Autonomous Consistency Technique in Distributed Database with Heterogeneous Requirements

Hideo Hanamura, Isao Kaji and Kinji Mori

Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro, Tokyo 125, Japan

Abstract. Recently, the diversified types of companies have been trying to cooperate among them to cope with the dynamic market and thus integration of their DBs with heterogeneous requirements is needed. But the conventional centralized approach has problems of fault-tolerance, real-time property and flexibility.

Here, the autonomous consistency technique in distributed DB and its architecture are proposed. In this architecture, each site has the autonomy to determine Allowable Volume and to update the DB independently using it. In addition, this volume can be managed dynamically and successfully through autonomous communication among sites, and the system can achieve the adaptation to unpredictable user requirements. As an experimental result, it is shown that this mechanism can adaptively achieves users heterogeneous requirements.

1 Background

As the Information Technology advances, the information sharing among companies is essential for business and the integration of their databases(DBs) becomes more important. Though such kinds of DBs have been integrated by centralized approach [2], it has problems of fault-tolerance, real-time property and flexibility. Therefore, the autonomous approach is proposed to solve these problems. In this paper, the stock management system in Supply Chain Management(SCM) is discussed to explain the new approach.

1.1 Needs in SCM

Only the makers and retailers are considered as the constituents in the SCM. The characteristics of heterogeneous requirements of makers and retailers are described as below.

Retailers Retailers are considered to be dealing in two kinds of products, regular products and non-regular products.

Regular products are usually in stock at retailers in enough quantity. When some customers order this kind of products, the retailers ship them from their own stock. If they do not have enough stock, they order them to makers. On the other hand, non-regular products are not usually in stock at retailers. When the retailers receive the order from customers, they order them to the makers and the makers manufacture them at that time.

Makers Makers deal in both regular products and non-regular products in the same way, namely, check the current stock and manufacture them, if necessary.

2 Approach

2.1 Assurance

The **Assurance** is defined as the achievement of the user satisfaction with heterogeneous requirements in the integrated system.

Assurance Usually, a system is constructed to achieve a single user requirement. But the integrated system has heterogeneous requirements, which are realized in each system before the integration. So, These requirements must be realized in the integrated one, even if they are contradictory. When these heterogeneous requirements are realized in the integrated system – if they are contradictory, the system satisfies them fairly –, it is defined that the integrated system realizes the assurance.

2.2 Goal

The following properties are required in the integrated DB system in SCM.

- Real-Time Property
- Fault Tolerance
- Assurance

As mentioned above, the system must realize the assurance. As for the non-regular products, the requirements are same between maker and retailer and the system realizes the **Immediate Update**, which propagates the result of update operation to all the system immediately.

While, in case of the regular products, the requirements are contradictory between maker and retailer. In this case, the real-time property of update at retailers site is given the priority and the result is propagated to all the system at the earliest. which is called as **Delay Update**.

3 Accelerator

3.1 Allowable Volume

The centralized approach is difficult to achieve the assurance when conflicting requirements coexist in a system or user requirements are changing rapidly. Because it does not provide the operational autonomy at each site which is essential in these cases. So, the autonomous consistency mechanism is introduced.

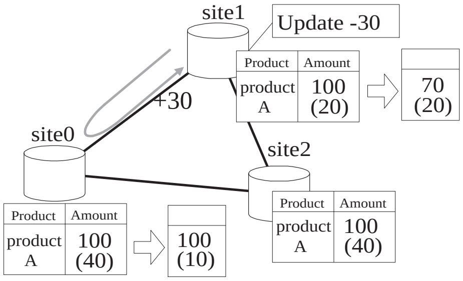

Fig. 1. Example of Allowable Volume

The attribute called as the Allowable Volume(AV) is introduced at each site. The AV is defined on each numeric data in each local DBs. Each site can update the numeric data within it autonomously – without any communication with others. The AV is not fixed volume allocated by some master site but is flexibly managed by communication among sites.

