DOMAIN-BASED DISTRIBUTED MEDIATION SYSTEM FOR LARGE-SCALE DATA INTEGRATION

Ghada Hasssan[1], Hamidah Ibrahim [2], Md.Nasir Sulaiman and Rasali Yakob[3]

[1,2,3,4]*Universiti Putra Malaysia, Malaysia*
[1]*oghada@hotmail.com, [2]hamidah@fsktm.upm.edu.my, [3]nasir@fsktm.upm.edu.my and [4]razali@fsktm.upm.edu.my*

ABSTRACT. The increasing use of computers and the development of communication infrastructures resulted in abundance of information on the networks and extremely need for querying and integrating data from large number of data sources, especially for scientific applications. Distributed mediation system performs this task fragmentally, where each mediator in the system architecture responsible for querying subset of data sources. To make such type of systems success in large scale, we must deal with two issues: logical distribution of the data sources and mediators in the system architecture, and logical interaction between the mediators. To handle these issues, this paper proposes a domain based distributed mediation system called Multi-Mediators System for Large scale Data Integration (MMSLDI). MMSLDI enables the reducing of the general query response time by eliminating unnecessary visits to the data sources that do not contribute to the answer of the query. Moreover, it satisfies the requirements of adaptation to various kinds of domains, decentralized control, and automation of its processes.

Keywords: data integration, Distributed mediation system, Multi-Mediators System for Large scale Data Integration (MMSLDI)

INTRODUCTION

Data integration is the process of combining heterogeneous data residing at different sources in order to provide the user with a unified view of these data (Fowler et al., 2004). Mediation architecture (Halvey et al., 2003) was proposed as a solution to integrate heterogeneous data sources in a specific domain by adding a layer between the application layer and the data sources in the system. A mediation layer handles the responsibilities of accessing the heterogeneous data sources and presents the integrated data to the user (Ezenwoye et al., 2004).

As the number of data sources increases, the centralized mediator architecture becomes an administrative and performance bottleneck. It has been identified in earlier research on mediator systems (Bassem & Samir, 2009) that distributed mediation architecture is necessary in order to avoid these bottlenecks while providing modularity, scalability, and reuse of specification in the integration process (Du & Shan, 1995).

In the distributed mediation architecture, the integration process is distributed among many mediators where mediators can access other mediators. The challenge in making such a system succeed at a large scale is twofold (Gardarin, Dragan, & Yeh, 2008). First, we need a simple concept for logical distribution of data sources. Second, we need efficient mechanisms for logical interaction between the mediators with respect to distributed query routing and execution that capable of handling well costly computations and large data transfers.

In data integration, the variation in the meaning of data sources content will affect the amount of useful data that can be obtained. Accordingly, there is a need to address the issue of Logical distribution of the data sources from the content perspective, which aid in eliminating unnecessary visited to data sources that cannot contribute to the answer of the query in order to reduce the query execution time, network load, and resource consumption.

Two methods exist for logical interaction between mediators. The first one uses one mediator to query other mediators such as in (Aboulnaga, El Gebaly, & Wong, 2007). This method is not preferable because the distributed mediation architecture is to avoid having any component that constitutes a central point of failure and when the numbers of mediators increase it reproduces the problem of single mediator, especially for large number of data sources. The second method uses P2P fashion to route the query between mediators such as in (Ezenwoye et al., 2004).

Current P2P routing techniques, however, are either unscalable, unrelaiable, slow response time or lack robustness. Usually they are based on one of the following techniques: pure p2p (Jurczyk, Xiong, & Sunderam, 2008), central index (Katchaounov, 2003), distributed index (Ege et al., 2004; Gardarin, Dragan, & Yeh, 2008; Huebsch et al., 2005), and super peer (Halvey et al., 2003; Loser et al., 2003; Beneventano et al., 2005). The description of these techniques and their drawbacks as follows.

In pure P2P, each peer floods the query to the directly connected peers, which themselves flood to their peers, until the request is answered or a maximum number of flooding steps occur. This may result in many redundant computations performed by each of the underlying peer, as well as in many redundant network accesses and data transfers.

The objective of peer indexing is to allow peers to select from an index the best neighbor to send a query to, rather than flooding or random selection. In a centralized index, a single server keeps references to data on many peers. The drawback of this method is when the node that holds the central index fails, the system will crash.

Distributed index techniques, such as Distributed Hash Table (DHT) algorithms are based on maintaining some knowledge by each node about some other nodes (but not all). The general purpose of these algorithms is to map a value onto a key using a hash function. Also this method has drawbacks when each peer joins the system needs to add DHT and when a data source joins the system needs to add new index to DHTs. Thus, each join or leaving of even a single peer requires considerable overhead in reorganization for all tables.

