Postprint: Organization Methods for Space Astronomical Satellite Data Based on Heterogeneous Databases

Authors: Yang Xiaoyan, Sun Xiaojuan, Shi Tao, Liu Zhiqi, Jizhou Tong

Date: 2022-01-14T14:52:04+00:00

Abstract

As the volume of data acquired by space astronomical satellites continues to grow, data applications are playing an increasingly significant role. In existing ground systems for astronomical satellites, data storage methods and organization approaches vary considerably, with data volumes reaching petabyte scale and exhibiting continuous growth. This makes it impossible to rapidly search for and extract characteristic parameters, thereby failing to meet the timeliness requirements of data applications for queries. This paper proposes a novel data organization methodology for space astronomical satellites. By parsing and extracting massive characteristic parameters from the data, it establishes correlations among observation time, spatial location, and characteristic parameters, thereby enabling multi-source data organization within a unified spatiotemporal framework; concurrently, it adopts a heterogeneous storage architecture that integrates relational and non-relational databases to design a massive characteristic parameter storage management system. When applied to a space science satellite big data application platform system, experimental results utilizing data from the Hard X-ray Modulation Telescope satellite demonstrate that the system satisfactorily fulfills requirements for data retrieval based on temporal and spatial constraints. Compared with relational database-based data organization approaches, data retrieval efficiency is significantly improved under identical query modes; furthermore, as data storage volume increases, the system demonstrates stable scalability.

Full Text

Data Organization Method for Space Astronomical Satellites Based on Heterogeneous Databases

YANG Xiaoyan[1],[2], **SUN Xiaojuan**[1],[2],[3], **SHI Tao**[1],[2], **LIU Zhiqi**[1], **TONG Jizhou**[4]

[1]Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China
[2]Key Laboratory of Technology in Geo-spatial Information Processing and Application System, Chinese Academy of Sciences, Beijing 100190, China
[3]University of Chinese Academy of Sciences, Beijing 100149, China
[4]National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China

Abstract: As space astronomical satellites generate increasingly large volumes of data, data applications have become progressively more important. Existing ground systems for astronomical satellites employ diverse data storage methods and organization schemes, with data volumes reaching petabyte scales and continuing to grow. This makes rapid feature parameter lookup and extraction difficult, preventing these systems from meeting the timeliness requirements of data applications. This paper proposes a novel data organization method for space astronomical satellites that extracts massive feature parameters from data files and establishes associations between observation time, spatial location, and these parameters, thereby enabling multi-source data organization within a unified spatiotemporal framework. The method employs a heterogeneous storage approach combining relational and non-relational databases, and designs a massive feature parameter storage management system. When applied to the Space Science Satellite Big Data Application Platform, experimental results using Hard X-ray Modulation Telescope (HXMT) satellite data demonstrate that the system effectively satisfies requirements for data retrieval based on temporal and spatial conditions. Compared with relational database organization methods, data retrieval efficiency is significantly improved under identical query patterns, and the system exhibits stable scalability as data storage volumes increase.

Keywords: space astronomical satellite data; spatiotemporal indexing; data organization; non-relational database
Classification: P171.3

Since 2015, China has launched a series of space astronomical satellites, including the Dark Matter Particle Explorer (DAMPE), the Hard X-ray Modulation Telescope (HXMT), and the Gravitational Wave High-energy Electromagnetic Counterpart All-sky Monitor (GECAM). These missions have continuously acquired vast amounts of space astronomical observation data. Ground systems store and manage both raw satellite detection data and derived products at various processing levels, including edited and calibrated products. These data products represent detection results obtained by satellites under specific spatiotemporal conditions, containing information that characterizes space astronomical targets—such as particle types, counts, energy bands, incident trajectories, and energy deposition—as well as engineering data reflecting satellite platform and payload status, including attitude, orbital position, temperature, and pressure. Such data plays a crucial role in space astronomy research, satellite and payload health trend analysis, detection target analysis, mission planning support, and visualization of detection processes.

In current ground management systems for space science strategic pilot projects, space astronomical satellite data is stored in domain-specific formats such as FITS (Flexible Image Transport System) and ROOT (a data format developed by CERN) according to the space science data model proposed by the National Space Science Center. Data retrieval requires first locating data files, parsing their formats, and then extracting required feature parameters from specific positions within the files. Some data also necessitate physical quantity conversion and time correction. Because different satellite data products use different storage formats, the parameter extraction processes vary, resulting in complex and time-consuming operations. As data volumes continue to grow, database retrieval times increase, making real-time data access increasingly difficult to guarantee. File-granularity storage and organization systems currently in use cannot satisfy the demands for real-time data retrieval.

