Smooth ERP Migration by Using Next Generation Distributed ERP Systems*

Lars Frank, Copenhagen Business School, Howitzvej 60, DK-2000 Frederiksberg, Denmark; E-mail: frank@cbs.dk

ABSTRACT

In a distributed ERP (Enterprise Resource Planning) system, the different local ERP systems are integrated in such a way that each local system can use the resources/stocks managed by the other local ERP systems. Businesses with branch offices may benefit greatly from such systems. In this paper, we will describe how a distributed ERP system can be used to convert/migrate to a new ERP system in a smooth way. The smooth migration property is a consequence of the distributed ERP systems ability to allow different ERP systems to operate on the same stock table. That is, the distributed ERP system can use the stock file from the old ERP system while customers over time are converted to the new distributed ERP system. As this type of converting does not need to take place overnight, it may reduce the risk and peak load for manpower in the converting period.

Keywords: ERP migration, converting risks, converting time, distributed ERP systems.

1. INTRODUCTION

Converting to a new ERP system is normally done overnight for the following reasons:

- Most major companies can only live a short time without an ERP system or the old legacy systems.
- It is not possible to run the new and the old ERP system in parallel as product stocks normally cannot be updated from both systems in a smooth way.

Converting overnight may be risky as the new system cannot be tested in a full scale production environment because the employees rarely have time to use both systems in the test period. A distributed ERP system may reduce the converting risk as such a system will make it possible to update the stock table from both the new and the old ERP systems. This makes it possible to do the converting over time.

A distributed ERP system used for converting must be able to update heterogeneous databases, which will present a problem with the ACID properties. The extended transaction model used in this paper implement the global atomicity property by using compensatable, pivot and retriable subtransactions in that order as described in e.g. Frank and Zahle [2].

The global consistency property is not implemented but it is the responsibility of the application designers to use countermeasures against the isolation anomalies in such a way that the database always will converge towards a consistent state.

The paper is organized as follows: First, we will describe the replication designs we recommend for distributed ERP systems that make it possible for different ERP systems to operate in parallel. Next, we will describe the architecture of a distributed ERP system as such a system is used in the new converting method. Finally, we will describe the method for smoothly converting to a new ERP system.

Related Research: The different replication designs used in this paper are described and evaluated in [4]. The extended transaction model used in this paper is *The Countermeasure Transaction Model* [2]. This transaction model owes many of its properties to e.g. [4], [5], [6] and [7].

2. THE MOST IMPORTANT REPLICATION DESIGNS

In general, replication methods involve n copies of some data where n must be greater than 1. The basic replication designs storing n different copies of some data are defined to be n-safe, 2-safe, 1-safe or 0-safe, respectively, when n, 2, 1 or 0 of the n copies are consistent and up-to-date at normal operation. In autonomous databases, it is normally only possible to use the 1-safe or 0-safe replication designs and, therefore, only these designs are used in this paper.

In general, it is not possible to select one of the replication designs as the best because all the designs have very different properties. However, if one knows the requirements of application, it is possible to select the most inexpensive design fulfilling those needs. This is illustrated by Frank [3].

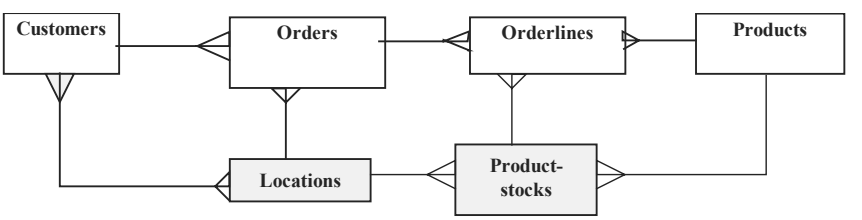
3. ARCHITECTURE OF A DISTRIBUTED ERP SYSTEM

In the following, we will describe in broad outline the table design of some of the most important tables in a distributed ERP system. In a distributed ERP system each local ERP system must have local autonomy to operate in disconnected mode. In the following design, we have also tried to optimize performance, availability and possibility of recovery.

The Product table is vital in all locations. We recommend that the Product table is replicated by using the basic 1-safe design with a central primary copy and secondary copies in all the other locations. We do not think it is necessary to use the n-safe, 0-safe, or expensive versions of the 1-safe designs, because normally the file is only updated from the central location when product prices are changed or new products are introduced. Therefore, updates can be executed in advance with an operation date. This means that in case of failure, it is possible to defer all updates of the Product table until the failure has been repaired, i.e. no lost transactions will occur. In other words, no risks are taken by using the inexpensive basic 1-safe design.

We will recommend that the Customer records for customers who deal with more than one branch office are fragmented and 0-safe with primary copy commit in the central location. The secondary copies will be stored in all the locations where a

Figure 1. Entity-relationship diagram for a distributed ERP system



Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

customer has placed orders previously. The balance of a customer must be updated each time an Order is invoiced, and, therefore, the balance is first updated in the local Customer record. Later, the updating should be committed globally at the central primary copy. The other involved remote branch offices will (sooner or later) receive the updates committed globally by the update of the central primary copy. We will recommend that the Customer records for local customers are fragmented and 1-safe with commutative updates. The primary copies are stored in the locations that deal with the local customers and the secondary copies are stored in the central location.

We will recommend that the Order table is fragmented with the basic 1-safe design as an Order record is normally used only in the location where the corresponding sale took place. However, for backup and data warehousing a secondary copy can be stored in the central location.

Suppose the Product-stock table has a record that includes the quantity of each product in stock at each store location. We recommend that the Product-stock table has the no-replication design, as it may be too expensive to replicate the data on-line. The table is also fragmented in such a way that each store location has its own Product-stock records. (A snapshot copy of the local Product-stock tables should be stored for backup in the central location).

