



基于微服务架构的 FOLIO 模块建设探索、实践与思考

——以北京大学图书馆闭架图书叫号系统为例

□王昊贤* 周义刚 张乃帅 张娟

摘要 在传统图书馆集成管理系统转型换代之际,“老系统”和“新服务”如何衔接以满足图书馆发展需求和提供读者服务成为图书馆建设的技术难点。文章从微服务与 FOLIO 架构作为切入点,以北京大学图书馆闭架图书叫号系统为例进行探索,介绍了基于微服务架构的 FOLIO 模块实践的设计思路、系统架构和技术使用,并通过分析系统应用数据,提出了优化图书馆服务的对策和建议;从微服务架构技术优势、应用场景和微服务架构下的图书馆角色转变等角度进行了讨论,总结了微服务技术在图书馆管理和服务中的创新意义,可为其他图书馆提供参考借鉴。

关键词 FOLIO 微服务 北京大学图书馆 叫号系统

分类号 G252

DOI 10.16603/j.issn1002-1027.2020.04.004

1 引言

无所不在的网络、无所不在的计算、无所不在的数据、无所不在的知识和无所不在的想象已成为当下大学生的主要生活方式和工作模式。随着作为互联网世代的“90后”和“00后”逐渐成为图书馆的主要用户群体,用户的需求已不再局限于图书馆的传统服务,而更趋向于自动化和智能化,这对图书馆的工作提出了新要求,传统的图书馆工作流程和服务方式与新时代的用户需求之间产生了明显的代沟。

信息科学技术的发展使得传统的图书馆集成管理系统(ILS)在资源管理和业务流程管理上受到了冲击,“新一代图书馆服务平台”成为学界的讨论热点。早在2013年殷红、刘炜就指出原有的“图书馆集成管理系统”已完全丧失了“集成”性^[1]。面对这一困境,肖铮、林俊伟认为新一代图书馆服务平台应具有:平台化、开放性、易扩展性和云端化等特征^[2]。陈武等认为下一代图书馆服务平台应充分利用最新的云计算技术和架构的优点,以降低软硬件维护成本^[3]。吴建中认为,下一代图书馆技术要把体现增值功能、突出用户本位、适应事业发展放在重要位置,同时要加强系统的扩展性、安全性和个性化^[4]。周义刚、聂华结合北京大学图书馆的实际需求,提出

了北京大学图书馆的需求框架^[5]。如何开发能满足时代要求和用户需求的平台,邵波、张文竹结合南京大学图书馆系统平台的实践讨论下一代图书馆平台的难点和对策^[6];刘素清从用户体验视角进行研究,分析得出开源项目 FOLIO 的推出为图书馆服务平台的选择提供了新的思路,其核心目标是建设一个开源的、基于云服务的应用程序框架,以实现不同开发团队之间的协作^[7]。2018年,中国高等教育文献保障系统(CALIS)新一代图书馆服务平台联盟成立,为基于 FOLIO 的新一代图书馆服务平台建设打下了坚实的基础^[8]。结合读者需求,抓住 FOLIO 发展、探索与应用的契机,2018年,北京大学图书馆运用基于 FOLIO 的微服务框架进行了闭架图书叫号系统开发,实现了通过开源程序来管理馆藏资源和提升图书馆服务质量的目标。同年,CALIS 也运用 FOLIO 架构升级了 CALIS 馆际互借系统。

2 微服务架构与 FOLIO

2.1 单体架构、SOA 与微服务架构

单体架构应用是功能集中、代码和数据中心化、一个发布包、部署后运行在同一进程的应用程序。面向服务的架构(Service Oriented Architecture,

* 通讯作者:王昊贤,ORCID:000-0003-3344-3089,邮箱:wanghx@lib.pku.edu.cn。



SOA)是一个组件模型,它将应用程序的不同功能单元进行拆分,并通过这些业务系统之间定义良好的接口和契约联系起来^[9]。微服务(Micro-services)架构 SOA 有相同之处,在 SOA 之上升华,是一些具有足够小的粒度、能够相互协作且自治的服务体系,每个服务仅关注于完成一个具体业务并能够良好地完成该功能^[10-12]。微服务架构强调的是业务彻底的组件化和“微”服务化,原有的单个业务系统会拆分为多个可以独立开发、设计、运行的微服务,这些微服务之间通过 API 完成交互。

单体架构、SOA 与微服务架构如图 1 所示,SOA 的总线是一根管道,用来连接各业务应用,为了集成不同的子系统、不同的协议服务,总线做了消息的转解释和路由工作,让不同的服务互联互通。微服务中的 API 网关是一个独立的服务,是系统的唯一入口,所有的客户端和消息端都通过统一的网关接入微服务。不同的微服务间也是通过 API 进行轻量级通讯。单体架构、SOA 与微服务架构在组件大小、耦合程度、团队架构等方面的对比如表 1 所示。

