CIGMA: aCtive Inventory service in Global e-MArket based on efficient catalog management

Su Myeon Kim, Kyungmin Cho, Inseok Hwang, Seungwoo Kang, Heung-Kyu Lee

Department of Computer Science
Korea Advanced Institute of Science and Technology
373-1 Kusung-Dong, Yoosung-Ku,
Taejon, 305-701, Korea

Junehwa Song

IBM T. J. Watson Research Center
P.O. Box 218, Yorktown Heights, N.Y.,10598, USA

ABSTRACT

A fully Internet-connected business environment is subject to frequent changes. To ordinary customers, online shopping under such a dynamic environment can be frustrating. We propose a new E-commerce service called the CIGMA to support online customers under such an environment. The CIGMA provides catalog comparison and purchase mediation services over multiple shopping sites for ordinary online customers. The service is based on up-to-date information by reflecting the frequent changes in catalog information in real-time. It also matches the desire of online customers for fast response. This paper describes the CIGMA along with an overview of its catalog management schemes.

Keywords

e-commerce, catalog comparison, dynamic content, cache, real-time update

1. INTRODUCTION

The Internet is significantly altering the way business is conducted, opening up many challenges and opportunities to both merchants and customers. One of the notable characteristics of a fully Internet-connected business world is that many aspects of a business environment will be subject to frequent changes. Merchants are actively adapting to such environments. However, to ordinary customers, these dynamics are a serious source of inconvenience; they do not feel comfortable shopping in a market where the status of an item, *e.g.* inventory, delivery condition, price, etc., may change several times a day. Customers will want services to overcome such difficulties and to get their share of benefits. However, existing service models and systems do not effectively reflect such environments and support ordinary customers.

Let's consider the source of such dynamics from the ordinary customers' point of view to discover potential inconvenience in on-line shopping. First, information on sales items may undergo constant changes. The case of Cisco presents a good example. By effectively adapting their business processes to the Internet environment, Cisco has shortened the term of their business closing to every six hours. Currently, they are trying to further shorten it to three hours. This means that they can accurately estimate the cost of their products, and hence change the sales conditions, every three hours. However, supposing that the sales conditions for popular goods are changed several times a day, customers may not be sure whether the conditions they saw at the merchant site ten minutes ago are correct. Thus, they can fall in a situation in which making shopping decision becomes very difficult. Second, there could be a number of similar shopping sites on the Internet. For example, there are 7,513 registered sites in the Yahoo booksellers' directory[1]. Thus, it is difficult for online customers to be sure if the sites they have chosen are the best ones. Thus, they may wander around to look for other shopping sites with better sales conditions. To make matters worse, on-line merchants compete to provide better sales conditions than others. Such competition may cause a chain reaction among similar shopping sites within a short time[2]. After all, online shopping under highly dynamic business environment can be described in a sentence: "NO ONE KNOWS HOW TO PURCHASE GOODS UNDER THE BEST CONDITION"

To address the problem of online shopping under such a dynamic e-commerce environment, we have proposed a new EC service called "the aCtive Inventory service for Global eMarket" (CIGMA) [16, 15]. The CIGMA provides catalog comparison services as well as one-stop shopping services over multiple shopping sites for ordinary online customers.

SAC 2003 Melbourne, Florida, USA

[1]From Yahoo Web site(www.yahoo.com) on Mar. 14th, 2002

[2]Price War is the term used in economics to indicate such a chain reaction in price changes [8]

An important characteristic of the CIGMA is that the services is based on up-to-date information by reflecting frequent changes in catalog information in real-time. In addition, the service matches the desire of the online customers for fast response. This is possible since the service is based on efficient catalog management schemes.

From a system's perspective, in the core of the CIGMA system[3] is the catalog caching system. It collects and maintains the catalog data from different merchants, and retrieves appropriate ones upon customers' requests. Also, the performance of the CIGMA is largely dependent on that of the catalog cache. In [16], we described the CIGMA focusing on the merchant-side interface including service setup and deployment procedures. In this paper, we further describe the CIGMA along with an overview of its high performance catalog caching system.

