

基于Spring Cloud的人才智库遴选系统的设计与实现

黄 静, 朱 旭

(浙江理工大学信息学院, 浙江 杭州 310018)

✉syhj_sy@163.com; zhiyitec@qq.com



摘 要: 为解决当下人才信息评定难度大, 筛选过程复杂、筛选效率低下等问题, 提出并实现了基于Spring Cloud(分布式服务)的人才智库遴选系统。系统采用Spring Cloud框架, 由于不同领域的人才评估指标的侧重有所不同, 根据人才智库的建设及对人才信息评定的需求, 文章主要针对科研型人才构建人才标签, 从基础信息、个人经历、荣誉资质三个方向进行评估, 同时使用图数据库存储海量的人才数据及人才之间的关联关系。以国家“千人计划”的人才评定为例, 通过将人才信息进行标签化处理, 并通过特定的评定流程对人才数据进行评估, 旨在建立起科学、客观的人才评定体系, 并能对评定出的人才数据进行相应的遴选。

关键词: 人才智库; 人才遴选; 人才评定; Spring Cloud; 图数据库

中图分类号: TP311.5 **文献标识码:** A

Design and Implementation of Talent Think Tank Selection System based on Spring Cloud

HUANG Jing, ZHU Xu

(School of Information, Zhejiang Sci-Tech University, Hangzhou 310018, China)

✉syhj_sy@163.com; zhiyitec@qq.com

Abstract: In order to solve the problems of difficult talent information evaluation, complex screening process and low screening efficiency, this paper proposes and implements a talent think tank selection system based on Spring Cloud (distributed service). Spring Cloud framework is adopted in the system. Due to the different emphasis of talent evaluation indicators in different fields, considering the construction of talent think tank and the needs for talent information evaluation, talent tags for scientific research talents are mainly constructed in the paper, and evaluation is carried out from three aspects: basic information, personal experience, and honor and quality. At the same time, the graph database is used to store massive talent data and the relationship between talents. Taking talents evaluation of the National "Thousand Talents Program" as an example, talent information is tagged and evaluated through a specific evaluation process, with the aim of establishing a scientific and objective talent evaluation system, and selecting the evaluated talent data accordingly.

Keywords: talent think tank; talent selection; talent assessment; Spring Cloud; figure database

1 引言(Introduction)

现代经济和生产效率的快速增长, 人才在我国经济发展和社会进步的发展过程中起到至关重要的作用, 在此过程中对专家信息的评定也越来越重要^[1]。虽然政府加大了对专家信息评审的投入, 但是随着互联网发展带来的信息爆炸问题, 使当前专家信息筛选过程变得越来越复杂, 专家信息评定错误率高, 难以满足专家信息资源管理的实际应用要求, 同时传统的人工筛选方式工作量巨大且很难找到真正满足国家、企

业需求的人才^[2]。

本文提出基于Spring Cloud的人才智库遴选系统, 以Spring Cloud为框架, 采用前后端分离的方式进行构建; 通过系统的业务需求将整个系统架构划分为若干个微服务模块, 保证每个服务的职责单一化, 通过这种方式实现系统架构的去中心化^[3]。为了支持海量的人才数据存储问题, 每一个模块下均有自己独立的数据库, 同时人才与人才之间庞大又复杂的关联关系, 造成传统的关系型数据库已经无法处理运

算,由此引入了图数据库^[4],它能提高人才检索速度,对人才信息标签之间的关联关系进行构建,使人才的遴选效率和质量得以较大的提升。

2 微服务架构(Microservice architecture)

随着服务端架构的不断迭代和更新,系统架构也逐渐由单体架构演变为面向服务的SOA(Service-Oriented Architecture)架构,由于SOA架构过度依赖消息总线,微服务MSA(Micro Service Architect)架构应运而生,它也是目前使用最广泛、最流行的分布式架构^[5]。

单体架构也被称为MVC(Model View Controller)架构,它由数据层、业务层和视图层三个部分构成。单体架构虽然具有易上线、易测试的优点,但是随着需求数量和开发人员不断增加,代码库中的代码快速增长,单体应用变得越来越臃肿,可维护性、灵活性逐渐降低,使得单体架构的维护成为一大难题,开发成本越来越高^[6]。因为所有的业务耦合在一起,会使代码愈发复杂和冗余,同时使用人数的剧增,也会使系统快速达到并发瓶颈,导致系统崩溃。此外,代码的耦合度较高及功能的复用,造成功能的优化也会变得更加复杂。常见的单体架构示意图如图1所示。

