ROLES OF AGENTS IN DATA-INTENSIVE WEB SITES

HENDA BEN GHEZALA
henda.bg@cck.rnu.tn

ABDELAZIZ ABDELLATIF
abdelaziz.abdellatif@fst.rnu.tn

ALI BEN AMMAR
ali.benammar@isd.rnu.tn

Ecole Nationale des Sciences
Faculté des Sciences de Tunis
Institut Supérieur de Documentation de l'Informatique

Abstract: The Data-Intensive Web Sites provide access to a large number of Web pages whose content is dynamically extracted from structured databases. They may be used for shopping or paying in e-commerce, for reading news in a newspaper site or to consult digital library. In this context, users often need rich content and fresh data. Several techniques have been developed to meet the demand for faster and more efficient access to the DIWS. Among them a main role is acquired by the replication, the caching, the materialization, and the refreshing of data. Software agents have proved to be a good tool that may give a high performance results on the Web. In this paper, we address their cases of use in the DIWS. We discuss cases in which agents can be used to improve the data management performance. The aim is to specify tasks that may profit from the increase development in agent technologies.

Keywords: Data-Intensive Web Sites, Agents, Optimization, Performance, Data management.

Introduction

The Data-Intensive Web Sites (DIWS) provide access to a large number of Web pages whose content is dynamically extracted from structured databases. They serve to integrate and summarize Web services that may be distributed and heterogeneous. They may be used for shopping or paying in e-commerce, for reading news in a newspaper site or to consult digital library. Their source databases are generally distributed, heterogeneous, and with dynamic content. The user queries are, sometimes, personalized that is they are tailored to the style and the needs of each individual. They often demand rich content and fresh data. In this context, data management will be highly complex. It consists in integrating, updating and rapidly accessing data.

Several techniques have been developed to meet the demand for faster and more efficient access to the DIWS. Among them a main role is acquired by the replication, the caching, the materialization, and the refreshing of data. Software agents are demonstrated to be a good tool that may give high performance results in such environment [40,48,51,16]. An agent is a software entity, situated in an environment, where it acts autonomously and flexibly to reach some objectives [39]. A multi-agent system is a distributed system composed of a set of collaborative agents. It is used to perform distributed tasks.

In this paper, we address the use of agents in the DIWS. We discus the cases in which agents can be used to improve the data management performance. The aim is to specify tasks that may profit from the increase development in agent technologies. Our discussion is based on the study of some works involving agents to resolve data management problems.

1.The paper is organized as follows. Section 2 presents some related works. Section 3 presents the concept of DIWS. Section 4 describes in more detail the concept of agent. In section 5 we discuss the use of agents in some tasks of data management in DIWS. Section 6 concludes.

2. Related works

Recently, there has been a lot of interest in the use of software agents in several domains. The main applications in which intelligent agents can be involved are identified in [37]. A framework to integrate agents into the use of the World Wide Web is designed and implemented in [16]. In this framework, the agents filter information, initiate communication, monitor events, and perform tasks. The aim is to improve the usability and usefulness of the World Wide Web. [48] Surveys several agent-mediated e-commerce systems and [41] presents an example of such systems that automate the hotel

reservation in tourism domain. In this paper we will limit our concern to the DIWS domain. By studying these related works and others we will identify the data management tasks which need software agents to be optimized.

3. Data-Intensive Web Sites (DIWS)

Data-Intensive Web Sites (DIWS) provide access to a large number of Web pages whose content is dynamically extracted from structured databases [28]. Today, they become necessary for allowing some e-commerce tasks or to access dynamic information. Their architecture includes a database management system (DBMS) layer, a site server layer and the client. Thus, a new kind of pages, dynamically generated, and a new architecture were born. We have no more the traditional couple of a Web client and a Web server, but a third part is added, the application program, running on the Web server to receive the request from the client, retrieve the relevant data from the database and then pack them into HTML or XML format. Newspaper sites and shopping ones are examples of such architecture. Several tools and approaches for developing such systems are presented in [43]. For each kind of application, a set of web development tools are specified. The performance problem of DIWS lies in addressing the latency reduction of page produced by the site and the quality of data presented to the clients. Firstly, because returning Web page may require costly interaction with the database system. So, the net effect of this situation is network congestion, high client perceived latency, Web server overload and slow response times for Web severs. Secondly, because the quality of data is of crucial importance, especially for applications that must always serve fresh data (e.g. providers of stock prices, sports scores).