We recommend that the Orderline table is fragmented and 0-safe with primary copy commit in the store location and a secondary copy stored in the location that created the order. In other words, first the Orderline should be created in the database of the order location and later it should be confirmed in a store location where a primary copy of the Orderline is created if the quantity ordered is in stock.

4. CONVERTING TO THE NEW ERP SYSTEM

An overnight conversion may be risky as the new system cannot be tested in a full scale production environment because the employees rarely have time to use both systems in the test period. A distributed ERP system may reduce the conversion risk as such a system will make it possible to update the stock table from both the new and the old ERP systems. This makes it possible to do the conversion over time.

The idea is to run the distributed ERP system in the same location as the ERP system that is going to be converted. All the replicated tables described in example of the previous section are created in the new distributed ERP system except the stock table. By using a special interface, it should be possible to access the stock table in the old ERP system through the new distributed ERP system.

It is possible to create a test department with customers in the distributed ERP system and test how to sell and restore the stocks in a production environment. At the same time, it is possible to use the old ERP system as usual. After a test period, it is possible to convert the sales departments one by one. When all sales departments have been converted to the distributed ERP system, the Stock file may be converted too. Now, the sale module of the old system can be deleted. Other ERP modules may be converted in the same way, but it is important not to delete old tables before all modules that use the old tables have been converted. If the company does not need a distributed version of the ERP system when the conversion has been completed, a non distributed version of the new ERP system can use the converted ERP database of the distributed ERP system. The conversion method described above is not simple. Therefore, the method is only recommended if the overnight conversion is too risky or costly compared to the new method. Another problem with the new conversion method is that special software is needed if the old and new ERP systems are heterogeneous. The special software is necessary to access the old stock file from the distributed ERP system. Special software is also needed for the different replication methods as the replication methods of the distributed ERP system only can be used between homogeneous ERP systems. On the other hand, a great deal of the replication programming has to be done anyway as the tables from the old ERP system must be converted to the new ERP system.

ACKNOWLEDGMENTS

*This research is partly funded by the 'Danish Foundation of Advanced Technology Research' as part of the 'Third Generation Enterprise Resource Planning Systems' (3gERP) project, a collaborative project between Department of Informatics at Copenhagen Business School, Department of Computer Science at Copenhagen University, and Microsoft Business Systems.

REFERENCES

[1] Y. Breibart, H. Garcia-Molina and A. Silberschatz, "Overview of Multidatabase Transaction Management", *VLDB Journal*, 2, pp 181-239.

[2] L. Frank, and Torben Zahle (1998), Semantic ACID Properties in Multidatabases Using Remote Procedure Calls and Update Propagations, *Software - Practice & Experience*, Vol.28, 1998, pp77-98.

[3] L. Frank, 'Architecture for Integration of Distributed ERP Systems and E-commerce Systems', *Industrial Management and Data Systems (IMDS)*, Vol. 104(5), 2004, pp 418-429.

[4] H. Garcia-Molina and K. Salem, "Sagas", *ACM SIGMOD Conf*, 1987, pp 249-259.

[5] S. Mehrotra, R. Rastogi, H. Korth, and A. Silberschatz, "A transaction model for multi-database systems"*, Proc International Conference on Distributed Computing Systems*, 1992, pp 56-63.

[6] G. Weikum and H. J. Schek, 'Concepts and Applications of Multilevel Transactions and Open Nested Transactions', *A. Elmagarmid (ed.): Database Transaction Models for Advanced Applications*, Morgan Kaufmann, 1992, pp 515-553.

[7] A. Zhang, M. Nodine, B. Bhargava and O. Bukhres, 'Ensuring Relaxed Atomicity for Flexible Transactions in Multidatabase Systems', *Proc ACM SIGMOD Conf,* 1994, pp 67-78.

[8] J. Gray and A. Reuter, Transaction Processing, *Morgan Kaufman*, 1993.

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

0 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/proceeding-paper/smooth-erp-migration-using-next/33240

Related Content

Learning With Games and Digital Stories in Visual Programming
Wilfred W. F. Lau (2018). *Encyclopedia of Information Science and Technology, Fourth Edition (pp. 3309-3316).*
www.irma-international.org/chapter/learning-with-games-and-digital-stories-in-visual-programming/184042

Vertical Integration of Science: An Approach to Including Information, Knowledge and Its Organization
Emilia Currás (2012). *Systems Science and Collaborative Information Systems: Theories, Practices and New Research (pp. 1-16).*
www.irma-international.org/chapter/vertical-integration-science/61283

An Efficient Complex Event Processing Algorithm Based on NFA-HTBTS for Massive RFID Event Stream
Jianhua Wang, Shilei Lu, Yubin Lanand Lianglun Cheng (2018). *International Journal of Information Technologies and Systems Approach (pp. 18-30).*
www.irma-international.org/article/an-efficient-complex-event-processing-algorithm-based-on-nfa-htbts-for-massive-rfid-event-stream/204601

Privacy Aware Access Control: A Literature Survey and Novel Framework
Rekha Bhatiaand Manpreet Singh Gujral (2017). *International Journal of Information Technologies and Systems Approach (pp. 17-30).*
www.irma-international.org/article/privacy-aware-access-control/178221

The Past, Present, and Future of UML
Rebecca Plattand Nik Thompson (2018). *Encyclopedia of Information Science and Technology, Fourth Edition (pp. 7481-7487).*
www.irma-international.org/chapter/the-past-present-and-future-of-uml/184445