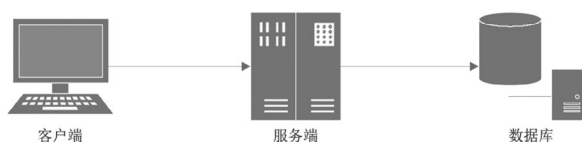


图1 单体架构示意图

Fig.1 Schematic diagram of single structure

微服务是从服务接口到业务逻辑层再到数据持久层,无论是逻辑上还是业务上都是独立和相互隔离的,不同于SOA架构那样需要服务总线接入。微服务是一种系统架构的理念^[7],它主要是将系统从中间层分解,将系统拆分成很多个小应用,服务与服务之间独立运行,互不影响,服务之间通过RPC(Remote Produce Call,远程过程调用)进行通信,保证数据传输的可靠性;微服务实现了数据、业务及代码之间的解耦,各个微服务之间独立开发,一个微服务只关注一个特定的业务功能,所以它的业务清晰、代码量较少。以微服务架构搭建出的应用是由若干个微服务构建而成的,而开发和维护单个微服务相对简单,所以整个应用也会被维持在一个可控状态。在微服务架构中,不同微服务之间可以使用不同的技术栈,可以结合项目业务及团队的特点,合理地地为每个微服务选择适用的技术。微服务架构示意图如图2所示。

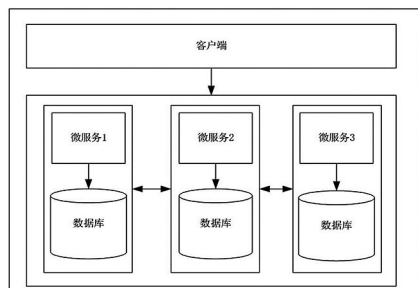


图2 微服务架构示意图

Fig.2 Schematic diagram of microservice architecture

3 系统相关技术选型(System related technology selection)

本系统前后端相互分离,独立部署。后端采用微服务的思想进行设计,每一个微服务均使用Spring Boot技术,结合Spring Cloud搭建整体的微服务架构;其中,数据库设计包含两种类型:关系型数据库采用MySQL 8.0.18,用于存储人才基础数据;非关系型数据库采用Neo4j,用于存储人才的标签数据及人才之间的关联关系。使用Redis(Remote Dictionary Server,远程字典服务器)作为系统的数据缓存,前端使用Vue.js进行平台的搭建。

Spring Boot^[8]是一个基于Java的开源框架,它可以帮助我们快速地搭建起一个微服务应用。传统的SSH(Struts, Spring, Hibernate)、SSM(Spring, SpringMVC, MyBatis)等架构开发复杂臃肿、耦合度高, Spring Boot大大简化了服务端开发过程,并提供了一套完整的解决方案。Spring Boot基于Spring 4.0设计,在Spring原有的优秀特性基础上,新增了自动化配置,通过提供一组依赖项解决大型项目依赖管理的问题。

Spring Cloud^[9]是分布式服务治理框架。Spring Boot能够快速开发单个微服务, Spring Cloud能够快速集成多个微服务,将微服务系统架构很好地应用到实际的应用开发中。使用Spring Cloud架构可以保证各个微服务之间有条不紊地协作,如利用“阿里巴巴”的Nacos实现高可用的服务注册中心及服务的配置管理等;为了使分布式系统更健壮,对依赖的服务使用Spring Cloud Hystrix(分布式服务容错保护)进行包装,通过线程隔离和熔断机制为系统的可用性保驾护航。系统还加入了网关服务组件Gateway用于请求的转发和服务集群的负载均衡。

Redis^[10]是一个高可用的缓存中间件。Redis的优秀特性不仅体现在读写速度上,同其他非关系型数据库相比,它还支持除字符串以外的多种数据结构,如列表、集合、哈希散列表等。Redis通常将全部的数据存储在内存中,性能十分高效,理论上扩展没有上限;由于它具有高并发的读写能力,所以在面对海量人才数据传输读取时,仍然能够轻松应对。

