

FOLIO 的技术选型与运营模式研究*

郭利敏 张 磊

摘 要 FOLIO 作为开源的“第三代图书馆服务平台”,提供了一种全新的模块化应用环境。FOLIO 由图书馆、开发商、服务商共同参与建设,工作流的架构不仅能够很好地支撑现有的业务,还能灵活地支持图书馆未来可能发生的未知服务。本文结合微服务介绍 FOLIO 平台,通过性能测试比较说明其技术选型更适合高并发性的网络环境要求;同时介绍了 FOLIO 社区组织和商业模式,提出国内图书馆可以从技术推广和角色转变两个方面切入,以更好地融合 FOLIO 的生态环境。图 1。表 1。参考文献 8。

关键词 图书馆服务平台 FOLIO 微服务

分类号 G250.7

Technology Selection and Operation Mode of FOLIO

GUO Limin & ZHANG Lei

ABSTRACT

FOLIO, as a first open source project of the third-generation library service platform, provides a new modular application environment, which is jointly constructed by libraries, developers and service providers. The architecture design of its workflow can not only well support the existing business but also flexibly support the unknown services that may happen in the future of the library. The purpose of this paper is to explore the FOLIO open source library service platform with its advanced technical architecture and how domestic libraries get involved in FOLIO ecosystem. In this paper, combined with microservices and FOLIO, we introduce the new ideas brought by microservices—“Organized around Business Capabilities” and “Products not Projects” and explain what they are. And then we give an overview of the FOLIO platform history, organizational and system architecture. It was organized by OLE and funded by EBSCO and combined librarians, developers and vendors. We show the 4 layers of FOLIO system architecture and do some performance test comparison between Vert.x and Spring framework under high concurrency. The results show that FOLIO’s technology selection has a high-performance asynchronous network that library can support large-scale applications. After that, we explain why FOLIO chose PostgreSQL by choosing JSON data format to store dynamic data structures without changing the database structure. The paper introduces FOLIO community organization and business model. In FOLIO community, there are 3 roles—

* 本文系国家自然科学基金一般项目“面向数字人文研究的图书馆开放数据体系构建与服务模式设计研究”(编号:18BTQ027)的研究成果之一。(This article is an outcome of the project “Research on Library Open Data System Construction and Service Model Design for Digital Humanities Research”(No. 18BTQ027) supported by National Social Science Foundation of China.)

通信作者:郭利敏,Email:lmguo@libnet.sh.cn, ORCID:0000-0002-8138-2762 (Correspondence should be addressed to GUO Limin,Email:lmguo@libnet.sh.cn, ORCID:0000-0002-8138-2762)

Tech Council, Product Council and Special Interest Groups, and the paper shows how community works and how FOLIO is driven by community. Considering the sustainable development of FOLIO, the paper also considered the business model. We think there are two business models: one is that the core is open source and the extension is charge, and the other is providing SaaS platform support. From the above, we think there are 2 ways for the domestic libraries to get involved in FOLIO ecosystem. The first is to build FOLIO Chinese community and do something about the translation, technology promotion Secondly, to face the changing, the library should do the role transformation, from projects to products. 1 fig. 1 tab. 8 refs.

KEY WORDS

Library service platform. FOLIO. Microservice.

0 引言

近年来随着信息技术的发展,以及大数据、云计算和人工智能的兴起,图书馆的管理与服务也悄然发生着变化。知识服务、空间服务、泛化的移动服务、全方位的个性化服务逐渐成为图书馆的主要功能,这也促使图书馆管理系统从以业务为中心的第二代系统向以服务为中心的第三代系统更迭。为满足新服务的需求,图书馆近年来在原有的第二代系统基础上引入或自主开发了越来越多的模块,如上海图书馆为解决 OPAC 无法满足移动端业务需求的问题,在原有的书目数据基础上构建了全新的全文检索系统^[1]。然而,由于缺乏统一的规划,这些新增系统间的数据共享和交换成本不断增加,随着时间的推移,部分系统还存在着信息孤岛的隐患,进而制约着图书馆服务的效率。