The update of DB with AV is illustrated in Fig. 1, in which each site has 40, 20 and 40 of AV for the product A, totally 100. The stock data of the product A is also 100. If some user updates -30 for product A at site 1, the AV is checked at the site 1 and both the AV and stock data are updated without any communication if the AV is enough. Otherwise, the site requests other sites to transfer the AV. In this figure, since the AV at site 1 is not enough to update, the site 1 requests the AV to site 0 and get +30, then the AV and data at site 1 are updated into 20 and 70.

3.2 System Model in SCM

The structure of the proposed system is shown in Fig. 2.

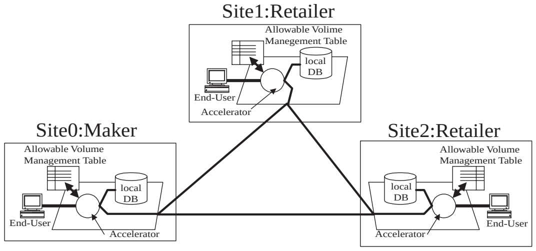

Fig. 2. System Model

In the proposed system, each site has a local DB and they participate in a distributed DB. One of local DBs is called as the base DB which is usually located at makers and works as the primary copy in the case of Immediate Update. The others are located at retailers. The content of all local DBs are the same, which include product names and amount of their stock. In addition, the classification between regular and non-regular products is known. All data are assumed to be delivered to all the sites initially from the base DB.

3.3 Accelerator

The accelerator is proposed to achieve both Immediate Update and Delay Update for realizing the assurance for makers and retailers. It is located at each site with Allowable Volume management table(AV table) and provides DBMS function and AV management function. The AV management function consists of three functions, which is **checking**, **selecting** and **deciding**.

The **checking** function is the function to check the type of user update request with AV table, Immediate Update and Delay Update, when the accelerator receives user update requests. The **selecting** function selects the site to request the transfer of the AV from/to. The **deciding** function decides the volume of AV needed to transfer from/to the other sites, depending on the situation to realize effective AV reallocation.

The accelerator realizes both Immediate Update and Delay Update by using these functions. In following, the behavior of the accelerator is explained.

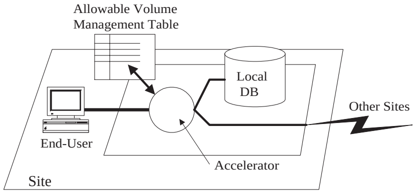

Fig. 3. Accelerator

Delay Update When the accelerator receives the update request, it checks using AV table whether it is Immediate Update or Delay Update. If the AV is defined on it, the accelerator distinguishes it as Delay Update and holds the necessary amount of AV in advance of the update of local DB. In this case, it is not necessary to lock the AV exclusively until the completion of whole transaction. Because the AV is not seen from end-users and if rollback of transaction occurs, the recovery of operation can be done by updating with opposite of update volume, which was used for the transaction at first. Then extra AV can be used by other process while one process accesses the same data.

If the AV is sufficient at the local site, the accelerator updates the AV and completes the update at the local site. But if the AV is not sufficient for update, the accelerator holds all the AV at the site and requests other sites for extra AV. The target site for the request is determined according to the strategy of the accelerator such as the order of the volume the other sites keep. On the other hand, the site receiving the request provides some amount of AV according to its strategy. When the requesting accelerator receives the AV, it checks whether it is sufficient for the update. It requests again to other sites if it is not sufficient. After it gets the enough volume, it updates both data and AV and remaining AV is stored at the local AV table. Otherwise, all accumulated AV is stored in the local AV table.

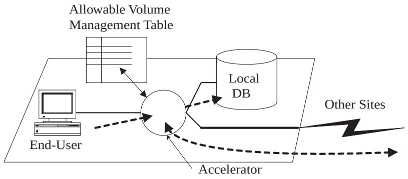

Fig. 4. Delay Update with AV

Immediate Update When the accelerator checks the user requests and finds the AV is not defined on the AV table, it deals with the request as Immediate Update based on primary copy scheme. The requesting accelerator works as the coordinator and the update is processed. At first, it locks the data at the local DB and it also sends the lock request to the other accelerators simultaneously. Then the operations for update are processed at all the sites and ready and commitment messages are exchanged. The requesting accelerator judges the completion of the update with the message from the accelerator at the base DB.

