

基于微服务架构的高校办事服务平台优化

王平¹,董国卿^{1†},夏凌云¹,蔡吉刚²

(1. 中国石油大学(华东)信息化建设处,山东 青岛 266580;

2. 中国石油大学(华东)古镇口校区综合办公室,山东 青岛 266580)

摘 要:在国家将数字技术广泛应用于管理服务的要求下,为了提升高校管理服务的便利化水平,对全校所有的管理服务进行整体性设计。同时在原有高校办事服务平台上基于微服务架构进行功能完善,通过将应用和服务分解成松耦合的组件,降低平台变化带来的影响,并对优化后的服务平台进行性能测试。测试结果表明,服务平台获取表单信息的平均响应时间约为 1.1 s,可以更好地提高用户满意度。

关键词:微服务架构;服务平台优化;性能测试

中图分类号:TP39

文献标识码:A

Optimization of University Service Platform Based on Microservices Architecture

WANG Ping¹, DONG Guoqing^{1†}, Xia Lingyun¹, CAI Jigang²

(1. Information Construction Department, China University of Petroleum (East China), Qingdao, Shandong 266580, China;

2. Guzhenkou Campus Comprehensive Office, China University of Petroleum (East China), Qingdao, Shandong 266580, China)

Abstract: Under the requirement of the state to widely apply digital technology to management services, in order to improve the facilitation level of university management services, the whole management services of the university are designed. Meanwhile, the functions of the original university service platform are improved based on microservices architecture, and the impact of platform changes is reduced by decomposing applications and services into loosely coupled components. The performance test of the optimized service platform is carried out, and the test results show that the average response time of the service platform to obtain the form information is about 1.1 s, which can better improve user satisfaction.

Key words: microservices architecture; service platform optimization; performance testing

早在 2018 年 4 月 20 日,习近平在全国网络安全和信息化工作会议上就提出“构建全流程一体化在线服务平台,让百姓少跑腿、信息多跑路,更好解决企业和群众反映强烈的办事难、办事慢、办事繁的问题。”^[1]2021 年 3 月 25 日,《教育部关于加强新时代教育管理信息化工作的通知》明确提出“精简归并不同层级、部门的同类事项,规范工作流程,形成办事指南。利用一体化教育服务平台,推动管理服务全程网上受理、网上办理和网上反馈,……

最大程度减少管理服务前置条件、精简流程环节、缩短办理时间。”^[2]2022 年 10 月,《国务院办公厅关于加快推进“一件事一次办”打造政务服务升级版的指导意见》又提出“推动数字技术广泛应用于政府管理服务,推动更多关联性强、办事需求量大跨部门、跨层级政务服务事项实现‘一件事一次办’,进一步提高企业和群众办事的体验感和获得感。”^[3]由此可见,国家对政务服务事项数字化转型持续深化。

收稿日期:2023-06-30

作者简介:王平(1979—),女,山东威海人,硕士,高级工程师,研究方向:计算机应用,信息化建设,数据挖掘。

†通信联系人,E-mail: donggq@upc.edu.cn

高校师生办事往往经历四个阶段:第一阶段是去各部门办理,即线下分散办理。该阶段办事需要根据各部门的要求分别准备材料,在各部门之间来回穿梭。第二阶段是去实体大厅办理,即线下集中办理。该阶段是建立一个实体大厅,将各部门的办事项集中在一起,统一在此实体大厅中进行办理,解决了办事分散的问题,为师生办事提供一个统一的物理空间,减少穿梭于业务部门间的时间和精力。第三阶段是登录各业务系统办理,即线上分散办理。该阶段是各部门建立自己的业务系统,将办事项搬到网上,高校师生只需要一台电脑就可以办事,解决了线下跑腿的问题,同时管理员也可以在线上审批,缩短了办理时间。第四阶段是登录“一站式”服务平台办理,即线上集中办理。该阶段是将学校的事项都集中在一个平台上,不需要师生来回切换不同的系统,避免登录多个系统的烦琐,并且可以实现多个部门的协同,打通了部门壁垒,提高了办事效率,但是仍存在两方面的不足,即线上服务缺乏整体性设计和平台功能不完备。前者多以管理本位出发,没有从师生服务角度进行办事流程设计,以集中解决重点问题为主,线上流程照搬线下工作模式,办事项相对琐碎,学校总的办事底数不清,不利于管理服务的进一步整合和提质增效。后者是服务平台统一身份认证提供的服务范围不足,未实现统一门户,缺少电子签章、服务评价等功能,不能满足用户的所有办事需求,不利于服务质量的持续提高。

