DOI:10.13878/j.cnki.jnuist.2019.05.014

陈石定[1]　刘翔[1]　汪应琼[1]

一种基于微服务架构的突发事件预警辅助决策系统的设计与实现

摘要

单体式架构应用在突发事件预警辅助决策系统开发中不能满足现实需求,本文引入了微服务架构设计开发该辅助决策系统.通过分析微服务架构在复杂系统中相对于传统单体式架构的应用优势,设计出一种基于微服务架构的突发事件预警信息发布辅助决策系统,该系统选用 Spring Cloud 微服务框架,并对其进行适当的扩展,创建了基于该系统设计的注册中心与网关.系统采用二三维一体化地理信息系统作为展示平台,通过接入各行业静态、危险源动态监测数据,根据设定的模型进行数据融合、处理,辅助进行预警信息生成、发布及应急处置阶段的指挥决策.所设计方案在湖北省突发事件预警发布辅助决策系统中得以实际应用,验证了该类系统使用微服务架构的合理性和有效性.

关键词

微服务架构;突发事件;预警信息发布;辅助决策系统;三维地理信息系统

中图分类号 TP319

文献标志码 A

收稿日期 2019-04-15

资助项目 湖北省突发事件预警信息发布系统建设项目(鄂发改审批服务(2017)45 号)

作者简介

陈石定,男,高级工程师,研究方向为公众气象服务、灾害应急管理、计算机信息系统.424216117@ qq.com

刘翔(通信作者),男,高级工程师,主要研究方向为灾害应急管理、地理信息系统、计算机信息系统.35161899@ qq.com

1　湖北省公众气象服务中心,武汉,430074

0　引言

权威、精准发布各类预警信息是做好防灾减灾和应对各种突发事件的关键和有效环节[1].目前,我国各省、市依托气象业务系统逐步建立了省、市、县级政府部门统一应用的突发事件预警信息发布平台,作为国家级突发事件预警发布平台的一个分支,承担突发事件预警信息的发布、管理及组织工作.为保证预警信息发布的精准性、及时性以及决策的科学性,提高所发布信息的针对性,达到信息精准、受众靶向的目标,需要有一个科学的辅助决策系统来支持进行突发事件的信息生成、智能发布和灾害的应对工作.突发事件预警信息发布辅助决策系统是针对突发事件的预警信息发布工作智能生成预警信息内容,并根据灾害落区情况,向有需要的人群靶向发布;同时,它也是根据灾害影响及严重程度,按照设定的应急预案进行应对辅助的一个智能化信息平台.

我国在突发事件预警辅助决策方面的研究及系统研发已有一定的基础及实际案例.陈炳洪等[2]介绍了基于“一张图”、“一张网”的广东省突发事件应急指挥决策辅助系统,系统可以展示各类数据,并通过数据融合及致灾模型,实现了气象的影响预报模型与风险等级预警.黄成南等[3]研究了基于三维地理信息系统 Google Earth 的突发事件预警发布辅助决策系统,实现了各应急单位行业数据接入和辅助进行预警信息发布.周洁等[4]提出了将各类数据进行融合,通过大数据分析技术,为应急决策提供依据.曹蕾[5]对预警系统进行了详细的研究,搭建了基于 G/S 模式的突发事件时空平行演化平台,对突发事件中海量、异构的时空数据进行分析识别、组织管理、态势推演和可视化表达,在突发事件预警及应急指挥方面可以起到智能支持作用.我国国土、水利、安监、海事等行业也建设了针对本行业相关灾害的决策支持系统,同样偏重于解决在事件发生后如何进行数据协同、应急管理.刘峰博等[6]研究了大数据技术在轨道交通应急方面的辅助决策应用,通过大数据处理技术构建了应急辅助决策系统中的大数据处理框架,实现了在轨道交通应急辅助决策方面的应用.

这些研究成果或者系统开发应用成果对于突发事件预警、应急指挥决策起到了一定的辅助支持作用.然而,上述成果主要偏重于在

突发事件发生后如何进行辅助预警信息发布,或者在事件发生后对如何进行应急指挥决策提供支持,而对于如何智能生成预警信息内容,并将预警信息精准发布到相关受众,同时在应急阶段接收受众的灾情现场反馈相结合起来辅助进行灾害全流程处置的研究较少.为此,基于业务需求,本文设计并开发了突发事件预警辅助决策系统,实现了预警信息发布阶段通过模型智能生成预警信息,并通过接入的渠道、受众等信息,实现预警信息靶向发布,在应急处置阶段,根据事件处置应急预案及现场实际,靶向发布应急指令.同时,系统通过接收所接入的 APP 用户上报的现场详情,实时了解事件现场的进展,为决策者进行应急处置决策提供辅助.作为一个大型的分布式应用系统,突发事件预警信息发布辅助决策系统除了有运行在服务器上的应用外,同时还有运行在移动端的 APP 应用等.APP 与系统通过部署于云端的服务进行数据共享,系统功能复杂、模块众多,并考虑到后期的功能模块扩展,因此,在系统设计、开发部署时采用了当前比较流行的微服务架构,将各模块进行分解包装成一个个独立的服务进行独立部署,并通过集群进行管理.

1 突发事件预警信息发布辅助决策系统设计思想

1.1 系统介绍

突发事件预警信息发布辅助决策系统是针对突发事件的预警信息发布工作智能生成预警信息内容,并根据灾害落区情况,向有需要的人群靶向发布;同时,它也是根据灾害影响及严重程度,按照设定的应急预案进行应对辅助的一个智能化信息平台.系统利用大数据和云计算技术来挖掘气象信息与突发事件的相关性,构建多灾种影响链分析模型,按照实况(基础信息)、风险(预警信息)和事件等场景之间的相关性,建立全过程的气象应急保障自动化流程,提高突发事件综合研判分析与决策支撑能力,促进灾害应急辅助决策业务转型升级.

