IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 May 2024, accepted 20 June 2024, date of publication 28 June 2024, date of current version 23 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3420757

RESEARCH ARTICLE

Enhanced Remote Sensing Monitoring Through a Bimodal Cloud Infrastructure: A Dual-State Cloud Service Approach

YANG KAIJUN[1,2,3], LEI FAN[2,3], WEI JIDE[2,3], AND ZHANG ZHE[2,3]
[1]Institute of Advanced Studies, China University of Geosciences (Wuhan), Wuhan 430078, China
[2]Second Surveying and Mapping Institute of Hunan Province, Changsha 410029, China
[3]Key Laboratory of Natural Resources Monitoring and Supervision in Southern Hilly Region, Ministry of Natural Resources, Changsha 410029, China

Corresponding author: Lei Fan (632085233r@qq.com)

This work was supported in part by the Natural Science Foundation of Hunan Province under Grant 2024JJ8342, and in part by the Research Foundation of the Department of Natural Resources of Hunan Province under Grant 20230155CH.

ABSTRACT This study addresses significant challenges within the field of remote sensing monitoring, including operational inefficiency, data confidentiality concerns, high hardware costs, and issues with data management and distribution. To tackle these problems, we introduce a synergistic remote sensing monitoring framework that leverages a bimodal cloud infrastructure, facilitated by cloud services to provide on-demand resource allocation and efficient management. Our research focuses on designing and developing an integrated operating system that optimizes remote sensing monitoring processes and enhances operational efficiency through the use of a mutual scheduling mechanism and rapid data indexing capabilities. The system is underpinned by a dual-state cloud service mechanism, combining the Memory Cloud (Flash Cloud) known for its high-speed data processing and the Storage Cloud (Persistent Cloud) for long-term data retention. This dual-state approach establishes a multi-level caching system to ensure quick access to frequently requested spatial data. Additionally, a two-tier security system is implemented to safeguard data integrity and confidentiality. Our ‘‘YunYao’’ geographic information service rendering engine, operating on this dual-state cloud platform, demonstrated remarkable performance advantages over mainstream platforms in identical testing environments. Specifically, it outperformed ArcGIS Desktop by over two times, exceeded GeoServer by more than four times, and was over seven times faster than ArcGIS Server in rendering speeds. Experimental and practical applications have shown that our system streamlines routine workflows and enhances work efficiency, making it a critical reference for remote sensing monitoring. Furthermore, a comparative analysis was conducted to quantitatively demonstrate the superior performance of our method in handling large volumes of remote sensing data(Including satellite imagery and UAV imagery). Despite these advancements, the integration of cloud service technology in the field of satellite remote sensing requires further development, particularly regarding the establishment of private clouds and the internal collaborative computing mechanisms within the remote sensing domain. Our research paves the way for future advancements and the eventual full integration of cloud service models into remote sensing monitoring.

INDEX TERMS Cloud service, remote sensing monitoring, bimodal cloud, load balancing.

I. INTRODUCTION

Remote sensing technology is a cornerstone for monitoring and managing land resources, with applications ranging from agricultural surveillance to urban development. Despite the pivotal role it plays, the current landscape of remote sensing

The associate editor coordinating the review of this manuscript and approving it for publication was Nitin Gupta.

monitoring systems is fraught with operational inefficiencies, data security vulnerabilities, escalating hardware costs, and challenges in data management and distribution. These issues not only impede the scalability of monitoring efforts but also pose significant barriers to the adoption of remote sensing in various scientific and practical domains. Our research introduces a transformative approach to these challenges by proposing a synergistic remote sensing monitoring framework that leverages a bimodal cloud infrastructure. This framework is designed to be a significant advancement over the state of the art, offering a dual-state cloud service model that integrates memory and storage clouds to optimize data processing and access speeds. The system's originality lies in its ability to streamline operational workflows, enhance individual efficiency, and provide a robust security framework for data integrity and confidentiality. The primary research question this study addresses is: How can a bimodal cloud infrastructure enhance the efficiency, security, and scalability of remote sensing monitoring systems? To answer this, we have developed a coordinated operating system that integrates advanced scheduling mechanisms, rapid data indexing, and a two-layer security system. Our contributions include the design of a memory cloud for high-speed data processing and a storage cloud for persistent data retention, coupled with a comprehensive platform for collaborative image processing and information extraction. The "beyond state-of-the-art" aspect of our proposed study is multifaceted. Our contributions include:

(1) The design of a memory cloud for high-speed data processing and a storage cloud for persistent data retention, coupled with a comprehensive platform for collaborative image processing and information extraction.

(2) The innovative integration of memory and storage clouds to create a dual-state cloud service mechanism that significantly improves data access and processing speeds.

(3) A two-tier security system that ensures an unprecedented level of data security and integrity, safeguarding data at rest and during transit.

(4) The development and implementation of the "Yun-Yao" geographic information service rendering engine, which demonstrated superior performance over mainstream platforms, offering a quantum leap in rendering speeds that can potentially revolutionize the field of remote sensing. The motivation behind our research is to facilitate a more efficient, secure, and cost-effective remote sensing monitoring framework that not only enhances current capabilities but also paves the way for future advancements in the field. By addressing the identified challenges and providing a robust solution, we aim to contribute to the broader goal of sustainable development and environmental monitoring through the innovative use of cloud technology.

The motivation behind our research is to facilitate a more efficient, secure, and cost-effective remote sensing monitoring framework that not only enhances current capabilities but also paves the way for future advancements in the field. By addressing the identified challenges and providing a robust solution, we aim to contribute to the broader goal of sustainable development and environmental monitoring through the innovative use of cloud technology.

II. RELATED WORK

The integration of cloud computing with remote sensing has been a focal point for enhancing the efficiency and scalability of monitoring systems. Several studies have explored the application of cloud services in the realm of remote sensing, each contributing to the body of knowledge in different ways.

Becker-Reshef et al. [1] introduced the Global Agriculture Monitoring (GLAM) project, emphasizing the utility of coarse-resolution earth observations for global cropland monitoring. This work laid the foundation for leveraging satellite data at a global scale and demonstrated the potential for cloud-assisted data analysis.

Fuhu and Jinnian [2] delved into the technical aspects of converting remote sensing into cloud services, providing valuable insights into the integration of remote sensing technologies with cloud-based services.

Amani et al. [3] conducted a comprehensive review on the application of Google Earth Engine for managing remote sensing big data applications, highlighting the platform's significance in handling large datasets.

Sun et al. [4] proposed an efficient framework for processing remote sensing big data in cloud environments, showcasing the feasibility of cloud-based solutions for managing and analyzing vast amounts of data.

Ferreira et al. [5] described the use of remote sensing images and cloud services on AWS for land use and cover monitoring, further emphasizing the role of cloud platforms in remote sensing applications.

However, these studies have not fully explored the adaptive characteristics of dual-state cloud systems for various types of geographic information data. There is a gap in the literature regarding the comparative analysis of data interaction modes between storage and memory clouds based on geographic information data models.

To address this, our study introduces a novel approach that not only builds upon the existing body of work but also incorporates metaheuristic algorithms to optimize the scheduling and processing of remote sensing data within the cloud infrastructure. Metaheuristic approaches, such as genetic algorithms, particle swarm optimization, and ant colony optimization, are known for their effectiveness in solving complex optimization problems and can be adapted to enhance the efficiency of remote sensing data processing in a cloud environment [1], [2], [3], [4], [5], [6], [7].

