

SODDA – A SERVICE-ORIENTED DISTRIBUTED DATABASE ARCHITECTURE

Breno Mansur Rabelo

Centro EData – Universidade do Estado de Minas Gerais, Belo Horizonte, MG, Brazil

Clodoveu Augusto Davis Jr.

Instituto de Informática – Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte, MG, Brazil

Keywords: Distributed database, service-oriented architecture, web services, database management system, heterogeneous data sources, data integration, distributed query processing.

Abstract: The dissemination of information systems over computer networks has reinforced the interest on distributed database management systems (DDBMS). A review on the architecture of such systems is motivated by the wide availability of networking resources, which allows the cost of communication among the nodes to be reduced. Besides, the coming of age of syntactic interoperability standards, such as XML, and of service-based networking allow for new alternatives for the implementation and deployment of distributed databases. This paper proposes the adoption of elements from service-oriented architectures for the implementation of the connections among distributed database components, thus configuring a service-oriented distributed database architecture.

1 INTRODUCTION

Organizations are actively pursuing the integration of their systems with those of their partners, with the intention of achieving improvements in their production chains and logistics processes. Managing distributed data, stored in physically distinct and independent servers, is often required. As the volume of distributed data grows, the use of distributed database management techniques increases in importance. With such techniques, one can build an integrated database from isolated and independent segments. The cost of high-capacity servers and the operational decentralization of several types of organizations are among the reasons for this tendency.

Kossman (2000) points out that, even though good ideas have been presented by research initiatives on distributed database management systems (DDBMS), the prototypes that have been developed did not make it to commercial tools. Kossman (2000) believes that such research may have taken place ahead of the ideal time, mainly because of the lack of a communications infrastructure as stable and inexpensive as the Internet. With the growing

availability of network communication resources, through the Internet, along with growing quality of service and a strong trend towards cost reduction, some of the obstacles in the path of developing distributed databases have vanished. However, the theoretical background in this field still does not consider peculiar aspects related to the use of Internet standards such as XML (Bray, Paoli et al. 2006), Web services (Alonso, Casati et al. 2004), SOAP (Simple Object Access Protocol) (Curbera, Duftler et al. 2002) and SOA (Service-Oriented Architecture) (Endrei, Ang et al. 2004; Huhns and Singh 2005), among others. Such recent technological elements can be put together, in a review of DDBMS concepts, using SOA to implement Internet-based data exchange. Current demand for distributed databases makes it possible to envision the use of previously developed ideas in a more favorable technological context, thus generating interesting alternatives to centralized DBMS.

Recent works (Campbell 2005; Tok and Bressan 2006) present initiatives related to the integration of SOA and DBMS, thus proposing a service-oriented database architecture (SODA). We propose an extension of SODA, as a new architecture for Internet-

based distributed database management using SOA-based communication.

The remainder of this paper is organized as follows. Section 2 presents related work. Sections 3 and 4 present the proposed architecture and a prototype implementation. Finally, Section 5 presents conclusions and notes for future work.

2 SODA

SODA represents a step towards the integration between database systems and SOA. SODA uses Web services to publish functions and procedures that can handle the content of databases across the Internet. SODA has been proposed (Campbell 2005) in the context of the creation of SOA support structures in the core of Microsoft's SQL Server. The paper presents details on a novel HTTP daemon in the DBMS core, so that the DB server and the Web server can be independent, while allowing the DBMS to respond to HTTP requests.

According to Tok and Bressan (2006), several commercial database management systems are beginning to include features for the integration between Web services and stored procedures, and mention SQL Server as an example. However, this integration is superficial; the database query processor ends up ignoring the communications impact in its cost calculations.

We agree with the notion that the ideas behind the SODA architecture, as proposed by Campbell (2005), can lead to interesting technological solutions to many distributed database issues which have been previously documented. We present a proposal for the evolution of SODA towards distributed databases, in a SOA-based context.

3 SODDA

SODDA works out the main issues involving communication needs in distributed database environments. Basically, SODDA merges DDBMS concepts with new technologies and initiatives associated to SOA and the Internet.

We propose using service-oriented protocols over the Internet as the principal means of communication for a DDBMS. This approach makes it necessary to review the situations in which communications take place among DDBMS nodes, so that Web services and SOA features can be used

as an alternative. We describe the necessary adaptations next.

SODDA uses Web services to coordinate operations among distributed database nodes. Each node includes a Web service to coordinate the local database, and which is capable to respond to a client data provider, called the SODDA Hub, when the node receives requests for queries or other database operations (Figure 1). SODDA Hub can be seen as a common connectivity middleware. It runs on the client's side and it is responsible for all interactions with the SOA components of SODDA. Therefore, in the application developer's point of view, SODDA is a fully transparent and modular distributed data provider that can be used in a way that is similar to OLEDB, JDBC or ODBC.

