Web Mining and its SQL based Parallel Execution

Masaru Kitsuregawa, Takahiko Shintani, Iko Pramudiono
Institute of Industrial Science, The University of Tokyo
7-22-1 Roppongi, Minato-ku, Tokyo 106, Japan
{kitsure,shintani,iko}@tkl.iis.u-tokyo.ac.jp

Abstract

Web mining can be classified into two categories, Web access log mining and Web structure mining. We performed association rule mining and sequence pattern mining against the access log which was accumulated at NTT Software Mobile Info Search portal site. Detail web log mining process and the rules we derived are reported in this paper. The parallel association rule mining is explored on large scale PC cluster system. Parallelism is key to improve the performance. We achieved substantial speed up through parallel SQL execution.

1. Introduction

The internet, particularly the web has emerged as largest information pool available in the world. However, most of internet users still do not enjoy the benefits since the wealth of information in the web are largely unstructured and unindexed.

Recent studies also show that the growth of web space is so dramatic by any measures. The number of indexable home pages is approximated to reach 800 million pages by July 1999, and it keeps growing[12]. The same study in January 1998 only recorded 320 million pages.

Therefore the research to provide web users better information quality while stay scaled well with the growth of the web is one of the today hottest topics. The goal is to provide personalized information retrieval mechanism that match the need of each individual user. "One to one marketing" on the web also has similar objectives. The development of the web personalization technologies will certainly benefit e-commerce too. Most researchers agree that the goal can be accomplished by extracting the characteristics of the web and web users behavior. The process in general is called web mining. Thus web mining is the extraction of interesting and potentially useful patterns and implicit information from artifacts or activity related to web.

Web mining can be divided into two categories :

1. Web usage mining

   Customization involves learning about an individual user's preference/interests based on access patterns. The access patterns can be obtained by mining the access log of a web site. Thus, customization aids in providing users with pages, sites, and advertisements that are of interest to them. It may also be possible for sites to automatically optimize their design and organization based on observed user patterns.

2. Web structure mining

   Some approaches has been conducted to mine web structure. Most of them take into account hyperlink information and apply some techniques from graph theory. Interesting concepts such as hubs and authority are invented[11]. Hyperlinks information has been also used to rank search results[6].

Here we report some of the web mining techniques based on association rule that can be accomplished by some modified SQL queries on relational database. The integration of web with database techniques has drawn attention from researchers. Some have proposed query languages for the web that is similar with SQL such as Squel[9] and WebSQL[5]. They emphasize better organization of web data managed in relation database way. We extend this concept for real applications of web mining. We also address the performance problem by performing in parallel the execution of SQL queries on high cost performance PC cluster.

Nowadays, even most of commercial RDBMS are already equipped with parallel processing ability. In near future, we can expect it will become a standard for RDBMS. Although currently available massive parallel hardware are expensive, the development of PC

cluster using commodity PCs will make parallel processing with RDBMS affordable choice for web mining application.

In this paper, we focused on the mining access log using association rule discovery techniques. We show some mining results from web log of Mobile Info Search(MIS), a location-aware search engine. Usage mining of this unique site could give some interesting insights into the behavior of mobile device users.

Although the amount of log at MIS is not so large, generally at large portal site it tends to be very large. The log can reach several tens of GB per day. Just one day log is not enough for mining. If we are going to use several weeks log, then we have to handle more than one terabyte of data. Single PC server cannot process such huge amount of data with reasonable amount of time.

2. Web Usage Mining for Portal Site

2.1. Web Access Log Mining

Access log of a web site records every user requests sent to the web server. From the access log we can know which pages were visited by the user, what kind of CGI request he submitted, when was the access and it also tells to some extent where the user come from. Using those information, we can modify the web site to satisfy the need of users better by providing better site map, change layout of the pages and the placement of links etc. [13] has proposed the concept of adaptive web site that dynamically optimize the structure of web pages based on the users access pattern. Some data mining techniques has been applied on web logs to predict future user behavior and to derive marketing intelligence[17][7][8]. Currently many e-commerce applications also provides limited personalization based on access log analysis. Some pioneers such as Amazon.com have achieved considerable success.

Here we will show the mining process of real web site. We have collaborated with NTT Software to analyze the usage of a unique search engine called Mobile Info Search.