针对上述两个问题,基于微服务架构对高校办事服务平台进行优化设计。通过调研,从“办事者”的角度把多个关联程度大、办理频率高、办理时间相对集中的“单项事”打包为“一件事”,精简流程,缩短办事时间。通过将应用和服务分解成松耦合的组件,完善服务平台功能,降低平台变化带来的影响,实现统一入口、统一认证、数据共享和业务联动,进而提升部门效率,通过评价闭环改进服务,进而提高用户办事的体验感和获得感。

1 高校办事服务平台优化

1.1 架构设计

对用户体验反馈进行研究分析,得出用户需求为增加功能模块和对现有流程进行优化再造。基于微服务架构,在保留原有投资平台的基础上,研究设计服务平台的框架,如图 1 所示。

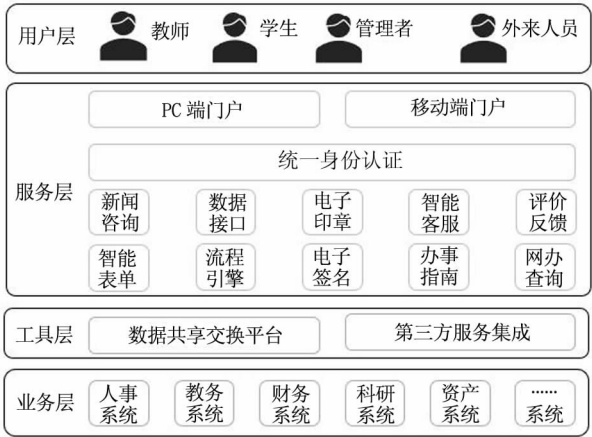


图 1 服务平台框架图

(1)用户层

用户层是该框架的最上层,主要是定义该平台对外的服务群体,优化后的服务平台在原有服务师生和管理者的基础上,增加外来人员,扩大平台的服务范围。

(2)服务层

服务层是该框架的核心部分,为了方便用户随时随地都可以访问使用,设计了 PC 端门户和移动端门户。因为用户层增加了外来人员,所以为了让所有用户实现一次认证后,便可在各系统或服务模块间无缝衔接,设计了统一身份认证,增加注册和激活功能。除此之外,在原有智能表单、流程引擎等功能的基础上,还新增新闻咨询、数据接口、电子印章、智能客服、评价反馈等多项功能。所有办事流程都可以通过数据接口获取其他事项中的数据,达到“一次填报、多次复用”的效果,减少填报量,减轻管理者审核工作量。用户还可以通过订阅的方式获取自己感兴趣的新闻咨询,平台也可以根据用户分类发送不同的通知和消息,实现从“人找信息”到“信息找人”的转变,避免“信息满天飞”的现象。

(3)工具层

工具层是该框架的关键,它通过数据共享交换平台为服务平台提供学校数据仓库中已有的数据,也将伴随式采集到的数据汇聚到数据仓库中,方便校内业务系统间的数据共享。通过第三方服务集成工具,将业务系统中已经实现的服务集成到服务平台上,实现办事服务统一入口。

(4)业务层

业务层是该框架的基础,它产生的数据通过工具层提供给服务层,也可以通过工具层获取服务层采集的数据。

1.2 线上服务整体设计

当前学校使用的服务平台是从管理者的视角出发,各部门通过该平台将线下业务搬到线上,让学校师生办事方便,部门管理员审批快捷,可以满足师生办事的最基本需求,但是师生有时候不知道该找哪个部门,只能从成堆的事项中去翻找,而且有些事项相互之间有关联,师生在办事之初也不清楚,这就延长了师生办事的时间,也对办事部门有情绪,极大地降低了师生的幸福感。因此,要从服务师生的角度出发,对学校各部门的办事项进行摸底,并对每个事项进行调研,在充分理解业务的前提下,减少不必要的环节和前置条件,对服务事项进行整体性设计,将多个关联程度大、办理频率高、办理时间相对集中的“单项事”打包为“一件事”,实现场景化改造,确保部门之间、线上线下的有机衔接。

以“入职一件事”为例,它包含入职手续办理、社保医保登记、财务账号开通、宿舍申请、校园卡开通、电子邮箱申请六件单项事,在当前的“一站式”服务平台上是分散在四个部门中,通过一件事改革将这六件单项事归并为“入职一件事”,如图 2 所示。

