Design of Real-time Database in Small Satellite Integration Testing System

Li Yin$^{1,a,*}$, Wang Cuilian$^{2,b}$ and Wu Haichao$^{1,c}$

$^1$DFH Satellite Co. Ltd., Beijing 100094, China
$^2$Beijing Institute of Spacecraft System Engineering, Beijing 100094, China
$^a$yin_li_bit@126.com, $^b$cuijian_wang_bit@126.com, $^c$wuhaichao2017@126.com

Keywords: small satellite; SQL Server relational database; real-time database; JSON serialization; SqlBulkCopy class.

Abstract: Aiming at reducing the degree of difficulty in data access and data migration of the real-time database, this paper presents a design method for real-time database in small satellite integration testing system based on SQL Server relational database. This real-time database adopts JSON serialization and WriteToServer method in SqlBulkCopy class to batch import the telemetry data into the database. This method realizes real time telemetry data storage and then sets up the real-time database.

1. Introduction

Real-time database [1] is a branch of database system development, which is the product of the combination of real-time system technology and database technology. The real-time database is mainly used to store real-time test data generated during the integration test of small satellites [2-3], including engineering value and source code of telemetry parameter, frame source code, packet source code, telecommand data, test data and instructions of ground equipment.

The real-time database requires high real-time performance, high data throughput, general system platform, interface specification and other characteristics. In order to meet these requirements, the existing real-time database is usually designed as a data platform with integrated management and control, and the mature relational database interface features are shielded. Meanwhile, the existing real-time database usually adopts so complicated compression technology, which makes it difficult to implement data access and data migration. In view of the above problems, this paper proposes a design method of building a real-time database in small satellite integration testing system based on relational database. This method can effectively reduce the construction time of real-time database, and has been successfully applied in many satellites ground integration test.

2. Software Architecture

2.1. Overall Structure

The main structure of the real-time database in small satellite integration testing system based on
relational database is shown in Figure 1. The real-time database adopts client/server (c/s) structure, the server is SQL Server relational database, and the client is real-time telemetry data processing software. The database, the package basic table, the parameter basic table and the package data table are created in SQL Server relational database. The above database and tables all have mature relational database interface characteristics, and data query, acquisition and processing support standard SQL statements. The processing software is realized through software programming and developed in C# programming language, and the development environment is visual studio 2010.

![Diagram of real-time database structure](image)

**Figure 1** Structure of real-time database

### 2.2. Design of Processing Software

The processing software is the core of the real-time database and adopts modular design method [4]. The functional module includes database management module, basic info query module, data preprocessing module, data bulk storage module and telemetry data query module. The system structure of the functional module is shown in the above figure.

Database management module receives configuration information, including database IP, database name, etc. After obtaining the storage path of the server SQL Server relational database, the database, the package basic table, the parameter basic table and the package data table are sequentially created, and the database configuration success message is sent. Besides, this module provides an interface for updating the package basic table and the parameter basic table, such as adding, deleting, and modifying and other operations.

The basic info query module receives the query request and sequentially completes the query operations of the package basic table and the parameter basic table. The query result of the package basic table is consist of ID, code, name, status and other package properties. The query result of the parameter basic table includes ID, code, name, type, upper limit value, lower limit value, processing...
method, length, offset, mask, and status. After the query operation is finished, the query result is returned to the external module.

Data preprocessing module receives real-time telemetry data, including package ID, package source code, onboard time, local time, parameter ID, parameter source code, and parameter engineering value. A data cache is established to store all received real-time telemetry data.

According to the transmission rate of real-time telemetry data, the data bulk storage module sets the timer. In the timer response function, the real-time telemetry data is taken out of the data cache in turn. JSON serialization is adopted to convert package ID, package source code, parameter ID, parameter source code and parameter engineering value into JSON string, then the string is stored in DataTable object with onboard time and local time. Create SqlBulkCopy object and use WriteToServer method to batch insert all DataTable object into real-time database.

Telemetry data query module supports two kinds of timestamp query operations: onboard time and local time. The query result includes onboard time, local time and JSON string. JSON string is parsed into parameter code, source code and engineering value, and then the parsed result is returned to the external module.