Recently, much research has been devoted to improving Web performance by reducing latency and bandwidth consumption, and increasing server scalability and availability. Proposed solutions include predictive prefetching, caching and materialization of Web objects, and architecting network and Web servers for scalability and availability. These solutions are beneficial but need to be yet improved to accommodate the continuously growing number of web users and services. In section 5 we will discuss the possibility of using agents to improve these solutions.

4. Agents

According to [51], an agent is computational entity which:

- Acts on behalf of others entities in an autonomous fashion;
- Performs its actions with some level of proactivity and/or reactiveness ;
- Exhibits some level of the key attributes of learning, co-operation and mobility.

Software agents are, today, more important because:

- More and more every-day tasks are computer-based;
- The world is in a midst of an information revolution, resulting in vast amount of dynamic and unstructured information;
- Increasingly, more users are untrained;
- And therefore users require agents to assist them in order to understand the technically complex world we are in the process of creating.

According to [51], a mobile agent is a software entity which exists in a software environment. It inherits some of the characteristics of an agent. A mobile agent must contain all of the following models: an agent model, a life-cycle model, a computational model, a security model, a communication model and finally a navigation model.

According to [51], a multi-agent system is able:

- To solve problems that are too large for a centralised single agent to deal with due to resource limitations or the sheer risk of having one centralised system;
- To allow for the interconnecting and interoperation of multiple existing legacy system, e.g. expert systems, decision support systems;
- To provide solutions which draw from distributed information sources;
- To provide solutions where the expertise is distributed, e.g., in health care provisioning;
- To enhance speed (if communication is kept minimal), reliability (capability to recover from the failure of individual components with graceful degradation performance), extensibility (capability to alter the number of processors applied to a problem), the ability to tolerate uncertain data and knowledge;
- To offer conceptual clarity and simplicity of design.

In section 5 we will give examples that illustrate these abilities. Then, we will deduce where and how agents can be used in DIWS to make profit from their abilities.

5. Improving DIWS performance by using agents

As we have seen above, several solutions have been developed to improve web performance. In major

cases, these solutions are still valid for DIWS. We may classify them into three groups: data integration solutions, data update solutions, and data access optimization solutions. In the rest of this section we will see how these solutions can be improved by using agents.

5.1 Data integration

To construct a client web page in DIWS environment, data should be extracted from different sources and then integrated. The integration needs metadata that describe the data semantic and the mapping approach from database to Web page. Several techniques of semantic Web like ontology [60] have been used to perform the integration. So, the integration needs three main tasks: searching metadata, constructing and updating the mapping approach, and composing the Web page to the client.

Software agents have been used in several domains needing integration [13,15,33,38,49,51]. The main agent types that we see more relevant for the integration are:

- Wrapper agents that convert the source information and react to source changes ;
- Integrator agents that manage global data view, transform and subdivide queries, integrate and formulate responses.

Since source data and user queries are high dynamic in a DIWS, these two agent types may be more adequate to optimize the integration process. For complex queries asking replicated and distributed data, integration agents will greatly decrease their response time.