This table provides a side-by-side comparison of our proposed study with other significant works in the field. It highlights the originality of our contribution, particularly in the integration of metaheuristic approaches to optimize the dual-state cloud infrastructure for remote sensing data processing.

TABLE 1. Comparative analysis of remote sensing cloud services with metaheuristic approaches.

Study	Focus Area	Cloud Platform	Metaheuristic Integration	Data Processing Optimization	Security Measures
Becker-Reshef et al., 2010	Global cropland monitoring	-	No	Coarse-resolution analysis	-
REN Fuhu and WANG Jinnian, 2012	Cloud service conversion	-	No	Technical research	-
Amani et al., 2020	Big data management	Google Earth Engine	No	Comprehensive review	-
Sun et al., 2019	Efficient data processing	Cloud computing	No	Framework development	-
Ferreira et al., 2020	Land use monitoring	AWS	No	Application of services	-
Our Proposed Study	Bimodal cloud infrastructure	Dual-state Cloud Model	**Yes**	Scheduling and processing	Two-layer security

III. MATERIALS AND METHODS

A. SIMULATION ENVIRONMENT DESCRIPTION

For the simulation of our remote sensing monitoring platform, we have meticulously constructed an environment on the Amazon Web Services (AWS) cloud infrastructure. AWS was chosen for its global reach, scalability, and the ability to provide high-performance computing resources that are essential for handling the complex processing requirements of remote sensing data.

The hardware configuration within AWS for our simulation environment is as follows:

CPU: The simulation utilizes 4 instances of the Intel Xeon E7-4850 v4 processor, with a base frequency of 2.1GHz and a turbo frequency of up to 2.8GHz. Each instance is equipped with 64 cores and 128 threads, enabling concurrent processing of large datasets typical in remote sensing applications.

Memory: The system is supported by 256GB of RAM, which is critical for the efficient handling of large-scale data processing and analysis tasks.

Storage: We have allocated 20TB of high-throughput storage to accommodate the vast influx of data generated by remote sensing technologies. This storage solution is designed to balance speed with capacity to meet the demands of data-intensive workflows.

Graphics Processing: For tasks requiring intense graphical computation, such as image rendering and analysis, we have integrated 4 NVIDIA Tesla V100 GPUs with 16GB of GDDR5 memory each. These GPUs are optimized for handling the parallel processing requirements of remote sensing imagery.

Networking: A 10 Gigabit network interface is employed to ensure rapid data transfer within the AWS cloud environment, which is crucial for real-time processing and analysis of remote sensing data.

To harness the power of AWS, we have utilized a combination of services, including:

EC2 (Elastic Compute Cloud): For the dynamic allocation of compute resources, allowing us to scale our CPU and GPU instances based on demand.

EBS (Elastic Block Store): High-performance block storage for our large datasets, providing consistent and low-latency performance.

VPC (Virtual Private Cloud): To create a secure and isolated network within the AWS environment, ensuring the privacy and integrity of our remote sensing data.

Rationale for AWS Selection

AWS was selected due to its:

Global Presence: Allowing for the simulation of a platform that can operate across different geographical regions.

Service Reliability: Benefiting from AWS's proven track record of service uptime and reliability.

Customizability: The ability to customize compute and storage resources to fit the specific needs of our remote sensing monitoring platform.

Security: AWS provides a robust security framework that aligns with our stringent data protection requirements.

While AWS offers a powerful platform for simulation, it is important to note that the actual performance may vary due to the virtualized nature of cloud resources. Additionally, network latency and data transfer costs are factors that need to be considered when operating on a global scale.

The simulation environment on AWS provides a robust platform for evaluating the performance and scalability of our remote sensing monitoring platform under conditions that closely emulate real-world operational scenarios. The detailed configuration allows for a comprehensive assessment of the system's capabilities, paving the way for future deployment and optimization.

B. SYSTEM OBJECTIVES

The system, underpinned by cloud services, establishes a comprehensive platform within the private cloud service framework. This platform is designed for collaborative image processing, information extraction, result analysis, thematic presentation, and various auxiliary functions related to remote sensing. At its core, a central processing cluster takes charge of processing and analyzing both vector and raster data. It also operates within a dual-state cloud service mechanism accessible to all platform users.The dual-state cloud system consists of two key components: the Memory Cloud (Flash Cloud) and the Storage Cloud (Persistent Cloud).

The Memory Cloud primarily utilizes memory as its storage medium, renowned for its rapid and efficient data read-write capabilities and computational prowess. However, unlike storage devices such as disk arrays and hard drives, the Memory Cloud lacks the same level of data storage persistence. This characteristic results in a less stable data retention

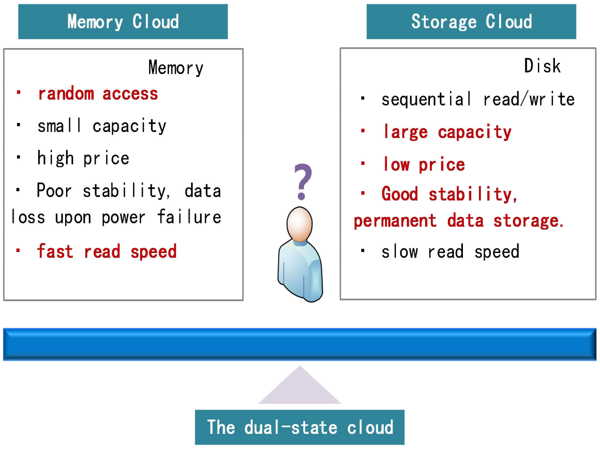

FIGURE 1. Dual-state cloud characteristics.

state, often described as a “flash state” due to its non-steady nature [8], [9], [10].

Figure 1 illustrates the difference between the Memory Cloud and the Storage Cloud that constitute the Bipolar Cloud, providing a detailed comparison of the advantages and disadvantages of the two approaches.

Additionally, the system comprises numerous terminal operation devices available for technicians to schedule and process computing resources on the central processing cluster. Unlike the current environment, terminal devices in the cloud service model only require lightweight devices with browser scheduling capabilities. There is no need to equip each terminal with high-performance or large storage hardware. All terminals access their assigned tasks and submit results by logging in through a browser to their respective accounts. This means that mobile devices (such as smartphones or iPads) can play a role in scheduling computing resources within the private cloud, enabling all devices to share the entire resource pool.

Taking Hunan Province as an example, the system adopts a dual-layer cloud mechanism on the cloud side. All original remote sensing image data is transmitted via the public network from the national primary center in a 1+31 pattern, receiving real-time domestically produced satellite images within the transit scope. The primary center utilizes a commercial cloud service model, employing Alibaba Cloud technology for image management and transmission. Upon receiving the image data pushed by the upper-layer cloud, the Hunan sub-center, as the lower-layer cloud, utilizes a private cloud service mechanism to disseminate computing services across the entire internal network. Within the entire business flow system, the dual-state cloud efficiently cooperates to maintain the stable operation of the computing resource pool.

Figure 2 illustrates the hierarchical architecture of the entire system, divided into the Operational Support Layer, Data Layer, Cloud Services Layer, and User Layer. The periphery also demonstrates the Security Assurance System and Standard Specification System.

