ChinaRxiv 中X

AI translation · View original & related papers at
chinarxiv.org/items/chinaxiv-201711.01182

Postprint: Database System Selection for GWAC Massive Star Catalog Data Processing

Authors: Wan Meng, Wu Chao, Ying Zhang, Xu Yang, Wei Jianyan

Date: 2017-10-11T00:00:00+00:00

Abstract

To address the challenges of big data management and real-time processing for China' s wide-field ground-based wide-angle camera array, a design scheme for a time-series data processing and management system based on the column-store database MonetDB is proposed. This scheme fully leverages MonetDB' s characteristic as a unified database platform that integrates data processing and management, and realizes the design concept of "bringing computation to the data" by embedding core data processing algorithms such as cross-identification directly within the database. Simultaneously, multiple key technologies of this scheme have been researched and tested: TPC-H benchmark performance testing; big data loading capability testing and optimization research; implementation and testing of the Zone algorithm based on MonetDB; and testing of customizable function development capabilities. Preliminary research results demonstrate the feasibility of column-store technology, while also providing a detailed introduction to this design scheme. The proposed massive star catalog data processing application scheme, designed based on the column-store MonetDB database, constitutes an efficient astronomical database solution that integrates data processing and management.

Full Text

A Study on Database System Selection for Massive Astronomical Catalog Data Processing

Authors: Yang Xu, Jianyan Wei, Chao Wu, Meng Wan, Ying Zhang
Affiliations:
1. National Astronomical Observatories, Chinese Academy of Sciences, Beijing
2. University of Chinese Academy of Sciences, Beijing
3. Centrum Wiskunde & Informatica, Amsterdam, 1098 XG

Abstract

To address the big data management and real-time processing challenges posed by China' s Ground Wide Angle Camera (GWAC), this paper proposes a time-series data processing and management solution based on a column-store database. This design fully leverages the integrated data processing and management capabilities of the MonetDB database platform by embedding core data processing algorithms such as cross-matching directly into the database, thereby realizing the "bring computation to data" design philosophy. We conducted comprehensive benchmark performance tests, big data loading capability tests with optimization studies, and research and testing of key MonetDB technologies: algorithm implementation and testing, and customizable function development capabilities. Preliminary research results demonstrate the feasibility of column-store database design for massive astronomical catalog storage. This paper presents a detailed introduction to the proposed MonetDB-based solution for massive catalog data processing, which represents an efficient astronomical database solution that integrates data processing and management.

Keywords: Astronomical database; Architecture design; MonetDB; Real-time analysis; Cross-matching

1. Introduction

Modern developments in astronomical observation and data processing technologies have enabled time-domain astronomy to evolve toward larger fields of view and higher temporal sampling rates, injecting new vitality into contemporary time-domain observations. Discoveries of supernovae, particularly early-stage explosions, rapid-response observations of gamma-ray burst optical afterglows, and detection and follow-up observations of microlensing events all benefit from these technological advances. China' s under-construction GWAC consists of an array of wide-field telescopes, each equipped with 4k $\times$ 4k CCDs, providing continuous observations of fixed sky fields for 4-5 hours per night with a 15-second sampling interval. The entire camera array covers a field of view of 5,400 square degrees. In terms of both field size and temporal sampling frequency, GWAC possesses unique advantages for time-domain astronomy.

However, these capabilities present enormous challenges for data management and processing. The GWAC catalog data specifications indicate approximately 5.4×10^4 records per image. With 36 telescopes in the array, this generates 86,400 images per night, totaling 1.5×10^2 TB of data. The database management system must meet three critical requirements: (1) Rapid big data ingestion capability, ensuring all observation catalogs from one night are loaded within 1.5 hours before the next observation night begins; (2) Real-time association computation capability for high-density massive catalogs, matching each night' s catalog data with reference catalogs to generate light curves; and (3) The ability to complete all data processing within the 15-second sampling interval.

The most straightforward database management and processing system design would use a database solely for storage while performing computations through external programs. However, this approach suffers from continuous data exchange between the database and peripheral programs, causing unnecessary I/O overhead and lacking holistic optimization across multiple program components. Database expert Jim Gray successfully developed the SkyServer survey database management system, proposing the design principle of "bringing computation to data rather than moving data to computation." Inspired by this philosophy, we propose integrating GWAC' s data processing and management into a unified database platform.