Super-peer is a node that acts as a centralized server to a subset of clients peers. Clients submit queries to their super-peer and receive results from it. Moreover, super-peers are also connected to each other, routing messages among them, and submitting and answering queries on behalf of their clients and also interconnected super-peers (Yang & Garcia-Molina, 2003). Super-peers introduce a single-point of failure for its clients, and still there is a need to make them more reliable.

Based on the proposed query routing techniques and their drawbacks, we come to a conclusion that additional research is needed to address the issues of query routing in large scale data integration systems, and to present a new solution that improve the efficiency and fault tolerance of query routing techniques for such systems.

This paper provides a description of the proposed data integration system (MMSLDI). The logical distribution of data sources and mediators is given in the next section. The details of the logical interaction between the mediators, and the query routing mechanism are presented in the latter sections.

LOGICAL DISTRIBUTION OF DATA SURCES AND MEDIATORS IN MMSLDI

Our proposed integration system (MMSLDI) which illustrated in **Figure 2** is a distributed mediator system that consists of a set of sub-systems called mapping systems and a set of data sources. The mapping system is a mediator that facilitates querying of a set of heterogeneous data sources that are distributed over multiple sites. This set of sources is called the domain of the mapping system. The mapping system provides the required components to process user queries that their results are obtained from its domain.

Each domain consists of a mapping system called Sub-Main Mapping System (SMMS) that acts as a header of the domain, and a set of Sub-Domains (SD) that organize data sources. Each sub-domain encompasses a Mapping System (MS) as a header and a set of data sources.

The header of each domain acts as a method of interaction of its domain in the integration system. The interaction refers to the forwarding of a specific set of user queries that their answers can be obtained from the other domains, or the forwarding of the answers that can be obtained from the domain to the other domains. In this architecture, initially, the set of all data sources of a particular field (e.g. biological field) are registered and then are grouped into SDs. The grouping is performed based on the topics that are provided by the sources.

The direct communications with the data sources for submitting queries and obtaining their answers are performed by the MSs of the sub-domains. The user interacts directly with the MSs for posing queries.

The different domains are linked together through the Main Mapping System (MMS) that acts as an intermediate system for forwarding a set of user queries that do not have answers in the domains where they are posed to the other domains, and returning the answers of these queries to their original domains. The MMS has the widest view of all domains in the integration system in that it stores the metadata of a full version of the domain's schema, the information about the data sources that are included in each domain, and the set of correspondences between the universal schema and the data sources schema. Both SMMS and MS store only the metadata that are related to their domains.

Based on the above description of the architecture, we can formally define MMSLDI as follows.

Definition. MMSLDI is a 5-tuple $<S, M, D, I, U>$, where:
$S = \{s_1, \ldots, s_n\}$ is a finite set of available data sources of a particular field.
$M = \{m_1, \ldots, m_m\}$ is a finite set of mapping systems that are required to facilitate the processing of the user queries. This set contains the following categories of the mapping systems:
- MSs that have direct communications with the users and a specific set of sources.
- SMMSs that are responsible for a particular set of MSs to facilitate their interactions.
- MMS that acts as the main medium for the interaction between SMMSs.

$D = \{d_1, \ldots, d_l\}$ is a finite set of domains that group together subsets of the elements of both S and M. Each domain has an element of M as a header and a set of sub domains.
$I = \{i_1, i_2, i_3\}$ is a finite set of mechanisms of interaction between the elements of M after they are grouped in the elements of D.
$U = \{u_1, \ldots, u_n\}$ is a set of users that interest in obtaining specific information from the particular field. Each user interacts directly with specific elements of M for query posing.

