



ISSN 2278 – 0211 (Online)

Federated System - A Technique for Efficient Data Mining

Sweta Vinay Kamat

Assistant Professor, Department of Computer Engineering
Shree Rayeshwar Institute of Engineering and IT, Shiroda, Goa, India

Abstract:

Distributed computing plays an important role in the Data Mining process for several reasons. First, Data Mining often requires huge amount of resources in storage space and computation time. To make systems scalable, it is important to develop mechanisms that distribute the work load among several sites in a flexible way. Second, data is often inherently distributed into several databases, making a centralized processing of this data very inefficient and prone to security risks. Federated System for Data Mining explores techniques of how to apply Data mining in a non-centralized way. A federated System may be composed of a heterogeneous collection of databases, in which case it lets applications look at data in a more unified way without having to duplicate it across databases or make multiple queries and manually combine the results. A Federated System will help to overcome differences by allowing the user, who will be provided a common user interface made available at the client terminal to access or modify the data in any of the databases available in the network

Keywords: Federated, database, distributed, computing, data mining, heterogenous, virtual, query

1. Introduction

A federated system [1] is a type of DBMS which transparently integrates multiple autonomous database systems into a single federated database. The constituent databases are interconnected via a computer network [2]. A federated database (or virtual database) is the fully-integrated, logical composite of all constituent databases in a federated database system.

Through abstraction, federated database systems can provide a uniform front-end user interface, enabling users and clients to store and retrieve data in multiple noncontiguous databases with a single query [3] even if the constituent databases are heterogeneous. A federated database system will be able to decompose the query into sub queries for submission to the relevant constituent DBMS's, after which the system will composite the result sets of the sub queries. Because various database management systems employ different query languages, federated database systems can apply wrappers [4] to the sub queries to translate them into the appropriate query languages.

In today's business environment, it is becoming increasingly difficult to access current, accurate, and complete information about the people or entities for which information is stored across an organization. As organizations merge and grow, information about the same entity is dispersed across multiple disparate systems and databases, and there might be several different versions of the information of varying quality. Information becomes fragmented, duplicated, unreliable, and hard to locate. A single source of authoritative, reliable, and sustainable data is needed. As soon as data about the same entities begins to be stored in multiple departments, locations, and applications, the need for this single source becomes apparent.

2. System Architecture

Federated system is a way of integrating data from heterogeneous database sources. In today's rapidly growing business world we see cases of company's merging to expand and flourish. In such cases it is often inevitable that the companies use different data schema's to store their information. Converting data or manually combining data of such huge firms is not a feasible solution.

This system will allow users and applications in such cases to overcome the problem of heterogeneous forms of data storage across a network. With the federated approach the user can access data without having to know where it is actually located on the network and in which format it is

