

基于微服务架构的管道地质灾害监测预警系统^①



郇 凯, 黄佳为, 王 鑫, 凌 诚

(北京化工大学 信息科学与技术学院, 北京 100029)

通信作者: 凌 诚, E-mail: s0897918@126.com

摘 要: 随着国家管网建设的不断发展, 系统业务与灾害数据信息规模大幅增长, 同时为防范长距离管道运输过程中易受地质灾害影响发生事故的发生, 开发基于微服务框架的管道地质灾害监测预警系统. 系统采用前后端分离开发模式, 具有高并发、低耦合、高可用、易扩展等特点, 集观测、上报、研究、风险评估、预报预警功能于一体. 目前系统长期稳定运行于国家管网分公司, 有效解决灾害预警、灾害处理、灾害信息管理等问题, 为长距离运输管道行业提供了有效解决方案.

关键词: 微服务架构; 地质灾害; 监测预警

引用格式: 郇凯, 黄佳为, 王鑫, 凌诚. 基于微服务架构的管道地质灾害监测预警系统. 计算机系统应用, 2022, 31(3): 65-74. <http://www.c-s-a.org.cn/1003-3254/8399.html>

Pipeline Geological Disaster Monitoring and Early Warning System Based on Microservice Architecture

HUAN Kai, HUANG Jia-Wei, WANG Xin, LING Cheng

(College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China)

Abstract: With the continuous development of the national pipeline network, the scales of system business and disaster data have increased significantly. A pipeline geological disaster monitoring and early warning system based on the microservice architecture is developed to prevent the occurrence of accidents that are vulnerable to geological disasters during long-distance pipeline transportation. The system adopts the mode of separated front-end and back-end development and has the characteristics of high concurrency, low coupling, high availability, and easy expansion. It integrates observation, reporting, research, risk assessment, and forecasting and early warning functions. At present, the system has been operating stably in national pipeline network branches for a long time, effectively settling disaster early warning, disaster handling, disaster information management, and other issues. The proposed system provides effective solutions for the long-distance transportation pipeline industry.

Key words: microservice architecture; geological disasters; monitoring and early warning

中国地貌类型多样、地质环境复杂、灾害性天气频发, 是世界上地质灾害最严重的国家之一^[1]. 灾害种类主要包括滑坡、崩塌、泥石流、危岩、河道水毁、地裂缝、地面沉降等. 据自然资源部地调局环境监测院发布的《环境监测院发布 2019 年全国地质灾害通报》显示, 2019 年, 全国共发生地质灾害 6 181 起, 共

造成 211 人死亡、13 人失踪、75 人受伤, 直接经济损失 27.7 亿元.

随着我国油气管道建设里程的不断增长, 越来越多的油气管道穿越复杂的地质环境. 管道铺设地点大多位置偏僻、自然环境差, 在复杂地质条件地区, 地质灾害频发会对管道的运行状态产生重要影响^[2-5]. 长距

① 收稿时间: 2021-06-04; 修改时间: 2021-07-07; 采用时间: 2021-07-13; csa 在线出版时间: 2022-01-24

离运输管道具有运输距离长、运行压力高、易燃易爆、点多、线长等特点,沿线地质灾害类型繁多、成因复杂,且危及范围广。发生地质灾害的区域往往交通不便,人迹罕至,隐蔽性高,灾害点地理信息难以精准掌握。暴露于地表外的管道还将会遭到第三方破坏或空气腐蚀;重则造成管道变形、破裂、断裂,甚至介质泄漏等紧急情况^[6,7]。利用现代信息技术手段建立具有良好实用性与可扩展性的信息化集成平台,对灾害点监测数据进行自动化分析与预警,并进行数据可视化呈现,能够为防灾救灾提供辅助决策,对提升地质灾害防治工作的信息化水平具有积极意义^[8-10]。

1 系统架构概述

1.1 系统业务需求

2019年12月国家管网成立,对中石油、中石化、中海油三大公司油气输送业务进行资源整合。随着地质灾害系统接入部门不断增多,原有单服务器架构已不足以应对业务需求。

(1) 地质灾害数据呈现出多源性、多时相性、异构性、多尺度、多分辨率等特点,地质灾害数据日益呈现出海量发展趋势^[11]。随着部门的不断增多,系统接收的数据也成几何级增长,导致服务器压力陡增。2021年6月1日至2021年6月30日,仅一个公司就向平台数据库推送地质灾害数据168 747条,平均每天约5 625条数据。传统单服务器架构难以承受如此大的数据处理量。使用微服务架构,可以分区域部署多个数据库,对数据进行拆分,可以显著降低服务器的压力。

(2) 由于系统接入部门增多,部门业务不同,需求不断变更,应用规模不断扩大,在单体架构下代码会变得非常复杂且耦合度较高,同时不同人员开发代码后会遗留不同问题,维护模块化结构非常困难,不仅增加了开发难度,也影响代码质量。使用微服务架构,可以对业务进行拆分,不同业务由不同微服务实现,每个微服务相互独立,互不影响。可以显著提高团队之间的开发效率。

(3) 由于系统一线应用人员众多,使用过程中难免出现问题,在单体架构下,某个功能的问题可能会导致系统的崩溃。使用微服务架构,各个微服务之间相互独立,同时可以采用断路器、熔断等方案降低系统崩溃的可能性。