2.3. Workflow

| Create Real-time database, package basic table, parameter basic table and package data table |
| Establish the data cache to receive real-time telemetry data |
| Start the timer; JSON serialization real-time telemetry data |
| Create DataTable object to store JSON string |
| Create SqlBulkCopy object and set properties |
| Use WriteToServer method to batch insert all DataTable object into real-time database |

Figure 2 Flow chart of the process software

First, the processing software connects to the remote database, and real-time database, package basic table, parameter basic table and package data table is created in it. Second, establish the data cache to receive real-time telemetry data. Third, start the timer to handle real-time telemetry data in
the data cache. Last, all of telemetry data is stored in real-time by the joint work of JSON serialization and WriteToServer method in SqlBulkCopy class. The workflow of the processing software is shown in Figure 2.

2.3.1. JSON Serialization

As a lightweight data transmission format, JSON [5] can exchange data between languages. JSON is a subset of JavaScript specifications, easily to read and code, and it can be parsed by browsers that support JavaScript. Compared with XML, it reduces the performance and compatibility problems brought by parsing. During JSON serialization, the process of converting the object state into a maintainable format. This paper uses JSON serialization to convert real-time telemetry data into JSON string, realizing lightweight compression of real-time telemetry data and saving storage space. At the same time, it simplifies the difficulty of data table design and real-time telemetry data access.

2.3.2. SqlBulk Copy Class

SqlBulkCopy [6-7] is a new class in the .NET Framework 2.0, belongs to System.Data.SqlClient namespace, and mainly provides the ability to efficiently bulk copy data from other data sources to SQL Server database. The principle of SqlBulkCopy is to use the bcp command provided by SQL Server to batch data copy. The SqlBulkCopy class contains WriteToServer method, which can handle data types including DataRow array, DataTable, and DataReader. This paper chooses DataTable as the data container. The real-time telemetry data transmission rate is high and the data volume of the DataTable is large. Therefore, it is necessary to set the batch size property of the SqlBulkCopy object so that the loaded data is stored in the database in batches. At the end of each batch copy, the rows in the batch are sent to the SQL Server database.

There are several methods to copy a large amount of data into a database in batches: ExecuteNonQuery method in SqlCommand class, Update method in SQLDataAdapter class, and WriteToServer method in SqlBulkCopy class adopted in this paper. 10,000 pieces of data was inserted into database on the same client machine to test the operating time of the three methods, the result is shown in Figure 3. Compared with other two methods, WriteToServer method in SqlBulkCopy class has obviously improved writing speed, which can avoid the problem of slow software running caused by data bulk copy.

```
ExecuteNonQuery method
Insert Success Time:108722 ms

Update method
Insert Success Time:15961 ms

WriteToServer method
Insert Success Time:1132 ms
```

Figure 3 Comparison of the run time

3. System Application and Characteristics

The real-time database designed in this paper can receive data from multiple data source, such as satellite real-time telemetry data, network telemetry data, telemetry data files, etc. Telemetry data
file is served as the data source. There are 20 telemetry data packets when telemetry data file is played at normal speed, so if the telemetry file is played at triple speed for 1 hour, it will generate 216,000 packets. The long-term running effect of the real-time database is tested by circularly inputting 216000 records. The result is shown in Figure 4. The average time is about 50,000 ms, which is far less than the actual generation time of 1 hour, and there is no significant increase in recording time during long-term operation.

![Figure 4 Result of the run time](image)

At present, the real-time database has been used in many satellites ground integration test. It has the following characteristics.

1) The real-time database has the mature relational database interface features, and real-time telemetry data can be directly obtained by SQL Server client software. The real-time database supports Microsoft ADO object interface. Besides, data query, acquisition and processing of the real-time database can be realized by standard SQL statements.

2) JSON serialization is used for lightweight compression of real-time telemetry data, which saves storage space and simplifies the design difficulty of package data table. The real-time telemetry data is stored in SQL Server relational database that provides great convenience for the data transplantation in the late of the satellite ground integration test.

3) Compared with ExecuteNonQuery method and Update method, WriteToServer method has obviously improved writing speed, which can avoid the problem of slow software running caused by real-time telemetry data bulk copy to the database.

4) The real-time database can run alone or work together with other software, effectively reduce the construction time of real-time database, and greatly improve the work efficiency of the satellite ground integration test.

4. Conclusion

This paper proposes a design method of building a real-time database in small satellite integration testing system based on relational database. The telemetry data is stored in real-time by the joint work of JSON serialization and WriteToServer method in SqlBulkCopy class. This
real-time database has been used in many satellites ground integration test. Meanwhile, it can also be used as the real-time database and the historical database during satellite in-orbit to assist administrative staff in the operation and control management of the satellite. The future research and development will focus on further optimize the design of the data structure of the real-time database to improve the performance and reliability.

References