

Designing an Architecture for Improving Web Query Processing in Heterogeneous Databases Access

Mohd Kamir Yusof^{1st}, Ahmad Faisal Amri Abidin^{2nd}, Surayati Usop^{3rd}, Sufian Mat Deris^{4th}

Universiti Sultan Zainal Abidin

Terengganu, Malaysia

+609 6653117

mohdkamir@unisza.edu.my^{1st}

ABSTRACT

World Wide Web is important tool to web users. This tool enables the web users gain information about education, entertainment, etc. In this research, we investigate the issue for data access to heterogeneous data sources. Web query processing is issue for access the heterogeneous data sources. In this research, new architecture for improving web query processing in heterogeneous databases was presented. In this architecture components such initial query, exploit query, refinement query and match query is implemented in order to improve web query processing. Meanwhile, new algorithm designed to support this architecture. Based on approaches above, we also present architecture and prototype of a system for heterogeneous database access.

Keywords

Data warehouse, heterogeneous database, linguistic, semantic, web query processing.

1. INTRODUCTION

Web query is important element to be considered in 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 get information about education, carrier, tourism, etc. Information is important to web users. Web users can surfing internet using laptop, notebook, PDA, etc anywhere and anytime to get the information. These information can help web users to make any decision, research, etc. Most of web applications stored data into a single database. This database normally stored lot of data and provide lot of information to web users. In web applications, normally involved two main component; interaction between web application and web server. Web application allows web users to search any information and also display the information (result) to web users. Meanwhile, in web server, all data stored into a single database. Figure 1 shows the interaction between web application and web server.

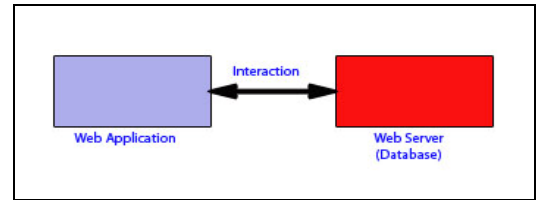


Figure 1. Interaction between Web Application and Web Server

The issue for a single database are capacity and web query processing. The big issue in web based application is web query processing. This issue occur once a single database stored lot of information (combining data such as tourism, health, education, etc). Theoretically, web query processing will become slow if a size or capacity of single database is too big. Once web query processing become slow, web users will feel give up to search any information from this web based application. Heterogeneous databases one of the best solution for web query processing issue. Heterogenous databases will separate data sources with different database (such as data source1 about tourism, data sources about education, etc). However, issue about web query processing in heterogenous databases still exists caused by difficult to determine what the really users wants [1]. Several research effort have been attempt to solve this problem such as query refinement, etc. These efforts are to help the user identify better terms for a query [2][3]. Nevertheless, there is no techniques that enables to identify when the results of the query are relevant [1]. That way, a new architecture with good methodology is needed to improve web processing in heterogenous databases access.

2. LITERATURE REVIEW

Most of application or web based applications work with heterogeneous databases. Hospital is one example of heterogeneous databases approach. Heterogeneous databases system (HDBS) 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 [10]. 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 [10]. 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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WIMS'11, May 25-27, 2011 Sogndal, Norway

Copyright © 2011 ACM 978-1-4503-0148-0/11/05... \$10.00.

steps [11]. 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 [5]. A lot of research in the past focuses on developing methodologies for querying heterogeneous data sources. Integration of data from existing databases in a distributed environment will give the impact of operations on the databases [6]. One of the approaches in database integration is unified global integration [5]. 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 [5]. This approach extracts 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 a architecture for improving web query processing for heterogeneous databases. In this research, data warehouse, semantic, database integration, linguistic and clustering was studied to produce architecture and come out with a prototype for experiment and testing purpose.

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 [4]. 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.

