引文格式：闫晗辉,蒋喆,张明达,等. 海量多源遥感影像元数据存储与分析应用方法[J]. 地理空间信息, 2025, 23(10):114-119. doi:10. 3969/j.issn. 1672-4623.2025.10.025

海量多源遥感影像元数据存储与分析应用方法

闫晗辉[1]，蒋　喆[2*]，张明达[1]，郭治宏[1]
（1. 湖北大学资源环境学院，湖北 武汉 430062；2. 广西水利电力职业技术学院，广西 南宁 530023）

摘　要：为实现海量多源遥感影像元数据的统一组织和高效管理，基于NASA UMM（统一元数据模型）和数据仓库星型模型，建立了适用于快速检索和分析的遥感影像元数据统一表达方式，提出了元数据集群化存储方法和基于元数据的遥感影像空间覆盖程度分析方法，并对比分析了关系型数据库PostGIS集群与非关系型数据库MongoDB集群在功能、扩展性和性能等方面的优劣。研究结果表明，MongoDB集群在扩展性和性能上更具优势，能够实现多源遥感影像快速混合检索和时空覆盖程度分析。研究成果有利于辅助大范围、长时序的遥感数据查询。

关键词：影像元数据；分布式存储；影像检索；元数据分析
中图分类号：P237　　**文献标志码：**B　　**文章编号：**1672-4623（2025）10-0114-06

Application Methods for Metadata Storage and Analysis of Massive Multi-source Remote Sensing Images

YAN Hanhui[1], JIANG Zhe[2], ZHANG Mingda[1], GUO Zhihong[1]
(1. Faculty of Resources and Environmental Science, Hubei University, Wuhan 430062, China;
2. Guangxi College of Water Resources and Electric Power, Nanning 530023, China)

Abstract: To achieve unified organization and efficient management of massive multi-source remote sensing image metadata, based on NASA's unified metadata model and the star schema of data warehouses, we established a unified expression method for remote sensing image metadata for rapid retrieval and analysis, proposed a metadata clustering storage method and a spatial coverage analysis method for remote sensing images based on metadata, and conducted a comparative analysis between the relational database PostGIS cluster and the non-relational database MongoDB cluster in terms of functionality, scalability, and performance. The results indicate that MongoDB cluster has advantages in scalability and performance, enabling rapid hybrid retrieval of multi-source remote sensing images and spatio-temporal coverage analysis. The research outcomes are beneficial for large-scale, long-term remote sensing data query.

Key words: image metadata, distributed storage, image retrieval, metadata analysis

目前，多源遥感影像元数据的统一管理可为用户提供一致的数据检索入口，支撑多源元数据联合分析，但仍存在如下问题：①不同平台提供的遥感影像元数据信息各异，元数据模型和接口不统一，难以组合检索；②传统的关系型数据库存储影像元数据，往往会遇到性能上的瓶颈，导致查询效率低，并且横向扩展能力差[1-2]，难以实现海量元数据场景下的高可用和易扩展。

目前，在元数据模型方面，已有一些元数据标准来规范多源异构元数据的表达[3-8]。但针对海量多源异构的遥感影像元数据的分布式储存方案以及使用元数据进行时空覆盖程度的分析与应用的研究仍然有限。

本文聚焦海量遥感元数据联合检索与分析应用，基于NASA UMM（统一元数据模型）和数据仓库星型模型，建立了适用于快速检索和分析的遥感影像元数据统一表达模型；提出了元数据集群化存储方法，借助MongoDB分片集群储存海量元数据，并实现了遥感元数据的组合条件快速检索和影像的覆盖程度分析。

1　元数据统一组织模型

通用、标准的元数据模型是多源、异构的遥感元数据的统一存储与检索的基础。UMM是NASA提出的用于遥感影像元数据组织的规范模型，它定义了统一的元数据字段和格式，来统一不同传感器和任务的遥感元数据。本文选用了其中的UMM-G（UMM-Gran-

