Design and Implementation of MUSER' s Negative Database Interface (Postprint)

Authors: Shi Congming, Zhang Xiaoli, Wang Feng, Dai Wei, Yang Qiuping

Date: 2018-05-29T00:00:00+00:00

Abstract

The MingantU SpEctral Radioheliograph (MUSER) has now entered routine observation and will inevitably generate massive observational data. To address the heliograph' s requirement for efficient management of these data and provision of efficient retrieval services, a negative database management system has been proposed and applied to the heliograph. The design and implementation of the negative database interface are introduced in detail, and the effectiveness and performance of the designed interface are verified through experiments. The negative database management system can not only effectively solve the problems at hand, but also provide a reference for negative database management systems for next-generation telescopes.

Full Text

Preamble

Design and Implementation of the MUSER Negative Database Interfaces

Congming Shi[1,2], Xiaoli Zhang[2], Feng Wang[1,2,3], Wei Dai[2,3], Qiuping Yang[2,3]
[1] Faculty of Management and Economics, Kunming University of Science and Technology, Kunming 650093, China
[2] Key Laboratory of Applications of Computer Technology of Yunnan Province, Kunming University of Science and Technology, Kunming 650500, China
[3] Yunnan Observatories, Chinese Academy of Sciences, Kunming 650011, China

Abstract

The Mingantu Spectral Radioheliograph (MUSER) has entered routine observation and will inevitably generate massive amounts of observational data. To

address the need for efficient management and retrieval of this data, a negative database management system has been proposed and implemented for MUSER. This paper details the design and implementation of the negative database interfaces and validates their effectiveness and performance through experiments. The negative database management system not only effectively solves the challenges faced by MUSER but also provides a valuable reference for next-generation telescope negative database management systems.

Keywords: Interface design; Negative database; MUSER

The concept of negative databases has been applied to survey methods, enabling the protection of respondents' privacy information. Reference [5] proposes a security protection algorithm using negative database concepts to provide an additional security layer for generic positive databases. Reference [6] applies negative database concepts to biometric databases, achieving privacy protection for biometric data. Reference [7] implements a network-based data management system using negative database concepts, offering a higher level of data security.

The negative databases discussed in the above literature are primarily applied in privacy protection and data security, where deriving the positive database from a negative database is an NP-complete problem. However, the negative database designed for MUSER not only reduces the storage capacity required for metadata of massive data frames but also enables rapid derivation of metadata information corresponding to data frames from retrieved complement information.

2. Interface Design and Implementation

We have further refined the functionality of the negative database prototype system and abstracted the negative database interfaces based on this foundation. This abstraction facilitates porting or integrating MUSER' s negative database system into other telescope data management systems with minimal code modifications. Following a brief introduction to MUSER' s data characteristics, complement theory, and record format, we focus on the design and implementation of several key negative database interfaces.

2.1 Data Characteristics of the Mingantu Spectral Radioheliograph

The current data storage system organizes data hierarchically by directory, file, and data frame. All observational data are stored as files, with low-frequency and high-frequency array files placed in separate directories. The 19,200 consecutive data frames generated each minute are encapsulated into a single file, with each frame containing observation date, time, band, polarization, visibility data, autocorrelation data, and other information. File names follow the "YYYYMMDDhhmm" format (YYYY: year, MM: month, DD: day, hh: hour, mm: minute) based on the observation date and time of the first frame in the file. Due to limitations in the current storage system performance, data frames are randomly lost when written to disk, making it impossible to guarantee that

all 19,200 frames encapsulated in a file belong to the same minute. Additionally, since observation is initiated manually, it cannot be guaranteed that the start button is pressed exactly at a whole minute or second. These two factors cause data frames from the same minute of observation to be stored across two data files.