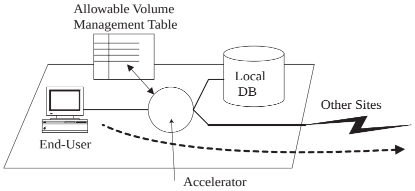

Fig. 5. Immediate Update

3.4 AV management

Each site has its own strategy to manage AV to realize the real-time property and the assurance. But the optimal AV allocation algorithm using global information is not suitable for AV management. Because it is more important to adapt to changing user requirements rapidly. While determining the mechanism of AV management, it is essential to calculate the volume of AV transfer using local information and to make AV circulate among the sites. Then the system achieves the real-time property and the assurance.

4 Simulation

The simulation for the Delay Update is done with the proposed mechanism. A maker and two retailers are modeled as Fig. 2. For AV management, the algorithm which is proposed in the research for electronic commerce money distributed system [1] is utilized. That is, the AV management is occurred in the case when the Allowable Volume is insufficient for update. The requested site is selected according to the amount of AV the site keeps, which information is collected at the necessary communication for AV management and may not be current data. In addition, the requested AV is the amount of shortage needed for the completion of update. And allocated AV is half of the amount of AV that the site keeps.

In the simulation, the number of data items in local DB is 100. In site 0, data is updated to increase the volume by at most 20% of the initial amount of data randomly. On the other hand, at site 1 and site 2, it is updated to decrease at most 10% randomly. The result is shown in the Fig. 6 and Table. 1.

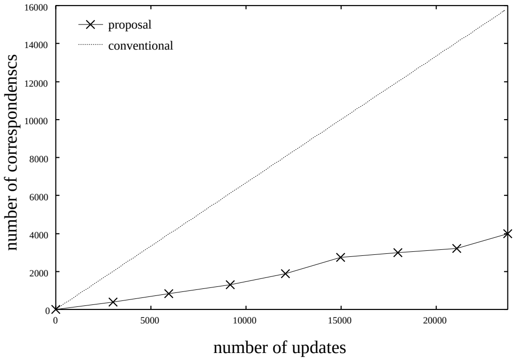

Fig. 6. Number of Updates vs Number of Correspondences

Site	1000	2000	3000	4000	5000	6000	7000	8000
Site1	195	416	669	973	1441	1516	1614	1997
Site2	195	422	634	905	1335	1473	1591	1996

Table 1. Number of Correspondences for Update

In the Fig. 6, the total number of updates in the system is described as the horizontal axis and the number of correspondences for update is the vertical axle (2 messages are counted as 1 correspondence). The line "conventional" shows the number of correspondences for update by the conventional centralized way. The figure shows that the proposed way decreases the correspondences by 75% and most of the update is completed within the local site. Thus the real-time property is attained.

In the Table. 1, it shows the number of correspondences for update in each site. In proposed way, the numbers are almost same between site 1 and site 2 and increases very slowly. That is, it is shown that the real-time property is fairly achieved at the retailer sites – site 1 and site 2.

As the result, in the case of Delay Update, the proposed mechanism is shown to improve the real-time property by decreasing the number of correspondences and to realize the assurance for retailers.

5 Conclusion

In this paper, it is focused on the problem of the heterogeneity of the requirements according to the needs of cooperation among companies in SCM. Then the autonomous approach is proposed to solve this problem. In the proposal, each site can determine the AV, by autonomous communication among them, according to the changing situations. And the data can be updated autonomously at the local site within it without any communication to realize fault tolerance. By the accelerator, both Delay Update and Immediate Update are realized to achieve the assurance for maker and retailer. Then, by simulation, it is shown that the real-time property and the assurance for retailers are achieved. As a result, the proposed mechanism is shown to realize the real-time property, fault-tolerance and assurance, which attains users heterogeneous requirements.

References

1. H. Kawazoe, T. Shibuya, and T. Tokuyama. "optimal on-line algorithms for an electronic commerce money distribution system". In *Proceedings of the 10th ACM SIAM Symposium on Discrete Algorithms (SODA 99)*, pages 527–536, 1999.
2. Amit P. Sheth and James A.Larson. "federated database systems for managing distributed, heterogeneous, and autonomous databases". *ACM Comupting Surveys*, 22(3):183–236, 1990.