ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, No 5, May 2013

Design and Implementation of Structured and Unstructured Data Querying System in Heterogeneous Environment

Thu Zar Mon

***Abstract*— In real world applications, data sharing and integration system design is challenging task with the problem. The system currently is only used to integrate and query on structured data (e.g., data of a relational database). To solve this problem, this system proposed an integrated environment that is provided for accessing, querying and sharing structured data (e.g., data of a relational database) and unstructured file-based data (e.g., data stored in a text or binary file). The system is provided via Open Grid Service Architecture Data Access and Integration (OGSA-DAI) services and supported by the Globus Toolkit and that is allowed database operations on both the structured and unstructured data.**

***Index Terms*— OGSA-DAI, structured data, unstructured file-based data, data integration and Globus toolkit**

I. INTRODUCTION

In a variety of scientific fields, grid had been developed with storage, processing, and availability of data. A grid is a collection of distributed computing resources available over a local or wide-area network that appears to an end user or application as one large virtual computing system. In many research purposes, data should not only be stored and rather needs to be readily accessible and integrated. Both of structured and unstructured data are included in resources although the considered shared resources are only files or unstructured data. Data can be stored in various formats, including in a relational database or as a file. A relational database can include a collection of relations, frequently known as tables that can correspond to a logical structure in which data can be stored. Unstructured data consists of any data stored in an unstructured format at an atomic level. Furthermore, unstructured data can be divided into two basic categories: bitmap objects (such as video, image, and audio files) and textual objects (such as spreadsheets, presentations, documents and email). Both of them can be treated as a string of bits. The unstructured data is usually managed by operating system.

The system provided the data abstraction to overcome issues of heterogeneity of data sets including relational tables and text files (comma separated files, CSV) that achieved via Open Grid Services Architecture Data Access and Integration (OGSA-DAI) services and supported by the Globus Toolkit. The Globus provide diverse services related to security and data management based on standard specifications of OGSA. Globus Toolkit is the most popular middleware for file-based data access and sharing. OGSA-DAI is a middleware which has adopted a service oriented architecture (SOA) solution for integration data and grids through the use of web services. When a client wants to make a request to an OGSA-DAI data service, it invokes a web service operation on the data service using a perform document. A perform document is an XML document describing the request that the client wants to be executed defined by linking together a sequence of activities. An activity is an OGSA-DAI construct corresponding to a specific task that should be performed. The output of one activity can be linked to the input of another to perform a number of tasks in sequence. A range of activities is supported by OGSA-DAI, falling into the broad categories of relational activities, XML activities, delivery activities, transformation activities and file activities [10]. To support the concept of data grid, the system uses a series of protocols and services (middleware) as well as a virtual repository. The data sources are hosted in MySQl Server 5.0, Oracle 10g and Microsoft SQL Server 2005 databases. The three main nodes are connected via Fast Ethernet switch (100Mbps). Data-sharing among large natural resource and environment supports the ability to find and acquire the desired data quickly by data-sharing among large natural resource and environment.

The structure of this paper is as follows. Section II introduces related works of the proposed system. Overview and Implementation of the system are discussed in Section III and IV. Section V describes the system operation. This is followed by some current applications. Section VI concludes the paper.

II. RELATED WORKS

A data integration system is an automated method for querying across multiple heterogeneous databases in a uniform way. There are many researches for data integration solution should be transparent to the user. They provide a convenient way of exposing data resources and often used to implement wrappers. In essence, a mediated schema is created to represent a particular application domain and data sources are mapped as views over the mediated schema that approached in research [1], [2] and [3].

The first proposal to use distributed query processing in a grid setting was the Polar* that was accessed using a grid-enabled version of Message Passing Interface (MPI) and

Manuscript received May, 2013.
***Ms. Thu Zar Mon**, Faculty of Information and Communication Technology, University of Technology(Yatanarpon Cyber City), Pyin Oo Lwin, Myanmar, 09-448542823,*

it supported execution of distributed queries. GRelC-Data Gather Service (DGS) [5] was another middleware approach for database access, management and integration.