To meet application requirements for real-time data access, feature parameters must be extracted from space astronomical data files to construct an efficient, parameter-level fine-grained data organization method. However, the sheer volume of feature parameters extracted from massive space astronomical data files presents a critical challenge for efficient organization and indexing.

1. Characteristics of Space Astronomical Satellite Data

Space astronomical satellites primarily observe various celestial targets in space. Their data includes scientific data characterizing these observation targets and engineering data reflecting satellite and payload status, exhibiting the following characteristics:

(1) Diverse data types, high temporal resolution, and massive volume. In terms of product content, space astronomical satellite data encompasses astronomical target scientific data, satellite platform engineering data, and payload engineering data. Regarding product levels, it includes edited products, calibrated products, and others. Each satellite differs in product content and levels. Taking DAMPE as an example, there are nine product levels with approximately ten categories per level, totaling over 100 categories. A half-hour data file from DAMPE's calibrated products contains about 120,000 particles, with each particle having parameters including deposited energy in each detector, hit positions, and incident trajectories. Based on a five-year mission lifetime, this generates approximately 10.51 billion records. Engineering data includes dozens of categories such as AOCC (Attitude and Orbit Control Computer) attitude data and

GPS positioning data, with most recorded at one-second intervals, some at two or four records per second. At one record per second, each satellite generates over 30 million records per year per data category. Estimating for a five-year mission with 35 data categories per satellite yields over 5 billion records per satellite. Total data volumes reach tens or even hundreds of billions of records, necessitating an efficient organization method for massive multi-source data.

(2) Spatiotemporal attributes requiring rapid multi-source data retrieval based on temporal and spatial conditions. Space astronomical satellite data can be expressed as (Time, RA, DEC, par1, par2, ···), where Time represents observation time, RA denotes the right ascension of the satellite' s field-of-view center at the observation time, DEC indicates the declination of the field-of-view center, and par1, par2 represent feature parameter values such as high-energy electron counts or payload engineering measurements. The spatiotemporal nature of this data requires unified processing of temporal and spatial attributes from multiple sources to construct feature parameter indexing and retrieval methods that support rapid multi-source data retrieval based on spatiotemporal conditions.

(3) Continuous data growth requiring scalable architecture. With existing satellites continuing operations and new satellites being launched, space astronomical satellite data volumes show a sustained growth trend. This requires building a distributed database storage system with good scalability in storage capacity, where retrieval efficiency remains essentially stable as storage capacity increases.

2. Related Work

Traditional relational databases struggle to meet the demands of massive space astronomical satellite data organization and rapid retrieval. Non-relational databases like HBase offer flexible data structures and strong horizontal scalability, making them more effective for big data organization than conventional structured databases. However, HBase only builds B+ tree indexes on primary keys, enabling fast queries based on primary keys but requiring full table scans for non-primary key queries, resulting in low efficiency. Space astronomical satellite data requires retrieval based on multiple attributes including time, right ascension, declination, and parameters, which HBase cannot efficiently support.

Researchers from various industries have investigated storing and retrieving massive spatiotemporal data using non-relational databases, following two main approaches. The first approach, adopted by scholars in geographic information, land resources, and space science, involves constructing spatiotemporal grid models and storing spatiotemporal data in non-relational databases using spatiotemporal encoding. For example, Zhang et al. proposed a distributed storage model for spatial vector data using a quadtree to establish spatial grids and constructing row keys from grid IDs and random codes for HBase storage. Kang et al. proposed the HTM-ST discrete spatiotemporal data organization model, which creates spatiotemporal coupling codes through temporal and spatial discretization and uses these codes as row keys for storing solar-terrestrial space data in HBase. Since HBase stores row keys in dictionary order, multi-attribute row key construction only supports point queries. For range queries, it requires judging topological relationships between spatiotemporal grids and query ranges at each level, continuously approximating the spatiotemporal range in the query conditions during subdivision, or performing full table scans, resulting in significantly high query latency.

The second approach, taken by computer science researchers, improves non-relational database retrieval efficiency by constructing multi-layer indexes. For instance, Ge et al. proposed a hierarchical indexing technique combining index tables and value tables with hot data caching. While this improves retrieval efficiency to some extent, it requires merging multi-column query results for multi-attribute range retrieval, failing to meet the real-time spatiotemporal range data acquisition needs of space science. Yuan et al. proposed a three-layer indexing technique called TA-index aimed at improving data ingestion efficiency, but its spatiotemporal range query requires multiple queries across index layers and database tables, resulting in long latency.

