

A Comparative Study Between the Management Methods of Semantic Cache

M. Ezziyyani, M. Bennouna, M. Essaaidi, L. Cherrat, M. Hlimi, M. En-naimi

Abstract — The diversity and the heterogeneity of distributed systems make the setting up of a mediation system an indispensable necessity. To reduce the time of answering the customer's requests, it is necessary to design a device destined to maintain the results of these requests in a memory for a posterior use in semantic cache. For of a good management of the cache, this paper proposes a comparative study between two methods of design and organization of the semantic cache, the one-region and multi-region. To improve the query performance, we have developed a semantic cache system on AXMed mediator, which applies innovative query analysis and rewriting techniques to answer client's queries using one-region method. Extensive experimental results on the size of memory and time response illustrate the performance improvement achieved by one-region over the multi-region method for a variety of situations.

Keywords — Cache, semantic Cache, request, restriction, XML, Mediator, Wrapper, attributes, Pages, AXMed.

I. INTRODUCTION

THE resources number available on the Web does not cease increasing, but heterogeneity and autonomy make very difficult the exploitation and the handling of these resources in an integrated environment. The information system mediators play a very important role to solve this problem by transparent collection and data integration stem from geographically distant resources, exploiting the network band-width [22]-[23]. Indeed, several studies were carried out on the data integration allow an integration of heterogeneous resources; but the optimization of the time response and the consumption of the network web resources still remain problems which influence on the performances of mediation systems. These problems can be resolved by introducing the cache mechanism [2]-[6].

There are several types of caches: cache by duplication, cache by identifier and semantic cache [1]. The Cache in mediation systems is characterized by the complexity of the determination methodology of requests restriction. To contribute to the solution of these problems, this article presents a technique for managing the semantic cache mechanism based on one-region method. Note that the

current version of AXMed Server is not equipped with a system for the cache management; the implementation of the cache in this system will enable us to validate our methodology [16]-[19].

II. THE SEMANTIC CACHE

The semantic cache is qualified as intelligent, because it exploits the semantics of the requests previously formulated to evaluate if a request should (entirely, partially or not) be sent on the network. A semantic cache is constituted essentially of a requests base containing all the requests, posed by the customers, in their canonical format; and a results base, which take the form of data warehouse, constituted by a set of node identifiers answering the requests which are contained in the requests base [7]-[9].

When a mediator, using a semantic cache mechanism, receives the customer request, this one is analyzed by the requests cache analyser engine, and then it is divided into two parts:

1. A local sub-request: which recovers the part of the answers available in the cache (local part) noted Locale (Q).
2. A complementary request: allowing to recover the missing data, from the distant resources, it is noted Rest (Q).

III. ORGANIZATION METHODS OF THE SEMANTIC CACHE

Mediator systems impose new methodology for the cache management and the data storage compared to traditional systems [1]. The cache data are divided into two parts: a history base which records all the requests predicates posed by the customers with their interrogation dates; and a results base which contains the requests answers, namely the data base. The design of the organization of the semantic cache in mediation systems can be modelled in two different ways [10]-[12]:

A. Organization Method « multi -region ».

For each source integrated by the mediation system we associate a storage space on the system level. This space contains several disjointed areas, each area consists of two entities: the first records the requests predicate and the second stores the execution result.

The study of the mediators working with such a method permits us to release a set of fundamental remarks:

1. For each new request, after the request rewriting phase, the cache manager searches the data recorded in

M. Ezziyyani, Abdelmalek Essaadi University, Tangier, Morocco (phone: 21264433918; fax: 21239393953; e-mail: ezziyyani@ieee.org).

M. Bennouna, Abdelmalek Essaadi University, Tanger, Morocco (Phone: 21261209409; Fax: 21239393953; e-mail: president@uae.ma)

M. Essaaidi, Abdelmalek Essaadi University, Tetuan, Morocco (Phone: 21261725992; Fax: 21239994500; e-mail: essaaidi@ieee.org)

the local cache for each sub-request. Such a research becomes very expansive for the data selection when the size of the cache is important. Thus, this method degrades the performance and the quality of the mediator in the case when the integrated resources number is high.