Unlike SODA, in which Web services are used only to expose stored procedures that can handle data, SODDA uses Web services as fundamental components of the communication among nodes. This is a typical activity in SOA. All operations that are submitted to database nodes are conducted through the SODDA Hub, which is also capable of accessing the distributed database's global catalog.

Direct access to local databases is encapsulated by two elements of the architecture: Node Wrappers and Data Source Wrappers. Web services act as Node Wrappers. In any node of the distributed environment there must be a Web service to respond to client invocations. To execute commands in the local context of a node, the Node Wrapper must use a Data Source Wrapper. These wrappers implement methods to adapt and convert standard SQL into specific query commands at each data source.

In a DDBMS, the global catalog stores data on fragments and their location. In SODDA, the global catalog is replaced by a catalog service, which provides information about the location of Node Wrappers. This brings location transparency, one of the main SOA advantages, to SODDA. The catalog service only contains locations and basic information on Node Wrappers, but includes the statistics service, which contains methods to retrieve data on communication and storage costs from every node, and to prepare these data for use in cost-based query optimization. The catalog service is accessed only to determine the nodes that are possibly involved in the operation, and the statistical data guides the selection of the best alternatives among redundant nodes. The SODDA Hub then exchanges XML documents directly with the Node Wrappers, in order to retrieve data to respond to a client request. This exchange is fundamentally important for the distributed query processing.

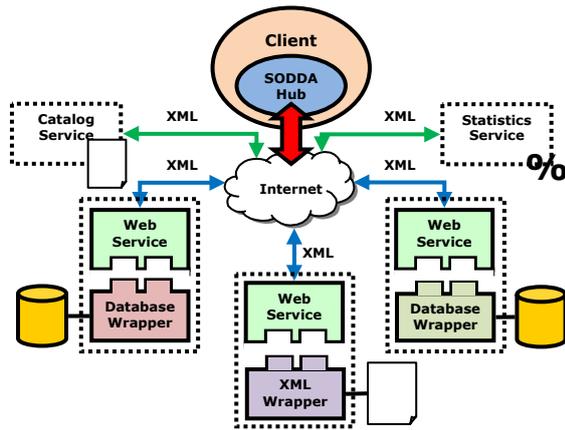


Figure 1: Basic layout of the service-oriented distributed database architecture.

Distributed query processing is naturally one of the most important functions in a DDBMS, since it is used in every data retrieval situation involving fragments that are in various sites. In SODDA, we assume that the distributed database nodes are managed by regular DBMSs with local autonomy. Therefore, these nodes also include the internal features of a conventional DBMS. Considering this, SODDA includes a query processor with optimization in two levels. The first level deals with distribution aspects, based on metadata from the catalog service and from the statistics service. The optimizer gathers statistics on the tables that are involved in a query, using a cost-based strategy supported by the statistics service. The second optimization level handles the part of the query which is transformed into a local query in the point of view of a given node, and executed using the native query processing and optimization methods available to the local DBMS. Since part of the optimization work is performed by the conventional DBMSs, SODDA can focus on distribution and consolidation issues.

The query processing mechanism in the SODDA architecture is based on the basic model presented by Kossman (2000). The query processor is divided into three parts: the Query Decomposition Engine (QDE), the Distribution Optimizer (DO), and the Distributed Execution Engine (DEE) (Figure 2). QDE is responsible for partitioning the queries, finding out which nodes are involved in the operation, decomposing the query into sub queries, based on the fragmentation schema. QDE generates query parts (QP), each of local scope, in regard to a DDBMS node. DO then must determine which tasks can be performed in parallel, building a tree to guide the work of joining the QP. The result of this phase

is the execution plan, obtained from a tree, built from the dependencies among the QP. Independent QP's, therefore, are always defined as leaves in this tree. Finally, the DEE traverses the tree generated by DO, and commands the execution of each QP. DEE can identify nodes which can receive query parts simultaneously, and which therefore can execute their tasks in parallel. When results are received, DEE dynamically creates a representation of global schema from the distributed environment, and copies partial results. Finally, DEE performs the required joins over partial results in a temporary swap database, called Dynamic Swap (DS), in order to generate the final results. The DS is either handled at the SODDA Hub's site, or at a different node, added to the network precisely to support thin or mobile clients.

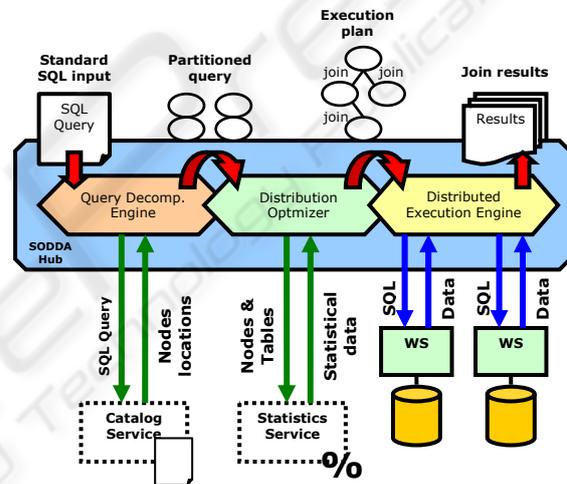


Figure 2: Distributed query processing in SODDA.