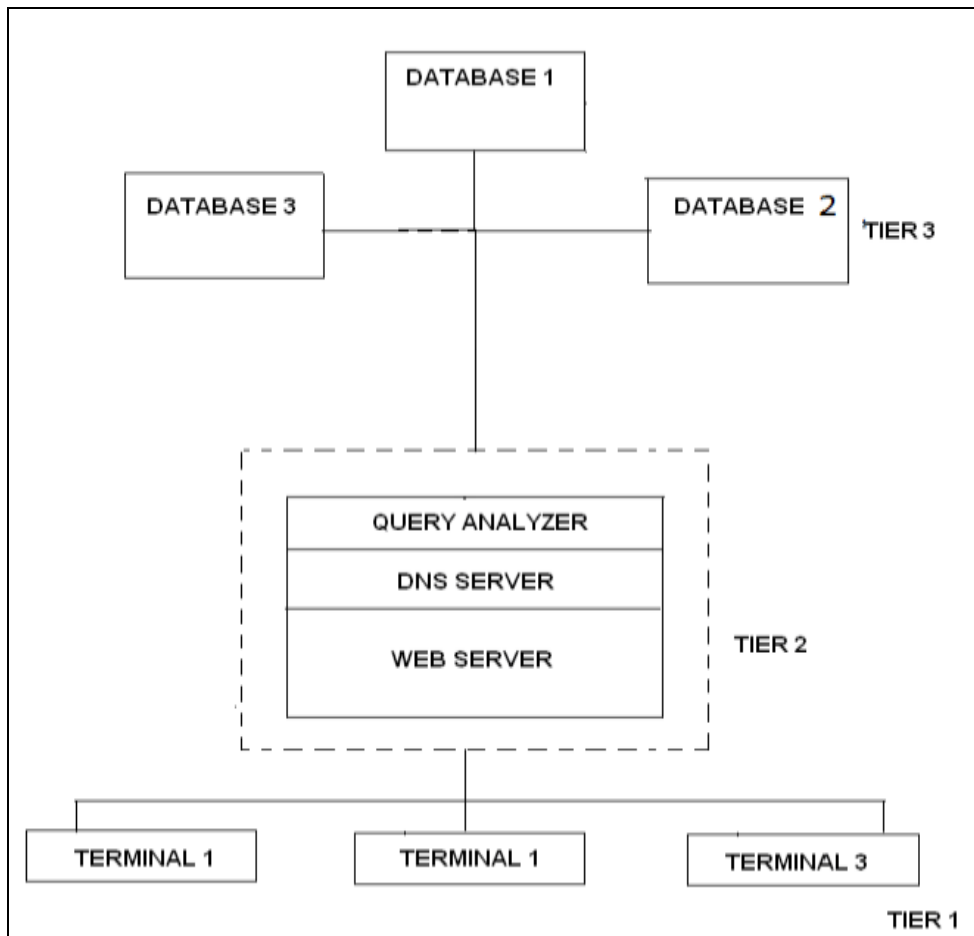


Figure 1: System Architecture

The system operates on 3 Tiers Architecture. Tier 1 provides the user interface; it provides the interface with which a common user or an admin can access or modify the databases available in the network. This can be implemented inside a web browser or as an application [e.g. Java Swing Application]

Tier 2 consists of the Web Server and the DNS Server. The Web server processes the input coming from the user interface, converts it into a generic query and hands it over to the DNS server. Now the DNS Server decides the database that it has to hit for getting the particular data. It also consists of a Query Analyzer. This converts the generic query coming from the web server side to the database specific query and then hits the correct database for data. This application will be installed on the server itself.

Tier 3 consists of the different databases. They may be of different types. This layer contains the data which has to be retrieved to the user terminal.

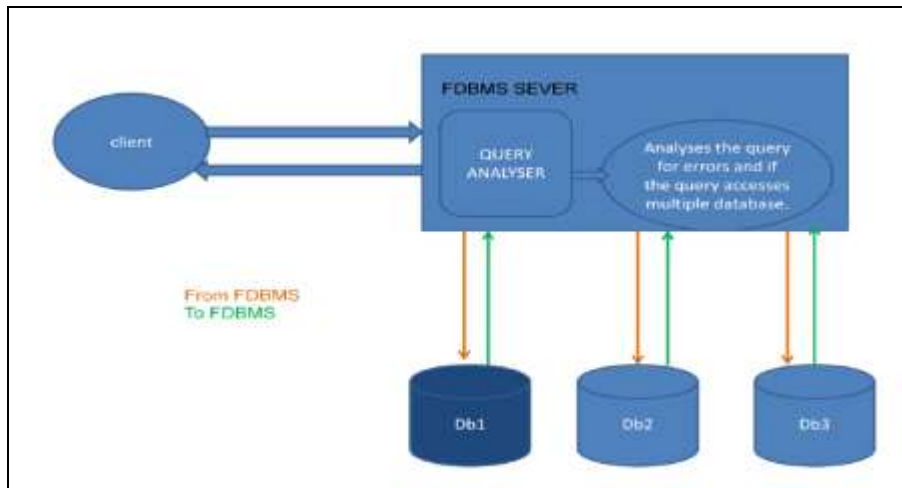


Figure 2: Overall System Design

A Federated is a Database Management System that allows the user to access and modify different databases spread over a network.