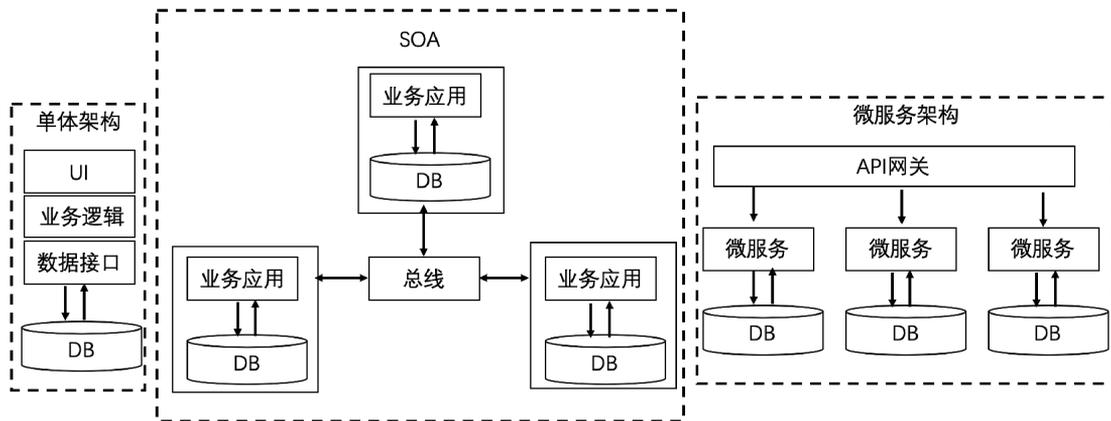


图 1 单体架构、SOA、微服务架构示意图

表 1 单体架构、SOA、微服务架构对比

功能	单体架构	SOA	微服务架构
组件大小	单组件	由多个子系统组成,组件较大	由多个微服务组成、组件较小
耦合	高度耦合	通常松耦合	总是松耦合
团队架构	——	一般为大型企业	一般为小型团队
管理	中央管理	中央管理,自顶向下开展实施	分散管理,自底向上开展实施
目标	完整功能	面向服务、功能	快速开发迭代

微服务的特性使其在系统设计开发应用中具有一定的优势,并且越是在规模庞大的软件项目中其优势越明显^[13]。优势包括:

(1)独立性。不同的服务间耦合程度低,有明确的任务和功能。

(2)技术选择灵活。服务间通过 API 进行通讯,开发团队可根据自身情况,选择适合的架构和开发语言。

(3)独立部署。更便捷的借助容器(如 Docker)独立部署,且进程隔离^[14]。

微服务架构自产生就被业界广泛应用,例如,新型云件 PaaS 平台^[15]、统一应用开发平台^[16]、电力云服务平台^[17]、城市一卡通手机充值支撑系统等^[18]。微服务架构也为新一代图书馆服务平台的建设提供了新的思路。与此同时,在现有图书馆集成管理系统退出历史舞台到新一代图书馆服务平台开始服务的过渡时期,可以基于微服务架构,将图书馆的新增服务统一设计、组织、管理。

2.2 FOLIO 微服务架构

FOLIO(The Future of Libraries is Open)项目成立于 2016 年,由图书馆、开发者和提供方共同发起,并于 2019 年发布 Aster 版本^[19]。FOLIO 平台(见图 2)采用微服务架构^[20],数据层支持 PostgreSQL、Oracle、MongoDB 等多种数据存储技术;系统层主要负责索引、日志、租户配置等;OKAPI 是 FOLIO 平台的网关,负责租户和 APPs 间的通信;应用层为应用程序,包括采访、编目、流通等传统模块,以及支持扩展模块;UI 层可以运用 React、VUE、Angular 等技术开发用户界面,以适应不同用户在不同场景下使用的需求。其中 OKAPI 是 FO-

2020 年 第 5 期
大學圖書館學報

LIO 的核心,各业务功能被划分成相互独立的微服务,将 RESTful API 接口信息以及代理或托管信息提交到 OKAPI。用户访问 OKAPI HTTP 服务,OKAPI 通过接口 URL 确定用户需要调用的 API,然后通过代理或托管信息找到实际服务的地址,将请求转发到实际服务的地址。