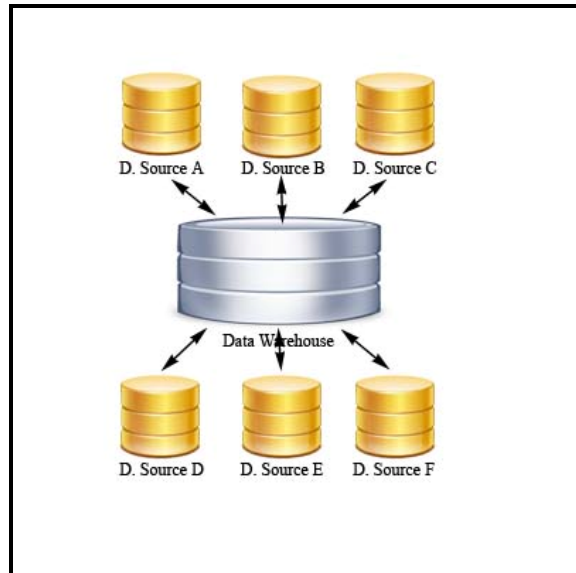


Figure 2: Warehouse Architecture

In figure 2, data schema from six different data sources will extracted and store into data warehouse. Physical data still belonging data sources. Mapping technique will aplpy to map data schema to pyhsical data.

4. ONTOLOGY IN WEB QUERY

A concept of the ontology may represent either semantic or linguistic information. The rationale is that ontology should represent the synonyms of each concept, when a concept is a compaund name composed for two or more nouns. For instances, web users request about “*WorkHard*”. A semantic concept in ontology called *WorkHard* and the ontology should also contain linguistic information which indicate that its name a noun comprised of two words: *Work* and *Hard*, and is the denotation of the word *Hard*. The ontologies in web query processing can help user get more relevant infromation. Most of ontologies used to support web queries neither represent the instances of the conceptualization they are modeling, nor the linguistic infromation about their concepts. However, the problem about web users statisfaction still exist, because to improve correctly the user request, linguistic external repositories is needed to identify which concepts in the ontology are related with the terms provided by the user.

5. METHODOLOGY

The methodology uses semantic, linguistic, clustering and data warehouse approach. This methodology focuses on two number of words for one sentences (such as Data Mining, Data Warehouses, etc). Two major parts in this methodology are client side and server side. Server side has three components: initial query, exploit query and possible refinement query. This methodology specific to length of the query equal to two such as Data Mining, DataWarehouse, WorkHard, Policitical Issues, etc.

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.

$$A \rightarrow IQ \quad (1)$$

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 = Number of words in A

$M \in \{n_1, n_2, n_3, \dots, n_i\}$

$M \cup n$

Example 1

A = Political Issues in Malaysia

$A_1 = \{\text{Political Issues}\}$

$A_2 = \{\text{in Malaysia}\}$

$n = 4$

N , number of words in A is 4, where data set in A exploit by space.

Example 2

A = Data Mining
 $A_1 = \{\text{Data Mining}\}$
 $n = 2$
 N, number of words in A is 2, where data set in A exploit by space.

c) Possible Refinement Query

In this process, the query will refinement based exploitation query process.

Q_n = Number of possible query
 $S \in \{Q_1, Q_2, Q_3, \dots, Q_n\}$

Example 1

A = Data Mining
 $A_1 = \{\text{Data Mining}\}$
 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) Matching Query

Finally, this query will match keywords in data warehouse.

```

Start
Set initial variable is Q
 $Q \rightarrow \{Q_1, Q_2, \dots, Q_n\}$ 
Loop
     $K = \{K_1, K_2, \dots, K_n\}$ 
    If ( $Q_i == K_i$ )
    {
        Goto searching( $Q_i, K_i$ );
    }
While Q is null
 $K \in M$ 
Loop
End
  
```

Figure 3: Algorithm in Client Side

Figure 3 shows the how the algorithm work with get the initial query until match to suitable keywords and data schema in data warehouse. Meanwhile, three processes in server side; searching, mapping and retrieving.

Example 1

$\beta = \{K_1, K_2, K_3, \dots, K_n\}$, where $K_1, K_2, \dots, K_n$ 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_i 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.

a) Mapping

Based on keywords was found, system maps to relevant data schema.

```

Start
Mapping(S)
{
     $M \rightarrow \{\text{data sources, } Y_i, \dots, Y_n\}$ 
}
End
  
```

Figure 4. Mapping Algorithm

Based on figure 4, system map directly to data sources. After that, system will search and retrieve data from this data sources. Number of data sources depends on the keywords found in temporary files.