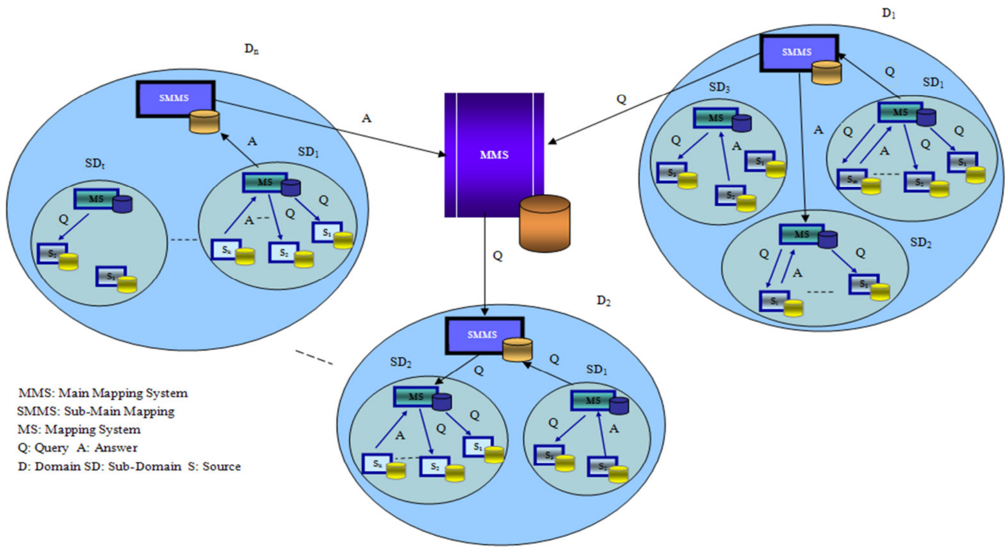

Figure 13.The Architecture of MMSLDI

LOGICAL INTERACTION BETWEEN MEDIATORS IN MMSLDI

The interaction between the different mapping systems in MMSLDI is performed through routing the query between them. For this purpose, domain based query routing mechanisms are proposed. In these mechanisms, the query is written using the ordinary query writing method as follows:

Select attribute_name(s)
From source_name
Where condition

Each MS receives a query from the user, extracts the query subject from the *Select* clause, and translates the query into a tentative query form without location operators (From clause). Then it forwards this query to other MS based on the routing algorithm. The receiver MS of the tentative query, checks the query subject against its local data. In the case that the answer exists, it fills the holes in the *From* clause by identifying a location operators containing potentially relevant data.

The routing mechanism to the other mapping systems is determined based on a similarity measure between the query subject and the domain schema. The user posts the query to any MS then the MS calculates the similarity (r) between the query subject and the domain schema. Based on the value of r $(0 \leq r \leq 1)$, the query can be forwarded from a MS to its parent or the child or the neighbors. For example, If r equals the highest value $(r=1)$, this indicates that the answer can be fully produced from the domain where the query is posted. Otherwise, the query can not be fully answered in its domain and it must be either fully $(r=0)$ or partially $(0< r <1)$ forwarded to the MMS to forward it to its neighbors. If the answer cannot be fully produced from the neighbors, SMMS forwards the query to MMS that forwards it to the domains where it can be answered.

If the query answer can be fully produced from the same domain, the interaction between the MSs can be realized through direct communications in a P2P manner. In order to implement this method, an **Inter-Domain Routing algorithm (IDR)** is designed. This algorithm allows peers that belong to the same domain to find the answer independently from their SMMS, This to avoid performance bottlenecks and single points of failure in SMMS.

If the query can not be produced from the same domain where the query is posted, the query must be forwarded to the SMMS to search between domains. To implement this method, a **Cross-Domain Routing Algorithm (CDR)** is designed. In this algorithm, super domain peers (SMMSs) are grouped together and allowed to find the answer independently to avoid performance bottlenecks and single points of failure in MMS.

Inter Domain Routing Algorithm

Each domain in our proposed system has its own schema. This schema is represented by a list of its attributes. If we assume D_1 schema has 20 attributes (A_1,…A_{20}), we can classify these attributes into groups according to their occurrences together. For example (G1:first name ,last name , G_2:professor name ,filed of study ,department).

Accordingly, D_1 schema can be represented as groups of attributes as follows:

G_1:A_1,A_2,…A_6

G_2:A_7…A_{10}

Each new joined node in the network sends its own schema information to the header of the domain. The header determines the attribute groups that belong to this node. Accordingly for each node in its domain, the header records the following information about the attribute groups:

N_1: G_1, G_3

N_2:G_4, G_2

Based on this information, the following facts can be deduced (~ means appear in):

G_1~N_1, N_5, N_{10}

G_2~N_4, N_8, N_3

Accordingly, additional interconnected groups of nodes are produced inside each domain (e.g. D_1 (G_1, G_2, G_3,….)). For each produced group, a coordinator is elected. This coordinator has a link to its group's nodes and the SMMS.

Each node when it joins the system, it will have links to the coordinators of the groups and the SMMS. When the node tests the incoming query against its domain schema, it can determine the attribute groups that the query belongs to in order to forward the query to the selected group coordinators.

Cross-Domain Routing Algorithm

This algorithm is dedicated to the SMMSs. In this algorithm, each group of SMMS nodes is organized into a ring topology. Each node has associated number *NN* ($1<NN<n$), where *n* is total number of nodes in each super domain. *n* is associated with the last node in the series and 1 is associated with the first node.

Each node has knowledge only about the contents of the two direct neighbors (i.e. the successor and predecessor) and stores only one address for remote node called *partner node*. Each node has two types of links: Direct-neighbor-link (DL) and Partner-link (PL). DL is used to link the node with its direct neighbors and the PL is used to link the node with its *partner node*. The message forwarding between *partner node*s will be in one direction, i.e. the anticlock wise direction.In the following, we illustrate how to compute the direct successor (*DNs*) and predecessor (*DNp*) neighbors and partner neighbor (*PN*) for each header node:

$$DNs = \left| NN+1 \right| \bmod n,\ 1<NN<n$$

$$DNp=\begin{cases} |NN-1| \bmod n, 1 < NN \le n \\ n, NN = 1 \end{cases}$$

$$PN= \left| NN+3 \right| \bmod n$$

Based on the aforementioned description, the CDR routing mechanism is as follows.