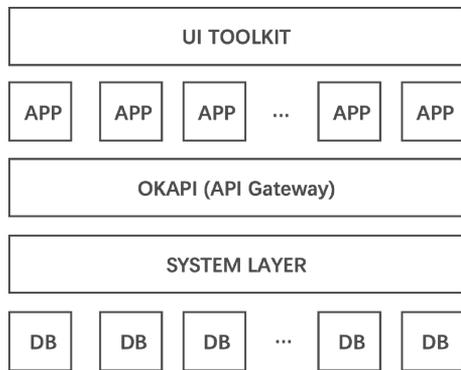


图 2 FOLIO 平台架构

完整的 FOLIO-APP 的基本构成(见图 3)包括 UI 模块、后端模块和存储模块,UI 模块负责前端页面交互,后端模块负责业务逻辑,存储模块负责实现特定的持久化措施。服务型 FOLIO-APP 主要包含后端业务模块和存储模块,业务逻辑型 FOLIO-APP 主要包含后端业务模块,前端型 FOLIO-APP 主要包含 UI 模块。



图 3 FOLIO-APP 的结构

3 FOLIO 在北京大学图书馆的实践

北京大学图书馆闭架借书处一直使用读者递交索书单,工作人员按条找书的传统服务方式,整个工作流程中存在着诸多困难。例如,有时读者手写不清、誊写信息错误,导致工作人员分单取书困难;由于空间有限,部分外文图书同一分类分放在不同的楼层,找书要依靠读者查询填写馆藏位置,如果读者不清楚借阅规则,则不得不进行二次检索,增加了借书难度;馆藏查询与借书时间不同步,读者填写索书单时图书可能已经是借出状态;读者借阅的是非闭架图书,却误到闭架借书处提交索书单,导致借书请求无法满足等。针对这一系列问题,为了将闭架借书服务从线下手工作业流程提升到线上系统操作,北京大学图书馆开发了闭架图书叫号系统(下文简称叫号系统),是基于微服务架构的 FOLIO 模块建设的一次有意义的探索和实践。

3.1 以用户为中心的设计

以用户为中心的设计(User-Centered Design),其核心思想就是在产品开发的每个阶段都把用户特征纳入考虑的范围。系统设计的初衷在于为读者提供更便捷的借书方式、缩短获取馆藏资源的时间,同时减轻工作人员的工作量,并直观分析用户行为,为完善工作流程,进一步提升工作效率提供依据。叫号系统对现有流程做出了优化调整,用户与系统交互的流程如图 4 所示。优化后,读者无需手工填写索书单,查询与提交请求一次完成,节省时间和精力,并省去了图书馆工作人员分单和索书单由运书电梯上行的环节,简化了借书流程,打印出来的索书单和索书号清晰准确,图书定位准确,可借状态准确,借书效率更高。

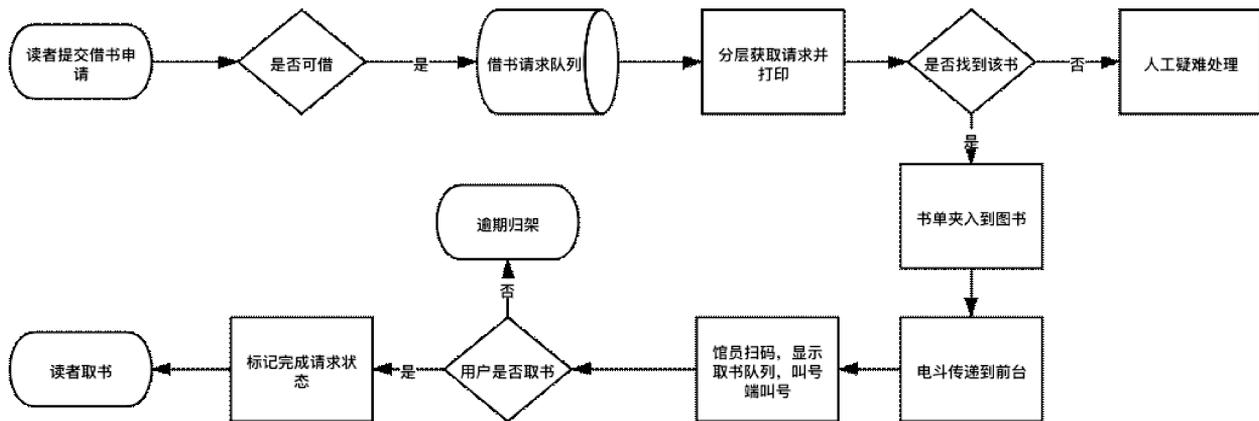


图 4 用户与叫号系统交互流程图



3.2 叫号系统架构与主要开发技术

在叫号系统的业务分析中,对叫号业务涉及到的统一认证、馆藏检索、叫号业务处理以及 FOLIO 的一系列基础服务等环节进行相应的微服务的分解,不同微服务之间的关系是低度耦合的,单项微服务只负责单一任务,不同微服务之间通过 RESTful 请求通讯,在保证 RESTful API 稳定的情况下,各模块升级互不影响。其中,统一认证服务主要负责用户的认证、身份信息、权限获取,并通过发送 RESTful 请求的方式与 FOLIO 基础服务中的用户模块进行通讯;馆藏检索服务主要负责与图书馆集成系统 Symphony 的通信、检索以及检索结果的处理;叫号业务处理服务主要负责读者请求的建立以及处理等;FOLIO 的基础服务包括模块认证、权限管理、用户管理等。叫号服务在 FOLIO 平台中的角色为应用层众多应用(APP)中的一个应用,叫号服务与 OKAPI 的关系如图 5 所示。

