Architecture for Accessing Heterogeneous Databases

Mohd Kamir Yusof, Ahmad Faisal Amri Abidin, Mohd Nordin Abdul Rahman
Faculty of Informatics, Universiti Sultan Zainal Abidin, Terengganu, Malaysia
Email: mohdkamir@unisza.edu.my

Abstract—This paper presents the architecture for accessing heterogeneous databases. Two major processes in this architecture are extracting SQL statement and ontology. The algorithms for extracting SQL statement were created and tested in order to improve time performance during searching and retrieving process. Ontology approach was implemented and combined with these algorithms. In ontology approach, web semantic was implemented in order to retrieve only relevant data from database. A prototype based on this architecture was developed using JAVA technology. JAVA technology was chosen because this technology have Jena library. This library is provide API and support SPARQL. Several experiments have been executed and tested. The result indicates this architecture able to improve web query processing in term of time. The result also indicates this architecture able to retrieve and displayed more relevant data to web users.

Index Terms—Heterogeneous Database, Data warehouse, Ontology, Semantic Web, Web query processing.

1. Introduction

Web query is an important element to be considered in architecture application development especially for web based application. The purpose of web based application is to provide the information to web users. In Malaysia, most of people using web based application such as Malaysia Government Portal (http://www.malaysia.gov.my) to access information about education, carrier, tourism, etc. The web users can surfing internet using laptop, notebook, PDA, etc anywhere and anytime to access the information. These information can help web users to make any decision in research, education, etc. Most of web applications stored data into a single database. A single database stored lot of data. Increasing number of data will reduce the space size of storage database. In this case, performance of web query processing become slow to search and retrieve lot of data from a single database. In web based applications, three main component involved; interaction among web user, web application and web server. User interface is provided to web user to enter information (a query) such as education, tourism, etc. The user interface also will display the information/results to web user. Web application server will get a query from web user. This query will manipulate before searching and retrieving process. Then, the web application server will communicate with database server for searching and retrieving process. After data is found, the information/results will display to web user through web user interface. Figure 1 shows the interaction among web user, web application server and database server.

Due to increasing number of data in a single database, a good approach to overcome this issue is heterogeneous databases approach. This approach very suitable implement in web based application especially involved with lot of data. Heterogeneous database means, data is stored in data sources (databases – different type of DBMS) and located at different places. For instance, in heterogeneous databases approach, different data sources are contains different data such as data source1 about tourism, data sources2 about education, etc). In example above, web server will connect to all data sources in order web user can access all data in these data sources. However, issue about web query processing in heterogeneous databases still exits caused by difficult to determine what the really users wants [7]. Several research effort has been attempt to solve this problem such as query refinement, etc. These effort is to help the user identify better terms for a query [1][4]. Nevertheless, there is no techniques that enables to identify when the results of the query are relevant [7]. In this research, architecture for accessing heterogeneous databases was designed and implemented in web based application. Experiments in this research focus on two
parameters; 1) number of relevance data and 2) response time (in seconds) to access the data.

2. Literature Review

Most of government sector in Malaysia, their web based applications work with heterogeneous database. Hospital is one example using heterogeneous databases approach. Heterogeneous databases system (HDDBS) might provide uniform access to electronic patient records in hospital computing environment that uses a MUMPS hierarchical database for storing patient demographic data and a Sybase relational database for storing patient laboratory results [9]. However, the core of the heterogeneous problem in hospital environment is that independently developed and maintained databases are heterogeneous with respect to their query models [9]. Heterogeneous databases approach also has been implemented in geospatial data. Implementation of heterogeneous databases in geospatial data involved three modules; a vector/vector integration module, a raster/vector integration module, and a databases module. Databases architecture was designed to preprocess inputs and to store and export results of the vector/vector and the raster/vector integration steps [8]. The unprecedented increase in the availability of information due to the success of the World Wide Web has generated an urgent need for new and robust methods that simplify the querying and integration of data [2]. A lot of research in the past focuses on developing methodologies for querying heterogeneous data sources. Integration data from existing databases in a distributed environment will give the impact of operations on the databases [14]. One of the approaches in database integration is unified global integration [2]. The purpose of this approach is to facilitate efficient global processing. However, this approach becomes hard to manage as the number and types of data sources increase. Another approach in database integration is system based on mediators and wrapper [2]. This approach is sophisticated that abstract the data sources from the users. The wrapper-mediator is remarkable scalable, and allows the integration of an increasing number of data sources. This research focuses on designing architecture for accessing heterogeneous databases in order to improve web query processing.

