Hindawi
Journal of Mathematics
Volume 2021, Article ID 3224190, 13 pages
https://doi.org/10.1155/2021/3224190

Research Article

Distributed Storage Strategy and Visual Analysis for Economic Big Data

Xiangli Chang[1] and Hailang Cui[2]

[1]*College of Economics and Management, Henan Vocational College of Quality Engineering, Pingdingshan, Henan 467000, China*
[2]*Business School, Yunnan University of Finance and Economics, Kunming, Yunnan 650221, China*

Correspondence should be addressed to Hailang Cui; zz1771@ynufe.edu.cn

Received 20 October 2021; Accepted 11 November 2021; Published 27 November 2021

Academic Editor: Miaochao Chen

With the increasing popularity of a large number of Internet-based services and a large number of services hosted on cloud platforms, a more powerful back-end storage system is needed to support these services. At present, it is very difficult or impossible to implement a distributed storage to meet all the above assumptions. Therefore, the focus of research is to limit different characteristics to design different distributed storage solutions to meet different usage scenarios. Economic big data should have the basic requirements of high storage efficiency and fast retrieval speed. The large number of small files and the diversity of file types make the storage and retrieval of economic big data face severe challenges. This paper is oriented to the application requirements of cross-modal analysis of economic big data. According to the source and characteristics of economic big data, the data types are analyzed and the database storage architecture and data storage structure of economic big data are designed. Taking into account the spatial, temporal, and semantic characteristics of economic big data, this paper proposes a unified coding method based on the spatiotemporal data multilevel division strategy combined with Geohash and Hilbert and spatiotemporal semantic constraints. A prototype system was constructed based on Mongo DB, and the performance of the multilevel partition algorithm proposed in this paper was verified by the prototype system based on the realization of data storage management functions. The Wiener distributed memory based on the principle of Wiener filter is used to store the workload of each workload distributed storage window in a distributed manner. For distributed storage workloads, this article adopts specific types of workloads. According to its periodicity, the workload is divided into distributed storage windows of specific duration. At the beginning of each distributed storage window, distributed storage is distributed to the next distributed storage window. Experiments and tests have verified the distributed storage strategy proposed in this article, which proves that the Wiener distributed storage solution can save platform resources and configuration costs while ensuring Service Level Agreement (SLA).

1. Introduction

With the continuous development of Internet technology, the amount of data generated and accumulated on the Internet has exploded. The information generated by people's activities on the Internet, such as posting Weibo, shopping, and comments, is eventually converted into binary data and stored. Cloud computing and Internet of Things technologies have further increased the speed and scale of data growth, and all walks of life are facing the challenge of big data [1–3]. In the era of big data, new data is continuously generated and accumulated every day. Massive data pushes the storage capacity of databases to the limit, making enterprises have to regularly expand databases to accommodate the ever-growing massive economic big data [4, 5].

With the rapid development of multisource sensor technology, different from traditional spatial data, economic big data not only has the massive and unstructured characteristics of traditional spatial data, but also has multimodal characteristics. Multimodal characteristics refer to the same spatiotemporal object with many different forms of data description, and different modal data have the characteristics of heterogeneous low-level features and high-level semantic correlation. The multimodal characteristics of

multisource economic big data provide the possibility to support multisource spatiotemporal data in-depth cross-modal analysis, mining, and application. Therefore, how to realize the storage of economic big data and provide database support for cross-modal analysis applications has become the focus of current research.

Most of the analysis of economic census data is still based on the methods of purely using statistics and artificial intelligence. Both methods are dedicated to pattern discovery and prediction, while the intelligent data analysis method continues these sophisticated mathematical foundations of artificial intelligence and statistics, encapsulates their technology, and provides them as a service to the staff. Workers can ignore the principles of these technologies and just focus on the problems they want to solve. Generally speaking, the intelligent data analysis method combines statistics, artificial intelligence theory, modern software technology, and computer science technology. Although the application of intelligent data analysis methods in the field of economic census is not perfect, research results in other fields have laid a solid theoretical foundation for the application of this technology in economic census.

Judging from the current development, the access efficiency and applicability of the spatial data engine are better than other methods. However, relational databases are only good at managing structured data. For unstructured spatiotemporal data, such as videos and images, they are usually stored as multimedia attribute data, and it is difficult to support cross-modal query analysis and in-depth application of unstructured spatiotemporal data. The traditional spatial data division ignores the temporal and semantic relevance that spatial data may have, and it is easy to separate the deep semantic relationships contained in spatial data and reduce the storage and management efficiency of public safety monitoring big data. This brings inconvenience to subsequent deep mining and analysis applications based on economic big data. Therefore, in the face of economic big data with multimodal characteristics, time and semantic relevance of the data must be taken into consideration. Research on distributed storage technology based on multilevel partition strategy has become a key issue to be solved urgently.

This article analyzes the main principles of the division of economic big data, discusses the mainstream division methods, and analyzes the characteristics and application of each method. Specifically, the technical contributions of this article can be summarized as follows:

First: a multilevel data division method based on the combination of Geohash and Hilbert is proposed. This method can take into account the spatial proximity, temporal relevance, and semantic similarity of economic big data under the premise of satisfying the efficiency of the algorithm.

Second: on the basis of Geohash coding and Hilbert coding, this paper integrates the time stamp and semantic coding table of the data and proposes a unified coding method with temporal and spatial semantic constraints, and divides the data elements into corresponding nodes based on the unified coding.

Third: this article has been tested in the NoSQL storage system. Based on the Wiener distributed storage workload, the performance of the cluster was adjusted. The results show that it can achieve the purpose of the original design, save resources, and reduce costs.

The rest of this article is organized as follows. Section 2 presents related work. Section 3 discusses the main technologies and methods of the distributed economic big data computing analysis platform. Section 4 designs a distributed storage strategy with temporal and spatial semantic constraints. In Section 5, a visual experiment and analysis are carried out. Section 6 summarizes the full text.

2. Related Work

Distributed storage system is significantly different from traditional centralized storage. It improves the data management defects of traditional storage methods and data storage security issues [6]. The distributed storage system separates and processes metadata and business data and distributes business data on low-cost server nodes, so that the scale of storage data and the node servers that access the storage can be linearly expanded [7, 8]. Of course, in the process of engineering application, many practical problems have been exposed. Major international companies and research institutions have invested a lot of manpower and material resources to study distributed storage systems [9]. The underlying storage, scheduling tasks, and interaction modes are hot topics of research [10]. Well-known systems include Google's "Google Storage," Amazon's S3 Simple Service, Apache's Hadoop, Twitter's Impala, vmware's "Vmare v Spere," and Microsoft's Windows Azure. The "Google Storage" cloud storage service is launched by Google, and the key storage technology GFS is a typical distributed file storage system [11]. Google is a well-known search company, and it ranks among the best in user searches. According to the company's characteristics, it designs a distributed storage system that mainly stores the webpages of major websites in the world, crawls them, and stores them. Everyone knows the data of the webpages. The data types of each web page are various [12]. There are audio and pictures. The difficulty in designing a distributed storage system can be imagined. At the same time for crawling web pages, we also need to perform big data analysis on each web page and analyze the theme features and key feature extraction of each web page [13]. We can imagine how huge the data scale is. At present, Google has more than 200 super Google File System (GFS) cluster service data centers in the world, and the daily processing data scale reaches 20 PB, and it is gradually increasing [14, 15]. Related scholars have proposed a load balancing algorithm for Hadoop Distributed File System (HDFS), which qualitatively considers the migration of hot metadata under high load, but there is no quantitative analysis and lack of dynamics and adaptability [16]. Researchers use Bloom filter technology; although they can quickly locate the location of the Minimum Discernible Signal (MDS), they still do not consider the issue of dynamic allocation of metadata based on access changes, so the

problem of load balancing is also not solved [17]. Related scholars use the Basic Balancing Load Algorithm (BBLA) for the initial load distribution and use the Incremental Balancing Load Algorithm (IBLA) to adjust to achieve the purpose of load balancing, but the parameters used when introducing the access delay and control functions are completely fixed, and the load distribution cannot be achieved well [18–20]. Both can be regarded as the concrete realization of static hashing and dynamic hashing respectively.