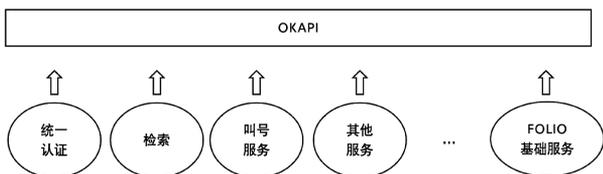


图 5 叫号服务与 OKAPI 的关系

FOLIO 的系统架构支持了前后端分离的开发方式,后端及存储模块负责业务逻辑,而前端模块主要负责呈现、用户交互等。从技术选型角度,FOLIO 支持了多语言、多框架的技术,具有很高的灵活性、适应性、兼容性和可扩展性,不同模块间互不影响,不同的开发团队可以根据其技术栈选择不同的技术。各技术和框架各有优势,在遵循 FOLIO OKAPI 支持的 RESTful 通讯模式,开发团队可选择适应业务流程所需的、团队擅长的语言和技术框架,而不仅仅局限于 FOLIO 已经使用的技术框架,例如 FOLIO OKAPI 运用 Vert.x 框架。FOLIO—APP 结构与叫号系统采用的主要技术如图 6 所示。

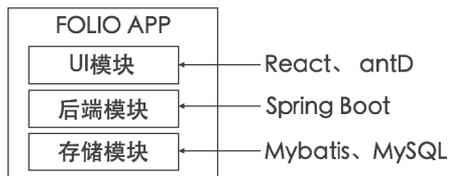


图 6 FOLIO—APP 结构与叫号系统采用的主要技术

后端模块支持多语言、多种技术开发,例如 Spring Boot、Nodejs、Django 等,具有很强的灵活性,适应不同技术团队的开发。叫号系统后端模块运用 RESTful 开发模式,采用 Spring Boot 技术将业务逻辑包装成 RESTful API 接口,与其他模块进行 RESTful 模式通讯,叫号系统模块的修改和升级不影响其他模块,其他模块在保证接口稳定的情况下的升级和扩展同样不影响叫号系统模块,保证了平台的兼容性和可扩展性。Spring Boot 是由 Pivotal 团队提供的框架,其设计目的是用来简化新 Spring 应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置,从而使开发人员不再需要定义样板化的配置。与 OKAPI 的通讯采用代理模式,即 OKAPI 不负责模块的运行,仅将请求转发给指定的 URL,后端业务模块将 API 接口信息和代理信息提交给 OKAPI。存储模块采用 Mybatis 和 MySQL 数据库,MyBatis 基于 Java 语言的持久层框架,支持定制化 SQL、存储过程以及高级映射,使用简单的 XML 或注解来配置和映射原生信息,将接口和 Java 的 POJOs(Plain Old Java Objects)映射成数据库中的记录。

前端模块采用 React、Ant Design 技术实现多终端的用户交互界面。React 起源于 Facebook 的内部项目,由起始的 JavaScript MVC 框架扩展成一套前后端 Web APP 解决方案,主要用于构建 UI,可适用于服务器、浏览器和移动终端,与 Angular、VUE 并称为目前流行的三大开源 UI 框架。React 的核心思想是封装组件,各个组件维护自己的状态和 UI,当状态变更,自动重新渲染整个组件。在叫号系统前端模块中,发挥前端控制层作用,并提供了前后端数据通讯的通道。Ant Design 是蚂蚁金服体验技术部出品的基于 React 的 UI 库,包含设计原则、控件规范和视觉尺寸以及配套的前端代码实现方案等,满足了多用户适应不同终端的需求。

3.3 部署上线服务

叫号系统采用了前后端分离的开发方式,也支持了前后端分开打包部署的方式,前后端模块可部署在相同或不同的服务上。叫号系统后端模块运用 Maven 打包方式,可直接发布为可运行 Jar 包。前端模块运用 Webpack 打包,生成若干个 HTML、JS、CSS 以及图片等静态文件。Nginx 是一个轻量级的高性能的 HTTP 和反向代理 Web 服务器,具

有支持高并发链接、占用内存较小、配置文件简洁、支持热部署等优势。因此,将前端模块部署到 Nginx 服务器,同时,为保障用户在使用系统服务时的个人信息安全,运用 Nginx 服务器反向代理的方式启用 HTTPS 服务。另外,叫号端采用了桌面虚拟化技术,通过终端设备连接到位于数据中心的虚拟桌面提供持续稳定的服务,保证了叫号端的稳定性。系统的部署采用了先进的技术,不仅使得部署过程便捷简单,并且保证了服务的安全性和稳定性。

