View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Institutional Knowledge at Singapore Management University

Singapore Management University
Institutional Knowledge at Singapore Management University

Research Collection School Of Information Systems School of Information Systems

4-1993

Federated Autonomous Databases: Project Overview

Satya PRABHAKAR
Honeywell Inc

Jiandong HUANG
Honeywell Inc

James RICHARDSON
Honeywell Inc

Jaideep Srivastava

Ee Peng LIM
Singapore Management University, eplim@smu.edu.sg

See next page for additional authors

DOI: https://doi.org/10.1109/RIDE.1993.281921

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research
Part of the Databases and Information Systems Commons

Citation

PRABHAKAR, Satya; HUANG, Jiandong; RICHARDSON, James; Srivastava, Jaideep; LIM, Ee Peng; NAVATHE, Sham; SAVARASE, Ashok; and FORESTI, Mark. Federated Autonomous Databases: Project Overview. (1993). *RIDE-IMS '93: Third International Workshop on Research Issues in Data Engineering: April 19-20, 1993, Vienna, Austria: Proceedings*. 216-219. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/923

This Conference Proceeding Article is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email libIR@smu.edu.sg.

Author
Satya PRABHAKAR, Jiandong HUANG, James RICHARDSON, Jaideep Srivastava, Ee Peng LIM, Sham NAVATHE, Ashok SAVARASE, and Mark FORESTI

This conference proceeding article is available at Institutional Knowledge at Singapore Management University: https://ink.library.smu.edu.sg/sis_research/923

Federated Autonomous Databases: Project Overview

Satya Prabhakar, Jiandong Huang, and James Richardson, Honeywell Inc., Minneapolis, MN
Jaideep Srivastava, Lim EePeng, and San-Yih Hwang, University of Minnesota, Minneapolis, MN
Sham Navathe and Ashok Savasare, Georgia Institute of Technology, Atlanta, GA
Mark Foresti, Rome Laboratory, Rome, NY

Abstract

This paper presents an overview of an ongoing research program, Federated Autonomous Databases, sponsored by Rome Laboratory (US Air Force) and conducted by Honeywell in collaboration with the University of Minnesota and Georgia Institute of Technology. This program is exploratory in nature and is aimed at understanding and solving, within the scope of the program definition, the problem of providing integrated access to distributed, heterogeneous and autonomous databases.

1: Program introduction

Honeywell, in collaboration with University of Minnesota and Georgia Institute of Technology, is conducting the Federated Autonomous Databases program for Rome Laboratory (Rome, NY), a research and development organization of the United States Air Force. This paper presents a technical overview of the project.

The program is exploratory in nature with the objective of defining the problem and issues involved in providing integrated access to distributed, autonomous and heterogeneous databases and, within the scope of the program, address some of the issues so identified. After completing requirements analysis, technical assessment of the state-of-the-art, developing a functional architecture and selecting a common data model, we identified schema integration, federated query processing and optimization and consistency management to be the three key areas to be emphasized. Other issues we have been working on are specification and negotiation of a (functional service) contract between users and local databases to allow greater local autonomy, specification and enforcement of weak global integrity constraints and security.

Section 2 of this paper presents our perspective of the problem of federated autonomous databases and describes the philosophy and details of our overall system design. Section 3 explores each major technićal areas in terms of the problems and our approaches. Section 4 concludes this paper. Appendix A contains a list of project papers to date.

2: Functional architecture

2.1: Problem definition

The requirements and design of a federated autonomous database system are different from those of a conventional distributed database system. The designers of distributed databases have the luxury of selecting suitable (typically homogeneous) component databases and designing compatible schemas and management schemes (concurrency control, vertical and horizontal partitioning, security, etc.) The design essentially is top-down with significant centralized control to maximize performance, integrity and security, with component databases playing a subordinate and, to some extent, captive role. In a federated context, however, the design is bottom-up in the sense that *preexisting heterogeneous* databases have to be integrated in a loosely-coupled manner to provide uniform access, while upholding their autonomy. Realizing meaningful integration of these heterogeneous and distributed databases while upholding local constraints and autonomy is quite challenging.

2.2: Architecture

Our federated database architecture is loosely-coupled comprises the following three types of entities which interact with each other on a client-server basis using a platform- and vendor-independent language and protocol.

- **Clients (Global Users):** A client is any global user or application plus any software necessary for him or her to communicate with any of the remote databases (either federated or component)

- **Component Databases:** A component database is an actual source of data and presents its data in terms of the common data model, through the use of a gateway. A component database, at a minimum, processes queries posed against it by the clients and returns the results, if any. A component database has the freedom to dynamically choose the set of services (such as transaction management) to different clients.