5.2 Data update

Data update may concern the modification of source data, the refreshment of data copies, or the refreshment of metadata in an integrator. This subsection deals with only the refreshment of data copies since in DIWS environments the data sources are, in general, managed by their owners and that the metadata refreshment is evoked here above. Data copies mean the data extracted from a source to be integrated in a Web page which is materialized or cached in a Web proxy or in a Web server. There are many works addressing Web caching data [1,11,14,25,27,28,45,46] and Webview materialization [2,4,5,7,9,17]. A key requirement for DIWS that provide dynamic data is to keep data copies up-to-date that is fresh and consistent with their original sources. The freshness of data [31] depends, in general, on the client tolerance and on data access frequency.

On the Web, there are many techniques to refresh derived data [3,6,24,30,35,42] but there is less use of agents. This may be because there are database tools, like triggers, that can perform such task. In [35], there is an attempt to use agents for capturing source updates. Agents are used in [22,57] to capture user needs and preferences which may lead to deducing user tolerance. i.e. accepting a data that is not refreshed along a period of time t, means that the user is satisfy if the data age is less than t. So, in DIWS, agents may be used for two updating tasks: capturing source changes and specifying data update frequencies based on user tolerance and access frequencies.

5.3 Data access optimization

A family of optimization techniques is developed to reduce the query response time in DIWS environment. It includes distribution, data caching and data materialization.

5.3.1 Distribution

With the increase in traffic on the web, popular Web sites get a large number of requests. Servers at these sites are sometimes unable to handle the large number of requests and clients to such sites experience long delays. One approach to overcome this problem is the distribution or replication of content over multiple servers. This approach is called Content Distribution Networks (CDN). It allows for client requests to be distributed over multiple servers. Several techniques have been suggested to distribute or replicate content over Web servers [18,29,34,36,52,53,54,61,64], and to direct client requests to multiple severs [12,19,26,32,55,56].

Based on their intelligence and learning capability, software agents can optimize query response time essentially in distributed environment. They are used to collect information on the Web [10,23,33,58,63]. In the DIWS context, query processing may be distributed over several mobile or source localised agents. Agents may also be used to dynamically searching the optimal processing plan for query in a given situation (server overload, data distribution,…).

5.3.2 Caching

DIWS satisfy, in most case, dynamic requests. The overhead for satisfying this kind of requests may be orders of magnitude more than the overhead for satisfying static requests. Dynamic requests often involve extensive back-end processing and invoke several database accesses. In order to reduce the overhead of generating dynamic data, it is often feasible to generate the data corresponding to a dynamic page once, store the page in a cache, and to serve subsequent requests to the page form cache

instead of invoking the server program again. However there are types of dynamic data that cannot be pre-computed and serviced from the cache. For example a personalized Web page that contains content specific to a client, such as the client's name, should not be cached.

The issues pertaining to the cache management are cache consistency and cache replacement. The purpose of a cache consistency mechanism is to ensure that cached data are essentially updated to reflect the changes to the original data. While, the purpose of a cache replacement mechanism is to decide which data will enter the cache when a client requests them and which data already in cache will be purged out in order to make space for the incoming data when the available space is not sufficient. The first issue is a data updating problem that is evoked in the subsection 5.2. We will now address the role of agents in the cache replacement problem and query processing.

Several replacement algorithms have been developed in literature [1,8,11,14,27,44]. They try to keep in cache the most valuable data. The value of datum is usually a function of several parameters, say access frequency, size, retrieval cost, frequency of update etc…. In [11], the authors propose to use fragments to allow partial caching of dynamic pages. Common information that needs to be included on multiple Web pages can be created as a fragment. In order to change the information on all pages, only the fragments need to be changed. In this context agents may be used to search common fragments and then to identify the appropriate ones that should be cached.

Intelligent agents may be used also to prefetching Web pages that will be probably highly accessed in the next period. That is they will prevent the cache content before receiving queries. Agents may be used also to transform some cached data in order to satisfy incoming queries that don't have cached solutions. For this reason, agents should analyze the queries and profit from their experience to provide good responses.

