

Federated Query Processing Service in Service Oriented Business Intelligence

M.Prabhavathy¹, K.Sivasankari²

Assistant Professor, Dept. of CSE, Dhirajlal Gandhi College of Technology, Salem, Tamilnadu, India^{1,2}

ABSTRACT: In current business scenario, most of the software is offered as services which are exposed using contracts by the service providers. The client or user pays the service provider for the duration of usage. In process of business intelligence data integration is inevitable. Data federation is a form of data integration which is implemented as a service.

The federated query service gets the input query, decomposes into sub queries, post the sub queries to different data sources, extract the required data from sources and integrate into a single virtual view. This paper proposes the algorithms for federated query processing service and uses distributed XML database for data federation. The experimental results are provided.

KEYWORDS: Business Intelligence, Service Oriented Architecture, Data Federation, Analytical Applications

I. INTRODUCTION

Business Intelligence is a broad category of applications and technologies to gather, store, analyze and provide access to data to help enterprise users make better business decisions. The above functionalities are provided as services in service oriented business intelligence framework. Data integration techniques are used to gather the data from distributed heterogeneous data stores.

Data Integration provides a unified view of the business data that is scattered throughout an

organization. Data Integration facilitates query over autonomous and heterogeneous data sources through a common and uniform schema (global schema).

1.1 DATA INTEGRATION TECHNIQUES

There are three main techniques used in data integration [W1]

- a. Data consolidation
- b. Data propagation
- c. Data federation

Data propagation involves replication of data in different locations from different sources. These applications operate online and push data to target store. Updates to a source system may be propagated asynchronously or synchronously to the target system. Enterprise application integration (EAI) is the technology that supports data propagation.

Data federation provides a single virtual view of data from multiple heterogeneous sources without permanently moving or replicating to a new location and enables applications to see the dispersed data as though it resided in a single data source. Data federation is an approach to real-time data integration which is easier and cost effective than data consolidation for certain type of application. Enterprise Information integration (EII) is the technology that supports data federation. A key distinction of EII compared to other integration technologies is that data is not permanently moved or replicated into a new location or server. The source data remains where they are and results persist in the server only as needed for caching.

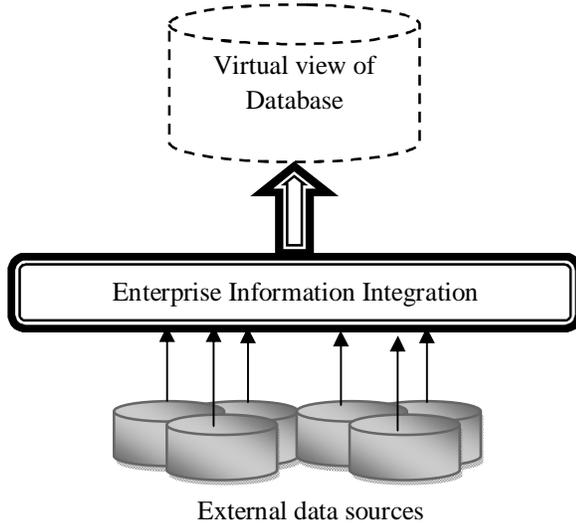


Figure-1 Data Federation

The data federation provides access to current data and removes the need for data consolidation. It pulls data from data source on demand basis.

The data federation is implemented as service in service oriented business intelligence framework. A Service-Oriented Architecture is an application architecture where the functionalities are defined as independent services, with well defined interfaces, which can be called in built-in sequences to build business processes [R1].

This architecture provides a framework that allows heterogeneity, integration and reusability of the participant components in a flexible environment. In this context, it is interesting to analyze how a Federated System [R2] can be designed within the ideas proposed by the Service-Oriented Architecture.

II. RELATED WORK

Integration of heterogeneous distributed data source can be addressed either by replicating the data in a consolidated data store or by creating integrated view on-demand. The later method is more appropriate for dynamically changing business environment.

A few data federation techniques already available in literature are Integration Brokers for Heterogeneous Information Source [R3], Dynamic data

Integration using Web Service [R4], Service Oriented Architecture for Federated Database System [R5], A Multi layered Service Oriented Architecture for Business Intelligence [R6], and Three tier Architecture for Service Oriented Business Intelligence [R7].