美国著名的图书馆自动化系统专家 Marshall Breeding 于 2012 年提出了图书馆服务平台 (Library Services Platforms, LSP) 的概念^[2]。这一概念中,Library——明确平台的使用对象是图书馆,通过技术帮助图书馆实现自动化;Services——指出其技术架构为面向服务的软件架构,通过对外暴露网络服务和 API 接口来进一步促进图书馆向用户提供服务;Platform——表明该系统为平台化的服务系统,可通过平台的 API 来扩展功能,创建与其他系统的连接以及动态数据交互等^[3]。图书馆服务平台与当前使用

的第二代图书馆软件系统的单体架构不同,是一种全新的软件架构。在 LSP 的概念下,多家厂商基于 SOA 架构推出了下一代图书馆服务平台,如 OCLC 的 WorldShare Management Services、ExLibris 的 Alma、Innovative Interfaces 的 Sierra、Serials Solutions 的 Intota 等^[4-5]。

近年来随着软件技术的发展,基于云平台的微服务架构已逐步成为软件系统开发的主流,而这也为图书馆服务平台提供了可行的技术实现方案,即借助云计算和微服务架构为图书馆提供一个可伸缩、可扩展、持续改进迭代的图书馆服务平台。

1 基于微服务的开源图书馆服务平台——FOLIO

1.1 微服务简介

微服务自 2012 年被提出以来,作为一种新型的架构风格,不仅继承了传统的 SOA 架构的基础,还在理论和工程实践中形成了新的标准,发展至今已成为当下比较流行的一种软件架构。2014 年,马丁·富勒在 *Microservices*^[6] 一文中对微服务做出了纲领性的定义:在结构上,将原有从技术角度拆分的组件,升级为依业务拆分的独立运行的服务(这些服务有各自的实现,有自由数据,服务间以智能端点和哑管道的方式通信);在工程上,从产品而非项目的角度进行设计,强调迭代、自动化和面向故障的设计方法;从人员上,构成微服务的团队是跨职能的,

包含开发过程所要求的所有技能。运用微服务架构能够在很大程度上提高系统的伸缩性,方便业务之间的协作,消除系统中的信息孤岛。

微服务带来的不只是技术的再创新,对于研发团队组成、团队职能都提供了新的理念。

(1) 围绕业务功能的组织:在微服务中倾向围绕业务功能的组织来划分服务,这些服务针对其所在的业务领域提供多层次的软件实现,包括用户界面、持久化存储,以及任何对外部的协作性操作。因此,构成微服务的团队是跨职能的,包含开发过程所要求的所有技能:用户体验、数据库和项目管理。跨职能的团队同时还负责构建和运营每个产品,每个产品被分割成许多单个的服务并通过消息总线进行通信。

(2) 产品制而非项目制:目前大多数应用程序开发工作都采用项目制,一旦项目验收后,软件将移交给维护人员进行运营维护,构建软件的项目团队将被解散或指派别的项目。对于微服务而言,更倾向于认为团队应该在其整个生命周期内拥有产品,并对其负责。如亚马逊提出的概念“you build, you run it”,要求开发团队对生产中的软件负全部责任。因此,为了对软件进行支持,开发人员不得不在日常接触他们的软件,了解软件在生产中的各种情况(如软件bug分析、用户使用行为分析、运行中产生的新的需求的分析等),同时还需要增加与用户的联系;开发者要保持产品心态,与业务能力紧密相连,不是将软件视为项目要完成的功能,而是存在一种持续的关系,这种关系可以更容易地将开发者与用户联系在一起,持续改进项目。

1.2 FOLIO 简介

FOLIO 是由开放图书馆基金会(Open Library Foundation, OLF)主办,依托 OLE(Open Library Environment)社区,为图书馆员、开发人员、服务提供商和供应商之间提供深入合作的平台,FOLIO 由 EBSCO 公司资助,是基于下一代图书馆服务平台理念设立的一个采用 Apache2.0 开源协议的开源项目。该平台立足现有的图书

馆业务,创建可持续的现代技术生态系统,以社区协作为导向,利用开源应用程序,为图书馆提供管理图书馆资源和扩展图书馆价值的功能。

在开放图书馆基金会的组织下,FOLIO 项目目前已有关于元数据管理、资源管理、用户管理、资源访问、国际化、用户隐私、系统运营和管理等数十个兴趣小组和三个区域兴趣小组。其社区讨论内容、视频和记录均以公开的形式发布到社区 wiki 中,代码则通过流行的 github 进行开源管理,任何人都可以在其基础上创建软件版本分支,修改代码,贡献代码。自 2016 年启动以来,已有包括图书馆、供应商和开发人员等 1 000 多名会员,共同参与系统的设计、开发、测试和实施。目前已进入第三阶段,预计至少有三家图书馆将应用 FOLIO 系统。2019 年第三季度,FOLIO 已在瑞典查尔莫斯大学图书馆上线^[7],并进行稳定性、性能和系统缺陷方面的测试。