Web data may be cached in several nodes of the network (DBMS, Web server, Proxy,…). In this case, agents may be distributed over the different nodes to manage caches. Their role will be to negotiate the relevant data to be cached, when to place data, how to compose the query responses from the distributed fragments.

5.3.3 Materialization

Similarly to traditional database views, the term Webviews is used on the web to mean Web pages that are automatically constructed from base data using a program or a DBMS query. The materialization approach consists in computing Webviews and storing them. Having a Webview materialized can potentially give significantly lower query response times, provided that the update workload is not heavy. Even if the Webview computation is not very expensive, by keeping it materialized we eliminate the latency of going to the DBMS every time which could lead to DBMS overloading.

According to [5], Webview materialization is different from Web caching: Webview materialization aims at eliminating the processing time needed for repeated generation, whereas Web caching strives to eliminate unnecessary data transmissions across the network.

The Webview materialization approach is similar to that of view materialization in a data warehouse [20, 21,47]. The main issues of the Webview materialization approach are: how to select dynamically the appropriate Webviews to be materialized, how to refresh materialized Webviews and how to distribute the storage of Webviews over several servers.

Here, agents may be used in the first task to search the needed information and parameters like the access frequency, the update frequency, the estimated size of Webviews; to decide which Webviews to materialize in a given situation (reserved space, overload constraints,…). The role of agents in the two other tasks will be as it is described in sections 5.3.1 and 5.2. In the query processing context, agents can reformulate query to be satisfied from the materialized Webviews or redirect query to the appropriate server having the responsive Webviews.

6. Conclusion

Today, software agents are frequently used on the Web to optimize several data management tasks. In this paper, we have addressed their role in DIWS. After describing the concepts of DIWS and agents, we have identified the main tasks of DIWS, in which agents can be involved. From the study of some applications of agents on the Web, we have concluded that, in a DIWS environment, software agents can enhance the performance of other techniques developed to perform three main functions: data integration, data update and data access optimization. In the future work we will, develop a multi- agent system that dynamically selects the appropriate Webviews to be materialized.

References:

[1] A. Iyengar, D. Rosu. Architecting Web sites for high performance. Scientific Programming 10(1): 75-89.2002

[2] A. Labrinidis, N. Roussopoulos. Adaptive WebView Materialization. WebDB'01: 85-90. 2001

[3] A. Labrinidis, N. Roussopoulos. Balancing Performance and Data Freshness in Web Database Servers. VLDB'03: 393-404. 2003

[4] A. Labrinidis, N. Roussopoulos. Online View Selection for the Web. 2002. http://citeseer.ist.psu.edu/cache/papers/cs/25933/http:zSzzSzwww.cs.umd.eduzSzLibraryzSzTRs zSzCS-TR-4343zSzCS-TR-4343.pdf/labrinidis02online.pdf

[5] A. Labrinidis, N. Roussopoulos. On the Materialization of WebViews. In ACM SIGMOD Workshop on the Web and Databases (WebDB '99): 79-84. 1999.

[6] A. Labrinidis, N. Roussopoulos. Update Propagation Strategies for Improving the Quality of Data on the Web. VLDB'01: 391-400. 2001

[7] A. Labrinidis , N. Roussopoulos. WebView Materialization. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. pp.367-378. May 15-18, 2000, Dallas, Texas, United States

[8] A. Labrinidis. Web-Aware Database Servers – I. 2002 http://www.cs.pitt.edu/~labrinid/courses/cs2001/webdb-12nov2002.pdf

[9] B. Christos , K. Agisilaos. Efficient Materialization of Dynamic Web Data to Improve Web Performance.15th International Conference on Computer Communication (ICCC 2002) Mumbai.India, August 11-14, 2002

[10] C. Edgar, G. Susan. Intelligent Information Agents for the World Wide Web. Technical Report ITTC-FY97- TR-11100-1, Information and Telecommunication Technology Center, The University of Kansas, Lawrence, KS, May 1997.