b) Searching

After get the possible query, system search any relevant keywords in data warehouse. For instance, based three possible query, system found eight keywords in data warehouse. After this result sends to mapping process.

```

Start
Seaching( $X_i, Y_i$ )
{
     $M \rightarrow \{X_i, Y_i\}$ 
    Mapping(M)
}
End
  
```

Figure 5. Searching Algorithm

Based on figure 5, system will start searching process after data sources location was identified.

c) Retrieving

In this process, system will retrieve data from certain data sources. Then, the relevant information will display to web user.

```

Start
 $x \rightarrow \text{Exploit Query}$ 
for ( $i=0; i < \text{count}(x); i++$ )
{
    for ( $j=0; j < \text{count}(y)$ )
    {
        if found  $x[i] == \text{keywords}[i]$ 
             $\text{keywords}[i] \rightarrow \text{data sources}$ 
    }
     $M \rightarrow \text{Result}$ 
}
Display result
End
  
```

Figure 6. Algorithm in Server Side

Example 1

Q → Initial Query
X → Exploit Query

Q = {Data Mining}
X → Exploit {Data Mining}
X₁ → Data
X₂ → Mining

Assign possible queries
Y₁ → X₁.X₂ → {Data Mining}
Y₂ → X₂.X₁ → {Mining Data}

⋮
⋮
⋮
Y_n → X_n.X_{n-1}

Matching with keywords (K_i) in temporary files

If Y_n = K_i

keywords[i] → data sources

Else

Y_n → temporary files // store in temporary files

Assign data sources

keywords[i] → data sources

Finally, search data based keywords and map directly to data sources. After searching and retrieving process, display the result to web users, M → Result. In retrieving process, sentence-query similarity was implemented. This formula was implemented:-

$$C(s, q) = \left\{ \left(\frac{|s \cap q|}{|q|} \right)^k \mid |s \cap q| \geq \alpha |q| \right\} \quad (2)$$

Here $|s \cap q|$ is the number of index terms in common between s and q. Number of index terms in q by |q|. The exponent k is a parameter that is used for controlling importance of the number of

terms in common. As k increases, big $|s \cap q|$ becomes more important. A threshold function t is used to ignore sentence-query similarity if $|s \cap q|$ is too small compared with |q|. The current value of t are set as follows: t(1) = 2, t(2) = 1, t(3) = 2, t(4) = 2, t(5) = 2 and t(i) = 3 for i ≥ 6. For all sentences s_i, 1 ≤ i ≤ l, in document d having l sentences the C values are summed to get the sentence-query similarity sim₁(d, q) between d and q;

$$Sim_1(d, q) = \sum_{i=1}^l C(s_i, q) \quad (3)$$

Where S_i are sentences in d. The RSV for a system that employ sentence-query similarity on top of the basic model is computed as:

$$RSV(d, q) = sim_0(d, q) + \alpha \cdot sim_1(d, q) \quad (4)$$

The coefficient α is used to control the weight of sentence-query similarity in the overall RSV value.

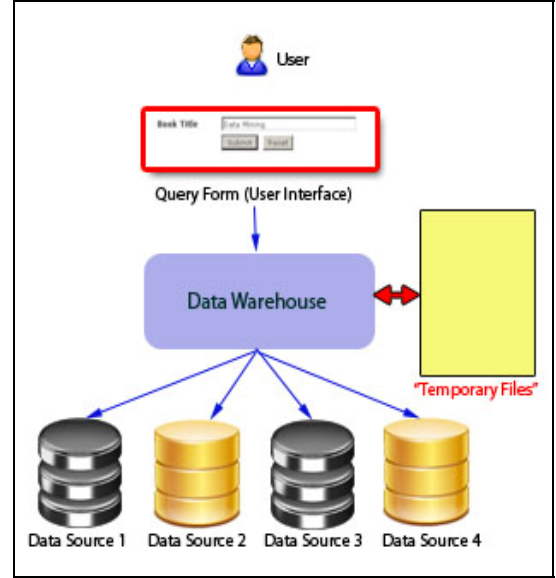


Figure 7. Data Flow for Retrieving Process

Based on figure 7, web user enter any keywords into user interface. After application receive the keywords, system automatically, search and match any keywords in temporary file. This temporary file stored at data warehouse. After searching process for keywords finish, server side will go only selected data sources. Based on figure above, only two data sources involved (yellow colour). System will search and retrieve data from these data sources. Finally, system will display the result to web user.