Nowadays, data cluster administrators usually adopt an oversupply strategy to configure the database when building a database system to ensure Service Level Agreement (SLA). This strategy is to reasonably estimate the service request load and prepare sufficient resources for users. This brings about a problem. Overallocation of resources causes a waste of resources and increases costs. How to adjust cluster resources so that services can maximize resource utilization without violating SLA restrictions, reduce resource waste, and save costs for customers will be the biggest challenge we face.

Currently, there are two types of mainstream storage databases in the cloud environment, relational and nonrelational [21]. Traditional relational DBMS does not support horizontal expansion and tends to perform poorly in a cloud environment. On the contrary, nonrelational databases eliminate performance problems when processing large amounts of data. Some different vendors adopt different strategies for managing data models to implement different NoSQL based on application requirements, but these implementations lack interoperability and data portability [22]. With the rapid growth of data volume, relational databases cannot scale horizontally, so the cost of using relational databases to build big data solutions will become very expensive. Secondly, relational databases are not good at solving unstructured data. Therefore, some social network and big data companies, such as Facebook and Google, realize that in the environment of massive data and diversified data types, nonrelational databases are the best solution [23]. Users can expand the number of data storage servers at any time according to the volume of data.

3. Main Technologies and Methods of Distributed Economic Big Data Computing and Analysis Platform

This section introduces the distributed storage technology and its principles at the core of the platform and analyzes the technical theoretical principles of the platform's distributed storage. Then we introduce the core big data calculation and analysis principles that will be used and based on the platform to realize the analysis and calculation of economic big data, which indirectly proves the feasibility of the distributed economic big data calculation and analysis platform.

3.1. Distributed File System. The DFS that the distributed economic big data computing and analysis platform plans to adopt is HDFS based on Hadoop technology. The so-called DFS is a network service that combines file systems distributed on different computers into a namespace and enables the entire network to share and easily access these files. As one of the key technologies of big data technology, DFS plays an important role in the field of big data and is one of the keys to the development of the whole big data technology. The most famous of the development of big data technology is the special HDFS based on Hadoop, which is based on distributed storage technology. HDFS is designed to be suitable for running on a computer cluster with a common configuration, that is, a distributed file system that can run on a general von Neumann-structured computer without special servers.

Specifically, HDFS has the following premise and design goals. The first is the support and compatibility of hardware errors, and hardware failure is a normal state of distributed data storage and computing clusters. A typical HDFS can contain a large number of server nodes, and each node stores part of the cluster data. Having so many system nodes, plus each system node has a certain probability of failure, means that if HDFS does not have a fault tolerance mechanism, it is easy to crash.

The master-slave structure is one of the characteristics of HDFS, and an HDFS cluster often has only one master server. The core technology of HDFS is to manage the data storage naming and block storage processing of the entire cluster on the master node to respond to customer access requests. In addition, the distributed data storage nodes of the cluster are also responsible for executing block creation, deletion, and copy instructions, which generally come from the cluster namespace management node. The system composition of the distributed cluster namespace management node of HDFS and the distributed data storage node of the cluster is shown in Figure 1.

The Namenode of the cluster, the cluster namespace management node, is responsible for recording the block location of each file distributed storage and controls the Datanodes, the data storage node, to complete the writing and reading of data. The software system of the cluster namespace management node and the distributed data storage node of the cluster is designed to run on general commercial machines. HDFS has native support for Java and can run Java code very efficiently. Using the Java language can bring a high degree of portability, which means that HDFS can be deployed in a very wide range of computing. A normal Hadoop cluster generally has only one namespace management node, and at the same time, there are corresponding separate cluster-decentralized data storage nodes on other computers in the cluster. Under special circumstances, HDFS also supports one node to run multiple virtual clusters of scattered data storage nodes and coexist with namespace nodes. The existence of a cluster namespace management node in the cluster greatly simplifies the system architecture. The cluster namespace management node is the arbitrator of all HDFS and the metadata of the repository. Because the system is designed to construct the cluster in such a way, user data will not flow through the cluster namespace management node.

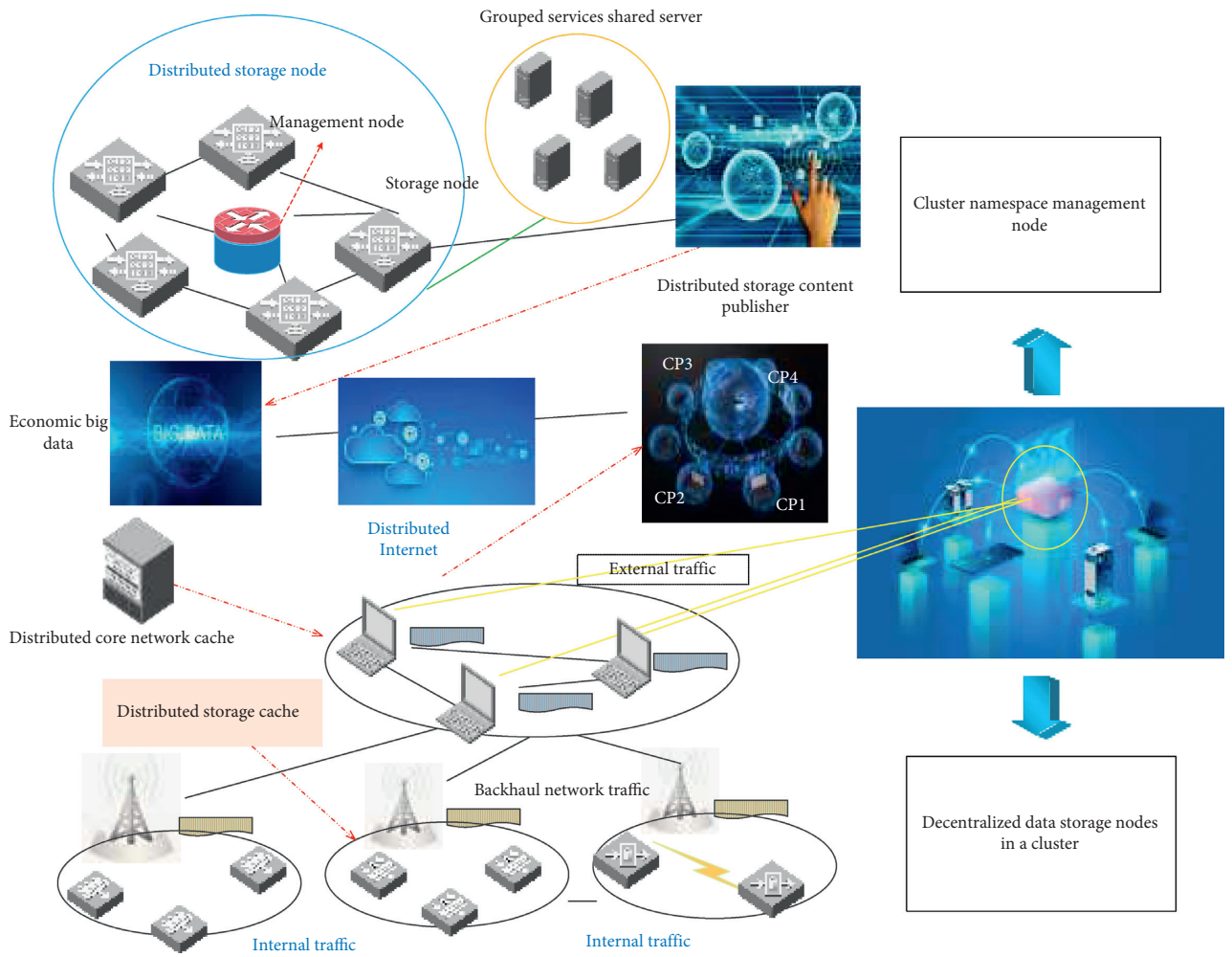

Figure 1: HDFS distributed cluster namespace management node and cluster-decentralized data storage node system composition.