[11] C. Mohan. Caching Technologies for Web Applications. In Proceedings of the 2001 VLDB Conference, Roma, Italy, September 2001.

[12] D. A. Johnson, G. C. Shoja. Request Routing Architecture in Content delivery Networks. Proceedings of 2003 International Conference on Internet Computing, June 23-26, 2003, Las Vegas, USA.

[13] D. Gilbert, M. Aparicio, B. Atkinson, S. Brady , J. Ciccarino, B. Grosof , O'Connor, P., Osisek, D., Pritko, S., Spagna, R. & Wilson L. The role of intelligent agents in the information infrastructure. IBM Report.1995.

[14] D. Katsaros , Y. Manolopoulos. Cache management for Web-powered databases. In Web-Powered Databases, pp 201-242. IDEA Group Publishing. 2002.

[15] F. Scilla, M.N. Huhns. Making Agents Secure on the Semantic Web. IEEE Internet Computing, pp 76-93, Nov/Dec 2002.

[16] G. Fischer, C.G. Thomas. Using Agents to improve the Usability and Usefulness of the World- Wide Web. Proceedings UM-96, Fifth International Conference on User Modeling, Hawaii, User Modeling, Inc. 1996, 5-12.

[17] G. Mecca, A. O. Mendelzon, P. Merialdo. Efficient Queries over Web Views. EDBT '98: 72-8. 1998

[18] G. Pierre , M. van Steen , A. S. Tannenbaum. Dynamically Selecting Optimal Distribution Strategies for Web Documents. IEEE Transactions on Computers, v.51 n.6, pp 637-651, June 2002

[19] H. Kabir, E. G. Manning, G. C. Shoja. Request-Routing Trends and Techniques in Content Distribution Network. Proc ICCIT 02, Dhaka, Bangladesh, pp 315-320. December 2002. ISBN 984-32-0450-6.

[20] H. Gupta and I.S. Mumick. Selection of Views to Materialize Under a Maintenance-Time Constraint. ICDT. 1999.

[21] H. Gupta. Selection of Views to Materialize in a Data Warehouse. ICDT. 1997.

[22] H. Liebermann. Letizia: An agent that assists Web browsing. In Proc. Intl. Conf. on AI, Montréal, Canada, August 1995.

[23] H. Lieberman, N. van Dyke, A. Vivacqua. Let's browse: a collaborative Web browsing agent. In Proc. Intl. Conf. on Intelligent User Interfaces, January 1999.

[24] I. Ari, E. L. Miller. Caching support for push-pull data dissemination using data snooping routers. In Proceedings of the 10th International Conference on Parallel and Distributed Systems (ICPADS'04). 2004

[25] J. Wang. A Survey of Web Caching Schemes for the Internet. ACM Computer Communication Review 29(5). 1999.

[26] J. Watts, S. Taylor. A Practical Approach to Dynamic Load Balancing. IEEE Trans. on Parallel and Distributed Systems, Vol. 9, NO. 3, March 1998, pp 235-248.

[27] K. Amiri, S. Park, R. Tewari, S. Padmanabhan. DBProxy: A Dynamic Data Cache for Web Applications. ICDE Conference 2003. pp 821–831.

[28] K. Yagoub, D. Florescu, V. Issarny, and P. Valduriez. Caching strategies for data intensive web sites. In Proceedings of the VLDB 2000