Figure 6 shows the algorithm in server side. After get possible query from exploitation process, system will search and find a suitable keywords. Then, the keywords will map to physical data source to retrieve the relevant data and display to web user.

6. EXPERIMENTAL RESULTS

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 5.

6.1 Implementation

The methodology has been implemented in a prototype using PHP and MySQL technologies. PHP is programming language to develop a interface engine based on system architecture in section 3. Meanwhile, MySQL is one of the Database Management System (DBMS) used to manage data in databases. This environment development was chosen because it would make the system portable and easily accessible through the World Wide Web. This development concentrate on two parts; client side and server side. On client side, this part receives keywords entered by web users. This keywords are called “initial query”. After that, this initial query will through exploitation query process, refinement query process and matching query process. On server

side, data schema from different data sources will extract and store into data warehouse (physical data still in data sources).

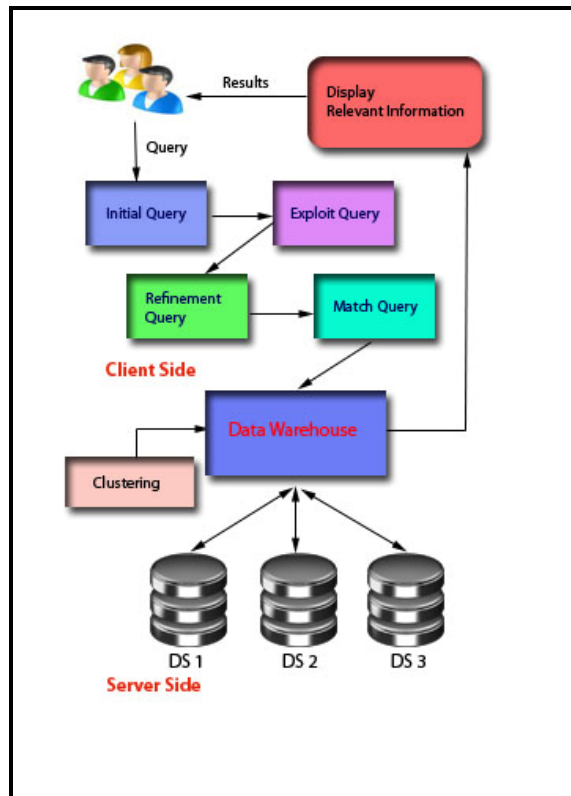


Figure 8. System Architecture

Figure 7 shows the system architecture from web users enter the query, processes in client side and server side, and finally, the system display a relevant information to web users.

Each data schema will assign with suitable keywords. The purpose of this assigning process is to make query matching with more easy and fast.

6.2 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.

Figure 9: Search Form

The user interface is a Web site which is allows the web users to enter initiate keywords (Fig 9). The user requests information through queries submitted to the system via an HTML form. Once

the query has been submitted, it is sent to the client side. Once this query has been submitted, this query will define as an *initial query*. After that, this initial query will exploit into two words; data and mining. Then, the next process is refinement possible query. This process will refine and display any possible query, for instance “data mining” or “mining data”. Last module in client side is matching process. This process will communicate to data warehouse in order to find and match a possible query with keywords in data warehouse. Meanwhile, on server side, keywords in data warehouse have been assigned to data schema. The code processes in client side are searching and retrieving. In this concept, data schema was stored in data warehouse, but physical data store in data sources. Once any change occurs in data sources, automatically data schema in data warehouse also changes. Based on the data schema, system directly maps to certain data sources. System automatically retrieves any possible and relevant information.

Product Code	Book Name	Price (RM)
1000109132	Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)	120.00
100342132	Introduction to Data Mining	78.00
10129167	Handbook of Statistical Analysis and Data Mining Applications	150.00
100123132	Data Mining: Concepts and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems)	230.00
102321132	The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)	60.00
104519132	Principles of Data Mining (Undergraduate Topics in Computer Science)	70.00
10239132	Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management	200.00

Figure 10: Result

In figure 10, all relevant information will display to web user. These all relevant information is based on keywords entered by web user and through processes or modules on client side and server side.