湖北省突发事件预警信息发布辅助决策系统是一个部署在省级,省、市、县三级多部门应用的系统,需要接入不同部门所管理的多种类数据.系统基于“一张图”,实现了 GIS 子系统、数据管理子系统、综合监控子系统、综合分析子系统、风险评估子系统、决策联动子系统、效益评估子系统、知识库管理子系统、移动终端决策 APP 子系统等多个子系统,包括服务器端、客户端、APP 应用等,是一个复杂的分布式应用系统,如图 1 所示.

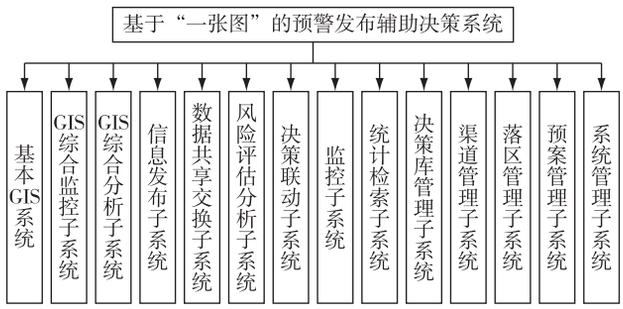

图 1 系统功能结构

Fig. 1 System function structure diagram

1.2 基于微服务架构的开发技术

传统 Web 软件系统的典型架构是单体架构,即将应用程序的所有功能打包成一个应用,每个应用是最小的交付和部署单元,应用部署后运行在统一进程中[7].单体架构的应用具有易于测试、部署的优点.但是,随着互联网的发展及企业应用的扩展,单体架构的应用也表现出代码包庞杂、难于维护、升级扩展困难等固有的缺陷.微服务架构应运而生.

微服务一词来源于计算机科学家 Fowler[8] 的一篇著名的博文,是一种软件设计风格[8].微服务架构是目前软件开发界的一个热门话题.所谓微服务架构,就是将整个系统分解成互相独立的功能模块单元,每个功能模块单元作为一个独立的应用提供服务,专注于具体的业务功能,独立存储数据,按照业务功能模块开发、测试及部署,系统的各个服务之间通过 HTTP API 协议关联起来,组成完整的应用系统,服务之间的通信协议通常采用 RESTful API[9].

相对于单体架构应用,基于微服务架构的分布式应用具备以下特点:

1) 复杂度可控.微服务架构通过将庞大复杂的整体应用分解成一组服务,每一个微服务只提供单一的功能,通过定义良好的接口清晰描述服务的边界,使复杂的功能通过模块化的方式呈现出来,整体上降低了系统的复杂度[10].微服务架构模式加强了一定程度的模块化.由于每一个服务体积小、复杂度低,降低了开发、维护的难度,提高了开发效率.

2) 灵活可扩展.每一个微服务功能相对比较单一,微服务架构模式使得每个服务可以独立扩展,进行单独的功能增减,使得整体变得非常灵活,有效降低风险.

3) 独立部署.系统的每一个微服务具备独立的

运行进程,所以每个微服务可以独立部署.使用相同的部署环境可以实现批量快速部署微服务.

4) 故障隔离.当某一组应用功能发生故障时,在单一进程的传统架构下,故障很有可能在进程内扩展,导致整个应用的运行故障.在微服务架构下,故障会被隔离在单个服务中,通过良好的设计,其他服务可通过退化等机制实现应用层面的容错.

因辅助决策系统首先是一个分布式应用,故在系统设计过程中,需要考虑以下内容:

1)系统包含有多种来源数据的集成,且在后期应用的过程中还可能会集成更多种类的数据;

2)系统集成多个不同的功能单元,接入不同的模型,需要与多个不同的第三方应用进行对接;

3)当出现达到预警模式要求的要素时,系统要能够超越气象灾害预警常规处理流程,如审核、签发、复核等操作,由系统根据实时监测数据自动进行判定,并生成对应的预警信息,进行发布;

4)系统支持决策指令的实时发布;

5)系统需要与众多如移动端 APP、微信公众号、微博等新媒体进行交互;

6)系统需要做到全天候 24 h 不间断监测、运行等因素.

因此,本系统在设计开发时需要采用当前比较流行的适用于分布式开发部署的可扩展软件架构,以做到快速开发部署,减少成本、降低风险.根据采用的平台,本文选用了目前企业 Web 应用开发中的微服务架构进行系统的设计与开发.

1.3 微服务架构的不足及应对措施

采用微服务架构进行分布式应用系统设计开发,可以解决单体系统变得庞大之后产生的难以维护等许多问题,但同样因为功能模块拆分成微服务,也会引发许多原本在单体应用中没有的问题和缺陷.这些问题及缺陷应对措施包括:

1) 服务的边界问题.在应用微服务架构进行系统分解的过程中,容易出现服务之间的边界无法很好界定的情况,造成分解后的微服务过大或者过小.单个微服务过大,会出现如同单体架构系统类似的缺陷,单个微服务过小,则导致管理大量服务的难题.因此,在进行系统设计时,需要对系统的功能、模块进行细致分析,找出最合理的边界,做到耦合度最小,同时满足功能需求.

2) 应用的复杂性.分布式应用相对于单体架构应用来说具有固有的复杂性,需要处理 RPC 或者进程间消息传递时的通信机制,特别是当消息传递过程中出现速度过慢或者不可用等局部失效的问题时,分布式应用的处理复杂性更高.在基于微服务架构的分布式应用中,只有充分考虑到网络延迟、容错、消息序列化、异步、版本控制、负载均衡等[11]对每一个微服务的影响,采取一套完整的策略和机制才能保证各个服务能够正常运行.