One of the characteristics of HDFS is that it has powerful data blocks, that is, a powerful distributed data shard placement strategy, and the placement strategy is based on a lot of experience and operating history of traditional file systems. The cluster name space management node identifies and identifies which physical location in the cluster the scattered data storage nodes of each cluster belong to, so as to identify the network topology according to the physical location and ensure optimal data storage.

3.2. Nonrelational Database Technology. Not Only SQL (NoSQL) generally refers to nonrelational databases. In the Internet 2.0 era, traditional relational databases will become very weak when dealing with extremely large-scale website data and become a bottleneck for data analysis. However, nonrelational databases have received people's attention and development due to their special characteristics. NoSQL database was created to solve the challenges brought by large-scale datasets and multiple data types, especially the data processing problems of big data. NoSQL databases are roughly divided into 4 categories, namely, key-value storage databases, columnar storage databases, document databases, and graph databases.

The databases that are suitable for economic big data are also HBase and Cassandra. Cassandra is designed to handle large data workloads that span multiple nodes without a single point of failure. Its architecture is based on understanding that system and hardware failures can occur. The problem of Cassandra address failure is to use a point-to-point distributed system with homogeneous node data distributed across all nodes in the cluster. Each node frequently exchanges status information itself, and other nodes use a point-to-point gossip communication protocol in the cluster. You write in sequence on each node and submit the log to capture write activity to ensure data durability. Then you index the data and write it into a memory structure called memtable, which is similar to write-back cache. Each time the memory structure is complete, data is written to the disk SSTable data file. All write operations are automatically partitioned and replicated throughout the cluster. Cassandra regularly consolidates SSTables using this process called compaction, which discards obsolete data marked as deleted tombstones. Various repair mechanisms across clusters to ensure that all data remains consistent. The schematic diagram of the economic big data distributed computing analysis platform is shown in Figure 2.

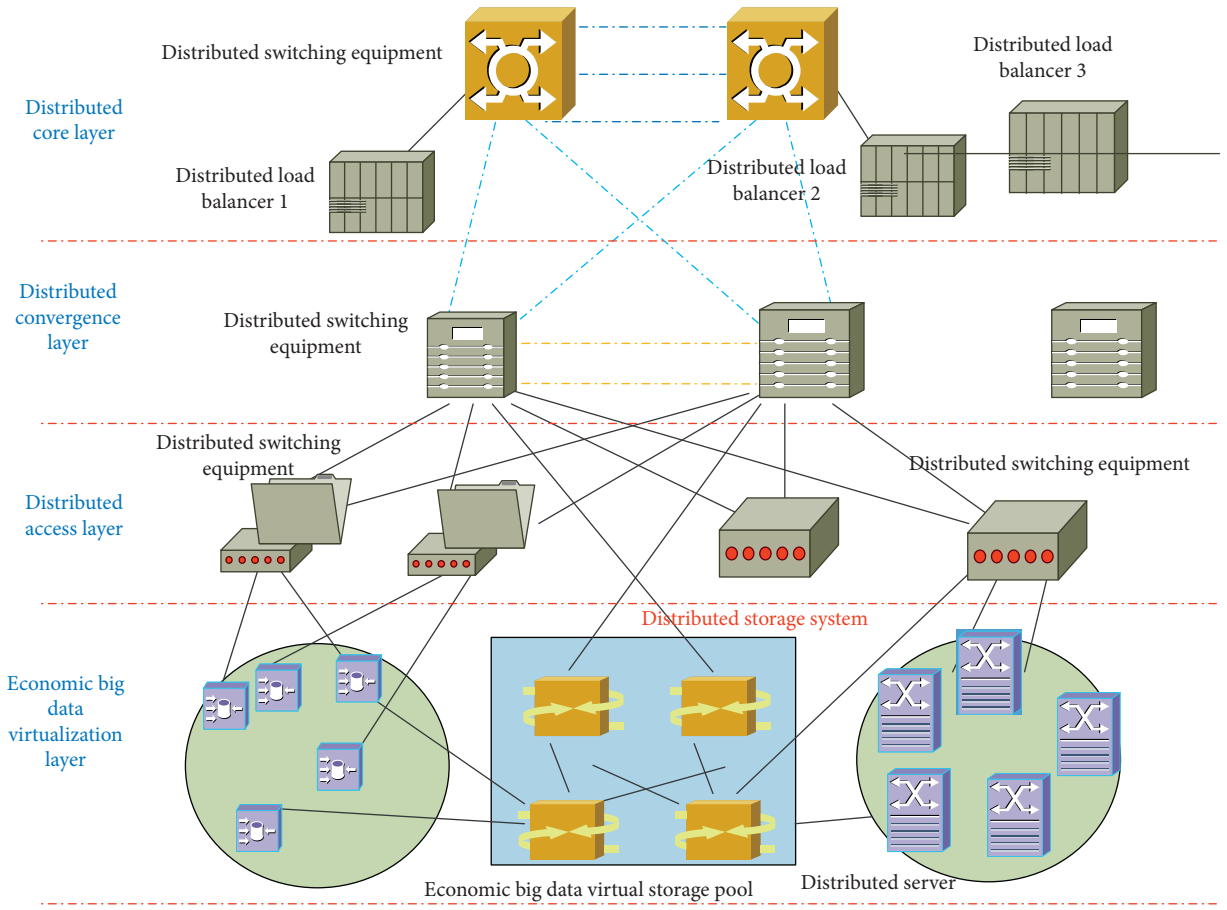

Figure 2: Schematic diagram of distributed computing analysis of economic big data.

Cassandra is a partitioned row storage database, and the rows are organized into the primary key required by the table. Cassandra's architecture allows any authorized user to connect to any data center and node to access data using Continuous Query Language (CQL). For ease of use, CQL uses SQL-like syntax and table data. Developers can access CQL, Dev Center through cqlsh, through the language of drivers and applications. Usually, a cluster has a number of different tables for each application.

Client read or write requests can be sent to any node in the cluster. When a client requests to connect to a node, the node acts as the coordinator of a specific client operation. The coordinator acts as a proxy for data requirements between client applications and nodes. The coordination in the ring determines which nodes should be requested based on the cluster configuration.

HBase is a "NoSQL database. "NoSQL" is a general term meaning that Relational DataBase Management System (RDBMS) databases do not support SQL as its main access language, but there are many types of NoSQL databases: an example of Berkeley DB's local NoSQL database. HBase is a distributed database. Technically speaking, HBase is indeed a "data storage" concept more accurate than the concept of "database" because it lacks many functions, especially compared to RDBMS, such as type column, secondary indexes, triggers, and advanced query languages.

3.3. Big Data Visualization Technology. The main principle of big data visualization technology is to display the results of big data processing and analysis through technical means in a variety of ways and with rich effects, so as to assist analysts in deeper understanding and mining of data results. Among them, big data visualization technology is very rich. From traditional data Office software to MATLAB, it can be used as big data visualization work. Here, the HyperText Markup Language (HTML) 5 selected by the platform is selected for a brief introduction because HTML5, as a new RIA implementation solution, can be used for big data visualization, and the front-end part of this platform also selects HTML5 as one of the implementation technologies. HTML5 is a markup language used to build and present content on the World Wide Web.