(4) 在单体架构开发过程中,不同业务需要在同一份代码中进行修改、测试、部署,开发效率低。使用微服务架构,不同业务对应不同模块,可以分别开发测试

与部署,全自动化,提高效率。

1.2 系统业务拆分

微服务架构将业务边界进行服务微化拆分,根据业务需求将业务拆分为多个独立运行、部署的子服务。从地质灾害监测预警需求角度分析可将服务拆分为地质灾害服务、巡查巡测服务、专业监测服务、预警预报服务、文件处理服务、用户管理服务、用户鉴权服务共7个微服务模块。其中,由于微服务相互调用之间需要用户权限进行配合,因此增加用户鉴权服务,可以保证各微服务之间接口调用、数据传递安全性。

1.3 系统技术选取

管道地质灾害预报预警系统使用前后端分离开发、独立部署的微服务架构。

(1) 数据库

PostgreSQL有丰富的几何类型,如点、线、面等,PostGIS在对象关系型数据库PostgreSQL上增加了存储管理空间数据的能力,符合并且实现了OpenGIS的一些规范,多年来在GIS领域处于优势地位。由于地质灾害数据一般都携带有空间地理数据信息,具有空间关系、空间位置、分类编码、非结构化、海量数据等特征,因此系统采用了PostgreSQL数据库。

(2) 客户端

相较于Angular和React两个流行框架,Vue.js参考了React的组件化和虚拟DOM技术,借鉴了Angular的模板与双向数据绑定技术,同时拥有更加简洁的代码与更小的体积,运行效率更高。因此采用Vue.js实现监测预警系统前端界面。

(3) 服务器

Spring Cloud与Dubbo是目前两款主流微服务框架,Dubbo作为一款轻量开源Java RPC框架,提供服务注册、调用与不完善的断路器功能,仅实现了服务治理。Spring Cloud则提供了更多的微服务解决方案,包括服务注册、调用、服务网关、断路器等,但目前部分组件停止维护与更新,且环境搭建与配置较为复杂。Spring Cloud Alibaba是阿里推出的Spring Cloud第二代实现标准,拥有更完善的解决方案,其整合了目前流行的技术,包括配置管理、服务发现、智能路由、负载均衡、熔断器、控制总线、集群状态等功能。可以协调分布式环境中各个系统,为各个服务提供模板配置。因此采用Spring Cloud Alibaba进行后台服务器开发实现,并提供RESTful风格接口,便于各端进行

数据获取与调用。

1.4 系统结构设计

系统划分为4个层级结构(如图1所示)。

(1) 基础设施层: 数据采集包括管道本体监测、野外监测设备、一线员工采集等方式. 物理组件包括服务器、防火墙、网络设备等物理组件。

(2) 数据资源层: 主要包括 PostgreSQL 数据与 Redis 缓存数据。

(3) 平台服务层: 主要包括业务需求各个微服务模块实现、应用层请求服务从网关实现以及各微服务组件实现。

(4) 系统应用层: 用户通过电脑 Web 端与移动 APP 端对系统进行访问。

1.5 系统架构设计

如图2所示. 系统分为外网部署与内网部署两部分, 外网部署为对应系统应用层, 包括 PC 端、Web 服务与移动端 APP 服务. 内网部署对应平台服务层与数据资源层, 主要包括网关服务、注册、配置中心、全文检索与日志存储。

