HYBRID FEDERATED DATA WAREHOUSE INTEGRATION MODEL: IMPLEMENTATION IN MUD CRABS CASE STUDY

Mustafa Man, W. Aezwani W.A. Bakar, Noraida Hj. Ali and Masita Abd. Jalil

Department of Computer Science
School of Informatics and Applied Mathematics
Universiti Malaysia Terengganu
21030 Kuala Terengganu, Terengganu.
mustafaman@umt.edu.my, beny2194@yahoo.com, aida@umt.edu.my, masita@umt.edu.my

ABSTRACT. *Data integration is considered as one of the hot issues to be solved especially in integrating unstructured data with multiple types and formats. This paper introduces a new model for integrating multiple types of heterogeneous data applying to mud crabs case study in Setiu Wetland (SW). The Hybrid Federated Data Warehouse (HyFeDWare) model combines two approaches which are Data Warehouse and Federated Database. Simulation result shows that the processing time for integration of unstructured biodiversity data of mud crabs are lesser than 2 seconds for 12 rows of 7 MB data. This model generally could be used to integrate any types and format of data in distributed environment.*

KEYWORDS: Data Integration, Data Warehouse, Federated database, Distributed Environment.

INTRODUCTION

Data integration involves combining the data from different sources and presents these data in a unified view (Mustafa, *et. al*, 2011). Many organizations are concerned about integrating the structured and unstructured data to retrieve valuable information and knowledge that are useful for the organizations in terms of their benefits (Aezwani, *et al*, 2010; Mustafa, *et al., 2012*). Data integration has become a very challenging task with consequent growing volume of the data especially in integrating data from distributed, heterogeneous and large volume of data sources (Mustafa, *et al*, 2011).

Data warehouse serves as a repository for an organization's historical data. Analysis, discovering new knowledge, data mining and generating report can be done by using the data stored in data warehouse (Jose, *et al*., 2009) Data warehouse is an approach that is used to integrate multiple sources of data and store the data for further analysis in order to provide or generate meaningful information for the organizations. Data warehouse focuses on data translation, extracting the data from different sources, transforming and converting the extracted data, and finally importing or loading it into the data warehouse (Catriel & Milo, 1999; A. Roth, *et al.,* 2002; Bowen, 2012).

Federated database or sometimes called virtual database is a method of providing a unified view and querying multiple databases as if the databases were one single entity. The federated database eliminates the need to duplicate all the accessible data from multiple databases (Christine & Spaccapietra, 1998). Federated database focuses on query translation which is different from the data warehouses that focus on data translation. All queries are executed at the sources by translating a query against federated database into a query against the sources (Ming & Fu, 2014).

Mud crab is one of the main natural resources that exist in Setiu Wetland (SW). It mainly contains three types of species which are *S. olivacea*, *S. transquebarica* and *S. Paramamosain.* A lot of studies and researches have been conducted at SW to collect data about mud crabs species, size, sex ratio, and weight of mud crabs (Ikhwanuddin, *et al.,* 2012). The data collected are commonly stored in electronic form and therefore it is vital to provide a good management system to ensure it is fully utilized. However the issue persists when, data storage is in disparate databases and this will cause some problem to researchers to obtain the data needed. In the worst case, each types of data mainly stored in multiple formats. Therefore, there must be a standardize data integration scheme to overcome the pertaining issues (Joan Bader *et al.,* 1999; Hossain *et al.,* 2012).

The rest of the paper is organized as follows. Section 2 describes the related works of data integration. Section 3 explains the proposed model. Section 4 outlines on the experimentation being conducted. This is followed by result and discussion in section 5. Finally, conclusion and future direction is reported in section 6.

RELATED WORK

There are multiple advantages for having data integrated. One of the advantages is that integrated data can contribute in minimizing the inconsistent data. With a lot of unmanaged inconsistency, this can have some impacts on the organizations or users who want to use the data. Data searching and collecting will be a difficult task with inconsistency of the data and data integration can help to reduce the inconsistency hence provide users an easier method to find the data needed.

