International Journal of *Geo-Information*

MDPI

Article

A Trajectory Big Data Storage Model Incorporating Partitioning and Spatio-Temporal Multidimensional Hierarchical Organization

Zhixin Yao [1,2], Jianqin Zhang [1,2,*], Taizeng Li [1,2] and Ying Ding [1,2]

[1] School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, Beijing 102616, China
[2] Key Laboratory of Urban Spatial Information, Natural Resources Ministry, Beijing 102616, China
* Correspondence: zhangjianqin@bucea.edu.cn

Abstract: Trajectory big data is suitable for distributed storage retrieval due to its fast update speed and huge data volume, but currently there are problems such as hot data writing, storage skew, high I/O overhead and slow retrieval speed. In order to solve the above problems, this paper proposes a trajectory big data model that incorporates data partitioning and spatio-temporal multi-perspective hierarchical organization. At the spatial level, the model partitions the trajectory data based on the Hilbert curve and combines the pre-partitioning mechanism to solve the problems of hot writing and storage skewing of the distributed database HBase; at the temporal level, the model takes days as the organizational unit, finely encodes them into a minute system and then fuses the data partitioning to build spatio-temporal hybrid encoding to hierarchically organize the trajectory data and solve the problems of efficient storage and retrieval of trajectory data. The experimental results show that the model can effectively improve the storage and retrieval speed of trajectory big data under different orders of magnitude, while ensuring relatively stable writing and query speed, which can provide an efficient data model for trajectory big data mining and analysis.

Keywords: trajectory data; spatio-temporal coding; distributed columnar storage; Hilbert partition; HBase write hot

Citation: Yao, Z.; Zhang, J.; Li, T.; Ding, Y. A Trajectory Big Data Storage Model Incorporating Partitioning and Spatio-Temporal Multidimensional Hierarchical Organization. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 621. https://doi.org/10.3390/ijgi11120621

Academic Editors: Hartwig H. Hochmair and Wolfgang Kainz

Received: 5 September 2022
Accepted: 11 December 2022
Published: 13 December 2022

1. Introduction

As a very important spatio-temporal big data, trajectory data contain temporal, spatial and attribute information, which not only has the characteristics of multi-source, massive and fast update, but also presents the complexity of multi-dimensional, semantic and spatio-temporal dynamic correlation [1]. At present, traditional relational spatial databases fail to meet the storage and query needs of large-scale spatial data [2], so distributed, non-relational spatial databases are born [3]. However, the traditional big data technologies in the existing distributed storage systems use a direct-write columnar storage mechanism and traversal of massive data retrieval [4], both of which have some degree of problems in big data processing techniques. Among them, the direct columnar storage mechanism leads to data "write fever" and data skewing, and the storage efficiency is low, which cannot meet the storage requirements of large-scale spatial data. At the same time, in the process of data retrieval, the traversal of data tables causes problems such as high overhead and slow retrieval speed of I/O computation. Although the above problems can be solved by using spatial data division and hierarchical organization, there are still many problems when taking into account the temporal continuity and spatial proximity of data objects. Therefore, in order to effectively balance the integrity, spatial proximity and data structure flexibility of trajectory data [1] and meet the needs of balancing the data among distributed nodes, the targeted organization and management of data from temporal and spatial perspectives is an important objective for the efficient storage and query of massive data [4].

In recent years, to improve the organization and management of spatial big data, many scholars have conducted a significant amount of research in the field of spatial data partitioning. Geohash coding and spatially filled curves are two commonly used methods for effective data dimensionality reduction partitioning. Geohash, as a spatial partitioning-based indexing model, is widely used for the indexing and retrieval of spatial data [5]. The literature [6–8] used Geohash strings as partitioning keys to partition and index large-scale geographic data, which effectively improved the data retrieval efficiency and accuracy; the literature [9] developed Geohash-Grid Tree for searching integrated data in heterogeneous data sources to cope with the data retrieval problem in mobile sensing environments. All the above studies have been validated effectively in terms of retrieval speed, but the data volume of the storage nodes cannot be balanced and the storage efficiency has not been illustrated accordingly.

Space-filling curves mostly use Hilbert curves and Z-curves to achieve data partitioning by dimensionality reduction. Generally speaking, Hilbert curve data have good clustering characteristics, but the mapping process is complicated. In contrast, Z-curve data have slightly worse clustering properties, but the mapping process is simple. In the literature [10], a new multi-level spatio-temporal grid index GeoSOT ST-index was constructed based on Z-curves to organize the trajectory data hierarchically, but it failed to take into account the uneven distribution of trajectory data.

In contrast to the wide application of Z-curves, there are relatively few studies using Hilbert curves for spatial data partitioning. Cao [4] and Wu [11] et al. used Hilbert curves to partition the space into small cells to cover the region of interest where the trajectory dataset is located with a grid, and mapped the trajectory data into different cells for retrieval. Based on the state view, Jiang et al. [12] proposed an efficient coding algorithm JFK-3HE and a decoding algorithm JFK-3HD to reduce the complexity of coding and decoding and improve the efficiency of coding and decoding of 3D Hilbert curves. Jiang and Wu et al. [12,13] focused on the study of 3D Hilbert curve cascade evolution and coding efficiency, and Jia et al. [14] improved the 2D Hilbert coding and decoding algorithm from the perspective of accounting for data skewing, but none of them fundamentally changed the status quo of data skewing. Yang et al. proposed an adaptive spatial partitioning method and Geohash coding point pair filtering strategy, which combine to accelerate the computational efficiency of the K function, but the repeated coding increases the time overhead [15]. Wu et al. used the state vector and its hierarchical evolution and degradation function to implement the computation of the neighboring lattice element Hilbert code and improved the computational efficiency of the linear octree neighboring lattice element [16]. Yang et al. designed an iterative method based on the non-uniform Hilbert space-filling curve fast generation algorithm based on the iterative method, which can generate more lightweight non-uniform Hilbert curves [17]. Meanwhile, in spatial data partitioning, the above methods fully consider the spatial information of spatio-temporal trajectory data, but fail to fully utilize their temporal information.

Data access speed is the key to judge whether the data index is satisfactory [18]. Grid indexing based on spatial partitioning has been widely used in recent years due to its simplicity of construction [19]. The literature [20,21] combined Geohash encoding with an HBase key-value storage structure based on the key-value storage structure of distributed databases [20,22] to construct fine and accurate indices with deepening grid hierarchy division. The literature [23] abstracts mobile data objects, constructs multi-level storage hierarchies and innovates storage and I/O mechanisms to improve I/O performance in managing data. The literature [24] builds a highly scalable distributed computing framework based on the multidimensional aggregated pyramid model, which supports the efficient interaction visualization under different queries in spatio-temporal and attribute multidimensions. All of the above indexing methods have efficient query performance in spatio-temporal dimensions [25], but have complex hierarchical partitioning, high I/O computational overhead and the problem of complex index construction.

In order to address the above problems, this paper focuses on exploring good spatial data partitioning strategies [26] by combining various big data processing frameworks [27–29] to construct a proven spatio-temporal hybrid index. Under the distributed computing framework Spark, the algorithmic process of Hilbert curve division of regions is studied to ensure the storage of spatio-temporal data in spatio-temporal nearest neighbors and to balance the data volume of each region; the spatial division scheme is extended to the temporal dimension, the slices are divided by days and the time is finely encoded into minutes to form a global spatio-temporal subdivision scheme. On this basis, this paper proposes a data storage model based on spatio-temporal multi-angle hierarchical organization. The model associates spatio-temporal data with the help of spatio-temporal cubes [30], which makes it easy for data mining [31] and visualization display [32]; meanwhile, it designs row keys to create spatio-temporal hybrid encoding, stores multidimensional spatio-temporal data down to a distributed HBase database and combines it with a pre-partitioning mechanism [33] to allocate storage resources to solve the problems of write heat and data skewing, which achieves efficient trajectory big data storage and retrieval. The model shows good query performance in both exact point query and spatio-temporal range query and can provide an effective support model for trajectory big data mining and analysis.

