

杨理臣,帅向华,董翔,等,2022. 基于 MongoDB 的地震风险评估综合地情数据服务平台设计与实现. 中国地震,38(1): 120~130.

基于 MongoDB 的地震风险评估 综合地情数据服务平台设计与实现

杨理臣¹⁾ 帅向华²⁾ 董翔³⁾ 刘钦²⁾ 李继赓²⁾

1)青海省地震局,西宁 810001

2)中国地震台网中心,北京 100045

3)山东省地震局,济南 250014

摘要 本文介绍了基于 MongoDB 的地震风险评估综合地情数据服务平台,以灵活的文档结构存储与查询为目标,使用 Web Service 技术,采用前后端分离的开发方式,给出了系统的总体设计以及关键技术和功能,实现了全国区县级地震风险评估综合地情数据更灵活地上传、编辑、查询等功能。通过建立全国统一的地震风险评估综合地情数据服务平台,对数据进行可视化展示,提高了全国综合地情数据的利用率和服务能力。

关键词: 综合地情数据 自由存储 前后端分离 MongoDB

[文章编号] 1001-4683(2022)01-0120-11 [中图分类号] P315 [文献标识码] A

0 引言

“十五”期间,中国地震局建设完成了地震应急指挥系统,地震应急基础数据库是整个地震应急指挥系统进行抗震救灾指挥决策的基础和核心(谢江丽等,2019)。地震应急基础数据库包括基础地理数据、社会经济统计数据、地震基础数据、灾害影响背景、救灾力量储备、震时应急联络和地震应急预案等方面内容。该数据库以 GIS 空间数据库的形式存储,在地震应急中发挥着积极的作用(梁芳,2013)。随着应急指挥技术系统的发展,数据库的内容也越来越丰富和精细,为了将庞杂的数据更好的展示和利用,需要利用数据挖掘、数据可视化等手段,通过文字、图片、表格等形式归纳总结出某一区域的特征,从而更好地服务于地震应急和地震风险评估领域。

为了快速掌握某一地区的基本情况,需要通过地震应急基础数据库综合整理全国各县的基本情况,关注地区的行政区划、人口经济、建筑特征、地震地质等方面的内容,提炼出

[收稿日期] 2020-12-25 [修定日期] 2021-07-25

[项目类别] 国家重点研发计划(2018YFC1504501)、地震应急青年重点任务(CEAEDEM202109)共同资助

[作者简介] 杨理臣,男,1982年生,高级工程师,主要从事地震灾害应急、地震信息化工作。

E-mail:18297108539@163.com

帅向华,通讯作者,女,1973年生,研究员,主要从事震害预测、地震灾害应急、地震信息化、GIS应用和遥感应用研究。E-mail:Shuaixhua@sina.com

能够服务于地震领域的地方情况概要信息,形成地震风险评估综合地情数据库。该综合地情数据库目前采用 Word 形式进行存储与展示,以区县为单位,将所有内容放在一个 Word 文件中,制作完成后打印备用。在使用过程中存在内容检索不便、展示方式单一等问题,并且零散的文件限制了信息共享和及时更新,易造成版本混乱,难以向全国范围提供服务。为了更好地存储与使用综合地情数据,亟需开发一套查询展示系统,提供统一平台面向全国范围服务。

近年来,非结构化数据的大量产生带动了大数据解决方案逐渐走向成熟,信息社会智能化程度得到大幅提高(雷德龙等,2014)。一些大型互联网公司对非结构化数据库 MongoDB 的使用已经颇为普遍。优酷每天的日均访问量超过 6 亿,仅在日常运营数据方面,每天产生的日志总量均在 TB 级别,其在线评论数据存储大部分已迁移到了 MongoDB 数据库;阿里开放云平台、天猫商城、新浪微博、亚马逊等知名互联网公司都在很多业务场景应用了 MongoDB 数据库管理技术。随着大数据可视化、机器学习等技术的不断发展,如何更好地将新技术、新成果以及交叉学科的研究进展应用于地震信息服务工作,也是当前地震信息化建设中亟须解决的问题(郑通彦等,2021)。针对现有综合地情数据库的不足,本文基于 MongoDB 数据库,以数据灵活的存储和共享为目标,使用 Web Service 技术实现综合地情数据的录入、更新与查询。同时提供版本管理功能,保证了数据的现势性和接口服务的一致性。采用前后端分离的开发方式,通过地震风险评估平台将数据可视化展示,并且可以根据各省份需求,利用同一个后台服务开发多种前端应用,为数据共享和应用提供更加多维、全面的数据服务形式。