4 系统应用数据分析

基于微服务框架开发的闭架图书叫号系统强大的统计功能为提升用户体验、改进管理方式提供了数据支撑。叫号系统于 2018 年 12 月 24 日试运行,试运行当日完成了 91 册图书的借阅请求,自试运行开始至 2019 年 9 月 30 日,共计有 10000 余名读者登录系统,产生了 124000 余条检索(含翻页)数据和 37000 余条图书借阅请求数据。如图 7 所示,96% 的图书借阅请求正常结束,可见读者借到书的成功率较高,但仍有 4% 的请求异常结束,其中 42% 为读者取消借阅申请,图书未找到的占 32%,逾期未取书的占 26%。

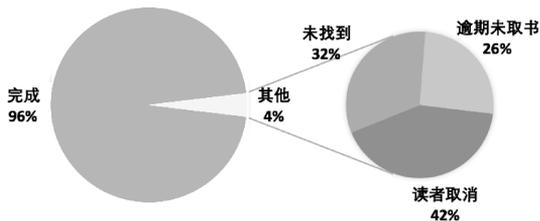


图 7 闭架图书借阅请求完成状态

完成借书流程时间指的是从读者检索提交借书申请到读者拿到书完成整个借书流程的时间差(见图 8)。据统计,一位读者最快 33 秒完成了整个闭架借书流程,1500 余条请求在 1 分钟内完成,半数以上请求在 6 分钟内完成整个借书流程,80% 的请求在 12 分钟内完成。

对读者提交借书请求的时间进行分析,如图 9 所示,10:00—11:00、12:00—13:00 和 15:00—16:00 出现三个峰值,这与学生的课间休息和午餐时间基本相符;8:00—10:00 与其他时段相比,借书请求数相对较少。分析得出的结论可为优化工作流程提供指导,如工作人员可集中在早晨完成其他工作,如调整闭架图书预约报表运行、处理时间和新书收登

等工作,在借书请求峰值时段以服务现场借书读者为主,减少读者等待时间。

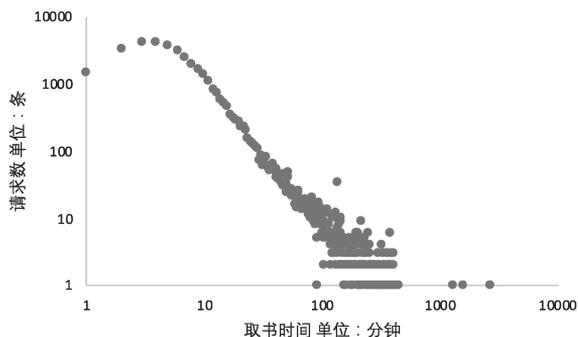


图 8 闭架图书完成借书流程时长

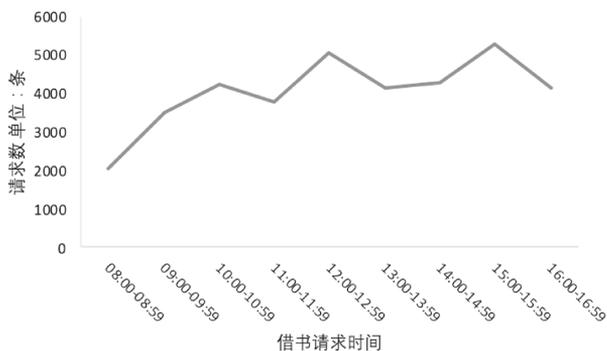


图 9 借书请求数量随时间的变化

按读者所借图书的分类号进行统计分析,排名前 10 位的图书类别如图 10 所示,包括 D(政治、法律)、T(工业技术)、I(文学)、B(哲学、宗教)、K(历史、地理)、O(数理科学和化学)、F(经济)、H(语言、文字)、G(文化、科学、教育、体育)、C(社会科学总论)。这 10 类的借书请求数量占总请求数的 87%,基本符合齐普夫定律,这一统计可为未来图书馆空间布局的调整和书库排架设计提供参考。

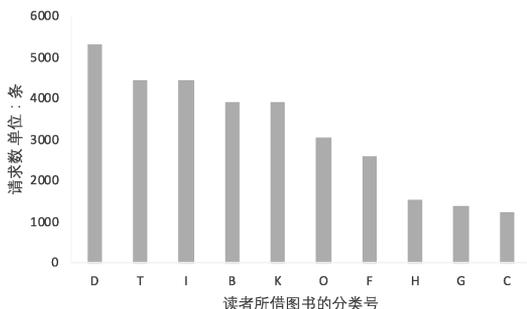


图 10 借书请求数量排名前 10 位的图书类别

5 微服务架构应用探索与思考

从叫号系统的技术原理和实践过程中可见,微



服务架构在当前图书馆转型时期是一个较好的选择。因其既可以针对一个具体的业务独立开发和运用,也可以在同一开发模式下实现兼容和合作,既能满足传统图书馆管理系统的服务升级要求,又能匹配新一代图书馆服务平台的多种要求。