Neo4j^[11]是一个高性能的图形数据库,具有高性能的图引擎,这使得它具了所有成熟数据库的特性。Neo4j是一种基于图论实现的高性能非关系型数据库,其数据存储结构和数据查询方式都是以图论为基础的,图数据库主要用于存储更多的连接数据。随着人才信息的增多,人才标签的复杂化,人才与标签之间和人才与人才之间的联系在基于传统的关系型数据库下很难处理关系运算,因此引入图数据库。

Vue.js^[12]基于渐进式JavaScript框架,可以帮助开发者快速高效地构建用户界面。Vue.js因为被设计为可以自底向上逐层应用,所以使与其他大型框架有了明显的区分。Vue.js只关注视图层,使其不仅易于上手,还能方便地与第三方库或既有项目整合。当与现代化的工具链及各种支持类库结合使

用时,Vue.js也完全能够为复杂的单页应用(SPA)提供驱动。

4 系统的设计与实现(System design and implementation)

4.1 系统总体架构设计

基于Spring Cloud的人才智库遴选系统的总体架构图,如图3所示。

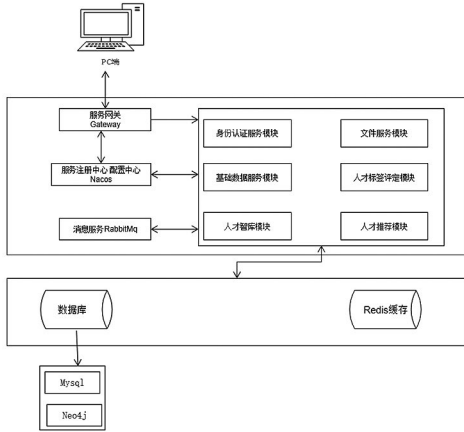


图3 系统总体架构图

Fig.3 Overall architecture diagram of the system

本系统设计主要分为四个部分,具体如下。

(1)数据库、Redis缓存:本系统将用户信息、人才信息、标签信息、人才基础数据(如顶级实验室数据、人才标准、国籍信息、地区产品导向等)存储至MySQL数据库中,便于进行人才信息匹配和人才信息过滤。考虑到人才之间的关联关系及人才关键信息的检索,将人才匹配成功后的标签数据和标签与人才、标签与标签的联系存储至Neo4j数据库。在人才标签匹配时,由于人才信息量较大,涉及标签信息数量较多,所以将人才的数据更新至缓存,从数据库匹配出的数据也暂存至缓存,待将人才类型匹配结束之后才清空缓存,达到减少与数据库的交互,进而提高系统的开发能力和人才标签匹配效率的目的。同时,本系统的身份校验基于JWT(JSON Web Token,网络令牌),所以在每次获取数据请求时进行的身份认证也会先访问缓存,若缓存中没有用户登录状态,则视为无效登录,禁止访问系统数据。由于系统有部分关键功能依赖于缓存,所以为了避免缓存穿透、缓存击穿、缓存雪崩等问题,在人才数据上加上数据的预处理,即上传人才数据时会先进行布隆过滤,同时是用分布式锁避免缓存击穿;缓存雪崩的主要原因是大批数据在同一时间过期导致,故在设置人才信息时,不设置过期时间,当人才信息评定完成之后再手动进行删除。

(2)服务端:将服务端根据业务需求划分为若干个微服务模块,保证每个服务职责单一,各司其职,同时依靠Spring Cloud提供的组件保证各个微服务模块的正常运行,如系统的统一访问入口Gateway,配置和注册中心Nacos,服务之间的通信Openfeign,服务的限流Sentinel等。整个系统的功能模块划分为用于每个服务统一身份认证的身份认证服务模块,

用于个人简历文件预览、简历上传等专门用于文件处理的文件服务模块,用于处理奖项、学历、企业经历、国家级项目的基础信息模块,用于筛选、录入、上传人才信息的人才智库模块,用于人才类型评定、标签匹配和打分的人才标签评定模块,用于保存人才与人才之间的关联关系和通过关键词检索人才数据的人才推荐模块。每个模块下都有自己对应的数据服务和数据管理。

(3)分布式事务:由于本系统一个功能的实现需要多个模块之间的协调才能实现,因此不同的微服务模块都有自己的数据库,此时就会出现一个服务必须调用多个服务下的数据库实例才能完成的情况。在分布式架构下,每个节点只知晓自己操作失败或者成功的状态,无法得知其他节点的状态。本系统基于RabbitMQ(高级消息队列)实现可靠的消息服务,解决服务间通信问题和不同微服务之间的事务问题。