2. Study Area and Dataset

2.1. Study Area

Xiamen is located in the south of Fujian Province, with a topography sloping from northwest to southeast, and is divided into six municipal districts. Among them, Siming District is the center of a multi-functional district and is adjacent to Huli District to the north, which is extremely rich in travel data. Jimei District and Haicang District have been important ports since ancient times and undergo a relatively dense travel of people. The former Tongan District was divided into two and split into Tongan District and Xiangan District, which has an extensive territory but fewer neighborhoods under its jurisdiction and the lowest density of people travel. The six districts of Xiamen city present sparse and dense crowd activities due to their functions and location characteristics, and the data of people travel in the study area are shown in Figure 1. From Figure 1, it can be seen that the person travel data in Xiamen are densely distributed in each neighborhood, and the data distribution reflects the neighborhood distribution of Xiamen city.

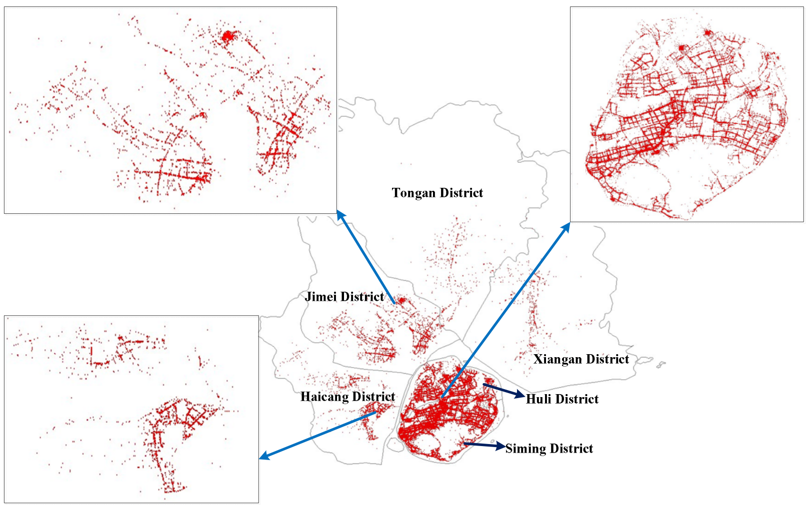


Figure 1. Example of experimental data.

2.2. Experimental Data

This experiment uses Xiamen city cruiser order data, selected from 31 May 2019, with a total of 248,161 data, each datum being a data object. The spatial dataset ranges from 117.908136° E to 118.337183° E and from 24.425163° N to 24.817838° N. The cruiser order data belongs to the OD data in the trajectory data, and each data object includes the O-point and D-point time information, spatial information and attribute information of the order, as shown in Table 1. Each cruiser order datum has problems such as data drift and incomplete latitude and longitude information during the collection process due to the equipment. Due to the fact that this study mainly focuses on the cruiser order information within Xiamen, it is necessary to screen out the data whose OD point location information is not within the study area. Therefore, the data are firstly cleaned to screen out the objects with data drift, incomplete latitude and longitude information, and the longitude and latitude information of boarding and alighting vehicles not within the study area, and then the latitude and longitude are corrected to complete the coordinate conversion. The processed data contained a total of 241,556 pieces of information, and the cleaning efficiency reached 97.34%.

Table 1. Data content.

Serial Number	Field Name	Field Meaning	Example
1	CAR_NO	License plate number	87a1f63390dcb590b94bea32e66e6d2a
2	GETON_DATE	Boarding time	31 May 2019 14:14:00
3	GETON_LONGITUDE	Boarding longitude (WGS84 GPS Standard)	118.1061
4	GETON_LATITUDE	Boarding latitude (WGS84 GPS Standard)	24.47037
5	GETOFF_DATE	Disembarkation time	31 May 2019 14:24:00
6	GETOFF_LONGITUDE	Downtime longitude (WGS84 GPS Standard)	118.115031
7	GETOFF_LATITUDE	Dismount latitude (WGS84 GPS Standard)	24.470555
8	PASS_MILE	Metered kilometers	2.2
9	NOPASS_MILE	Empty kilometers	1.2
10	WAITING_TIME	Waiting time	333

3. A Data Storage Model Incorporating Data Partitioning and Spatio-Temporal Multidimensional Hierarchical Organization

To effectively organize and manage spatio-temporal big data, this paper proposes a data storage model that incorporates data partitioning and a spatio-temporal multi-angle hierarchical organization, as shown in Figure 2. Under the Spark computing framework, the model first preprocesses the dataset to meet the requirements of application scenarios. Next, the spatial data partitioning is completed based on Hilbert curve hierarchical decomposition in space; the temporal partitioning is performed by day and finely coded in a minute system, forming a coding method from overall to local subdivision. The data are organized hierarchically from spatial and temporal perspectives to build a mixed temporal and spatial coding index. Finally, according to the principle of distributed storage and combined with the pre-partitioning mechanism, HBase data records are matched to the corresponding region area.

3.1. Principle of Hilbert Curve Space Partitioning

Both Hilbert curves and Z-curves recursively divide the filled region using a quadratic pattern to fill a two-dimensional or even a high-dimensional space with one-dimensional continuous straight lines [34]. Based on the extension direction of the Hilbert curve, similar entities have spatial proximity in the multidimensional space and uniquely encode the recursive region, as shown in Figure 3a. Distinct from the Hilbert curve, the Z-curve has spatial proximity in the local region, and the spatial jump is exceptionally evident when crossing the local region, as shown in Figure 3b.

As shown in Figure 4, according to the distribution of the dataset, some of the object numbers are organized into a grid, and it can be seen that due to the richness of urban land use types, different functional areas present different spatio-temporal clustering and dispersion of population, showing a non-homogeneous and dense distribution.