At this point, we observe that SODDA presents not only location transparency, but also access transparency, since the access to the distributed database is performed from the client's location, running the SODDA Hub, which concentrates services required to access the distributed database.

Beyond the query processor, a DDBMS must implement some other services to manage transactions and replication features of the distributed environment. We propose three additional services to that effect: the Distributed Transactions Manager (DTM), the Replicated Data Manager (RDM), and the Database Recovery Service (DRS).

The DTM is a mechanism that deals with transaction atomicity in a distributed database environment. In SODDA, the Hub coordinates the execution of the transaction. The original transaction, through the SODDA Hub, starts sub-transactions at each node involved in the process, and waits for their

completion. In case of failure at any particular node, the Hub issues messages, using Web services, to cause a rollback at every node involved. If every node completes its task successfully, the Hub issues messages to cause a commit at every node. The success in the implementation of this procedure requires the implementation of a Web services-based distributed commit protocol.

The RDM implements such a protocol to promote synchronization of data that exist in different nodes and are possibly replicated. RDM uses a publish/subscribe method. The nodes that subscribe to a topic receive change notifications. If a node becomes offline, the previously notified changes must be processed at the moment it becomes online again, before it can accept any further commands.

The DRS is a mechanism that is responsible for finding out whether or when a node becomes inaccessible, and for determining which redundant copy (if any) can replace it during the period of unavailability. This mechanism is integrated to the RDM since, if the distributed database manages redundant data, every modification is propagated using the publish/subscribe notification system, in order to ensure the synchronization. Therefore, available nodes receive changes immediately, and unavailable nodes receive updates later, in a queue.

4 SODDA PROTOTYPE

To validate proposed architecture, we have implemented a SODDA prototype as a Microsoft .NET data provider. However, we cannot provide many implementation details due to space limitations. Results obtained so far demonstrate the feasibility of the ideas and proposals presented here, along with a large potential to support future work, both in research and in commercial applications.

In order to test our prototype, we developed three Data Source Wrappers, respectively for SQL Server, SQL Server CE and Access. Support for data sources such as MySQL, Oracle, XML files, and CSV files is planned for future releases. The prototype currently only implements data retrieval features. Data manipulation is to be included in the next releases of the SODDA data provider.

5 CONCLUSIONS

SODDA intends to use some of the most interesting features of SOA to implement distributed databases. Expected benefits include easier implementation, lower communications costs, and greater access capability. The proposed statistics service facilitates query optimization, since it unifies the treatment of performance-oriented metadata and allows implementation of automatic statistics updating.

An extensive list of possibilities for future work presents itself at this stage. We observe that the distributed database nodes do not, necessarily, need to be managed by a full DBMS. Since they are accessed only through Web services, it would be possible to have other data sources, such as spreadsheets, Web pages and others. *Data Source Wrappers* can be written to use these sources as nodes on a SODDA-based environment. Another possibility involves the implementation of "hot-swapping" of nodes, making it possible to achieve full availability for, say, equipment maintenance, through the simple modification of catalog entries.

REFERENCES

- Alonso, G., F. Casati, et al. (2004). *Web-Services: Concepts, Architecture and Applications*, Springer Verlag.
- Bray, T., J. Paoli, et al. (2006). "Extensible Markup Language (XML) 1.0 (Fourth Edition) - W3C Recommendation 16 August 2006, edited in place 29 September 2006." Retrieved 24/04/2007, from <http://www.w3.org/TR/xml/>.
- Campbell, D. (2005). *Service Oriented Database Architecture: App Server-Lite?* Proceedings of the ACM SIGMOD International Conference on Management of Data.
- Curbera, F., M. Duftler, et al. (2002). *Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI*. IEEE Internet Computing. Vol 6: 86-93.
- Endrei, M., J. Ang, et al. (2004). *Patterns: Service-Oriented Architecture and Web Services*, International Business Machines Corporation (IBM).
- Huhns, M. and M. P. Singh (2005). *Service-Oriented Computing: Key Concepts and Principles*. IEEE Internet Computing. Vol 9: 75-81.
- Kossman, D. (2000). *The State of the Art in Distributed Query Processing*. ACM Computing Surveys. Vol. 32: 442-469.
- Tok, W. H. and S. Bressan (2006). *DBNet: A Service-Oriented Database Architecture*. Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA'06).