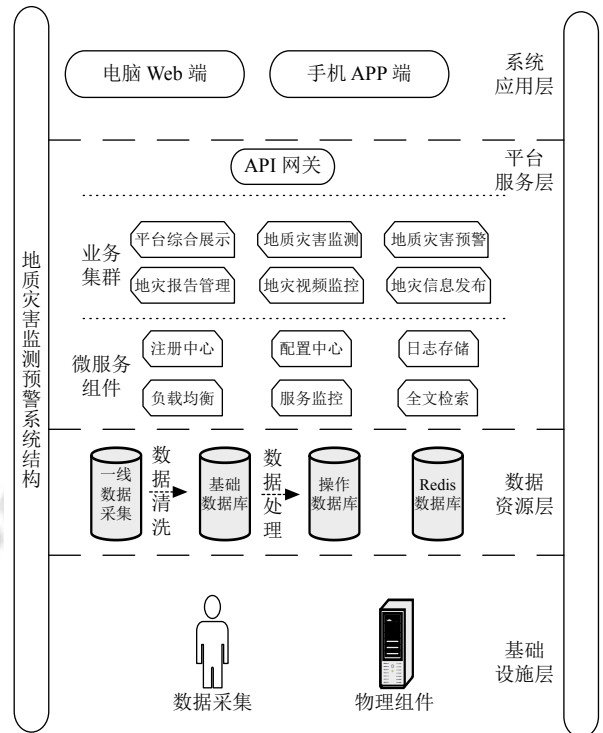


图1 系统体系结构图

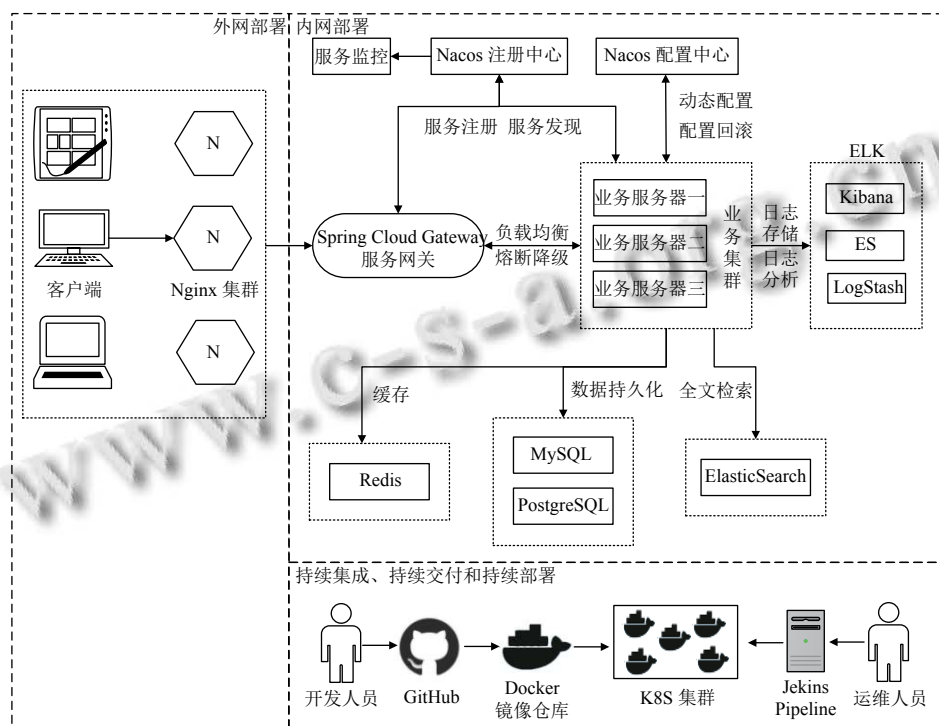


图2 系统架构图

网关服务是外界请求微服务接口的唯一入口, 常用功能包括路由转发、路由过滤、权限校验、限流控

制等. 注册中心用于记录各微服务地址并进行服务健康监测, 配置中心用于将项目中繁琐的配置文件进行

集中管理, 提供统一的请求接口, 并提供动态更新方案. 全文检索可以快速存储、搜索并分析海量数据. 日志存储用于记录系统日志并处理.

2 系统实现

管道地质灾害监测预警系统功能模块如图 3 分为后台服务器、前端客户端和后台管理系统 3 部分.