All the approaches described above only used to integrate and query on structured data. They still have gaps in transparency both of structured and unstructured data. Moreover, a major difficulty in database and files sharing stems from the inherent semantic and heterogeneity environment. Enabling the sharing and querying of distributed data without using a global schema is also one of the challenging tasks towards grid research community.

III. OVERVIEW OF PROPOSED SYSTEM

The overview of proposed system consists of three parts: (1) Data Grid Infrastructure, (2) Run Time System, and (3) User Interface.

1) The Data Grid Infrastructure consists of the grid middleware, which is achieved via Open Grid Services Architecture Data Access and Integration (OGSA - DAI) services and supported by the Globus toolkit deployed in different nodes of the network.
2) Run-Time System consists of web based user interface and document repository that are components deployed in a Web server (Tomcat Server) developed in Java technologies.
3) The User Interface from which users can submit search queries and the results back.

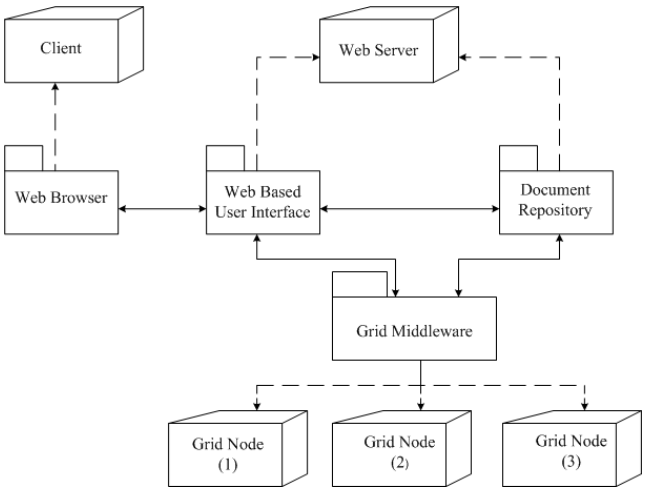

Figure1. Overview of Proposed System

A. Method

A method comprising: receiving structured data and unstructured data; integrating the structured data and the unstructured data, including: generating metadata from the unstructured data; and based on the generated metadata, configuring an abstraction layer to perform a database operation on both the structured data and the unstructured data; and providing an integrated view of the structured data and the unstructured data for display, the integrated view including a user interface for allowing the user to control the database operation of the abstraction layer.

The main objectives of this paper is the ability (1) to find and acquire the desired data quickly by data-sharing among large natural resource and environment (2) to solve lack of collaboration and storage capability problem (3) to deploy the integration framework in a service base grid architecture (4) to develop a decentralized framework for integration of heterogeneous data sources meeting the requirements of scalability, robustness and autonomy.

B. Data Abstraction Layer

Both the structured and unstructured data are allowed by a data abstraction layer. Inside the implementation of the OGSA-DAI server, a node has been selected as coordinator of the abstraction and unstructured data request process on this layer. The layer can provide document catalog that can be associated with schema which can define a set of fields that can track the most common attributes of documents, for e.g.; file name, file type, file size, creation date, modification date and metadata including author, title, subject and copyright information.

Data abstraction layer can include search interface that can provide functions of basic search and can organize generation of an index based on full text of the document being uploaded. The basic search can query both attributes of structured data and content of the unstructured data.

Generating the metadata from the unstructured data can include extracting the metadata from the document, and incorporating user created document attributes into the extracted metadata. Extracting the metadata can include determining a file type of the document. Configuring the abstraction layer can include storing metadata of the document and a reference of the document into one or more document description data fields of the database.

The system can insert the metadata and a document reference that refers to the unstructured data into a table of the database. The document reference can include a file name and a path relative to a root directory of the document repository.