3) 分区的数据库架构.在基于微服务架构的分布式应用中,一个微服务通常对应一个数据库.在复杂的应用系统中,同时更新多个业务主题的事务非常普遍,因此在微服务架构应用中,需要同步更新不同服务所使用的不同的数据库,才能够时刻确保不同的数据库系统中数据的一致性.

在系统设计中,有一个 CAP 原则.根据 CAP 原则,在一个较为复杂的分布式系统中,Consistency(一致性)、Availability(可用性)、Partition tolerance(分区容错性)三者需要彼此权衡,不可能同时得到满足,最多只能保证其中的两个方面.因此,需要为每个服务调用做不同的权衡,以解决微服务架构应用中的分布式数据库问题.

4) 部署的复杂性.相对于单体式架构应用,基于微服务架构的分布式应用有较多的微服务组成,而且这些微服务均实例化,作为独立进程运行,这就需要整个应用有较多的配置、部署、监控功能,再考虑到系统运行的可靠性及稳定性,需要有一定的冗余设计,通常采用集群化的部署方案.这使得系统的部署及维护的复杂性大大提高.

2 基于微服务架构的突发事件预警辅助决策系统设计

2.1 微服务框架选择

微服务框架有很多,不同的平台有不同的适用框架.JAVA 平台比较著名的微服务框架有 Dubbo 和 Spring Cloud 框架.经过对比及建设实际要求,选择 Spring Cloud 作为系统开发的微服务应用框架.Spring Cloud 使用 Spring 作为容器框架,SpringMVC 作为模型控制器框架.相较于其他的微服务框架,Spring 是真正的微服务框架,它提供了完整的组件支持,支持跨语言实现,使用简单方便且有强大的社区支持,支持自定义组件,是一套完整的分布式系统实现解决方案.

Spring Cloud[11-12] 是一个基于 Spring Boot 的云服务的分布式框架集合,核心功能包括分布式/版本

化配置、服务注册和发现、路由、服务和服务之间的调用、负载均衡、断路器、分布式消息传递等.它实现了从网关服务到服务发现,再到熔断机制等一系列流程,涵盖了实现分布式系统所需的基础软件组件.其组件架构如图 2 所示.

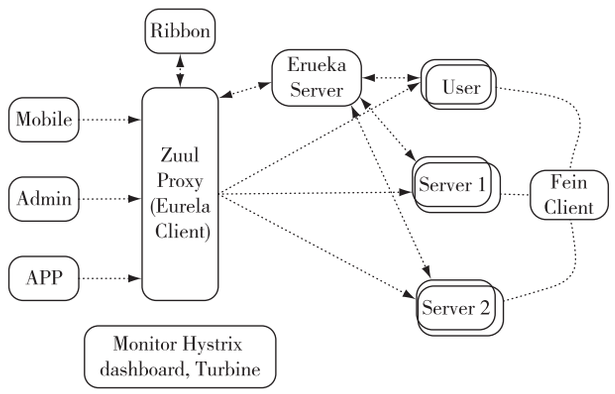

图 2　Spring Cloud 组件架构

Fig. 2　Spring Cloud component architecture

Spring Cloud 基于 Spring Boot,通过在代码中添加依赖和使用注解功能,使得系统的开发和部署变得简单.

2.2　系统微服务设计

根据系统设计,系统分为基础公共模块和业务处理模块.基础公共模块中包括对引用的各类第三方组件的封装、系统设计的各类公共组件模块等;业务处理模块包括进行业务处理所需要的处理、分析、模型等模块.根据前述的微服务架构,以及项目建设实际情况,系统将各功能模块设计为微服务,并将其分为两类:基础微服务和业务定制微服务.基础微服务提供基础功能,业务定制微服务根据功能设计实现各定制的业务功能.系统功能模块分类及相应的微服务设计如表 1 所示.

各微服务设计成独立运行的程序,可以部署在不同的服务器上.通过定义良好的服务接口,使用服务所提供的功能.

2.3　系统的微服务总体架构设计

预警信息发布辅助决策系统由 Spring Cloud 提供的基本组件和构建的业务微服务组成.通过使用 Zuul 组件开发自己的注册服务,实现服务注册,统一提供 REST API,使用 Ribbon 实现负载均衡[13].通过开发 API 网关,实现服务注册中心,并通过 Eureka 实现服务注册和服务发现.针对构建的业务微服务,注册到开发的注册服务中,并通过 Eureka 实现向外

表 1　系统设计的微服务列表

Table 1　UC1 microservice list of system design

序号	分类	微服务名称
1	基础微服务	GIS 组件服务
2		文档插件服务
3		权限管理服务
4		API 网关服务
5		服务注册管理服务
6		日志服务
⋮		⋮
1	业务微服务	信息发布服务
2		短信网关服务
3		预警网关服务
4		监控服务
5		新媒体管理服务
6		综合分析服务
7		风险评估服务
8		知识库管理服务
9		效益评估服务
⋮		⋮

订阅,提供微服务的具体功能.各微服务对应着独立的业务数据库.各库之间的数据有部分冗余.微服务应用架构设计如图 3 所示.

2.4　系统实现

2.4.1　动态信息接入

系统基于 GIS 地图,以各图层叠加的方式展示接入的各行业数据,包括静态数据、动态实时监测数据、历史数据等.静态数据包括监测点、灾害风险点、应急资源场所、应急物资存放信息等;动态数据包括气象部门的监测、预报数据,水利行业的水库、湖泊、河流水文监测数据,国土行业的灾害点实时监测数据,海事部门的船舶实时监测数据,其他部门的视频监控图像信息等;历史数据主要包括监测点(或监测区域)历史上发生的灾害情况数据等.这些数据通过数据拥有部门提供的接口接入到本系统中,并基于一张图,根据决策者需要分类有条件进行展示,同时根据致灾预警模型,对数据进行加工、融合处理后得出预测结果,并有针对性地生成各类预警信息.