0-8186-3710-2/93 $3.00 © 1993 IEEE

- **Federated Databases:** A federated database is a functional component that provides location transparent integrated access to multiple local databases. From a client perspective, federated database appears and behaves just like any other component database. A federated database appears and behaves as a client from a component database point-of-view. Each federated database presents an integrated schema (of two or more selected component database schemas) to its clients and processes their queries using federated query processing and optimization tools. A federated database encapsulates all the integration functionality into one entity that functions autonomously. There can be multiple federated databases, each providing integrated access to a selected set of component databases to an identified group of clients.

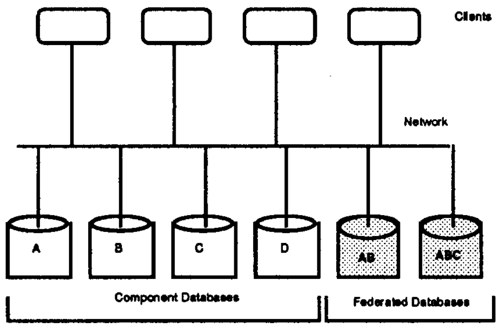

Figure 1. FADB Environment

As shown in Figure 1, an example federated autonomous database environment comprises multiple clients, local databases and federated databases operating autonomously. This environment contains two federated databases (shaded parts) which provided integrated access to databases A and B and to databases A, B and C, respectively. Federated databases AB and ABC are virtual in the sense that they do not contain actual data, but they appear and behave like other local databases to the clients. This architecture allows direct access to individual databases also.

The interfaces between clients and servers are defined by the ISO/ANSI standards: Remote Database Access (for database communication) and SQL3 (for data access). Since the interfaces are standardized, there is no constraint on allocation of function to platform. Entities (clients, component databases or federated databases) may be physically distributed. This flexible and loosely-couple architecture based on vendor-independent, recognized international standards, results in plug-and-play interoperability, extensibility and flexibility while supporting varying levels of integration.

Even though the architecture is loosely-coupled, it allows islands of tight integration. For example, component databases A and B may preclude direct access by users and allow integrated access only through a particular federated database.

3: Research approaches

As indicated in Section 1, we have been focusing on four technical areas that we perceived to be important: schema integration, federated query processing, consistency management and global integrity maintenance. Due to space limitation, we are able to only briefly describe our approaches to selected problems we addressed in each technical area.

3.1: Schema integration

Schema integration must resolve semantic differences among the export databases being integrated. Semantic heterogeneity occurs both at the schema level and at the instance level. Most approaches to schema integration resolve only schema-level conflicts such as homonyms, synonyms, differences of scale, and differences of structure. In the FADB project, we have focussed on how to resolve instance-level conflicts, in particular *entity identification* (how can you recognize and merge references to the same real-world object?) and *attribute value conflict* (what do you do if the export databases record different values for the same attribute of the same real-world object?).

Our work is based on the relational model, specifically SQL. The databases to be integrated are sets of relational tables. The output of the schema integration process is a set of views defined as query expressions over the original database tables. These query expressions include computations to merge references to the same object and resolve attribute value conflicts. The distributed query processor can use query substitution and optimization to produce efficient execution plans for queries against the views. We have defined an interactive schema integration process and the kinds of assertions required for instance-level integration. These assertions are used to generate the view definitions. The meaning of the assertions is defined using first order predicate logic. [Richardson et al. 1993].

There are several points worth noting about this approach to schema integration

- Schema integration resulting in global conceptual schemas is not mandatory. Schema integration is undertaken only when there is a need (identified by a set of users) for providing an integrated view of a selected parts of two or more component databases.

- Schema integration is not a one-time task carried out at design time. It can be done incrementally, view by view, even during system operation.

- The impact of a change to a table in an export schema is limited to the views in the integrated database that reference that table. The integrated database as a whole is not invalidated.

- While a database administrator (DBA) will often be responsible for schema integration, nothing (except any access restrictions) prevents users from creating their own views with the schema integrator. After all, a view is just a query that has been assigned a name and stored in a database. Different users may create different views to meet their application needs.

- The views created by the schema integrator can themselves be the input for further view definitions in the same and other integrated databases.

3.2: Federated query processing

In this area, we are focusing on the following two problems.

Definition of Integration Operators to Handle Instance Level Entity Identification and Attribute Value Conflict Problems: Schema integration and federated query processing in a federated environment that addresses instance level integration in addition to schema level integration, requires a new set of operators to resolve entity identification and attribute value conflict problems. We identified the following to be the operators necessary to address these problems: 1-way outerjoin operator, 2-way outerjoin operator, Key derivation operator and Generalized Attributed Derivation Operator. [Richardson et al. 1993 and Lim and Srivastava 1993]

Definition of a Multi-Phased Query Optimization Strategy: The query optimization function in a federated database environment is complicated because of (i) the possibility that the component databases may not provide statistical information to guide efficient processing, (ii) heterogeneity in local cost models and (iii) the component databases make autonomous databases as to the query optimization strategies. We are developing a multi-phased query processing technique that comprises the following steps: (i) Federated query planning phase, (ii) Local processing and feedback phase, (iii) Optimization phase, (iv) Inter-site processing phase, and (v) Result collection and integration phase. The feedback phase involves the local query agent at the component database site sending back information (e.g., result table sizes) that enables the federated query processor to take appropriate optimization decisions. [Lim and Srivastava 1993].

