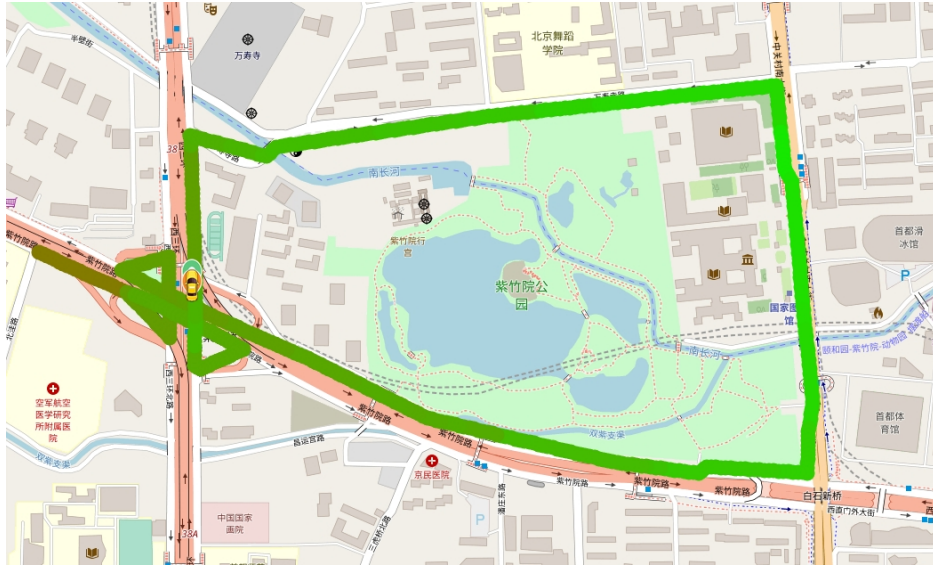


图 5-10 目标车辆多次交易情况展示

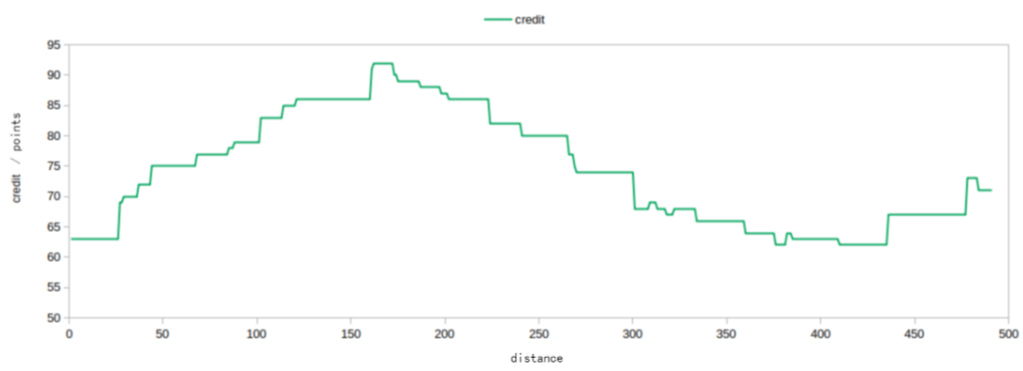


图 5-11 目标车辆多次交易情况展示

可以看到车辆初始信誉值较低，后在行驶过程中与周围车辆交易过程中完成成功验证，信誉值逐渐增加，且用户评价均为中评及以上，所以整体曲线都呈现上升趋势。车辆行驶至后半段时，到达高速公路，此处车辆较为密集，且定位产生一定偏差，导致车辆位置验证率明显降低，信誉值曲线也相对降低，末尾阶段通过验证成功率增加以及乘客好评有少许回升。

整体上分析信誉值结果可以看到对于车辆正常行驶行为，信誉值评估算法对其评估均呈正向状态，未出现低于初始值的情况。且在乘客评价良好的情况下，能够中和一部分车辆验证失败的损失，能够平衡交通状态等外部情况较差时的影响，同