针对发布的预警信息,系统实时在图上的对应区域进行高亮提醒,并使用相应的预警图标在预警区域进行标明,动态展现预警情况,如图 4 所示.

2.4.2　致灾预警模型接入

发布预警信息是一件严肃的事情,事关人民群众生产生活、防灾减灾,要有科学性、权威性、实时

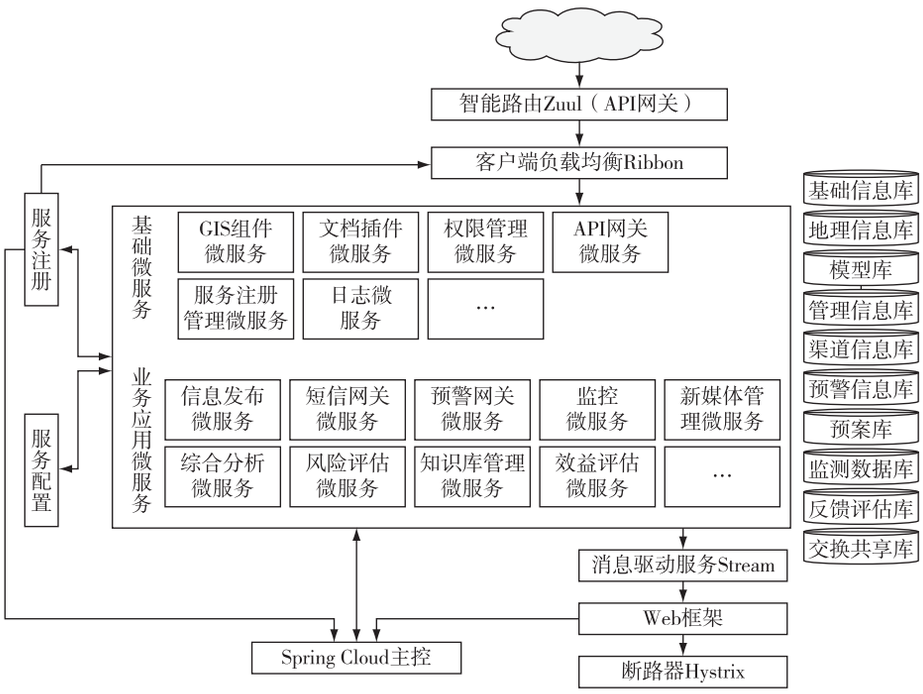

图 3　系统微服务架构设计图

Fig. 3　System microservice architecture design

图 4　接入数据展示

Fig. 4　Access data display

性.预警信息的缺发、滥发、错发都会导致严重的后果,故发布前需要仔细斟酌发布的内容、范围、等级、受众、时间等因素[3].本系统在发布流程上设计了多流程节点进行确认,在预警信息要素的获取上通过采用相应的科学模型进行计算,辅助生成预警信息内容.系统根据实际,引入或接入了多种模型,例如:暴雨洪水水淹模型接入.系统采用了目前研究、使用较多的二维水动力演进模型 Flood Area[14],计算指定区域内淹没范围.国内很多研究人员,如文献[15-17]使用该模型进行了暴雨洪水、淹没区域模拟,证明该模型用以界定洪水淹没范围并预警洪涝风险的精度较高.

Flood Area 模型的计算基于水动力方法,同时考虑当前栅格周围 8 个邻域单元,对邻域单元的泄入量使用 Manning-Stricker 公式[14]进行计算:

$$V = k_{st} \cdot r_{hy}^{2/3} \cdot I^{1/2}, \tag{1}$$

式中:k_{st}是反映地表粗糙情况对水流影响的系数,称之为 Stricker 系数,根据土地利用数据转换得到,系统中将使用到的类型建立表格,通过查表得到;r_{hy}是水力半径;I 为地形(或明渠)的坡度.

水流的淹没深度为淹没水位高程与地面高程之间的差值,淹没过程中的水流方向则由地形的坡向所决定[17].对地面上的任意一点,其坡向表征了该点高程值变化最大的方向,其计算公式[14]为

$$A_{spect} = 270 - \frac{360}{2\pi} \cdot \alpha\tan2\left[\frac{\partial z}{\partial y}, \frac{\partial z}{\partial x}\right], \tag{2}$$

式中:α 为地形坡度;$\frac{\partial z}{\partial y}$为南北方向的高程变化率,

$\frac{\partial z}{\partial x}$为东西方向高程变化率.

Flood Area 模型有 3 种基本的淹没情景:漫顶式、溃口式以及暴雨式.本系统由用户根据实际使用场景选择适合的淹没情景.

系统获取指定区域及周围临近各站点的实时降雨量预测预报数据及观测数据,采用泰森多边形面积权重法计算小时面雨量,然后根据该区域 DEM 分布,确定不同的地形类型,采用不同的 Flood Area 模型计算参数,分别计算致灾雨量、洪水淹没区、警戒区,并将计算结果在 GIS 地图上进行三维叠加模拟显示.同时,系统根据计算得到的致灾雨量数据进行判断并自动进行预警.在雨量达到设定的阈值时,根据计算所得到的淹没区域、警戒区域信息,提取该区域内的学校、村庄等各类场所信息及应急资源等要素,获取各应急责任人、区域内的各种预警信息发布渠道等,自动将其作为靶向发布的受众,提醒发布预警信息和应急指令,为防灾、减灾提前做好应对措施,给决策者或者系统使用者进行决策提供相应的辅助,并根据预案信息实现决策联动.

其他主要接入模型包括污染物扩散模型、城市内涝模型、地质灾害预警模型等,均采用微服务架构集合各种外部门数据和灾害公式模型做上述引用处理叠加呈现在 GIS 系统中.