Its core goal is to improve the latest multimedia language support while keeping it simple and readable by humans and a consistent understanding of computers and devices (web browsers, parsers, etc.). The purpose of HTML5 is to include not only HTML 4, but also XHTML 1 and HTML Document Object Model (DOM) level 2. After its predecessors HTML 4.01 and XHTML 1.1, HTML5 is a reflection of the commonly used HTML and XHTML on the World Wide Web with a mixture of characteristics introduced by various specifications, together with the introduced software products, such as Web browsers and established conventions. This is also an

attempt to define a separate markup language, which can be written in HTML or XHTML. It includes detailed processing models to encourage more interoperable implementations; it extends, improves, and closes markup available for documentation and introduces markup and application programming interface (api) for complex web applications.

For the same reason, HTML5 is also a potential candidate for cross-platform mobile applications. Many features of HTML5 are designed with low-power devices such as smart phones and tablets under consideration. In particular, HTML5 adds many new syntax features. These functions are to facilitate the inclusion and processing of network multimedia and graphic content without resorting to proprietary plug-ins and APIs. Other new page structure elements are designed to enrich the semantic content of the document. New attributes are introduced, some elements and attributes have been removed, and some elements have been redefined or standardized. APIs and Document Object Model (DOM) are no longer dispensable, but basic parts of the HTML5 specification defines some details to deal with invalid documents, so grammatical errors will be unified by all standard-compliant browsers and other user agents.

4. Distributed Storage Strategy Design with Temporal and Spatial Semantic Constraints

4.1. Economic Big Data Storage Strategy. Based on the massive characteristics of economic big data, traditional centralized storage can no longer meet the storage of such a large amount of data, so this article adopts a distributed storage strategy. Distributed storage is different from traditional centralized storage. It stores data slices on distributed storage nodes through data partitioning, thereby realizing distributed storage management of massive data.

Compared with distributed storage, the traditional centralized storage system uses a centralized storage server to store all data. The reliability, security, and scalability of the storage server become the performance bottleneck of the system and cannot meet the storage requirements of massive spatiotemporal data. The distributed storage system uses the horizontal expansion method to connect multiple distributed servers through the network or other means. Each distributed server stores a part of the block data obtained by data partitioning, so as to make the system reliable and usable. And storage performance has been significantly improved, while storage management costs are reduced, and there is no storage space limitation.

Distributed storage strategy provides distributed storage solutions for distributed systems, so that data can be stored on distributed storage devices through reasonable data partitioning. A good data partitioning strategy can effectively avoid possible data tilt problems in distributed storage systems. Therefore, the data partition method becomes the core of the distributed storage strategy, which divides the massive data into different devices in a balanced manner, so that the distributed storage system realizes the data load balance.

Since economic big data contains a large amount of video, image, remote sensing image, and trajectory data, these data contain rich time, space, and semantic information. Therefore, this article will comprehensively consider the space, time, and semantic information of economic big data and propose economic big data. Based on the relevant characteristics of economic big data and the above analysis, this article will focus on the distributed storage strategy of temporal and spatial semantic constraints.

The construction of the space filling curve is based on the premise of the sequential traversal of spatial discrete units. The discrete units adjacent in space are still adjacent on the filling curve, so the space filling curve has good spatial proximity characteristics. Therefore, it is used for the division of spatial data to ensure the proximity of spatial objects after the division of spatial data.

The reason why economic big data can be stored in a distributed manner is because of the inherent relevance of economic big data. The tighter the correlation between the data, the more accurate the distributed storage. The calculation scheme of Wiener distributed memory is given below.

The true value is

$$V_d(n) = s|N+n|. \tag{1}$$

The distributed storage value is

$$V(n) = \lim_{n\longrightarrow\infty} \sum_{i=1}^{n} x(N-m+i)\cdot h(m-1). \tag{2}$$

The error is

$$e(N+n) = s(N+n) - \widehat{s}(N+n). \tag{3}$$

As mentioned above, this article uses the principle of minimum mean square error to calculate distributed memory, so to minimize the mean square error, we must take the derivative of $h(m)$ and set it to zero. The h_j here is $h(m)$.

$$\frac{\partial E\left(e(N+n)^2\right)}{\partial h_j} = 0. \tag{4}$$

So we launched

$$E\left[\sum_i x_i \cdot x_j \cdot h_i - s(N+n)\right] = 0, \tag{5}$$

which is

$$R_{xyd}(k) = \lim_{m\longrightarrow\infty} \sum_{i=1}^{m} r_{xx}\cdot (i-k)\cdot h_{\text{opt}}(i). \tag{6}$$

We launched

$$R_{xyd}(k) = E(s(N+n+k)\cdot x(n-1+k)). \tag{7}$$

The transformation to Z domain is

$$s_{xyd}(Z) = Z^N \cdot S_{xs}(Z). \tag{8}$$

Therefore, the best solution for introducing noncausal Wiener distributed memory is

$$H_{\text{opt}}(z) = z^N \cdot \frac{S_{xs}(Z)}{S_{xx}(Z)}. \tag{9}$$

4.2. Spatiotemporal Multilevel Division Strategy of Economic Big Data. Data partition is the basis of distributed storage of data, and it is also a key factor affecting the performance of distributed database systems. Therefore, the core of distributed storage strategy is to establish a data partition method with temporal and spatial semantic constraints. Data partitioning is to divide the data into multiple independent, relatively small subdata blocks through certain partitioning rules and then store them on each distributed storage node, providing a basis for distributed query and analysis. Based on the analysis of existing spatial data division methods, this paper proposes a multilevel data division strategy based on the combination of Geo Hash and Hilbert curve.

Traditional data partitioning methods mainly consider data load balance and data integrity, while economic big data has rich spatial, temporal, and semantic relationships. Therefore, different from the traditional data partition method, the data partition method in this article not only needs to consider the data load balance of each node, but also needs to take into account the spatiotemporal and semantic relationship between economic big data.

Spatial proximity refers to the closeness of objects in space. Spatial proximity is a commonly used spatial relationship in spatial analysis. If the data in the same storage node has a certain spatial proximity, the amount of data exchange and the number of disk accesses of the node is significantly less when querying in a small space, and the I/O efficiency is higher; the partition algorithm and data distribution method distribute the spatially adjacent data blocks to different nodes, and multiple nodes can be used to provide multiple I/O during query, which improves the efficiency of data access and query. Time correlation refers to the proximity of data in time series. Different from other spatial data, economic big data requires not only the analysis and query of spatial proximity, but also the query and analysis of time-related data. The schematic diagram of distributed storage information interaction is shown in Figure 3.

Semantics refers to the meaning contained in the objects in the data. Because economic big data has multimodal characteristics, for the same spatial object, the association between multimodal data can be established through similar semantics, and multiple descriptions of the spatial object can be obtained, which is conducive to data analysis and in-depth mining.

Data integrity means that after data storage is completed, the data value in the data storage system should be the same as before storage, and in distributed storage, the total amount of data of each storage node should be the same as the total amount of data before storage. Data integrity is the basic requirement of data division. If data integrity is lost after data division, the data division method is considered invalid.

In order to ensure the performance of the distributed storage system, the data load balance of each distributed node should be ensured as much as possible when storing data. Compared with structured data of the same size, spatial data is mainly unstructured data with uneven distribution of data volume. Therefore, when storing spatial data, load balancing does not use the balanced number of data elements as the criterion, but uses the data of each node. Data volume balance is used as a basis for measuring load balance. This requires that in a distributed storage system, the data partitioning strategy should achieve load balancing of data partitions as much as possible, thereby improving the overall load balancing of the distributed storage system.