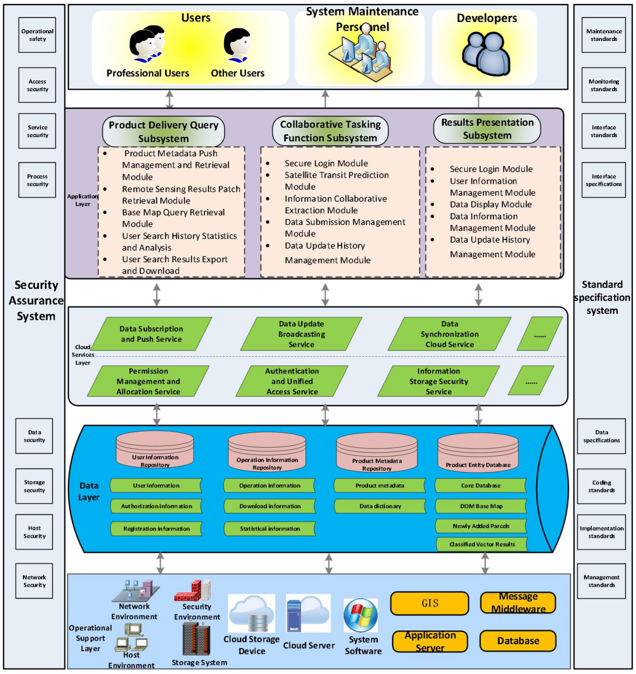

FIGURE 2. System overall architecture.

C. DATABASE DESIGN

The platform utilizes MongoDB as the foundational database, which is a collection-oriented NoSQL non-document-oriented database. MongoDB stores data in BSON (Binary Serialized Document Notation) formatted documents, which are collections of key-value pairs. Keys are strings, and values can be any type within the data type collection, including arrays and documents [11], [12], [13], [14], [15], [16], [17], [18], [19]. As a NoSQL database, MongoDB offers comprehensive index support, dynamic querying, query monitoring, and an effective load balancing mechanism. Each cluster comprises one or multiple mongod processes, responsible for MongoDB’s core services and data storage. Typically, each shard opens multiple services to enhance service availability. These mongod processes within the shard constitute a replica set, providing redundancy and fault tolerance.

D. HORIZONTALLY SCALABLE BIPOLAR CLOUD STORAGE ARCHITECTURE

In general, computer storage media comprises various types such as disks, solid-state drives, memory, CPU cache, and others. As the distance to the CPU decreases, the speed of the storage media increases, while the storage capacity decreases. The efficiency of spatial data access is closely related to the storage medium of spatial data. The overall improvement in spatial data access efficiency is accomplished by replacing slower storage devices with faster ones. Thus, for different storage media, the same set of spatial data may exist in different forms. This system, based on a bipolar cloud spatial data storage architecture, primarily divides into Memory Cloud and Storage Cloud. The system constructs a multi-level

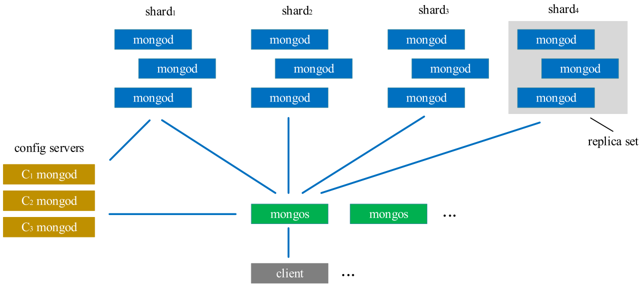

FIGURE 3. MongoDB cluster structure.

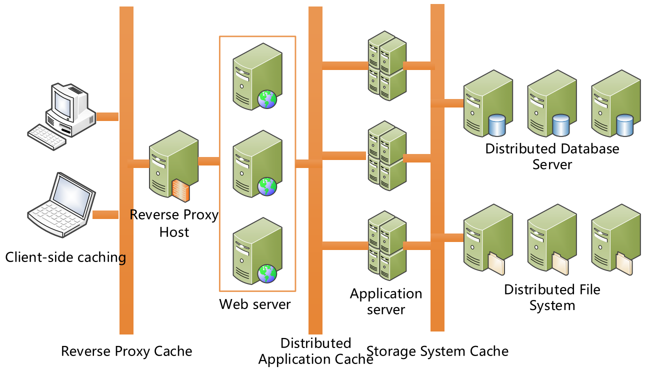

FIGURE 4. Multi-level caching system architecture.

caching mechanism to ensure that clients can access hot-spot spatial data as quickly as possible, categorized into Client-side cache, Reverse Proxy Cache, Distributed Application Cache, Database Cache, and Distributed File System Cache.

(1) Storage Cloud Design: Persistent storage of spatial data mainly adopts disk storage, including mechanical hard drives and solid-state drives. Spatial data storage methods can either be disk files or database records. The system's storage cloud primarily relies on cloud object storage for distributed storage, encompassing storage types like original image files, DOM results, and vector feature files.

(2) Memory Cloud Design: Memory computing stores the data to be analyzed within memory for processing, avoiding the performance bottleneck of disk I/O, significantly enhancing the system's execution efficiency, making it suitable for handling massive data and real-time response systems. Presently, 32GB, 64GB, or even larger memories are commonplace in servers, enabling large-scale computation based on memory cloud. The goal of memory computing is to enhance memory and CPU efficiency.

Regarding the caching system architecture, memory is a commonly used data caching medium, but in reality, data caching is more complex. The memory cloud storage types include geospatial processing results, cached tiles, cached vector information, etc. Memory cloud caching utilizes high-speed storage devices to cache frequently accessed hot-spot data, avoiding performance loss caused by frequent reads from slower devices. Additionally, it caches computational results to avoid resource waste and performance loss due to frequent redundant computations. The fundamental idea of data caching in this system is to store hot-spot data on high-speed devices and bring data as close to the client as possible. Moreover, it requires a well-designed scheduling strategy to ensure that the cached data remains hot-spot data, while timely removing less frequently accessed data from the cache.

(a)Client-Side Caching The closest to the user yields the best results, directly avoiding the need for client requests to the server's data. Client-side caching of spatial data encompasses both memory caching and local file caching. Data suitable for caching on the client-side typically includes static HTML, JavaScript, CSS, image files, as well as bulk DEM (Digital Elevation Model) and model data extensively used in grid data and three-dimensional systems. These data share a common trait: they exhibit very infrequent changes. Various client-side tools can devise and implement their unique client-side caching strategies. For instance, conventional 3D clients often opt to cache DEM data and model data to achieve higher efficiency.

(b)Reverse Proxy Cache The reverse proxy caches requests' results as (key, value) pairs in its cache, utilizing the request's URL as the key, ready for direct use upon subsequent visits. Initially intercepting the user's request, the reverse proxy first checks within its cache to determine if the request is cached. If found, it retrieves the data directly from the cache and sends it back to the client. Otherwise, it forwards the request to the Web server. The reverse proxy cache obviates the need for a substantial amount of disk I/O and server computing operations, significantly amplifying the system's throughput.