3. Data Warehouse

Data warehouse is a data repository that provide variety of data to web users. Data from different sources are located at single place. This single place is called “data repository or data warehouse”. Ideally, web users should be able to access data from the data warehouse without knowing either where data resides or the form in which it is stored [12]. In this research, virtual view approach and datamart approach in data warehouse have been applied. Through this combination approaches, data schema from all data sources will collect and store into data warehouse. These data schema will update automatically once any changing occurs.

4. Semantic Web

The purpose of web based application is to provide information to web users. Most of web based application is designed for humans to read, not for computer programs to manipulate data into meaningful information. Figure 4 shows four major layers involved in the semantics web.

XML as data structure language is allows users to add arbitrary structure to their documents. Programmer must know what the document creator means by each structure as XML alone does not capture information about the structures meaning [13]. RDF represent meaning, as users use metadata to describe Web resources and improve in RDF as encoded triples which states that particular resources (subjects) have properties (predicates) with certain value [5]. The second top layers
Ontology provides a shared and common understanding of a domain that can be communicated between people and heterogeneous and distributed application systems.

5. Ontology in Web Query

In this research, ontology was used in web query processing. The ontology approach was used to retrieve data from heterogeneous databases.

6. Architecture for Heterogeneous Database Access

In this section described about architecture for accessing heterogeneous databases. Two major components in this architecture. First is extracting SQL statement from web users and second is ontology. Four processes involved in extracting SQL statement which are assign the initial query, exploit the initial query, assign any possible query and refinement query. In ontology component, two processes which are semantic mapping and extracting wrapper ontology. This architecture focuses on two number of words for one sentences (such as Data Mining, Data Warehouses, etc). Figure 4 shows the processes flows in architecture for heterogeneous databases access.

6.1 Extraction SQL

Four processes involved which are assign the initial query, exploit the initial query, assign a possible query and possible refinement query.

a) Initial Query
Query from web users through application or system is initial query. For example, web user sends a query about “Political Issues”. System automatically assign this query as a initial query.

\[ A \rightarrow IQ \]  \hspace{1cm} (1)

Example 1
IQ = Political Issues
A = IQ (Political Issues)

b) Exploit Query
This process will exploit a query from web user. Once web user sends a query about “Political Issues in Malaysia”, this section will exploit into four words; Political, Issues, In, and Malaysia.

\[ n = \text{Number of words in } A \]
\[ M \subseteq \{n_1, n_2, n_3, ..., n_i\} \]

Example 1
A = Political Issues in Malaysia
A1 = {Political Issues
A2 = {in Malaysia
n = 4
N, number of words in A is 4, where data set in A exploit by space.

Example 2
A = Data Mining
A1 = {Data Mining
n = 2
N, number of words in A is 2, where data set in A exploit by space.

c) Assign a Possible Query
In this process, the query will refinement based exploitation query process.

\[ Qn = \text{Number of possible query} \]
\[ S \subseteq \{Q_1, Q_2, Q_3, ..., Q_n\} \]

Example 1
A = Data Mining
A1 = {Data Mining
Assign any possible query;
\[ \alpha_1 = \{\text{Data Mining}\} \]
\[ \alpha_2 = \{\text{Mining Data}\} \]
Number of possible query is 2, where \( S \in \{\text{Data Mining, Mining Data}\}.

d) Refinement Query
Finally, this query will match keywords in data warehouse.

\[
\text{Start}
\text{Set initial variable is } Q
\text{ } Q \Rightarrow \{Q_1, Q_2, \ldots, Q_n\}
\text{Loop}
\text{ } K = \{K_1, K_2, \ldots, K_n\}
\text{If } (Q_i = K_i)
\text{ } \{
\text{Goto searching}(Q_i, K_i);
\}
\text{While } Q \text{ is null}
\text{ } K \notin M
\text{Loop}
\text{End}
\]

Fig 5: Matching algorithm

Figure 5 shows the how the algorithm work with get the initial query until match to suitable keywords in data schema.

Example 1
\( \beta = \{K_1, K_2, K_3, \ldots, K_n\} \), where \( K_1, K_2, \ldots, K_n \) represents as a keywords. These keywords will store in temporary files at server side. \( Q \)

Step 1
Match \( Q_i \) with any keywords, \( K_i \) in temporary files.