(4)系统前端设计:本系统前端主要采用Vue.js框架,并配合Element-UI(桌面端组件库)和ECharts(商业级数据图表)实现良好的人机交互页面,通过数据的图表实现人才数据分析的可视化效果。

4.2 系统数据库设计

本系统数据库采用MySQL 8.0.18和Neo4j两种不同类型的数据库。由于Neo4j为图数据库,不同于关系型数据库,它不需要事先预定行列规范,所以在此不介绍Neo4j的设计,下面对MySQL数据库中部分表的结构设计进行说明。其中,人才基础信息数据库表结构如图4所示,人才基础信息数据库里包含人才所对应的基础标签数据,其中专家库(tb_profession)包含人才的姓名、年龄及系统匹配后人才类型等;其余的表皆为专家库的子表,包含人才的详细信息数据,如教育经历(education)、个人头衔(personal_title)、人才标准(tb_standard)、个人经历(experience)、专利(patent)及导师信息(tutor)等。

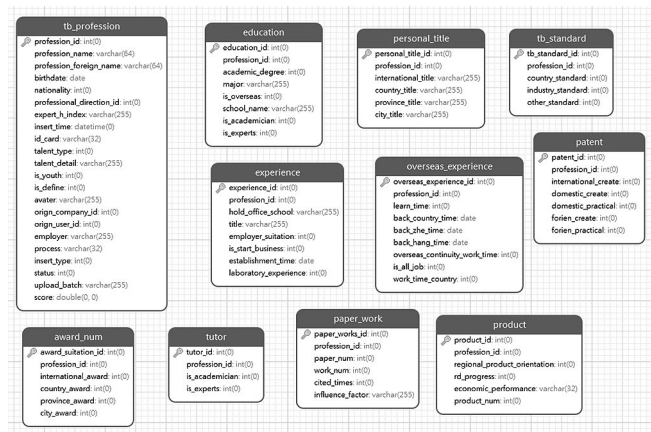


图4 人才基础信息数据库表结构图

Fig.4 Structure diagram of basic information database table of talents

图5为人才标签评定数据库表结构图,其中标签库(tb_tag)包含所有需要匹配的标签数据,通过条件库(query_

criteria)及条件关联库(query_criteria_association)匹配用户的人才遴选类型, 将匹配后的标签分值及结果保存至人才标签表(profession_tag), 同时将人才的人才遴选结果同步更新到人才基础信息数据库的专家信息表中。

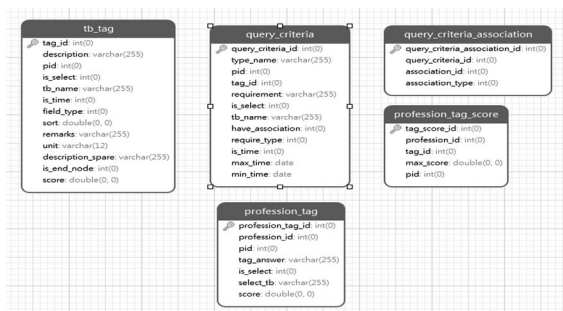


图5 人才标签评定数据库表结构图

Fig.5 Structure diagram of tag assessment database table of talents

4.3 系统服务端设计与实现

4.3.1 微服务框架设计

本系统基于Spring Cloud架构进行搭建, 当接口对服务端发起请求时, 每个请求都必须先经过网关, 然后由网关将请求进行统一的转发。本系统的服务端网关采用的是Spring Cloud全家桶中的微服务API网关Gateway^[13]。Gateway不仅提供了统一的路由方式, 还实现了网关的基本功能, 如网络请求安全、数据请求监控及服务的限流等。

由于不同微服务可能不在同一台服务器上, 甚至不在同一网络环境下, 所以服务间的通信只能依赖于网络, 而Openfeign则实现了不同服务之间的数据调用。为了防止服务雪崩, 本系统还引入了Spring Cloud Hystrix实现熔断机制, 当系统某个服务发生故障或异常时, 会直接熔断服务, 而不是一直等待该服务超时^[14]。