(c)Distributed Application Cache This is the cache within distributed application servers, serving as a cache for core business logic. The range of data cached within application servers is vast, covering grid image data, frequently used vector data, user permission data, address matching result data, server traffic monitoring status data, WFS (Web Feature Service) GetFeature query result data, WMS (Web Map Service) GetMap real-time map data, SQL query result data, database table records, real-time GPS signal data, and more. Essentially, almost any data that can be involved may potentially be cached here. Caching substantial volumes of data in this context often necessitates the use of distributed caching. The keys used for caching here may vary based on the content and data, requiring different key generation strategies.

(d)Database Cache Database caching involves caching data that is infrequently modified but frequently read, such as account data, frequently used spatial index data, SQL query results, etc. Once this data is cached, the database can directly return the data from memory to the application upon subsequent retrieval, eliminating the need for disk read operations and significantly improving speed.

(e)Distributed File System Cache Similar to database caching, the distributed file system cache is more straightforward. It involves caching file data blocks that are frequently

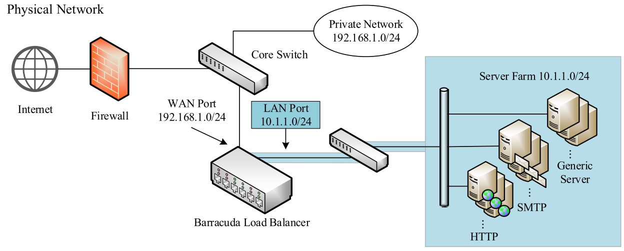

FIGURE 5. System load balancing mechanism.

read into memory to achieve faster access. Typically, server operating systems support the caching of data files.

E. CLOUD SERVICE LOAD BALANCING

In order to enable the cloud platform to respond rapidly and non-blockingly under multiple concurrent user conditions, the backend adopts a dual-system, multi-node Nginx load balancing mechanism that combines Linux and Windows. The load balancing is geographically structured into Local Load Balance and Global Load Balance. Local Load Balance refers to load balancing within the local server cluster, while Global Load Balance, also known as geographical load balancing, involves load balancing among server clusters placed in different geographic locations with distinct network structures.

Local Load Balance effectively addresses issues related to excessive data traffic and network overload without the need for investing in expensive, high-performance servers. It optimally utilizes existing equipment, avoiding data loss due to single-point server failures. Its flexible and diverse balancing strategies allocate data traffic reasonably among the servers within the cluster, allowing them to share the workload collectively. Even with upgrades or expansions to existing servers, it merely involves adding a new server to the server group without altering the existing network structure or disrupting ongoing services.

F. MATHEMATICAL MODELING

To facilitate a deeper understanding of the problem domain, we present a mathematical model that encapsulates the essence of the challenges faced in remote sensing monitoring and the advantages of our proposed bimodal cloud infrastructure.

Let D represent a set of diverse remote sensing data, where each element $d_i \in D$ corresponds to a specific data type or resolution. The goal is to efficiently store, index, and process this data in real-time to support various monitoring applications.

We define the operational efficiency (E) of a remote sensing monitoring system as a function of data processing time (T), data throughput (Θ), and security level (S):

$$E = f(T, \Theta, S)$$

The data processing time (T) is influenced by the hardware capabilities and the data management system. In our bimodal cloud infrastructure, we introduce a memory cloud (MC) for high-speed processing and a storage cloud (SC) for persistent data retention. The effectiveness of the memory cloud can be modeled as:

$$TMC = \frac{1}{1\ Processing\ Speed\ of\ MC}$$

Likewise, the storage cloud's contribution to the overall processing time is:

$$TSC = \frac{1}{Data\ Retrieval\ Speed\ of\ SC}$$

The total processing time T is a combination of memory cloud and storage cloud processing times, along with any synchronization overhead (O):

$$T = TMC + TSC + O$$

Data throughput (Θ) is a measure of the amount of data processed per unit time and can be calculated as:

$$\Theta = \frac{Volume\ of\ Processed\ Data}{\text{Time Interval}}$$

Security level (S) is a function of the implemented security mechanisms (M) and their effectiveness (e):

$$S = g(M, e)$$

Our proposed system incorporates a two-layer security system that not only protects data at rest but also during transit. The security mechanisms include encryption (E), access control (A), and intrusion detection (I):

$$M = \{E, A, I\}$$

The effectiveness e of these mechanisms is determined by their ability to prevent unauthorized access and ensure data integrity.

By optimizing T, maximizing Θ, and enhancing S, our bimodal cloud infrastructure aims to increase the operational efficiency E of remote sensing monitoring systems. This mathematical modeling provides a clear framework for understanding the problem domain and the focus of our research.

IV. RESULTS

The collaborative workflow, supported by the dual-state cloud, spans the entire remote sensing monitoring project and comprises several core stages: satellite image transit prediction, initial data quality inspection and archiving, ortho-product generation, change detection for land use information extraction, and result analysis. Each stage undergoes scheduling and processing at the terminal, while the cloud handles data processing and analysis in the Memory Cloud mode and stores and archives data in the Storage Cloud mode.

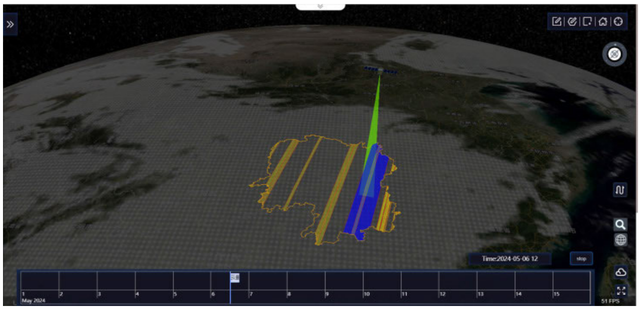

FIGURE 6. Satellite overflight prediction analysis.

A. SATELLITE IMAGE TRANSIT PREDICTION

Satellite orbit transit prediction offers reliable data support for the initial planning of remote sensing monitoring tasks. It provides comprehensive oversight of satellite coverage within the monitoring time frame to determine the necessity of supplementing additional data sources to achieve monitoring goals. Visual multi-satellite task simulation technology, adapted to planned multi-satellite observation tasks, integrates the characteristics of multiple domestic high-resolution remote sensing satellites like ZY1-02C, GF1, and GF2. This technology uses two-dimensional visualization to simulate the operating status and trajectories of multiple satellites, enabling comprehensive scheduling and coordinated observation of multiple high-resolution satellite tasks.

Using the SGP4/SDP4 (Simplified General Perturbation Version 4/Simplified Deep-space Perturbation Version 4) satellite orbit estimation method proposed by the North American Aerospace Defense Command, precise orbit estimation of the satellite is carried out based on the initial satellite motion state. This method accounts for perturbation forces such as Earth's non-spherical gravity, solar and lunar gravity, solar radiation pressure, and atmospheric drag. It demonstrates high predictive accuracy and superior predictive convergence effects [20], [21], [22], [23], [24], [25].

Specific features include:

Providing location search functionality: offering search capabilities for regions at or above the county level nationwide, highlighting the user-entered area and enabling its preservation in the observation area.

Image prediction functionality for observation areas: providing emergency-accessible data and area lists within the forthcoming 1 to 7 days.