Step 2
Hold data sources, DS, location if found, otherwise keep new keywords, \( Q_i \) in temporary files.

Step 3
Repeat step 1 and step 2 until data set \( Q \) equal to null.

6.2 Ontologies
Ontology approach was implemented in architecture above. Thr purpose of this implementation is to ensure web users will get a relevant information. Two processes involved in ontology components which are semantic mapping and extracting wrapper ontology.

6.2.1 Build Ontology – Based Query (Extracting Wrapper Ontology)
First step in implementation of ontology is to create a model for query relational database on ontology. Two phases were considered in this model which are offline ontology extraction and online query issuing. In offline ontology extraction, system extracts the explicit classes and relations from the relational schema. In online query, web users can issue semantic query to the system. Figure 6 shows the query relational database on ontology model [10].

Table 1 below shows the primary key, foreign key and their relationship. The purpose of this table is to show relation among tables.

<table>
<thead>
<tr>
<th>Relation</th>
<th>Primary Key</th>
<th>Foreign Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book(BookID, Name, Price, SupplierID, ISSN, CategoryID)</td>
<td>BookID</td>
<td>SupplierID, CategoryID</td>
</tr>
<tr>
<td>Supplier(SupplierID, BookID, SupplierName, Address, Tel, Email, CSupplierID)</td>
<td>SupplierID</td>
<td></td>
</tr>
<tr>
<td>BookType(BookTypeID, Name)</td>
<td>BookTypeID</td>
<td></td>
</tr>
<tr>
<td>CategorySupplier(CSupplierID, Name)</td>
<td>CSupplierID</td>
<td></td>
</tr>
</tbody>
</table>

Rule 1: If primary key of more than one relation is the same is same, merged in one ontological class and their attribute should be merged.

ex: Book rdf:type
    Book(BookID, Name, Price, ISSN, SupplierID, CategoryID)

Rule 2: If the primary of one relation is unique for that relation, and not contain the primary key in another relation, then that relation will be considered as one ontological class.

Ex: BookType rdf:type
    BookType(BookTypeID, Name)
Ex: CategorySupplier rdf:type
    CSupplier
This query is to find the book information (name, price, ISSN) where book name = "Data Mining".

**SQL Syntax**

```
SELECT Book.name, Book.price, Book.ISSN, Supplier.Name FROM Book, Supplier
WHERE Book.SupplierID = Supplier.SupplierID and Book.Name="Data Mining";
```

After executing this SQL statement, system will retrieve all related information about data mining from databases and display to web users.

### 7. Experiment and Analysis

In this section, we describe about implementation, sample query, analysis of proposed architecture, and validation component. The purpose of this section is to produce a prototype based on system architecture was designed in section 6.

#### 7.1 Implementation

The methodology has been implemented in a prototype using JAVA, HTML and four different Database Management System (DBMS). JAVA is programming language to develop a web application system based on architecture in figure 5. JAVA programming was choose because it is a powerful language and make a web application is portable and easily to accessible through World Wide Web. In JAVA, many RDF libraries are provided. The most complete library is Jena. Jena was develop to provide API that was designed specifically for the JAVA programming especially for web application development. Jena also provides ontology API and rules engine for basic inference RDF schemas. It is also support SPARQL by calling ARQ module [10].

#### 7.2 Sample Query

This section illustrates how our system works using a sample query. Suppose that a web user is looking to buy books. Assume that a web user is looking to buy “data mining” book.

SPARQL-Syntax

```
Select ? name ? price ? ISSN ? Supplier Name ?
WHERE {
  ?s ex: BookID ? x.
  ?x ex: name ? bname.
  ?x ex: ISSN ? bISSN.
  ?x ex: Supplier Name ? Sp.
}
```

Fig 8: Search Form
The user interface is allows web users to enter the information (Fig 8). This information is called a keyword. The user requests information through queries submitted to the system via an HTML form. The first process based on architecture in figure 5 is extracting a SQL statement. A SQL statement must through 4 sub processes in extracting SQL statement process. If the keyword is already exists in data schema, this information will submit to next process. The next process is ontology. Two sub processes in the ontology process are semantic mapping and extracting wrapper ontology. In these processes, searching and retrieving process from data sources occurred. The purpose of these processes is to search and retrieve only relevant information to the web users.

Fig 9: Result

In figure 9, all relevant information will display to web user. These all relevant information is based on a keyword entered by the web user.