With cached catalog data, the CIGMA effectively supports customers to shop online. Customers can easily choose the best sales conditions without exhaustive surfing over shopping sites. They can buy goods from different shopping sites at the CIGMA site without visiting original shopping sites. For merchants, the CIGMA can act as a kind of a sales agency that sells their items on behalf of them, and hence, provide them with a chance to increase their business profits.

1.1 High Performance Catalog Caching System

For an advanced service such as the CIGMA, an ordinary cache architecture such as proxy or reverse proxy is not appropriate. Existing solutions for dynamic data caching[9, 21, 11, 12, 18, 22, 7] is not appropriate, either (See Section5 for comparison with related work). Below, we describe the novel characteristics of the proposed catalog caching system. We believe that the proposed architecture can serve as a reference of an Internet cache for advanced services.

First, it is a high performance cache. Since the CIGMA confronts ordinary customers directly, the performance level of the cache should be very high to process a high rate of requests from lots of customers. In addition, it should be capable of managing a lot of interactions with merchant servers to keep the freshness of cached catalogs. To satisfy such requirements, it caches e-catalogs in the form of source data. With a source data caching, keeping cached data up-to-date can be done efficiently. In addition, we try to further improve the system performance via crafted design and implementation.

Second, the cache supports a cross-organizational service. To gather and manage different kinds of catalogs from heterogeneous shopping sites, a merchant wrapper is provided. Via the wrapper, the cache is able to deal with each different shopping site in a uniform manner.

Third, the cache is equipped with real-time update capability.

This paper is organized as follows. In Section 2, we sketch the design of the CIGMA. Section 3 gives a brief overview of the high performance catalog caching system. Section 4 briefly goes over the prototype implementation and the system performance. Section 5 discusses related work in two aspects: EC service model and dynamic data caching techniques. We conclude our work in Section 6. Due to space limit, we cannot go to the details of the CIGMA as well as the catalog caching system. More details can be found in our technical report[15] and other reports that will follow up.

[3]When required, we distinguish the CIGMA system from the service it provides, i.e., the CIGMA service. Otherwise, we interchangeably use the term "the CIGMA" to mean either the service or the system when there is no confusion.

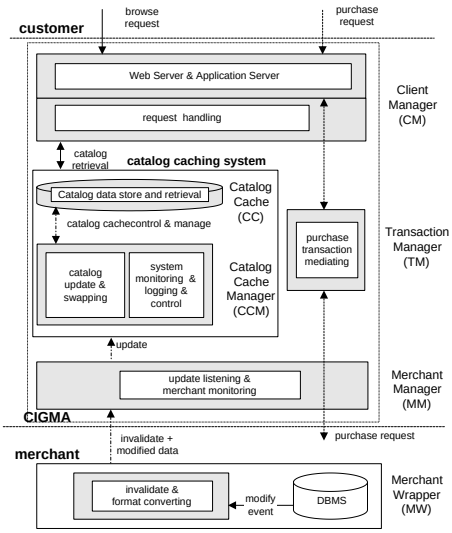


Figure 1: The CIGMA system architecture with service processing flows

2. OVERVIEW OF SYSTEM DESIGN

The CIGMA system has a modular structure. It consists of five server components and one remote component. The remote one, called Merchant Wrapper (MW), runs on each merchant server. The server components are Client Manager (CM), Merchant Manager (MM), Transaction Manager (TM), Catalog Cache (CC), and Catalog Cache manager (CCM). Figure 1 shows the CIGMA system along with a merchant's and a customer's. The function of each component will be described in Section 2.2.

2.1 Internal Service Flows

There are three kinds of external services in the CIGMA: catalog browse, item purchase, and catalog update. The first two are for customers and the last, for merchants with each request being handled differently. The internal service flows for these requests are shown in Figure 1.

The catalog browse request is the most frequent one. It is first received by the Web server, which is a part of the CM. The CM then parses the request and constructs an equivalent query string. It sends the query string to the CC to retrieve the requested information such as a catalog comparison table, an entire catalog, an item category list, etc. Lastly, it replies to the request with an HTML page which is dynamically generated with the query result.

To handle a purchase request, a safety mechanism is required because it generally includes important information such as a credit card number, address, phone number, etc. To secure the purchase transaction processing, the CM communicates with customers using the secure HTTP protocol. Then, it forwards the received data to the TM. The TM also forwards the data to a proper merchant server at once.