2. Several areas (local) can contain common data, in this case the cache will store the same results coming from various sources; witch requires an optional management mechanism for the data redundancy.
3. Difficulty for validating the data model: the cache update is an essential operation which aims to maintain the cache operational. It allows the rebuilding of the results and the predicates bases. This operation becomes very complicated with the increase of distant sources number, which complicates the decidability algorithm of the request content capacity.
4. Non extensibility: the integration of a new resource in the mediator requires the installation of a specific cache on the wrapper level to take into account a new resource.

B. Organization method « one-region ».

Throughout this article we will defend a new organization method of the cache areas to face the limitations of the previous method; we have fixed the following objectives:

1. Better data availability: the regrouping of data coming from the same source in the same region allows better availability of these data.
2. The same request rewriting algorithm for the interrogation of the resources can be used to ask the cache.
3. Data representing the same results can be stored in the same area.
4. Facilitation of the control of data security: the same source areas being gathered in the same entity, this makes it possible to apply the same rights of access to the sources.

To achieve these goals, we propose another method to manage the semantic cache; this one deploys a set of independent areas associated with the integrated sources. The results sent by the wrappers to the mediator are merged and stored in the same area related to the customer request, forming, thus, a unique area.

IV. COST ESTIMATION AND STUDY OF THE CACHE PERFORMANCE.

In this part we are interested in the impact of the cache management method on the mediation system performance, and in estimating of requests treatment cost when using the semantic cache.

The cache allows reducing the request execution time in two different ways. In the first, during the phase of optimization: the sub requests, already present in the cache, will not be optimized, allowing reducing the optimization time. In the second way, during the execution phase: the sub requests which are entirely or partly in the cache, will not be treated by the mediator and wrappers (or treated partially). So, the reduction of the execution time, by reducing the wrappers and mediator cost treatment and

the communications with distance sources.

A. Response time.

Among the criteria taken into account when calculating the cost in a mediation system we find: the communication cost, the treatment cost of the mediator, the cost of the various treatments on the wrappers and the cost of the network congestion.

$$\text{Total_cost} = \text{Communication_cost} + \text{Mediator_cost} + \text{Cache_cost} + \text{Wrapper_cost}$$

Thereafter, we will be interested in the various situations of the cache to carry out a comparative study between the two methods (with one-region and multi-region) in order to estimate the cost of the cache.

In the first case, the cache contains a total answer to the customer request; in this case the cost of the cache can be represented by:

- One-region :
 $\text{Cache_cost} = \text{Analysis_time} + \text{Interrogation_time}$

- Multi-region :
 $\text{Cache_cost} = \text{Analysis_time} + \text{Interrogation_time} + \text{Fragmentation_time} + \text{Reconstruction_result_time}$.

In the second case, the cache contains a partial answer, or doesn't contain any response, to the customer request; the cache cost can be represented by:

- One-region:
 $\text{Cache_cost} = \text{Analysis_time} + \text{Locale_interrogation_time} * \text{Distance_Interrogation_time}$.

- Multi-region:
 $\text{Cache_cost} = \text{Analysis_time} + \text{fragmentation_time} + \text{Locale_interrogation_time} * \text{Distance_Interrogation_time} + \text{Reconstruction_result_time}$.

The *Locale_interrogation_time* appears only in the case where the cache contains a partial answer of the request.

- Analysis and conclusion:

Since a preliminary decomposition of the user request is made by the mediator, the cache in the case *one_region*, exploits this decomposition (as well as the reconstitution of the result), contrary to the cache multi-regions, which requires a fragmentation, and an additional reconstitution, in this case, several regions are associated to each local source. Thus, the semantic cache *one_region* spends less time to answer the customer requests, because there is no need for fragmentation and reconstitution of the result.