Additionally, this technology integrates common GIS functionalities into the simulation system, facilitating future applications. It utilizes OpenLayers for visualization, ensuring real-time daily updates to satellite orbit data to ensure accurate satellite orbit prediction [26], [27], [28], [29].

B. PIXEL-LEVEL IMAGE PROCESSING CHAIN IN THE MEMORY CLOUD ENVIRONMENT

The cloud-end deploys PCI GeoImaging Accelerator (GXL), featuring characteristics such as multitasking parallel computing, GPU-accelerated graphics processing, and distributed processing. It automates time-consuming image processing tasks, including control point collection, image rectification, image mosaicking, and image fusion, throughout the image processing workflow, maximizing efficiency and minimizing the waiting time for manual image processing. Technicians can directly call upon computing resources through a web client, generating level-four image products. An experiment involving ortho-processing for 40 scenes of GF2 imagery showed an 8-10 times increase in overall efficiency compared to single-machine operations.

C. IMAGE SERVICE PUBLISHING

After generating ortho-image DOMs, these DOMs are published as cloud-based Web Map Services (WMS) using OGC (Open Geospatial Consortium) standards, accessible to all applications (using the OpenLayers framework) within the private cloud. OGC and ISO/TC211 jointly introduced spatial data interoperability specifications, including Web Map Service, Web Feature Service, Web Coverage Service, and the Geography Markup Language (GML). WMS utilizes geospatial data with geographic location information to create maps. This service efficiently transforms Storage Cloud data into Memory Cloud, offering robust browsing speeds. Dynamic access to heavyweight DOMs in the cloud-based browser mode is faster than browsing copies in a single-machine version, significantly reducing storage resource wastage and data redundancy.

It is prohibited to modify vector data requiring association, and administrative boundary vectors are made accessible through WMS calls. Modified vectors utilize WFS service mode for access, providing a bidirectional interactive service that allows users to modify vector attributes in addition to obtaining their boundaries. Technicians can directly modify and submit vectors based on their current permissions.

In our experiment, we conducted a detailed comparison between our developed Cloud Remote Sensing Geographic Information Service Engine and globally renowned Geographic Information Service Rendering Engines. The compared products primarily include ArcGIS Desktop, ArcGIS Server, GeoServer, among others. The main parameter for comparison was the number of rendered tiles loaded within a unit time, which intuitively reflects the rendering performance of the Geographic Information Service Engine. The experiment was conducted under identical software, hardware, and network conditions with the following specific configuration: (1) Specification: 4U Rack-mounted Server; (2) Processor: 4 Intel Xeon 6130 (2.1GHz/16C); (3) Memory: 4 units of 32GB DDR4 memory; (4) Hard Drive: 2 units of 300GB 15K RPM SAS hard drives; (5) RAID: Configured with SAS 9361 RAID (1G cache); (6) Network: Single-port 10 Gigabit optical network card (including multimode optical modules). The comparative experiments revealed that the performance of the Cloud Remote Sensing Geographic Information Service Engine, utilizing a bipolar cloud architecture, significantly surpassed other service engines.

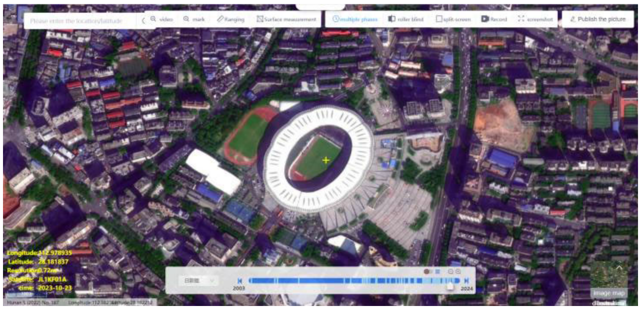

FIGURE 7. Platform image WMS service(mosaic image).

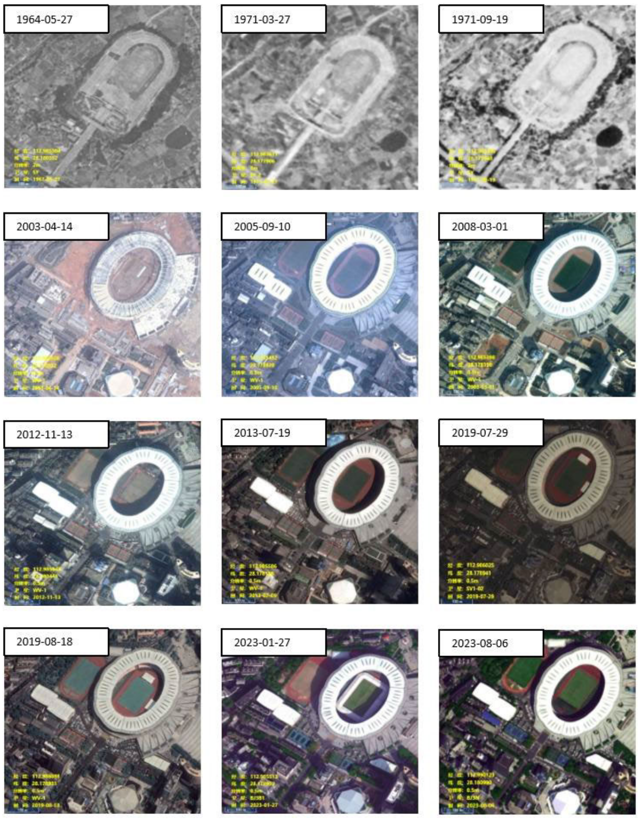

FIGURE 8. Platform image WMS service(large-scale remote sensing image time series).

It achieved a rendering rate exceeding 120 tiles per second, more than double that of ArcGIS Desktop, over four times that of GeoServer, and more than seven times that of ArcGIS Server.

D. COLLABORATIVE INFORMATION EXTRACTION PLATFORM

The information extraction system adopts a cloud-based WebGIS architecture, with MongoDB serving as the supporting database. To enhance user access efficiency, it deploys 100 Mbps fiber optics and 100TB-level data storage space for data storage and transmission services. Additionally, the backend system utilizes Node.js for web services, which encapsulates the Google V8 engine known for its rapid execution of JavaScript. Node.js offers optimizations for specific use cases, providing alternative APIs that enhance V8 performance in non-browser environments. The software environment supports the deployment of Node.js, and due to spatial data conversion requirements, the system simultaneously deploys GDAL (Geospatial Data Abstraction Library), an open-source raster spatial data conversion library operating under the X/MIT license. The frontend adopts the OpenLayers framework, loading map data from Tianditu (TianDiTu) and WMS/WFS services published by Geoserver as the basic framework, incorporating user login permissions control, GIS analysis, task assignment submission, and multiple sub-functional modules [30], [31], [32], [33], [34], [35], [36], [37], [38].

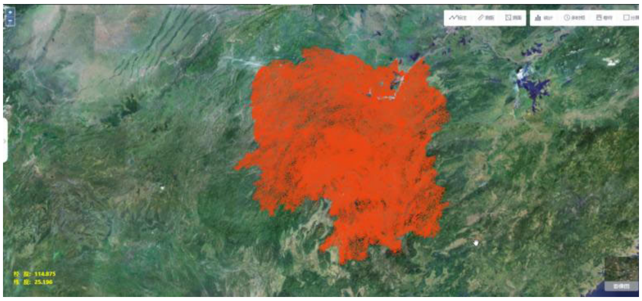

FIGURE 9. Platform vector WMS service.