5.1 微服务技术优势

(1)单一职责。FOLIO 采用微服务架构是一种高度模块化的设计,每个模块只负责与本模块相关的业务,当一个模块进行升级和改造时,其他模块的运行不受影响。例如,北京大学图书馆的统一认证采用了 CALIS 统一认证系统,同时,北京大学校内也支持 IAAA 的认证,如果认证方式进行了调整和修改,仅需对认证模块进行改造,不影响其他模块的运行,从理论上实现了热插拔。

(2)轻量级通信。如上文所述 OKAPI 是 FOLIO 的核心,统一认证、检索、叫号服务与 FOLIO 的一系列基础服务分别作为独立的“微服务”,通过托管或代理的模式将 API 接口提交给 OKAPI,OKAPI 对不同服务的 API 接口进行了统一管理。其他服务,例如送书上门服务也可包装成一个独立的微服务,通过相似的方式将 API 接口提交给 OKAPI,对服务平台进行扩展,运用 RESTful 的接口方式可实现不同模块、不同微服务之间的通讯。

(3)技术多样性。OKAPI 通过认证租户的方式,为各模块、微服务提供模块级别的权限控制。不同的微服务负责相对独立的业务,逻辑与其他微服务关系是低度耦合的,可由不同的开发团队运用不同的编程语言、不同的技术框架,并可独立部署,搭建在不同的物理或虚拟服务器上,形成服务云,仅需遵循 RESTful 的开发方式即可实现不同服务间的通讯。例如,FOLIO OKAPI 运用 Vert.x 框架,叫号系统后端使用了 Spring Boot 框架,两个模块分别由不同团队开发,选择了不同的技术,部署在不同的虚拟服务器上,但配合良好,体现出微服务架构的优势。

(4)独立性。FOLIO 的系统架构支持了前后端分离的开发方式,这种开发方式不同于传统前后端作为一个统一的整体开发方式,前端和后端开发相对独立,在开发前期对接口进行描述,前端和后端开发可同时进行,互不干扰。与此同时,后端模块将 API 接口提交给 OKAPI,API 接口由 OKAPI 统一管理,前端请求发送给 OKAPI,由 OKAPI 做请

求的统一分发而不必考虑服务的物理、虚拟服务器的具体位置、IP 等,因此,前端模块在发送请求时对后端是否为同一微服务不敏感,对不同微服务提交的 API 接口进行请求发送采用的方式基本相似,这样提高了前端开发的灵活性,降低了前后端的耦合度,为开发提供了便利。综合对系统结构的分析和微服务的拆分与开发应用,可以看出基于微服务架构的 FOLIO 模块建设中充分体现新一代图书馆服务平台的平台化、开放性、易扩展性、云端化等特征。

5.2 微服务应用场景

虽然北京师范大学图书馆、清华大学图书馆分别于 2017 年、2018 年上线了下一代图书馆服务平台,但大部分高校图书馆仍处在“老系统”向“新平台”的过渡时期,且在短时间内无法完成新老系统的更替。与此同时,随着互联网时代的到来,用户的需求出现爆发式增长,图书馆因原有系统存在的一系列问题,在面对新需求时显得捉襟见肘,在提供新服务的同时,对旧有系统的生命周期会有所顾虑。“老系统”一旦终止服役,新的服务是否会受到影响?影响程度如何?“老系统”终止服役的正确时机又在何时?

微服务架构的设计有可能最大程度地解决这一问题,将新的服务与旧有系统通信的部分包装成为独立的服务,进行数据获取、业务处理等,并向新服务提供 RESTful API,当新老平台进行切换时,运用同样的 API 接口,仅需对此服务进行切换,对新服务影响不大。例如叫号系统中,图书检索模块运用了基于 Symphony Web Service 的基础检索功能,但用户对检索结果的满意度不高,当图书馆替换集成系统 Symphony 或升级改造检索功能时,仅需对检索模块进行独立的升级即可。

同样,“老系统”中部分模块已经不能适应现有的业务逻辑,但其他模块仍可继续稳定服役,同样可以重新组织业务逻辑,并将与“老系统”的交互部分进行服务化,从而把业务从“老系统”中替换。依照这一思路,可以逐步分割“老系统”的功能,一方面适应新的业务需求,另一方面,“老系统”将会变得越来越“瘦”,实现“老系统”向“新平台”的稳步过渡。即使新一代图书馆服务平台选择了商业产品,对于扩展的新服务同样可以采用微服务的架构进行扩展,运用 FOLIO 的方式方法将这些服务进行统一管理。