A service oriented federated architecture using Integration broker for heterogeneous information sources has been proposed [R3]. This architecture is an amalgamation of Federated database management systems and service oriented architecture. The data access service is constructed using web service; these services are data intensive and are responsible for providing data from heterogeneous data sources. This methodology uses Data as a Service (DaaS) instead of Software as a Service (SaaS). The operational system receives a query from the user, identifies user access rights, locates the source and return the result to the user. The operational system consists of collection of services communicating between web service and user interface. The services are exposed using the Universal Description Discovery and Integration (UDDI) registry where the services have advertised. Clients look at the service definition in the registry and uses the WSDL definition to send messages or request directly to the service via Simple Object Access Protocol (SOAP). This approach is static because in Federated database management system architecture, data source registration and schema integration are assumed to occur 'once-for-all' at set-up time. Depending on user request, the system discovers the services dynamically, invokes them, and provides response to the service consumer. The major limitations of the system are its inability to handle dynamically changing business environment, non-fault tolerant and inability to integrate ontology services.

Dynamic data Integration is achieved using SaaS and late binding mechanism [R4]. It facilitates organizations to share resources in a constantly changing environment with integration of applications and systems. A Service Oriented data integration architecture (SODIA) has been designed to provide a dynamically unified view of data on demand from various autonomous, heterogeneous and distributed data sources. This method accepts internal and external changes of the organizations, including changes in data, data structure, constraints, permissions, data model and semantics. It employs meta-database to ensure any change from the backend data be managed within meta-database. This method provides semantic description of

service using domain ontology and extends UDDI to act as a semantic service registry. The data access control is extended to extract data from dynamic heterogeneous data sources. The domain ontology supports dynamic semantic inter operability. The major limitation of this approach is that it lacks full dynamic discovery and binding processes in a large scale, open and ever-changing computational environment.

An architecture for service oriented federated system. The architecture consists of two components - Federation component and Source component [R5]. The federation component consists of Federation controller, Query decomposer, Integrator, Rights manager and Invoker. Federation controller works as an intermediary between user and federation. Integrator generates the federated schema based on schema that arrives from each source and integrates the access rights. Query Decomposer decomposes the user query over the federation into specific queries to the sources. Invoker identifies the location of each source and invokes its services by their Service Controller. Rights manager will verify access rights that user has over the schema.

The source component consists of Service Controller, Rights manager, Schema filter, Query filter and data source connector. Service controller exports the

services of source and receives orders of the federation. Query filter receives the user query and verify the user rights before executing the query. Schema filter controls the exportation of schema from source. Data-source connector translates the model and query language of each source to the canonical model and language of federation. Rights manager verifies access rights of the user. The limitation of this method is that it does not support components in different languages; and does not integrate with federation management tools.

A multi-layered Service oriented Architecture for Business Intelligence has been proposed by [R6]. This Service Oriented Architecture for IT Performance Analytic [SOA-ITPA] uses five layers - Data source layer, ETL layer, Physical layer, Logical layer and Analytical application layer [Figure-2]. This architecture is built upon reusable components like web services. The main limitation of this method is that, it is difficult to create Integrated Reporting Data Store [RDS] in a changing business environment. As the user will not have access to data source for verification, change in data source schema, if any, requires RDS to be rebuild.

To overcome the limitations like inability to support changing business environment, lack of dynamic discovery and building real-time reporting data store three tier architecture for Service Oriented Business Intelligence has been proposed. This three tier architecture merges the logical layer and ETL layer in to service layer and eliminate the intermediate physical layer [Figure-3].

III. FEDERATED QUERY SERVICE

Federation is incorporated as a service on three tier architecture [Figure 4]. This includes communicating web services and a user interface.

Graphical User Interface (GUI) provides an interface for login, Access Rule Service to authenticate user and display of final result.

Access Rule Service is responsible for initial user authentication and subsequent authorization. Initial authentication is based on username and password. Once the user has been authenticated, access rule service determines the type of data resources the user can access.