2.2. Mobile Info Search(MIS)

Mobile Info Search (MIS) is a research project conducted by NTT Software Laboratory whose goal to provide location aware information from the internet by collecting, structuring, organizing, and filtering in a practicable form[14]. MIS employs a mediator architecture. Between users and information sources, MIS mediates database-type resources such as online maps,

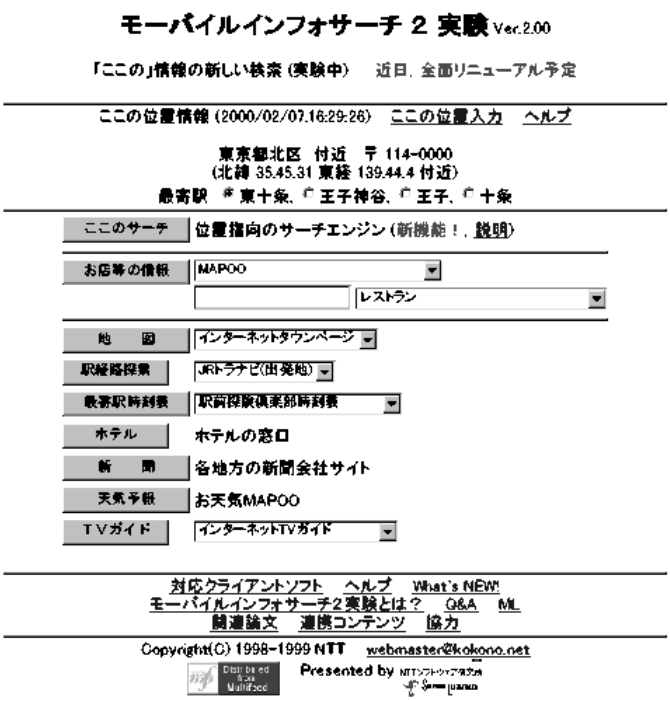

Figure 1. A snapshot of Mobile Info Search at *http://www.kokono.net*

internet "yellow-pages" etc. using *Location-Oriented Meta Search* and static files using *Location Oriented Robot-based Search*. Users input their location using address, nearest station, latitude-longitude or postal number. If the user has a Personal Handy Phone(PHS) or Geo Positioning System(GPS) unit, the user location is automatically obtained.

The site is available to the public since 1997. Its URL is *http://www.kokono.net*. In average 500 searches are performed on the site daily. A snapshot of this site is shown in Figure 1

MIS has two main functionalities :

1. Location Oriented Meta Search

   Many local information on the web are database-type, that is the information is stored in backbone database. In contrast to static pages, they are accessed through CGI program of WWW server. MIS provides a simple interface for local information services which have various search interfaces. It converts the location information and picks the suitable *wrapper* for the requested service. Example of database-type resources provided are shown in table 1.

2. Location-Oriented Robot-Based Search "kokono

Service	Location information used for the search
Maps	longitude-latitude
Yellow Pages	address (and categories ... etc)
Train Time Tables	station
Weather Reports	address or region
Hotel Guides	nearest station

Table 1. Database-type resources on the Internet

Search"

kokono Search provides the spatial search that searches the document close to a location. "kokono" is a Japanese word means *here*. *kokono Search* also employs "robot" to collects static documents from internet. While other search engines provide a keyword-based search, *kokono Search* do a location-based spatial search. It displays documents in the order of the distance between the location written in the document and the user's location.

3. Mining MIS Access Log and its Derived Rules

3.1. Preprocessing

We analyzed the users' searches from the access log recorded on the server between January and May 1999. There are 1035532 accesses on the log, but the log also consists image retrieval, searches without cookie and pages that do not have relation with search. Those logs were removed. Finally we had 25731 search logs to be mined.

- Access Log Format

  Each search log consists CGI parameters such as location information (*address*, *station*, *zip*), location acquisition method (*from*), resource type (*submit*), the name of resource (*shop_web*, *map_web*, *rail_web*, *station_web*, *tv_web*), the condition of search (*keyword*, *shop_cond*). We treat those parameters the same way as items in transaction data of retail sales. In addition, we generate some items explaining the time of access (*access_week*, *access_hour*).

  Example of a search log is shown in Figure 2.

- Taxonomy of Location

  Since names of places follow some kind of hierarchy, such as "city is a part of prefecture" or

```
0000000003 - - [01/Jan/1999:00:30:46  0900]
"GET /index.cgi?sel_st=0&NL=35.37.4.289&EL=138.33.45.315
&address=Yamanashi-ken,Koufu-shi,Oosato-machi
&station=Kokubo:Kaisumiyoshi:Minami-koufu:Jouei
&zip=400-0053&from=address&shop_web=townpage&keyword=
&shop_cond=blank&submit_map= Map&map_web=townpage
&rail_web=s_tranavi&station_web=ekimae&tv_web=tvguide
HTTP/1.1" 200 1389 "http://www.kokono.net/mis2/mis2-header
?date=1999/01/01.00:27:59&address=Yamanashi-ken,Koufu-shi,
Oosato-machi&NL=35.37.4.289&EL=138.33.45.315
&station=Kokubo:Kaisumiyoshi:Minami-koufu:Jouei
&zip=400-0053&from=address&keyword=& shop_web=townpage
&shop_cond=blank&map_web=townpage&station_web=&tv_web=tvguide"
"Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)"
"LastPoint=NL=35.37.4.289&EL=138.33.45.315
&address=Yamanashi-ken,Koufu-shi,Oosato-machi
&station=Kokubo:Kaisumiyoshi:Minami-koufu:Jouei
&zip=400-0053;  LastSelect=shop_web=townpage&shop_cond=blank
&keyword=&map_web=townpage&rail_web=s_tranavi
&station_web=ekimae&tv_web=tvguide; Apache=1; MIS=1" "-"
```

