The Zero-Delay Data Warehouse: Mobilizing Heterogeneous Databases

Eva Kühn

TECCO AG
Austria
info@tecco.at

Abstract

„Now is the time… for the real-time enterprise“: In spite of this assertion from Gartner Group the heterogeneity of today's IT environments and the increasing demands from mobile users are major obstacles for the creation of this vision. Yet its technical foundation is available: software architectures based on innovative middleware components that offer a level of abstraction superior to conventional middleware solutions, including distributed transactions and the seamless integration of mobile devices using open standards, crossing the borders between heterogeneous platforms and systems. Space based computing is a new middleware paradigm meeting these demands. As an example we present the real time build-up of data warehouses.

1. Introduction

In today's IT world data warehouses must be filled with data from an array of distributed and heterogeneous database sources. The general problem is an age old one; that of compatibility and complexity. In any large enterprise system there will exist many different DB platforms (ORACLE, DB2, INGRES etc…) running on many different operating systems and built on a colorful foundation of hardware. To further the problem in this system it is a surety that there will be a variety of DB schemas being used to model more or less the same data.

In addition, the real time enterprise requires data to be always up to date. This criterion enables a certain level of security at the level of decision making. A decision should always be made given valid and up to date data. e.g. Within the banking standard for Basel 2 exists a scenario where the decision of whether a customer may get a loan is based on how many loans he/she already has and the particulars of said loans. To take an example of an enterprise rule in the context of banking; In order to be entitled to a loan the aggregate combined total of a client's loans must not supercede a certain amount. For a decision to be made correctly, based on this very simple rule, highly up to date data is required. The problem is that the data concerning the outstanding loans could exist in different branches of the bank that may be located in different countries and/or even continents. Combined with the aforementioned compatibility and complexity issues this introduces a *world* of complexities into one very simple rule.

2. Space Based Computing Middleware Paradigm

CORSO (Coordinated Shared Object Spaces) is a patented technology for the management of distributed applications in heterogeneous IT environments that has been developed to reduce the mentioned complexity found in distributed systems. CORSO is based on research work

Proceedings of the 29th VLDB Conference, Berlin, Germany, 2003

carried out at Vienna University of Technology and is marketed by TECCO, a university spin-off. CORSO offers a new middleware paradigm based on the idea of a virtually shared memory [KuNo1998, Kuhn2001]. The IT landscape is "abstracted" by the CORSO space: Applications interact exclusively by reading and writing to shared objects in the space, independent of platforms, languages and communication standards. Application programmers see a very simple interface for:

- reading objects,
- writing objects,
- near-time notification,
- flexible transaction control [BuEK1993, Kuhn1994] and
- distributed process coordination.

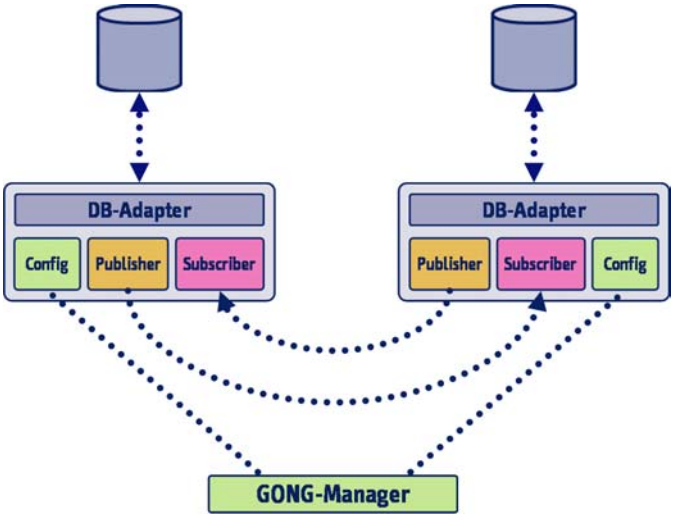
Load and data are distributed (replicated) throughout the network. CORSO's replication protocols keep the copies consistent, guaranteeing their secure and efficient diffusion. There is no need for a central server, consequently no bottleneck can ensue.
Data is never transported unnecessarily (automatic caching). Only changes to data are sent which leads to network traffic reduction. Preference settings determine whether data is sent to a host being needed there (eager pre-fetching) or whether it is only sent upon demand (lazy distribution) and whether it survives host failures (persistence). These configuration options are built-in and require no programming expense.

This "black-board" communication requires neither that the partner processes "know" one another (anonymity), nor that they run simultaneously (asynchronicity). This asynchronous communication is advantageous for publish/subscribe patterns.

CORSO, in contrast to other space based products like Java Spaces [FrHA1999, Ange2003], is more than a mere communication platform. CORSO permits the entire business-logic to be expressed in CORSO space with transaction safety, coordinated by one or more mediators.

