2008 International Conference on Computer Science and Software Engineering

Data Updating and Query in Real-time Data Warehouse System

Youchan Zhu
Information and Network
Management Center, North China
Electric Power University,
Baoding, 071003, China
hdzhuyc@163.com

Lei An
Information and Network
Management Center, North China
Electric Power University,
Baoding, 071003, China
hdal9936@126.com

Shuangxi Liu
Department of computer，
North China Electric Power
University, Baoding, 071003,
China
hdlsx@126.com

Abstract

The issue of data updating is the most important issue facing organizations deploying real-time data warehouse solutions. This paper discussed the real-time data warehouse implement method, proposed feasible real-time data warehouse architecture based on SOA. This SOA based real-time data warehouse architecture uses the web service to pack the various source database systems, and the various changed data was captured by the data capture web service. When it comes to the data transformation and flow, the update strategy based on web service and xml are used for the real data warehouse real-time updating. Also the multi-level real-time data cache is used for real-time data storage. Using this real-time data warehouse architecture, we can implement the real-time data warehouse easily.

I. INTORDUCTION

In today's fiercely competitive marketplace, companies have an insatiable need for information. Key to maintaining a competitive advantage is understanding what your customers want, what they need and the manner in which they want to receive your products or services. The ability to access meaningful data in a timely, efficient manner through the use of familiar query and analysis tools is critical to realizing competitive advantages. Equally important is the moving and sharing of data throughout an organization, between departments, offices and business partners [1].

Real-time data warehouse is the combination of the real-time behavior and the data warehouse [2]. Given the benefits of real-time data warehousing, it is difficult to understand why the "snapshot" copy process has prevailed. Currently, the dominant method of replenishing data warehouses and data marts is to use extraction, transformation and load (ETL) tools that "pull" data from source systems periodically – at the end of a day, week, or month – and provide a "snapshot" of your business data at a given moment in time. A real-time data warehousing solution and framework can commonly be divided into three fundamental tiers with data flows between them. The three layers are Presentation Layer, Architecture Layer, and Middleware Layer. These tiers or layers must be seamlessly integrated and function as one to ensure the immediate success and long-term benefits of a real-time data warehouse. The most important technology is the real-time data update of the data warehouse. But the traditional ETL process can not support the exact real time data updating. After the SOA technology appeared, we can do the real-time acquisition based on SOA and web service. Most researchers proposed the CTF (Capture-Conversion-Flow) process to update date in real-time data warehouse. CTF is a simple and effective data transfer technology in the heterogeneous systems [3]. This is the real-time date updating process. Data can be transferred to data warehouse phase table from the OLTP system in a very low-delay when changed.

In this paper, we will discuss the real-time data warehouse architecture, real-time data capture, data updating and query in the following there parts separately.

II. REAL-TIME DATA WAREHOUSE ARCHITECTURE

Figure 1 illustrates the multi-cache based real-time data warehouse architecture, mainly made up of OLTP systems, changed data capture(CDC), ETL, multi-level cache, data warehouse, real-time data integration(RDI), OLAP server and the applications. The key problems of real-time data warehouse foundation are changed data capture, data updating and real-time data query.

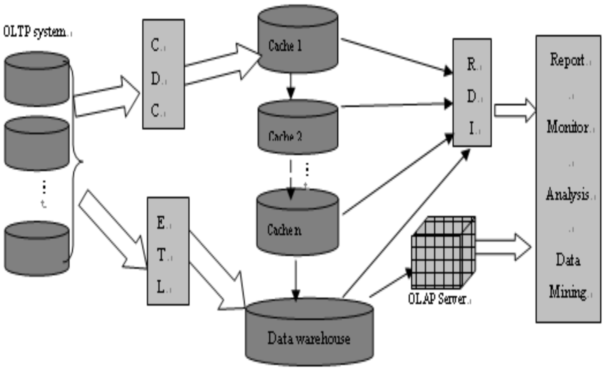

Figure 1. Real-time data warehouse architecture

ETL is playing an extremely important role in data warehouse's establishment and the maintenance process, it is the bridge between the data source and the data warehouse [3], simultaneously real-time data capture is the foundation of real-time data warehouse. In this article, the real-time data is captured by the web service, and changed data will be extracted to the real-time data cache.