A. BioMart

Managing the biological data is considered as a difficult task due to the complexity of the biological data and is usually not well-defined. The factors affecting the effort in combining are geographical distributed data sources and heterogeneous data. Therefore, it will encounter difficulties during presenting the required data in a unified view (Kasprzyk, 2011).

Ontario Institute for Cancer Research (OICR) and the European Bioinformatics Institute (EBI) have jointly developed a query-oriented data integration system called BioMart. This BioMart software adapts two fundamental concepts to tackle the problems mentioned before. The two concepts are data agnostic modeling and data federation. During data modeling, it is often to encounter tasks that are hard and consume a lot of time. As a solution to it, data agnostic modeling is used to simplify the tasks. Next, to be able to provide a unified view of interface, data federation is used to make it possible to organize multiple, different and geographical distributed database system.

The BioMart system provides a single point of access where it allows retrieval of the data from the biological data storage. Later, the BioMart Central Portal, web server interface is introduced to provide the function to access to variety of datasets. The dataset is queried independently by users to search over information for solving complex problems that may be situated at different geographical locations. A query is sent to BioMart Central Portal, then it accesses that data from data storage. The queries for BioMart are classified into two types. The first type of query is called metadata access. Information retrieval are done using these metadata access request to retrieve which databases, datasets, filters, attributes and associated formatters are made available. The second type is data access where query is used to request for accessing the required data that is available (Haider *et al.,* 2009)

B. PDBWiki

The name of PDBWiki came from the two separate system which are the Protein Data Bank (PDB) and Wikipedia (PDBWiki, 2015). This PDBWiki system can be said to combine some of the features from both systems where it merges the well-known wiki system and the PDB database of heterogeneous biological data. PDBWiki has a special feature which is community annotation that is used to overcome the problem faced by the centrally controlled databases of biological data. The problem stated above is that it does not allow editing by the users and eventually the findings will become out of date as it requires time to get permitted to alter the data. As a consequence, it will affect the accuracy of the data if there is a known mistake or error and it cannot be corrected as it requires more time to get the mistakes corrected. By introducing the PDBWiki, it provides annotation function to gather knowledge from different researchers from all around the world to ensure the data and information presented are the latest version and correct. The PDBWiki is a semi-structured approach where it has no restrictions to a particular format and it allows the users to write in free text comments. The PDBWiki provides a system in order to classify the annotations in a hierarchical way.

MediaWiki is used to implement the PDBWiki where MediaWiki has some nice key features that are versioning, notifications, watch-list and transparency. Besides that, MediaWiki has developed two extensions to provide better users with better experience using the system. The extensions are providing a user comment form and managing the image by adjusting the default image functionality. Weekly synchronization will be done with PDBWiki if any new data that is added or updated in PDB in which new entry in PDB will create a new page while the data that is updated will be recreated. However, the comments on the updated data will be saved in the system and will be able to link to the updated data entry. In order to process the updates in PDB, OpenMMS software package is the software needed and it will create a relational database. SQL queries are used to facilitate the update process with the data collected from relational database and finally with the help of Python Wikipedia Robot framework, the update is completed.

In the case of PDBWiki providing the annotation and users comments, it is said to have some similarities with the bug-tracking system. This is due to the ability to collect the reports of the problems found, feature requests and act as a discussion forum. In addition to that, experts from the field will be able to discuss the findings they found through this software where this function provides an easy way for them to have discussions. Furthermore, this bug-tracking like feature will surely facilitate to have a better and more accurate data (Stehr *et al.,* 2010).

Although the PDBWiki provides a number of benefits to the users, it has some shortcomings that are needed to be noted. Due to the semi-structured approach of the system, it has the possibility to generate some unstructured data that is hard to be integrated. The problem of integrating the unstructured data could lead to loss of valuable information that might be significant to the researches. Besides the short-coming, the authors who wrote the papers are less likely to get recognized. The authors want their work to be recognized worldwide however this software might not able to help them in this perspective. As a result, it will suffer from low contribution from users to propose their findings in PDBWiki.