1. Each node receives the query, it determines the source of the query
2. If the query comes from MMS, the node checks it against its local schema only and then forwards it to its underline mapping system.

3. If the query comes from MS, the node checks its local knowledge about the neighbor schemas. If it can be answered, then it directly forwards it to the specific direct neighbor. Else it will be forwarded through the partner link.
4. If the source of the query is one of its neighbors, the node must check its local schema and neighbor schemas to determine if the answer can be provided. Otherwise, the partner link will be followed.

To strictly guarantee that the query forwarding is not exceeding number of hops in the worst case, the Descending Factor (DF) is proposed. Before the first forwarding for the query through the current level, the query must be tagged with DF and the value of DF is decreased by one in each forwarding for the query in this level. The value of DF is calculated as follows.

DF=round $\left|(N\text{-}1)/3\right|$, where N is number of nodes in the ring.

CONCLUSION

To cope with the problem of integrating data from large number of heterogeneous data sources, the paper presented MMSLDI that represents a multi-mediators architecture for large scale data integration. MMSLDI encompassed mapping based architecture for logical distribution of data sources and mediators, and routing mechanisms for the interaction between the mediators. MMSLDI eliminates unnecessary visits to data sources that do not contribute to the answer of the query through forwarding the queries only to the appropriate mediators. This enables the reduction of the resources consumption and the network overhead, improving the response time of the queries, and supporting the scalability of the system.

REFERENCES

Aboulnaga, A., El Gebaly, K., & Wong, D. (2007). µBE: user guided source selection and schema mediation for internet scale data integration. *Proceedings of IEEE 23rd International Conference on Data Engineering (ICDE 2007)* (Istanbul, Turkey), 186 - 195.

Bassem, B., & Samir, M. (2009). Metadata driven integration model for large scale data integration. *Proceedings of 2009 IEEE/ACS International Conference on Computer Systems and Applications*, 41-46.

Beneventano, D., Bergamaschi, S., Guerra, F., and Vincini, M. (2005). Querying a super-peer in a schema-based super-peer network. *Proceedings of the 2005/2006 international conference on Databases, information systems, and peer-to-peer computing*, Trondheim, Norway.

Du, W., and Shan, M. (1995). Query processing in Pegasus. *Object-oriented multidatabase systems: a solution for advanced applications*, Prentice Hall International (UK) Ltd, Hertfordshire, UK.

Ege, R. K., Yang, L., Kharma, Q., and Ni, X. (2004). Three-layered mediator architecture based on DHT. *Proceedings of the International Symposium on Parallel Architcture, Algorithm and Networks (I-SPAN)* ,Hong Kong, China.

Ezenwoye, O., Ege, R.,Yang, L., and Kharma, Q. (2004). A mediation framework for multimedia delivery. *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia, Maryland*, 251 – 256.

Fowler, J., Perry, B., Nodine, M., and Bargmeyer, B. (1999). Agent-based semantic interoperability in InfoSleuth. SIGMOD Record, 28, 60–67.

Gardarin, G., Dragan, F., and Yeh, L. (2008). P2P semantic mediation of web sources. *Lecture Notes in Business Information Processing*. volume 3, 3-16.

Halvey, A. Y., Ives, Z. G., Mork, P., Tartarinov, I. (2003). Piazza: data management infrastructure for semantic web applications. *Proceedings of the twelfth international conference on World Wide Web*, 556- 567.

Huebsch, R., Chun, B.N., Hellerstein, J.M., Loo, B.T., Maniatis, P., Roscoe, T., Shenker, S., Stoica, I., Yumerefendi, A.R. (2005). *The architecture of pier: an internet-scale query processor. Proceedings of 2nd Conference on Innovative Data Systems Research CIDR*.

Jurczyk, P., Xiong, L., and Sunderam, V. (2008). DObjects: Enabling distributed data services for metacomputing platforms. *Proceedings of the VLDB Endowment*, 1432-1435.

Katchaounov, T. (2003). Query Processing for Peer Mediator Databases *(Doctoral Dissertation, Uppsala University).* Retrieved from user.it.uu.se/~udbl/Theses/TimourKatchaounovPhD.pdf

Loser, A., Siberski,W., Wolpers, M., Nejdl, W. (2003). Information integration in schema based peer-to-peer networks. *Proceedings of the 15th international conference on Advanced information systems engineering*, Klagenfurt, Austria.

Yang, B., Garcia-Molina, H. (2003). Designing a super-peer network. *Proceedings of the 19th International Conference on Data Engineering,* Los Alamitos, USA.