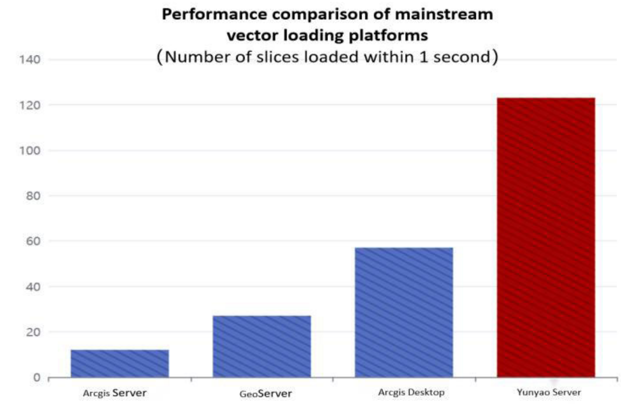

FIGURE 10. Performance comparison of mainstream vector loading platforms.

The information extraction platform employs a user hierarchical mechanism, granting different operational functionalities to information extractors, quality inspectors, and reviewers based on their permissions. Information extractors utilize web-based image comparison to assess and attribute image patches using loaded image bases and current recognition images, employing common GIS analysis tools for statistical analysis. Quality inspectors and reviewers have higher privileges, conducting comprehensive evaluations of the quality of personnel's outputs and entering the assessments into the system. All roles collaborate, participating in the overall cloud-based project implementation.

project: 1111222

FIGURE 11. Swiper information extraction module.

E. SECURITY ASSURANCE SYSTEM

The entire collaborative workflow involves image and result data that possess high levels of confidentiality. Therefore, the platform has been designed with two security assurance systems to ensure the security of the project implementation.

Firstly, when transmitting data from the upper-level cloud to the lower-level cloud, a data transmission layer security assurance system has been implemented. This transmission process utilizes the WebSocket data transmission protocol, revolutionizing the conventional one-way polling operation and enabling full-duplex communication between the browser and the server for active data push. It breaks through key technologies involving spatial metadata extraction synchronization, secure data push, and data integrity verification. It realizes active spatial information parsing, data packaging, dynamic key creation, secure push, key loading implementation, automated decryption, and automatic data unpacking in a one-stop push process. Additionally, in response to unstable networks and specific unexpected situations in various regions, a breakpoint resumption mechanism has been added to actively push and maximize the integrity and reliability of the data transmission link. Images are made available for download through a cloud server-generated download link, which must include user authentication information. To ensure the security of this information, all user information involved in the download link is encrypted using MD5 into non-plaintext strings. As the system provides open download links to the outside, to prevent malicious dissemination and unauthorized downloads of images, a random code parameter is set at the end of the link. The download link also has a specified expiration period. After the deadline, the cloud re-generates a random code to create a new download link, rendering the old link immediately invalid. Moreover, the upper-level cloud server can record user information and IP addresses during downloads, preventing the misuse of download links and data leaks.

Secondly, within the lower-level cloud, a strict user management system has been established, physically isolating operations from the external network, providing an additional layer of security. All ortho-image results are released in sliced form within the private cloud, accessible only to users with corresponding permission levels. Additionally, they cannot be copied or written into; for new results of operational vectors generated by staff, modifications are subjected to an application-based system. If modifications are required for the submitted results, users must submit a modification request through the user center to higher-level permission users for online approval before making the necessary changes.

V. RESULTS DISCUSSION AND ALGORITHM IMPROVEMENT

A. RESULTS DISCUSSION

Our proposed remote sensing monitoring framework has demonstrated significant improvements in operational efficiency through the innovative use of a bimodal cloud infrastructure. To provide a thorough comparison with existing approaches, we will discuss the operational efficiency of our framework in relation to relevant works in the field.Comparison with Relevant Works

Operational Efficiency: Traditional remote sensing monitoring systems often suffer from operational inefficiencies due to the lack of integrated data processing and management mechanisms. In contrast, our framework, through the dual-state cloud service mechanism, achieves operational efficiency by optimizing data processing and access speeds. For instance, while Becker-Reshef et al. [1] focused on global cropland monitoring using coarse-resolution earth observations, our system offers a more efficient approach by leveraging the Memory Cloud for rapid data processing.

Data Processing Speed: Our "YunYao" geographic information service rendering engine outperformed ArcGIS Desktop by over two times, exceeded GeoServer by more than four times, and was over seven times faster than ArcGIS Server in rendering speeds. This level of performance is unprecedented in the field and highlights the superior operational efficiency of our system.

Security and Integrity: Unlike other systems that may overlook the importance of data security, our framework implements a two-tier security system that ensures data integrity and confidentiality throughout the monitoring process. This is a critical aspect that sets our work apart from other remote sensing monitoring systems that may not prioritize data security to the same extent.

Scalability and Flexibility: Our system's design allows for scalability and flexibility, accommodating the varying demands of remote sensing monitoring tasks. This is in contrast to more rigid systems that may not easily adapt to changing data volumes or processing requirements.

Cost-Effectiveness: By utilizing cloud services and optimizing resource allocation, our framework reduces hardware costs associated with traditional remote sensing systems. This cost-effective approach is particularly beneficial for long-term monitoring projects where expenses can accumulate.

Furthermore, our system has shown excellent performance in handling large volumes of remote sensing data.

By integrating metaheuristic algorithms, such as genetic algorithms, particle swarm optimization, and ant colony optimization, our system achieves more efficient scheduling and processing of remote sensing data within the cloud infrastructure, effectively addressing complex optimization problems.

B. ALGORITHM IMPROVEMENT

The core improvements of our proposed approach are as follows:

(1) **Dual-State Cloud Service Mechanism**: By combining Memory Cloud and Storage Cloud, our system enables rapid access to frequently requested spatial data while maintaining long-term data persistence.

(2) **Multi-Level Caching System**: A multi-level caching mechanism is designed within our system, including client-side cache, reverse proxy cache, distributed application cache, database cache, and distributed file system cache, ensuring quick access to hotspot spatial data for clients.

(3) **Metaheuristic Algorithm Integration:** Our approach goes beyond existing work by incorporating metaheuristic algorithms to optimize the scheduling and processing of remote sensing data within the cloud infrastructure.

(4) **Two-Layer Security System:** A robust security framework is implemented, consisting of a data transmission layer and a strict user management system, ensuring data integrity and confidentiality.

Cloud Service Load Balancing: By balancing the load between local and global server clusters, our system effectively handles high concurrency under multi-user conditions, improving response speed and availability.

In summary, our proposed remote sensing monitoring framework offers significant enhancements in efficiency, security, and scalability. Through practical applications and comparative analysis, we have validated the superior performance of our framework in processing remote sensing data, providing a novel solution for the field of remote sensing monitoring. Future work will focus on further optimizing system performance and exploring additional applications of cloud service technologies in remote sensing monitoring.