C. Oracle Database 12c

Oracle Database 12c has been focusing on improving performance for the Unstructured Data query and analysis (Greenwald *et al.,* 2013). Next, improvement of integration has been done on these data types with other features in Oracle Database. Besides that, Oracle Database 12c also attempted to simplify the application code by moving more of the application logic and analytics associated with specific data types and analysis. Oracle Database 12c manages the data differently by determining the way the data is created and usage of the data. For the unstructured data, Oracle Database 12c provides a special management specific to manage XML, Text, Spatial, Network Data Model graphs and RDF Semantic graphs, and Multimedia and DICOM data.

Oracle Database 12c supports the unstructured data in the aspects of storage, data types, management, indexing and in-database analytics. To date, there are several types of Large Object (LOB) is used to store large size of data and it is suitable to use in managing semi-structured data and unstructured data. One such example of LOBs is Binary Large Objects (BLOBs) where it is adapted and served as a container to keep the unstructured data inside database tables. Besides that, in order to analyze and manipulate the manipulating XML documents, text, multimedia contents and geospatial information, incorporating intelligent data types and optimizing data structures with operators have been done by the Oracle Database.

Although the Oracle Database 12c provides a lot of advantages, and are crucial and are a big step forward in managing the unstructured data, the cost of implementing it is very high and it is usually unaffordable by all the organizations. In addition to that, the maintenance cost will be high due to the updates that are required to ensure the data is up to date. Therefore, it is suggested that more researches should be carried out to search for a lower cost yet high quality of data integration approach where all the organizations and users can enjoy the benefits of having unstructured data managed (Oracle, 2013). The summary of comparison method used in three approaches is depicted in table 1.

Table 1: Comparison of three model/system.

Features	**BioMart**	**PDBWiki**	**Oracle Database 12c**
Approach Used	Federated databasing	Wiki-based integration	Data warehousing
Support multiple data sources types	It can support and integrate multiple sources	It can support and integrate multiple sources	It can support and integrate multiple sources
Point of access	Single point of access.	Single point of access.	Single point of access.
Network Connectivity	It depends heavily on network connectivity.	It depends lesser on network connectivity. It needs to be access through browsers.	It eliminate the need of connecting the network because all the queries occur within the data warehouse.
Annotation ability	It does not allow annotate or edit by users.	It allows annotate or edit by users.	It does not allow annotate or edit by users.
Authorship	Authorship is recognized because it does not allow editing.	Authorship is not recognized.	Authorship is recognized because it does not allow editing.
Speed of retrieving data	It depends on the network connectivity and its queries	N/A	It retrieves data faster because the data is stored within the data warehouse.
Updates	It is always synchronizing with various databases.	It will synchronize with PDB weekly.	It has to continuously update data that is guided by human to keep the data up to date.
Efficiency	Efficiency depends on the network connectivity. It may cause low efficiency when network connectivity is low or unavailable.	It can be rated as moderate efficiency because it synchronizes data weekly which is said it is not always up to date.	High efficiency during data retrieval. However, the data has to be continuously updated to be efficient.
Cost	Cost saving due to the storage needed.	Low cost in maintenance.	High in cost to provide a storage and maintenance cost.

PROPOSED MODEL

It is obvious that data warehouse and federated database have their own strengths and weaknesses. Some of those weaknesses can be theoretically covered or overcame by each other. Figure 1 shows the proposed model called HyFedWare that combine both approaches (Federated Data and Data Warehouse) to increase efficiency and overcome some of the short comings of data warehouse and federated database that work separately.