1 总体设计

1.1 数据对象分析

地震风险评估综合地情数据库是对地震应急基础数据库的内容进行挖掘与总结所形成的能够综合描述各区县基本概况的文档资料。其内容由地理位置、行政区划、地形地貌、水系水库、气候特征、人口民族、社会经济、建筑特征、交通概况、学校教育、医疗卫生、重要目标、避难场所、地质构造、历史地震、区域特征、地方联络等条目组成,每个细分条目下包含文字形式的描述及图片、表格信息,如图 1、表 1 所示。该数据库可服务于地震监测、地震应急、震害防御等领域,可以快速了解、查询基本情况和地域特点,有利于地震风险评估等工作地开展,其数据的复杂性有以下两个特点:

(1) 数据条目的复杂性

由于全国不同地区对监测数据应用、震后应急需求、服务对象需求等差异,各省份往往自行决定文档条目内容的详细程度,或在满足规范、规定的基础上,根据实际需求自行增加新的条目内容,以便适应各省份的实际特点。如在青海等地区,寺院的分布较为重要,人口分布往往围绕寺院呈聚集状态,可以将寺院分布情况单独列出作为一个地域的特征。

(2) 条目内容的灵活性

数据条目可由文字、图片和表格三部分组成,表现形式多样。例如,一些省份将重要目标数据直接以文字、图片描述重要目标的数量和分布情况,一些省份则更细化地将重点目标以表格形式详细列出;而对于地方联络数据,一些省份以文字形式直接给出,一些省份则直

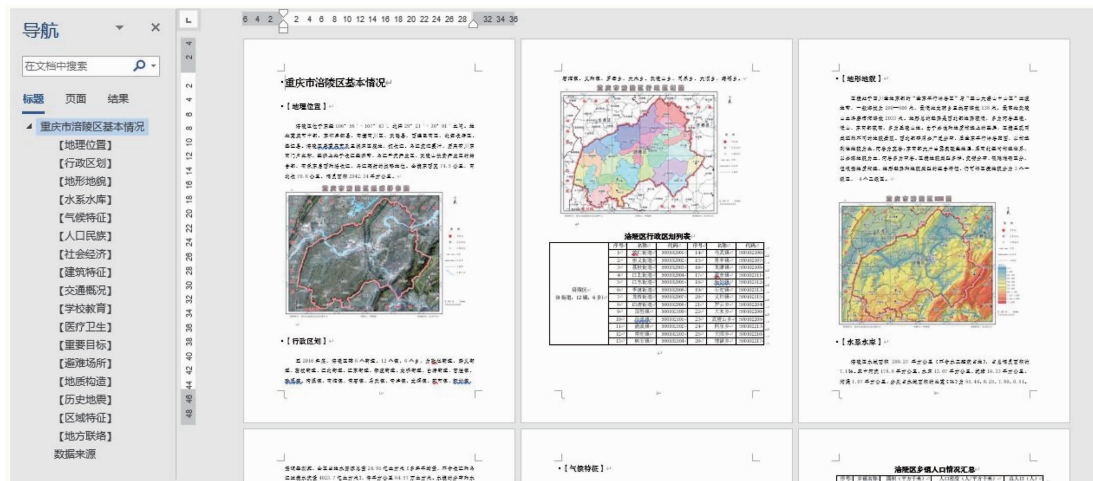


图 1 综合地情数据库样式

表 1 综合地情数据库条目内容

标题	内容	文字	图片	表格
地理位置	描述基本地理位置信息及距省会距离	有	有	无
行政区划	辖区基本情况及政府驻地	有	有	有
地形地貌	地形地貌、海拔情况描述	有	有	无
水系水库	水系、水库情况综述,病险水库情况	有	有	有
气候特征	气候情况综述,气温、降水、日照及自然灾害等	有	无	无
人口民族	人口和民族情况综述	有	有	有
社会经济	经济情况综述	有	有	有
建筑特征	设防烈度及各类建筑结构特征、抗震能力描述	有	有	有
交通概况	交通情况概述,公路、铁路、机场等	有	有	可选
学校教育	学校情况概述	有	有	可选
医疗卫生	医疗卫生情况概述	有	有	可选
重要目标	危险源、旅游景点、文物古迹等重要目标宏观描述	有	有	可选
避难场所	避难场所面积、分布情况等宏观描述	有	有	可选
地质构造	地质构造特征	有	有	无
历史地震	主要历史地震震例,包含最高烈度、伤亡情况、房屋倒塌情况等	有	有	有
区域特征	需要特别指出的地域特点	有	可选	可选
地方联络	地方政府、地震部门联络信息	可选	可选	无