如图 1 所示,FOLIO 平台架构由四层组成:最底层为系统层,负责数据存储、索引、日志管理,目前 FOLIO 支持关系型数据库 PostgreSQL 和非关系型数据库 MongoDB;第二层为消息网关层(也被称为 Okapi),为整个 FOLIO 平台的核心,其实质是一个基于代理服务的 API 网关,负责各子系统或模块之间的消息转发、权限控制、监控、负载均衡、缓存、请求整形和管理以及静态相应处理,通过 Okapi 可以实现系统的可配置、可扩展,由 Vert.x 实现;第三层为应用层,该层主要负责具体的业务实现,如电子资源管理、元数据管理、流通等,作为一个开源软件,FOLIO 探索的类似“图书馆系统应用商店”的商业模式也在此层实现;最后一层为 UI 层,为开发人员提供 UI 界面开发工具。

1.2.1 Vert.x

Vert.x 是一个非阻塞式 IO 模型,相较于阻塞式 IO 具有高效的请求处理能力,FOLIO 选择 Vert.x 实现其 Okapi(API 网管),主要目的是为了确保 Okapi 的高效性。但由于 Vert.x 是一个刚刚兴起的基于异步事件的开源框架,相较于 Spring 框架而言缺少丰富的整合插件以及熟悉

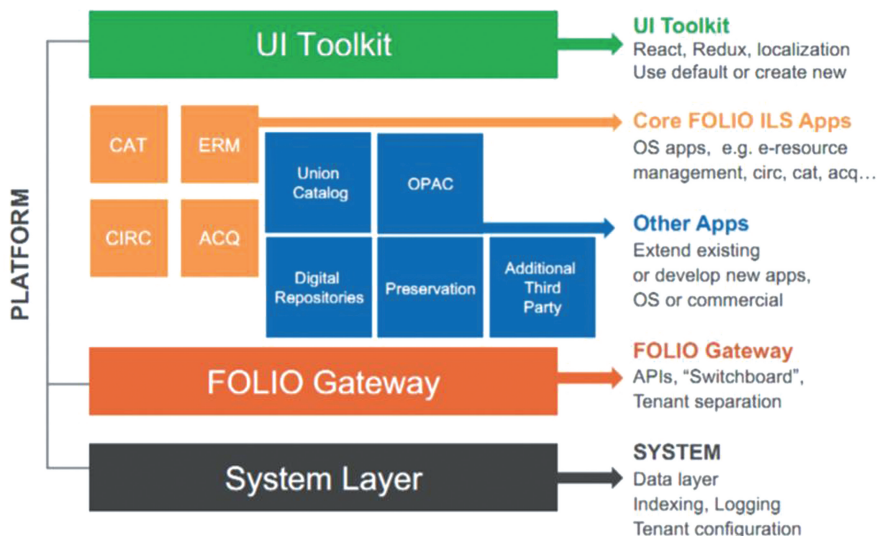


图 1 FOLIO 平台架构

的软件开发人员。考虑到在实际应用过程中,将 Vert.x 和 Spring 框架整合在一起是一个比较好的选择,既可以拥有 Vert.x 的高吞吐量,又可以使用 Spring 丰富的扩展库。所以笔者对 Vert.x JDBC 和 Vert.x 整合 Spring JPA 做了一个相应的性能测试。

如表 1 所示,测试有两个方案:方案一名称为“Spring”,采用 Vert.x-web 整合 Spring JPA,通过 HTTP PUT 对数据库进行写操作。方案二名称为“Vert.x”,采用 Vert.x-web 和 Vert.x-jdbc、

HTTP PUT 对数据库进行写操作,两个方案均采用默认配置,并没有进行额外的优化。测试环境为 Win10 操作系统,jdk1.8 64 位,cpu 为 i7-7700,机械硬盘,测试程序可于 github (<https://github.com/x19990416/vertx-examples>) 上获取。