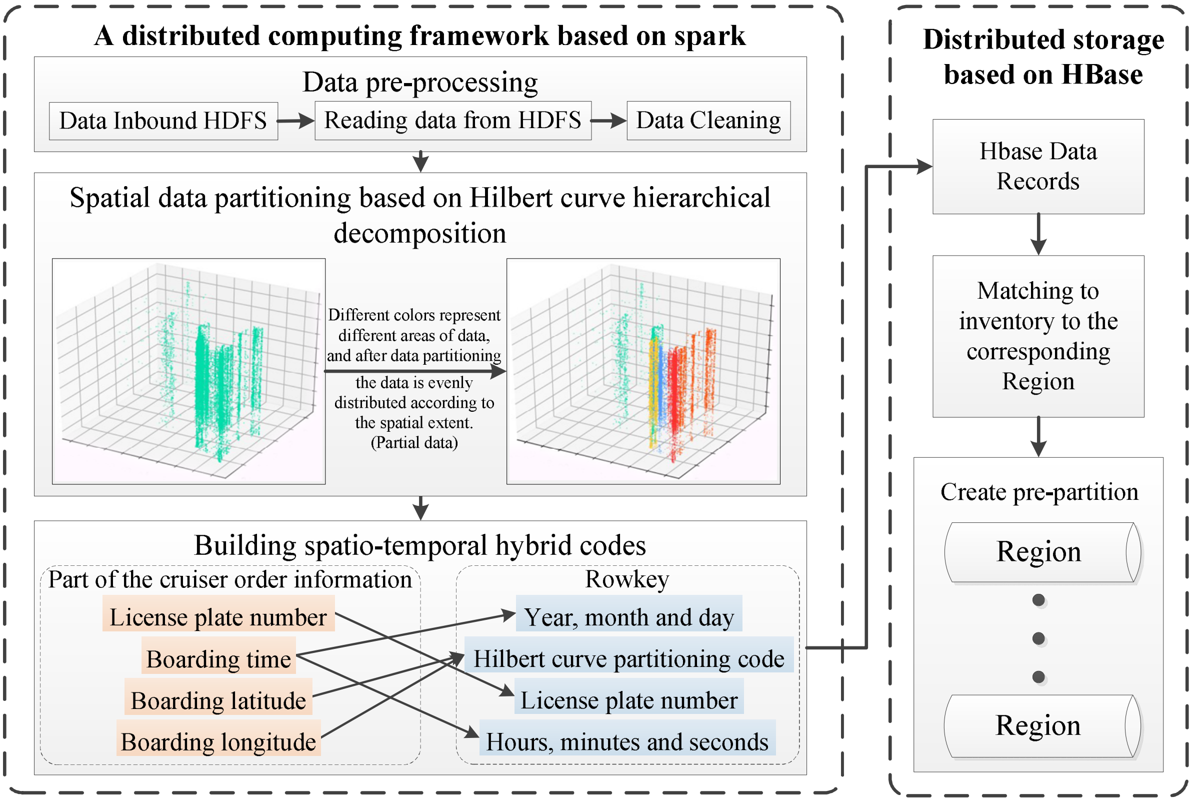


Figure 2. Data storage model incorporating data partitioning and spatio-temporal multidimensional hierarchical organization.

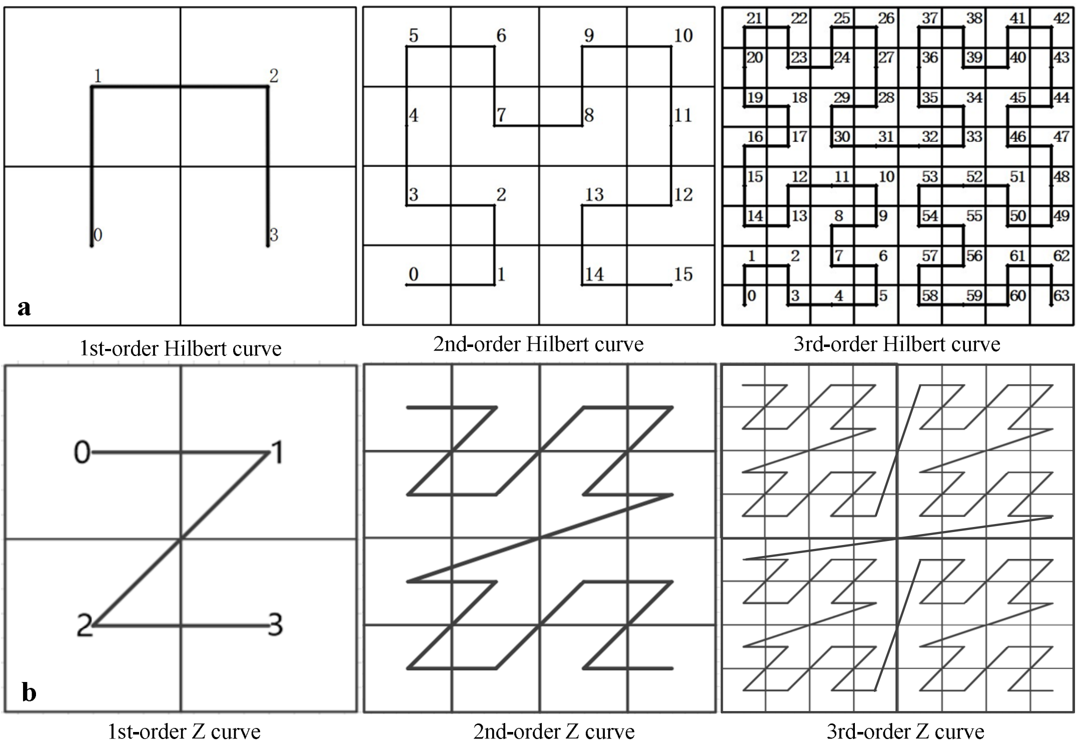


Figure 3. Space-filling curve schematic. (**a**) Coding principle of 1st, 2nd and 3rd order Hilbert curves, (**b**) Coding principle of 1st, 2nd and 3rd order Z-curves.

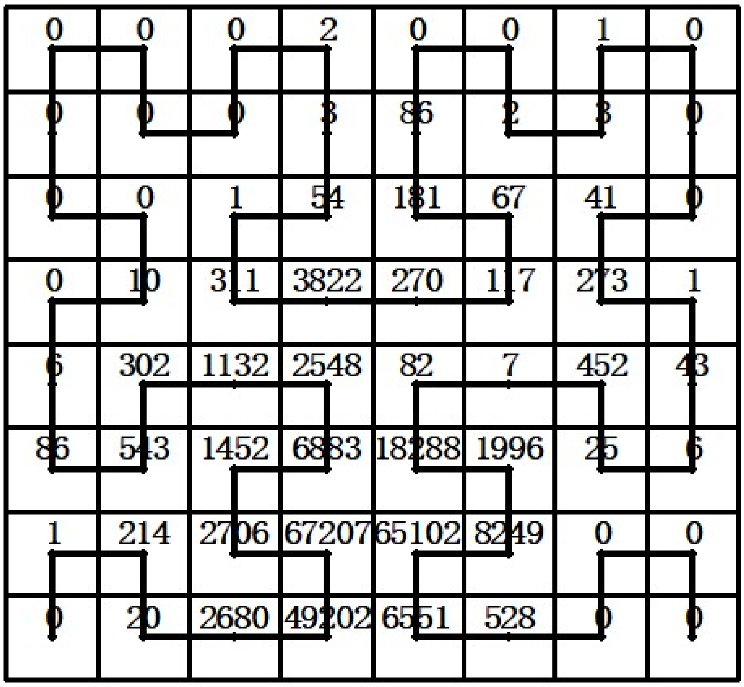


Figure 4. Distribution of the number of objects in the partial data grid (the data in the grid represent the number of objects in each area).

Considering the scope of the spatial dataset and the number of stored nodes, an initial low-order Hilbert curve is chosen to divide the entire study area into $2^k \times 2^k$ primed grids. As shown in the left side of Figure 5, the number of objects within each grid is counted, and the Hilbert curve coding rule is used to code all the objects in the space within each grid, establishing a one-to-many relationship between the initial Hilbert grid coding and the spatial objects. According to the initial order to determine the number of objects in the grid standard value, and according to the order of the initial fraction, Hilbert encoding will be the number of objects in space with the standard value in turn to do the judgment. As shown in Figure 6a,b, the areas exceeding the subgrid subdivision Hilbert grid is established until it is around the standard value or does not exceed the maximum number of decomposition levels to generate a new code and the number of objects in the grid. Considering that the number of objects in some of the grids is much smaller than the standard value, the grids can be further combined to generate new spatial data division regions, as shown in Figure 6c. Since the coding order follows the direction of the Hilbert curve extension, the number of objects in each partition can be relatively balanced while ensuring the spatial proximity of data objects in the same area. The process of the spatial data partitioning algorithm based on Hilbert curve hierarchical decomposition is shown in Figure 5, and the final number of divided regions and storage nodes is much smaller than the number of spatial objects.

In order to fully verify the spatial proximity when dividing regions in space using Hilbert curves, the results of dividing regions in space by Z-curves were compared. As shown in Figure 7a,b, the Z-curve is no different from the Hilbert curve in spatial hierarchy division, but due to the difference of encoding rules between the two space-filling curves, it can be seen by comparing Figures 6c and 7c that after the deep division and spatial region merging, and the formation of the irregular spatial region, the spatial region result of the Hilbert curve proximity is a complete region, while the Z-curve spatial region is fragmented and spatial spanning occurs between adjacent codes.