The result of the transaction is delivered from the merchant server to the customer in the opposite direction.

In the case of an update request, immediate processing is important. Each modification of source data at the merchant's DBMS is detected by the MW module. Then, the module constructs and sends an update request message to the MM at once. The MM forwards the message to the CCM after verifying the integrity of the message. Then, the CCM actually updates the cached data of the CC by composing a query string based on the message. After the CC update, the return value of the update operation (i.e., OK or NOT_OK) is forwarded to the MW in the reverse order.

2.2 Major Components

Client Manager (CM)

The CM takes charge of every communication with customers. It includes a Web server and a Web application server. The Web application server is required to generate responses with dynamic contents[4]. Via these servers occurs each interaction with customers. The CM receives both catalog browse requests and purchase requests from customers.

Merchant Manager (MM)

The MM manages most interactions with merchants. Upon a catalog modification, the MM receives an invalidation message from a merchant server. Then, the MM immediately notifies the CCM of the event to request the update of the CC.

The merchant server monitoring should be noted. There could be a certain situation such as network partitioning, merchant server failure and network congestion, in which the MM cannot receive invalidation messages. This may result in consistency problem in cached catalogs. To avoid this problem, the MM monitors heart beat messages from merchants to check the status of servers. If a merchant sends no messages during a predefined period, the MM notifies the CCM of the situation. Then, the CCM will initiate a suitable action. For example, the CCM may invalidate all the catalogs from the unreachable merchant server until the server responds again. A merchant is forced to send an empty message at the end of the predefined interval if there is no update.

Transaction manager (TM)

The TM mediates purchase transactions. It does not process the transaction on behalf of merchants, but simply relays all the purchase-related information, i.e., order form, purchase order data, and transaction result, etc., between a customer and a merchant site. This approach is taken to avoid the complication and overhead which may be incurred by related business issues.

The TM mediates purchase transactions in the order of time. An assurance for ordered processing is important when many requests arrive within a short period for a few popular goods with insufficient supply. A fair ordering is guaranteed by tagging messages with its original arrival time at the CM. A purchase request is processed according to the tag until the processing is completed.

[4]Popular web application servers are JAVA servlet, common gateway interface (CGI), active server page (ASP), etc.

Catalog Caching System (CCS)

Catalog Caching System is the most important and complex component in the CIGMA system. It is composed of Catalog Cache (CC) and Catalog Cache Manager (CCM). The CC stores and retrieves cached catalog data. Generally, the performance of storage module is critical to the overall caching performance. The CIGMA is expected to process a much higher number of update requests as well as customer requests than general shopping sites. Therefore, the performance of the CC, which store and manage cached catalogs, is even more critical to our catalog caching system. The CCM has three main functionalities: updating cache data, monitoring and controlling the system, and swapping catalog. The Catalog Caching System is described in detail in Section3.

Merchant Wrapper (MW)

The MW covers all the required works for a merchant to interact with the CIGMA. As shown in Figure 1, the MW is deployed on a merchant server for the merchant to join the CIGMA service. Two main functions of the MW are the update propagation and the heart beat message transmission.

The update propagation is performed according to the real-time update scheme, which is described in section 3.1. The MW also periodically sends out heart beat messages to the CIGMA, if there is no update event during a predefined interval. Although the heart beat message carries no information, the message itself is important because it notifies the CIGMA of the fact that the merchant server is alive. If there are neither update messages nor heart beat messages, the CIGMA regards the merchant as having some critical problems. Thus, the CIGMA can invalidate all catalog data from the merchant to avoid providing catalog data that may be stale.

3. CATALOG CACHING SYSTEM

As the CCS is a source data cache, it stores the source data in unit of database fields such as price, model, stock status, etc., as are saved in merchants' databases. From these source data, a catalog page is generated by Web applications upon each customer's request. Other approaches exist which cache catalog data in the form of a HTML/XML pages or fragments [9, 11, 12, 14] or a query result [19, 21, 22]. However, the source data caching is much more efficient especially in maintaining cached data up-to-date. For instance, the CC can invalidate only the modified fields when a catalog is changed at a merchant site. If other approaches were taken, it would have to invalidate all the related objects including the modified fields. As a matter of fact, source level caching is rather indispensable for the CIGMA and its performance. This is because field level operations over cached catalogs, like sorting according to a specific database field, are essential in providing the CIGMA service.