利用 Jmeter 对不同的方案进行并发测试。总体来说,并发数在 600 以内时 Vert.x 的性能优于 Spring;从 700 开始进入数据库的 IO 瓶颈,出现大量的数据库写错误,由于 Spring 缓存机制

表 1 Spring 和 Vert.x 在不同并发数下的性能比较

并发数	方案	样本	异常率	均值 (ms)	标准差	中位数
500	Spring	1 000	0	18 437	7 664.75	24 095
	Vert.x		0	15 024	6 398.91	19 301
600	Spring	1 200	0	20 010	8 376.48	26 026
	Vert.x		0	18 451	8 099.39	24 323
700	Spring	1 400	50.36%	23 817	9 054.90	30 002
	Vert.x		56.71%	22 540	9 657.25	30 001
800	Spring	1 600	60.81%	24 204	9 018.05	30 002
	Vert.x		61.06%	24 150	9 069.90	30 001
1 000	Spring	2 000	68.80%	26 003	7 916.3	30 003
	Vert.x		70.20%	25 366	8 399.88	30 002
10 000	Spring	20 000	96.99%	23 724	12 384.83	30 281
	Vert.x		97.01%	22 321	13 255.41	30 107

使得其写数据库的错误率略低于 Vert. x。

结合以上性能测试和 Spring 框架在国内市场的实际情况,笔者认为整合 Vert. x 与 Spring 对图书馆行业应用领域来讲是一个比较合适的选择,既可以充分发挥 Vert. x 的高吞吐量,实现与 okapi 核心框架无缝对接,又可以使用 Spring 丰富的扩展包。

1.2.2 PostgreSQL

据 DB-ENGINES^[8]最新的数据库排名所示,排名前四的关系型数据库分别为 Oracle、MySQL、SQL Server 和 PostgreSQL,其中 MySQL 和 PostgreSQL 均为开源数据库,MySQL 的综合排名要高于 PostgreSQL,那么为什么 FOLIO 要选择 PostgreSQL 呢?笔者认为这与数据库对 JSON 格式的支持程度有关。

FOLIO 的目的是构建可持续的现代图书馆技术生态系统,如何更好地支持将来可能发生的需求是 FOLIO 不可避免的问题。就数据库底层设计而言,是无法预知将来数据之间的组织关系的,故关系型数据库这种行列明确的数据存储形式不能很好地满足这方面的需求。以现在图书馆用户运用微信、微博等第三方认证为例,早期的用户名和密码认证方式存储的字段结构并不能满足第三方认证的需求,故图书馆业务需求需要依照第三方认证的需要修改表结构,以增加相应的字段。

为了解决这个问题,FOLIO 做了两种方案,其一是支持非关系型数据库 MongoDB(数据以 JSON 格式存储),其二是采用关系型数据库,但其内部数据采用 JSON 格式存储,那么在对 JSON 格式的支持方面,提供专属 JSON 字段(JSONB 数据格式)的 PostgreSQL 数据库要明显优于 MySQL(支持 JSON 函数查询)。

2 FOLIO 的运营模式——开放共赢

一个可持续发展的开源项目首先必须要建设一个良好的生态环境,不仅要吸引开发者参与其中,还需要商业公司提供项目的产品化应

用。开发者专注技术,负责产品实现,商业公司关注市场,提供产品化意见,双方共同合作来不断完善项目。

2.1 社区驱动

FOLIO 作为一个社区驱动的开源项目已有一整套完善的体系架构,由技术委员会(Tech Council, TC)与产品委员会(Product Council, PC)组成。项目中的一系列需求由技术委员会与产品委员会共同完成,如共同指导开源社区以保证技术目标和方向的一致;解决平台长期愿景、连续性和稳定性的技术问题;主动识别需要处理的技术问题和资源需求;协调项目中的各方利益问题。

(1)技术委员会:采取准入制,其委员任命由 FOLIO 利益相关方批准,确保各方在委员会中至少有一个席位。技术委员会以协商方式做出决定,在无法达成共识的情况下由委员会成员投票决定。其职责范围包括:明确总体技术方向;定义项目贡献代码的流程和标准;协调开发者与产品使用者之间的关系以减少冲突(如产品设计理念间的冲突);向产品委员会提出需要努力和确定优先顺序的技术问题的建议;回应产品委员会关于技术问题的要求。