基于微服务架构的 FOLIO 模式,不仅适用于新一代图书馆平台,同样可以管理“周边”服务。图书馆并非只有图书馆集成系统或新一代图书馆服务平台,还有一系列的“周边”服务系统,少则几个,多则几百个,这些系统由不同的厂商或图书馆自身进行开发,采用不同的技术架构,再加上有些服务的开发商已经不存在,或者当初负责开发的图书馆员存在调离等情况,使得后续的系统运行和维护困难重重。而基于微服务架构的 FOLIO 模式,不仅适用于新一代图书馆平台,同样可以把这些“周边”服务进行统一管理和运行维护。服务可以由不同的团队采用不同的技术进行开发,仅需遵循 RESTful 的通用开发模式,将接口交于 OKAPI 统一管理。

5.3 图书馆角色的转变

图书馆集成管理系统提供的服务是图书馆传统服务的核心,包括采访、编目、流通等业务,并且业务流程与系统流程是高度耦合的。然而,大多数商业产品的系统流程,特别是境外公司的图书馆集成管理系统,与国内图书馆的业务流程或多或少都存在差异,尤其是采访模块,用外来的系统流程适应本地业务流程是一件相对困难的事情。

而基于微服务架构的开源平台方式让图书馆从被动的使用方转化为主动的需求方,拥有很高的自主选择权。图书馆可以提出符合自己业务流程的需求,交由开发方进行开发,或者在已有的模块中选择适合自己业务流程的模块。同时,本身具有开发能力的图书馆,也可以进行模块的自主开发,仅需遵循相应的规则,选择适合的开发语言和技术框架。

6 总结与展望

在互联网技术爆炸的背景下,社会正经历着方方面面的重大的、影响深刻的变革,图书馆也正在经历这些变革。变革影响到传统图书馆提供的服务内容、服务方式,促使图书馆自身不断提高对技术的应用。变革也影响到图书馆的用户,用户学习方式和行为习惯的改变要求图书馆提供的服务和新技术的应用要相互渗透和融合。只有进一步构建便于用户检索和利用文献的平台,才能充分利用馆藏资源为广大读者提供便捷的服务。微服务架构、FOLIO 等的出现为解决这一问题提供了新的思路。北京大学图书馆闭架图书叫号系统的设计、开发及上线为图书馆赢得了用户,帮助图书馆树立了新的

形象。

首先,叫号系统针对不同用户需求在设计和实践的过程中对 FOLIO 架构进行了分析和研究,兼顾了现役图书馆集成系统 Symphony、现有工作流程和新一代图书馆服务平台的新理念、新技术。在“老系统”向下一代平台的过渡时期,叫号系统运用 FOLIO 的技术和平台较好地满足了新时代的用户需求,更好地解决了“老系统”不能解决或者极难解决的问题。运用微服务架构技术,新老平台的更替对叫号系统等新建的系统影响不大,从理论和实践上,实现了新老平台的平稳过渡,为其他图书馆解决此类问题提供了思路。

其次,在设计和开发的过程,虽然叫号系统作为一个较小的应用,但打通了基于微服务架构的 FOLIO 模块开发的整个流程,打通了各技术开发示例,实现了各模块之间的轻量级通讯,并且扩展性较好,对实现技术的依赖性较小,微服务部署在不同的物理或虚拟服务器上,同时,不同的微服务由 OKAPI 统一管理,有效解决了各项服务分散运行和维护的问题,证明了 FOLIO 的平台化、开放性、易扩展性、云端化等优势,为新一代图书馆服务平台的设计和建设提供了范例。

叫号系统是图书馆运用 FOLIO 的一次探索,但也存在一些不足,在自动化部署和轻运维方面还未进行讨论和尝试,且叫号系统相对新一代图书馆平台来说是一个相对较为简单的模块,未涉及 FOLIO 处理复杂业务的实践。下一步的工作应主要包括实现系统的自动部署和轻运维工作以及非开架图书的统一管理等,向构建个性化的新一代图书馆服务平台迈进。

参考文献

- 1 殷红,刘炜.新一代图书馆服务系统:功能评价与愿景展望[J].中国图书馆学报,2013,39(5):26-33.
- 2 肖铮,林俊伟.用微服务构架下一代图书馆服务平台——以 FOLIO 为例[J].图书馆杂志,2018,37(11):63-69.
- 3 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,2013,31(6):82-87.
- 4 吴建中.人·技术·价值观——关于下一代图书馆技术的思考[J].图书馆,2019(4):1-4.
- 5 周义刚,聂华.新一代图书馆服务平台调研及思考——基于北京大学图书馆的需求[J].图书馆杂志,2019,38(2):69-78.
- 6 邵波,张文竹.下一代图书馆系统平台的实践与思考[J].图书馆情报工作,2019,63(1):98-104.
- 7 刘素清.基于用户体验视角的图书馆服务平台选择思考[J].图