To address the need for organizing and querying massive space astronomical satellite data based on both temporal and spatial attributes, this paper proposes a novel data organization method. The approach first parses data files to extract massive feature parameters, establishing associations between observation time, spatial location, and these parameters to achieve multi-source data organization under a unified spatiotemporal framework. It then combines the flexible data structures and horizontal scalability of non-relational databases with the multi-column range query advantages of relational databases. A distributed database partitioning approach constructs an HBase cluster for storing massive feature parameters, while relational database table partitioning stores spatiotemporal index data to support retrieval from both temporal and spatial dimensions.

3.1 Feature Parameter Extraction

Existing space astronomical satellite data is stored as files in ground management systems. Feature parameter extraction represents the first step toward efficient organization of space astronomical big data. Based on FITSIO and ROOT parsing frameworks, a data parsing algorithm is constructed to accommodate parameter extraction requirements for various data formats from existing satellites. The main steps are: (1) Pre-configure parameters to be extracted for each satellite data type to generate extraction requirements; (2) Acquire satellite data product files and identify satellite name, data type, and storage format; (3) Match satellite name and data type against the extraction requirements from step 1; (4) For FITS files, call FITSIO to extract parameter values and observation times; for ROOT files, call the ROOT parsing framework; for CSV, DAT, and other common formats, directly extract parameters and observation times; (5) Perform physical quantity conversion on extracted parameters as needed, such as converting payload temperature and pressure values from onboard electrical signals to physically meaningful values; (6) Calculate satellite field-of-view position information (RA, DEC) moment by moment based on satellite attitude data.

3.2 Feature Parameter Storage

To address storage requirements for massive feature parameter time-series data, this paper proposes a feature parameter storage structure based on an HBase cluster (Figure 1), using parameter table partitioning plus time-based partitioning to support data retrieval by time point or time range.

First, massive feature parameters are divided into different parameter groups (Group1, Group2, etc.) at the granularity of individual parameters or several correlated parameters, with separate parameter tables created for each group. Interrelated parameters such as attitude quaternions, orbital position XYZ coordinates, and orbital six-element sets are stored together in group tables—for example, parameters A, B, and C form one group, while parameters U and V form another. Remaining parameters are stored in separate tables. This approach enhances storage flexibility and management convenience while supporting concurrent queries across multiple parameter tables, thereby improving multi-parameter query efficiency.

Second, based on the temporal frequency of various parameters, independent time partition indexes are created for each parameter table according to time ranges. For instance, in Figure 1, Table1 with higher-frequency parameters uses five time units as partition spans (t1, t6, t11, ···), while TableN with lower-frequency parameters uses ten time units (t1, t11, ···). This partitioning design distributes massive parameter sets into different regions by time range, enabling partition index lookups for corresponding time periods during retrieval and supporting concurrent queries across multiple partitions to further improve query efficiency.

3.3 Spatiotemporal Index Storage

The spatiotemporal index represents the relationship between observation time and field-of-view center position, requiring storage of Time, RA, and DEC fields. It must support joint retrieval based on temporal and spatial ranges—that is, retrieving data using range conditions on Time, RA, and DEC fields. While HBase excels at fast retrieval via row keys or row key ranges, it suffers from low efficiency for non-primary key queries requiring full table scans. Relational databases using SQL queries are well-suited for multi-column value queries, satisfying both point queries (retrieving data by specified time and position) and range queries (retrieving data by time and spatial ranges). Therefore, this paper stores spatiotemporal index data in MySQL relational database tables.

During satellite observation, one spatiotemporal index record is generated per second, with observation time as a sequentially increasing value. In spatiotemporal index tables, the Time field serves as the primary key. With one record per second, each satellite generates over 30 million records annually. MySQL table retrieval efficiency degrades significantly when data reaches tens of millions of records. Horizontal table partitioning can resolve bottlenecks caused by extremely large data volumes and high loads, improving retrieval efficiency.

Since typical application scenarios in this work involve hourly-level data requests that most likely query single tables, controlling single-table data volume at the million-record level ensures efficient retrieval. Therefore, during spatiotemporal index storage, tables are horizontally partitioned by month based on the observation Time field, resulting in sub-tables containing over 2 million records each. Joint queries across two tables have been tested and show no significant difference in latency compared to single-table queries. However, if application scenarios change—for example, requiring longer retrieval durations with frequent joint queries across multiple tables—the MySQL partitioning scheme may need adjustment.