(2)产品委员会:以 OLE 成员为主,由利益相关方任命。虽然每个成员代表一个利益相关者,但产品委员会的目标是进一步实现 FOLIO 项目的总体目标。其职责包括:确定路线图中的优先事项,评估影响路线图的战略发展;协调 FOLIO 社区发展合作伙伴的工作;协调并鼓励社区参与 FOLIO;协调兴趣小组意见和工作,确保兴趣小组围绕共同感兴趣的对话、协商和设计。

(3)特殊兴趣小组:由产品委员会正式批准成立,并为具有相似技术需求、地理区域或热门兴趣的 FOLIO 参与者提供论坛。通过创建文档,对 FOLIO 模块设计原型的反应以及代码片段的理解来对 FOLIO 功能形成共识。它们允许用户交流想法,为 FOLIO 项目制定共同目标,并使用广泛使用的软件研究和开发新的交换标准。目前有元数据管理、资源管理、用户管理等

11 个特殊兴趣小组。

FOLIO 的社区驱动模式为利益各方提供了相互协商的平台,也使图书馆从一个被动的图书馆系统产品消费者转型成为产品设计者,参与到图书馆服务平台的设计中;另一方面,图书馆消费终端的建议也使得系统能够更好地为图书馆业务服务,进而提升图书馆整体服务创新的能力。有了多方的参与以及公开环境下的协商机制,FOLIO 能够更好地完成现代技术生态系统构建。

2.2 商业模式

现今而言,开源更多的是一种商业决策。FOLIO 也是 EBSCO 公司所做的一个商业决策,EBSCO 不仅资助开放图书馆基金会,同时还作为开发商和供应商为社区提供技术支撑,这是一个典型的商业参与开源社区的项目模式。图书馆员作为主题专家贡献他们的专业知识;开发商根据主题专家的意见创建平台及其模块;供应商将为项目提供实施、托管和支持服务。为了维护良好的社区生态,吸引更多的开发者参与其中,合适的商业模式是保证社区良好运作的必要条件之一,笔者认为 FOLIO 可能会通过如下两种模式进行商业运营。

(1) 开放核心,扩展收费:希望用户能够购买其增值产品,通常涉及一个强大的免费并开源的核心产品。在核心产品周围,商业实体提供增加或扩展其功能的专有软件,这些附加组件作为商业软件出售,与支持和服务捆绑销售,如 Confluent、Cloudera、Elastic。这种模式很符合 FOLIO 现有的架构,尤其从其提供的“微服务应用商店”的功能来看,大多会采用此种模式。

(2) 云/托管:FOLIO 的 SaaS 特性决定了其天生就和托管模式结合在一起。用户可以自行部署和管理,也能够使用别的厂商开发的模块或者工具套件中的源代码。若客户想实现一个真正的多租户环境的话,就必须使用托管的平台。此种方式可以产生明显的效益,而且具有可持续发展模式。此种模式对于技术能力不强、数据量不是很大的图书馆而言非常有用。

通过云/托管,结合应用商店的个性化选配组件,可以定制出专属于图书馆自身的业务系统。但现在这种模式最大的挑战来自于如何实施数据保护、保证正常运行时间和系统安全等。FOLIO 若要实现此种模式,还需要加强其系统安全方面的设计和验证。

3 国内如何落地 FOLIO

国内 FOLIO 支持者包括上海图书馆、上海交通大学图书馆、CALIS 等,同时吸引了国内一些图书馆系统相关的商业公司参与其中。目前,上海图书馆已组织志愿者团队对 FOLIO 界面进行了初步汉化。CALIS 也将自身的图书馆管理系统的相应功能作为 FOLIO 的模块接入其中,构建了 CLSP 并尝试部署到阿里云上。但 FOLIO 在国内落地依然面临很多问题,如:国内外图书馆业务流程不同,使得 FOLIO 的开源模块并不能做到开箱即用,那么必然会涉及 FOLIO 相关的技术普及和推广;FOLIO 采用的技术标准是否适应国内开发环境;国内的图书馆应该如何参与到 FOLIO 社区等。