接以表格形式列出。

由此可见,形式为半结构化的数据需要在选择数据库时充分考虑其无固定模式和可横向扩展的特性。传统关系型数据库的存储模式难以横向扩展,限制了数据模式的灵活存储,也造成了较高的维护成本。因此,引入非关系型数据库 MongoDB 将彻底解决传统关系型数据库在这类非结构性数据的存储和处理上存在的问题,实现文档存储的灵活自主。

在文件组织上,综合地情数据库共汇集了全国共 3100 余个区县的信息,其文件名命名

规范为“区划代码(区划名称).更新日期.docx”的形式,如“500101(重庆市万州区).201712.docx”。各省份提交数据时,以更新日期来区分地情数据的版本新旧。随着时间的推移,其文件数量也不断增加。因此,在查看文件时需要从各种版本的区县地情文件中找到最新的信息,使用数据库进行设计时也要考虑到归档单元管理的方便和版本检索的需求。

1.2 系统总体架构设计

基于综合地情文档结构类型的特点,本平台使用 MongoDB 数据库存储文档,利用 Web Service 技术对数据提供统一管理的服务接口,系统架构如图 2 所示。

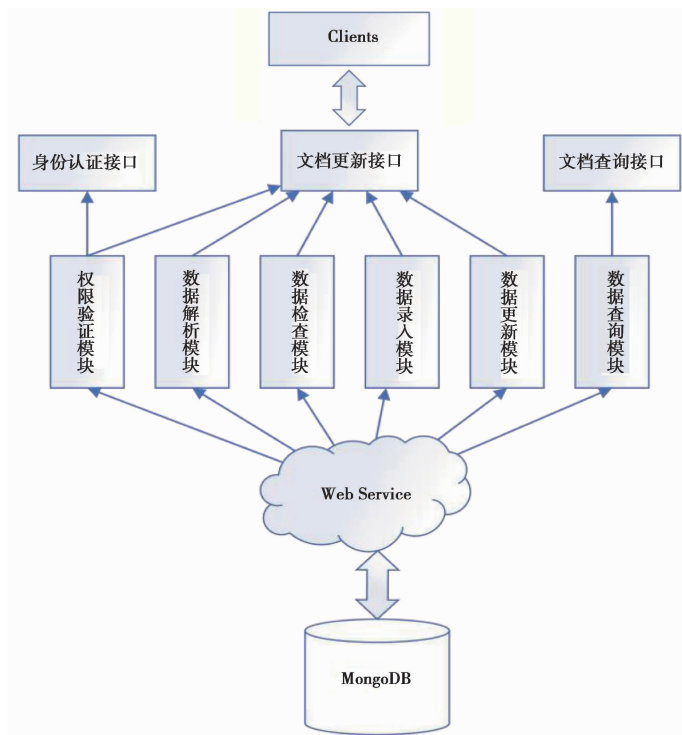


图 2 系统总体架构图

Web Service 技术提供的云服务作为整个系统的核心,提供了综合地情数据管理所需要的所有接口服务,利用该平台接口可使得任何客户端具有综合地情文档管理的能力。其流程如下:首先,通过身份认证接口获得接口授权;然后,利用文档更新接口将综合地情文档上传到服务端,服务端自动解析文件内容,验证文档内容的合法性,并根据文件命名规则比对数据库中的数据,进行文档录入或更新操作;最后,利用文档查询接口从数据库中检索所需要的信息,并根据数据内容和系统需求进行个性化的可视化展示。