Conference, pp 188-199. 2000.
[29] Lei Gao, Michael Dahlin, Amol Nayate, Jiandan Zheng, Arun Iyengar. Improving Availability and Performance with Application-Specific Data Replication. IEEE Trans. Knowl. Data Eng. 17(1): 106-120. 2005
[30] M. Bhide, K.Ramamritham, And P. Shenoy. Efficiently Maintaining Stock Portfoilis Up-To-Date On The Web. IEEE Reasearch Issues In Data Engineering (RIDE'02) Workshop, March 2002.
[31] M. Bouzeghoub, V. Peralta. A Framework for Analysis of Data Freshness. IQIS 2004. pp 59-67.
[32] M. Conti, E. Gregori, F. Panzieri. Load distribution among replicated web servers: A QOS based approach.
in Second Workshop on Internet Server Performance, May 1999.
[33] M. Côté, N. Troudi. NetSA: Une Architecture Multiagent pour la Recherche sur Internet. Expertise Informatique, vol. 3(3). 1998.
[34] M. Karlsson , C. Karamanolis. Choosing Replica Placement Heuristics for Wide-Area Systems. Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04), pp 350-359, March 24-26, 2004
[35] N. Ashish, D. Kulkarni and Y. Wang. Source Update Capture in Information Agents. Proceedings of IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03), Acapulco, Mexico. August 9-10, 2003.
[36] N. A. John. A Study of Replicated and Distributed Web Content. Thesis, Departement of Computer Science, Worcester Polytechnic Institute, August 2002.
http://www.wpi.edu/Pubs/ETD/Available/etd-0810102-160719/unrestricted/john.pdf
[37] N. Jennings , M. Wooldridge. Applications of intelligent agents. Chapter 1 in Agent Technology: Foundations, Applications, and Markets. Springer, 1998.
[38] N. Gibbins, S. Harris, N. Shadbolt. Agent-based Semantic Web Services. WWW2003, May, 2003.
[39] N.R. Jennings, M. Wooldrige, K. Sycara. A Roadmap of agent research and development. Int Journal of Autonomous Agents and Multi-Agent Systems, 1(1):7-38,1998.
[40] N.R.Jennings. An Agent-based approach for building complex software systems. Communications of the ACM, Vol.44, No.4. April 2001.
[41] O. Dikenelli, N.Y. Topaloglu, C. Erdur, O. -nalir. An Agent-Mediated E-Commerce Framework For Tourism Domain.
http://www.srdc.metu.edu.tr/webpage/projects/hermesproject/documents/agent-tourism.pdf
[42] P. Deolasee, A. Katkar, A. Panchbudhe, K. Ramamritham, and P. Shenoy. Adaptive push-pull: Disseminating dynamic web data. In Proceedings of the 10th International World Wide Web Conference, pages 265-274, Hong Kong, China, May 2001.
[43] P. Fraternali. Tools and Approaches for Developing Data-Intensive Web Applications: A Survey. In ACM Computing Surveys Volume 31:3, pp 227-263. 1999.
[44] Q. Luo, J. F. Naughton, R. Krishnamurthy, P. Cao, Y. Li. Active Query Caching for Database Web Servers. WebDB 2000. pp 29-34. 2000.
[45] Q. Luo, J. Naughton. Form-based proxy caching for database-backed web sites. In Proceedings of the 2001 VLDB Conference, pages 191-200, September 2001.
[46] Q. Luo, J. Naughton, R. Krishnamurthy, P. Cao and Y. Li. Active Query Caching for Database Web Servers.WebDB, 2000.
[47] R. Chirkova, A. Halevy, D. Suciu. Formal Perspective on the View Selection Problem. In Proc. VLDB, Roma, Italy, pp 59-68. 2001.
[48] R.Guttman, A.Moukas, P.Maes. Agent-mediated electronic commerce: A survey. Knowledge Engineering
Review, 13(2):147-159, 1998.
[49] R. J. Bayardo, W. Bohrer, R. Brice, A. Cichocki, J. Fowler, A. Helal, V. Kashyap, T. Ksiezyk, G. Martin, M. Nodine, M. Rashid, M. Rusinkiewicz, R. Shea, C. Unnikrishnan, A. Unruh, D. Woelk. Infosleuth: agent-based semantic integration of information in open and dynamic environments. In Huhns M. N. et Singh M. P., éditeurs, Reading in Agents, pages 205–216, SF, CA, 1998. Morgan Kaufmann.
[50] S. Bergamaschi___ , G. Cabri_ , F. Guerra_ , L. Leonardi_ , M. Vincini_ , F. Zambonelli_. Mobile Agents for
Information Integration. 2001.
http://dbgroup.unimo.it/Miks/publications/papers/cia2001.pdf
[51] S. Green, L. Hurst, B. Nangle, P. Cunningham, F. Somers, R. Evans. Software Agents: A Review. Intelligent
Agents Group (IAG) report TCD-CS-1997-06, Trinity College Dublin, May 1997.
[52] S. Sivasubramanian, M. Szymaniak, G. Pierre, M. van Steen. Web Replica Hosting Systems Design. Internal Report IR-CS-001, June 14, 2004. http://www.cs.vu.nl/pub/papers/globe/IR-CS-001.03.pdf
[53] T. Rotaru , H. N‹geli. Dynamic load balancing by diffusion in heterogeneous systems. Journal of Parallel and Distributed Computing, v.64 n.4, p.481-497, April 2004
[54] V. Cardellini , M. Colajanni , P. S. Yu.