3.1 技术推广与普及

如前文所述,FOLIO 是基于小众的 Vert. x 实现的,与国内广泛使用的 Spring 全技术栈相比,使用并熟悉其框架的技术人员非常少,CALIS 作为 FOLIO 国内的先行者,也仅仅使用了 FOLIO 的 Okapi 作为微服务的发现注册框架,其余的诸如用户认证、采访等功能模块都采用了自己原有的技术栈,前端则完全抛弃了 FOLIO 的 UI 框架,这虽然也是一种技术方案,但在本质上相当于抛弃了 FOLIO 的整体架构,无法与社区的技术标准同步。

中文社区是开源项目推广和普及最常用的方式,笔者认为可依托图书馆构建 FOLIO 的中文开源社区进行推广,国内图书馆员在作为主题专家为 FOLIO 官方社区提供国内的专业知识、保持与社区紧密联系的同时,可以将官方社区的内容包括实施经验等及时进行翻译整理提供到中文

社区,提供一个国内交流沟通的平台;图书馆也可以在社区中寻找活跃的开发员作为开发伙伴。

3.2 图书馆的角色转变

长久以来,图书馆一直以项目外包的形式进行IT系统的开发,即图书馆作为甲方提出具体需求,通过项目外包方式由乙方承接并负责实现,最后由图书馆技术部门负责验收和后期的运维。在这个过程中,由于并没有相应的外包开发标准和规范,这并不利于IT项目的移植和技术积累。然而,对于FOLIO这种“微服务商店”的模式而言,一套完整的开发设计规范是不可或缺的。所以,图书馆可以借此契机制定外包开发的标准,实现从需求、实现、测试、验收和运维的标准化,并结合checkstyle、findbugs、jenkins、docker等流行的IT技术手段,构建持续集成环境。

图书馆作为FOLIO的消费终端,也需要积

极探索FOLIO在国内图书馆的应用实践,主动参与到FOLIO项目的研发过程中,加强产品意识,从之前的项目验收方和使用方转变成为项目的拥有者,积极参与到项目的整个流程中,贯穿需求、设计、开发、测试和验收的各个阶段。

4 结语

FOLIO作为一个由商业公司支持的图书馆系统开源项目,使得图书馆能够从以往的图书馆系统的使用者转变为图书馆系统的参与者和拥有者,打破了以往图书馆对于图书馆系统被动接受的局面。无论FOLIO之后会如何,其开放的理念必然会逐步渗入到图书馆中,图书馆行业中必然也会有越来越多的开源项目。对于图书馆而言,更多的是要做好观念的转变,积极参与到开源的大潮当中,未来的图书馆必然是开放的。

参考文献

- [1] 郭利敏. 基于Solr的IPAC书目信息系统整合[J]. 微型电脑应用, 2013, 29(3): 4-7. (Guo Limin. Based on Solr IPAC bibliographic information system integration devices [J]. Microcomputer Applications, 2013, 29(3): 4-7.)
- [2] Breeding M. Automation marketplace 2012: agents of change[EB/OL]. [2019-06-18]. <https://librarytechnology.org/document/16717>.
- [3] Breeding M. Library services platforms: an update of the current environment[EB/OL]. [2019-06-08]. <https://librarytechnology.org/document/20269>.
- [4] 陈武, 王平, 周虹. 下一代图书馆服务平台初探[J]. 大学图书馆学报, 2013(6): 82-87. (Chen Wu, Wang Ping, Zhou Hong. Comparative study on next generation library services platforms [J]. Journal of Academic Libraries, 2013(6): 82-87.)
- [5] 包凌, 赵以安. 国外下一代图书馆自动化系统的实践与发展趋势研究[J]. 图书馆学研究, 2013(9): 58-65. (Bao Ling, Zhao Yi'an. A study on practice and trends of the next generation integrated library systems [J]. Research on Library Science, 2013(9): 58-65.)
- [6] Lewis J, Fowler M. Microservices[EB/OL]. [2019-06-18]. <https://www.martinfowler.com/articles/microservices.html>.
- [7] FOLIO roadmap [EB/OL]. [2019-06-20]. https://wiki.folio.org/display/pc/folio+roadmap?preview=/141053_8/25724741/milestones.png#roadmap-deliverables.
- [8] DB-Engines ranking[EB/OL]. [2019-06-20]. <https://db-engines.com/en/ranking>.

郭利敏 上海图书馆系统网络中心工程师。上海 200031。

张磊 上海图书馆系统网络中心正高级工程师。上海 200031。

(收稿日期:2019-09-11;修回日期:2019-11-13)