The SQL statement for joint queries is as follows:

```
select Time from Table1 where Time>=?5 and Time<=?6 and RA>=?1 and RA<=?2 and DEC>=?3 and DE
union
select Time from Table2 where Time>=?5 and Time<=?6 and RA>=?1 and RA<=?2 and DEC>=?3 and DE
```

4. Application-Oriented Data Retrieval

The proposed method supports feature parameter retrieval based on temporal and spatial conditions. According to combinations of temporal and spatial retrieval criteria, data retrieval requirements can be categorized into eight cases: time point, time range, spatial point, spatial range, time point + spatial point, time point + spatial range, time range + spatial point, and time range + spatial range.

For retrieval requests containing only temporal information, multiple parallel retrieval tasks are initiated against target parameter tables. For time-point queries, the HBase `get` method (single key-value lookup) is used; for time-range queries, the HBase `scan` method (range lookup based on key start and end) is employed. Results from multiple tasks are merged upon completion.

For requests containing spatial information, spatiotemporal index tables are first queried to obtain matching time information, which is then used to retrieve parameter tables.

Using the example of retrieving parameters A, B, C, and W for RA range (r1, r3), DEC range (d1, d2), and observation time range (t1, t100), the retrieval process is illustrated in Figure 2: (1) Query spatiotemporal index tables using conditions “r1<RA<r3 and d1<DEC<d2 and t1<Time<t100” to obtain result set ; (2) Query parameters A, B, C, and W using condition “t1<Time<t100” . Since parameter W is stored in a different table from A, B, and C, separate retrieval

tasks are generated for the ABC parameter table and the W parameter table, executed concurrently; (3) For the ABC parameter table, first query partition indexes, then simultaneously retrieve matching partition tables to obtain result sets and ; (4) For the W parameter table, similarly query partition indexes and simultaneously retrieve matching partitions to obtain result sets and ; (5) Merge result sets , , , and to produce the final result set .

5.1 Experimental Design

Following the proposed method, a test system comprising an HBase cluster plus MySQL (hereinafter "HeteroDB") was built on three virtual servers with 4-core CPUs and 32GB memory each. The test data consisted of HXMT satellite data from 00:00 on September 1, 2021, to 00:00 on October 31, 2021, comprising approximately 5 million records. A comparison system using only MySQL to store HXMT detection data from the same period (hereinafter "MySQL") was also established.

As query retrieval services constitute the core of data organization and management, this paper compares the two systems based on data retrieval efficiency. Three typical scenarios were selected: retrieving certain parameters by time range, retrieving multiple parameters by time range, and retrieving certain parameters by combined time and spatial ranges. The retrieval conditions cover temporal and spatial dimensions, with parameters ranging from single to multiple types. By setting identical retrieval conditions, the efficiency of both systems was compared. To avoid impacts from unstable virtual server resources and other random errors, all retrievals were performed in two separate time periods, with recorded values representing the average of ten retrieval operations.

Experiment 1 compares response speeds of the two systems under temporal retrieval scenarios. Experiment 2 tests the HeteroDB system' s response speed for time + space retrieval scenarios. Experiment 3 further validates the scalability of the proposed method under increasing data volumes by expanding HBase database records from 5 million to 80 million and testing the relationship between retrieval latency and data scale using identical retrieval conditions.

5.2 Experimental Results and Analysis

(1) Experiment 1: Temporal Retrieval Results and Analysis

Experiment 1 was configured with 5 million table records, retrieving specified parameters using time range conditions with spans of 1, 2, 3, and 4 hours, and parameter counts of 1 and 3. Identical retrieval conditions were applied to both HeteroDB and MySQL systems.

Test results (Table 1) show that for small time spans and single-parameter retrieval (Scenario 1-1), MySQL latency is comparable to HeteroDB. However, as time spans and parameter counts increase, MySQL latency grows significantly—in Scenarios 2-4, MySQL takes over 80 times longer than HeteroDB. This is due to MySQL' s balanced binary tree indexing mechanism, which requires multiple lookups during retrieval. As retrieval duration and parameter count increase, lookup frequency rises, causing retrieval efficiency to decline exponentially. HeteroDB leverages HBase' s dictionary-ordered row key storage mechanism using time as row keys, combined with parameter table partitioning and time-based partitioning with parallel querying, thereby improving retrieval efficiency and achieving notable improvements in temporal retrieval scenarios.

(2) Experiment 2: Spatiotemporal Joint Retrieval Results and Analysis

Experiment 2 was configured with 5 million table records, retrieving a single parameter using combined conditions of time range, RA range, and DEC range to compare response speeds for spatiotemporal joint retrieval.