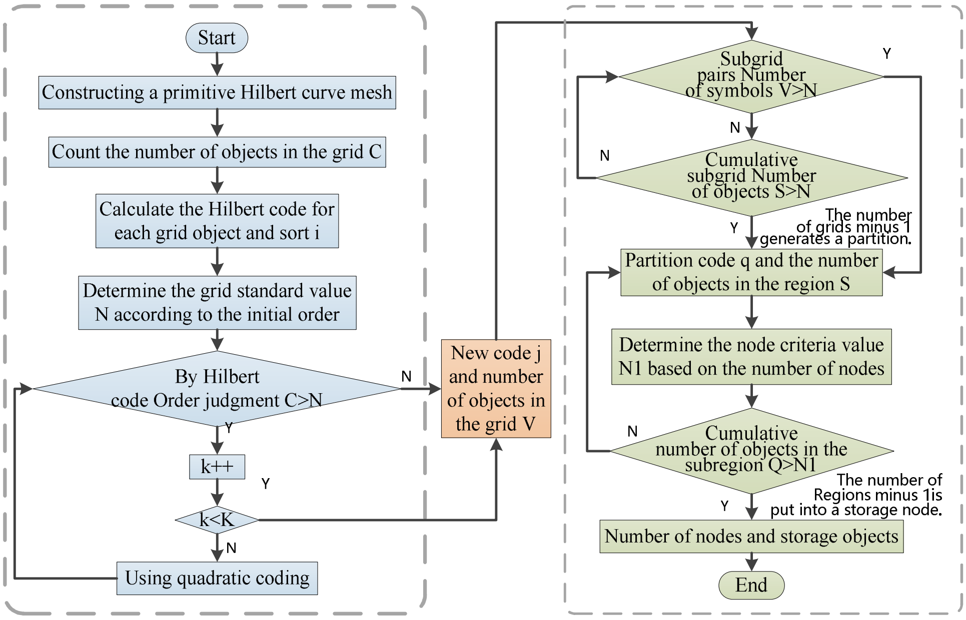


Figure 5. Spatial data partitioning algorithm flow based on Hilbert curve hierarchical decomposition.

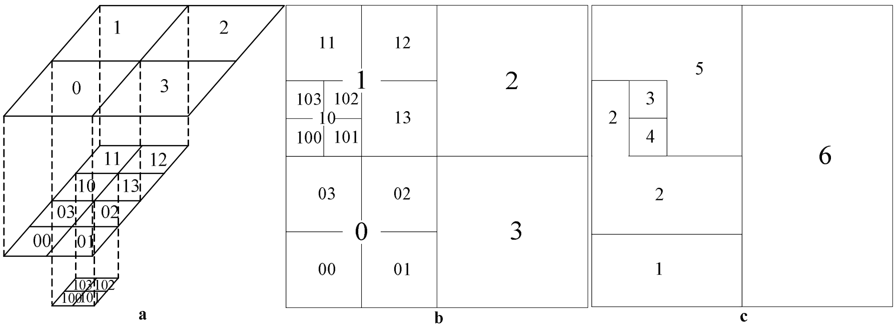


Figure 6. Hilbert curve hierarchy principle. (**a**) Three dimensional representation of spatial region divided by Hilbert curve, (**b**) Two dimensional representation of spatial region divided by Hilbert curve, (**c**) Two dimensional representation of merged spatial regions.

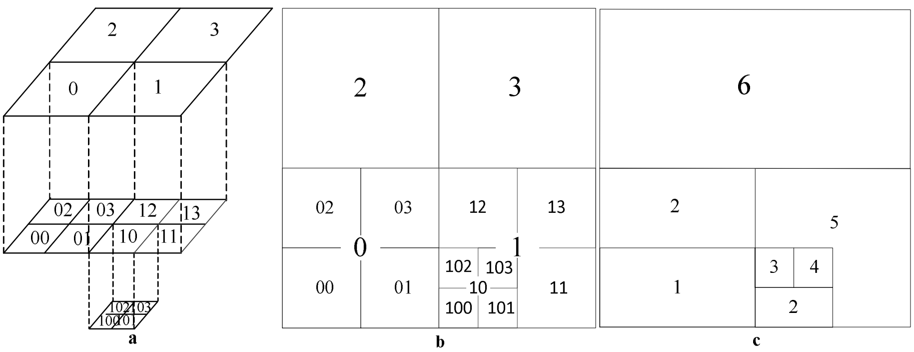


Figure 7. Z-curve hierarchy principle. (**a**) Three dimensional representation of spatial region divided by Z-curve, (**b**) Two dimensional representation of spatial region divided by Z-curve, (**c**) Two dimensional representation of merged spatial regions.

3.2. Multi-Level Spatio-Temporal Coding Model Based on Hilbert Curve Partitioning

To achieve efficient data retrieval and good visual analysis of spatio-temporal data trajectories, the mainstream data structures mostly use spatio-temporal cubes [30]. The spatio-temporal cube as a normative space to manage spatio-temporal objects has the advantage of storing data objects as well as connections between objects, which can organize, store and analyze geospatial big data in data integration [35]. To cope with accurate object localization and event detection, a continuous deepening of the grid hierarchy division and an establishment of multi-level indices are mostly used to achieve fine indexing, such as the spatio-temporal cube with internal structure representation [36]. However, when dealing with multiple object queries, it is necessary to retrieve them from a larger range to facilitate improved retrieval efficiency. Therefore, a multilevel spatio-temporal coding model based on Hilbert curve partitioning can be borrowed from the data organization model of the spatio-temporal cube, where the X and Y axes represent geographic space and the Z axis represents time. The model first performs coarse-grained spatial region division, and then organizes data from multiple temporal scales.

At the spatial level, according to the Hilbert curve coding principle, the spatial area is divided according to the extension direction of the Hilbert curve, so the neighboring objects are stored in the same area, and the study area is divided into irregular areas with relatively balanced data volume by fully taking into account the dense degree of data objects and spatial location relationship. At the temporal level, people travel intensively, which can generate a large amount of data every day, so the data are organized in days. The time structure of the collected data types in hours, minutes and seconds is converted into a minute system, coded as 0-1439, and specified to use a four-digit representation to unify the data structure.

The data without Hilbert curve-based partitioned multi-level temporal coding model as shown in Figure 8a are not organized and can only be written into the database in the order in which they are collected, and the pre-partitioning mechanism of the distributed database HBase cannot be used. As shown in Figure 8b, after spatial domain partitioning and temporal multi-scale organization, each data object has data partitioning and temporal coding. Then, according to the pre-partitioning mechanism, the storage location of the partitioned region is set so that the neighboring regions are stored in the same node. The multi-level spatio-temporal coding model based on Hilbert curve partitioning does not perform uniform subdivision in the spatial domain and uses coarse-grained partitioning, as well as divides time slices by days in the temporal domain and uses minute-based fine-grained coding. The objects stored in the same partitioned region have spatial connectivity, which facilitates the analysis of trajectory data within a region. The entire time cycle of urban data is divided into day-based organization, and a space-time cube is constructed for each time slice with minute-based fine-grained coding, which fully considers the temporal characteristics of the data and the temporal information can be fully expressed. The scalable cube storage processing method can meet the demand of highly dynamic trajectory spatio-temporal data management with high scalability, which not only improves the storage efficiency, but also enhances the data expression and spatial association effect.