Dynamic Load Balancing on Web-Server Systems. IEEE Internet Computing, v.3 n.3, p.28-39, May 1999

[55] V.Cardellini , M. Colajanni , P. S. Yu. Request Redirection Algorithms for Distributed Web Systems. IEEE Transactions on Parallel and Distributed Systems, v.14 n.4, p.355-368, April 2003

[56] V. Ghini, F. Panzieri, M. Roccetti. Client-centered Load Distribution: A Mechanism for Constructing Responsive Web Services. Proceedings of the 34th Hawaii International Conference on System Sciences – 2001.

[57] V. Robles, E. Menasalvas, S. Millڻn, M. Pérez, E. Hochsztain, O.Marbڻn, J. Peڤa, A. Tasistro. Beyond user clicks: an algorithm and an agent-based architecture to discover user behaviour. Proceedings of the ECML/PKDD 22-26 September 2003. Cavtat. Dubrovnik, Croatia.

[58] W. Jirapanthong, T. T. Sunetnanta. An XML-Based Multi-Agents Model for Information Retrieval on WWW. In Proceedings of the 4th National Computer Science and Engineering Conference (NCSEC2000), Queen Sirikit National Convention Center, Bangkok, Thailand, November 16-17 .2000.

[59] W. Li, W. Hsiung, D. V. Kalashnikov, R. Sion, O. Po, D. Agrawal, K. S. Candan. Issues and Evaluations of Caching Solutions for Web Application Acceleration. In Proceedings of the 28th Very Large Data Bases Conference. Hong Kong, China, August 2002.

[60] X. Zhu, S. Gauch, L. Gerhard, N. Kral, A. Pretschner. Ontology-Based Web Site Mapping for Information Exploration. Proc. 8th Intl. Conf. on Information and Knowledge Management (CIKM '99), pp. 188-194, Kansas City, MO. November 1999.

[61] Y. Chen, L. Qiu, W. Chen, L. Nguyen, R. H. Katz. Efficient and adaptive web replication using content clustering. IEEE J. Sel. Areas Commun. 21, 6 (Aug.), 979--994. 2003.

[62] Y. Kotidis , N. Roussopoulos. DynaMat: a dynamic view management system for data warehouses. ACM SIGMOD Record, v.28 n.2, p.371-382, June 1999.

[63] Y. Xirouhakis, G.Votsis, K.Karpouzis, S.Kollias. Efficient Browsing in Multimedia Databases using Intelligent Agents and Content-Based Retrieval Schemes. IEEE International Workshop on Multimedia Signal Processing (MMSP'98), Los Angeles, California, USA.1998.

[64] Y. Zhang, B. Krishnamurthy, C. Wills. On the Use and Performance of Content Distribution Networks .In ACM SIGCOMM Internet Measurement Workshop. 2001