C. Flow Diagram of needed functionality

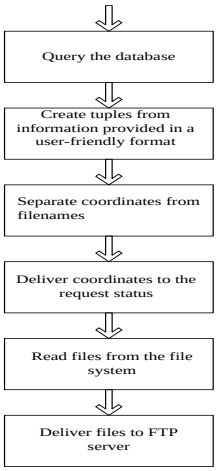

Figure 2. Functionality Flow Diagram

A relational database contains information about files

1) Coordinates Columns e.g, x,y,z
2) Filename Column e.g, filename

To provide data access and integration, the system uses the globus toolkit and OGSA-DAI software as the grid middleware and accesses flat files like CSV, EMBL and SwissPort files. The screenshots of interfaces are at present

stored in a file system and their path is stored in the database. Usually the database and file servers are kept behind firewalls. One possibility to evade the problem of severe security threat is to use OGSA-DAI to deploy the database and to use GridFTP or Grid Reliable File Transfer in globus toolkit to move files around in grid. In both cases the database and file servers just need to open ports for limited number of machines where OGSA-DAI and GridFTP servers are running.

D. Architecture of the Model

The architecture of the model is shown in figure 3. Database are deployed as data service resources, which contain all the information about the database like their physical location and ports, the JDBC drivers that are required to access the database and the user access rights. A data services exposes the data service resource in a web container, which could be a globus container or Apache Tomcat server.

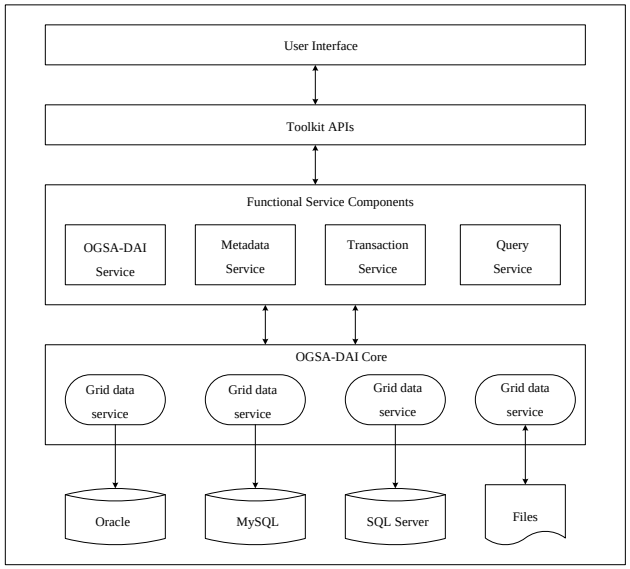

Figure 3. Architecture of the model

The data sources are hosted in MySQl Server 5.0, Oracle 10g and Microsoft SQL Server 2005 databases. The three main nodes are connected via Fast Ethernet switch (100Mbps). Data-sharing among large natural resource and environment supports the ability to find and acquire the desired data quickly by data-sharing among large natural resource and environment. The data service resources perform on behalf of a client. Factory is a service to create a data service instance to access a specific data source.

E. Process of OGSA-DAI

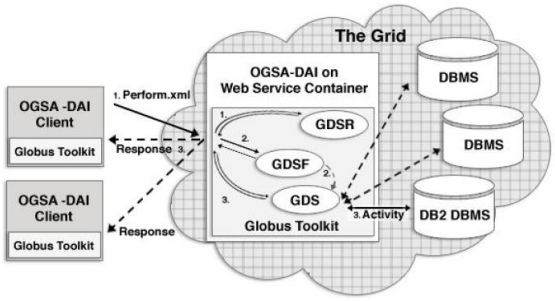

Figure 4. OGSA-DAI process

The process of OGSA-DAI is shown in figure 4.

GDS (Grid Data Service): Provides the access end point for a client and holds the client session with that data resource. A GDS is created by a GDSF.

GDSF (Grid Data Service Factory): Is defined to represent the point of a data resource on a grid. It is through a GDSF that a data resource's capabilities and metadata are exposed.