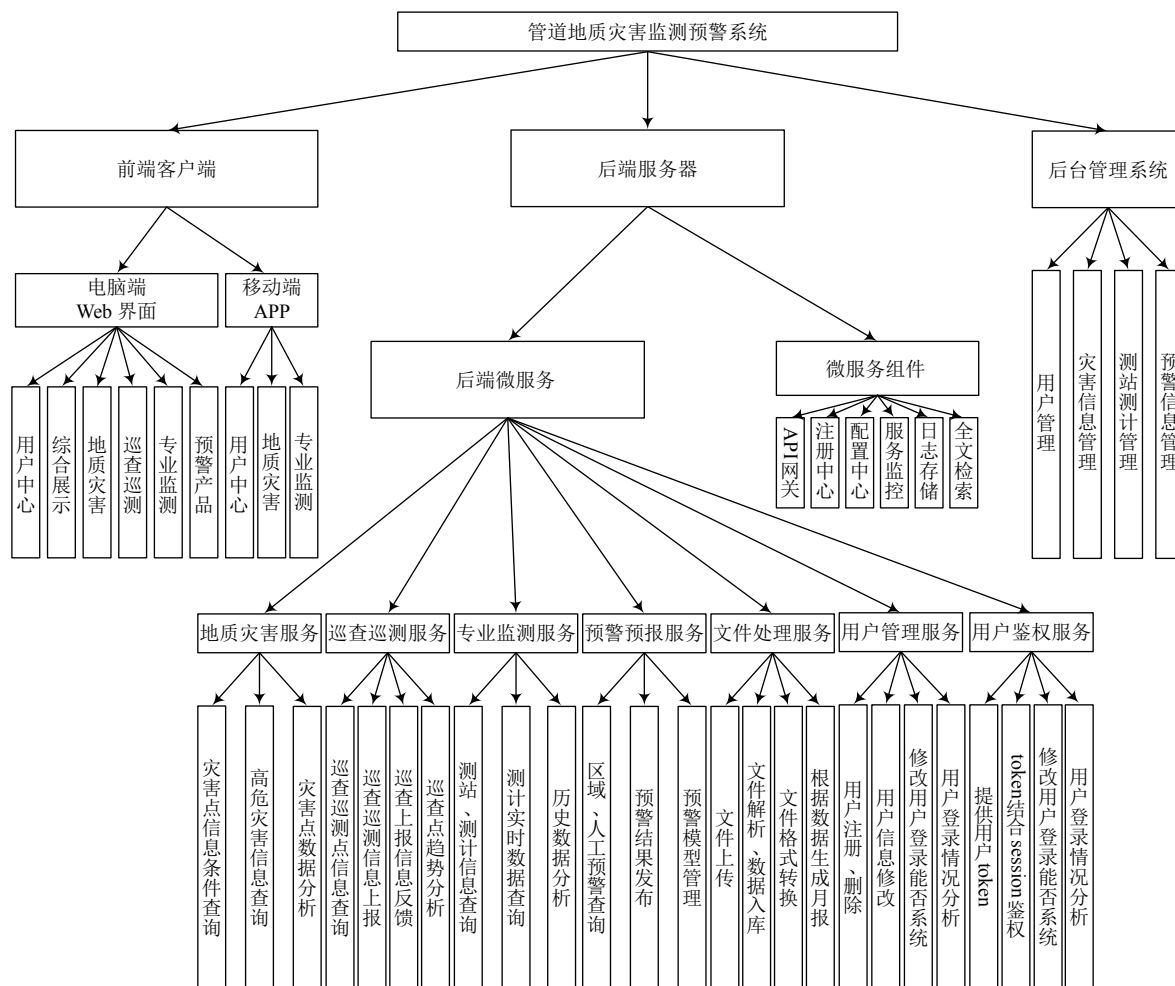


图 3 管道地质灾害监测预警系统功能模块

2.1 微服务架构实现

2.1.1 服务注册与服务发现

随着系统业务量的不断增加, 系统功能变得更加复杂, 微服务的数量也会同步增加. 微服务的架构决定了整个系统的业务功能是由大量的服务结构支撑的, 若采用手动管理和维护服务目录的方式则无法保证系统的稳定运行, 因此需要引入服务注册中心^[12,13]. 微服务将自身的服务名称与服务地址注册到注册中心中, 其他服务可以通过查询注册中心中信息发现服务, 并获取其接口进行远程调用. Spring Cloud 中提供 Eureka Server 注册中心, Spring Cloud Alibaba 中提供 Spring

Cloud Alibaba-Nacos 作为注册中心.

相较于 Eureka Server, Nacos 拥有更好的负载均衡策略, 支持 DNS 访问协议和动态更新, 同时支持配置中心, 便于将配置统一集中管理. 在部署方面, Eureka 需要创建单独的 Spring Boot 项目并加载 Eureka 服务, Nacos 只需要在官网下载 Jar 包即可启动服务, 无须创建新的应用实例. Eureka 2.0 不再开源, 开发者无法得到更好的技术支持. 综合考虑选择 Spring Cloud Alibaba-Nacos 作为系统的注册中心.

启动 Nacos 并运行项目后, 可以在 Nacos 监控界面查看各个微服务的注册、使用情况, 如图 4 所示.

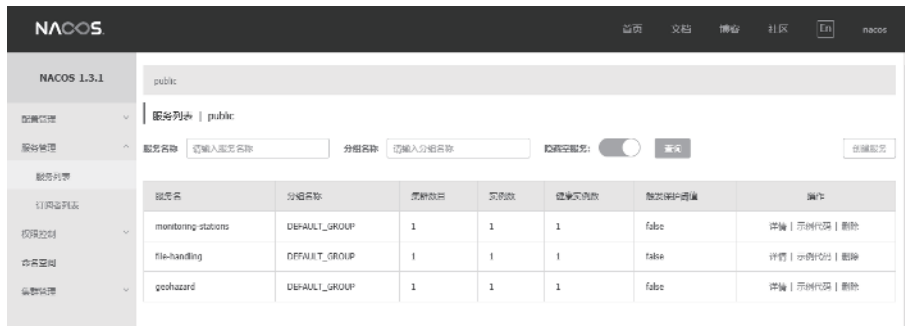


图4 Nacos Server 查看微服务注册列表

服务注册到配置中心后,可以通过 Spring Cloud OpenFeign 进行远程调用. Spring Cloud Feign 在 Netflix-Feign 的基础上扩展了对 Spring MVC 注解的支持,使用 @FeignClient 注解声明接口为远程客户端,并配置远程接口地址,即可对其他微服务进行远程调用.

2.1.2 网关服务与负载均衡

微服务是由很多服务组合而成的系统,每个服务都需要鉴权、限流、权限校验等功能^[14]. Spring Cloud Gateway 是替代 Netflix Zuul 的一套解决方案.网关是系统的唯一入口,其核心是一系列过滤器,不管是来自于客户端的请求,还是服务器内部调用,都需要经过网关,通过过滤器将请求转发到对应的微服务.

网关的核心为路由 (route)、断言 (predicate)、过滤器 (filter) 组成.路由信息由 ID、目的 URL、一组断言工厂与一组过滤器组成,如果当前断言为真,说明请求 URL 与配置路由匹配,进行路由转发至目的 URL.断言为 Spring 5.0 中的 ServerWebExchange 判定请求 URL 是否正确的一组参数,允许开发者定义任何匹配信息,如请求路径、匹配时间、Cookie 信息等.过滤器为标准的 Spring WebFilter,匹配成功后对请求和响应进行修改处理.

在微服务架构中,相同微服务可能部署在不同服务器上,为了更好地选择不同服务器使用,保证服务器不会出现阻塞与闲置,可以负载均衡地调用每一个服务器,提高系统健壮性.常见的负载均衡算法包括:

(1) 轮询:为第一个请求选择健康池中的第一个后端服务器,然后按顺序往后依次选择,直到最后一个选择完继续循环.

(2) 最小连接:优先选择连接数最少,也就是压力最小的后端服务器.