3.3: Consistency management

The following is a brief description of our work on FADB consistency management is three areas: recovery, concurrency control, and performance studies.

Formal analysis and formulation of recovery mechanisms: Handling global transaction failure, due to concurrency control or site related reasons, is difficult in the presence of local autonomy. Due to autonomy, the participating DBMSs may not support any visible atomic commitment protocol (e.g. 2-phase commit protocol). Our goal is to systematically analyze the recovery problem, to derive conditions for ensuring global serializability in the presence of failure, and to propose recovery algorithms that impose fewer restrictions and minimize the interference to the concurrency. In [Hwang and Srivastava 1992], we identify Local Recoverability as a property of execution at a site. We also derive necessary and sufficient conditions for solving the problem. Furthermore, we propose a recovery mechanism that takes care of both problems by preventing local and other global transactions from conflicting with a resubmitted global transaction during the interval that a prepared subtransaction aborts and its resubmission commits. This recovery mechanism masks the occurrence of failure so that any FADBS concurrency control algorithm designed for a failure free environment, can now be used to achieve global serializability even in the presence of failure.

Development of concurrency control algorithms: A number of concurrency control algorithms have been proposed in the literature. The objective of our work is to develop algorithms that capitalize on the advantages of the existing algorithms and to overcome their intrinsic problems such as violation of local autonomy, high abort rate, and low concurrency. We have developed a algorithm which is able to dynamically adjust global serialization order (called, DAGSO), and proved that DAGSO ensures global serializability and that it is deadlock free [Hwang et al. 1992]. We are also developing a distributed algorithm which provides the fault-tolerance capability [Huang et al. 1992].

Performance studies of concurrency control algorithms: Until now, the performance issue of global concurrency control in federated database systems has not been studied, even though many algorithms have been proposed. This work investigates the performance impact of global concurrency control on global transactions and explored the implications of performance interference between global and local concurrency control algorithms. In particular, we have studied five global concurrency control algorithms, including our new algorithm DAGSO. To conduct the performance study, we developed a federated database simulation model, which contains all the major functional components of global as well as local transaction processing systems. Furthermore, we have implemented those algorithms on a simulation testbed and evaluated them under alternative assumptions about local database system resources and global transaction behavior. The results of this work have been reported elsewhere [Huang et al. 1993].

3.4: Global integrity maintenance

It has been generally concluded that enforcement of global integrity (consistency) constraints in a federated database environment comprising autonomous and heterogeneous databases violates local autonomy and so is of little value. Our approach argues that the definition and enforcement of global integrity constraints has several advantages including maintaining inter-database consistency, defining an accurate integrated view of the world, automating the process of bringing the databases to a consistent state and playing a significant role in federated query optimization. Further, since strict global integrity enforcement necessarily compromises local autonomy, we are developing an active integrity maintenance technique in a federated environment that enforces inter-database consistency in a *weak* manner, without compromising local autonomy. [Prabhakar, Richardson and Foresti 1993].

4: Acknowledgments

This program is sponsored by Rome Laboratory, Rome, NY under the contract #F30602-91-C-0128. The technical monitor for this program is Mr. Mark Foresti. We thank Mr. Joseph Giordano of Rome Laboratory for his encouragement and support.

For a collection of FADB papers, please email or call Satya Prabhakar (satya@ssdc.honeywell.com) at 612-951-7134 (USA).

5: Project publications

Lim, E., Srivastava, J., Prabhakar, S. and Richardson, J., "Entity Identification in Database Integration," to appear in the proceedings of *Data Engineering* Conference, Vienna, 1993.

Richardson, J., Prabhakar, S., Srivastava, J., Lim, E., Navathe, S. and Savasare, A., "Instance Level Integration in Federated Autonomous Databases," submitted to *SIGMOD* 1993.

Prabhakar, S., Richardson, J., Srivastava, J. and Lim, E., "Instance Level Integration in Federated Autonomous Databases," published in the proceedings of *HICSS*, 1993.

Hwang, S. and Srivastava, J., "Transaction Recovery in Federated Autonomous Databases," submitted to *International Journal of Distributed and Parallel Databases.*

Huang, J, Hwang, S. and Srivastava, J., "Concurrency Control in Federated Autonomous Databases: A Dynamic Optimistic Approach," submitted to 13th International Conference on *Distributed Computing Systems* 1993.

Prabhakar, S., Richardson, J. and Foresti, M., "Global Integrity Maintenance in Federated Autonomous Databases," to appear in the proceedings of *TIMS/ORSA* Conference, 1993.