Selecting an appropriate database platform is critical, as traditional databases lack rapid big data processing capabilities. While the LSST project developed SciDB specifically for astronomical big data needs, it remains in active development with unproven stability and practicality. MonetDB, an open-source in-memory column-store database platform, offers significant advantages: small storage footprint, runtime query optimization, and an internal storage model fundamentally different from traditional databases. In MonetDB, relational tables are vertically partitioned, with each column stored in a separate binary association table (BAT) as key-value pairs (OID, value). This column-store approach enables queries to access only required columns, achieving high compression ratios and cache hit rates when similar data types reside in continuous memory.

MonetDB' s kernel is built on a programmable relational algebra machine using array-like structures, maximizing hardware performance for real-time user response. Its cache-conscious data structures and algorithms in the execution engine can optimize multi-level memory systems at runtime. For large tables where query statements represent a significant proportion—particularly when the ratio of accessed columns to total columns is small—column-store databases substantially reduce query time. This makes them ideal for Online Analytical Processing (OLAP) applications requiring rapid results, which matches GWAC' s query service requirements. MonetDB has proven successful in astronomical applications, including the LOFAR radio telescope survey project' s TKP pipeline managing 40 TB of annual catalog data, and has been tested against SDSS DR7 data.

2. Performance Evaluation

2.1 TPC-H Benchmark Testing and Comparison We conducted TPC-H benchmarks to evaluate MonetDB' s performance compared to traditional row-store databases PostgreSQL and MySQL. TPC-H simulates decision-support applications, similar to the analytical services provided by astronomical databases. The test suite includes 22 complex SELECT queries. Using the DBGEN tool to generate test data, we loaded data into all three databases in their initial states without any additional operations, then executed the query sequence once to measure total response time.

The test environment (Table 1) used a Dawn A650Server with Dual-Core AMD Opteron 2214 HE processor, 4 GB RAM, Fedora release 18 OS, MonetDB Feb2013-SP2, PostgreSQL 9.2.0, and MySQL 5.6.16. With SF=1 (1 GB) data volume, MonetDB demonstrated remarkable advantages: single-query speedup of at least 1/137 compared to PostgreSQL and 1/15 compared to MySQL. The total query time showed MonetDB' s clear superiority for analytical workloads with complex queries.

Parameters of the testing platform

The performance difference primarily stems from MonetDB's column-store architecture. For large tables with high query-to-total-column ratios, column-store databases excel. This explains why MonetDB' s response speed for queries Q18, Q19, and Q21 on the large Lineitem table leads by dozens of times. The results confirm that MonetDB is exceptionally well-suited for GWAC's analytical query requirements.

2.2 Data Loading Performance Tests Data loading tests comprised two scenarios: (1) analyzing loading speed changes as data volume increases, and (2) investigating factors affecting loading speed. We simulated 10,000 rows of catalog data per image and continuously loaded it into test databases.

Using MonetDB' s COPY BINARY INTO command for bulk import:

```
COPY BINARY INTO tablename FROM ('path_{{{to}}_{{file}}}0', 'path_{{{to}}_{{file}}}1', ...,
```

Results showed that as single-table data increased, MonetDB maintained stable performance with an average load time of 0.94 seconds per catalog file, while PostgreSQL averaged 6.0 seconds. MonetDB exhibited occasional spikes every few dozen files due to disk I/O operations when writing data to disk—a memory database characteristic that requires optimization.

Pressure testing with 86,400 binary catalogs (2.47 TB total, 1.4 MB per file) on a server with 2× Intel Xeon E5-2690 v2 CPUs and 256 GB RAM demonstrated that MonetDB' s loading speed decreased noticeably after 30,000 files, with average load time increasing to 1.75 seconds. However, this still meets project requirements. An interesting finding revealed an inverse relationship between file size and loading speed: larger files (16 MB) loaded faster than smaller ones (1-2 MB) due to better disk I/O utilization, suggesting optimization potential through file size adjustment.

[Figure 3: see original paper] MonetDB' s single table data loading test
[Figure 4: see original paper] PostgreSQL' s single table data loading test
[Figure 5: see original paper] 86,400 binary catalogs loading time curve
[Figure 6: see original paper] Different data block loading speed test

2.3 Cross-Matching Algorithm Implementation Cross-matching is the key algorithm for transient source detection and light curve generation in GWAC surveys. Effective partitioning strategies are essential. By dividing the sky into horizontal zones by declination, each source acquires a zone attribute, dramatically reducing comparison operations. This zone algorithm can be integrated directly into the database.