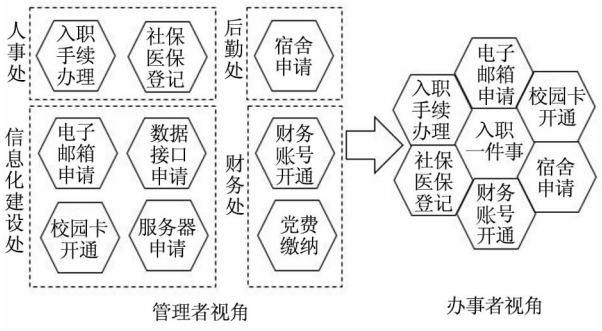


图 2 场景化改造例图

1.3 服务平台功能完善

1.3.1 总体技术路线

该平台采用 B/S 架构,RESTful 架构风格,基于微服务架构的设计模式,前后端分离,主要使用的技术语言是 .net,数据缓存采用高性能的 Key-Value 数据库 Redis,主要负责生产过程中频繁请求的静态、常量等数据的高速缓存,支持字符串、哈希表、链表、集合、有序集合、位图、hyperloglogs 等数据类型,内置复制、Lua 脚本、LRU 收回、事务以及不同级别磁盘持久化功能,同时通过 RedisSentinel 提供高可用,通过 RedisCluster 提供自动分区。构建用户界面采用渐进式框架 VueJS。它采

用自底向上增量开发的设计,其核心库只关注视图层,非常容易与已有项目整合,且有能力驱动采用单文件组件和复杂单页应用。数据库采用 MySQL,并配置了 Cluster,扩容时只需对数据库进行平行扩展即可,具有安全风险低、操作便捷的特点。开放性数据接口采用业界先进的 OAuth2.0 标准规范,在保障安全的前提下,更方便地为开发者带来应用接入服务。

1.3.2 微服务架构技术

微服务^[4]这一概念是由 Martin Fowler 于 2012 年提出的,它的基本思想是将应用系统按照业务功能和技术拆分成彼此独立的最小组件,每个组件都是一个微服务,服务之间通过消息代理、事件流和 REST API 的组合相互通信,如图 3 所示,这些服务拥有自己的技术栈,它们之间可以使用不同的技术栈单独部署,因此基于微服务架构的系统平台可以在不涉及整个应用程序的情况下添加新的功能,代码的更新和修改非常简便,每个功能组件可以独立伸缩,围绕这些组件来创建应用,这些应用可以独立进行开发和管理,在系统平台需要不断优化增加新的功能模块进行完善时,微服务架构的适应性越好。^[5]

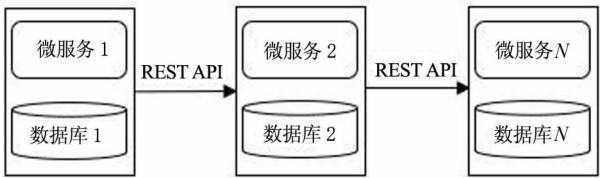


图 3 微服务架构

1.3.3 微服务开发

传统业务功能开发因变动而引起所有代码的重新修改和发布,而微服务架构因为将业务功能拆分为相对独立的微服务组件,更侧重于每个微服务的快速迭代,降低了平台变动带来的影响。以智能表调用电子印章为例说明微服务系统的开发流程。

智能表单微服务使用消息队列发送一个请求消息到电子印章微服务,并等待响应消息。在发送请求消息时,生成一个唯一的标识符作为消息的一部分,并将这个标识符作为参数传递给 ReceiveAsync 方法。电子印章微服务在处理请求并返回响应消息时,将这个标识符作为响应消息的一

部分返回。智能表单微服务在发送请求消息后,等待响应并返回响应消息的 Body 作为执行结果。

```
//智能表单
public class Intelligent_form
{
    private readonly IMessageQueue _messageQueue;
    public Intelligent_form (IMessageQueue messageQueue)
    {
        _messageQueue = messageQueue;
    }
    public async Task<string> CallE_seal ()
    {
        var correlationId = Guid.NewGuid();
        var message = new Message ("E_seal", "get_values", null, correlationId);
        await _messageQueue.SendAsync (message);
        var response = await _messageQueue.ReceiveAsync (correlationId);
        return response.Body.ToString();
    }
}
//电子印章
public class E_seal: IMessageHandler
{
    public Task< Message > HandleMessageAsync (Message message)
    {
        if (message.Action == "get_values")
        {
            var values = new string[] { "value1", "value2" };
            var response = new Message (null, null, values, message.CorrelationId);
            return Task.FromResult(response);
        }
        else
        {
            throw new InvalidOperationException ("Invalid action.");
        }
    }
}
//控制器
[ApiController]
[Route(" [controller]")]
public class MyController: ControllerBase
{
    private readonly Intelligent_form _intelligent_form;
    public MyController(Intelligent_form intelligent_form)
    {
        _intelligent_form = intelligent_form;
    }
    [HttpGet]
    public async Task<IActionResult> Get()
    {
        var result = await _intelligent_form.CallE_seal ();
        return Ok(result);
    }
}
//启动类
public class Program
{
    {
        public static void Main(string[] args)
        {
            CreateHostBuilder (args).Build().Run();
        }
        public static IHostBuilder CreateHostBuilder (string[] args) =>
            Host.CreateDefaultBuilder (args)
                .ConfigureServices( (hostContext, services) =>
```

```
{.....});
        public class MessageQueueListener: BackgroundService
        {
            {.....}
        }
    }
```