4.3. Attribute Division Method. The attribute division method is to group the field values according to a specific relationship according to a certain field of the data and divide the data of the same group into the same node. Common attribute division methods include range division method, round-robin method, and simple hash method. Due to the uneven size of the data metadata of spatial data, the use of attribute division method is prone to data skew. For example, the range division method divides the data according to the range of field values. Therefore, when the field distribution is uneven, there may be a large number of data elements in some ranges and a small amount of metadata in some range data, which results in data skew. In addition, the attribute division method does not consider the spatial relationship of spatial data, which is likely to cause the separation of the temporal and spatial semantic relationship of spatial data. The round-robin method divides data elements into different data blocks in turn to ensure that the number of data elements in the data block is approximately equal, but for unstructured data, due to the uneven size of a single data element, the data division of this method is more balanced. The simple hash method usually first performs a hash operation on the divided fields and then takes the remainder of the operation result to determine the result of the division. It is simple and efficient. Similar to the round-robin method, the simple hash method is suitable for fixed size.

4.4. K-Means Clustering Method. The K-Means clustering method mainly uses the clustering method to make the sum of square errors of each spatial object to be divided from the cluster center of the data block where it is located to the minimum.

We read the position coordinate value of each spatial object in the spatial dataset; if it is a line or area object, the smallest bounding rectangle is used to calculate the coordinate value of its center point; according to the set number of data blocks k, we randomly generate k data block centers, calculate the distance from the coordinate value of each space object to the center of each data block, and divide it into the closest data block; according to the current division of each data block, we recalculate the data block center and then iterate repeatedly until the center of the data block no longer changes or the sum of squared errors meets the requirements.

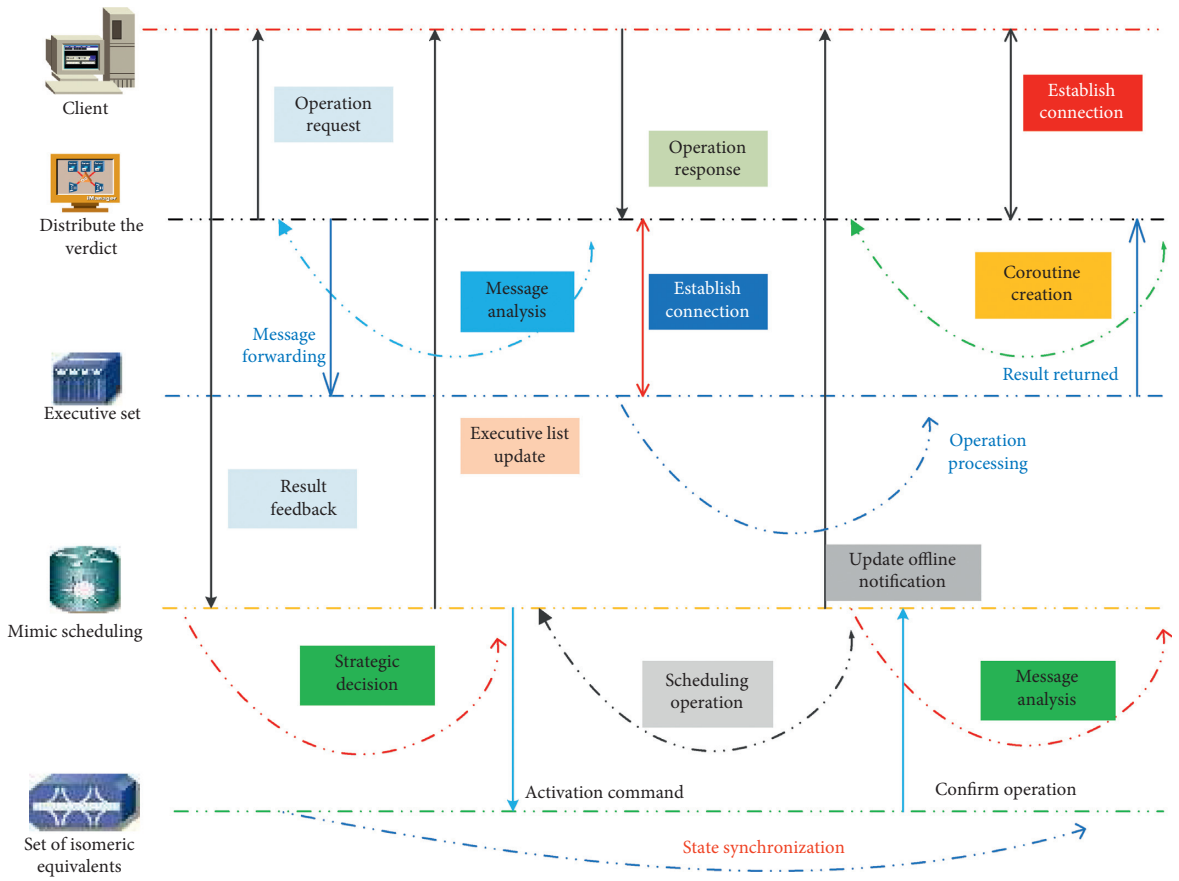

Figure 3: Schematic diagram of distributed storage information interaction.

Although K-Means clustering can keep the spatial data of the same partition with a certain spatial proximity and balance the amount of data at each node through a certain data distribution method, the selection of the initial cluster center and noise and isolated points will seriously affect the clustering of each spatial data block after K-Means clustering results in poor clustering effect, thereby reducing the clustering within the data block.

4.5. Multilevel Data Division Based on the Combination of Geohash and Hilbert. Since the attribute division method is not suitable for unstructured spatial data, the algorithm complexity of the K-Means clustering method is too high. Therefore, this paper studies the spatial data division method based on the space filling curve method. Considering the time correlation and semantic similarity of economic big data, this paper proposes a multilevel spatial data division strategy based on the combination of Geohash and Hilbert. The algorithm flow of multilevel data partition based on the combination of Geohash and Hilbert is shown in Figure 4.

5. Visual Experiment and Analysis

5.1. Economic Big Data Load Generation. Due to the hardware limitations of OpenStack, we used Docker for this experiment platform, used containers as virtual machine instances, and used 3 ordinary PCs as load generators to build NoSQL storage clusters on the private cloud platform. The NoSQL cluster chose to use the open-source Cassandra. Cassandra has a good copy mechanism and a configurable data migration scheme, so we choose to use Cassandra.

Here we use the workload simulation model to evaluate the algorithm. The main goal of this part is to implement a load generator. We use python to simulate and implement this workload generator under different request types. The workload is simulated from the access log of the economic big data. The access log of the economic big data contains the number of visits per hour for each page. First, we need to prepare the cleaning log data to remove the noise data in the log. Mainly we remove some meaningless pages such as homepage, search page, etc., which occupy a large part of log records. After that, we select the most visited 10% of the remaining records and discard the remaining pages. Mainly because 10% of the main visited pages account for 80% of the entire workload, and it is more meaningful to study these 10% of the most visited pages than to study the remaining 90% of the pages because many of the remaining pages are only visited one per hour to the inactive page twice. After cleaning the required workload data, because the economic big data only provides page browsing and does not provide write operations, we randomly select 5% of the sorted data to convert into write operations. After the workload of economic big data, we summarized the relevant parameters of writing our own workload generator as shown in Figure 5.