Applications can be added to or taken from a running system without requiring that existing components must be rebooted. A new arrival contacts a running CORSO server instance which registers it for participation in the entire virtual space. Clients can also connect directly to a host upon which a CORSO server is running. In this way the space grows organically and offers unlimited scalability.

In contrast to traditional client/server based or message oriented middleware technologies CORSO holds needed data in a cache. The programming effort concerning communication and coordination is heavily reduced. ROI computations for an insurance company business scenario reach up to 80% of code reduction.



CORSO V.3 is available for UNIX, Linux, Windows and z/OS platforms, as well as mobile devices. Language bindings are supported for C/C++, Java and .NET.

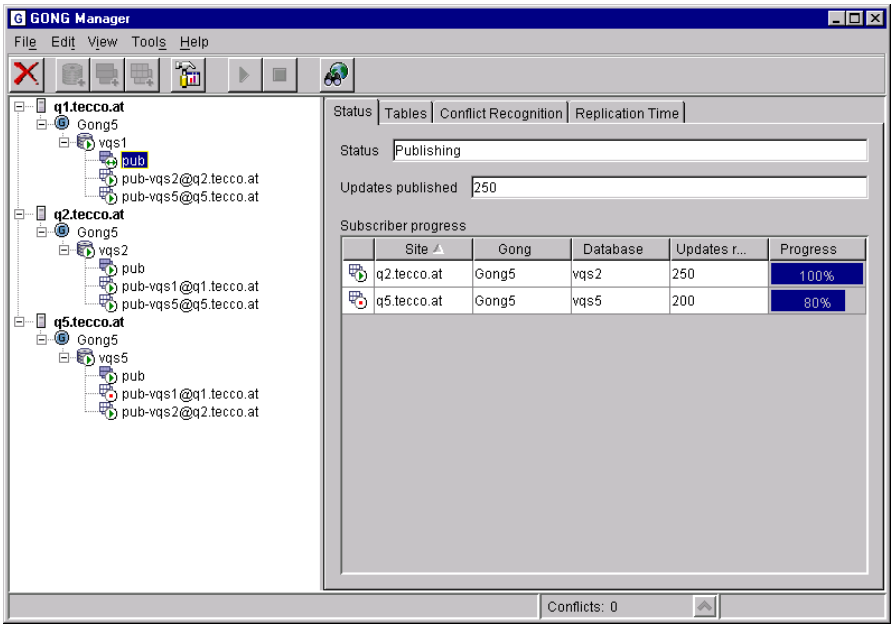
3. Heterogeneous Database Replication with GONG

GONG (General Database Notification Gateway) is a further TECCO product [Bind2003] based on the CORSO middleware, taking advantage of the asynchronous and reliable communication mechanisms. GONG offers uni- or bi-directional replication of data between homogeneous and/or heterogeneous distributed databases – without changing existing applications and without programming effort.

GONG employs a publish/subscribe pattern to control the communication between different databases and consists of three components: GONG Manager, GONG Database Adapter and GONG Service.

- **GONG Manager** serves in configuring and administering the entire replication context.
- **GONG Database Adapter** represents the interface to an individual database model, implemented using the vendor's API.
- **GONG Service** offers automatic recovery after system and network outages, dependable asynchronous communication, data compression on transmission, automatic garbage collection of the transferred data as well as of the database log tables and efficient network-wide data distribution. The GONG Service reads a log of changes written by triggers associated with a given local database and publishes them to a shared object space implemented in CORSO. Other, remote GONG Services are thus automatically and instantly notified of the changes affecting them; expensive polling is thus avoided. The changes are replicated with transaction safety and with the certainty that they can neither be lost nor applied more than once remotely. GONG supports mobile devices despite volatile connections: Changes are applied the moment a previously unreachable device returns online. A schedule for replication (e.g. continuously in near-time or in certain intervals) is fully configurable.

In contrast to data replication supported by database vendors, GONG supports heterogeneous replication in near-time and is very easy to install and administrate.



GONG can be used in many business scenarios: mirror, backup, distribution to branches, replication from branches to a central server (data warehouse), or proxy replication over many levels in a tree structure (e.g. mobile employees of an insurance company that report their data to regional servers from where the data are replicated back to a central server) and full bi-directional multi-master replication (peer-to-peer).

With bi-directional replication, semantic conflicts may occur, which are automatically detected by GONG and which are resolved according to configurable rules.

As databases can differ in their data schemas and in an heterogeneous environment a data warehouse will not always need all data of its sources, GONG supports filtering of data (e.g. selection of certain records according to selectable criteria) as well as data transformation (e.g. time stamp of DB2 into date time format of MS-SQL server).

For very large databases an optimized full loading of all data to be replicated from source into destination database is supported (e.g. DB2 load). This loading is required for the initialization of the replication and can be done also during runtime (re-synchronization).

GONG V.3 supports 24x7 operation (with automatic recovery after network, system and database errors). It is available for Windows and UNIX platforms, supporting adapters for ORACLE, MS-SQL Server and DB2 (also on the mainframe).