DAISGRs (Data Access Integration Service Grid Registry): GDSF's may be located on the grid through the use of DAISGR with which GDSFs may register to expose their capabilities and metadata to aid service/data discovery.

The client sends a XML document called perform document, which specifies the activities to be executed on the data service. Unstructured data are stored in a file system and their file location address is stored in a database.

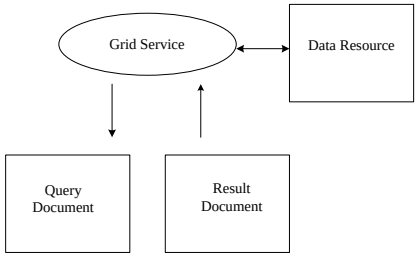

Figure 5. Grid Data Service (GDS) mode of operation

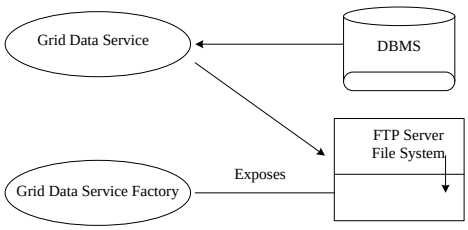

Figure 6. GDS delivery of file system

In figure 5 and 6, simple data access scenario is shown

1) A client contacts a DAISGR first to locate the GDSFs.
2) Accesses suitable GDSFs directly to find out more about their properties and the data resources they represent.
3) Asks GDSF to instantiate a GDS

4) Accesses resource by sending the GDS the GDS-perform document.

F. Metadata and Metadata Storage Model

Data integration is done through Filters' capability metadata. Metadata is stored in local file system as a flat file. OGSA-DAI services provide metadata about the DBMS. Also metadata is provided about the capabilities of that DBMS that are being exposed to the Grid through the service interfaces as well as any inherent capabilities of the service themselves.

Metadata storage model:

- Metadata is kept in Catalog Service (MCS)
- MCS enables attribute-based querying
- Metadata is for the datasets, data can be anything (binary, text ...)
- Data integration is done through XML based activity file mixing activities (in SQL queries) and metadata

For relational Databases, the Database schema may be extracted from the service, which may be helpful for higher level services such as distributed query processing.

OGSA-DAI use any data, and they have predefined Database schema to enable querying and accessing data. Global persistent identifiers for naming files support for metadata to describe the location and ownership of files. Support for descriptive metadata to support discovery through digital library query mechanism.

IV. IMPLEMENTATION OF THE SYSTEM

In proposed system, Globus Toolkit offers a core set of services for file access and management. OGSA-DAI can run alongside Globus Toolkit if data security or fast data transfer (using GridFTP) is needed. All nodes in the system are connected via Fast Ethernet switch (100Mbps). One node of the system is the master node which has to manage the data sources. The document repository included relational database tables defined for unstructured data metadata management. The following tools are needed to implement the services of the system. They are (1) Java Programming Language, (2) Apache Tomcat Web Container, and (3) OGSA-DAI 4.1 GT 4.2.0. This only works with Globus Toolkit 4.2.0.

V. SYSTEM OPERATIONS

The system not only has some static features but also has dynamic snapshots such as the system interfaces in various level, grid nodes and file systems.

Figure 7 shows the operation classes in system. All the operations can be generally divided into 2 groups, the first is information querying and the second is data transferring related. The objectives of dividing operations into classes and groups are for a better and clearer logical structure, and to get maximum code reuse.