Table 3: Results and Comparison

<table>
<thead>
<tr>
<th>Sample Query</th>
<th>Number of Relevance Data</th>
<th>Response Time (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Heterogeneous Database Architecture</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Mining</td>
<td>20</td>
<td>0.20</td>
</tr>
<tr>
<td>Data Warehouse</td>
<td>40</td>
<td>0.45</td>
</tr>
<tr>
<td>Information Retrieval</td>
<td>15</td>
<td>0.15</td>
</tr>
<tr>
<td>Database Integration</td>
<td>8</td>
<td>0.10</td>
</tr>
<tr>
<td>Education</td>
<td>9</td>
<td>0.14</td>
</tr>
<tr>
<td><strong>Proposed Heterogeneous Database Architecture</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Mining</td>
<td>25</td>
<td>0.18</td>
</tr>
<tr>
<td>Data Warehouse</td>
<td>31</td>
<td>0.33</td>
</tr>
<tr>
<td>Information Retrieval</td>
<td>15</td>
<td>0.12</td>
</tr>
<tr>
<td>Database Integration</td>
<td>8</td>
<td>0.09</td>
</tr>
<tr>
<td>Education</td>
<td>13</td>
<td>0.10</td>
</tr>
<tr>
<td>University</td>
<td>15</td>
<td>0.11</td>
</tr>
</tbody>
</table>

7.3 Analysis of Proposed Architecture

Table 3 shows the comparison between proposed heterogeneous database architecture and heterogeneous database architecture. This comparison based on number of relevance data and response time (in seconds). Several sample queries have been tested in experiments. Based on table above, the result indicates number of relevance data for “Data Mining” is 20 in both of these architectures. But, response time shows proposed heterogeneous database architecture was decrease about 20% from 0.20 seconds to 0.18 seconds. In “Data Warehouse”, number of relevance data for proposed heterogeneous database architecture was deducted from 45 to 34 compared to heterogeneous database architecture. Meanwhile, the response time for proposed heterogeneous database architecture was decreased about 30% from 0.33 seconds to 0.18 seconds. In others queries sample in table 3, such as Information Retrieval, Education and University, the results indicates number of relevance data for proposed heterogeneous database architecture was decreased about 5% to 10% average compared to heterogeneous database architecture. Meanwhile, in term of response time the proposed heterogeneous database architecture was decreased about 10% to 15% compared to heterogeneous database architecture. However, the result indicates number of relevance data for “Database Integration” is same for both of these architectures, but in term of response time the proposed heterogeneous database architecture was decreased about 10% from 0.10 seconds to 0.09 seconds. In conclusion, the result in table 3 indicates implementation of algorithm in extracting SQL statement and ontology approach was improved the web application performance in term of number of relevance data and response time.

8. Conclusion and Future Work

In conclusion, this paper was presented implementation of proposed architecture for accessing heterogeneous databases. This research focuses on improving web query processing. Based on experiments and results above, this architecture is able to improve web query processing in heterogeneous databases access in term of relevance data and response time. The main advantage in this research, the proposed architecture is very efficient and suitable for two numbers of words (such as Data Mining, Information Retrieval, etc). In future work, this proposed architecture will enhance in order it can support more than two numbers of words (such as Introduction to Data Mining, Advanced Research in Computer Science, etc).

References

Mohd Kamir Yusof obtained her Master of Computer Science from Faculty of Computer Science and Information System, Universiti Teknologi Malaysia in 2008. Currently, he is a Lecturer at Department of Computer Science, Faculty of Informatics, Universiti Sultan Zainal Abidin (UniSZA), Kuala Terengganu, Terengganu, Malaysia. His main research areas include information retrieval, database integration and web semantics.

Ahmad Faisal Amri Abidin obtained his Master of Computer Science from Faculty of Computer Science and Information Technology, Universiti Putra Malaysia in 2008. Currently, he is a Lecturer at Department of Computer Sciences, Faculty of Informatics, Universiti Sultan Zainal Abidin. His main research areas include computer security, mobile computing and computer networks.

Mohd Nordin Abdul Rahman obtained her PhD in Computer Science from Universiti Malaysia Terengganu in 2008. Currently, he is a Lecturer at Department of Computer Science, Faculty of Informatics, Universiti Sultan Zainal Abidin, and Terengganu, Malaysia. His main research areas include web services, cloud computing and software engineering.