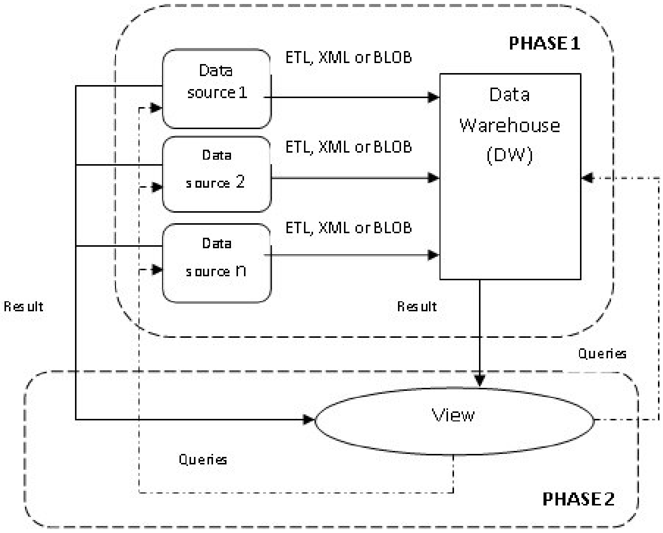

Figure 1: The Hybrid Federated Data Warehouse (HyFeDWare) model

Based on figure 1, in phase 1, the data needed in the data sources will be extracted by using the Extract, Transform and Load (ETL) process defined in each data sources. Then the extracted data will be standardized into the same format or common schema since all the data sources might be in different format or using different schema. Therefore, it is needed to define a common schema in order to integrate the data. In phase 2, once the query is received, the results will be returned to provide the view of the data required. The response result may be different prior to the queries being sent. For example, two different searches will be provided which is different into general search and specific search. With specifying the searches, the machine will know which place to look for the data.

Prior to conversion of formats and standardization, the types of outputs produced would be in Extensible Markup Language (XML) and Binary Big Object (BLOB) formats that can be fitted into data warehouse. There is another alternative in which the outputs are not loaded into the data warehouse. This is where the federated database approach fits in the proposed model. The request fetches the outputs and views them in a unified way as data are from one single database. This proposed model acquires a smaller repository size rather than having a bigger size of storage for storing all integrated data from data sources. Data warehouse serves as a repository to store *'general data'* about mud crabs. General data refers to data that is said to have fewer tendencies to change or fixed data.

General or fixed data is stored in data warehouse and this indicates that the other data which is not stored in data warehouse is fetched or retrieved using the federated method where data is directly obtained from the data sources. The specific data or non-fixed data of the mud crabs will contain the data which will frequently change or vary across time.

As the general overview, the proposed model integrates the data about mud crabs from different databases which maybe in different formats and geographically apart. The data are extracted and the format is standardized. This is due to different databases might use different types of formats to store the data. Therefore, there exists the need to standardize the data to be integrated.

EXPERIMENTATION

A. Experimental Design

Experiments are performed on Windows 7 Home Premium, 64-bit Operating System, and Intel Core-i5 2410 M, 2.3GHz and 4GB memory. The machine specification is provided as a benchmark for the measurement. Simulation is conducted on data sources with the number of rows of integrated data and sizes of those data. Mud crabs data is in flat file i.e. text format (pdf, docx, xls) and image format (jpg) in blob type. The raw data is then processed from unstructured to structured format using TOS (Talend Open Studio) software (Bowen, 2012) by combining multiple types of data in different types of database format (Oracle, 2013). Performance is measured in terms of seconds, minutes or hours depending on the size of data input.

In data warehouse approach, it is designed by using TOS where it provides very useful functions to perform all processes needed. The result of the data warehouse approach can be obtained by running the designed flow of process inside TOS as illustrated in Fig. 2. It shows how unstructured data in database is being integrated into data warehouse.

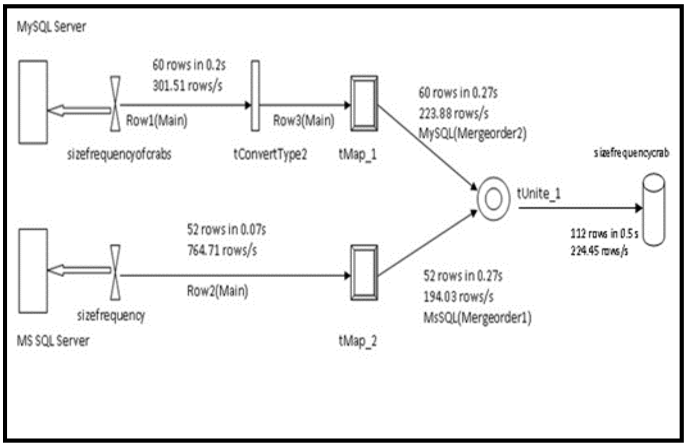

Figure 2: Result of running the job designed for the process flow of integrating data into data warehouse using TOS

In figure 2, we test this approach with the real data (secondary data) of 60 rows of frequency size of mud crabs in MySQL database format and 52 rows of frequency size of mud crabs in MSSQL database format. The result shows that this approach can integrate the two different formats of database easily.