2 关键技术

2.1 模式自由存储

传统的结构化数据库系统通常事先规定好数据库表格的结构,一旦涉及到数据库表格结构的调整便会带来大量的工作。全国现有 2800 余个区县级行政区划单位进行综合地情

文档类型的存储,未来扩充到乡镇级地情数据库后,行政区划单位数量将增加到 4 万个以上。地情文档每个条目均可包含文字、图片和表格等动态内容,且各省份可根据实际需求扩展本地化特色的条目内容。不同的地区各有特色,内容需求具有个性化,采用传统的结构化数据库存储方式,修改一个字段将影响全国所有地区的数据存储,难以实现各省份内容的灵活调整。因此,实现一个模式自由的存储框架具有重要意义。

MongoDB 是一个基于分布式文件存储的非结构化数据库,旨在为 Web 应用提供可扩展的高性能数据存储解决方案,是功能最丰富、最贴近关系数据库的非关系型数据库。它是一种面向集合、模式自由、文档型的数据库 (Banker, 2011),采用类似 JSON 的格式存储文档型的数据,该类型的数据存储不需要固定的模式,无需多余操作就能实现横向扩展,可用来存储较为复杂的数据类型。

图 3 展示了利用 MongoDB 存储的文档结构图,对于每一个区县的综合地情文档,采用一个独立的 Collection 存储,其名称为行政区划 ID,而其条目则由基本信息 (Info) 和拆分为多个条目文档的 Section 表示,基本信息存储了该区县的名称、编码和最近更新版本等信息,每个 Section 由必须的字段条目标题 (title)、文字内容 (content) 和版本 (version) 以及可选内容如图件 (figures) 和表格 (tables) 组成。

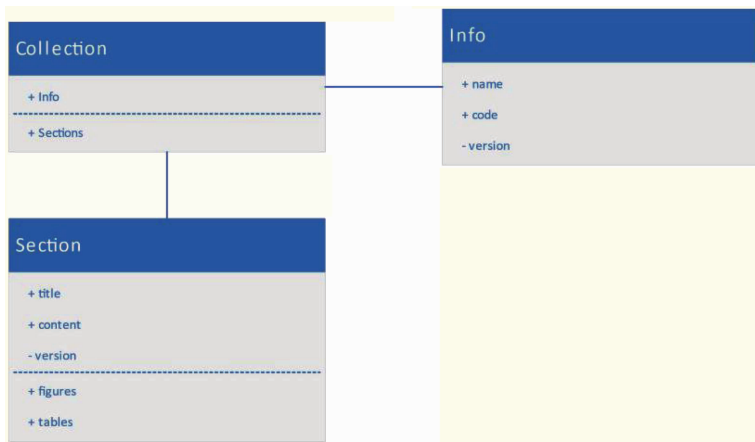


图 3 MongoDB 存储的文档结构图

在存储时,由于采用 JSON 的“键值对”模式,将传统数据表中“行”的概念转换成了更加灵活的文档模型 (Banker, 2012)。数据库可扩展性方面,当新增一类数据时只需相应地新增增加一个新的条目文档即可,这种更自由的存储模式,相较于传统的结构化数据库系统管理,大大降低了维护成本。

该存储方式有如下 3 个优点:①每个区县的 Collection 对应一个独立的综合地情文档,方便查询与管理;②不固定每个文档的条目内容,可根据实际情况灵活地扩展条目内容,满足各省级用户的个性化存储需求;③每个条目具有版本标识,可针对某一条目进行单独的更新与版本管理。

2.2 前后端分离

传统的开发模式将前后端代码耦合在一起,造成代码的可读性及可扩展性不高,给后期

的维护和扩展增加了难度,增加了维护和扩展成本(王建等,2020),难以满足不同用户对于综合地情数据库的个性化使用需求。前后端分离的核心思想是前端调用后端的 RESTFUL API 接口,通过 JSON 数据进行交互,从而将数据整合到前端的业务流程和数据展示中(孙一笑等,2019)。这种架构已经成为互联网项目开发的业界标准使用方式,本文通过 Nginx+WSGI 的方式有效地对前端和后端的开发进行解耦,所有业务逻辑处理均通过后端实现,确保系统的安全性(王亚楠等,2014;肖明魁,2015)。并且,前后端分离会为以后的大型分布式架构、弹性计算架构、微服务架构、多端化服务(各种客户端,如浏览器、桌面、安卓、IOS 等)夯实基础。