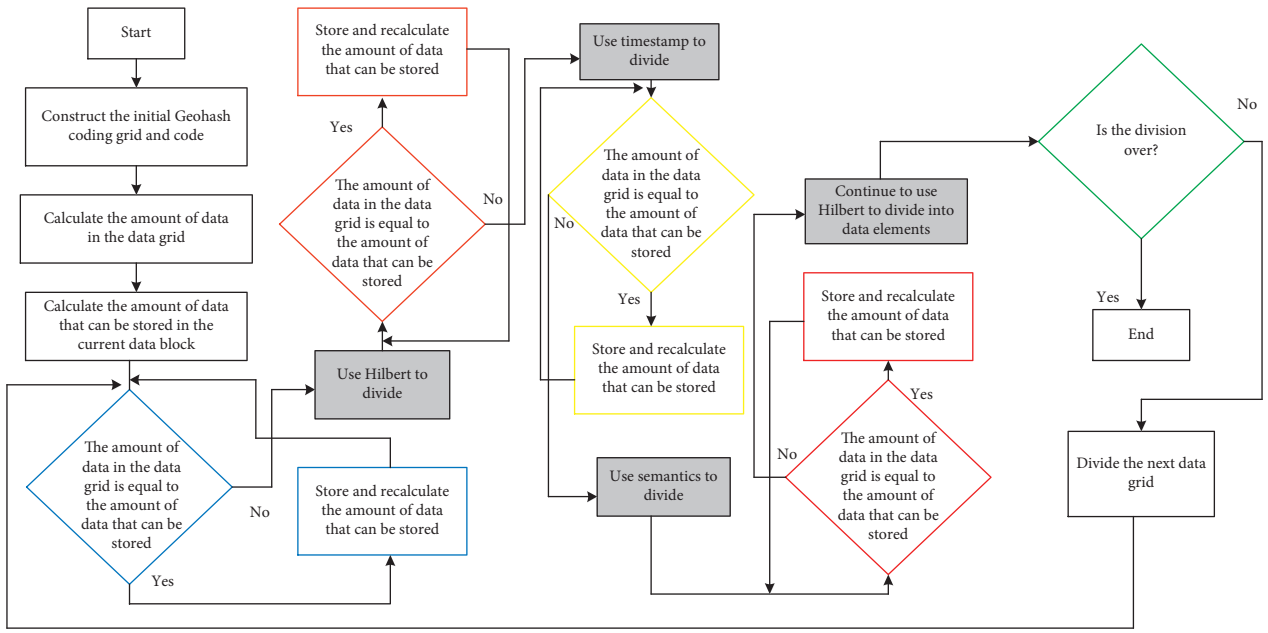

Figure 4: Flowchart of multilevel data partition algorithm based on the combination of Geohash and Hilbert.

We set up the load generator according to the parameters and send the compiled workload file to the load generator. The load generator initiates read and write requests to the NoSQL cluster based on the characteristics of the files we have organized. Our experimental environment uses 30 PCs as load generators, 20 of which initiate read requests and 10 initiate write requests. The thread of each client is set to 10 in this experiment. All the keys in the storage cluster have the same format (key1 ,. . ., key100000), and the value is a string of random length. Here, we select the monitoring data of read request access workload collected by 6 load generators for visual display, as shown in Figure 6. The workload fluctuates on a daily basis. We can regard the change of the workload as a stable and random process, so we can use the principle of Wiener distributed storage for distributed storage.

5.2. Visualization of Distributed Storage Workload. In this article, the accuracy of distributed storage of workloads is the key to whether the entire automatic deployment framework is reasonable, and it is also the core part of this article. The load distributed storage algorithm is a Wiener distributed memory derived from the extrapolation method in the principle of Wiener filtering. The process of distributed storage is to first divide the time series of the workload into several distributed storage windows (PW) in units of one hour. Each time is the distributed storage starting in the current distributed storage window and the next distributed storage workload of the window.

Based on the ideal polymerization time, we adjusted the polymerization time to 1 hour and performed visual analysis with 240-hour observations to highlight the experimental effect. We use aggregated workloads for distributed storage. After many tests, the distributed storage effect is best when the length of Wiener distributed storage is set to 2.

Figure 7 shows the distributed storage effect of the workload after the workload type of the simulated economic big data has been reduced year-on-year. From the figure, we can clearly understand that the distributed storage workload algorithm studied in this article can well complete the distributed storage task.

5.3. Visualization of Performance Evaluation. We select the actual value and distributed storage value of 50 distributed storage windows for evaluation. Our distributed storage strategy is the trend of correct distributed storage of workload, that is, the resource configuration of the storage cluster can be adjusted according to the workload of distributed storage. It only needs to verify whether the cluster violates the SLA standard of the agreement after adjusting the resource configuration using distributed storage workloads.

In accordance with the workload distributed storage algorithm distributed storage workload, this paper adjusts the cluster performance after the resource configuration of the NoSQL cluster in real time. It is mainly to monitor the response time of the NoSQL cluster to client read and write

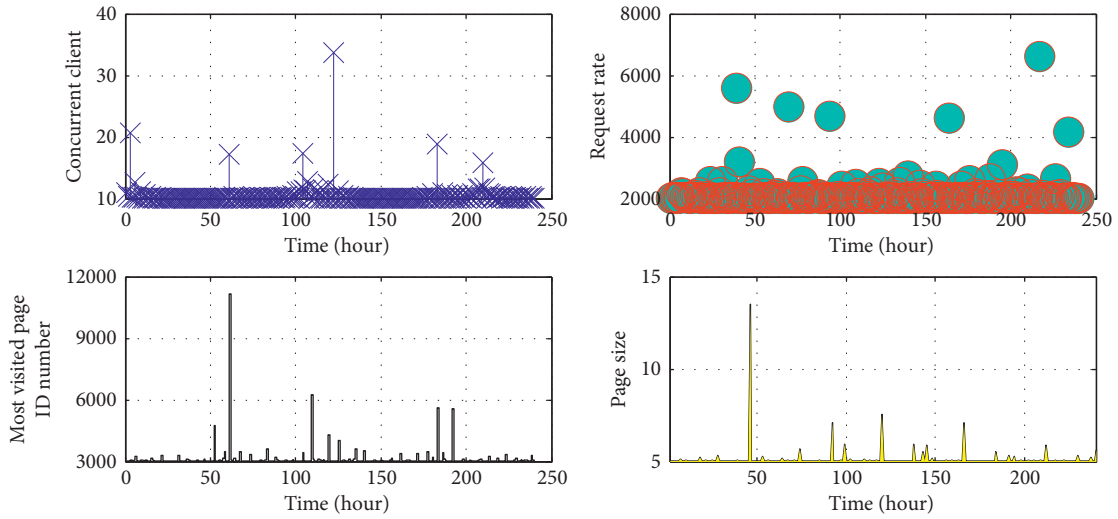

Figure 5: Visualization of load generation parameters.

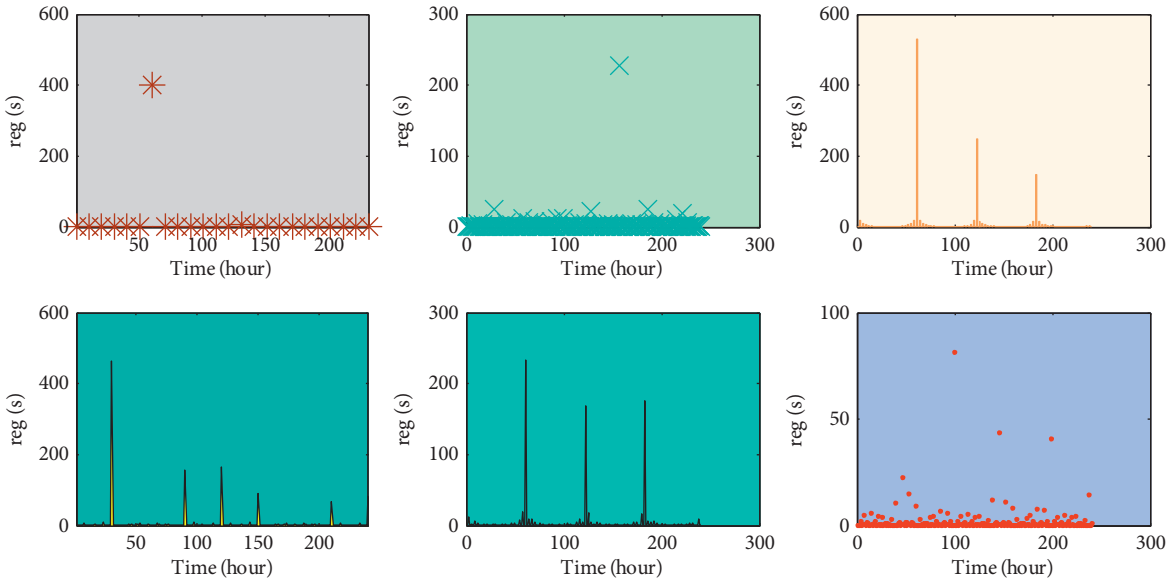

Figure 6: The load effect diagram generated by the 6 load generators in the experiment.

requests and the CPU utilization of the entire cluster to observe the performance of the cluster.