Table 1: Results and Comparison

	Sample Query	Number of Relevance Data	Response Time (Seconds)
Heterogeneous Database Architecture	Data Mining	20	0.20
	Data Warehouse	45	0.45
	Information Retrieval	12	0.15
	Database Integration	7	0.10
	Education	8	0.13
	University	7	0.14
Proposed Heterogeneous Database Architecture	Data Mining	20	0.18
	Data Warehouse	34	0.33
	Information Retrieval	10	0.12
	Database Integration	7	0.09
	Education	5	0.10
	University	6	0.11

6.3 Analysis of Proposed Architecture

Table 1 shows the number of sample queries have been executed and the number of relevance data was display to web users. Table above also, shows the comparison between heterogeneous databases architecture and proposed heterogeneous databases architecture in term of response time. According to table, web response time for web query processing using proposed architecture is better compared to heterogeneous databases architecture. Designing algorithms in proposed heterogeneous databases architecture was implemented in this architecture and the results also shows web query processing was improved. Components in this architecture such as initial query exploit query, refinement query and match query help a proposed architecture become efficient in term response time for web query processing. Clustering data in data warehouse also was implemented in order to support other components so that this architecture can be implemented and produce a better results.

6.4 Validation

Sample query have been executed in section 5.2 with the results shown in figure 10. The results shows 6 relevant information based on keywords "Data Mining". These sample query show the improvement of web query processing using combination among semantic, linguistic, data warehouse and clustering approach.

7. CONCLUSION & FUTURE WORK

Based on methodology and experiment above, results show that this architecture can be used for accessing data from different data sources. This architecture is able to improve query processing and display an accurate or relevant result to web users. In this paper, we present semantic, linguistic, data warehouse and clustering approach, a prototype for heterogeneous databases and illustrated them with example. However, this architecture has a potential to improve with added suitable components or modules. These addition components or modules can make a system more efficient and effective especially for web query processing.

8. ACKNOWLEDGMENTS

Special thanks are owned to my friend Che Mat Ismail for support and advice.

9. REFERENCES

- [1] Jordi Coness, Veda C.Storey, Vijayan Sugumaran. Improving web-query processing through semantic web knowledge. *Data & Knowledge Engineering* 66 (2008), 18-34.
- [2] J.Hartman et. al. Ontology based query refinement for semantic portal, in: *integrated publication and information system to virtual information and knowledge environment 2005*, 2005, pp. 41-50
- [3] B. Velez et al. Fast and effective query refinement, in: *Proceeding of the 20th annual international ACM SIGIR Conference on Research Development in Information Retrieval*, 1997, pp. 6-15.
- [4] Shi Ming Huang, Tung-Hsiang Chou, Jia-Lang Seng. Data warehouse enhancement: A semantic cube model approach. *Information Science* 177 (2007), 2238 – 2254.
- [5] Samuel Robert Collins, Shamkant Navathe, Leo Mark. XML schema mapping for heterogeneous database access. *Information and Software Technology* 44(2002), 251-257.
- [6] M. Bright, A. Hurson, S. Pakzad. A taxonomy and current issues in multidatabase system, *IEEE Computer* 25(3) (1992), 50-60.
- [7] Jie Cao, Wen Hou, Tingyou Cai. Research of Heterogeneous Database Integration System Based on E-Business, *IEEE* 2008, 186-189.
- [8] Carole Goble, Robert Stevens. State of the Nation in Data Integration for Bioinformatics. *Journal of Informatics* 41(2008), 687 – 693.
- [9] Eui Kyu Park, Dong Yul Ra, Myung Gil Jang. Techniques for Improving Web Retrieval Effectiveness. *Information Processing and Management* 41(2005), 1207 – 1233.
- [10] Walter Sujansky. Heterogeneous Database Integration in Biomedicine. *Journal of Biomedical Informatics* 34(2001), 285 – 298.
- [11] Matthias Butenuth, Guido V. Gosseln, Michael Tiedge, Christian Heipke, Udo Lipect, Monika Sester. Integration of Heterogeneous Geospatial Data in a Federated Database. *Journal of Photogrammetry & Remote Sensing* 62(2007), 328 – 346.
- [12] Jia Lang Seng, I.L Kong. A Schema and Ontology Aided Intelligent Information Integration. *Expert System with Application* 36(2009), 10538 – 10550.