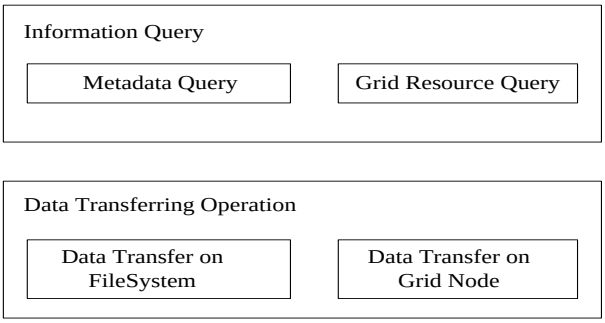

Figure 7. System Operation Schema

A. The Interaction of Executing an SQL Query on Remote Server

To process access data stored in relational database, the interaction of executing an SQL query on a remote server is needed. A typical OGSA-DAI client-service interaction involves a client running an SQL query through a remote OGSA-DAI service that then returns the query response, typically some data, in an XML document. This interaction involves the following six steps: The client sends a request containing the SQL query in SOAP message to an OGSA-DAI service.

1) The server extracts the request from the SOAP message, and the SQL query is executed on the relational database.
2) The query results are returned from the relational database to the OGSA-DAI server as a set of Java ResultSet objects.
3) The server converts the Java ResultSet objects into a format suitable for transmission back to the client, such as WebRowSet.
4) This data is sent back to the client in a SOAP message.
5) The client receives the SOAP message, unpacks the data, and converts it back to a ResultSet object (assuming this is a Java client).

The use of an alternative intermediate delivery format, namely, Comma Separated Values (CSV) was investigated. Although the embedding of metadata has weaker support than the case with the WebRowSet format, this CSV format uses space more efficiently and is easier to parse.

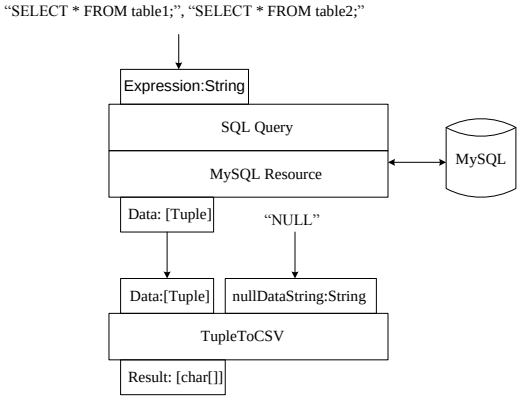

Figure 8. Activity inputs, outputs and resources

In analyzing the changing of data format, we also noticed that the difference of using between WebRowSet and *Comma Separated Values* (CSV) as an intermediate delivery format. Using WebRowSet as an intermediate delivery format added

a significant amount of XML mark-up that increased the amount of data that needed to be transferred between the client and server and the parsing of the messages out of XML could be slow thus possibly incurring an unnecessary overhead. By calculating the space required to represent the same result in each format, the reduction in data size in going from a WebRowSet to a CSV format can be estimated. This can be done by calculating the number of extra characters needed to describe a row of data.

B. *Transferring Binary Data*

To provide access to files stored on a server's file system, a little variation on the previous use case involves using OGSA-DAI. Normally, these could be large binary data files, stored in a file system, for these files stored with the associated metadata separately in a relational database. By using the OGSA-DAI delivery mechanisms, the client queries the databases to locate any files of interest are retrieved. Files are separately retrieved from the SOAP interactions for data transport efficiency. However, it can be more convenient for a client to receive the data back in a SOAP response message rather than using an alternative delivery mechanism.

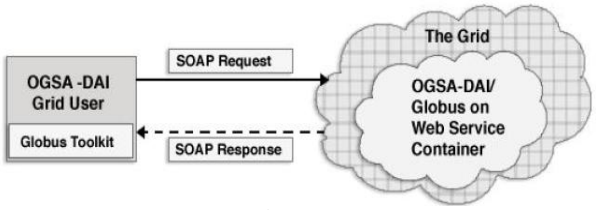

Figure 9. Overview of OGSA-DAI's process

We have covered both of these concerns by using the messages of SOAP with attachments. This approach importantly reduces the time required to process SOAP messages and permits binary data transfer to take place without necessitating Base64 encoding.