In this architecture, data updating is based on the xml and message queue. Using the xml as transmit data format has shielded the isomer source systems. Also the multi-level

978-0-7695-3336-0/08 $25.00 © 2008 IEEE
DOI 10.1109/CSSE.2008.78

cache we used for real-time data storage facilitates the data updating and the real-time data query.

In the following we will discuss the implementation of the data capture, data updating and the real-time query of the cache separately.

III. REAL-TIME DATA CAPTURE BASED ON WEB SERVICE

Today, more and more businesses using a data warehouse are beginning to realize they cannot achieve point-in-time consistency without continuous, real-time change data capture. There are several techniques used by data integration/replenishment software to move data. Essentially, integration tools either push or pull data on an event driven or polling basis.

In this article we use Web Services as the base data capture structure. Figure 2 described a marketing group's data capture, the system including many distributed isomer subsystems (various subsidiary companies), a data gather service and the registration organization. Each isomer subsystem has a Web Services server, Web Services shield the isomer subsystem's internal detail, announces its service connection outward, can respond the gather service's request, and provide the service. UDDI (Universal Description Discover and Integration) is the register center on the internet, and provide the register service. Web Services provides the registration service on Interact the organization. Users can look up for the available Web Services through UDDI registration center. The function of the data gather service is capture the changed data from each Web the Services. The data gather service exchange the XML data with Web service through SOAP/http (Simple Object Access Protocol).

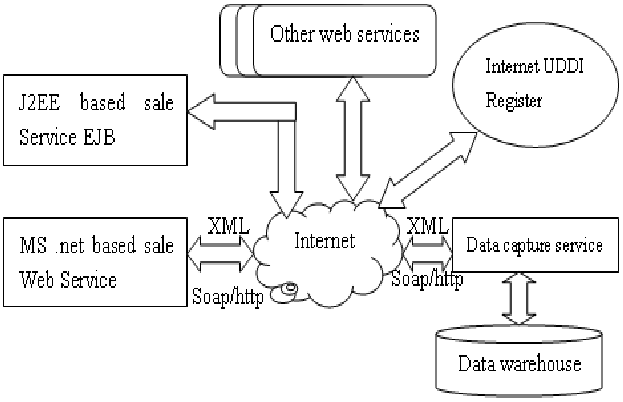

Figure 2. Data capture base on web service

The gather service module captures the changed data by Web Service. Web Services receiving the asynchronous information in XML form. In this paper, the data gather service is complete by Web Services, uses the XML documents to carry on the asynchronous exchange. When gather service needs to gather the data, it translates the content catalog into the XML format query message. The Web Services will return response messages after accepts the request. The response message is encapsulated a SOAP envelope.

IV. DATA TRANSMIT BASED ON XML AND MESSAGE QUEUE MECHANISM

Message queue service is a kind of loose coupling distribute application integration mode. In this mode, the sending and receiving are asynchronous. That is to say the sender and receiver can execute the other codes without wait for the succeed message from the other side. This method greatly increased the capability of affair process. The message send mechanism has the resume ability while breakdown, which make it possible for the integrating of the sender and receiver which were built on different physical platforms.

In the data transmission process, the XML form is used as the intermediate data format, which provides a unified data accessing, transmission format. And it shields the difference of the data format stored in the different relational databases; facilitates the integration of the heterogeneous data source.

In this paper, the mmessage queue manager (MQM) is used for the message management. It is the "heart" of the message middleware. It provides a message queue interface for the queue and message management of the procedure, in order to facilitate the procedure communication.