项目来源：湖北省自然科学基金面上项目（2022CFB119）。
第一作者简介：闫晗辉（2001—），硕士，主要从事空间数据组织管理工作，E-mail：yanhanhui2001a@163.com。
通信作者：蒋喆（1979—），硕士，副教授，主要从事测绘数字遥感专业教学与研究工作，E-mail：274748468@qq.com。

ule）和 UMM-C（UMM-Collection）模型规范，并结合数据仓库星型模型构建了一套表达遥感影像数据及其集合数据的元数据模型。如图 1 所示。

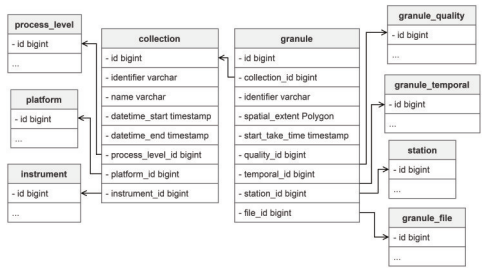


图 1　遥感元数据模型类图

2　元数据集群化存储设计

截至 2024 年 7 月 1 日，全部的高分、Landsat 和 Sentinel 系列平台可获取的元数据总量已达到 1.2 亿条左右，数据量非常庞大，若采用传统单节点关系型数据库储存全部的元数据，会导致查询效率显著下降。这是因为关系型数据库大多采用 B+树索引，当数据量达到一定的阈值时，B+树深度的增加会显著增加磁盘 I/O 负载。除了海量数据导致的性能问题，单节点关系型数据库也难以保证系统的可扩展性与高可用性[9]。

水平分片可以有效地缓解此类问题，水平分片是一种将大量数据分散存储至多个数据库或表的方法，该方法根据一个或多个字段值进行分片。借助多个节点分散储存海量遥感元数据，可以有效缓解单节点的性能问题。

本文分别实现了基于 ShardingSphere-Proxy + PostGIS 的关系型数据库分片存储方案，和基于 MongoDB 分片集群的非关系型数据库分片存储方案，并在功能和性能上对这 2 种方案进行了对比。

2.1　分片策略

为保证分布式存储下每个记录 ID 的唯一性，本文所有的事实表和维度表的主键 ID 均采用雪花 ID 算法生成。

此外，为了避免分片算法差异带来的查询性能影响，本文的关系型数据库分片储存方案与非关系型数据库分片储存方案均采用基于主键 ID 的哈希范围分片算法作为分片方式。在插入或者查询数据时，会首先使用 Hash 函数计算 ID 得到一个长度统一的哈希值，然后根据全部记录的 ID 的哈希值的极值以及集群的节点个数划定多个平均的范围，每个范围对应一个数据库分片节点，最后依据对应记录 ID 的哈希值的所在范围将该记录分配到对应的节点中，或者从对应的节点中查找该记录，该算法的执行机制如图 2 所示。

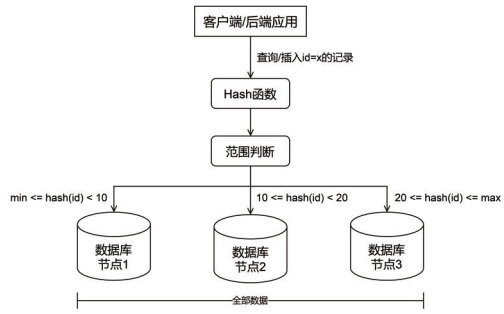


图 2　数据库集群的哈希范围分片算法

以图 2 为例，当前数据库分片集群中有 3 个数据库节点，假设某条数据的 ID 在经过哈希函数计算后得到的值 15，介于 10~20 之间，那么这条数据就会被分配至第 2 个节点中。通过计算 ID 的哈希值并进行分片，能够打乱原本的顺序，使得所有数据能够平均分散在不同的分片节点上。

2.2　分片集群存储方式

2.2.1　基于 PostGIS 的分片集群存储

PostGIS 是一个用于 PostgreSQL 数据库的扩展，为地理信息系统（GIS）数据提供了存储和查询功能，并提供了上百个能够对地理数据进行检索或者实时处理的 SQL 聚集函数，能够覆盖绝大多数对空间数据的存储和检索场景[10]。

然而，PostGIS 本身不支持分片集群部署，需要借助 ShardingSphere-Proxy 实现数据分片。ShardingSphere-Proxy 是一款开源分布式数据库中间件，能够将数据分散储存至关系型数据库集群，以实现数据的水平分片。