1.3.4 服务平台优化效果图

以教师一件事主题集成服务为例,将入职手续办理、社保医保登记、财务账号开通、宿舍申请、校园卡开通、电子邮箱申请六个单独事项打包为“入职一件事”,按照总体框架中用户层分为教师、学生、管理者、外来人员进行设计,服务平台优化后PC端统一门户界面如图 4 所示。为了方便用户随时随地使用平台,设计了移动端的统一门户界面,如图 5 所示。

2 性能测试

用户使用平台的体验效果决定了用户体验感,它除了与页面设计、平台功能等有关,还与平台的性能有关,因此需要对平台的每秒事务处理量和平均响应时间进行测试。

2.1 环境部署

- 搭载的应用环境如下:
 - 应用服务器:CPU 为 24 核;内存为 16 G;硬盘为 500 G。
 - 数据服务器:CPU 为 24 核;内存为 24 G;硬盘为 500 G。
 - 测试服务器:CPU 为 4 核;内存为 8 G;硬盘为 500 G。

2.2 测试原理

用户所有的业务请求都是根据负载均衡策略直达应用服务器,然后应用服务器会根据业务需要跟 MySQL 数据库/Redis 缓存中的一个或者几个发生交互。通过测试工具不断增加线程使服务器不断增加压力来观察服务器的处理能力。当应用服务器的资源利用率接近 100%的极限,取此时的每秒事务处理量和响应时间来判断服务器端系统用户请求的处理能力。在性能测试中,需要重点关注事务的响应时长,主机的中央处理器、输入输出、内存、带宽占用等指标^[6]。

2.3 测试结果

下面采用工具 JMeter 5.4 进行测试,设置最大并发量为 100 用户,压测时采取不断加压的方式,直至测试接口响应时间超过 1 s,同时对其他资源进行监控,如数据库的连接,.net 的使用,网卡的使用等。测试场景如表 1 所示。



图 4 优化后的 PC 端界面



图 5 移动端界面

不断增加并发数,直至接口平均响应时间超过

1 s,性能测试结果如表 2 所示。测试结果详情如图 6、7 所示。从测试结果可以看出该平台高可用。

表 1 测试场景

接口名称	获取表单信息
测试参数	id=1433
测试方法	get
成功标志	Status code: 200
缓存预热	无
响应断言	操作成功
测试时间	5 min
并发数量	200

3 结 论

当前高校办事服务平台多数存在缺乏整体性设计和功能不完备问题,为此,基于微服务架构技术进行高校办事服务平台优化。该研究将业务功能拆分为微服务,逐步完善服务平台功能;整合碎片化、条线化、分散化的服务事项,从被动受理向主动服务转变,实现模式创新;打通管理条块间的信息壁垒与职能边界,推动部门的权限、资源合理配置,重新建构与再造部门职能,实现了职能重构;使

信息的传递和处理更加迅速和便捷,提升管理者的工作效率,通过更全面的数据分析手段促进“经验型决策”到“数据型决策”的转变,实现能力提升;解

决师生与职能部门间的信息不对称难题,打造师生与学校双向互动和共同参与的学校治理新形态,实现多元参与,进而提升学校治理能力。

表 2 性能测试结果

接口名称	并发量	事务数	每秒事务数	平均响应时间/ms				
				平均	最大	90 %	95 %	99 %
获取表单信息	200	51686	157.82	1120	2055	1417	1489	1657