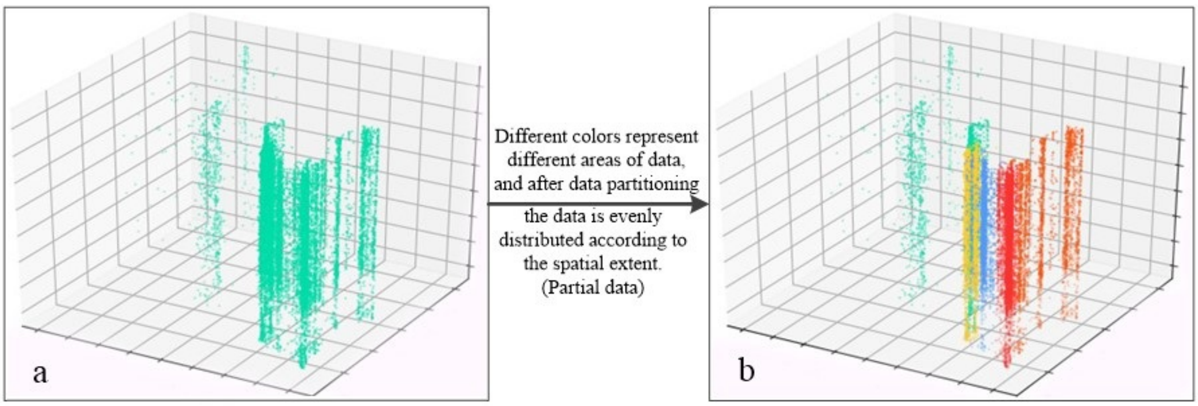


Figure 8. Multi-level spatio-temporal coding model based on Hilbert curve partitioning (partial data of the study area). (**a**) Data distribution without spatial data partition, (**b**) Data distribution for partitioning spatial data (Color represents partition).

3.3. HBase RowKey Design

The distributed database HBase is a highly reliable, column-oriented, scalable distributed database mainly used to store unstructured and semi-structured loose data [37]. If data are stored centrally without intervention, this will lead to write heat problems and needs to be combined with a pre-partitioning mechanism to allocate storage space in advance. The Hilbert curve is used to divide the spatial regions with relatively balanced data volume and assign the same storage node for the neighboring regions. Additionally, design RowKey and storage table are used according to the characteristics of HBase key-value storage. The data partition coding and temporal coding based on the Hilbert curve partition multilevel spatio-temporal coding model are effectively combined to uniquely mark each data object. The literature [38] designed three RowKey structures, time–space, space–time and mixed space–time, and proved that time–space RowKey and space–time RowKey have better query performance only in a single information dimension, while mixed space–time RowKey has better query efficiency in both space-time dimensions, effectively proving the advantages of the mixed space–time index. For this reason, the RowKey encoding structure in this paper is expressed in the way of year, month and day + division area encoding value + user ID + time and minute to ensure the uniqueness of RowKey values in each table. As shown in Figure 9, the storage table structure consists of RowKey, Column Family and Time Stamp, and the column family includes car number CAR_NO, area code M(O), passenger boarding longitude and latitude, passenger alighting longitude and latitude, boarding and alighting time and matched into the specified region pre-partition according to the rules. Therefore, according to the above temporal and spatial mixed row key design, it can meet the retrieval demand of precise query of trajectory and temporal and spatial wide range query.

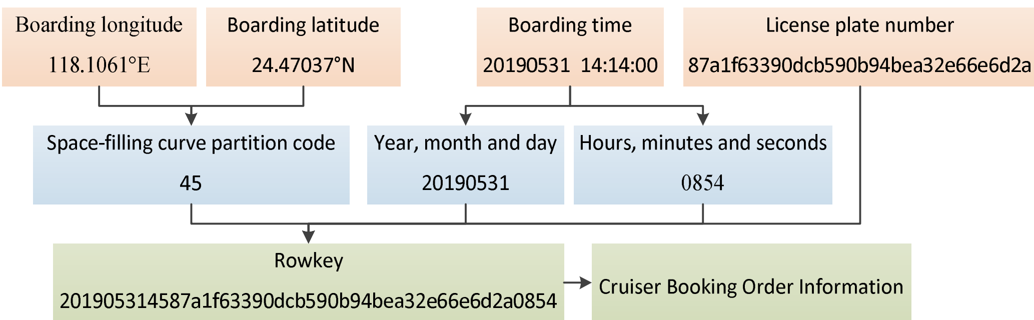


Figure 9. RowKey coding structure.

4. Validation Experiments and Analysis of Results

4.1. Experimental Environment

The distributed computing cluster consists of one Master node and two Slave nodes. Each server is configured with a CUP Intel Core i7-9700 @ 3.00 GHz octa-core, 16 GB of RAM and CentOS 6.9 under Linux. JDK, Spark, Hadoop, HBase and Zookeeper frameworks are installed on each node separately, and the versions of each framework are shown in Table 2. Firstly, the data partitioning is completed according to the Hilbert curve spatial partitioning principle in Section 3.1. To verify the improvement of the multi-level spatio-temporal coding model in both write speed and retrieval speed, this study designs two comparison experiments. The first one is the speed comparison of HBase table data pre-partitioned write and automatic partitioning write schemes, the second one is the speed comparison of different batches of data based on different retrieval conditions under the two write schemes.

Table 2. Framework version.

Frame Name	Version Number
JDK	JDK1.8
Spark	Spark-2.2.0-bin-2.6.0-cdh5.14.0
Hadoop	hadoop-2.6.0-cdh5.14.0
HBase	hbase-1.2.0-cdh5.14.0
Zookeeper	zookeeper-3.4.5-cdh5.14.0

4.2. Hilbert Curve Space Partition Determination

Different from GPS trajectory data, OD trajectory data only record the starting and ending position of each data object without considering the trajectory's morphological characteristics, and only mark the valid OD trajectory. The aggregation indices of the dataset are calculated as −807.435044, −933.402774 and −927.308349. When comparing the three data, it can be seen that the data aggregation degree is the lowest and the dispersion degree is the highest when the trajectory data are labeled with O point data. In order to achieve the effective labeling of data and make full use of data distribution to divide data hierarchy, the starting position is chosen to label each data object as the basis for data partitioning. According to the data distribution and the number of storage nodes in the study area, the initial partitioning order is selected first. The standard value of the number of objects in the grid is obtained from the ratio of the total number of objects to the number of initial order grids, which is recorded as Equation (1), and the standard value of 2774 is calculated according to Equation (1). The grids are coded in turn, the number of objects in the grid is calculated while the grids are compared with the standard value in turn to find the grids that need further subdivision. Through several experiments, the statistics of experimental results are shown in Table 3.

$$N_0 = \frac{N}{2^{k_0}} \tag{1}$$

where N_0 is the standard value of the number of grid objects; N is the total number of data entries; 2^{k0} is the number of grids.

Table 3. Summary of experimental results.

Serial Number	K Value	Edge	Number of Grids	Number of Grids Greater than Standard Value
1	3	8	8×8	8
2	4	16	16×16	18
3	5	32	32×32	22
4	6	64	64×64	6
5	7	128	128×128	1

When divided to the cumulative order of 7, there is only one grid in which the amount of data is greater than the standard value, but because the amount of data is very close to the standard value, there is no need to perform hierarchical division to avoid unnecessary computational overhead; therefore, all the grid codes and object numbers of the cumulative division level of 7 are obtained. The grid division is based on the maximum minimum longitude and maximum minimum latitude of the spatial dataset range, and there is the problem of there not being data in some grids and of the number of objects being much smaller than the standard data volume of the grid. Therefore, it is necessary to reorganize the spatial area by Hilbert curve coding. The standard value is still used as the division criterion, and the grid data are accumulated and divided into the next region when the standard value is exceeded. A total of 84 redefined regions are finally obtained, and the Hilbert curve partitioning is shown in Figure 10a. Meanwhile, the Z-curve partitioning results are compared, and it can be seen from Figure 10b that the number of partitions is lesser, but some regions completely overlap, and there are even small regions within large regions, and the partitioning results do not have reference ability.