2.2.2　基于 MongoDB 分片集群存储

虽然通过 ShardingSphere-Proxy 配合 PostGIS 集群可以实现数据的水平分片，但是其可扩展性仍然难以保证。例如当数据量持续增加、需要新增 PostGIS 节点时，则需要根据已有的数据 ID 哈希的极值，重新计算每个分片对应的范围，并对已有的数据进行重新分片和迁移操作，这增加了数据维护的成本；此外由于 PostGIS 本身不支持数据分片，额外引入 ShardingSphere-Proxy 中间件也增加了系统整体的复杂程度。

针对 PostGIS 分片集群在可扩展性方面的问题，采用原生支持分布式的 NoSQL（Not Only SQL，非关系型数据库）可以解决该问题。MongoDB 是一款非关系型文档数据库，采用 BSON（Binary JSON）格式储存数据，且自身支持副本集集群和分片集群的部署方式[11-13]。

比起 PostGIS 外键关联的表结构，MongoDB 的内嵌文档结构可以避免查询时频繁的表连接操作，减少

数据库对磁盘的I/O操作，进一步提高查询效率。

2.3 PostGIS分片集群和MongoDB分片集群对比

2.3.1 数据库功能特性对比分析

PostGIS作为关系型数据库，对地理数据的储存和查询支持非常丰富，虽然可以借助ShardingSphere-Proxy实现数据的水平分片以解决性能瓶颈问题，但是其可扩展性仍然难以保证。

MongoDB作为非关系型数据库，除了性能占优势之外，还原生支持数据的分布式分片存储，并且支持动态添加分片节点并完成数据的迁移和平衡操作[14]，可扩展性强。虽然MongoDB也支持GeoJSON格式地理数据的存储和检索，但是其地理处理功能比起PostGIS仍有所欠缺。

综上所述，对于存储大量包含地理信息并需要按地理条件进行检索的数据（例如遥感影像元数据），PostGIS分片方案和MongoDB分片方案都是可行的选择。然而，仍需从地理数据支持的程度和查询效率等方面进行权衡，以选择最合适的解决方案。

表1 PostGIS和MongoDB特性比较

数据库	类型	性能	地理支持	集群支持
PostGIS	关系型	海量数据下存在瓶颈	非常丰富	本身不支持，需引入中间件
MongoDB	非关系型-文档	高性能	有限的地理支持	原生支持集群，可扩展性好

2.3.2 查询性能测试与对比

为了对比不同数据库在查询性能方面的差异，本文在同样的条件下对PostGIS集群和MongoDB集群的查询速度进行了测试，数据库集群均使用Docker容器方式部署，每个分片集群均由3个数据节点构成，机器配置为：CPU：Ryzen 5 7600X，内存：64 GB 5 600 MHz，硬盘：NVMe SSD 1TB PCIe 4.0×4；同时还选用了全部高分元数据（截至2024年5月31日）和Landsat元数据（截至2024年7月3日）解析并分别导入至2个数据库集群，共计24 928 834条记录。目前系统支持组合下列条件进行查询：集合、云量、时间范围和空间范围条件，因此对2个数据库的相关的字段均建立索引覆盖查询条件。

由于PostGIS支持SP-GiST和GiST2种空间索引，且不同的空间索引在查询性能上也有所差异，因此本文在多个相同的查询条件下，同时对比在使用PostGIS SP-GiST索引、PostGIS GiST索引和MongoDB时的查询时间消耗。

首先以时间跨度为主要变量进行查询性能比较，在固定集合条件为Landsat8 Level 1和Level 2集合、空间范围为湖北省边界（包括相交）、云量条件为0~90%的情况下，逐步增加查询时间跨度进行查询，实验结果如图3所示。

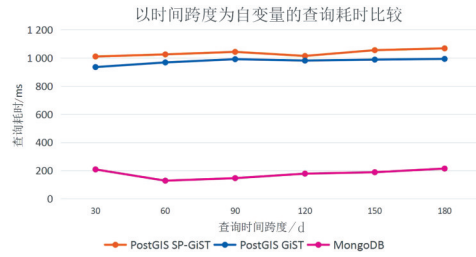


图3 以时间跨度为自变量的查询耗时比较

用户查询时选择的影像来源集合的数量也会对查询速度造成影响，因此实验还将在固定空间范围条件为湖北省边界（包括相交）、时间范围为2024-01-01—2024-06-30、云量条件为0~90%的情况下，依次比较选择1~10个影像集合进行查询的速度，实验结果如图4所示。