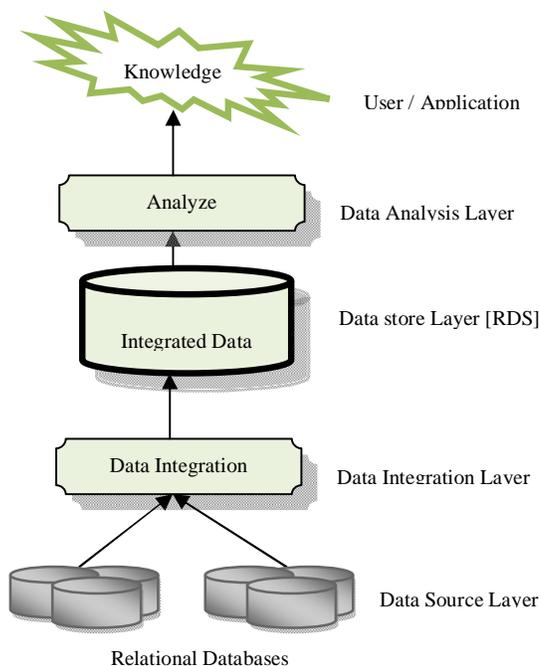


Figure – 2 Multi layered Traditional BI [R6] Architecture

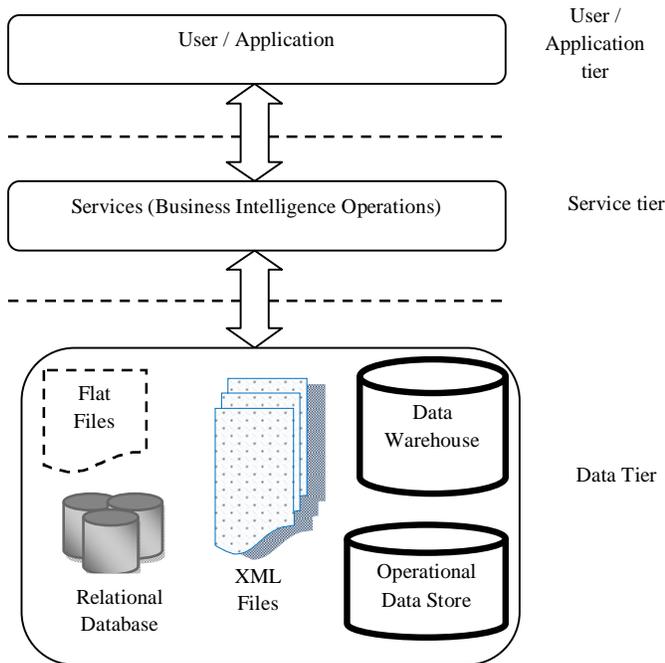


Figure – 3 SOBI Architecture

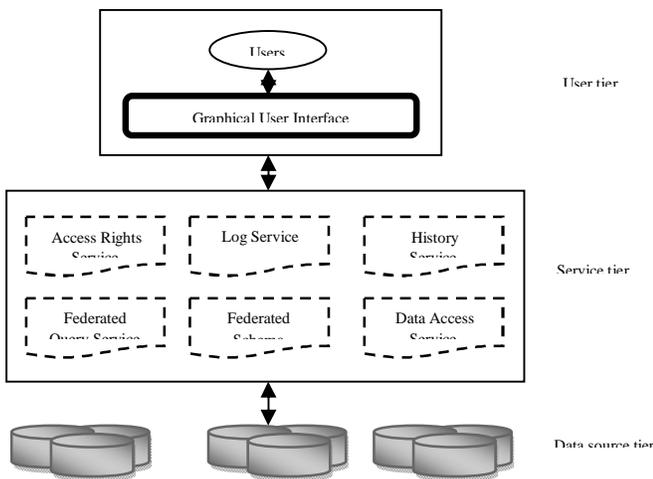


Figure-4 Architecture for Federated Query Service in SOBI

Federated Schema Service maintains federated schema which is constructed by integrating export schemas of individual data sources. It also maintains the export schema of individual data sources that wish to participate in federation.

Federated Query Service is responsible for decomposing the user query into set of local queries in consultation with federated schema service and integrating results from data access service into a federated record.

Data Access Service is responsible for providing data from their respective data sources in a transparent way.

History service will maintain all federated schemas created along with respective query and the log service will maintain transaction details of federation.

On a request from authenticated user for data in multiple heterogeneous data sources, a xpath query based on global schema of federated system for data sources is constructed and forwarded to service tier.

In the service tier, federated schema service maintains the local schemas of different data sources that wish to participate in federation. A local schema is the description of data expressed in native model of database system. As the data retrieved from multiple data sources is in XML form, schemas are also maintained as XML. Federated schema is constructed by integrating local schemas, irrespective of heterogeneities in local schemas.