(3) 散列:根据请求源的 IP 的散列 (hash) 来选择

要转发的服务器,可以一定程度上保证特定用户能连接到相同的服务器.

Spring Cloud Gateway 可以结合 Ribbon 实现负载均衡,由于不同服务器之间性能存在差异,因此对轮询进行优化,根据硬件性能配置实例负载的权重,从而更好地利用服务器资源.对于每个请求,遍历集群中的所有可用后端,同时累加所有服务器的权重,从集群中选取最大的服务器,作为本次选定的后端.

2.1.3 全文搜索

由于系统业务量不断扩大,所产生的灾害数据量也不断增长,目前数据量已达百万级别,单纯依靠数据库进行数据搜索查询效率较低,同时无法很好地进行数据分析,快速准确查询获取数据成为提升系统性能的关键点.

ElasticSearch (ES) 是一个基于 Apache Lucene (TM) 的开源分布式可扩展实时搜索和分析引擎,可以快速地存储、搜索和分析海量数据,支持 RESTful 风格.相较于传统数据库,ES 使用倒排索引技术优化索引速度,采取用空间换时间的策略,使得搜索性能显著提高.全文搜索系统结构如图 5 所示.

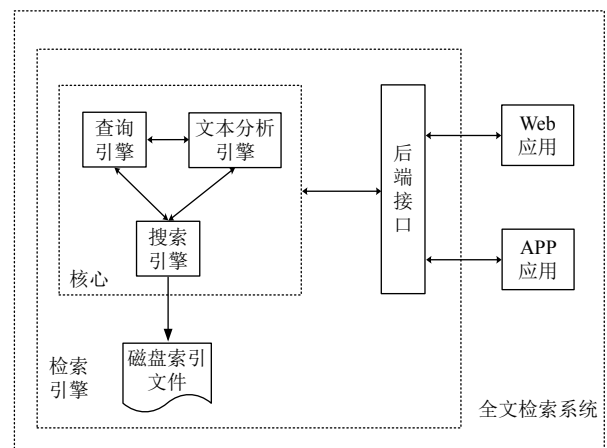


图5 全文搜索系统结构

2.2 前端客户端与后台管理系统实现

客户端界面与后台管理系统均采用 Vue.js 进行开发实现, 引入了 UI 组件库 ElementUI、WebGIS 客户端开发 JavaScript 类库 Openlayer (2D) 与 Cesium (3D)、图表组件 (Highcharts) 等前端开发组件库. 客户端不同模块通过 Axios 向服务器网关发送网络请求, 服务器网关通过断言将网络请求转发到对应微服务模块中, 并返回相应数据.

客户端工作流程如图 6 所示, 在 Vue 工程中, 视图、数据、结构分离, 使数据的更改更为简单, 不需要进行逻辑代码的修改, 只需要操作数据就能完成相关操作. 所有视图和组件均封装在以 .vue 为后缀的文件中. 每一个 .vue 文件中都由 3 部分组成——< template>、< script>、< style>. 其中, template 封装 HTML 界面视图文件内容; script 封装 JavaScript 脚本文件内容; style 封装 CSS 样式文件内容.

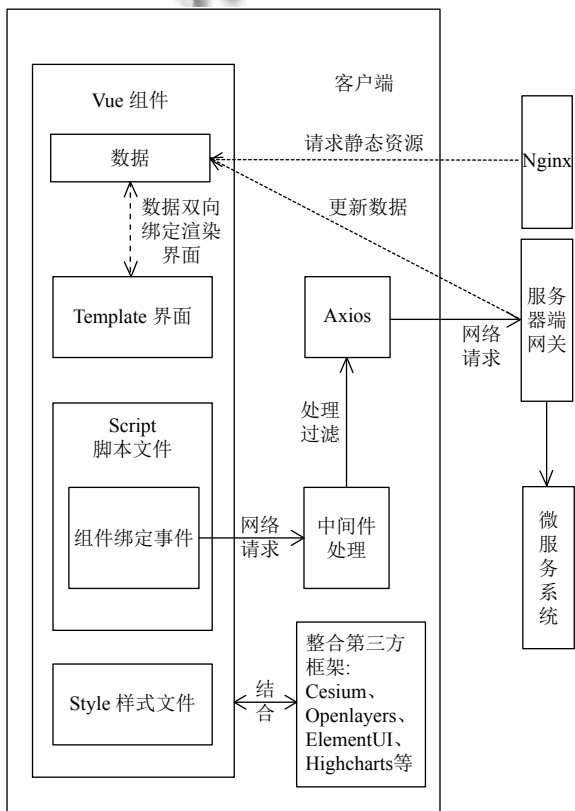


图 6 客户端工作示意图

2.3 持续集成、持续交付与自动化部署

传统的项目开发中, 每次新版本部署需要等全部需求功能完成后, 才能统一进行部署, 并且项目系统与所需

软件采用手动部署的方式, 即每次都需要发布、更新并连接到服务器进行手动部署新版本, 所需人力成本较高.

为了提高部署效率, 节省人力成本, 系统采用 Docker 作为代码与系统运行环境容器, 建立 K8S 集群, 通过 Jenkins Pipeline 实现持续集成、交付与自动化部署. 其执行流程如图 7 所示.