The result of time taken versus numbers of rows is shown in figure 3. This figure illustrates the time taken for the whole process of data warehousing and transformed into a line graph to give a better view on the overall performance.

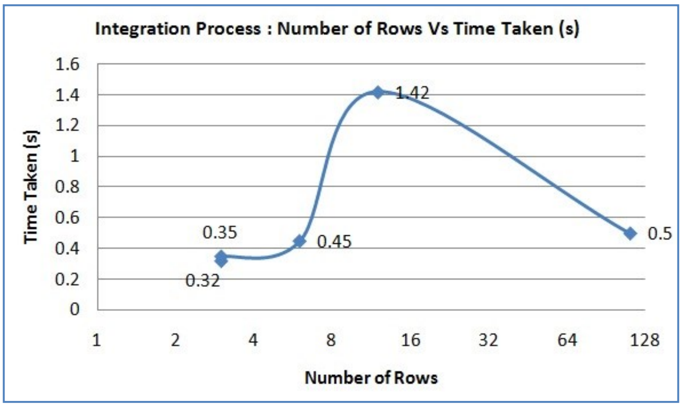

Figure 3: Line graph for different rows against time taken for integrating into data warehouse

Meanwhile, in federated database approach, measurement criteria are in terms of the retrieval speed of the data from data sources. Therefore, a simple web page is designed in order to visualize the retrieval time (time taken to load the page). Time taken to display the unstructured data will be longer as compared to structured data due to the size of the data being retrieved. However, the time taken to load all the data can be said to be acceptable although the time taken is longer.

The loading of unstructured data is a bit slower in web browser. This could have been caused by the processor speed and memory used by the machine during processing as the memory usage has gone up from 75 to 85 percent. The memory used by the machine boosted up to 76 percent with only Google Chrome v33.0.1750.154m, XAMPP Control Panel v3.2.1 and NetBeans IDE 7.3 opened. The image data from all database types is retrieved by queries using the hybrid processing tool called HyFeDWare. The performance comparison between the proposed model and the two previous approaches is tabulated in table 2.

Table 2: Performance Comparison of Proposed Model and Previous Model

Criteria	**Data Warehouse**	**Federated Database**	**Hybrid Federated Data Warehouse**
Network Dependency	Low	High	Medium
Time required to get latest data	Longer due to updates needed to be done to DW and cannot access to the latest data immediately	Short because it fetches data directly from the data sources.	Short because it fetches latest data from data sources directly.
Retrieval speed	Fast because data is stored inside DW and can be accessed directly.	Vary. This is because it highly dependent to the network connection.	Vary. This is because it has to depend on requested data. If the general data is requested, the speed of retrieving will be fast whereas specific data will depend on the network connection.
Cost	High in purchasing DW and maintenance fees.	Low. This is because it does not need to purchase large size of repository.	Medium. Smaller size of DW will be purchased to store the general data.
Efficiency	High efficiency in retrieving data. However, DW need to be regularly updated to be efficient.	Depend on the network connection to measure the efficiency in retrieving data.	Medium. Depend on which data to be retrieved.

From table 2, it can be seen that the proposed model, HyFedWare performs moderately in between data warehouse and federated database approaches. Though, the result shows a significant minor improvement as to compare with previous approaches.

CONCLUSION & FUTURE WORK

It could be concluded from the experimental results that the model can be further improved to increase the efficiency of the integration process especially in handling unstructured data in different types of databases. This could be done through having a deeper study about Talend Open Studio to have a simpler method of process that results in lesser line of codes.

With fewer lines of codes, efficiency could be achieved by reducing response time to increase performance.