As in federated system data sources are autonomous, all participating data sources follow their own schemas which typically differs from the global schema. These queries cannot be directly employed to query local sources due to the different structures of the global schema and local schema. In order to access the data from these sources, the input query must be decomposed in to sub-queries. Each of such sub-query conforms to the structure of a local source's schema, that can be executed to get the desired data.

3.1 PROPOSED QUERY DECOMPOSITION ALGORITHM

Query decomposition service accepts xpath query built based on global schema and outputs the sub-query that conforms to the local schema for which the sub query need to be generated. The xpath query based on global schema of the form $/p_1/p_2/p_3/.../p_{n-1}/p_n$ is evaluated from right to left as in xpath query. It is the actual result which the user wants to get from federated system. The local schema tree is traversed from leaf to root level.

If p_n does not exist in local schema, it can be concluded that there is no sub-query for this schema.

Otherwise if p_n is found at a node in the local schema tree, the subsequent searches for node p_i for ($i = (n-1) \dots 1$) will be performed from the ancestor nodes of the matched node. Instead of searching the whole tree only the ancestor nodes of last matched node need to be searched. This can significantly reduce the search time. If node p_i is found, then p_i is concatenated with p_n to form query.

The algorithm for query decomposition is shown below
Algorithm:

Input : userquery Q_{global} based on global schema, local schema S

Output: sub-query for S

Service splitter(userquery, local schema)

1. Initialize the variables 'query' and 'subquery' to NULL.
2. Initialize the arrays parts[], condition[] and value[] to NULL.
3. Split Q_{global} (global schema) and store it in an array parts[].
4. For each parts[i] repeat
 - If parts[i] has condition and value then
 - i. Split them and store in condition[i] and value[i] arrays respectively.
 - ii. Create subquery based on local schema with condition
 - Else
 - i. Create subquery based on local schema
5. Return subquery.

Data access service accepts the sub-query from query decomposition service. This xpath sub-query conforms to local schema of data source in which data access service runs as opposed to user query based on global schema before decomposition.

This service executes the xpath query and outputs a dataset that contains the requested data found in this data source. Each data source has an instance of data access service running at a port receiving the xpath sub-query, executes the query and provides the dataset.

The algorithm for sub query execution and data set generation is as follows

Algorithm:

Input : xpath subquery

Output : Dataset containing requested data from respective XML data source

Service Dataaccess (subquery)

6. Create a navigator XNav for XML Document
7. Evaluate subquery using XNav
8. Create an node iterator XNodeIter for nodes selected with Xnav
9. While there exist nodes in XNodeIter
10. If node has attributes
11. While there exist node.attribute
12. add the value to the Dataset
13. End While
14. If node has child elements
15. While node.child elements
16. add the value to the Dataset
17. End While
18. return Dataset

Result Integrator Service accepts the dataset from different data access service containing the data from multiple data sources. It combines the datasets eliminating any duplicates, orders the data for consistent display of records and sorts the records if required. The combined dataset is displayed to the user.

Algorithm:

1. Input : Dataset DSA,DSB....DSn
2. Output : Merged Dataset DSM
3. Service ResultIntegrator (DSA,DSB....DSn)
4. Dataset DSM := null
5. DSM :=Merge(DSA,DSB....DSn)
6. DSM :=Remove_duplicate_records(DSM)
7. DSM
- :=Sort(DSM,sortcolumnname,ASC/DESC)
8. return DSM

IV. EXPERIMENTAL RESULTS

For experimental study a database containing 10,000 records is used. The databases which serve as data sources are geographically distributed and run in heterogeneous platform with structural and semantic heterogeneity. The data are used are pertinent to car sales and resale of an organization which have number of branches. All these data sources are initially converted to XML databases to suite xpath query decomposition. The local schemas are used for source identification and federated schema generation. Initially 2000 records were populated. The web services were executed from local host. The data federation is performed for the given

query. The above process was repeated by incrementing records by 1000 for every run, till the count reaches 10000 records. The time taken to execute same query increased proportionally [Figure -5].

The web services were placed in a remote server. And the above process was repeated. The time taken to execute query is not proportional with number of records [Figure -6]. The factors like server load, bandwidth availability affects the execution time.

4.1 INFERENCES

The services should be coarse grained for business intelligence operation like data federation. Service composition of fine grain service is difficult for such operations because different service providers design services input in different way. A separate mapping layer is required for service composition to overcome the above difficulty.