The message queue manager uses the network equipment existed (e.g. TCP/IP, SNA or SPX) to transmit the message to other queue manager through the channel. The main function of the message manager is shown as following:

- Manage the message queue for the application procedure;
- Provide the program interface for application procedure (Message Queue Interface- MQI);
- Transmit the message to the other queue manager based on the existed network equipment.
- Update the database and queue simultaneously, so the PUT/GET can run simultaneously;
- Do the message partition and the reorganization if necessary, the manager can combine some messages to one physical message and transmit it to the destination, then the destination carries on automatic split and reorganization (considering the capability);

The data transmit mode based on massage queue and XML is shown in Figure 3. Each cache maintains a massage queue, and managed by the message queue manager. The date format transmitted in the queue is XML. The final data loads to the true data warehouse similarly passes through a message queue maintains by the data warehouse. After passed through all caches, the data was loaded to the static data warehouse. The system uses the multistage real-time data caches to store the real-time data, simultaneously different cache stores data with different freshness, i.e. the data was generated in different time. Along with the time increased, the preliminary cache data will flow to high-level cache through message queue Q, i.e. cache will be defers to the respective update cycle to carry on the data the refresh, for example: In the system, Cache-0, Cache-1, Cache-2, the Cache-3 update cycle respectively is 5 minute, 10 minute, 30 minute, 60 minutes, they deposit data in 5 minute, 10 minute, 30 minute, 60 minute separately.

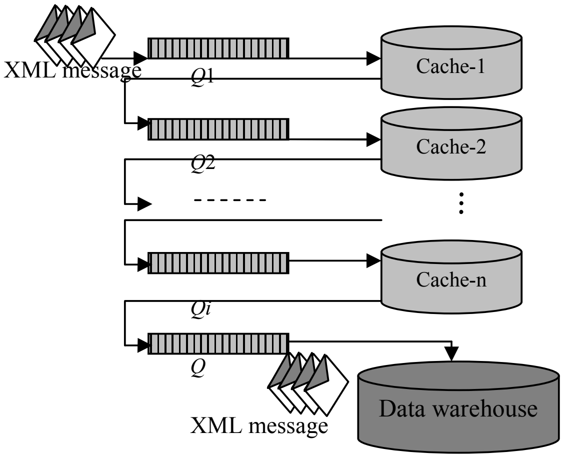

Figure 3. Updating process

V. Real-time data query

The main purpose is to reduce query load in the real-time data warehouse, while the real-time data query load equalization on the caches, the system's stability will be increased. For example: Cache-0, Cache-1, Cache-2, the Cache-3 update cycle are 0 minute, 10 minute, 30 minute, 60 minutes respectively, the data freshness degree are 0, 1, 2, 3 respectively, if the data freshness degree which inquires Q needs is 20 minutes, then it can be satisfied on Cache-2, and the query service will combine the data in Cache-2--Cache-n and the data warehouse together to complete this query. This method assigns the query in each data cache lightens the data warehouse burden, reduced the conflict occurrence. In some situations, the frequent query may concentrate on the data in a period, and makes massive inquiries, increases the query load of the relative caches. In order to solve this problem, we can adjust the update cycle of the cache while implements the data warehouse. If some buffer's load is too heavy, reduces this buffer's update cycle, the data quantity of this cache will be decreased, and the query load will be decreased.

VI. Conclusions

In this article, we analyzed the data update technologies of real-time data warehouse, proposed a new process for the real-time data update. In this data update technology, the changed data will be captured by the data capture service. The SOA technology facilitates system's expansion and reduces the implementation cost of real-time data warehouse. When come to the real-time transmit, use the message queue service and XML data format, realize the data drip loading in real-time data warehouse. This data updating process facilitates the construction of the real-time data warehouse.

References

[1] John Vandermay. Considerations for Building a Real-time Data Warehouse. Data Mirror Corporation. <http://www.dmreview.com/>.

[2] J. Langseth. Real-Time Data Warehousing: Challenges and Solutions. DSSResources.COM, 2004.

[3] YANG Le. Design and Implementation of Real-time data extraction Mechanism in data warehousing[D].Beijing: Beijing University of Posts and Telecommunications.2007.03.

[4] HU Zequn,LI Hua,WU Zhongfu. Data Collection Based on Web Services[J]. Journal of Chongqing University. 2004.27 (5).34-37.

[5] Liya Wu, Gilad Barash, Claudio Bartolini. A Service-oriented Architecture for Business Intelligence.IEEE International Conference on Service-Oriented Computing and Applications.2007.

[6] L. Agosta. “Data Strategy Adviser: Advance from Traditional to Dynamic Data Warehousing,” DMReview.com,December, 2006.