1. Through the GUI the user will be able to refer to the available databases and tables specific to those databases.
2. Upon referring, the user can fire a query.
3. The query can be of any database over the network.
4. The query will be analyzed and if necessary split and then translated to the specific database by the query analyzer.
5. The specific database will be hit to execute the query and give back the result to the user.

3. Implementation Results and Testing

The following are the Snapshots of the System upon implementation

3.1. Application User Interface on Start up

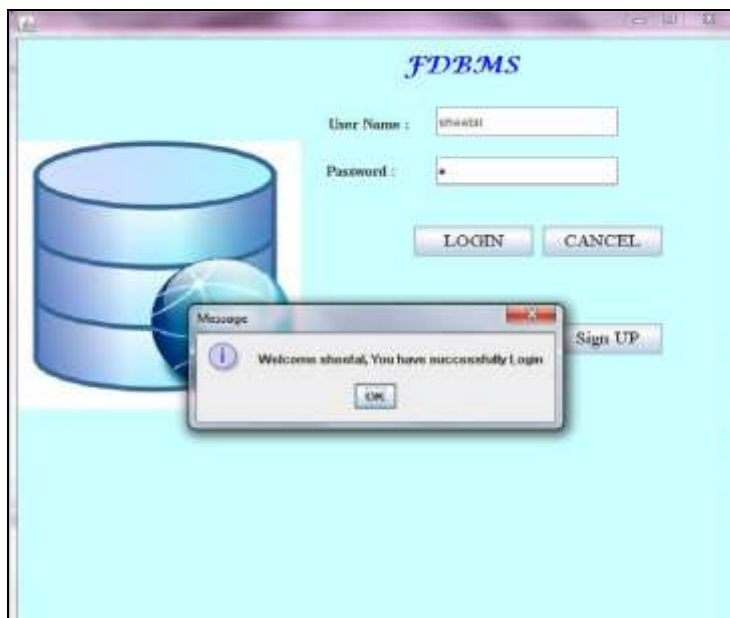


Figure 3 Application Interface

3.2. New User Can Register by Entering the Required Details



Figure 4: New User Registration

3.3. This Page Provides the User with IP address of all PC's in Network and Corresponding Database on Them.

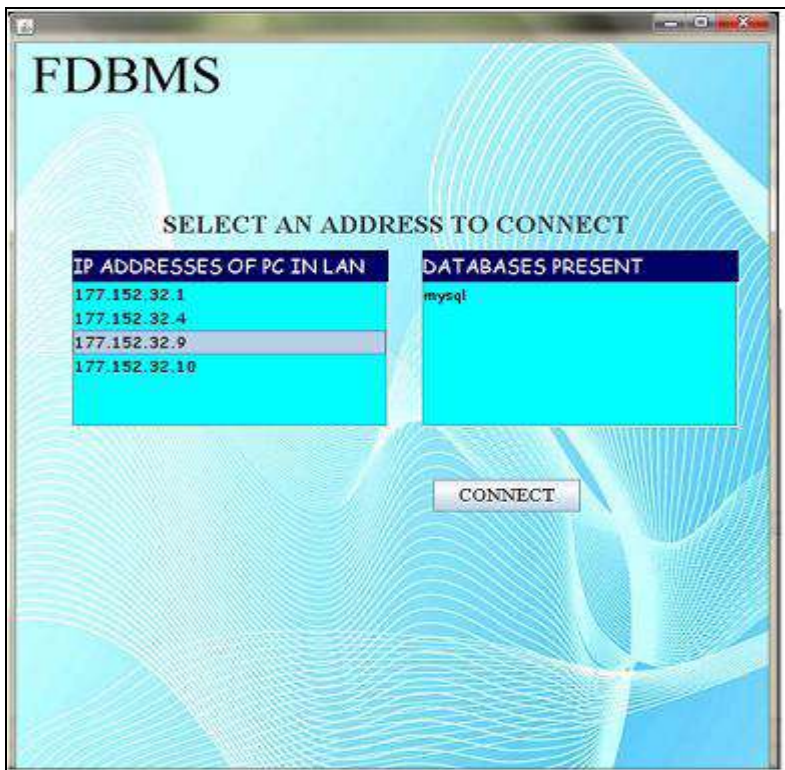


Figure 5: Network connection with distributed databases

3.4. Home Page

According to the role of the user, he will get different home pages to view. Normal user can only view data. Administrator will have all the privileges provided by application.