2.4.3 发布渠道接入

辅助决策系统对于各部门、行业已建的各类发布渠道,如大喇叭、LED、短信、12121 语音外呼、FTP、传真、微信公众号、微博、APP 等,可以通过接口实现接入,并通过 Web Service 回调机制实现状态监测,通过 TCP/IP 的连接状态实现心跳监测,并通过表格或图形化模式直观显示各接入渠道管理平台中所管理的发布渠道的实时连接状态.

在辅助决策系统中,如果监测的要素中遇到有超限情况,或者有比较紧急的预警信息或通知消息、决策指令,可以通过系统中的模型直接生成预警信号或通知消息,通过靶向发布方式或者指定受众发布的形式进行发布.若是有影响较大、涉及面较广的预警信号,需要通过严格的发布流程进行处理,则可直接转到预警信息发布界面,通过录入、审核、签发等规定的发布流程进行处理.

2.4.4 应急支持辅助

系统设计实现了在突发事件已经发生情况下的应急辅助支持.系统可以通过决策联动、预案指令、靶向发布等功能或技术,执行视频会商及会商纪要记录,启动应急预案进行突发事件处置,通过靶向发布技术对预警信息精准发布,通过 APP 上报事件处置情况及灾害详情供决策者了解现场现状等.辅助支持详情信息可在一张图上实时叠加展示,如图 5 所示.

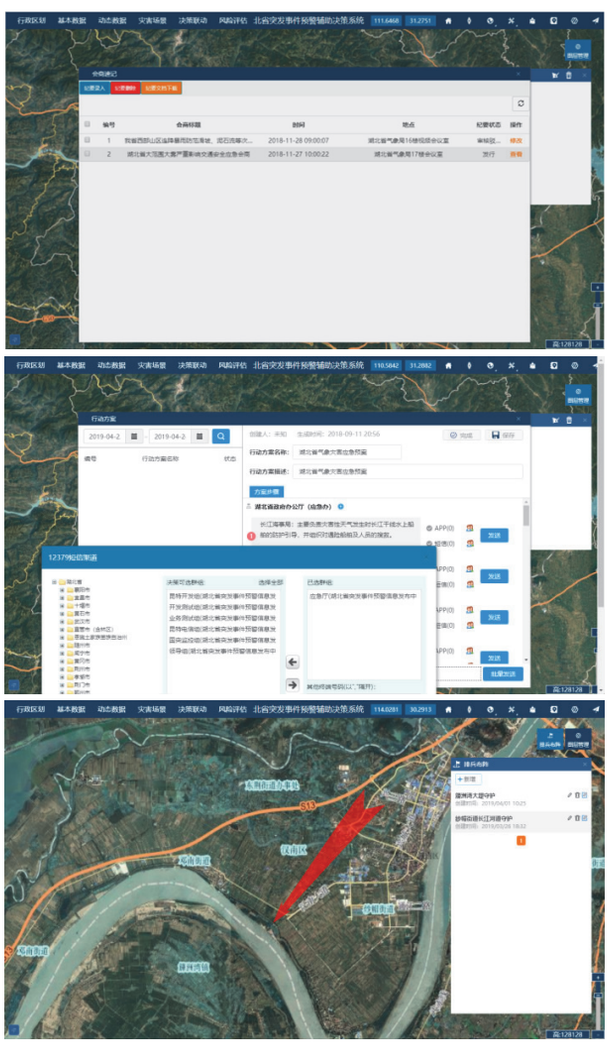

图 5 应急指挥辅助

Fig. 5 Emergency command assistance

2.4.5 灾害四维复盘

针对已经发生的灾害情况,需要做好灾害事件发生过程中各要素过程的详尽记录,事后可以进行灾害发生过程、处置过程的复盘查看,为今后同类事件的处置提供借鉴.

灾害四维复盘以事件进展的时间先后顺序,以二维/三维 GIS 为基础画面,综合展示各种观测数据、评估分析、联动情况,及各关键人员的行动轨迹,直观显示突发事件的演变、处置情况.

3 基于微服务架构的系统开发核心流程处理示例

突发事件预警辅助决策系统采用 IntelliJIDEA 开发平台,Maven 项目管理和综合工具进行系统构建.

在基于 Maven 的系统开发中使用微服务,其流程如图 6 所示[14].

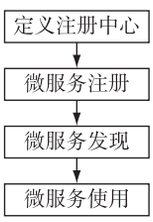

图 6 微服务开发使用流程[14]

Fig. 6 Microservice development process[14]

3.1 搭建(定义)服务注册中心

1)使用 IntelliJIDEA,创建新应用项目 Service-Registry,把该项目作为注册中心的服务.创建完成后先在配置管理文件,例如以下定义项目配置文件 pom.xml,在其中配置必需的依赖:

```
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-eureka-server</artifactId>
</dependency>
```

2)实现注册应用类 ServiceRegistryApplication

例如在代码 ServiceRegistryApplication.java 文件中类声明的前面,添加注解,通过 @EnableEurekaServer 注解启动一个服务注册中心,提供给其他应用进行对话.

```
@ EnableEurekaServer
public class ServiceRegistryApplication {
    public static void main(String[ ]args) {
SpringApplication. run ( ServiceRegistryApplication. class, args);
    }
}
```

3)配置访问路径及相关参数

例如定义 application.yml 配置文件,参数可如下进行:

```
#应用(服务名称):service-registry
spring:
  application:
    name:service-registry
# 定义注册服务运行时提供服务用的端口
server:
#设置该服务注册中心的端口号,此处设为 8761
    port:8761
# 定义 Eureka server
eureka:
  instance:
# 设置该服务注册中心的 hostname
    hostname:localhost
    port: ${server.port}
  client:
#由于该应用是一个服务注册中心,而不是普通的应用,默认情况下,这个应用会向注册中心(即本身)注册它自己,为此需设置为 false 以禁止这种默认行为.
  registerWithEureka:false
#服务注册中心本身的职责就是维护服务实例,不需要去检索其他服务.设置为 false
  fetchRegistry:false
  serviceUrl:
  defaultZone:
http://${eureka.instance.hostname}:${server.port}/eureka/
    server:
  waitTimeInMsWhenSyncEmpty:0
    enable-sel f-preservation:false
```

其中,server.port = 8761,表明通过 SpringEureka 的访问地址,可以访问 http://localhost:8761 以查看注册的微服务信息.