如图 8 所示, 相较于传统手动部署, 自动部署时间从 10 min 提升至 20 s 以内, 大大节省了部署时间成本.

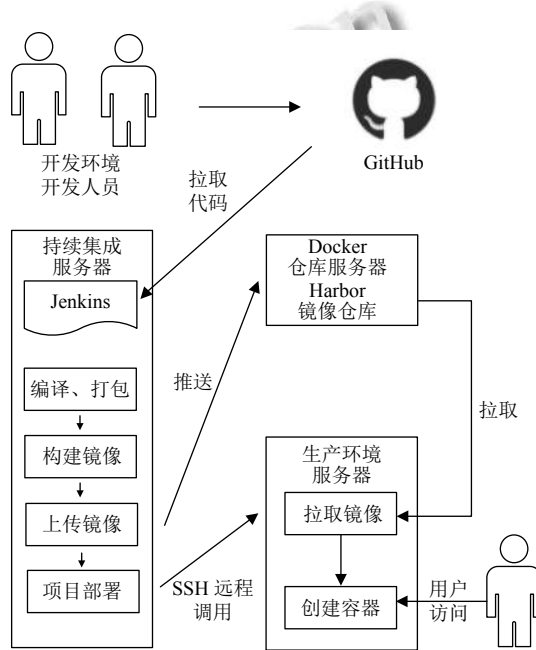


图 7 持续集成、交付与自动部署流程

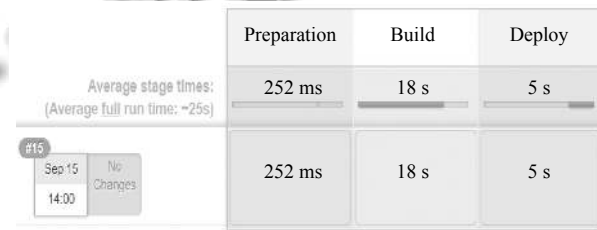


图 8 部署成功效果图

3 并发测试与系统展示

3.1 系统性能压测

系统使用开源压力测试工具 Apache JMeter 对系统地质灾害微服务模块进行高并发性能测试. 测试环境如表 1 所示. 测试数据如图 9 所示.

其中测试数据中线程数为虚拟用户数. Ramp-Up Period 为准备时长, 即设置的虚拟用户数需要多长时间

间全部启动. 循环次数为每个线程发送的请求次数. 例如图中测试数据为每个线程发送 100 次请求, 总请求数为 50 000 次.

性能压测结果如图 10, 详细信息如表 2.

表 1 测试环境

测试环境	客户端	服务器K8S集群
系统	Windows 10 Intel(R) Core(TM) i7-8700	Windows Server 2008 Inter(R) Xeon(R) Gold
CPU	CPU @ 3.20 GHz 3.19 GHz	5218 CPU @ 2.30 GHz 2.29 GHz
RAM	16 GB	32 GB
网络环境	3 Gb/s	18 Gb/s

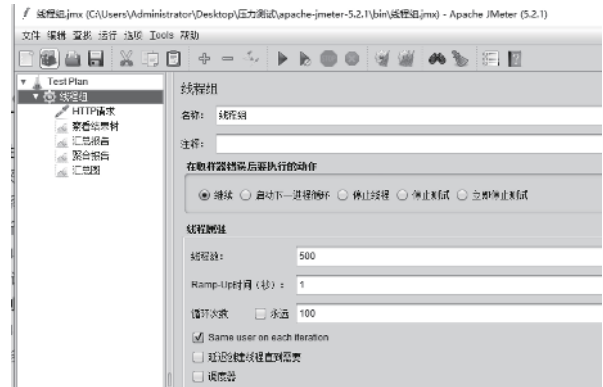


图 9 性能压测测试数据

Latency	# of requests	平均量	中位数	90% 百分位	95% 百分位	99% 百分位	最小量	总人数	完成 %	吞吐量	输出 KBitsec	发送 KBitsec
HTTP 请求	50000	267	218	446	563	703	43	1833	0.00%	1753.7/sec	4276.38	202.09
TOTAL	50000	267	218	446	563	703	43	1833	0.00%	1753.7/sec	4276.38	202.09

图 10 性能压测结果

表 2 测试结果

指标	微服务架构	单体架构
平均请求时间 (ms)	267	862
99%请求消耗时间	446 ms	1.67 s
请求消耗时间最大值	1 833 ms	5.36 s
异常率 (%)	0	3
吞吐量 (1/s)	1 753.7	489.3

由性能压测结果可以看出, 采取微服务架构显著提高了系统的高并发性能, 并大大提高了接口吞吐量, 可以让更多部门参与到系统使用中.

由系统性能压测结果可以看出, 在相同硬件环境下, 采用微服务架构显著提高了高并发下接口的平均请求时间, 并提高了系统的接口吞吐量, 可以显著提高用户体验, 使得系统实现了高性能、高可用.

3.2 系统展示

3.2.1 综合展示

综合展示 (如图 11 所示), 将系统分成不同模块进行展示. 可以纵览系统所有监测数据, 并对预警做出处理. 具有高效、直观、便于操作的优点.

3.2.2 地质灾害展示

地质灾害信息包括信息展示、文件上传、报告查询、灾害管理组成, 主要采用以表格 Datagrid 的方式

进行呈现, 具有多条件查询、模糊查询、分页查询的功能 (如图 12 所示).