To manage a large volume of catalog data efficiently for the CIGMA service, a catalog cache requires some basic functionality of DBMS. However, conventional disk based databases are too heavy and complex to provide a high throughput required for an Internet cache. Thus, we designed the CC to be highly light-weighted. We use main memory as the primary storage. The use of main memory significantly improves the performance because, in general, a disk I/O is a fairly heavy operation. Thus, the CC can be

considered to be a kind of a main memory DB. In addition, the following characteristics have been added to further improve the performance.

The CCM takes charge of updating cache data. Upon an update request from MM, the CCM composes a proper query string based on the incoming request. Then, it updates cache data by sending the query to CC. After the update, the return value of the update operation (i.e., OK or NOT_OK) is forwarded to the MW through the MM.

The CCM monitors the status of the system and controls system components. For example, the CCM monitors the frequency of the requests from customers and merchants and collects its statistics. Based on the collected data, it controls major facilities of the components to improve the performance.

3.1 Real-time update mechanism

It is important to keep the data in the CC up-to-date in the CIGMA service. Thus, any modifications in catalog data in merchant databases should be promptly reflected to those in the CC. In addition, the update mechanism should be efficient since a high number of update requests are expected.

The update scheme is based on server invalidation. The merchant server instantly identifies any modification in the database, and initiates an update in the cache by sending out an invalidation message. Thus, an update is propagated to the CIGMA with very small delay.

The instant identification of update is done based on trigger mechanisms in merchant databases. Using a trigger mechanism, the update process can be done very efficiently. It is so since, via the mechanism, changes can be detected in the unit of a field and thus, fine-grained invalidation is possible. Trigger mechanisms are provided in most popular database systems such as Oracle, DB2, Sybase, MySql, PostgresSQL, etc.

Currently, time-to-live (TTL) based schemes are most popularly used as cache consistency mechanisms in the Internet [17]. However, TTL-based schemes are not proper for the CIGMA service since they cannot quickly propagate updates to a cache. Prompt propagation of updates may be achieved if a cache frequently polls changes in servers in a very small interval. However, this will incur excessive overhead to the cache. On the contrary, server-push style approaches can more quickly reflect changes in original data. However, it has been reported that the update schemes of server-push style experience heavy server-side load [10]. This load comes from the fact that a server should keep track of all the caches which hold copies of its data per data item basis. That is, the server has to handle a very high number of caches. However, the CIGMA can avoid this burden. As the CIGMA service requires an explicit permission from a merchant, the merchant server can control the number of contracted caches.

3.2 Catalog Conversion

The CC should integrate catalogs from different merchant sites. Merchants generally use different catalog formats and item categories from each other. The CIGMA also has its own catalog format and category. Therefore, merchants' catalogs have to be re-categorized and their formats, converted to match those of the CIGMA.

For this conversion, the CIGMA uses the Catalog Mapping Table (CMT). The CMT is a table which contains all description for catalog conversion and re-categorization. That is, the exact specification of the conversion is done on the CMT before starting a service and once a service is started, the conversion is automatically processed. The usage of the CMT is not only for a merchant server to convert its catalogs to those of the CIGMA. In converse, the CIGMA also uses the table when it needs to retrieve a field from the merchant DB.