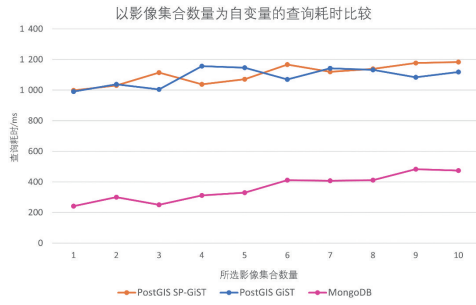


图4 以影像集合数量为自变量的查询耗时比较

为了比较不同的索引以及数据库在不同空间范围的情况下查询性能，实验还将在固定集合条件为Landsat8 Level1和Level2集合、时间范围为2024-01-01—2024-06-30、云量条件为0~90%的情况下，依次比较市级范围（宜昌市）、省级范围（湖北省）和全国级别范围（全中国范围）的查询速度，实验结果如表2所示。

表2 不同空间范围大小的查询耗时比较

空间范围	PostGIS SP-GiST耗时/ms	PostGIS GiST耗时/ms	MongoDB耗时/ms
市级	1 009	992	152
省级	1 005	1 002	163
国家级	2 210	762	1 076

通过实验可以得出：

1）在其他条件相同、只改变查询时间跨度的情况下，PostGIS使用SP-GiST索引的单次查询耗时在1 010~1 068 ms之间，平均耗时为1 035 ms；PostGIS使用GiST索引的单次查询耗时在935~993 ms之间，

平均耗时为976 ms左右；MongoDB的单次查询耗时在128~214 ms之间，平均查询耗时为177 ms左右。此外，逐步增加时间跨度对单次查询的速度影响非常小。

2）在其他条件相同、只改变选择的影像来源集合数量的情况下，PostGIS使用SP-GiST索引的单次查询耗时在998~1 184 ms之间，平均耗时为1 103.8 ms左右；PostGIS使用GiST索引的单次查询耗时在991~1 157 ms之间，平均耗时为1 088.4 ms左右；MongoDB的单次查询耗时在242~483 ms之间，平均查询耗时为362.1 ms左右。此外，选择的影像来源集合的数量越多，单次查询的耗时越长，且对MongoDB的查询耗时影响更加显著。

3）在其他条件相同、只改变查询的空间范围大小的情况下，PostGIS使用GiST索引和使用SP-GiST索引耗时基本相当，而MongoDB耗时总体偏小。

可见，在储存和检索海量数据的情况下，PostGIS的GiST索引的查询速度略优于SP-GiST索引，但是两者的查询耗时都显著高于MongoDB的查询耗时，MongoDB比PostGIS有显著的性能提升。

3　元数据的检索与分析方法

遥感影像能否覆盖感兴趣的空间和时间范围，是用户检索数据时关心的首要问题。在大范围、长时序应用场景，单一来源的影像产品往往无法直接覆盖研究区，需要联合多种遥感影像产品[15]。针对此，本文提出了基于组合条件的影像检索和遥感影像覆盖程度分析方法。

3.1　元数据的规范化查询

无论是高级检索功能，还是影像的覆盖程度分析功能，都需要根据用户提供的组合条件来完成元数据检索和查询工作。为了提高功能的重用性，建立了一个组合条件模型表示所有查询条件参数的总和，其中包含了所有表示查询条件的字段，是元数据检索API的主要参数，查询条件模型如图5所示。