通过开发一个综合地情数据服务作为独立的后台,实现综合地情文档在数据库中的存储、查询等具体业务功能,使得各省份都可以利用统一平台维护自己的数据,也可以查询到全国其他省份的信息,实现了地情信息的集中共享与属地维护。利用模式自由存储技术可以将地情数据条目和内容在满足规范的基础上,根据自身的需要进行自由扩展,满足各省份的个性化需求,方便数据直接整合进各自的业务系统当中。由于后台 REST 服务无状态具有客户端无关性的特性,从而可以实现一个数据服务后台、多种前端的应用模式。如利用本文的后台系统服务,山东将其整合到地震应急信息发布系统中,实现震后基本信息的微信发布;青海将其整合进地震应急快速评估系统中,实现了在地震发生时自动查询震中所在区域的基础信息,并在震情基本情况的报告中自动产出。

另外,利用 MongoDB 的分布式集群数据库部署可以动态扩展节点,避免单点故障给数据带来丢失的可能性,提高数据的访问量的性能(申德荣等,2013)。利用前后端分离方式实现的后台服务无状态的接口处理方式,搭载负载均衡配置,使后台服务可以根据访问量的提高,通过扩展服务节点的方式提高服务能力,基于云平台进行地震风险评估综合地情数据服务,满足系统高并发的需求。

综上,前后端分离的方式有如下 3 个优点:①实现全国统一的数据服务接口和平台,免除了数据分发和版本管理的混乱局面;②一个后台、多种前端的开发方式,方便各省份将数据直接整合进各自的系统当中,提高数据的利用率;③通过云端部署,使得服务具有高并发能力,满足多应用同时访问的压力需求。

3 功能实现

3.1 后台接口

本文使用 Python 语言,利用 Falcon 框架构建基于综合地情数据库的 Web Service 系统。Falcon 是一个简约的 WSGI 库,用于构建快速的 Web API 和应用程序后端。在构建 HTTP API 时,其他框架会带来大量依赖和不必要的抽象模型,而 Falcon 具有 HTTP 和 REST 架构风格简洁设计的优点,可以用于快速构建后台接口框架。

实现的接口如表 2 所示。其中接口 1 提供用户身份验证功能,验证通过后给出 token,通过验证 token 的有效性,从而验证接口 2~4 的调用权限。接口 2 为通过 GET 方法列出当前数据库中所有的综合地情文档列表。接口 3 通过 POST 方法上传一个 Word 形式的综合地情文档,系统可自动根据文档接口分析出文档包含的各个条目,将更新日期作为版本号存储到数据库中。接口 4 通过 GET 方法查询某一区县的综合地情文档内容,可通过传入 Section

表 2 后台重要接口实现

序号	接口方法	接口名称	接口功能
1	GET	/API/token	身份验证接口
2	GET	/API/annal	查询所有综合地情文档列表
3	POST	/API/annal	上传综合地情文档
4	GET	/API/annal/{code}	查询某一区县综合地情文档

参数调取某个单一条目内容,也可通过传入 version 查询指定版本的数据。如果不指定 version 参数,接口自动返回最新的条目版本。

接口 3 和接口 4 为整个系统的核心功能,分别承担了综合地情数据上传和查询的关键业务逻辑。采用该接口可以方便地对 MongoDB 中的综合地情数据库进行文档的存储和修正,同时,系统设计的接口可供后续系统继续发开应用(de Villemereuil,2012)。

3.1.1 身份验证接口

由于后台服务需要在全国地震系统中进行应用,为了确保接口调用符合权限,防止非法访问或修改资源,需要根据用户身份换取资源访问的令牌,并使用令牌执行后续的接口调用操作。通过令牌携带的信息包括用户身份标识、时间戳和签名等内容。采用令牌方式的优点为服务端无状态化,可适配分布式的后台服务系统,并支持移动端设备的安全和可跨区域访问,其身份验证流程为:

- (1)前端系统使用用户名和密码请求验证;
- (2)后台服务收到请求后,验证用户身份,并签发 token 发给前端;
- (3)前端系统收到 token 后,即可携带 token 向后台服务请求资源;
- (4)后台服务收到请求后,验证 token 的有效性,识别调用接口的用户身份,并根据用户身份判断其操作权限,验证通过后,即可返回请求数据(图 4)。