Performance for federated database can be improved by using UnityJDBC and SQuirrel SQL Client for the next testing process. These tools could serve as the engine that translates SQL queries into different understandable SQL queries for respective different databases.

In addition, the ontology concept can also be applied to help in identifying the unstructured data especially in PDF files and Microsoft Word Documents. Occurrences of each word in a text can be recorded and the top 5 highest occurrences could be chosen and set as keywords of the text files. This will help in searching for the useful data as required by the researchers.

ACKNOWLEDGMENT

The authors wish to express sincere gratitude to Assoc. Prof. Dr. Mohd Ikhwanuddin Abdullah who has guided and provided the needed data and information for this research and also to Ministry of Education Malaysia (MOE) for their support through Niche Research Grant Scheme (NRGS).

REFERENCES

Aezwani, W.A.B. et al., 2010,"SIDIF: Location based technique as a determinant of effectiveness and efficiency in artificial reefs development project."*Information Technology (ITSim), 2010 International Symposium in*. Vol. 2. IEEE,.

Bowen, J. (2012). *Getting Started with Talend Open Studio for Data Integration*. Packt Publishing Ltd.

Catriel, B. & Milo, T. 1999. Schemas for Integration and Translation of Structured and Semi-Structured Data. Database Theory—ICDT'99. Springer Berlin Heidelberg, 296-313.

Christine, P. and Spaccapietra. S., 1998, "Issues and approaches of database integration." *Communications of the ACM* 41.5es : 166-178.

Greenwald, R., Stackowiak, R. & Stern, J. (2013). *Oracle Essentials: Oracle Database 12c*. " O'Reilly Media, Inc.".

Haider, S., Ballester, B., Smedley, D., Zhang, J., Rice, P. & Kasprzyk, A. (2009). BioMart Central Portal—unified access to biological data. *Nucleic acids research*, *37*(suppl 2), W23-W27.

Hossain, M., Harari, N., Semere, D., Mårtensson, P., Ng, A. & Andersson, M. (2012). Integrated modeling and application of standardized data schema. In*5th Swedish Production Symposium,(SPS12), 6-8 November, 2012, Linköping, Sweden*. The Swedish Production Academy.

Ikhwanuddin, M. et al., 2012, "Improved hatchery-rearing techniques for juvenile production of blue swimming crab, Portunus pelagicus (Linnaeus, 1758)."*Aquaculture Research* 43.9 : 1251-1259.

Joan Bader, C. H., Razo, J., Madnick, S. & Siegel, M. (1999). An analysis of data standardization across a capital markets/financial services firm.

Jose, Z., Pardillo, J. & Trujillo, J. 2009. A UML Profile for the Conceptual Modeling Of Data-Mining With Time-Series In Data Warehouses. Information and Software Technology 51, 6: 977-992.

Kasprzyk, A. (2011). BioMart: driving a paradigm change in biological data management. *Database*, *2011*, bar049.

Ming Shuai, W. and Fu. X. F., 2014, "A Method of Heterogeneous Data Integration Based on SOA." *Applied Mechanics and Materials* 536 : 494-498.

Mustafa, M. et al. 2011, "Designing multiple types of spatial and non spatial databases integration model using formal specification approach." *Software Engineering (MySEC), 2011 5th Malaysian Conference in*. IEEE,.

Mustafa, M. et al. 2012, "Integration Model for Multiple Types of Spatial and Non Spatial Databases." *Signal Processing and Information Technology*. Springer Berlin Heidelberg,. 95-101.

Oracle (2013). *Unstructured Data Management with Oracle Database 12c*. Retrieved October 29, 2013 from http://www.oracle.com/technetwork/database/information-management/unstructured-data-management-wp-12c-1896121.pdf

Roth, M. A. et al., 2002, "Information integration: A new generation of information technology." *IBM Systems Journal* 41.4: 563-577.

Stehr, H., Duarte, J. M., Lappe, M., Bhak, J. & Bolser, D. M. (2010). PDBWiki: added value through community annotation of the Protein Data Bank.*Database*, *2010*, baq009.