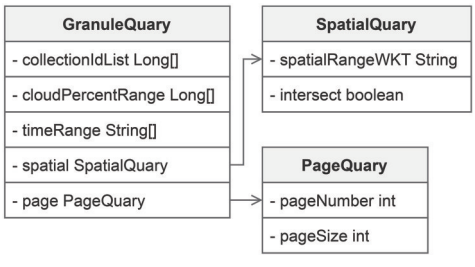


图5　查询条件模型UML图

图6表示了通过建造者模式的相关接口，将用户的查询参数构建为对应的MongoDB查询语句。

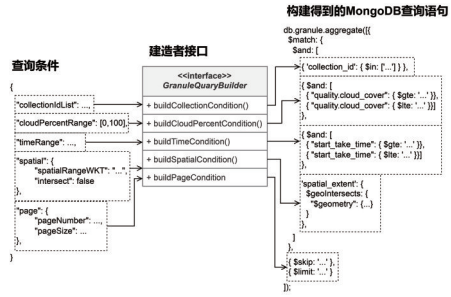


图6　基于建造者模式思想的查询语句构建

通过建造者模式接口将查询语句或者查询对象的构建过程规范化，只需要使用同一套构建流程，就可以构建出适用于MongoDB的查询对象，此外基于上述的构建流程，同样还可以实现构建出PostGIS查询语句乃至其他数据库的查询语句，将复杂的构建过程和原始参数、结果查询语句解耦合；当条件字段增加时，只需增加对应的建造者接口方法并实现，提高系统的可维护性和可扩展性。

3.2　影像的时空覆盖程度分析

针对大尺度、长时序的遥感应用场景，跨卫星平台的数据检索是前提，提出多源遥感元数据的时空覆盖程度分析方法，用户通过自定义的组合条件，即可获取某一个或者几个集合的影像在某一段时间内、对兴趣区域的覆盖程度。

通过用户提交的自定义条件，可以实现查询所有符合条件的影像记录，并按照影像时间升序排列，将每一幅影像覆盖到研究区域直到全部覆盖完成，若有两幅影像重叠程度过高，则会舍弃其中一幅影像，最后计算所有影像对研究区域的空间覆盖程度。

在计算空间覆盖程度时，将会通过影像与研究区域重合的部分进行计算，并去除研究区内影像重合的部分，图7展示了2幅影像和研究区的重合情况示例。

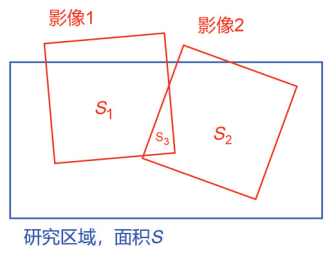


图7　影像和研究区域重合面积计算

假设研究区域的面积为 S，2幅影像与研究区的重合部分面积分别为 S_1 和 S_2，2幅影像相互重合且位于研究区域内的部分面积为 S_3，那么就可以根据式（1）得出这2幅影像对研究区域的空间覆盖程度 C 的值，使用百分比表示。

$$C=\frac{S_1+S_2-S_3}{S}\times 100\% \tag{1}$$

在计算空间覆盖程度的同时，还会根据查询结果的时间信息，计算影像在查询条件指定时间段内的覆盖程度。具体来说，通过已查询到的影像时间信息，精确计算给定条件时间段内哪些子时间段包含影像数据，精度达到天数级别。

最终，查询条件对应的影像数据在研究区的空间覆盖程度及其在查询时间段内的时间覆盖程度将一并返回，并在用户界面上进行可视化展示。通过结合时间和空间这2个维度的覆盖程度，可以为用户提供更加直接、精确的参考信息。

4 原型系统设计与实现

4.1 系统总体架构

系统总体采用的是B/S的前后端分离架构，其中前端负责发起获取数据的请求，并实现数据的可视化，后端则负责从数据库中检索数据并实现一些业务功能，系统整体的架构图如图8所示。

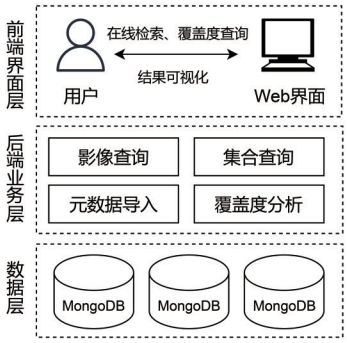


图8 系统整体架构图

数据层主要通过MongoDB分片集群储存全部遥感影像元数据；后端业务层调用数据库，负责将数据插入到数据库或者从数据库检索数据；前端则通过HTTP请求调用后端，获取到相应的结果并可视化。