由于模块划分较多, 配置繁杂和冗余, 为了解决配置文件相对分散、配置文件无法区分环境及配置文件无法实时更新等问题, 引入基于Nacos的配置中心, 各个服务按需获取自己的配置。与Spring Cloud Eureka(分布式注册中心)不同, Nacos不需要引入额外的组件如Spring Cloud Bus(分布式消息总线)等实现配置的动态刷新, Nacos采用Netty保持Tcp(传输控制协议)的长连接实现实时推送, 当配置中心的各种参数有更新时, 也能通知各个服务实时同步最新的信息, 实现动态更新。

服务注册与发现中心采用的也是Spring Cloud Alibaba Nacos(阿里巴巴分布式注册中心)^[15], Nacos不仅具备良好的图形化界面和服务在线管理, 还极大地简化了整个微服务架构的复杂度和冗余度。

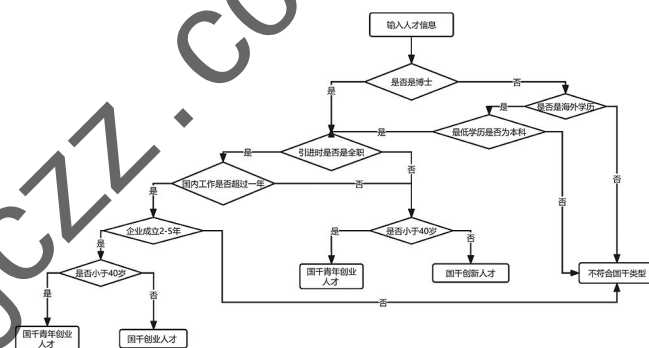
4.3.2 微服务功能设计

根据人才智库遴选流程, 将系统划分为身份认证模块、基础数据模块、人才标签评定模块、人才智库模块、人才推荐模块。

(1)身份认证模块。身份认证模块主要分为用户信息管理和身份信息校验两个功能。用户信息管理即对登录用户的姓名、密码、邮箱、角色等进行管理; 身份信息校验会拦截所有需要身份校验的请求, 并对其中的JWT信息进行校验, 包含登录信息、角色权限的校验, 只有校验成功才可以继续访问接口。

(2)基础数据模块。基础数据模块为其他微服务模块提供了基础的数据支持, 由于每个模块之间的数据有耦合, 所以把耦合部分提取出来作为公共服务模块, 如在录入人才数据时需要国籍列表、奖项列表、顶级实验室列表等数据, 但是在进行人才标签匹配时也需要这些数据做比对, 此时通过一个单独的微服务将这些数据统一进行管理。

(3)人才标签评定模块。该模块为系统的核心业务模块, 主要负责人才数据的标签匹配、判断人才的类型、对人才评分等操作。如图6所示的国家“千人计划”(全称为国家海外高层次人才引进计划)人才评定流程。



参考文献(References)

- [1] 赵猛.多元化人才培养下的高校奖学金评定制度研究[J].才智,2015(29):129.
- [2] 赵晓荣.基于胜任特征模型的辽宁省普通高校“百千万人才”遴选指标体系研究[D].大连:大连工业大学,2012.
- [3] 刘子豪.分布式一致性ADMM算法研究及其在RS译码中的应用[D].西安:西安电子科技大学,2021.
- [4] 韩鹏昊.基于图数据库的科技咨询数据资源管理平台研究与实现[D].北京:北京邮电大学,2021.
- [5] 吴兆跃.基于SpringCloud微服务架构的人工翻译系统的设计与实现[D].北京:首都经济贸易大学,2020.
- [6] 崔海涛,章程,丁翔,等.面向微服务架构的开发组织适应性评估框架[J].软件学报,2021,32(05):1256-1283.
- [7] 郑明钊,张建强.基于微服务的大平台系统架构演进探讨[J].软件,2017,38(12):165-169.
- [8] 梁莹.基于Spring Boot的教师企业实践管理系统的设计与实现[D].南宁:广西大学,2021.
- [9] SUN J N. Design and implementation of dormitory repair management system based on Spring Cloud Microservices[J].