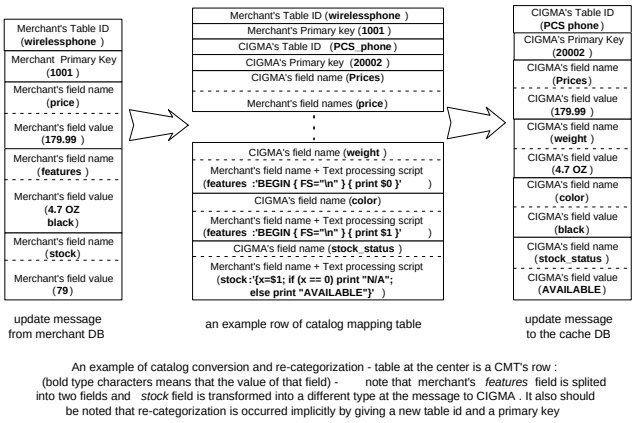


Figure 2: An example of Catalog Mapping Table

The CMT includes a merchant DB's table name and primary key as well as those of the CC. Then, it specifies, using an AWK-like[5] script language, how field names and values in a merchant catalog database is changed to those of CC. Figure 2 shows a row of a CMT example with an incoming update message from a merchant DB and an out-going update message to the cache, respectively. It shows that the item "wirelessphone" is recategorized into the item, "PCS phone". In the figure, the field *features* in the merchant DB is split into two fields: *weight* and *color*. It also should be noted that the type of field *stock* in the incoming message can be transformed into a semantically different type in an out-going message. This semantic transformation is especially important since it enable merchants to control the external appearance of their catalogs.

4. DISCUSSION

We have implemented a prototype of the CIGMA system and two sample shopping sites.[6] Figure 3 shows an snapshot of the CIGMA Web site. We are currently making various performance measurements for the CIGMA system [15]. Due to the limit of the space, in this paper, we do not provide details of the implementation nor performance study. Interested readers are referred to our full paper[15]. Below, we just give a very brief report on the throughput of the CIGMA system upon catalog browsing and comparison requests under a simplistic environment.

[5]AWK is a popular utility in UNIX environment. It enables us to specify what kind of data you are interested in and the operations to be performed when that data is found.

[6]You can view the CIGMA and the two sample sites. The URL of the CIGMA is http://nc9.kaist.ac.kr/cigma/. The two sample merchant sites can be accessed via http://nc2.kaist.ac.kr/merchant/ and http://nc2.kaist.ac.kr/merchant2/.

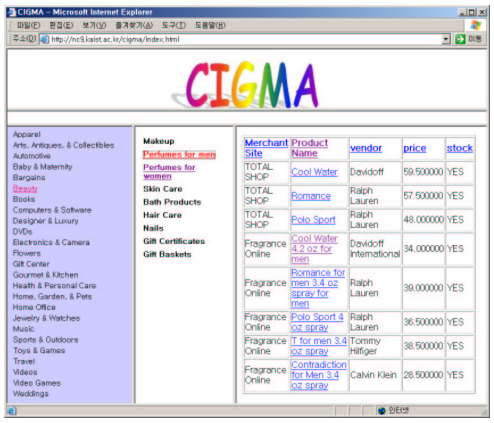
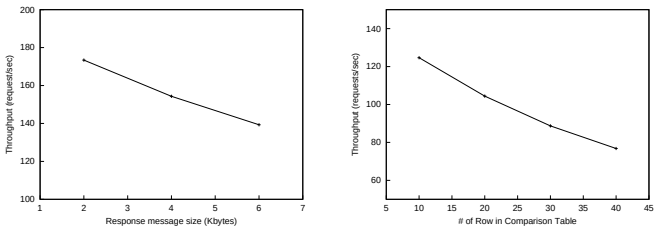


Figure 3: Snapshot of the CIMGA Web site.

The reported measurement was done on a single-node deployment. Under multi-node deployment, we can expect much higher performance numbers than those reported here. The node has a Pentium III 1Ghz CPU and 2GB of main memory. The Red Hat Linux 7.2 was used as the operating system. Apache Web server 1.3.20 and Tomcat 3.2.3 were used as the Web and application servers. Sun JAVA SDK 3.1 was used as the Java virtual machine. As the CIGMA's catalog cache, FastDB version 2.37 was used. The CIGMA server and other systems used to emulate merchants or customers were connected via a 100MB local Ethernet. To test the performance under customer's browse requests, we used the Httperf [13].

Figure 4 (a) shows the throughput of catalog browsing requests as a function of catalog size. The CIGMA can process 173 requests per second when the catalog size is 2 Kbytes. It degrades to 154 and 139 when the catalog size increases to 4 and 6 Kbytes, respectively. The throughput under catalog comparison requests is measured as a function of requested table size. Figure 4 (b) shows that the throughput is from 125 to 77 when the number of items in the comparison table varies from 10 to 40.