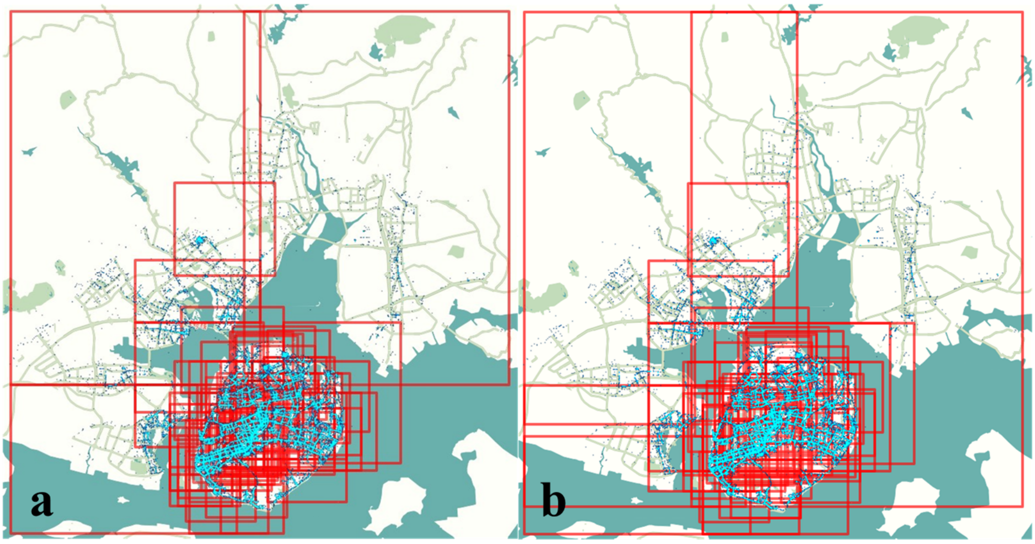


Figure 10. Data partition map. (**a**) Final spatial partition result of Hilbert curve, (**b**) Final spatial partition result of Z-curve.

The Hilbert curve is used to ensure the proximity between the divided spatial data, follow the principle of one-to-many spatial regions, avoid the dense division of the whole spatial range by using the hierarchical decomposition of a few subgrids, effectively reduce the calculation and sorting time of Hilbert curve coding of spatial objects and greatly improve the computational efficiency. By calculating the average data volume in the division region and the size of spatial objects in the subgrid, the appropriate parameters of hierarchical decomposition are determined to achieve the relative balance of spatial data volume in each division region.

4.3. Write Speed Comparison Analysis

Experiment 1 does not set up pre-partitioning, and RowKey is encoded as data order; Experiment 2 sets up pre-partitioning according to the dataset and the number of storage nodes, and RowKey is encoded using the Section 3.3 row key design structure, including two types of Hilbert curve and Z-curve. In the above experimental environment, the two schemes are used for 50,000, 100,000, 150,000 and 250,000 large batch writing experiments.

The experimental results are shown in Figure 11. After dividing the region according to the space-filling curve, the write speed of setting pre-partitioning is within the storage limit of region partitioning, and the data write speed is maintained in a relatively stable state as the data batch increases; the data write speed of not setting pre-partitioning is close to a linear increase with the increase in data batch. At the order of 50,000, the data write speed without pre-partitioning is better than the write speed with pre-partitioning, which is mainly because, at a small order of magnitude, HBase first completes resource allocation and then deposits data in a slightly higher time than the direct write speed. As the order of magnitude increases, the write speed with pre-partitioning is significantly better than that without pre-partitioning, which effectively avoids the computation time consumption caused by HBase's continuous self-partitioning. The experimental results show the advantages of the spatial data partitioning strategy in this paper in terms of storage resource allocation and improving the write heat problem.

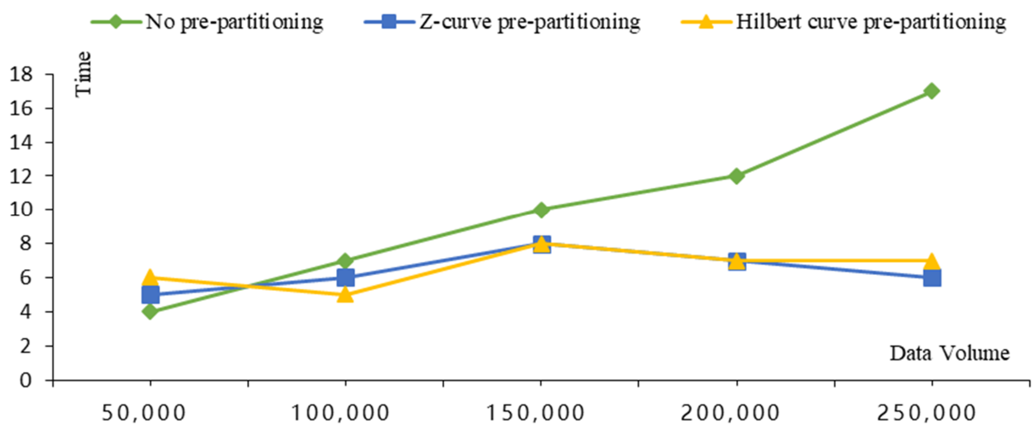
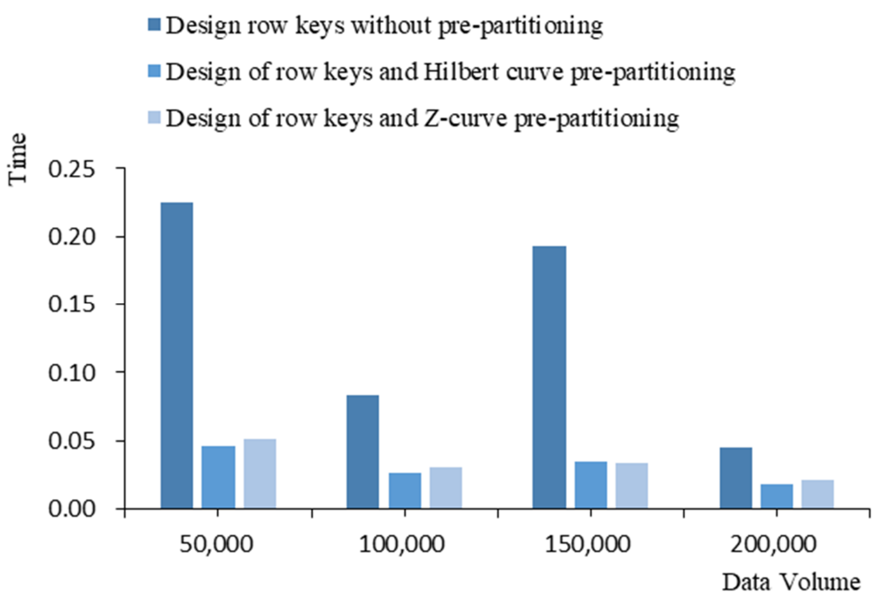


Figure 11. Write speed comparison chart.

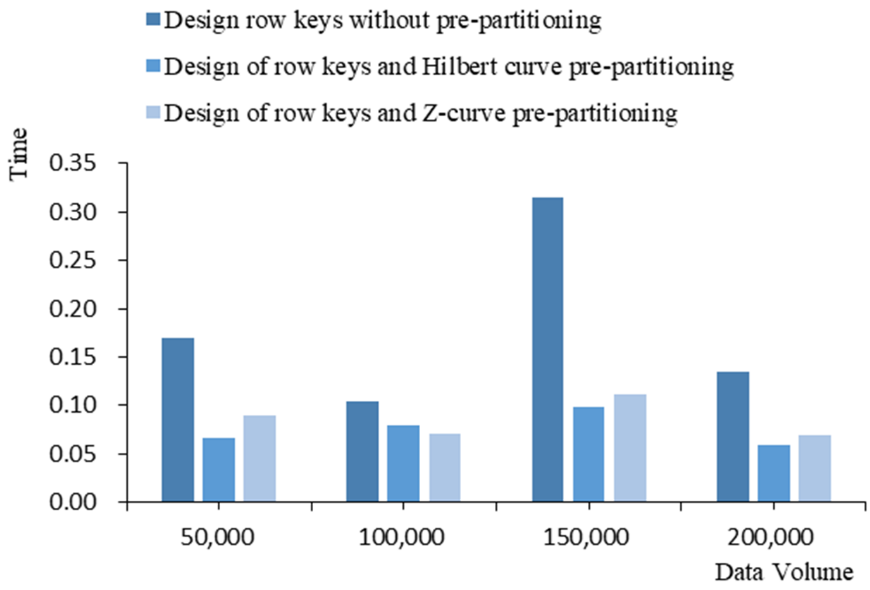
4.4. Comparative Analysis of Search Speed

After completing each batch-level data entry in Experiment 1, the retrieval efficiency is explored in three cases, which are single track object data retrieval with fixed row keys, range track object data retrieval with fixed row keys and regional track object data retrieval with fuzzy row keys. The average value is taken for multiple experiments to compare the query speed. From Figure 12, it can be seen that the retrieval speed of the designed row key combined with the pre-partitioning mechanism is significantly improved than that without the pre-partitioning mechanism under all three query conditions.