Figure 8 shows the CPU utilization before and after adding an instance in a certain distributed storage window of the monitoring cluster. It can be clearly seen that when the request load is too high to add a storage instance, the CPU utilization of the cluster will be significantly reduced until an average level. Here we need to count the data after one minute because the cluster will make a data migration plan after adding storage nodes, and the data of each node has reached a relative balance.

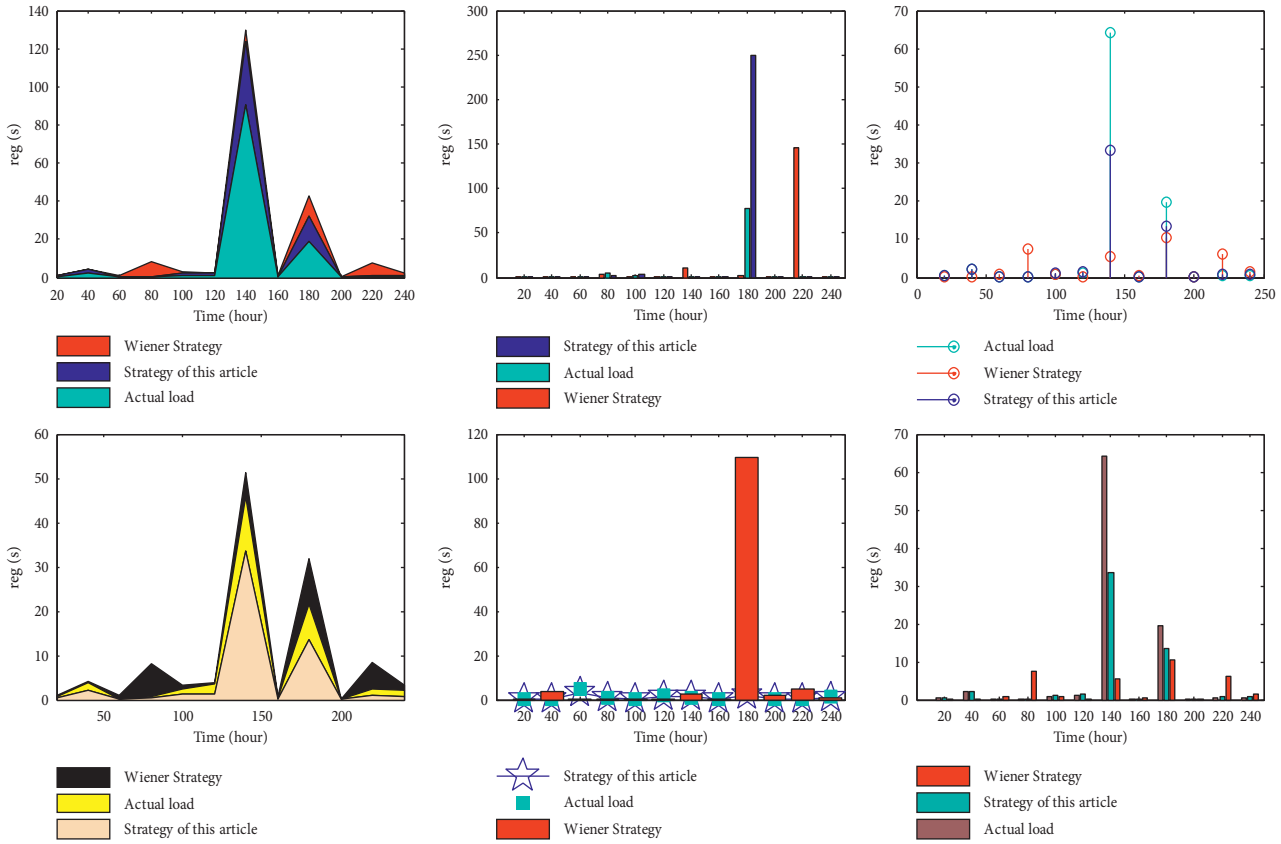

Figure 7: The economic big data workload and distributed storage effect of the 6 load generators in the experiment.

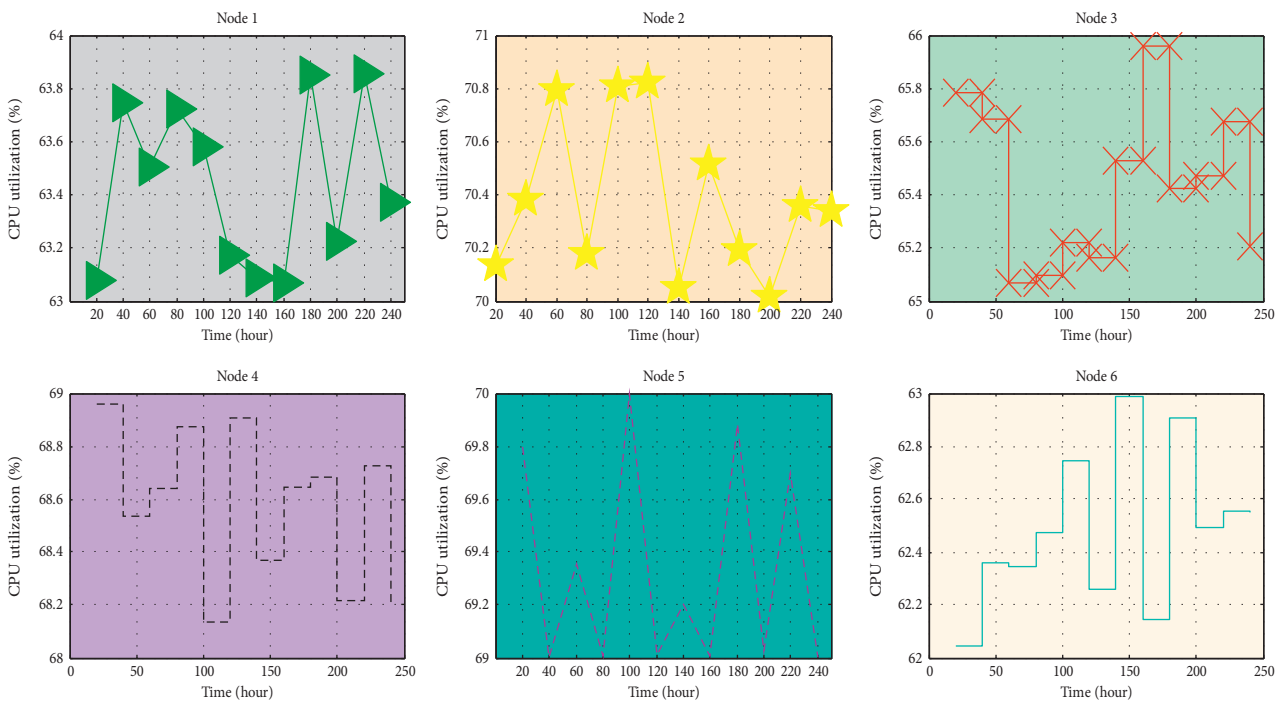

Figure 8: CPU monitoring after adding nodes 1 to 6.