2.2 Complement Theory

Assume a known universal set U, a dataset A, and dataset C as the complement of dataset A. The negative database management system is based on the following assumptions: (1) the universal set U is known and can be precisely defined; (2) all records can be derived from given initial conditions; and (3) dataset A can be derived from its complement C. The MUSER negative database management system designed based on complement theory can store minimal lost-frame information and ensure no loss of metadata for any data frames already saved in files through extensive logical derivation operations.

2.3 Record Format

We designed a record format to represent the logical relationships between data files, observation date-time, and data frames. For convenience, we denote this format as Record, with its specific structure shown in [Figure 1: see original paper]. Record consists of datetime, file1, and file2. The datetime component comprises year (yyyy), month (MM), day (dd), hour (hh), and minute (mm). The file1 component includes filename, time of the first frame in the file (timeF), time of the last frame in the file (timeL), and the logical relationship of frames within the file (FrameR). Both timeF and timeL share the same format, comprising year (yyyy), month (MM), day (dd), hour (hh), minute (mm), second (ss), and millisecond (ffffff). The logical relationship of frames within a file is described by start offset (S), band of the frame at start offset (B), polarization of the frame at start offset (P), end offset (E), and cumulative number of lost frames (CA).

The related interfaces and classes are shown in [Figure 2: see original paper].

2.4 Interface Design and Implementation

Based on MUSER' s data characteristics, complement theory, and the specific record format designed for these characteristics, we have designed corresponding interface functions and classes for MUSER' s negative database, as illustrated in [Figure 2: see original paper]. These interfaces and classes are primarily used for constructing records from observational data files, deriving corresponding data frame information from data records, and connecting to the underlying database. The negative database mainly includes interfaces for underlying database availability validation, telescope data file operations, database synchronization (initialization), data retrieval, and metadata reconstruction. Additionally, the design and implementation of MUSER' s negative database interfaces

comply with the Python DB-API specification to some extent, which reduces interdependencies among system components, enhances cohesion within components, decreases coupling between components, and thereby improves system maintainability and extensibility.

2.4.1 Underlying Database Availability Validation Interface The underlying database availability validation interface (validate_underlying_database_availability) currently supports only MySQL and Redis databases, with plans to add support for other databases in future work. This interface primarily validates whether the underlying database specified in the configuration file (db_config.xml) is supported, whether the corresponding database modules are installed, and whether other database connection-related configuration information is valid. The pseudocode for this interface is shown in [Figure 3: see original paper].

2.4.2 Telescope Data File Operation Interfaces The telescope data file operation interfaces are primarily based on the muserdata.py module in the MUSER Data Processing System (MUSEROS), providing operations for MUSER-specific raw data files (RawData[8]). These interfaces, shown in [Figure 4: see original paper], mainly include sub-interfaces for locating frame positions given frame sequences, retrieving metadata such as observation date-time from data frames, opening observational data files, closing observational data files, and skipping a specific number of frames. The core functionality of database synchronization and data retrieval interfaces heavily depends on these telescope data file operation interfaces.

2.4.3 Database Synchronization (Initialization) Interface The database synchronization (initialization) interface primarily provides functions for initial database setup and subsequent periodic extraction of relevant metadata from newly added observational data files for record construction, record generation using this metadata, and storage of records into the underlying database. The pseudocode implementation of this interface is shown in [Figure 5: see original paper], and it represents a core interface of the MUSER negative database.

2.4.4 Data Retrieval Interface The current data retrieval interface provides data retrieval functionality based on given start query date-time (Tstart), end query date-time (Tend), band information, and polarization mode, with more complex query conditions to be improved in future work. The pseudocode implementation of this interface is shown in [Figure 6: see original paper], and it is also a core interface of the negative database. Additionally, the data retrieval interface calls other interface functions such as the metadata reconstruction function (refactor_frame_metadata) and the conversion function (convert_filter_start_end_index) that transforms Tstart and Tend into start and end query frame ranges (IndexS, IndexE).