VI. CONCLUSION

The establishment of a collaborative remote sensing monitoring system under the dual-state cloud service mode is an exploration within the current cloud service model in the field of remote sensing. Contrasted with the traditional manual single-machine operation as the primary operational mode, the cloud service collaborative working mode has significantly liberated the waiting time and data redundancy within the operational workflow. Spanning the entire service chain from satellite prediction to result publication, it greatly optimizes production efficiency, minimizing the time cycle from raw data to service delivery. Furthermore, the novel management mode under the cloud service model maximizes the security of data and results. Shifting from handling big data to lightweight operations, users under the cloud service model no longer need to focus on the data processing process, freeing up more time to explore and analyze the value of image data. Presently, this research technology has been fully applied in the ‘‘Natural Resources Satellite Remote Sensing Cloud Service Platform’’ of the Ministry of Natural Resources and the ‘‘Satellite Cloud Remote Sensing’’ system of the Hunan Provincial Department of Natural Resources, both achieving favorable application effects.

The ‘‘YunYao’’ geographic information service rendering engine, as demonstrated, has achieved remarkable performance in remote sensing monitoring through the innovative dual-state cloud infrastructure. Our framework’s efficiency, security, and scalability present a significant advancement in the field of remote sensing technology. However, as with any research, there are limitations that need to be acknowledged and addressed in future work.

FUTURE WORK

Our future work will focus on several key areas to enhance the proposed framework:

Algorithm Optimization: We plan to further optimize the metaheuristic algorithms used for data scheduling and processing to handle an increased volume of remote sensing data.

Cloud Service Integration: We aim to improve the integration of cloud services, particularly in the establishment of private clouds and internal collaborative computing mechanisms within the remote sensing domain.

User Interface Enhancement: The development of a more intuitive user interface for non-expert users to leverage the power of remote sensing monitoring will be pursued.

Performance Benchmarking: We will conduct extensive benchmarking against other state-of-the-art systems to validate and improve our framework’s performance.

Scalability Testing: Further testing will be carried out to evaluate the system’s scalability under various loads and conditions.

LIMITATIONS

It is important to distinguish between the theoretical and practical limitations of our study:

Theoretical Limitations:

Modeling Assumptions: The mathematical model used in our study makes certain assumptions about data distribution and processing times, which may not hold true for all real-world scenarios.

Algorithm Complexity: The metaheuristic algorithms, while effective, have a computational complexity that could be limiting for extremely large-scale data sets.

Security Model Generalizability: The two-layer security system, though robust, may not be directly applicable to all cloud service providers due to varying security protocols and compliance requirements.

Practical Limitations:

Hardware Dependency: The system’s performance is inherently dependent on the underlying hardware resources, which may limit its applicability in resource-constrained environments.

Data Transfer Bottlenecks: High volumes of data transfer between memory and storage clouds can create bottlenecks, affecting overall system responsiveness.

Cost-Effectiveness: The financial implications of maintaining a dual-state cloud infrastructure, especially for small-scale operations, need to be considered.

User Training and Adoption: There may be a learning curve associated with adopting new technology, which could affect user acceptance and the technology's practical implementation.

By acknowledging these limitations, we aim to provide a comprehensive understanding of the scope and applicability of our research. Future work will be directed at mitigating these limitations and further refining the proposed framework to meet the evolving demands of remote sensing monitoring.

The satellite remote sensing domain's cloud service technology still requires further advancement. Particularly, in the establishment of private clouds, internal collaborative computing and deployment mechanisms in the remote sensing field lack mature reference cases. For instance, GIS analysis functions on the web interface do not yet match the speed of standalone software. Load balancing mechanisms are insufficient to handle a vast number of visitors, and there is still a considerable geometric exponential increase in hardware investment required in the cloud. In the long term, these issues represent challenges that the cloud service model needs to overcome, ultimately integrating the cloud service model completely into the remote sensing monitoring domain.

REFERENCES

[1] I. Becker-Reshef, C. Justice, M. Sullivan, E. Vermote, C. Tucker, A. Anyamba, J. Small, E. Pak, E. Masuoka, J. Schmaltz, M. Hansen, K. Pittman, C. Birkett, D. Williams, C. Reynolds, and B. Doorn, "Monitoring global croplands with coarse resolution Earth observations: The global agriculture monitoring (GLAM) project," *Remote Sens.*, vol. 2, no. 6, pp. 1589–1609, Jun. 2010, doi: 10.3390/rs2061589.

[2] F. Ren and J. Wang, "Turning remote sensing to cloud services: Technical research and experiment," *Nat. Remote Sens. Bull.*, vol. 16, no. 6, pp. 1331–1346, 2012.

[3] M. Amani, A. Ghorbanian, S. A. Ahmadi, M. Kakooei, A. Moghimi, S. M. Mirmazloumi, S. H. A. Moghaddam, S. Mahdavi, M. Ghahremanloo, S. Parsian, Q. Wu, and B. Brisco, "Google Earth engine cloud computing platform for remote sensing big data applications: A comprehensive review," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5326–5350, 2020.

[4] J. Sun, Y. Zhang, Z. Wu, Y. Zhu, X. Yin, Z. Ding, Z. Wei, J. Plaza, and A. Plaza, "An efficient and scalable framework for processing remotely sensed big data in cloud computing environments," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4294–4308, Jul. 2019.

[5] K. R. Ferreira, G. R. Queiroz, G. Camara, R. C. M. Souza, L. Vinhas, R. F. B. Marujo, R. E. O. Simoes, C. A. F. Noronha, R. W. Costa, J. S. Arcanjo, V. C. F. Gomes, and M. C. Zaglia, "Using remote sensing images and cloud services on aws to improve land use and cover monitoring," in *Proc. IEEE Latin Amer. GRSS ISPRS Remote Sens. Conf. (LAGIRS)*, Mar. 2020, pp. 558–562.

[6] B. Wu, F. Tian, M. Zhang, H. Zeng, and Y. Zeng, "Cloud services with big data provide a solution for monitoring and tracking sustainable development goals," *Geography Sustainability*, vol. 1, no. 1, pp. 25–32, Mar. 2020.

[7] K. Chen and W. M. Zheng, "Cloud computing: System instances and current research," *J. Softw.*, vol. 20, no. 5, pp. 1337–1348, 2009.

[8] D.-G. Feng, M. Zhang, Y. Zhang, and Z. Xu, "Study on cloud computing security," *J. Softw.*, vol. 22, no. 1, pp. 71–83, Mar. 2011, doi: 10.3724/sp.j.1001.2011.03958.

[9] R. Li, "A RAM-drive cloud-based strategy for the high-resolution remote sensing data storage and computation integration," Zhejiang Univ., Hangzhou, China, Tech. Rep. 01, 2014.

[10] C. E. Soulard, J. J. Walker, and R. E. Petrakis, "Implementation of a surface water extent model in Cambodia using cloud-based remote sensing," *Remote Sens.*, vol. 12, no. 6, p. 984, Mar. 2020.

[11] C. Xu, X. Du, Z. Yan, and X. Fan, "ScienceEarth: A big data platform for remote sensing data processing," *Remote Sens.*, vol. 12, no. 4, p. 607, Feb. 2020.

[12] S. Bradshaw, E. Brazil, and K. Chodorow, *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. Sebastopol, CA, USA: O'Reilly Media, 2019.

[13] H. Matallah, G. Belalem, and K. Bouamrane, "Comparative study between the MySQL relational database and the MongoDB NoSQL database," *Int. J. Softw. Sci. Comput. Intell.*, vol. 13, no. 3, pp. 38–63, Jul. 2021.