3.2 微服务注册

在完成服务注册中心的搭建后,将其他的服务添加到 Eureka 的服务治理体系中,作为服务的提供者提供相应的功能服务.下面以 user-service 微服务为例介绍微服务的创建及注册流程.

1)创建新应用模块,例如定义为 user-service,并在 pom.xml 配置对 Eureka 的依赖.

配置情况如下:

```
<artifactId>user-service</artifactId>
<packaging>jar</packaging>
<name>user-service</name>
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-eureka</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

2)创建微服务启动类 UserServiceApplication,并

在该类中实现设计好的相应功能.

```
@SpringBootApplication
@EnableFeignClients
@MapperScan("com.user.service.mapper")
@ServletComponentScan("com.user.service")
public class UserServiceApplication extends BaseApplication {
public static void main(String[]args) {
SpringApplication.run(UserServiceApplication.class,args);
  }
}
```

在主类中通过加上@EnableDiscoveryClient 注解,可以激活 Eureka 中的 DiscoveryClient 实现,实现对该服务中的相应服务信息的输出.

3)在 application.yml 或者 application.porperties 文件中配置用户的访问参数,通过 spring.application.name 属性为该服务进行命名,再通过 eureka.client.serviceUrl.defaultZone 属性来指定上一步构建的服务中心的地址.配置情况如下:

```
spring:
  application:
    name:user-service

# Define the port where the Widget Foundry server would be running
server:port:8881

# Define the Eureka server that handles service registration.
eureka:
  instance:
    port:8761
client:
serviceUrl:
defaultZone:http://localhost:${eureka.instance.port}/eureka/
```

通过 server.port=8881,服务注册中心可以发现该服务,并在服务列表中列出该服务的信息.通过在浏览器输入 http://localhost:8761,即可打开 Eureka 信息面板,查看已有注册的微服务.

3.3 微服务发现和路由

经过注册后的微服务可以作为服务提供者为服务消费者提供相应服务.该步骤通过服务发现实现.在实际的实现过程中,为有效降低维护路由规则与服务实例表的难度,解决微服务接口访问时的前置校验冗余,通常使用 API 网关服务,将所有的外部客户端访问通过它进行调度和过滤.因此在本系统中,实现了 API 网关功能.

1)创建 API 网关服务 api-gateway,也在 pom.xml 配置相关依赖:

API 网关服务基于 NexflixZuul 实现.在配置文件中需要引入相应的对 spring-cloud-starter-auul 的依赖.例如:

```
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-eureka</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-zuul</artifactId>
</dependency>
<dependency>
<groupId>com.github.mthizo247</groupId>
<artifactId>spring-cloud-netflix-zuul-websocket</artifactId>
<version>1.0.0</version>
</dependency>
```

2)创建应用主类,并使用@EnableZuulProxy 注解开启 Zuul 的 API 网关服务功能:

```
@SpringBootApplication
//开启 Zuul 的 API 网关服务功能
@EnableZuulProxy
//该应用是一个 Eureka 客户端应用.该注解表明该应用自动具备了发现服务的能力.
@EnableDiscoveryClient
public class ApiGatewayApplication {
public static void main(String[]args) {
SpringApplication.run(ApiGatewayApplication.class,args);
}
}
```

3)在配置文件汇总配置 Zuul 应用的基础信息,如应用名称、服务端口号等.

配置例如:

```
spring:
aop:
proxyTargetClass:true
  application:
    name:api-gateway
# 定义 API 网关服务运行时提供服务的端口
server:
  port:5555
# Define the Eureka server that handles service registration
eureka:
  instance:
    port:8761
```

```
    client:
  serviceUrl:
  defaultZone:http://localhost:${eureka.instance.port}/eureka/
  zuul:
    routes:
      user-service:/user/**
      demo-service:/demo/**
      system-service:/system/**
      ……
    add-proxy-headers:true
  ribbon:
    eureka:
      enabled:true
```

3.4 使用微服务

在 API 中注册的微服务可以向整个系统提供该微服务所具备的功能,即实现服务消费.

1)创建一个使用 User-service 服务的 Spring Boot 基础工程 uaa-service,在 pom.xml 中引入相应的依赖.

配置例如:

```
<name>uaa-service</name>
<description>登录和鉴权服务</description>

<dependencies>
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-eureka</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
……
</dependencies>
```

2)创建应用的入口类.在入口类的实现文件中通过使用@EnableDiscoveryClient 注解让该应用注册为 Eureka 的客户端应用,以获得服务发现能力.

```
@SpringBootApplication
@MapperScan("com.uaa.service.mapper")
//通过使用@EnableDiscoveryClient 注解让该应用注册为 Eureka 的客户端应用,以获得服务发现能力.
@EnableDiscoveryClient
@ServletComponentScan("com.uaa.service")
public class UaaServiceApplication extends WebMvcConfigurerAdapter {
public static void main(String[]args) {
SpringApplication.run(UaaServiceApplication.class,args);
  }
}
```

3)创建 Controller 类接口并实现接口:

```
@RestController
public class Controller implements Service {

@ApiOperation(value="得到用户",notes="查询…")
@RequestMapping(value="/getUser")
@ResponseBody
public User getUser ( ){
  User u=ThreadVariable.getUser( );
  return u;
}
}
```

4)最后,在.properties 配置文件中配置 Eureka 服务注册中心的位置,同时设置该服务消费者的端口.

经过上述步骤,可以通过向 http://localhost:port/uaa-service 发起 GET 请求,实现对服务中相关功能的调用.

3.5 系统应用场景案例模拟

本文选取系统中的监测数据超限时自动发送通知消息功能和灾害四维复盘模块进行系统的应用模拟.

3.5.1 站点监测数据超限自动发送通知消息模拟

系统定时从接入的监测站点获取实时的监测数据,然后与设定监控的站点的阈值进行比对,当是实测值超过阈值引发告警提示时,系统后台自动通过设定渠道发布告警提示信息.发布的告警提示信息可在发布管理平台中查看,如图 7 所示.

3.5.2 灾害四维复盘模拟

系统假设某一区域可能会发生一次灾害,根据业务处理流程,模拟该次灾害发生前后的数据收集融合处理情况、应急处置情况等,事后进行复盘展示,为以后处置该类事件提供参考.

图 8 展示了复盘处置过程中的 3 个显示界面.首先收集各站点监测数据、气象部门的预测预报数据,并进行数据融合处理,根据设定的阈值、模型进行判断处理.当超阈值时,图 8 在可能发生灾害区域进行信息提示,并将可能的灾害落区绘制显示.

图 7　辅助决策系统自动发送的告警通知消息

Fig. 7　Alarm notification message automatically sent by the assistant decision system

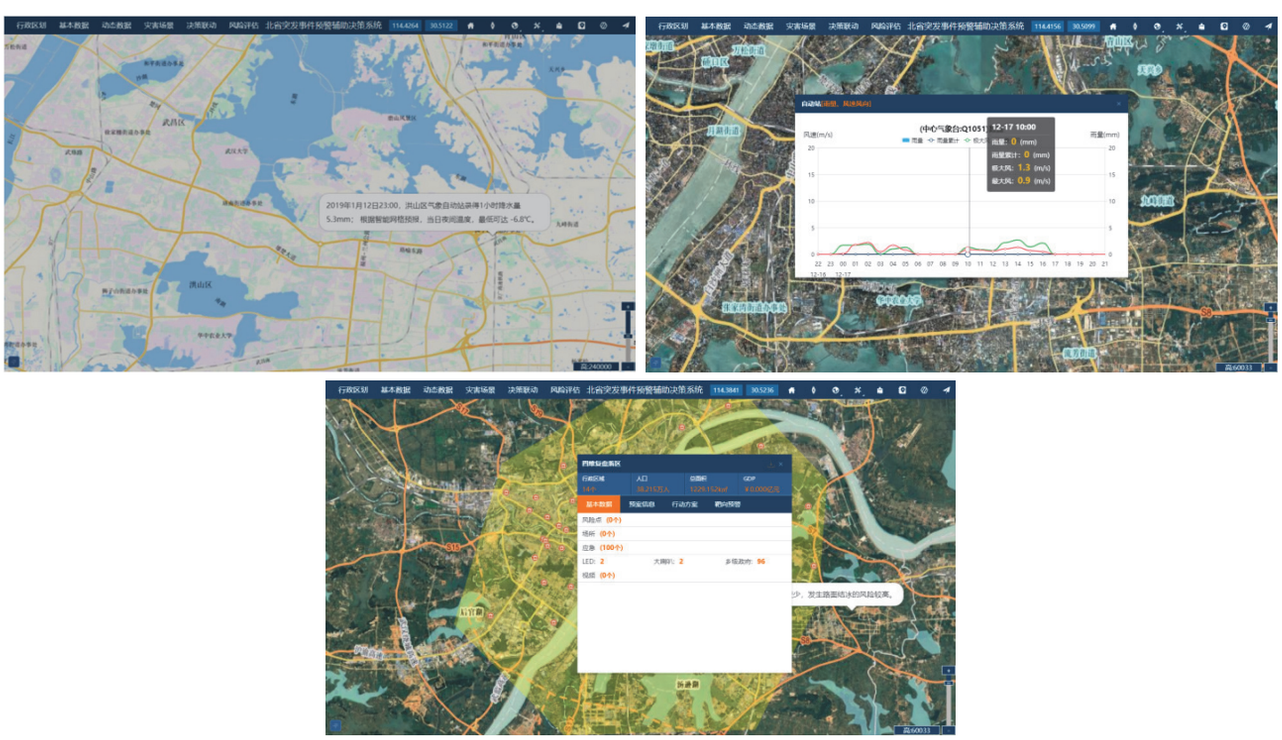

图 8　灾害四维复盘模拟

Fig. 8　Disaster four-dimensional complex simulation

系统根据落区情况,统计出该落区的相关要素,并检索与该落区相关的应急处置预案,根据预案生成相应的处置指令,执行靶向发布操作.同时,系统接收相应的 APP 用户上报的灾情信息进行显示.

4　结束语及讨论

基于 Spring Cloud 架构的微服务框架已经在湖北省突发事件预警辅助决策系统中得以应用,系统实现了融合多部门数据的灾害预警发布、应急指挥“一张图”,验证了微服务架构用于复杂的辅助决策的可行性与实用性,减小了系统各功能模块间的耦合度,更有利于应用系统的扩充和完善,与采用传统单体架构应用的开发过程相比较,降低了开发的难度,与外省同类系统相比提高了系统运行效率,也更

便于发布渠道等扩展系统的对接.

本系统今后的重点研究方向是,随着接入更多的实时数据,通过开发新的微服务,以接入更多的预测预警模型,并将大数据分析技术更好地应用于各类灾害模型,使之不断完善,同时利用已广泛建设的视频监控系统,借助成熟的图像处理技术,使预测、预报更为精确,提升预警信息发布的及时性和覆盖面,提高灾情现场态势监测能力,更好地服务于防灾减灾.

参考文献

References

[1] 郑国光.加快推进预警信息发布系统建设 努力实现权威精准快速安全发布[J].中国应急管理,2016(5):72-74
ZHENG Guoguang. Accelerating the construction of early warning information release system, strive to achieve authoritative accurate, fast and secure release[J]. China Emergency Management,2016(5):72-74

[2] 陈炳洪,曾沁,张毅,等.广东省突发事件应急指挥决策辅助系统的建设[J].广东气象,2018,40(1):39-42
CHEN Binghong, ZENG Qin, ZHANG Yi, et al. Construction of emergency command and decision support system for emergency situations in guangdong province[J]. Guangdong Meteorology,2018,40(1):39-42

[3] 黄成南,王文波,陈妍.肇庆市突发事件预警信息发布决策辅助系统的设计与应用[J].广东气象,2017,39(1):61-64
HUANG Chengnan, WANG Wenbo, CHEN Yan. Design and application of decision support system for early warning information release in Zhaoqing city[J]. Guangdong Meteorology,2017,39(1):61-64

[4] 周洁,许丽佳,林倩.面向辅助决策的突发事件预警信息发布系统研究:以北京市突发事件预警信息发布系统为例[J].科技风,2018(22):76,86
ZHOU Jie, XU Lijia, LIN Qian. Research on emergency warning information release system for assistant decision-making: taking beijing emergency early warning information release system as an example[J]. Technology Wind,2018(22):76,86

[5] 曹蕾.基于 G/S 模式的突发事件时空平行演化方法研究与实现[D].成都:成都理工大学,2013
CAO Lei. The research and implementation of parallel space-time evolution method based on the G/S Model of emergency[D]. Chengdu: Chengdu University of Technology,2013

[6] 刘峰博,干叶婷,周峰.大数据技术在轨道交通应急辅助决策中的应用设计[J].华东交通大学学报,2016,32(2):56-62
LIU Fengbo, GAN Yeting, ZHOU Feng. Application design of big data technologies in emergency decision supporting system[J]. Journal of East China Jiaotong University,2016,32(2):56-62

[7] 张晶,王琰洁,黄小锋,等.一种微服务框架的实现[J].计算机系统应用,2017,26(4):82-86
ZHANG Jing, WANG Yanjie, HUANG Xiaofeng, et al. Implementationof microservice architecture[J]. Computer Systems & Applications,2017,26(4):82-86

[8] Fowler M. Microservices[DB/OL].(2018-03-25).https://martinfowler.com/articles/microservices.html