4.2 原型系统实现

原型系统的后端服务基于Spring Boot框架开发，通过MongoDB Spring Boot Starter实现对MongoDB访问的支持。在数据导入方面，后端服务支持从本地批量导入JSON格式的高分和Sentinel元数据以及CSV格式的Landsat元数据，并统一整理为UMM模型，最后批量导入至数据库，为了提高导入数据的性能，使用了虚拟线程（Virtual Threads）技术来实现数据的批量导入功能，在I/O密集型的任务场景中虚拟线程有着更高的性能[16]。在业务功能方面，后端系统提供了组合条件检索API，用户可以根据影像数据的维度信息（如时间、空间、云量等）进行检索。此外，还可以通过组合条件完成影像的覆盖程度分析工作。

前端页面采用Vue 3 + Cesium框架进行开发，用户交互式设定查询条件，检索到的影像能够在三维的场景下可视化其覆盖范围或者缩略图，同时用户还能得到一个查询的影像结果列表，通过列表用户可以查看每一幅影像的全部详细信息，包括影像的时间维度、质量维度、站点等信息。

本系统实现了三维场景下的遥感影像元数据检索、影像缩略图浏览以及影像覆盖度分析功能。图9展示了该系统通过多种组合条件查询高分2号和Landsat 8 Level 1集合组合，覆盖湖北省区域，时间为2024年3月到4月份的影像记录。通过选中查询结果列表的影像可以将影像的覆盖范围可视化在三维地球视图中，同时，用户也可以查看其中一幅影像更多详细的元数据信息。

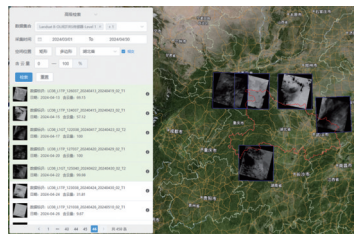


图9 组合条件的影像检索（影像来自天地图）

图10展示了系统通过多种组合条件查询高分2号和Landsat8 Level 1集合组合，在2024年5月对湖北省区域的时空覆盖程度及其可视化效果。

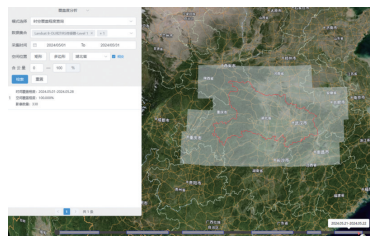


图10 高分2号和Landsat 8 Level 1集合组合在2024年5月对湖北省区域的时空覆盖程度（影像来自天地图）

在图10中，可通过左侧结果列表查看具体的时空覆盖程度数值，在地图上可查看空间范围的覆盖程度，此外下方的时间条也显示了时间覆盖程度，即查询时间段内哪些子时间段是有影像记录的，整个时间条代表给定的查询条件时间段，而白色部分则是该时间段内包含影像数据的子时间段。

5 结 语

本文聚焦于海量遥感影像元数据的储存和应用，首先基于UMM元数据模型和数据仓库星型模型对不同来源的元数据进行了统一的表达；然后搭建了Post-

GIS与MongoDB分片集群来完成数据的储存工作，并对2种储存方案从功能方面和性能方面进行了比较与测试，证明了MongoDB比PostGIS有着更高的查询性能；最后实现了影像高级检索功能和影像的时空覆盖程度分析功能，使用户能够更加灵活、有效地利用元数据。本文研究成果有助于实现复杂场景下海量遥感影像元数据的快速查询。

参考文献

[1] 张扬. 基于NoSQL的遥感影像存储系统的研究[D]. 北京:中国电子科技集团公司电子科学研究院,2019

[2] 朱建坤. 海量多源异构遥感影像的高效管理与调度方法研究[D]. 淮南:安徽理工大学,2023

[3] NASA. Unified Metadata Model. [EB/OL].(2021-07-15)[2024-07-16]. https://www. earthdata. nasa. gov/unified-metadata-model-umm