时也能鼓励司机为乘客提供更好的乘车体验。但图中下降部分也能说明该平衡效果并不会改变信誉值变化趋势，仅会平滑变化曲线，使得车辆信誉值不会过度下降。

## 5.8 本章小结

本章首先对于实验和运行的区块链平台部署进行了介绍，后介绍了相关测试工具。在第3节中对于浏览器端使用时延进行了测试，通过实验证明相比于直接与合约接口进行交互，用户通过浏览器端完成交易的过程存在一定的时延，但是该延迟均在接受范围之内，通过一定时间的牺牲，实现更为友好的数据交互窗口是有必要的。对于信誉值评估模型方面，首先分模块测试了各模块的影响效果，可以证明该算法对于在单因素影响的情况下表现良好，在不同场景下的使用结果均能满足预期。然后对于各模块影响参数进行测试选择，从系统对时间和交互失败结果的敏感度、信誉整体走势、数据波动性等方面进行评价，综合考虑得到了表现较好的参数。通过性能测试，测试了区块链后台对于 `write` 和 `read` 两类请求的响应时间，证明了将区块链应用于车辆调度系统中的可行性。最后设计在不同场景下，追踪测试一辆车辆，观察其信誉值变化，实验结果证明了该系统在真实数据场景下的实用性与可行性。

## 结 论

在智能交通高速发展的今天，如何提升人们出行的便捷程度和安全性是一个重要的话题。作为智能交通的重要一环，在线车辆调度即网约车已经成为了人们出行的重要选择，而区块链技术的发展为车辆调度系统的改革提供了新的机会。然而目前研究工作多聚焦于研究区块链的共识机制以及调度算法优化，未曾有较为完善的以区块链作为数据处理和存储平台的调度系统实现，缺乏实际应用。

本文针对上述问题，在实验室前期工作的基础上，对区块链车辆调度系统进行整合，完成了用户界面设计。该工作对于降低用户使用难度有着极大的作用，同时通过分析用户行为逻辑，可以逆向完善调度逻辑，补充必要信息，这对于系统的进一步优化也有着重要意义。

针对车联网应用中的安全问题，本文结合实验室早期车载自组网中车辆信誉评估算法设计<sup>[20]</sup>，分析完善了信誉值影响因素因子，使得该评价模型能够更加贴近真实使用场景，对于车辆信誉能够有更加全面真实的评估管理。通过相关实验证明本文提出的信誉值评估方案的合理性和科学性，针对不同场景均有较好的适用效果。此外。还将信誉评估模型整合至区块链车辆调度系统中，乘客能够查看匹配司机的信誉信息。司机也能实时查看自己的信誉变化情况，进一步完善了调度系统的设计。

整体而言，本文工作较好的完成了一套基于区块链的车辆调度系统的完善与运行。但仍有可拓展的研究方向和不足之处：

1. 针对用户身份认证部分仍有欠缺，如何保证准入用户身份的真实性，避免冒认身份等安全问题仍然等待解决。
2. 当前调度算法未能充分利用环境信息，可以将路况等交通信息纳入车辆调度逻辑的考虑范围内，使其对于真实道路情况能有更加灵活的应对。
3. 信誉值评估算法可以加入更多特殊场景的设计，比如极端恶意事件的举报等特殊事件管理，使其能够针对不同场景有更好的响应。