[9] 张峰.微服务技术构建大规模 web 系统的研究[J].科技创新与应用,2017,22:48-49
ZHANG Feng. Research on building large-scale web system with microservice technology[J]. Technology Innovation and Application,2017,22:48-49

[10] 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
ZHENG Mingzhao, ZHANG Jianqiang. Discussion on the evolution of big platform system architecture based on microservice[J]. Computer Engineering & Software,2017,38(12):165-169

[11] Richardson C. Introduction to microservices [EB/OL]. [2019-04-01]. https://www.nginx.com/blog/introduction-to-microservices

[12] Alberto R. Spring Cloud[EB/OL].[2019-04-01].http://spring.io/projects/spring-cloud,2018

[13] 翟永超.Spring Cloud 微服务实战[M].北京:电子工业出版社,2017
ZHAI Yongchao. Spring Cloud microservices in action[M]. Beijing: Publishing House of Electronics Industry,2011

[14] Geomer. Flodarea-arcview extention for calculating flooded areas (user manual version 10.3)[M]. Heidelberg,2015

[15] 谢五三,田红,卢燕宇.基于 FloodArea 模型的大通河流域暴雨洪涝灾害风险评估[J].暴雨灾害,2015,34(4):384-387
XIE Wusan, TIAN Hong, LU Yanyu. Risk evaluation of rainstorm and flood disasters in Datong river basin based on the FloodArea model[J]. Torrential Rain and Disasters,2015,34(4):384-387

[16] 苗茜,谢志清,曾燕,等.基于统计-FloodArea 模型的平原水网区致灾临界雨量研究[J].自然资源学报,2018,33(9):1563-1574
MIAO Qian, XIE Zhiqing, ZENG Yan, et al. Research of critical rainfalls in wet-net plains based on statistical method and FloodArea[J]. Journal of Natural Resources,2018,33(9):1563-1574

[17] 张磊,王文,文明章,等.基于“FloodArea”模型的山洪灾害精细化预警方法研究[J].复旦学报(自然科学版),2015,54(3):282-287
ZHANG Lei, WANG Wen, WEN Mingzhang, et al. Research on refined early-warning method of mountain flood disaster based on FloodArea[J]. Journal of Fudan University(Natural Science),2015,54(3):282-287

Design method of early warning assistant decision system for microservice architecture

CHEN Shiding[1] LIU Xiang[1] WANG Yingqiong[1]
1 Hubei Public Meteorological Service Center,Wuhan 430074

Abstract A single architecture cannot meet the demand for the development of an emergency early warning auxiliary decision system.In order to solve this problem,a microservice architecture is introduced to design and develop the system.By analyzing the advantages of applying a microservice architecture in a complex systeminstead of a traditional monomer-type structure,we designed a microservice architecture based on the emergency early warning information release auxiliary decision system.This system selects the Spring Cloud service framework,and makes an appropriate extension on itto create the registry and gateway based on the system design.The system adopts a 2D and 3D integrated geographic information system as the display platform.By accessing the static monitoring data and hazard source dymamic monitoring data on various industries,the system can integrate and processes data according to the set mode,and assists in command and decision-making in the early warning information generation,release,and emergency response stages.This design scheme has been applied tothe Hubei province emergency early warning and release aided decision system.The rationality and effectiveness of this kind of system,which uses a microservice architecture,has been verified.

Key words microservice architecture;emergency;early warning information release;assistant decision-making system;3DGIS