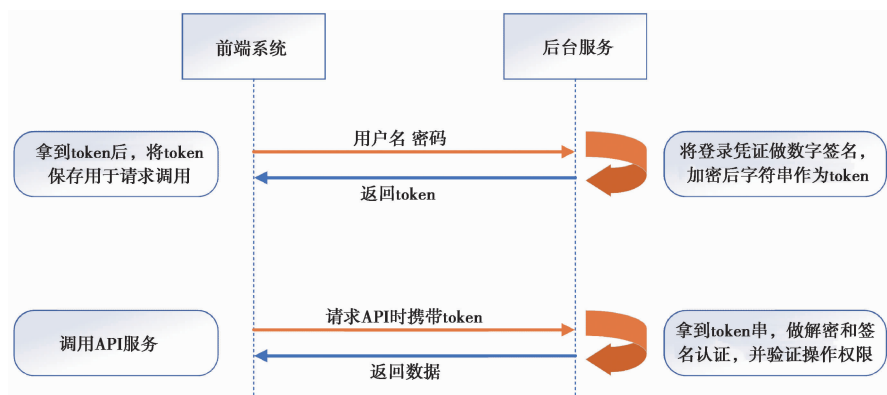


图 4 token 身份验证流程

3.1.2 综合地情文档上传接口

各省级用户可以通过综合地情文档上传接口维护各自的地情文档数据,服务接受 Word 格式的综合地情文档。系统通过文件名解析文档的区划代码、区划名称和版本信息,利用

docx 模块解析文件的具体内容,通过分析文档中各条目内容解析出该条目所包含的文字、图片和表格等信息,并在通过内容校验后将文档内容入库。流程如图 5 所示。

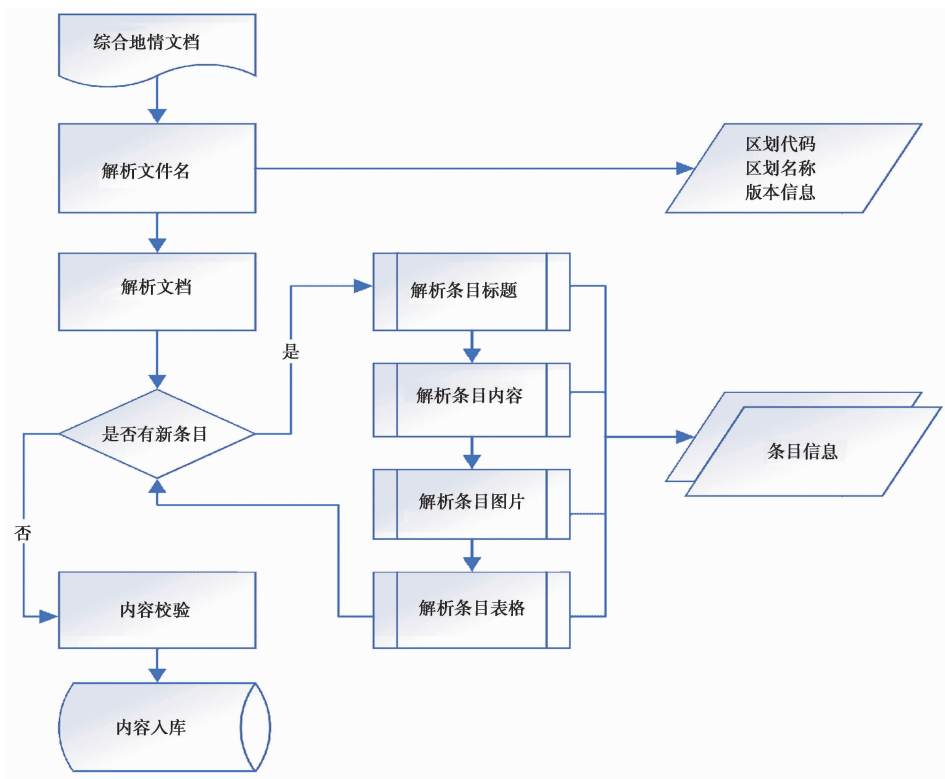


图 5 文档解析流程

3.1.3 综合地情文档查询接口

用户可通过查询接口获取文档的内容列表,并通过传入区划代码查询指定的文档,系统默认会返回最新版本的内容,也可通过传入条目信息和内容信息查询指定条目的文字、图片或表格信息。在系统开发应用时,可以在前端结合 GIS 形式的全国区划数据,通过空间查询、属性查询等可视化检索方式,获取查询区域的区划代码,将区划代码传入后台,实现便捷、GIS 可视化的数据检索和展示。