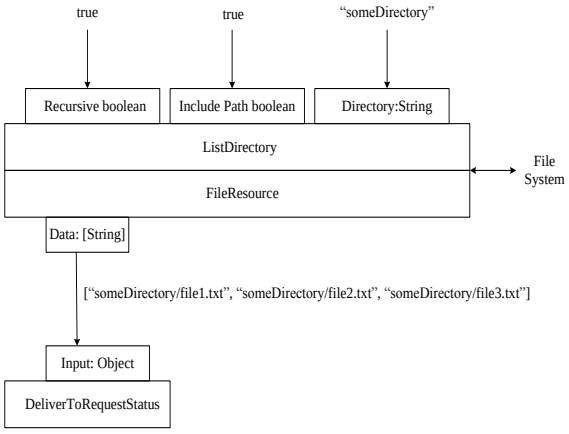

Figure 10. A simple example of list files in a file system

VI. Conclusion

In this paper, we proposed a system for integrating data resources in the heterogeneous environment. The system distinguishes the managed data into two categories, namely structured and unstructured data (flat files). The data grid middleware used for virtualization is separated based on the two categories of data. In our design, we use OGSA-DAI and Globus as the data grid middleware. The combination of the two data grids completely handles all kinds of data types. Hence this system can improve the accessibility, integration and management of the heterogeneous data sources.

Acknowledgment

We foremost thanks go to Professor Dr. Aung Win, the Principal of the Technology of University in Yatanarpon Cyber City, for welcoming our research and giving a hand for us. Next, we would like to thank Professor Dr. Soe Soe Khaing, the Head of the Department of Information and Communication Technology in our university, for giving a chance to fulfill my goal. Moreover, I also wish to thank to the other members of our department for encouraging us and offering guidelines about our research. Finally, our thanks go to our families and friends for all the love and kindness they gave us.

References

[1] Wiederhold & Genesereth, 'The mediator–wrapper architecture', 1997.
[2] Jackson et al, 'The OGSA-DAI (Open Grid Services Architecture Data Access and Integration) middleware', 2007.
[3] M. Nedim Alpdemir, Arijit Mukherjee…, 'OGSA-DQP: A Service-Based Distributed Query Processor for the Grid', 2005.
[4] Aloisio, G., Cafaro, M., Fiore, S., Mirto, M. and Vadacca, S.: '*Grelc Data Gather Service: A Step towards P2P Production Grids'*, SAC, (2007).
[5] *GRelC: 'Grid relational catalog project'*, (2007) http://grelc.unile.it/2007. [4] Jackson et al, 'The OGSA-DAI (Open Grid Services Architecture Data Access and Integration) middleware', 2007.
[6] Dafang Zhuang , Wen Yuan, Jiyuan Liu, Dongsheng Qiu, Tao Ming, 'The Unstructured Data Sharing System for Natural Resources and Environment Science Data of the Chinese Academy of Science' in Data Science Journal, Volume 6, Supplement, 20 October 2007.
[7] Manuel Garcia Ruiz, Alvin Garcia Chaves, 'mantisGRID: A Grid Platform for DICOM Medical Images Management in Colombia and Latin America', 3 February 2010.
[8] Aan Kurniawan, 'Educational Resource Sharing in the Heterogeneous Environments using Data Grid', Faculty of Computer Science, University of Indonesia, 2009.
[9] Pan, H.Research on the Interoperability Architecture of the Digital Library Grid, 2007, in IFIP International Federation for Information Processing, Volume 251, Integration and Innovation Orient to E-Society Volume 1, Wang, W.(Eds), (Boston: Springer), pp.
[10] Steven Lynden a Arijit Mukherjee b Alastair C. Hume c Alvaro A.A. Fernandes a Norman W. "The Design and Implementation of OGSA-DQP: A Service-Based Distributed Query Processor", 2010.
[11] Mike Jackson, Amy Krause. "OGSA-DAI practical – developing workflows and clients", GridKa School 2008.
[12] Susan Malaika, Dirk Hain. "Accessing DB2 Universal Database using the Globus Toolkit and OGSA-DAI", 2003.