(a) (b)

Figure 4: (a) Performance of item browse requests in the function of catalog size (b) Performance of catalog comparison table as the function of result table size

The throughput under catalog comparison requests is smaller than that under catalog browse requests. We conjecture that the performance difference comes mainly from the difference in the overhead to access databases. Generally, to construct a comparison table, a separate database access is required for each row.

5. RELATED WORK

EC service models

In the view of the EC service model, price comparison services [1, 2, 3] are similar to our work. They provide catalogs for an item by gathering them from many merchant sites. However, they are not supported by an automated real-time update scheme. Many services update their data manually or use schemes based on periodic or aperiodic polling [2, 3, 1]. Compared to the CIGMA's case, we may say that those services help online customers in a best-effort style.

AddAll [4] provides price comparison service based on instant catalog searching and gathering upon customer's requests. Thus, it can provide fresh information about an item. However, it also has a serious problem: the response time may be very long because it has to visit many shopping sites in an on-demand fashion. In addition, it issues multiple Web requests per each customer request and thus may cause a heavy traffic on the Internet.

These price comparison services are not a real shopping service. Neither of them have purchase transaction mediation functionality. Thus, customers have to visit the original shopping site to buy a selected item.

The B2B marketplaces [5, 6] intermediates between customers and merchants, and provides a set of services to support on-line purchasing. But, contrary to the CIGMA, most B2B marketplace model is designed and available only for business customers and/or transactions of a large volume. In addition, since B2B marketplaces do not deal with ordinary customers directly, performance concern is less serious in their design.

Techniques to cache dynamic data

Recently, a number of researches have proposed techniques for dynamic data caching [9, 21, 11, 12, 18, 22, 7]. These techniques have been proposed mainly as the scalability solution for ordinary Web services, noting that the generation of dynamic data becomes a major bottleneck. The CCS is different from those because it focuses on the provision of new service, i.e., cross-organizational data services, based on the cached information. The CCS is also different from others in that other caches can be considered as reverse proxies that are used within the contexts of specific servers, whereas the CCS is closer to a proxy that operates along with a number of multiple merchant sites. Below, we classify related techniques in terms of caching units.

HTML page caching stores HTML pages generated upon client's requests [14, 9]. In the view of cache hit gain, i.e. cost saving upon a cache hit, this approach is most advantageous; it saves the cost of query processing as well as that of HTML page generation. However, this approach lacks flexibility; for instance, it is not useful in caching personalized Web pages. Hit ratios for personalized pages will be very low because only specific clients will access those pages. In addition, modification to a common part may result in update of numerous pages.

XML/HTML fragment caching stores XML or HTML fragments which are parts of generated HTML pages. The system proposed in [12] provides an algorithm for efficient update propagation to HTML fragments stored in cache. However, this system requires an administrator to map the re-

lationships between the updates and the fragments affected by their updates. Fragment caching can provide more flexibility than HTML page caching.

Query result caching stores query results in caches. The advantage lies in that it removes the query processing step. Form-based Proxy Caching [20] comes under this category.

The caching schemes summarized above can be called *derived data caching*. On the contrary, the CCS is a *source data cache*. As mentioned, the source level caching is effective in providing advanced services, as shown in the CIGMA service, since fine-grained operation over the cached data is possible. More importantly, source data caching significantly reduces the cost of keeping data up-to-date. In the case of derived data caching, the semantic information of derived data may become different from that of the original data. Thus, it is difficult to automatically find derived data affected by data updates in the original servers. However, source level caching does not reduce the cost of the query processing, rather it moves the cost of the query processing from back-end servers to caches.

6. CONCLUSIONS

Despite many advantages of the EC, online shopping is an overloaded activity for most ordinary customers due to a highly dynamic e-commerce environment. For instance, they are not sure if a chosen sales condition is really a good one even after exhaustive Web surfing. The CIGMA addresses the problems in on-line shopping under such a dynamic environment. It suggests the best sales conditions over multiple shopping sites and provides a convenient shopping environment for customers. Thus, customers can save time and money by using the service. The CIGMA also helps merchants increase their business profit because the CIGMA can act as a sales agency.