Testing on a mini-GWAC dataset with 15,218 and 12,540 rows respectively showed MonetDB' s superior performance. Using SQL with zone filtering:

```
SELECT x0.id AS id1, t0.id AS id2
FROM extractedcatalog x0, template t0
WHERE x0.zone BETWEEN floor(t0.decl-0.0056) AND floor(t0.decl+0.0056)
  AND x0.decl BETWEEN t0.decl-0.0056 AND t0.decl+0.0056
  AND 3600*DEGREES(2*ASIN(SQRT((x0.x-t0.x)*(x0.x-t0.x)
  + (x0.y-t0.y)*(x0.y-t0.y)
  + (x0.z-t0.z)*(x0.z-t0.z))/2)) < 20;
```

The zone algorithm completed cross-matching in approximately 4.3 seconds, significantly faster than PostgreSQL. Table 2 compares performance with and without zone optimization.

15k × 12k rows cross-match performance with/without zone

2.4 User-Defined Function Testing MonetDB provides User-Defined Functions (UDFs) through a multi-layer architecture (Figure 7) that allows flexible extension of core functionality. SQL statements are parsed into MonetDB Assembly Language (MAL), then compiled into Binary Association Tables (BATs) for kernel execution.

We developed a ColumnCopy UDF for bulk data import. The implementation process involved: (1) adding the function to the MAL list, (2) creating SQL wrapper, (3) updating Makefile dependencies, (4) implementing the C code, and (5) compiling with the server. This function appends data from files directly to column storage, providing a flexible interface for future system development.

[Figure 7: see original paper] MonetDB architecture and its relationship with UDF

3. Preliminary System Design

Based on test results, we propose a distributed database architecture (Figure 8) where each camera corresponds to one database unit, all managed by a control server. This design enables physical data partitioning while the control server handles database unit management and query coordination.

Table 3 defines the core tables: - **image**: Unique identifier for each raw image - **targets**: Point sources extracted by SExtractor from each image - **uniquecatalog**: Unique sources with averaged measurements - **associatedsource**: Links between targets and uniquecatalog entries - **transient**: Detected transient sources - **tempuniquecatalog**: Temporary table for new source processing

[Figure 8: see original paper] Distributed architecture of database array
Table names and functions in the distributed database

The system processes data through: (1) cross-matching observed catalogs with template catalogs, (2) generating light curves for matched sources, (3) identifying transients from unmatched sources, and (4) performing relative flux calibration using median flux ratios (R_m) to normalize magnitudes: normmag $= -2.5\log(\text{flux}/R_m)$.

4. Discussion and Summary

China' s GWAC faces unprecedented challenges in big data processing and management, generating 1.5×10^5 records per camera per night across 36 telescopes with real-time processing requirements. Our proposed solution leverages MonetDB' s integrated data processing and management capabilities, embedding core algorithms within the database to minimize I/O overhead between separate systems.

Preliminary research confirms feasibility through: (1) TPC-H benchmarks showing MonetDB' s superior analytical query performance over PostgreSQL and MySQL; (2) Data loading tests demonstrating stable performance with 1.75-second average ingestion time per catalog, though optimization is needed for memory-to-disk flushing; (3) Cross-matching algorithm implementation achieving 4.3 seconds for 15k × 12k record matching, with potential for further optimization; and (4) UDF development providing flexible extensibility.

The distributed architecture design theoretically enables efficient transient search and massive light curve management. However, actual development requires extensive optimization and debugging. This research establishes a foundation for MonetDB' s application in Chinese astronomy big data processing, offering a viable integrated solution for GWAC' s challenges.

References

1. Xu Y, Wu C, Wan M, et al. A fast cross-identification algorithm for searching optical transient sources. Astronomical Research & Technology—Publications of National Astronomical Observatories of China, 273-282.
2. Luo Q, Zhao Y, Wang S, et al. Accelerating astronomical image subtraction on heterogeneous processors. 2013 IEEE 9th International Conference on eScience. 2013.
3. Zhao B, Luo Q, Wu C. Parallelizing astronomical source extraction on the GPU. 2013 IEEE 9th International Conference on eScience. 2013, 70-77.
4. Wang S, Zhao Y, Luo Q, et al. Accelerating in-memory cross match of astronomical catalogs. 2013 IEEE 9th International Conference on eScience. 2013, 326-333.
5. Szalay A S, Blakeley J A. Gray' s laws: database-centric computing in science. Microsoft Research. 2009, 5-11.
6. Scheers L H A. Transient and variable radio sources in the LOFAR sky.

7. Ivanova M, Nes N, Goncalves R, et al. MonetDB/SQL meets SkyServer: an architecture for the challenges of a scientific database. 19th International Conference on Scientific and Statistical Database Management. 2007.

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

Source: ChinaXiv —Machine translation. Verify with original.