3.2 前端展示

前后端分离架构实现了一个后台、多种前端,并使前端可以提供多种应用。图 6 为目前已建成的地震风险评估展示平台。

通过后台接口的调用对原有固定的 Word 形式的综合地情文档进行拆解,针对不同部分的内容进行更为直观地展示。将传统形式单一的文件查询方式转变为通过接口进行有针对性的查询方式,并可将文档中的表格内容变成柱状图、折线图和环状图等更直观形象的方式,可以通过交互的方式更直观地查询和了解当地的具体情况。

此外,该系统还可查询某一条目的原始信息,将原有的文字、图片和表格还原集中展示,方便全面了解这一地区的详细数据,如图 7 所示。

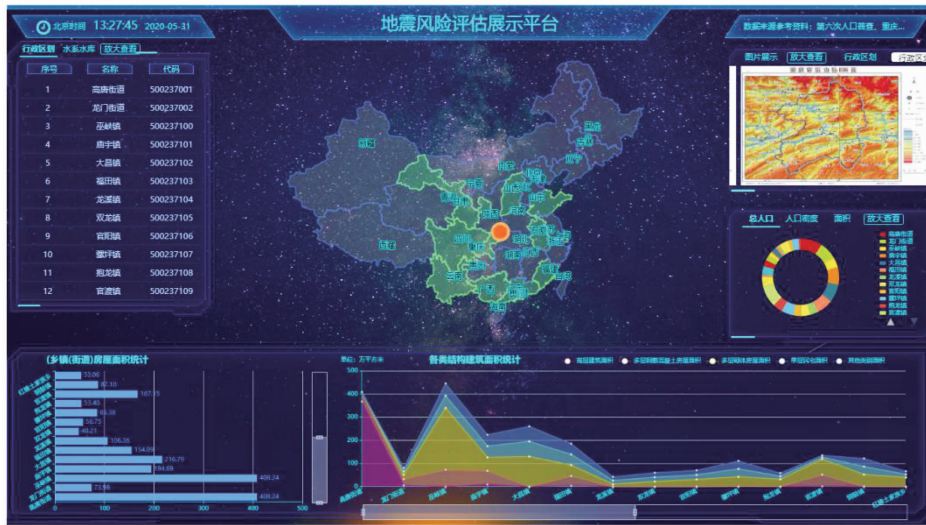


图6 地震风险评估展示平台界面

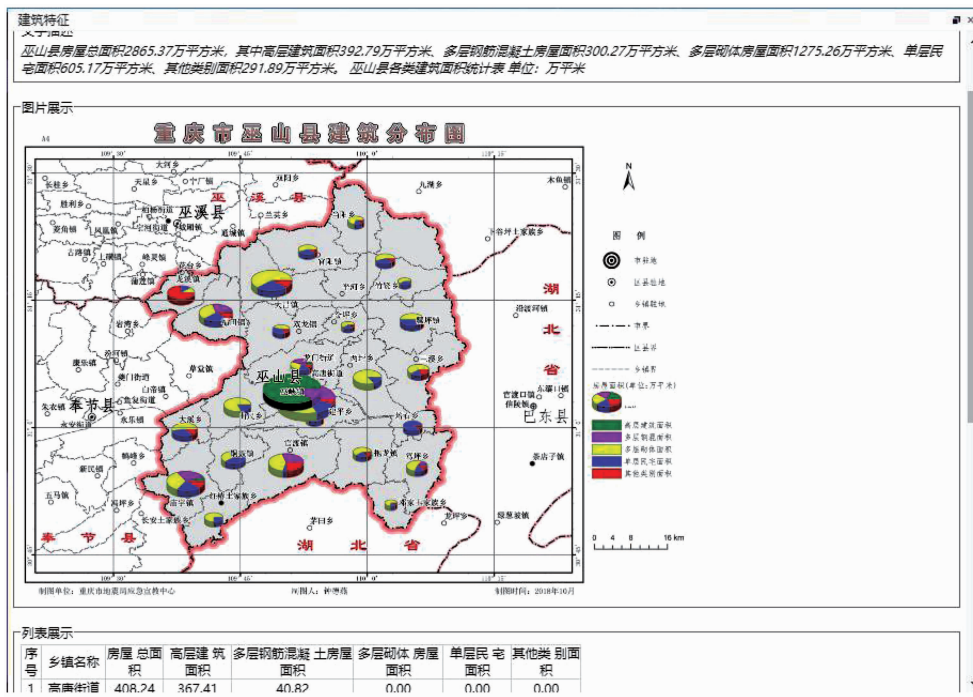


图7 综合地情条目内容展示界面

3.3 部署平台

该系统部署于中国地震台网中心云平台。云计算平台通过虚拟化等技术,根据上层应用达到资源共享、快速满足业务等需求,可以有效节约资源、提高用户服务质量(郭凯等, 2020)。其中,MongoDB 采用共用服务平台提供的数据库存储服务,后台接口使用云平台提