- 书情报工作,2018,62(14):49-54.
- 8 CALIS.CALIS 新一代图书馆服务平台联盟[EB/OL].[2019-02-20].<http://community.calis.edu.cn/index.html>.
- 9 惠澜,殷环环,顾焯青.“面向未来的图书馆事业和图书馆学发展研究”高端论坛综述[J].大学图书馆学报,2019,37(3):5-12.
- 10 Thónes J. Microservices[J], IEEE Software, 2015, 32(1): 1-116.
- 11 郑天民.微服务设计原理与架构[M].北京:人民邮电出版社,2018:9.
- 12 王磊.微服务架构与实践[M].北京:电子工业出版社,2015:39.
- 13 李贞昊.微服务架构的发展与影响分析[J].信息系统工程,2017(1):154-155.
- 14 邓杰文,曹彩凤.微服务若干关键问题研究[J].五邑大学学报(自然科学版),2016,30(2):49-54.
- 15 郭栋,王伟,曾国荪.一种基于微服务架构的新型云件 PaaS 平台[J].信息网络安全,2015(11):15-20.
- 16 李春阳,刘迪,崔蔚,等.基于微服务架构的统一应用开发平台[J].计算机系统应用,2017,26(4):43-48.
- 17 杨俊伟,纪鑫,胡强新.基于微服务架构的电力云服务平台[J].电力信息与通信技术,2017,15(1):8-12.
- 18 温晓丽,苏浩伟,陈欢,等.基于 SpringBoot 微服务架构的城市一卡通手机充值支撑系统研究[J].电子产品世界,2017,24(10):59-62.
- 19 FOLIO.Open source library services platform[EB/OL].[2020-03-11].<https://www.FOLIO.org/>.
- 20 Murray P.FOLIO Wiki[EB/OL].[2020-03-11]. <https://wiki.FOLIO.org/>.
- 作者单位:北京大学图书馆,北京,100871
收稿日期:2019年10月8日
修回日期:2020年3月12日
(责任编辑:关志英)

Exploration and Practice of FOLIO Module Construction Based on Microservice Architecture

—The Case of the Queue Management System of Closed Stacks of Peking University Library

Wang Haoxian Zhou Yigang Zhang Naishuai Zhang Juan

Abstract: With the transformation and upgrading of the traditional integrated library systems, how to connect the old system and the new service to meet the demand of library development and reader service becomes a technical challenge. To explore the micro-service and FOLIO architecture, this article takes the queue management system of closed stacks of Peking University Library as object to introduce the design ideas, system architecture and technology usage. Besides proposing an improved method of library services based on analysis of the database operations of the system, this article also discusses the technical advantages, application scenarios of micro-service architecture and the changing of library roles under this architecture. Finally, it suggests that the micro-service applied on library service and management is innovative.

Keywords: FOLIO; Micro-service Architecture; Peking University Library; Queue Management System

封面照片简介:哈尔滨工业大学图书馆

哈尔滨工业大学图书馆建于1920年,是黑龙江省高校图书情报工作委员会秘书处挂靠单位,同时肩负着CALIS黑龙江省文献信息服务中心、CASHL成员馆及CADAL三级服务中心的职能,现有两座独立馆舍和建筑分馆,总面积为3.9万平方米。截至2020年5月,图书馆已拥有传统型馆藏320万册/件,电子图书172万种,数据库136个,已形成与学校院系设置相适应的多学科、多语种、多载体、综合性的资源体系。

2019年,哈尔滨工业大学知识产权信息服务中心成为首批高校国家知识产权信息服务中心之一。在服务本校科研工作的同时,还为其他院校的科研工作及地区经济的发展提供有力支撑。在校园文化建设中,图书馆创立“书香苑”阅读推广品牌,开展多种多样的阅读推广活动,培养学生社会主义核心价值观,使其内化于心、外化于行、固化于性,让阅读成为育人育才的助力。

2020年初,完成两个校区馆舍的升级改造,新增解压体验室、全新激光影院、微课录播室、3D体验空间、智能机器人等高科技读者体验空间,利用科学技术的发展提升图书馆服务内容与服务水平,为读者提供方便快捷、智能舒适的阅读服务。改造后的图书馆将建成以读者为中心,以服务为基石,促进学习、协作和研究的全新服务模式,为学校教育教学、科学研究、高效管理、社会服务等方面提供有力支撑。

哈尔滨工业大学图书馆正朝着建设“现代化、研究型、开放文明的世界一流大学图书馆”的发展目标扎实奋进。

2020年
第5期

大學
圖書館
學報