An architecture of a new catalog cache system, CCS, is briefly described which comprises the core of the CIGMA system. Technical components of the CCS are sketched with a brief report on performance. We believe that the proposed architecture can serve as example of an Internet cache for advanced Internet services.

Acknowledgments

This work was partly supported by the Korea Science and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Center(AITrc).

7. REFERENCES

[1] http://www.nextag.com/,NexTag - The Search Engine for Shoppers.

[2] http://www.mysimon.com/,mySimon - Compare products and prices from around the Web.

[3] http://www.dealtime.com/,DealTime - compare products, prices & stores.

[4] http://www.addall.com/,AddALL - Book Search and Price Comparison.

[5] http://service.ariba.com/, Ariba Supplier and Buyer Network.

[6] http://www.biz2biz.com/Marketplace/, Biz2Biz B2B Marketplace.

[7] A. Brown, G. Kar, and A. Keller. An active approach to characterizing dynamic dependencies for problem determination in a distributed environment. In *Proc. of IFIP/IEEE International Symposium on Integrated Network Management*, pages 377 – 390, May 2001.

[8] L. M. B. Cabral. *Introduction to Industrial Organization* . MIT Press, 2001.

[9] K. S. Candan, W.-S. Li, Q. Luo, W.-P. Hsiung, and D. Agrawal. Enabling dynamic content caching for database-driven web sites. In *Proceedings of SIGMOD'2001*, pages 532 – 543, May 2001.

[10] P. Cao and C. Liu. Maintaining strong cache consistency in the World Wide Web. *Computers, IEEE Trans on*, 47(4):445–457, April 1998.

[11] J. Challenger, A. Iyengar, and P. Dantzig. A scalable system for consistently caching dynamic web data. In *Proceedings of INFOCOM '99*, pages 294–303, 1999.

[12] J. Challenger, A. Iyengar, K. Witting, C. Ferstat, and P. Reed. A publishing system for efficiently creating dynamic web content. In *Proceedings of INFOCOM '00*, pages 844–853, 2000.

[13] David Mosberger and Tai Jin. httperf: A Tool for Measuring Web Server Performance. *Performance Evaluation Review*, 26(3):31–37, December 1998.

[14] V. Holmedahl, B. Smith, and T. Yang. Cooperative caching of dynamic content on a distributed web server. In *Proc. of the 7th IEEE International Symposium on High Performance Distributed Computing*, pages 243–250, 1998.

[15] S. M. Kim, S. Kang, H.-K. Lee, and J. Song. CIGMA: aCtive Inventory service in Global e-MArket for enabling one-stop shopping over internet shopping sites. Technical Report CS-TR-2002-178, Korea Advanced Institute of Science and Technology (KAIST).

[16] S. M. Kim, S. Kang, H.-K. Lee, and J. Song. CIGMA: aCtive Inventory service in Global e-Market for enabling one-stop shopping over Internet shopping sites. In *Proceedings of the 3rd International Symposium on Electronic Commerce (ISEC-02)*, 2002.

[17] B. Krishnamurthy and J. Rexford. *Web Protocols and Practice - HTTP/1.1, Networking Protocols, Caching and Traffic Measurement*. Addison-Wesley, 2001.

[18] A. Labrinidis and N. Roussopoulos. Webview materialization. In *Proceedings of SIGMOD'2000*, pages 32 – 35, May 2000.

[19] Q. Luo, S. Krishnamurthy, C. Mohan, H. Pirahesh, H. Woo, B. G. Lindsay, and J. F. Naughton. Middle-tier database caching for e-business. In *Proc. of SIGMOD 2002*, June 2002.

[20] Q. Luo and J. F. Naughton. Form-based proxy caching for database-backed web sites. In *Proc. of the Int. Conf. on Very Large Data Bases (VLDB)*, September 2001.

[21] Qiong Luo and Jeffrey F. Naughton and Rajasekar Krishnamurthy and Pei Cao and Yunrui Li. Active query caching for database web servers. In *Proceedings of WebDB'2000*, pages 32 – 35, 2000.

[22] K. Yagoub, D. Florescu, C. C. Andrei, and V. Issarny. Building and customizing data-intensive web site using weave. In *Proc. of the Int. Conf. on Very Large Data Bases (VLDB)*, September 2000.