In this experiment the schemas were located in a web server. In case if we use composite service, composed of web services belongs to different service providers. The XML schema of different local data base is to be communicated using common message format.

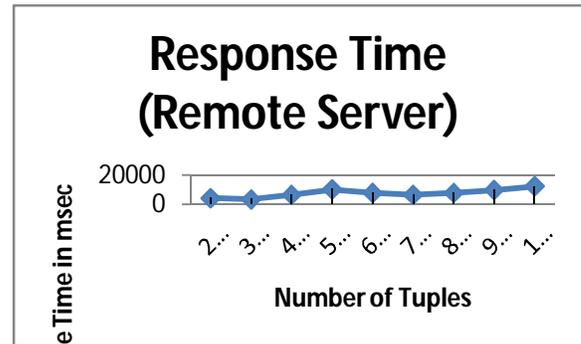


Figure – 6 Response Time for datafederation (web services are located remote server)

The performance of algorithm can be evaluated in an ideal environment where the server load is uniform.

Experimentally it is found that the proposed algorithm is efficient for data federation in service oriented business intelligence.

V. CONCLUSIONS AND FUTURE ENHANCEMENT

The data federation service in service oriented business intelligence was implemented. All services used to perform the operation are coarse grained and belongs to one service provider. Since the process is implemented using services they are independent of underlying software and hardware. The proposed algorithm efficiently integrates the distributed data efficiently. The algorithm is scalable and reliable.

The services like ad-hoc query processing can be designed for effective decision making. A common mapping layer can be proposed for business intelligence operation to have service composed of federated query service in service oriented business intelligence. simple services of different service providers. An efficient service composition mechanism is required.

REFERENCES

- [R1] Thomas Erl, "Service Oriented Architecture: Concept, Technology and Design", Pearson Education, 2009.
- [R2] A. Sheth and J. Larson, "Federated database system for managing distributed, heterogeneous and autonomous databases", ACM Computing Surveys, 22,3, pp 183-236, 1990.

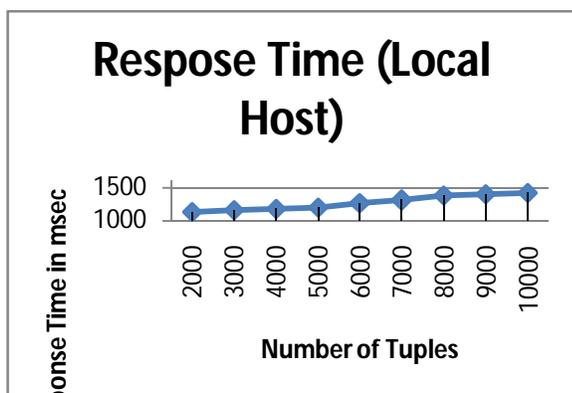


Figure – 5 Response Time for datafederation (web services are located in local host)

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization,

Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHANICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

- [R3] I. Kotsiopoulos, J. Keane, M. Turner, P. J. Layzell and F. Zhu, "IBHIS: Integration Brokers for Heterogeneous Information Sources", Proceedings of COMSAC' 04, IEEE Computer Society Press, 2004
- [R4] F. Zhu, Mark Turner, I. Kotsiopoulos, K. Bennett, M. Russell, D. Budgen, P. Brereton, J. Keane, P. J. Layzell, M. Rigby and J. Xu, "Dynamic Data Integration using Web Services", Proceedings of IEEE International Conference on Web Services (ICWS'04), 2004.
- [R5] D. Calegari, M. Viera and R. Motz, "Design of a Service Oriented Architecture for Federated System", Journal of JCS&T, 5,4 pp 167-172, 2005.
- [R6] Liya wu, Barash G, Bartolini C, "A Service Oriented Architecture for Business Intelligence", IEEE International Conference on Service

Oriented Computing and Applications 2007 SOCA, 19-20 pp 279-285, 2007.

[R7] M. Nageswara Guptha, P. T. Rajan, A. Chitra, "Data Sourcing using Federated Database System in Service Oriented Architecture", CSI Communications, April – 2009, Vol 33, No.1, pp12-18.

[R8] Le Thi Thu Thuy, Doan Dai Duong, "Bottom-Up Strategy for Query Decomposition", Digital Information System, Dec – 2006, pp 215-221.

[W1] www.it.toolbox.com