3. Interface Effectiveness Validation

The hardware environment used for testing the performance of the designed MUSER negative database interfaces includes: an Intel 24-core Xeon(R) E5-2620 v2 @2.10GHz processor, 64 GB DDR3 memory, and 6TB hard disk. The software environment consists of CentOS 7.4 and Python 2.7. The databases used are Redis 4.0 and MySQL 5.7. The test data comprises 400 observational data files (1,572 GB) from routine MUSER-II observations and 400 observational data files (768 GB) from routine low-frequency array observations.

We primarily tested the performance of the designed data initialization and retrieval interfaces based on the currently supported Redis and MySQL underlying databases.

3.1 Database Synchronization (Initialization) Interface Performance Test

To ensure the validity of the database initialization interface performance test, we repeated the database initialization experiment 1,000 times and used the average results as the final measurements. The comparison of average database initialization times is shown in [Figure 7: see original paper]. For the same observational data files, the initialization speed using the in-memory Redis database is slightly faster than MySQL. For different observational data files, the high-frequency array with larger data volume is slightly slower than the low-frequency array with smaller data volume.

3.2 Data Retrieval Interface Performance Test

We tested the response time for retrieving 1, 8, 80, 160, 320, and 640 consecutive data frames from the database. To ensure the validity and accuracy of the retrieval response times, we repeated each data retrieval operation 100,000 times and used the average response time as the final measurement. The comparison of average data retrieval response times is shown in [Figure 8: see original paper]. For the same data, the retrieval performance of the Redis-based negative database is 2 to 6 times faster than the MySQL-based negative database. For both low-frequency and high-frequency arrays, the retrieval performance of negative databases using the same underlying database is essentially identical. As the number of retrieved data frames increases, the retrieval performance of the MySQL-based negative database degrades more rapidly than that of the Redis-based negative database.

4. Conclusion

This paper briefly introduced the theoretical foundation and record format involved in the MUSER negative database management system and discussed in detail the interface design and implementation of the negative database management system. Through experiments, we validated that the designed and imple-

mented interfaces exhibit good robustness, generality, and effectiveness. However, since the negative database was originally designed to address MUSER' s massive data management challenges and uses its specific raw data file format rather than generic formats such as UVFITS or FITS-IDI, the universality and portability of the negative database interfaces require further verification. Additionally, the interfaces in MUSER' s negative database need further improvement. The research results presented in this paper can provide a valuable reference for data management in other next-generation telescope systems.

References

[1] Yan Y, Zhang J, Wang W, et al. The Chinese spectral radioheliograph—CSRH[J]. Earth, Moon, and Planets, 2009, 104(1-4): 97-100.

[2] Shi Congming, Wang Feng, Deng Hui, et al. High-performance Negative Database for Massive Data Management System of The Mingantu Spectral Radioheliograph[J]. Publications of the Astronomical Society of the Pacific, 2017, 129(978): 1-10.

[3] Esponda F, Forrest S, Helman P. Enhancing privacy through negative representations of data[R]. New Mexico: University of New Mexico, 2004.

[4] Esponda F, Forrest S, Helman P. Negative representations of information[J]. International Journal of Information Security, 2009, 8(5): 331-349.

[5] Dubey G, Khurana V, Sachdeva S. Implementing security technique on generic database[C]//Eighth International Conference on Contemporary Computing, 2015: 370-375.

[6] Bringer J, Chabanne H. Negative databases for biometric data[C]//Proceedings of the 12th ACM Workshop on Multimedia and Security. 2010: 55-62.

[7] Patel A, Sharma N, Eirinaki M. Negative database for data security[C]//International Conference on Computing, Engineering and Information. 2009: 67-70.

[8] Wang F, Mei Y, Deng H, et al. Distributed data-processing pipeline for Mingantu Ultrawide Spectral Radioheliograph[J]. Publications of the Astronomical Society of the Pacific, 2015, 127(950): 383-396.

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