3.2.3 地灾概览展示

通过柱状图、饼图与折线图的形式, 直观展示当前部门与下属部门的地质灾害信息 (如图 13 所示), 为日常监管提供有效数据支持.

3.2.4 综合分析展示

将数据以曲线形式展示, 主要展示各个部门、统计历史数据与数据变化规律 (如图 14 所示), 并且可以观察以“年”“月”“日”“总”为单位的不同时间段内的信息, 可以直观有效掌握不同时间段内数据变化情况.

4 系统应用效果

地质灾害监测预警系统自 2018 年 11 月上线以来, 不断进行优化升级, 目前稳定运行于某国有公司, 供 5 条分管道、55 个子分公司与管理处部门、35 个巡线队部门协同使用; 设立监测站 203 个, 包含监测测计 595 个; 已经开展含水率、地下水位等 28 类要素检测, 布置传感器 1 698 套; 建设雨量站记录地质灾害诱发因子数据, 包括管道沿线 2 411 处, 滑坡、泥石流等灾害数据, 包括县市灾害数据 175 100 个, 预警反馈灾害点

雨量站、水利行业提供的管道沿线江河汛期雨情数据及 493 处水文站监测数据、管道专用 173 处雨量站;

记录各地崩塌 2 个、管道沿线灾害点 38 383 个; 累计接收各类灾害信息数据 236 143 654 条。



图 11 综合展示

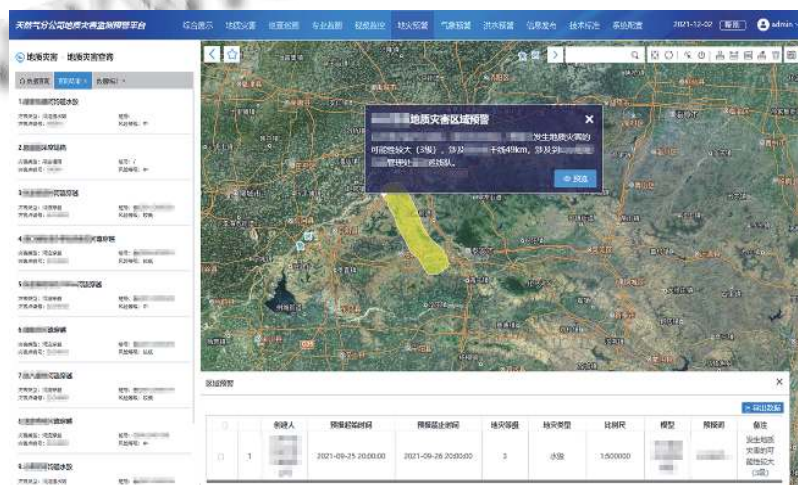


图 12 地质灾害查询

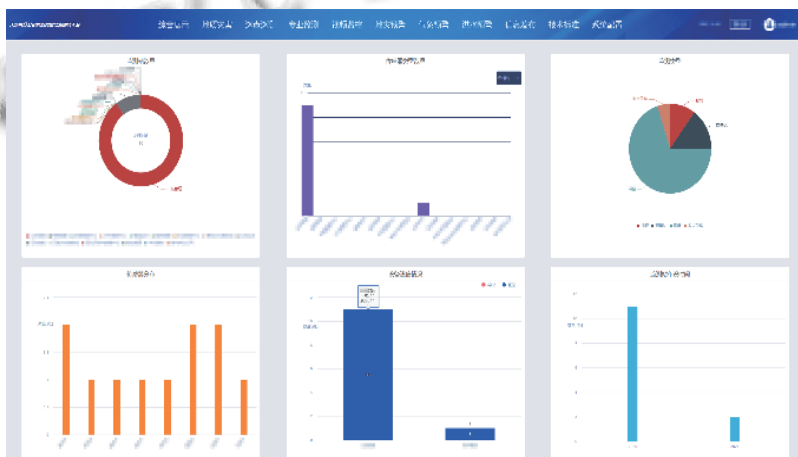


图 13 地灾概览

系统全年无间隔提供预报预警服务, 2019–2021 年期间接收预警数据如图 15 所示, 接收灾害点数据如表 3 所示. 地质灾害上传、文件上报等由之前的 1–2 天缩短至 5 min; 实时监测系统与自动报警功能的存在, 将事故处理时间由原来的 1–2 h 缩短至 15 min 以内, 大大降低了事故发生的概率. 同时, 在每年 4–9 月汛期, 系统为国有管道开展全国区域油气管道、油气田地质灾害预报预警、区域地灾预报预警服务, 保障在汛期管道的正常运行.

由于管道地质灾害大多发生在人迹较少的地区,

以往存在监测难、统计难、管理难等问题, 使用系统后, 可以实时监测数据状态、统计信息解决上述问题. 当测站、测计发生故障报警时, 可以快速定位故障地点, 即刻派维修人员进行抢修, 确保管道运行正常. 管道地质灾害监测预警系统界面简洁, 操作方便, 实现了地质灾害的预警预报与监测的功能, 使得地质灾害工作在巡查勘测、灾害上报、灾害分析、灾害管理、预警预报等工作变得便捷高效. 相较于传统的 Excel 表格办公上报等方式, 节省了大量人力成本、免去了专业人员信息上传更新与查询统计等琐碎操作.