供的 uWSGI 和负载均衡服务,前端展示平台使用虚拟主机服务。通过云平台的服务能力,使系统具有弹性伸缩、资源共享的特点,从而提高系统的资源利用率和服务的可靠性。

4 结语

由于综合地情数据内容形式表达的复杂性和文档条目的灵活伸缩性,使其内容查询和应用服务成为亟需解决的问题。本文充分利用了 MongoDB 的非结构化文档存储的优点,提出了适用于动态条目的地震风险评估展示平台的总体设计、关键技术和功能实现,利用前后端分离技术,最大化地提高了综合地情数据库的服务能力,利用统一后台服务可将数据集成到多种应用,满足数据利用和展示的多样化需求。由于目前系统服务资源是以虚拟主机为单位进行分配,在分配资源上存在一定的浪费,为了更好地提高系统利用率和发挥云平台的效能,下一步可使用容器技术进行升级部署,提高云平台资源的利用率。

参考文献

- 郭凯,姜立新,刘晓雨,等,2020. 地震异构云管理平台的技术架构设计. 中国地震,36(1):153~161.
- 雷德龙,郭殿升,陈崇成,等,2014. 基于 MongoDB 的矢量空间数据云存储与处理系统. 地球信息科学,16(4):507~516.
- 梁芳,2013. 城市与减灾. 地震应急基础数据库的建设和发展. 城市与减灾,(6):28~30.
- 申德荣,于戈,王习特,等,2013. 支持大数据管理的 NoSQL 系统研究综述. 软件学报,24(8):1786~1803.
- 孙一笑,张玉军,孙宇成,等,2019. 基于 WebAPI 前后端完全分离的软件开发模式. 信息与电脑,(6):96~97.
- 王建,罗政,张希,等,2020. Web 项目前后端分离的设计与实现. 软件工程,23(4):22~24.
- 王亚楠,吴华瑞,黄锋,2014. 高并发 Web 应用系统的性能优化分析与研究. 计算机工程与设计,35(8):2976~2981.
- 肖明魁,2015. 基于 Nginx 负载均衡技术初探. 科技展望,25(36):125.
- 谢江丽,李帅,姚远,2019. 地震应急数据库中人口数据预测——以乌鲁木齐市为例. 中国地震,35(2):389~398.
- 郑通彦,文鑫涛,李华玥,等,2021. 地震应急信息速报视图自动发布系统的设计与实现. 中国地震,37(1):206~215.
- Banker K,2011. MongoDB in Action. Greenwich:Manning Publications Co.,312.
- Banker K,2012. MongoDB 实战. 丁雪丰,译. 北京:人民邮电出版社,10.
- de Villemereuil P, Wells J A, Edwards R D, et al, 2012. Bayesian models for comparative analysis integrating phylogenetic uncertainty. BMC Evol Biol,12(1):102.

MongoDB-based Design and Implementation of Integrated Situation Data Service Platform of Earthquake Risk Assessment

Yang Lichen¹⁾ Shuai Xianghua²⁾ Dong Xiang³⁾ Liu Qin²⁾ Li Jigeng²⁾

1) QingHai Earthquake Agency, Xining 810001, China

2) China Earthquake Networks Center, Beijing 100045, China

3) Shandong Earthquake Agency, Jinan 250014, China

Abstract This paper we introduced the integrated situation data service platform of earthquake risk assessment based on MongoDB. We take flexible document structure storage and query as the goal, use Webservice technology and frontend-backend separation development mode, carry out the system design, key technology and function realization of the system. The system can upload, edit and query the integrated situation data. Through the establishment of a national unified integrated situation data service platform for earthquake risk assessment, multiple applications is realized, and service ability of national integrated situation data are improved.

Keywords: Situation database; Schema-free; Frontend-backend separation; MongoDB