Test results (Table 2) demonstrate that the proposed method effectively supports data retrieval using combined temporal and spatial ranges. In Scenario 3-1 (1-hour time span, 10° RA/DEC spans), HeteroDB retrieval latency is 26.3 ms; under the same conditions, MySQL latency is essentially equivalent. However, as temporal and spatial ranges expand, MySQL latency far exceeds HeteroDB' s—in Scenario 3-4 (4-hour time span, 10° RA/DEC spans), MySQL takes approximately 5.3 times longer than HeteroDB.

(3) Experiment 3: Scalability Results and Analysis

To verify HeteroDB' s retrieval performance across different data scales, test data was progressively expanded to 10 million, 20 million, 40 million, and 80 million records, using identical retrieval scenarios to test system efficiency. Test results (Table 3) indicate that within the tested data volume range, HeteroDB' s retrieval efficiency for both temporal and spatiotemporal retrieval scenarios remains essentially stable as data volume increases.

This stability stems from HeteroDB' s time-partitioned parameter table storage approach, which supports simultaneous retrieval across multiple partitions meeting query conditions. Consequently, as data volume grows and time partitions increase, the multi-partition parallel retrieval mechanism maintains stable efficiency. As data volume continues to increase further, when time partition counts become excessive and server resources insufficient, retrieval efficiency will inevitably decline gradually. At that point, adding more nodes to the HBase distributed database can maintain retrieval efficiency.

6. Conclusion

To address the need for rapid acquisition of specified parameters from space astronomical satellite data within temporal and spatial ranges, this paper proposes an efficient method for organizing massive data. Unlike existing file-granularity management approaches, this method establishes relationships between observation time, spatial location, and feature parameters, integrating various parameters from data files into a unified spatiotemporal framework. It utilizes an

HBase distributed database to store various parameters and a MySQL relational database to store spatiotemporal indexes, achieving fine-grained data organization and management. The improvements include: (1) Separating satellite data file parsing, parameter extraction, and physical quantity conversion from traditional data acquisition processes, simplifying data retrieval; (2) Building a distributed database cluster with independent tables for parameters and time-range partitioning to support parallel temporal retrieval and improve efficiency; (3) Employing relational database table partitioning to store spatiotemporal index data, enabling retrieval from both temporal and spatial dimensions; (4) Providing good scalability for data volume growth from increasing observation times and parameter types, adapting to continuous data growth requirements.

Simulation results demonstrate that the proposed method significantly improves data retrieval efficiency and meets application needs for real-time acquisition of space astronomical satellite data.

References

[1] Xiong Senlin, et al. Framework for Space Science Data Organization[J]. Journal of Agricultural Big Data, 2019, 1(4): 30-36.

[2] Definition of the Flexible Image Transport System (FITS), version 2.1b, December 9, 2005. http://fits.gsfc.nasa.gov/standard21b.html

[3] Yang Xiaoyan, et al. Application Research of FITS Variable-Length Arrays in DAMPE Data Storage[J]. Astronomical Research & Technology, 2018(02): 176-180.

[4] Rademakers F, Brun R. ROOT: An Object-Oriented Data Analysis Framework[J]. Nuclear Instruments & Methods in Physics Research, 1998, 389(1/2): 81-86.

[5] Cui Chenzhou, et al. Search and Location of FITS Data Files[J]. Astronomical Research & Technology, 2008(02): 116-123.

[6] Ma Fuli, et al. Design and Implementation of GECAM Preprocessing Pipeline[J]. Astronomical Research & Technology, https://doi.org/10.14005/j.cnki.issn1672-7673.20210611.001

[7] Zhang Jia, et al. Storage and Computing Optimization of Large Scale Distributed Spatial Vector Data[J]. Computer Systems & Applications, 2020, 29(12): 251-256.

[8] Kang Donghe, Zou Ziming, et al. HTM-ST: A Data Model Supporting Spatio-temporal Coupled Computation for Solar-terrestrial System[J]. Journal of Geo-information Science, 2017, 19(6): 735-743. DOI:10.3724/SP.J.1047.2017.00735

[9] Ge Wei, et al. HiBase: A Hierarchical Indexing Mechanism and System for Efficient HBase Query[J]. Chinese Journal of Computers, 2016, 39(1): 140-153.

[10] Yuan Maolin, et al. Efficient Spatio-temporal Classification Index Under HBase[J]. Journal of Chinese Computer Systems, 2017, 38(6): 1231-1236.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.