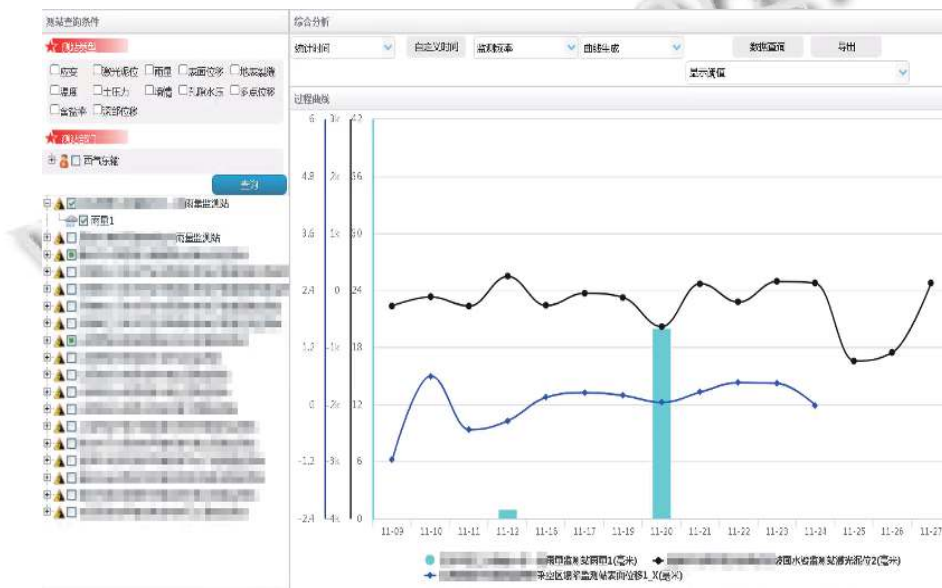


图 14 综合分析

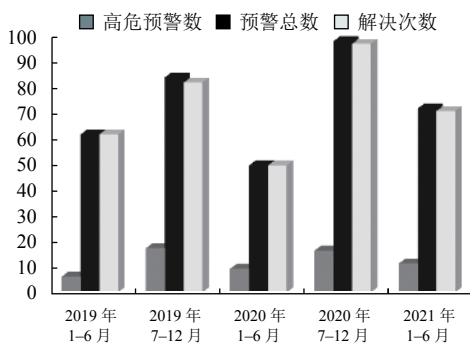


图 15 2019–2021 年期间预警信息

表 3 2019–2021 年期间灾害点信息

灾害点类型	个数	录入个数
地质灾害点	1 979	1 979
巡查巡测点	1 123	1 123

同时, 系统架构层面采用微服务架构, 显著提高了系统高并发访问性能, 并使用持续集成与自动部署, 加速了系统开发迭代周期.

参考文献

- 李忠权, 冷小鹏, 梁军. 基于 SOA 的地质灾害实时监测预警平台设计. 成都理工大学学报 (自然科学版), 2018, 45(5): 606–614.
- 吴玉良, 徐国瀚, 王国付, 等. 滑坡灾害中埋地管道稳定性分析. 中国安全生产科学技术, 2018, 14(12): 73–77.
- 辜冬梅, 姚安林, 尹旭东, 等. 基于层次-模糊评价法的山区油气管道地质灾害易发性研究. 中国安全生产科学技术, 2012, 8(5): 52–57.
- 张伯君. 山体滑坡区域内长输埋地油气管道强度研究 [博士学位论文]. 杭州: 浙江大学, 2013.

- 5 张晶, 刘兴荣, 杨军. 涩宁兰输气管道沿线地质灾害分析及危险性分级. 防灾科技学院学报, 2010, 12(2): 70-73.
- 6 伍运霖, 邓清禄, 安鹏举, 等. 呼包鄂管道鄂尔多斯段水毁灾害特征及防护工程优化. 中国地质灾害与防治学报, 2017, 28(4): 95-102.
- 7 袁巍华, 吴玉国, 王国付, 等. 水毁灾害中埋地管道稳定性研究. 中国安全生产科学技术, 2017, 13(9): 90-95.
- 8 张春山, 张业成, 胡景江, 等. 中国地质灾害时空分布特征与形成条件. 第四纪研究, 2000, 20(6): 559-566.
- 9 殷跃平. 中国地质灾害减灾战略初步研究. 中国地质灾害与防治学报, 2004, 15(2): 4-11.
- 10 黄露, 谢忠, 罗显刚. 地质灾害监测预警信息共享机制研究. 测绘科学, 2016, 41(5): 55-59.
- 11 王帅永, 唐川, 何敬, 等. 无人机在强震区地质灾害精细调查中的应用研究. 工程地质学报, 2016, 24(4): 713-719.
- 12 陈德权. 微服务架构的自然资源外业核查系统设计与实现. 地理空间信息, 2021, 19(5): 123-126.
- 13 顾芒芒, 吴铭程. 基于 Spring Cloud 实现任务调度微服务化的设计与实现. 工业控制计算机, 2021, 34(3): 117-119.
- 14 朱瑞新. 基于微服务架构社会化发行系统设计与实现. 网络安全技术与应用, 2021, (4): 121-125.