[4] 王志宝,赵亮,马轩,等. 大规模卫星遥感数据资源目录技术研究[J]. 计算机与数字工程,2023,51(10):2 293-2 299

[5] 秦强,王晏民,黄明. 基于MongoDB的海量遥感影像大数据存储[J]. 北京建筑大学学报,2015,31(1):62-66

[6] 王慧民. 福建省多源遥感影像云服务平台设计与实现[J]. 地理空间信息,2023,21(3):102-105

[7] Huang K, Li G, Wang J. Rapid Retrieval Strategy for Massive Remote Sensing Metadata Based on GeoHash Coding[J]. Remote Sensing Letters,2018,9(11):1 070-1 078

[8] Wang S, Li G, Yu W, et al. Requirement-driven Remote Sensing Metadata Planning and Online Acquisition Method for Large-scale Heterogeneous Data[J]. Geo-Spatial Information Science,2022,25(2):169-181

[9] Apache Software Foundation. Apache Sharding Sphere-features-sharding. [EB/OL]. (2022-08-09) [2024-07-17]. https://shardingsphere. apache. org/document/current/en/features/sharding/

[10] PostGIS, OSGeo,PostGIS. [EB/OL]. (2024-07-06)[2024-07-21]. https://postgis. net/

[11] Cornelia G, Robert G, Pecherle G, et al. A Comparative Study: MongoDB vs. MySQL[C]//13th International Conference on Engineering of Modern Electric Systems (EMES),2015

[12] Sethi S, Panda S. SQL or NoSQL-practical Aspect and Rational Behind Choosing Data Stores[J]. Journal of Computer and Communications, 2024,12(8):155-174

[13] Mongo D B. MongoDB 2dsphere Index.[EB/OL]. (2024-06-28) [2024-07-21]. https://www. mongodb. com/docs/manual/core/indexes/index-types/geospatial/2dsphere/

[14] Mongo D B. Mongo D B Add Shards to Cluster. [EB/OL]. (2024-06-28)[2024-07-21]. https://www. mongodb. com/docs/manual/tutorial/add-shards-to-shard-cluster/

[15] 霍嘉婷,赵展,朱秀丽. 全球地表覆盖数据辅助多源影像融合提取城市不透水面[J]. 测绘通报,2024(2):19-25

[16] Open J D K. JEP 444: Virtual Threads. [EB/OL]. (2024-07-10)[2024-07-23]. https://openjdk. org/jeps/444

（上接第105页）

270-279

[10] 于昊辰,尹登玉. 新质生产力发展与土地资源配置:历史逻辑、理论逻辑与现实逻辑[J]. 中国土地科学,2024,38(5):29-37

[11] 姬冬梅,李淑杰. 农用地整理潜力测算与分级研究:以延边朝鲜族自治州为例[J]. 东北师大学报(自然科学版),2021,53(3):107-114

[12] 向敬伟. 全生命周期视角下流域国土空间生态保护修复研究[J]. 中国土地科学,2024,38(5):21-28

[13] 龚华,仝德,张楚婧,等. 城乡融合视角下的全域土地综合整治模式优化[J]. 规划师,2023,39(12):38-44

[14] 施釉超,金晓斌,韩博,等. 面向资源认知变化的耕地评价研究进展与展望[J]. 中国农业大学学报,2024,29(8):237-250

[15] 郭维红,蒋非非,马静,等. 2000—2020年中国耕地旱改水空间格局演变及其适应性管理[J]. 农业工程学报,2024,40(4):292-303

[16] 刘金花,王雨晴,蒋海明,等. 进出平衡视角下耕地恢复潜力重点区域识别[J]. 农业工程学报,2023,39(12):216-225

[17] 牛文涛,褚珺祎,董安洞. 加快建设农业强国背景下高标准农田建设质量的区域差异及分布动态演进[J]. 宏观质量研究,2024,12(4):86-100

[18] 梁琳,金光敏. 农业强国背景下高标准农田建设提质升级的路径研究:以吉林省为例[J]. 税务与经济,2024,45(4):106-112

[19] 杜建平,邵景安,周春蓉,等. 基于生态适宜度和三角模型的煤矿临时建设用地复垦决策研究[J]. 自然资源学报,2018,33(11):1 872-1 885

[20] 林嘉琪,李宽,柯春鹏,等. 基于TOPSIS的广东省农村建设用地拆旧复垦成效评价[J]. 资源开发与市场,2023,39(5):538-546