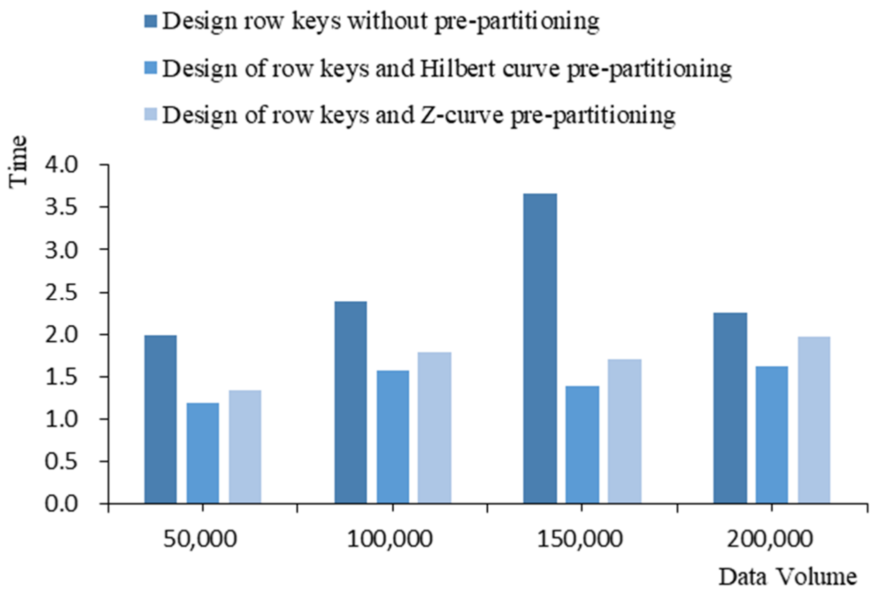


(**a**)

Figure 12. *Cont.*



(**b**)



(**c**)

Figure 12. Retrieval speed comparison chart. (**a**) Single-trajectory data retrieval, (**b**) Range trajectory object data retrieval, (**c**) Regional trajectory object data retrieval.

When retrieving data from a single-track object with fixed row key, one is randomly selected from all data objects at each batch level and queried under two writing schemes, respectively. As shown in Figure 12a, under the design row key without combining the pre-partitioning mechanism, there is a significant query speed fluctuation by querying only one data. On the contrary, under the design row key combined with the pre-partitioning mechanism, the query speed is not only effectively improved, but the query speed is also in a relatively smooth state, and there is no significant difference between the retrieval speed of the two space-filling curves. As shown in Figure 12b, when retrieving the range data with fixed row keys, this experiment randomly selects 20 adjacent data under each batch level data for verification, and the retrieval speed is significantly improved. As shown in Figure 12c, in the query of regional data with fuzzy row keys, the partition code retrieval of each batch-level data can be randomly selected for verification, and 430, 392, 366 and

432 objects corresponding to 3440, 3136, 2928 and 3456 data are selected for query speed comparison, respectively. As can be seen from Figure 12, the retrieval speed is significantly improved, and the query gap time of different levels of data volume is also reduced, which fully proves the effectiveness of the spatio-temporal hybrid retrieval method in this paper. Meanwhile, the results of the Hilbert curve dividing spatial regions and constituting row keys are slightly better than those of the Z-curve. In summary, the Hilbert curve is better than the Z-curve both in dividing spatial regions and in constructing mixed spatio-temporal indices comparing multiple retrieval modes.

5. Conclusions

In order to solve the problems of hot data writing, storage skew, high I/O overhead and slow retrieval speed in the process of distributed storage and retrieval of trajectory big data, this paper proposes a spatio-temporal multi-angle hierarchical organization data storage model incorporating data partitioning. The model is based on the Spark computing framework, exploring the algorithmic process of dividing data regions by Hilbert curves, organizing data from spatio-temporal multi-levels, constructing spatio-temporal hybrid indices and allocating storage resources in combination with a pre-partitioning mechanism. The experiments are conducted with Xiamen city cruiser OD data as an example for model computation and validation analysis, and the results show that the model can effectively improve data storage and retrieval speed with relatively stable write and query speed under different orders of magnitude and application scenarios, as well as provide an effective data support model for trajectory big data mining and analysis.

The proposed spatio-temporal multidimensional hierarchical data storage model with fused data partitioning in this paper has carried out validation studies using only a single data source. Therefore, the next step will be to fuse multiple sources of data, such as public transportation and cabs, to further improve the retrieval efficiency of the data model in multi-application scenarios and extend the usability of the model.

Author Contributions: Conceptualization, Zhixin Yao and Jianqin Zhang; methodology, Zhixin Yao; software, Zhixin Yao and Taizeng Li; validation, Zhixin Yao; formal analysis, Zhixin Yao; investigation, Zhixin Yao and Jianqin Zhang; resources, Jianqin Zhang and Ying Ding; data curation, Ying Ding; writing—original draft preparation, Zhixin Yao; writing—review and editing, Jianqin Zhang; visualization, Zhixin Yao; supervision, Ying Ding; project administration, Jianqin Zhang; funding acquisition, Jianqin Zhang. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Beijing Natural Science Foundation (grant No. 8202013), the National Science Foundation of China (grant No. 41771413), and National key R&D plan project (grant No. 2019YFC1804900).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhou, Y.; Chen, Q.; Shan, B.; Jiang, F.; Pang, Y. A Distributed Storage Strategy for Trajectory Data Based On Nosql Database. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; IEEE: Piscataway, NJ, USA, 2019.
2. Tian, R.; Zhai, H.; Zhang, W.; Wang, F.; Guan, Y. A Survey of Spatio-Temporal Big Data Indexing Methods in Distributed Environment. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 4132–4155. [CrossRef]
3. Pimpalkar, A.P.; Raj, R.J.R. Influence of pre-processing strategies on the performance of ML classifiers exploiting TF-IDF and BOW features. *ADCAIJ: Adv. Distrib. Comput. Artif. Intell. J.* **2020**, *9*, 49. [CrossRef]
4. Cao, B.Y.; Feng, H.S.; Liang, J.H.; Li, X. Using Hilbert curve and Cassandra technology to realize spatiotemporal big data storage and indexing. *J. Wuhan Univ.* **2021**, *46*, 620–629.
5. Xiang, L.G.; Wang, D.H.; Gong, J.Y. Geohash coding organization and efficient range query of large-scale trajectory data. *J. Wuhan Univ.* **2017**, *42*, 21–27.