B. The size of the cache.

Another factor which has to be taken into account is the consumption of the disk space for the data storage in the cache. The storage space depends on the organisation method, on the type of the results, and on the applicability domain of the distant sources. Let us consider the following parameters:

resp_ave_size: the response average size of the distant source to a customer request.

source_numbr: the number of distant sources.

resp_size(R_i): average size of the response on the R_i request (it is supposed that request R_i is a characteristic request).

resp_dup_size: average size of the response duplicated in the cache.

P: probability to have a new answer, from distant source, which is not registered in the cache. We suppose that it is the same for all sources (for simplicity).

• Multi_ region method

$$\text{resp_size}(R_1) = \text{resp_ave_size} * \text{source_numbr} * P_1$$

$$\text{resp_size}(R_2) = \text{resp_ave_size} * \text{source_numbr} * P_2$$

$$\text{resp_size}(R_n) = \text{resp_ave_size} * \text{source_numbr} * P_n$$

So the total cache size after n request becomes:

$$\text{Tot_size} = \text{resp_ave_size} * \text{source_numbr} * \sum P_i$$

We suppose that every new request is put for the first time, and that the answer of this request is generated by all the considered sources, P_i become:

$$P_i = \frac{m-i}{m}$$

Where **m** is the number of record stored in the considered source, the Tot_size after **n** request becomes:

$$\text{Tot_size} = \text{resp_ave_size} * \text{source_numbr} * \sum_{i=1}^n \frac{m-i}{m}$$

• One region method :

Since all the answers coming from different sources are stored in one region; another factor intervenes in this case: the duplication factor D.

$$D = \frac{\text{source_number} * \text{counting the same result on request } R_{n-1}}{\text{total_source_number}}$$

The total cache size after n request in this case is:

$$\text{Tot_size}(n) = \text{resp_ave_size} * \text{source_numbr} * \sum_{i=1}^n \frac{m-i}{m} * (1-D), (D < 1).$$

Analysis and conclusion

In the first experiment, we have fixed the two parameters: the factor of duplication D and the average number of the recordings M and we have varied the number of requests N, we will obtain the following graph:

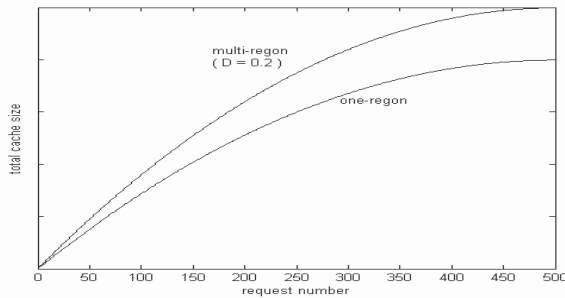


Fig.1 Comparison the cache sizes between One-region and Multi-region (For D=0.2).

According to the figure (Fig.1), the size of the cache multi-region is always higher than that of one-region and the difference becomes important with the increase in the

number of requests. That means that the duplication of data recorded in the regions of the cache Multi-region (according to the factor of duplication) influence on the size of cache in comparison with the One-region cache.

In the second test one can follow the effect of the variation of the duplication factor D on the size of the cache. According to the graph of the figure (Fig.2) one notices that if the duplication factor D decreases the size of the cache increases. That shows that size of the Multi-region cache (when D=0) is always higher than all the possible sizes of the one-region cache according to various values' of the duplication factor D.

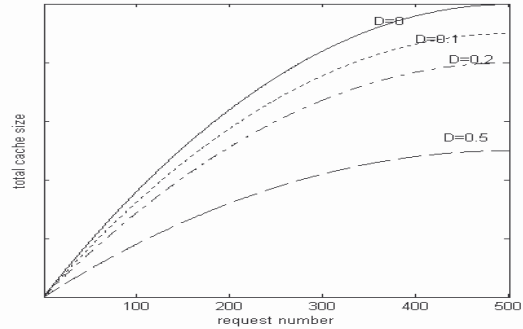


Fig.2: Influence of the duplication factor D on the cache size.