[14] P. Filip and L. Cegan, "Comparison of MySQL and MongoDB with focus on performance," in *Proc. Int. Conf. Informat., Multimedia, Cyber Inf. Syst. (ICIMCIS)*, Nov. 2020, pp. 184–187.

[15] S. Palanisamy and P. SuvithaVani, "A survey on RDBMS and NoSQL databases MySQL vs MongoDB," in *Proc. Int. Conf. Comput. Commun. Informat. (ICCCI)*, Jan. 2020, pp. 1–7.

[16] J. M. Medina, I. J. Blanco, and O. Pons, "A fuzzy database engine for mongoDB," *Int. J. Intell. Syst.*, vol. 37, no. 9, pp. 5691–5724, Sep. 2022.

[17] A. Chauhan, "A review on various aspects of MongoDb databases," *Int. J. Eng. Res. Sci. Technol.*, vol. 8, no. 5, pp. 90–92, 2019.

[18] I. Kuyumdzhiev, "Comparing backup and restore efficiency in MySQL, MS SQL server and MongoDB," in *Proc. Int. Multidisciplinary Sci. GeoConf.*, 2019, vol. 19, no. 2, pp. 167–174.

[19] G. Harrison and M. Harrison, *MongoDB Performance Tuning: Optimizing MongoDB Databases and Their Applications*. Berkeley, CA, USA: Apress, 2021.

[20] A. Makris, K. Tserpes, G. Spiliopoulos, and D. Anagnostopoulos, "Performance evaluation of MongoDB and PostgreSQL for spatio-temporal data," in *Proc. EDBT/ICDT Workshops*, 2019, pp. 5–7.

[21] P. Dutt, M. Mutyalarao, P. Bhanumathy, T. R. Saritha Kumari, D. Negi, A. K. Anilkumar, A. Kumar, and V. Ashok, "Assessment of in-house algorithms on re-entry time prediction of uncontrolled space objects," *Adv. Space Res.*, vol. 72, no. 7, pp. 2535–2551, Oct. 2023.

[22] P. Hickson, "OCS: A flexible observatory control system for robotic telescopes with application to detection and characterization of orbital debris," in *Proc. 1st Int. Orbital Debris Conf.*, Sugar Land, TX, USA, 2019, pp. 6–8.

[23] A. Mukundan and H.-C. Wang, "Simplified approach to detect satellite maneuvers using TLE data and simplified perturbation model utilizing orbital element variation," *Appl. Sci.*, vol. 11, no. 21, p. 10181, Oct. 2021.

[24] G. Jadala, G. N. Meedinti, and R. Delhibabu, "Satellite orbit prediction using a machine learning approach," Tech. Rep., 2022.

[25] X. Xiao-Li and X. Yong-Qing, "Study on the orbit prediction errors of space objects based on historical TLE data," *Chin. Astron. Astrophys.*, vol. 43, no. 4, pp. 563–578, Oct. 2019.

[26] G. Wen, Y. Xu, C. Liu, and Z. Zhou, "A simulation system of space object images for space situation awareness," in *Proc. 15th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, Nov. 2022, pp. 1–6.

[27] E. Hazzard, *Openlayers 2.10 Beginner's Guide*. Packt Publishing Ltd, 2011.

[28] A. Maiti, S. Majumdar, S. Shukla, S. R. Koti, and P. K. Gupta, "An open source web-gis based precise satellite tracking and visualisation tool using two line element data," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 4, pp. 109–114, 2018.

[29] V. K. Sharma, E. Amminedu, G. S. Rao, P. V. Nagamani, K. R. Mohan Rao, and V. Bhanumurthy, "Assessing the potential of open-source libraries for managing satellite data products—A case study on disaster management," *Ann. GIS*, vol. 23, no. 1, pp. 55–65, Jan. 2017.

[30] J. A. Gwinn, P. M. Thomas, and J. F. Kielkopf, "Satellite bands in the emission spectrum of cesium," *J. Chem. Phys.*, vol. 48, no. 2, pp. 568–572, Jan. 1968.

[31] G. Krylov, M. Patrou, G. W. Dueck, and J. Siu, "The evolution of garbage collection in V8: Google's Javascript engine," in *Proc. 9th Medit. Conf. Embedded Comput. (MECO)*, Jun. 2020, pp. 1–6.

[32] J. K. Martinsen, H. Grahn, and A. Isberg, "Combining thread-level speculation and just-in-time compilation in Google's V8 Javascript engine," *Concurrency Comput., Pract. Exp.*, vol. 29, no. 1, Jan. 2017, Art. no. e3826.

[33] C. D. Michaelis and D. P. Ames, ''Web feature service (WFS) and web map service (WMS),'' 2008, doi: 10.1007/978-0-387-35973-1_1480. [Online]. Available: http://www.opengeospatial.org
[34] C. D. Michaelis and D. P. Ames, ''Considerations for implementing OGC WMS and WFS specifications in a desktop GIS,'' *J. Geograph. Inf. Syst.*, vol. 4, no. 2, pp. 161–167, 2012.
[35] X. Song, Y. Kono, and M. Shibayama, ''The development of web mapping application using open source GIS solution,'' in *Proc. Int. Symp. Geoinformat. Spatial Infrastruct. Develop. Earth Allied Sci.*, vol. 2004, Hanoi, Vietnam, pp. 1–6.
[36] P. Lemenkova and O. Debeir, ''GDAL and PROJ libraries integrated with GRASS GIS for terrain modelling of the georeferenced raster image,'' *Technologies*, vol. 11, no. 2, p. 46, Mar. 2023.
[37] Y. Jiang, M. Sun, and C. Yang, ''A generic framework for using multidimensional Earth observation data in GIS,'' *Remote Sens.*, vol. 8, no. 5, p. 382, May 2016.
[38] J. Masó, X. Pons, and R. Singh, ''OGC WMTS and OSGeo TMS standards: Motivations, history and differences,'' in *Proc. FOSS4G*, 2010.

YANG KAIJUN received the B.S. and M.S. degrees in surveying and mapping engineering from Wuhan University, Wuhan, China, in 2013 and 2015, respectively. He is currently pursuing the Ph.D. degree with China University of Geosciences, Wuhan. He is a Senior Engineer with the Second Surveying and Mapping Institute of Hunan Province. His primary research interests include remote sensing informatization and intelligent processing.

LEI FAN received the B.S. and M.S. degrees in Chinese painting from Xi'an Academy of Fine Arts, Xi'an, China, in 2005 and 2008, respectively. He is a Senior Engineer with the Second Surveying and Mapping Institute of Hunan Province. His primary research interests include remote sensing informatization and agricultural remote sensing.

WEI JIDE received the B.S. and M.S. degrees in geographic information system engineering from Fuzhou University, Fuzhou, China, in 2008 and 2012, respectively. He is a Senior Engineer with the Second Surveying and Mapping Institute of Hunan Province. His primary research interests include remote sensing informatization and agricultural remote sensing.

ZHANG ZHE received the B.S. degree in computer science from the Central South University of Forestry and Technology, Changsha, China, in 2012. He is a Senior Engineer with the Second Surveying and Mapping Institute of Hunan Province. His primary research interests include remote sensing system development and UAV data processing.

• • •