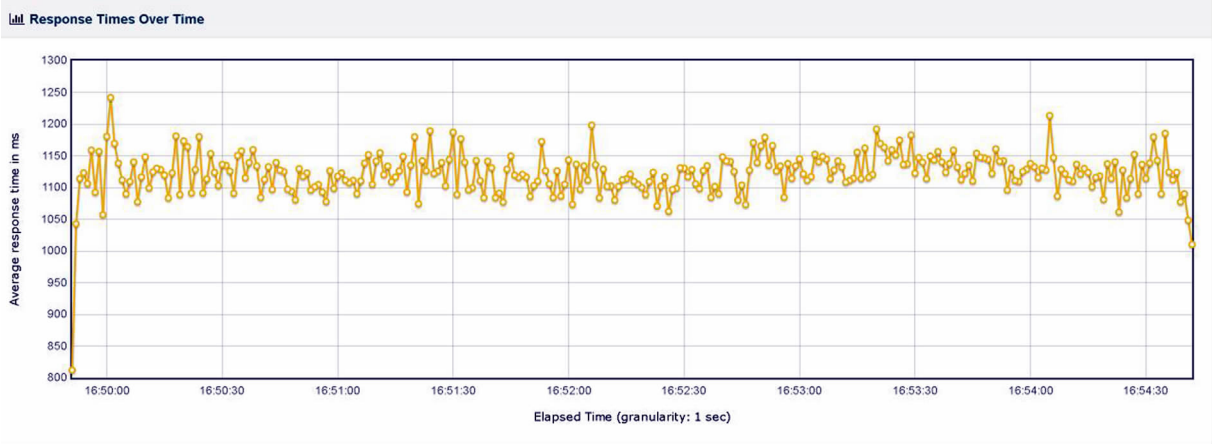


图 6 获取表单信息响应时间测试结果

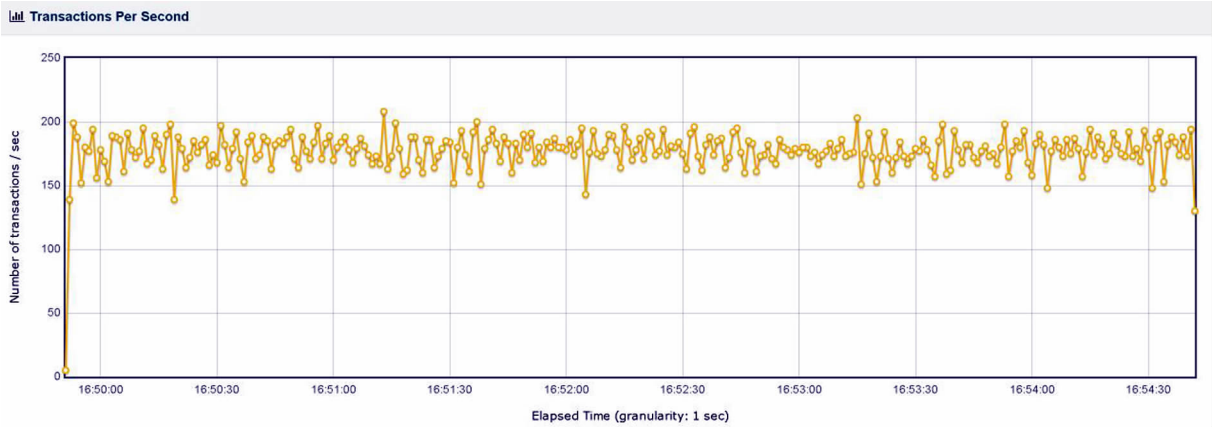


图 7 获取表单信息每秒事务数测试结果

参考文献

[1] 习近平出席全国网络安全和信息化工作会议并发表重要讲话[EB/OL]. http://www.gov.cn/xinwen/2018-04/21/content_5284783.htm.

[2] 教育部关于加强新时代教育管理信息化工作的通知[EB/OL]. http://www.moe.gov.cn/srcsite/A16/s3342/202103/t20210322_521669.html.

[3] 国务院办公厅关于加快推进“一件事一次办”打造政务服务

升级版的指导意见[EB/OL]. http://www.gov.cn/zhengce/content/2022-10/03/content_5715693.htm.

[4] 微服务架构_百度百科[EB/OL]. https://baike.baidu.com/item/%E5%BE%AE%E6%9C%8D%E5%8A%A1%E6%9E%B6%E6%9E%84?fromModule=lemma_search-box.

[5] 李鹏,程建远.基于微服务的地质保障系统架构与应用[J].煤田地质与勘探,2022,50(1):118-127.

[6] 唐承玲,王虎,李光平,等.基于JMeter的Web性能测试研究[J].电脑与电信,2021(6):65-68.