According to the two formulas of the cache size we can conclude that the size of the cache one-region is lower than the one in multi-region case in the various possible situations. But the importance of the difference depends on the duplication factor, the result average size and the number of distant sources.

When we use a cache mechanism, the communication rate and the data exchange decrease on a certain level of interrogation, consequently the cost estimate take into account other additive parameters in addition to the basic parameters studied previously.

IV ONE-REGION SEMANTIC CACHE SERVER STRUCTURE.

To face the problems previously cited, we propose a new architecture for the management of the semantic cache; it is based on the organization of the cache on the mediator level instead of the wrappers one. In this architecture the cache server manages only one data integrated source composed of two different native sources storing the predicates of all the customer requests, and the results coming from the wrappers. The architecture of the cache server is based on six basic modules:

1. Analyzer of the customer requests: It is the core of the cache server; it manages the determination of the requests restriction.
2. Distant data search engine: this software layer allows the interrogation of the wrappers by complementary requests.
3. Safety, history and access to the data control: it is a complementary module which ensures the security access and the exploitation of the data recorded in the cache.
4. The Requests base manager: it allows the update of the requests base according to the algorithm predefined.

5. The results base Manager: it ensures coherence between the requests base and the cache's data.
6. Results Presenter to the customer: it intervenes in the final phase to present the result of interrogation to the customer.

For storage, our approach uses an XML native data base, Xindice [13]-[15]. This choice is justified by several important factors which can be summarized by the fact that the data can be stored and extracted in their initial forms as soon as possible, moreover; it is well adapted to the exchange model of the mediator use[20]-[22].

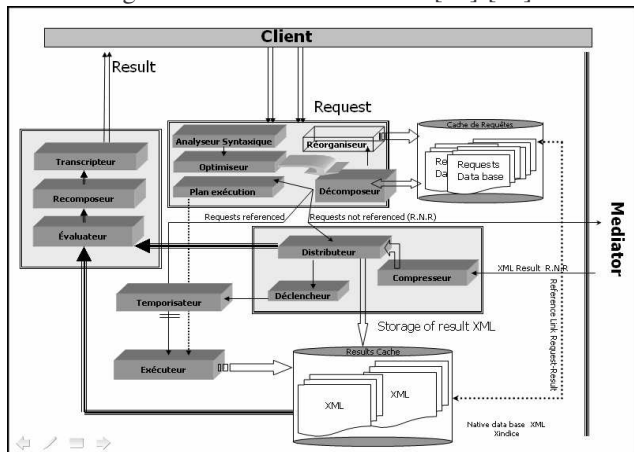


Fig.3: One region semantic cache architecture

V. CONCLUSION.

We have presented in this paper a cache management method for the semantic cache for mediator systems; this method is based on the organisation of the cache in one global region, aiming to better data availability and preservation of networks resources. A One-region semantic cache server structure was presented, it is based on the organization of the cache on the mediator level instead of the wrappers one. The impact of the cache management method on the mediation system performance was carried out using cost estimation. Our results show that the use of one-region cache solution can reduce considerably the execution time and the cache size.