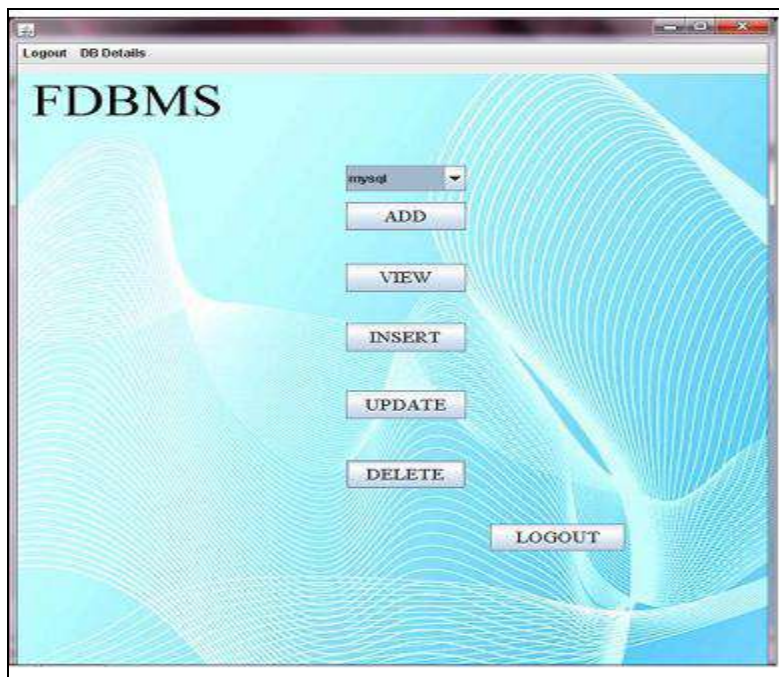


Figure 6: Home page for administrator

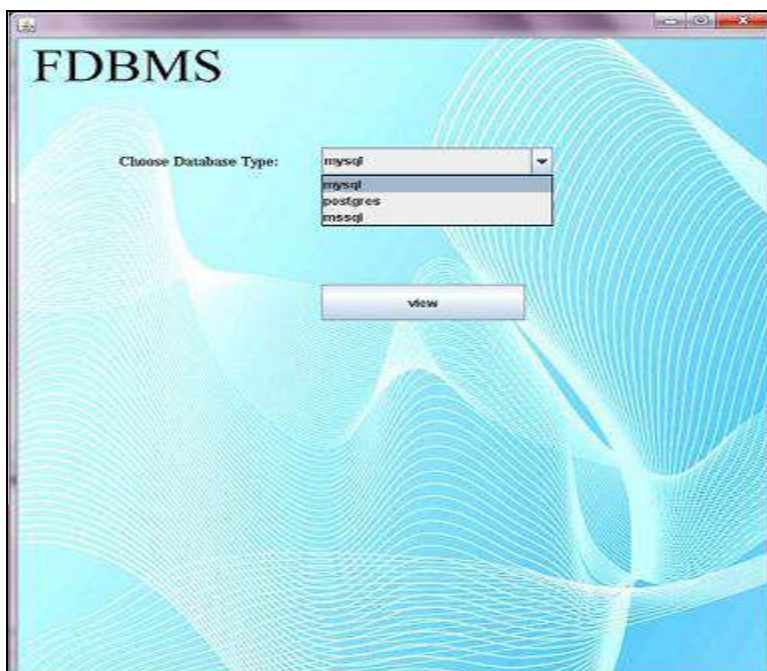


Figure 7: Home page for User

The implementation of the various functions on the home page such as add, delete, update, view and insert can be implemented as in a normal database management system as if it is at a single location.