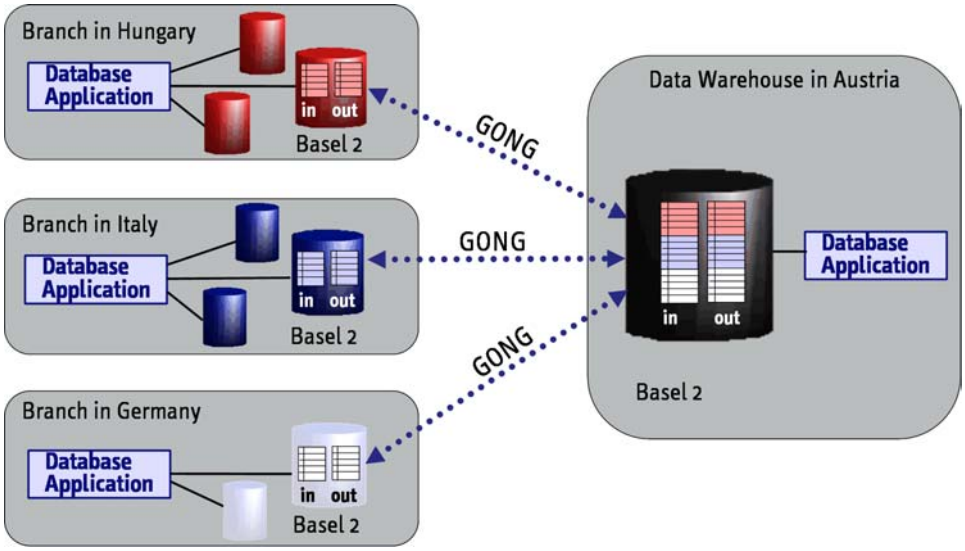
4. Business Scenario: Zero-Delay Data Warehouse with GONG

Given the following scenario:

Big bank Europe has branches situated in multiple countries around Europe. Currently they wish to make use of GONG to synchronize data between some of their banks located in three different countries using rules from the Basel 2 standard. The banks concerned are located in Hungary, Italy and Germany respectively. They wish to use a data warehouse based in Austria, due to its geographically central location, as their central repository.

Using GONG they create an in-out table located at their central repository in Austria. Into this table they then define rules, again based on the Basel 2 standard, to govern data propagation to

the other branches. The Italian branches are for now only interested in customers' credit history and current open loans. The German branches; however, would also like to stay informed of their clients account balance and previous transactions. Using a publisher-subscriber pattern GONG directs the data to those parties concerned. In a nutshell each system can choose the information it wishes to publish and/or subscribe to. For each set of data they can also configure when this set is to be moved. The data can be configured to be propagated eagerly (real-time), at certain times (batch), or as it is needed (lazy).



5. Conclusion

The real time enterprise can take advantage of GONG's database replication for superior management of their data warehouse enabling zero-delay business in order to assist in the daily running and decision making process of the enterprise. Network problems: outages, data errors etc. are made transparent with GONG as GONG bases its communication upon CORSO middleware. Another advantage gained by using GONG is that network traffic is reduced as only the changes are propagated. A further advantage of using this solution is that no expensive additional hardware is needed. There are also many different out-of-the-box solutions provided which have been proven to work easily and flawlessly and all with the minimum of programming effort. Overall GONG enables a zero-delay data warehouse which assists in providing confidence in the data available to every branch of the enterprise.

References

[Ange2003] Bernhard Angerer, Space-Based Programming, 03/19/2003, http://www.onjava.com/pub/a/onjava/2003/03/19/java_spaces.htm

[Bind2003] Thomas Binder, GONG User Manual, 2003/06/24, TECCO Software Entwicklung AG

[BuEK1993] Omran Bukhres, Ahmed K. Elmagarmid, Eva Kühn, Implementations of the Flex Transaction Model, In: Data Engineering Bulletin, Special Issue on Workflow Applications, June 1993.

[FrHA1999] Eric Freeman, Susanne Hupfer, Ken Arnold, Java Spaces – Principles, Patterns, and Practice, Addison Wesley 1999.

[Kuhn1994] Eva Kühn, Fault-Tolerance for Communicating Multidatabase Transactions, In: Proceedings of the 27th Hawaii International Conference on System Sciences (HICSS), ACM, IEEE, January 4-7, Wailea, Maui, Hawaii, 1994.

[KuNo1998] Eva Kühn, Georg Nozicka, Post-Client/Server Coordination Tools, In: Coordination Technology for Collaborative Applications, Wolfram Cohen, Gustaf Neumann (eds.), Springer Series Lecture Notes in Computer Science, 1998.

[Kuhn2001] Eva Kühn, Virtual Shared Memory for Distributed Architecture, Nova Science Publishers, 2001.