REFERENCES

- [1] B. Chidlovskii, C. Roncancio and M.L. Schneider, "Semantic cache mechanism for heterogeneous Web querying," *Computer Networks* 31 1999, pp. 1347-1360.
- [2] B. Chidlovskii and U. Borghoff, "Signature file methods for semantic query caching," "2nd European Conf. on Digital Libraries, LNCS 1513, Crete, Greece, September, 1998.
- [3] Q. Luo, J.F. Naughton, R. Krishnamurthy, P. Crao and Y. Li. Active, "Query Caching for Database Web Serves," *Lecture Notes en Computer Science*, 2001, 92-102.
- [4] C. Li, E.A. Rundensteiner and S. Wang, "Xcache - A semantic Caching System for XML Queries," *ACM SIGMOD'2002*, June 4-6, 2002.
- [5] S. Adali, K.S. Candan, Y. Papakonstantinou and V.S. Subrahmanian, "Query caching and optimization in distributed mediator systems," *Proc. SIGMOD'96 Conference*, 1996, pp. 137-148.
- [6] Chen Li., "Computing Complete Answers to Queries with Binding Restrictions," *Department of Computer Science, Stanford University*.
- [7] F. Afrati, C. Li and P. Mitra, "On containment of conjunctive queries with arithmetic comparisons,"

- [8] L. Haas, D. Kossman, E. Wimmers and J. Yang, "Optimizing Queries across Drivers Data Sources," *23rd Very Large Data Bases*, August 1008, Athens, Greece, 1997.
- [9] S. Dar, M.J. Franklin, B. Jonsson, D. Srivastava and M. Tan, "Semantic data caching and replacement," *Proc 22nd VLDB Conference*, Bombay, India, 1996, pp. 330-341.
- [10] Q. Luo, R. Krishnamurthy, Y. Li, P. Cao and JF Naughton, "Active Query Caching for Database Server," *Computer Science Departement, University of Wisconsin-Madison*.
- [11] P.T. Martin and J.I. Russell, "Data caching strategies for distributed full text retrieval systems, *Information Systems*," (16) (1) (1991)1 - 11.
- [12] L. Manolescu, D. Florescu and D. Kossman, "Answering XML Queries over Heterogeneous Data Sources," *27th Very Large Data Bases*, Roma, Italy, Sept, 1994, pp. 241-2560.
- [13] J. Shanmugasundaram, J. Kiernan, E. Shekita, C. Fan and J. Funderburk, , "Querying XML Views of relational Data, *27th Very Large Data Bases*," Roma, Italy, Sept 2001, pp. 241-2560.
- [14] A.Y. Levi and A. Rajaraman and J.J. Ordille, "Querying heterogeneous information sources using source descriptions," *Proc. 22nd VLDB Conference*, Bombay, India, 1996, pp. 251-262.
- [15] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland and Y. Papakonstantinou, J. Ullman, J. Widom, "The TSIMMIS Projet : Integration of Heterogeneous Information Sources," *IPSIJ Conference*, Tokyo, Japan, Octobre, 1994, pp. 7-187.
- [16] M. Ezziyyani, M. Bennouna and M. Saaidi, "Optimisation de cache sémantique pour les médiateurs des systèmes hétérogènes," *CIRO'2005*, Marakkech.- Morocco, Vol 4 , Mai 2005, pp. 69-97.
- [17] M. Ezziyyani, M. Bennouna and M. Saaidi, "Information Systems Integration through Web Services Technologies," *Information and Communication Technologies International Symposium, ICTIS'2005*, Vol 1, Tetuan - Morocco, Juin 2005, pp. 454-457.
- [18] M. Ezziyyani, M. Bennouna and M. Saaidi, "Advanced XML Mediator For Heterogeneous Information Systems," *Information and Communication Technologies International Symposium, ICTIS'2005*, Vol 1, Tetuan - Morocco, Juin 2005, pp. 64-71.
- [19] G. Gardarin, A. Mensch, T. Dang-Ngoc, L. Smit, "Integrating Heterogeneous Data Sources with XML and XQuery. e-XMLMedia,"
- [20] D. Chamberlin, D. Florescu, J. Robie, J. Siméon, M. Stefanescu, "XQuery : A Query Language for XML," *W3C*, 2004.
- [21] A. Serge, B. Omar, M. Ioana, M. Tova, R. W., "Actice XML : Peer-to-Peer Data and Web services Integration," *Proceeding of the 28th VLDB Conference*, Hong Kong, China, 2002.