4. Conclusion

4.1. Conclusion

Federated database management system will help to solve the difficulty of converting or manually integrating data from different database schemas. This approach will help users to retrieve data from any system which is connected in the network and on retrieving, the user will be able to do the necessary changes and save the changes to the desired location or the same location from where the data was retrieved.

The user can transparently access and model data stored in the integrated DBs and/or file systems via the interface provided by the integration layer. The component systems (DBs or file systems) integrated in the FDBS are autonomous and can also be used as stand-alone systems. The integration layer realizes the global data model and distributed query processing and transaction management. It manages the global schema integrating the various local schema and other meta data. It finds a lot of useful applications in the modern business enterprises. In general, a federated system is useful in any situation in which there are multiple sources of data, and a need to combine the information from these various sources.

4.2. Further Enhancements

The user interface can be extended to be used with the help of web browser. That will enable the user to perform data retrieval more conveniently and easily accessible. This will eliminate the need for a custom made client software application.

A database search which will be based on artificial Intelligence concept can be incorporated to the existing system thereby enabling the user to have the provision of giving only the keywords for database query and at the same time obtaining the required result very efficiently.

The multiple database access queries can have more number of features. When implemented, it can be used for the cross-checking and inter-database data retrieval efficiently. In general, a federated system is useful in any situation in which there are multiple sources of data, and a need to combine the information from these various sources of data. Federated technologies provide a common unified interface to diverse data, for example in the case of merging banks and so on.

5. References

- i. federated database management system
<http://www.ibm.com/developerworks/db2/library/techarticle/0203haas/0203haas.html>
http://en.wikipedia.org/wiki/Federated_database_system
- ii. computer network
http://en.wikipedia.org/wiki/Computer_network
<http://www.ibm.com/developerworks/db2/library/techarticle/0203haas/0203haas.html>
- iii. query

- <http://www.informit.com/articles/article.aspx?p%3D29661>
http://wikipedia.org/wiki/Query_language
- iv. wrappers
<http://www.ibm.com/developerworks/db2/library/techarticle/0203haas/0203haas.html>
- v. transparency
<http://www.ibm.com/developerworks/db2/library/techarticle/0203haas/0203haas.html>
- vi. heterogeneity
<http://www.ibm.com/developerworks/db2/library/techarticle/0203haas/0203haas.html>
- vii. autonomy
<http://www.ibm.com/developerworks/db2/library/techarticle/0203haas/0203haas.html>
- viii. SAX
http://en.wikipedia.org/wiki/Simple_API_for_XML
- ix. XML
http://www.w3schools.com/xml/xml_what_is.asp
- x. DOM
<http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001>
- xi. client server architecture
<http://www.wisegeek.com/what-is-client-server-architecture.html>
- xii. RMI(Remote Method Invocation)
http://en.wikipedia.org/wiki/Java_remote_method_invokation
- xiii. Domain Name System
http://en.wikipedia.org/wiki/Domain_Name_System
- xiv. Web server
http://what_is.techtarget.com/definition/0,,sid9_gci213606,00.html
http://en.wikipedia.org/wiki/Web_server
- xv. Apache
<http://www.modulehosting.com/apache.html>
- xvi. Internet Information Server
http://en.wikipedia.org/wiki/Internet_Information_Services
- xvii. data Mining
[http://www-ai.cs.uni-dortmund.de/auto?self=\\$ejr31cyc](http://www-ai.cs.uni-dortmund.de/auto?self=$ejr31cyc)
- xviii. XML
http://www.w3schools.com/xml/xml_what_is.asp
- xix. DOM
<http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001>
- xx. client server architecture
<http://www.wisegeek.com/what-is-client-server-architecture.html>
- xxi. RMI(Remote Method Invocation)
http://en.wikipedia.org/wiki/Java_remote_method_invokation
- xxii. Domain Name System
http://en.wikipedia.org/wiki/Domain_Name_System
- xxiii. Web server
http://what_is.techtarget.com/definition/0,,sid9_gci213606,00.html
http://en.wikipedia.org/wiki/Web_server
- xxiv. Apache
<http://www.modulehosting.com/apache.html>
- xxv. Internet Information Server
http://en.wikipedia.org/wiki/Internet_Information_Services