6. Conclusion

This paper proposes a distributed storage strategy with temporal and spatial semantic constraints. In view of the massive characteristics of multisource economic big data, from the perspective of time, space, and semantics, the current mainstream data division methods are analyzed, and the advantages and disadvantages of each method are compared, and a multilevel data division based on the combination of Geohash and Hilbert is proposed. On this basis, with the time stamp and semantic coding table, a unified coding method with temporal and spatial semantic constraints is established, which provides data storage for the efficient distributed storage of massive multisource economic big data and the cross-modal management, analysis, and application of economic big data. In this article, we designed a set of automatic deployment frameworks for NoSQL data clusters on cloud platforms. The core is to use distributed storage workloads to deploy the resource configuration of the cluster in advance, so that the data cluster can save the use of storage resources and save customer costs while meeting SLA restrictions. The purpose can be well achieved by means of distributed storage workloads. However, there are always errors in distributed storage, and the errors in distributed storage may be large at some point, which poses a great challenge to our research. In the actual economic big data environment, because there are many sensitive factors, resource allocation is more stringent. Since the workload will be different, the conditions of distributed storage will also be different, so we need to adjust our distributed storage strategy according to the characteristics of the workload. The multilevel data partition method based on the combination of Geohash and Hilbert only considers the temporal, spatial, and semantic associations between the data. Due to the richness and complexity of data semantics, this paper only considers simple semantics in the process of data partitioning. Further research on the semantic constraints of data partitioning can be carried out to make semantic constraints more abundant and accurate. This will help in-depth analysis and mining of massive economic big data.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this study.

References

[1] F. Chen, M. Chen, Z. Xu, J. M. Guerrero, and L. Y. Wang, "Distributed noise-resilient economic dispatch strategy for islanded microgrids," *IET Generation, Transmission & Distribution*, vol. 13, no. 14, pp. 3029–3039, 2019.

[2] B. Bai, S. Xiong, B. Song, and M. Xiaoming, "Economic analysis of distributed solar photovoltaics with reused electric vehicle batteries as energy storage systems in China," *Renewable and Sustainable Energy Reviews*, vol. 109, pp. 213–229, 2019.

[3] A. Mohamed, M. K. Najafabadi, Y. B. Wah, E. A. K. Zaman, and R. Maskat, "The state of the art and taxonomy of big data analytics: view from new big data framework," *Artificial Intelligence Review*, vol. 53, no. 2, pp. 989–1037, 2020.

[4] M. Resch, J. Bühler, B. Schachler, R. Kunert, A. Meier, and A. Sumper, "Technical and economic comparison of grid supportive vanadium redox flow batteries for primary control reserve and community electricity storage in Germany," *International Journal of Energy Research*, vol. 43, no. 1, pp. 337–357, 2019.

[5] S. Wang and Z. Fu, "Thermodynamic and economic analysis of solar assisted CCHP-ORC system with DME as fuel," *Energy Conversion and Management*, vol. 186, pp. 535–545, 2019.

[6] A. G. Polyakova, M. P. Loginov, and E. V. Strelnikov, "Managerial decision support algorithm based on network analysis and big data," *International Journal of Civil Engineering & Technology*, vol. 10, no. 2, pp. 291–300, 2019.

[7] M. J. Ghadi, S. Ghavidel, A. Rajabi, A. Azizivahed, L. Li, and J. Zhang, "A review on economic and technical operation of active distribution systems," *Renewable and Sustainable Energy Reviews*, vol. 104, pp. 38–53, 2019.

[8] R. Dubey, A. Gunasekaran, S. J. Childe, C. Blome, and T. Papadopoulos, "Big data and predictive analytics and manufacturing performance: integrating institutional theory, resource-based view and big data culture," *British Journal of Management*, vol. 30, no. 2, pp. 341–361, 2019.

[9] B. P. Bhattarai, S. Paudyal, Y. Luo et al., "Big data analytics in smart grids: state-of-the-art, challenges, opportunities, and future directions," *IET Smart Grid*, vol. 2, no. 2, pp. 141–154, 2019.

[10] S. Gupta, S. Modgil, and A. Gunasekaran, "Big data in lean six sigma: a review and further research directions," *International Journal of Production Research*, vol. 58, no. 3, pp. 947–969, 2020.

[11] S. Garg, A. Singh, K. Kaur et al., "Edge computing-based security framework for big data analytics in VANETs," *IEEE Network*, vol. 33, no. 2, pp. 72–81, 2019.

[12] R. Kesharwani, Z. Sun, C. Dagli, and H. Xiong, "Moving second generation biofuel manufacturing forward: investigating economic viability and environmental sustainability considering two strategies for supply chain restructuring," *Applied Energy*, vol. 242, pp. 1467–1496, 2019.

[13] R. M. Bennett, M. Pickering, and J. Sargent, "Transformations, transitions, or tall tales? A global review of the uptake and impact of NoSQL, blockchain, and big data analytics on the land administration sector," *Land Use Policy*, vol. 83, pp. 435–448, 2019.

[14] A. Lorestani, G. B. Gharehpetian, and M. H. Nazari, "Optimal sizing and techno-economic analysis of energy- and cost-efficient standalone multi-carrier microgrid," *Energy*, vol. 178, pp. 751–764, 2019.

[15] E. Hossain, I. Khan, F. Un-Noor, S. S. Sikander, and M. S. H. Sunny, "Application of big data and machine learning in smart grid, and associated security concerns: a review," *IEEE Access*, vol. 7, pp. 13960–13988, 2019.

[16] L. Yao and Z. Ge, "Distributed parallel deep learning of Hierarchical Extreme Learning Machine for multimode quality prediction with big process data," *Engineering Applications of Artificial Intelligence*, vol. 81, pp. 450–465, 2019.

[17] M. Huang, W. Liu, T. Wang et al., "A game-based economic model for price decision making in cyber-physical-social systems," *IEEE Access*, vol. 7, pp. 111559–111576, 2019.

[18] G. Arbia, C. Ghiringhelli, and A. Mira, "Estimation of spatial econometric linear models with large datasets: how big can spatial Big Data be?" *Regional Science and Urban Economics*, vol. 76, pp. 67–73, 2019.

[19] A. Colmenar-Santos, A.-M. Muñoz-Gómez, E. Rosales-Asensio, and Á. López-Rey, "Electric vehicle charging strategy to support renewable energy sources in Europe 2050 low-carbon scenario," *Energy*, vol. 183, pp. 61–74, 2019.

[20] M.-W. Tian, H.-C. Yuen, S.-R. Yan, and W.-L. Huang, "The multiple selections of fostering applications of hydrogen energy by integrating economic and industrial evaluation of different regions," *International Journal of Hydrogen Energy*, vol. 44, no. 56, pp. 29390–29398, 2019.

[21] R. Hou, Y. Kong, B. Cai, and H. Liu, "Unstructured big data analysis algorithm and simulation of Internet of Things based on machine learning," *Neural Computing & Applications*, vol. 32, no. 10, pp. 5399–5407, 2020.

[22] J. Grayson, "Big data analytics and sustainable urbanism in Internet of Things-enabled smart governance," *Geopolitics, History, and International Relations*, vol. 12, no. 2, pp. 23–29, 2020.

[23] X. Yan and L. Jingyi, "Analyze the changes and countermeasures of economic responsibility audit in the era of big data," *Academic Journal of Business & Management*, vol. 3, no. 2, pp. 89–92, 2021.