6. Ai Jawarneh, I.M.; Bellavista, P.; Corradi, A.; Foschini, L.; Montanari, R. Efficient QoS-Aware Spatial Join Processing for Scalable NoSQL Storage Frameworks. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 2437–2449. [CrossRef]
7. Zhou, C.; Lu, H.M.; Xiang, Y.; Wu, J.; Wang, F. GeohashTile: Vector Geographic Data Display Method Based on Geohash. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 418. [CrossRef]
8. Huang, K.Y.; Li, G.Q.; Wang, J. Rapid retrieval strategy for massive remote sensing metadata based on GeoHash coding. *Remote Sens. Lett.* **2019**, *10*, 111–119. [CrossRef]
9. Zhou, Y.C.; De, S.; Wang, W.; Moessner, K.; Palaniswami, M.S. Spatial Indexing for Data Searching in Mobile Sensing Environments. *Sensors* **2017**, *17*, 1427. [CrossRef]
10. Qian, C.; Yi, C.; Cheng, C.; Wei, X.; Zhang, H. Geosot-based spatiotemporal index of massive trajectory data. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 284. [CrossRef]
11. Wu, Y.H.; Cao, X.F. Hilbert code index method for spatiotemporal data in virtual battlefield environment. *J. Wuhan Univ.* **2020**, *45*, 1403–1411.
12. Jiang, L.Y.; Li, B.B.; Li, M.J.; Chen, Y.; Ding, J. Efficient 3D Hilbert Curve Encoding and Decoding Algorithms. *Chin. J. Electron.* **2022**, *31*, 277–284.
13. Wu, Y.H.; Cao, X.F.; Yu, A.Z.; Sun, W.Z. Three-dimensional Hilbert curve hierarchical evolution model and coding calculation. *J. Surv. Mapp.* **2022**, *51*, 104–114.
14. Jia, L.Y.; Kong, M.; Wang, W.C.; Li, M.J.; You, J.G.; Ding, J.M. A two-dimensional Hilbert codec algorithm under skewed data distribution. *J. Tsinghua Univ.* **2022**, *62*, 1426–1434.
15. Kang, Y.X.; Gui, Z.P.; Ding, J.C.; Wu, J.H.; Wu, Y.H. Parallel Ripley's K-function based on Hilbert space partitioning and Geohash indexing. *J. Geomat.* **2022**, *24*, 74–86.
16. Wu, Y.H.; Cao, X.F. Neighborhood lattice element computation algorithm for Hilbert octree. *J. Wuhan Univ.* **2022**, *47*, 613–622.
17. Yang, F.; Hua, X.; Yang, Z.K.; Li, X.; Zhao, X.K.; Zhang, X.N. A fast algorithm for filling curve generation in non-uniform Hilbert space based on iterative method. *J. Wuhan Univ.* **2022**, 1–15. [CrossRef]
18. Xia, J.Z.; Yang, C.W.; Li, Q.Q. Building a spatiotemporal index for Earth Observation Big Data. *Int. J. Appl. Earth Obs. Geoinf.* **2018**, *73*, 245–252. [CrossRef]
19. Zhang, K.; Shang, S.; Yuan, N.J.; Yang, Y. Towards efficient search for activity trajectories. In Proceedings of the 2013 IEEE 29th International Conference on Data Engineering (ICDE), Brisbane, QLD, Australia, 8–12 April 2013; IEEE: Piscataway, NJ, USA, 2013.
20. Le, H.V.; Takasu, A. G-HBase: A High Performance Geographical Database Based on HBase. *IEICE Trans. Inf. Syst.* **2018**, *E101D*, 1053–1065. [CrossRef]
21. Zhang, J.W.; Yang, C.; Yang, Q.; Lin, Y.; Zhang, Y. HGeoHashBase: An optimized storage model of spatial objects for location-based services. *Front. Comput. Sci.* **2020**, *14*, 208–218. [CrossRef]
22. Kumar, S.; Madria, S.; Linderman, M. M-Grid: A distributed framework for multidimensional indexing and querying of location based data. *Distrib. Parallel Databases* **2017**, *35*, 55–81. [CrossRef]
23. Wadhw, B.; Byna, S.; Butt, A.R. Toward transparent data management in multi-layer storage hierarchy of hpc systems. In Proceedings of the 2018 IEEE International Conference on Cloud Engineering (IC2E), Orlando, FL, USA, 17–20 April 2018; IEEE: Piscataway, NJ, USA, 2018.
24. Guan, X.; Xie, C.; Han, L.; Zeng, Y.; Shen, D.; Xing, W. Map-vis: A distributed spatio-temporal big data visualization framework based on a multi-dimensional aggregation pyramid model. *Appl. Sci.* **2020**, *10*, 598. [CrossRef]
25. Guan, X.; Bo, C.; Li, Z.; Yu, Y. ST-hash: An efficient spatiotemporal index for massive trajectory data in a NoSQL database. In Proceedings of the 2017 25th International Conference on Geoinformatics, Buffalo, NY, USA, 2–4 August 2017; IEEE: Piscataway, NJ, USA, 2017.
26. Zhou, Y.; Zhu, Q.; Zhang, Y.T. Spatial data partition method based on hierarchical decomposition of Hilbert curve. *Geogr. Geogr. Inf. Sci.* **2007**, *4*, 13–17.
27. Le, P.; Wu, Z.Y.; Shangguan, B.Y. Design and implementation of distributed spatial data storage structure based on spark. *J. Wuhan Univ.* **2018**, *43*, 2295–2302.
28. Huang, Z.; Chen, Y.R.; Wan, L.; Peng, X. GeoSpark SQL: An Effective Framework Enabling Spatial Queries on Spark. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 285. [CrossRef]
29. Lei, B. A Hadoop-Based Spatial Computation Framework for Large-Scale AIS Data. In Proceedings of the 2019 IEEE 2nd International Conference on Elsectronics Technology (ICET), Chengdu, China, 10–13 May 2019; IEEE: Piscataway, NJ, USA, 2019.
30. Chen, W.; Huang, Z.S.; Wu, F.R.; Zhu, M.; Guan, H.; Maciejewski, R. VAUD: A Visual Analysis Approach for Exploring Spatio-Temporal Urban Data. *IEEE Trans. Vis. Comput. Graph.* **2018**, *24*, 2636–2648. [CrossRef]
31. Zhang, Y.; Lin, Y.P. An interactive method for identifying the stay points of the trajectory of moving objects. *J. Vis. Commun. Image Represent.* **2019**, *59*, 387–392. [CrossRef]
32. Kim, S.; Jeong, S.; Woo, I.; Jang, Y.; Maciejewski, R.; Ebert, D.S. Data Flow Analysis and Visualization for Spatiotemporal Statistical Data without Trajectory Information. *IEEE Trans. Vis. Comput. Graph.* **2018**, *24*, 1287–1300. [CrossRef]
33. Li, Z.; Zhao, Z.M. Geohash: Trajectory data index method based on historical data pre-partitioning. In Proceedings of the 2021 7th International Conference on Big Data Computing and Communications (BigCom), Deqing, China, 13–15 August 2021; IEEE: Piscataway, NJ, USA, 2021.

34. Wu, M.G. Hilbert filling curve and space division method of point data set for spatial distribution pattern detection Chinese. *J. Image Graph.* **2013**, *18*, 1336–1342.
35. Lu, Y.L.; Li, J.W.; Ye, S.X.; Jiang, J.W.; Yin, M.; Zhou, Y.L. GIS spatiotemporal big data organization method based on extended stream data cube. *J. Bull. Surv. Mapp.* **2018**, *8*, 115–118.
36. Bach, B.; Dragicevic, P.; Archambault, D.; Hurter, C.; Carpendale, S. A Descriptive Framework for Temporal Data Visualizations Based on Generalized Space-Time Cubes. *Comput. Graph. Forum* **2017**, *36*, 36–61. [CrossRef]
37. Chen, D.N.; Guan, X.F.; Han, L.X.; Xiang, L.G.; Wu, H.Y. VA HBase: An adaptive distributed management scheme for vector data. *J. Wuhan Univ.* **2021**, *46*, 1–11.
38. Li, X. Discussion on traffic flow data storage and index model based on spark/HBase. *J. Geogr. Geogr. Inf. Science* **2019**, *35*, 1–8.