## 参考文献

- [1] De Vany A. Capacity Utilization Under Alternative Regulatory Constraints: An Analysis of Taxi Markets[J]. Journal of Political Economy, 1975, 83: 83-94.
- [2] Lee X t, ZHAO C g. Research on Market Regulation of Online Car-hailing Travel Based on Tripartite Game[C]//2019 International Conference on Modeling, Analysis, Simulation Technologies and Applications (MASTA 2019)(2019 年建模、分析、仿真技术与应用国际会议) 论文集. 杭州, 2019: 81-88.
- [3] He F, Shen Z J M. Modeling taxi services with smartphone-based e-hailing applications[J]. Transportation research, Part C. Emerging technologies, 2015, 58C(Sep. Pt.A): 93-106.
- [4] Li Q, Malip A, Martin K M, et al. A Reputation-Based Announcement Scheme for VANETs[J]. IEEE Transactions on Vehicular Technology, 2012, 61(9): 4095-4108.
- [5] 蔡晓晴, 邓尧, 张亮, 等. 区块链原理及其核心技术[J]. 计算机学报, 2021, 44(1): 84-131.
- [6] Dua A, Kumar N, Bawa S. A systematic review on routing protocols for Vehicular Ad Hoc Networks[J]. Vehicular Communications, 2014, 1(1): 33-52.
- [7] 钟克华, 游东宝, 苏炳辉. 一种基于 Geohash 方法和地图路线规划的行程匹配算法[J]. 汽车电器, 2019(8): 16-18.
- [8] 杨柳青. 区块链技术在车联网中的应用研究[J]. 中国新通信, 2018, 20(6): 110.
- [9] 林康杰. 基于联盟区块链的移动车辆资源调度策略研究[D]. 2021.
- [10] 范君, 李茹, 张祎航. 一种适用于车联网任务调度的轻量级区块链架构[J]. 应用科学学报, 2021, 39(1): 99-108.
- [11] Uler H, Hartenstein H, Vollmer D, et al. Location-Based Routing for Vehicular Ad-Hoc Networks[J]., 2002, 7.
- [12] Huang X, Yu R, Kang J, et al. Distributed Reputation Management for Secure and Efficient Vehicular Edge Computing and Networks[J]. IEEE Access, 2017, 5: 25408-25420.
- [13] 李志展, 王永利, 宫小泽. 基于 DAG 分布式账本的车联网信誉值模型[J]. 信息安全研究, 2022, 8(1): 55-61.
- [14] Gurung S, Lin D, Squicciarini A, et al. Information-Oriented Trustworthiness Evaluation in Vehicular Ad-hoc Networks[C]//Network and system security : Madrid, (sp), 2013: 94-108.
- [15] 孙士奇. 区块链技术的发展及应用[J]. 信息系统工程, 2018(10): 85-86, 88.
- [16] 栾方军, 张鹏旭, 曹科研. Geohash 编码在出租车巡游路线推荐中的应用[J]. 计算机与数字工程, 2020, 48(12): 2836-2842.
- [17] 殷凤梅, 陈鸿. Geohash 编码的 k 匿名位置隐私保护方案[J]. 武汉大学学报（理学版）, 2022, 68(1): 73-82.
- [18] 徐川, 丁颖祎, 罗丽, 等. 车联网中基于位置服务的个性化位置隐私保护[J]. 软件学报, 2022, 33(2): 699-716.
- [19] 张海亮, 张征. 基于 GeoHash 索引的 A 算法优化[J]. 火力与指挥控制, 2021, 46(6): 78-83.
- [20] 吴玥仪. 车联网信誉评估算法的设计与实现[D]. 清华大学, 2021.
- [21] Javali C, Revadigar G, Rasmussen K B, et al. I Am Alice, I Was in Wonderland: Secure Location Proof Generation and Verification Protocol[C]//2016 IEEE 41st Conference on Local Computer Networks (LCN). 2016: 477-485.
- [22] 李晓薇. Vue.js 前端应用技术分析[J]. 网络安全技术与应用, 2022(4): 44-45.

- [23] Zhou C, Lu H, Xiang Y, et al. GeohashTile: Vector Geographic Data Display Method Based on Geohash[J]. ISPRS International Journal of Geo-Information, 2020, 9(7): 418.
- [24] Kerrache C A, Calafate C T, Cano J C, et al. Trust Management for Vehicular Networks: An Adversary-Oriented Overview[J]. IEEE Access, 2016, 4: 9293-9307.

## 致 谢

感谢这明媚春日。疫情之下，在宿舍还算明亮的灯光里写下这篇致谢，2022年，是在北理的第四年，也是疫情的第二年。大学生活因为疫情仿佛割裂成了两个世界，口罩、消毒水、封禁、隔离……已经成了司空见惯，两年的大学时光就好像抽屉里的口罩，被收了起来。还好春天没有失约。

感谢所有师友。大学四年，还算平路坦途，是这路上或许匆匆而过，或许伴于左右的他们带着我一点点成长。感谢陆慧梅老师和向勇老师在这段时间对我的帮助，从引导我做自己想做的东西开始，到在过程中事无巨细的指导与解惑，两位老师给予我前进方向上的指引；实验室各位学长学姐的帮助同样让我受益匪浅。还有这四年里所有的授课老师，同窗，都给予我莫大的帮助，相携而行。

大学四年，感谢所有遇到的人，感谢所有遇到的欢愉与悲苦，都被岁月轻抚。