(上接第62页)

户,并且数据的呈现方式多样化,符合用户的不同需求。本文通过构建SQL Model,将自然语言问题转化为查询语句应用于政务智能搜索引擎系统。

目前,NL2SQL还是一个比较新的技术,尤其是将其应用到电子政务领域,是一次创新性探索。通过将NL2SQL技术与实际业务需求相结合,能够有效地提升相关部门的行政工作效率。本文在保证数据安全、数据权限可控的基础上,围绕NL2SQL技术应用的关键难点和场景要求,开发了政务智能搜索引擎,采用用户提出自然语言搜索需求,解析引擎将自然语言搜索需求转换成SQL语言搜索需求的应用方式,将用户的自然语言输入转化为可用的SQL语句,然后自动输出用户所需的数据,在降低数据应用复杂度的同时,也实现了一切数据皆可查的目的,并且支持多维度复杂查询,使得业务部门数据搜索应用难度降低,极大地提高了政务数据搜索效率。

参考文献(References)

- [1] NOROUZIFARD M, DAVARPANAH S H, SHENASSA M H. Using natural language processing in order to create SQL queries[C]// IEEE. 2008 International Conference on Computer and Communication Engineering, Piscataway: Institute of Electrical and Electronics Engineers, 2008:600-604.
- [2] RAO G, AGARWAL C, CHAUDHRY S, et al. Natural language query processing using semantic grammar[J]. International Journal on Computer Science and Engineering, 2010, 2(2):219-223.
- [3] CHAUDHARI P P. Natural language statement to SQL query translator[J]. International Journal of Computer Applications, 2013, 82(5):18-22.
- [4] GHOSH P K, DEY S, SENGUPTA S. Automatic sql query formation from natural language query[J]. International Journal of Computer Applications, 2014, 975(1):8882-8887.

- International Core Journal of Engineering, 2021, 7(1):201-204.
- [10] 孔晓阳,代真虎.基于WebSocket与Redis的高性能Web组态系统设计[J].信息技术与标准化,2021(03):51-56.
- [11] 卜令彬.基于Neo4j图数据库的地理场景数据组织方法[D].南京:南京师范大学,2020.
- [12] 李成仁.基于Vue.js的单页面WebGIS可视化框架研究与实现[J].地理空间信息,2020,18(05):83-86,98.
- [13] 王燕,伍军,周云霞.基于Spring_Cloud微服务构架的局域网智能管理系统设计[J].软件,2022,43(01):8-10.
- [14] 肖飞龙,张爽,胡志凌.基于Neo4j的疾病预防与控制知识图谱研究[J].电子技术与软件工程,2021(22):180-182.
- [15] 周永圣,侯峰裕,孙雯,等.基于SpringCloud微服务架构的进销存管理系统的设计与实现[J].工业控制计算机,2018, 31(11):129-130,133.

作者简介:

黄静(1965-),女,博士,教授.研究领域:通信工程,大数据,深度学习.

朱旭(1997-),男,硕士生.研究领域:软件工程,深度学习,大数据.

- [5] REINALDIA F, WIDAGDO T E. Natural language interfaces to database (NLIDB): Question handling and unit conversion[C]// IEEE. 2014 International Conference on Data and Software Engineering (ICODSE), Piscataway: Institute of Electrical and Electronics Engineers, 2014:1-6.
- [6] SINGH G, SOLANKI A. An algorithm to transform natural language into SQL queries for relational databases[J]. Selforganizology, 2016, 3(3):100-116.
- [7] DHAMDHERE K, MCCURLEY K S, NAHMIAS R, et al. Analyza: Exploring data with conversation[C]// ACM. Proceedings of the 22nd International Conference on Intelligent User Interfaces, New York: Association for Computing Machinery, 2017: 493-504.
- [8] GAO Y, LOU J G, ZHANG D. A hybrid semantic parsing approach for tabular data analysis[J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2019, 2(2):1752-1754.
- [9] 孟小峰,王珊.数据库自然语言查询系统Nchiql中语义依存树向SQL的转换[J].中文信息学报,2001,15(5):40-45.
- [10] 于龙,殷建平,张国敏.关于数据库查询的自然语言接口研究[J].计算机工程与科学,2005,12(1):36-37.
- [11] 郝亮,张文东,袁春风.一种数据库汉语查询接口的设计与实现[J].计算机技术与发展,2010,20(6):13-17.
- [12] 杨鹤标,陈力.自然语言向SQL代码的转化方法[J].计算机工程,2011,37(023):72-74.

作者简介:

姚俊华(1973-),男,硕士,高级工程师.研究领域:智慧城市数字政府.

汤代佳(1975-),男,硕士,高级工程师.研究领域:智慧城市.