Figure 2. Example of an access log

"a town is a part of a city", we introduce taxonomy between them. We do this by adding items on part of CGI parameter *address*. For example, if we have an entry in CGI parameters entry [address=Yamanashi-ken, Koufu-shi, Oo-satomachi], we can add 2 items as ancestors : [address= Yamanashi-ken, Koufu-shi] at city level and [address=Yamanashi-ken] at prefecture level. In Japanese, "ken" means prefecture and "shi" means city.

- Transformation to Transaction Table

  Finally we have the access log is transformed into transaction table ready for association rule mining. Part of transaction table that corresponds to log entry in Figure 2 is shown in Table 2

3.2. Association Rule Mining

[1] first suggested the problem of finding association rule from large database. An example of association rule mining is finding "if a customer buys A and B then 90% of them buy also C" in transaction databases of large retail organizations. This 90% value is called *confidence* of the rule. Another important parameter is *support* of an itemset, such as {A,B,C}, which is defined as the percentage of the itemset contained in the entire transactions. For above example, *confidence* can also be measured as support({A,B,C}) divided by support({A,B}).

We show some results in Table 3 and 4. Beside common parameters such as *confidence* and *support*, we also

Relation *LOG*

Log ID	User ID	Item
001	003	address=Yamanashi-ken ,Koufu-shi,Oosato-machi
001	003	address=Yamanashi-ken,Koufu-shi,
001	003	address=Yamanashi-ken,
001	003	station=Kokubo: Kaisumiyoshi:Minami-koufu:Jouei
001	003	zip=400-0053
001	003	from=address
001	003	submit_map=Map
001	003	map_web=townpage

Table 2. Representation of access log in relational database

use *user* that indicate the percentage of users logs that contain the rule, i.e. number of users contain the rule divided by total number of users. An user can have some sessions if the user accesses the web site several times and each session usually consists of several transactions.

When we set minimum support to 0.05 %, we got more than 1 million rules. We used following heuristics to extract interesting rules :

1. Put higher minimum confidence restriction to rules with low *support*.

2. Remove item with small contribution to *confidence* of the rule.

   If an item is removed from a rule and its *confidence* is only degraded a little, the item has little importance to the rule.

Those rules can be used to improve the value of web site. We can identify from the rules some access patterns of users that access this web site. For example, from the first rule in Table 3 we know that though Akihabara is a well known place in Tokyo for electronic appliances/parts shopping, user that searches around Akihabara station will probably looks for restaurant. From this unexpected result, we can prefetch information of restaurant around Akihabara station to reduce access time, we can also provide links to this kind of user to make his navigation easier or offer proper advertisement banner. In addition, learning users behavior provides hint for business chance for example the first rule tell us the shortcoming of restaurants in Akihabara area.

Other results in Table 3 show some specific behaviour of people at a certain area in Japan who use this site.

In Table 4, first three results show some correlation between the time of searching and the conditions used for searching. The last three results show how the method used by users to specify the location affects the search conditions. We can use this knowledge to infer what the search conditions a user might choose.

Not so many good restaurants in Akihabara ?
[keyword=][address=Tokyo,][station=Akihabara] ⇒ [shop_cond=restaurant]
In Hokkaido, people looks for gasoline stand at night from its address
[access_hour=20][address=Hokkaido,][from=address] [shop_web=townpage] ⇒ [shop_cond=gasoline]
People from Gifu-ken quite often searches for restaurants
[address=Gifu-ken,][shop_web=townpage] ⇒ [shop_cond=restaurant]
However people from Gifu-ken search for hotels on Saturday
[access_week=Sat][address=Gifu-ken,] [shop_web=townpage] ⇒ [shop_cond=hotel]
People from Gifu-ken must search for hotel around stations
[address=Gifu-ken,][shop_web=townpage] [station=Kouyama] ⇒ [shop_cond=hotel]

Table 3. Some results of MIS log mining with regard to search condition

Most frequent searches for restaurants around 16:00 if they start from address on Friday
[access_week=Fri][from=address][shop_cond=restaurant] ⇒ [access_hour=16]
Most frequent searches for department store stand at 20:00 if start from address.
[from=address][shop_cond=department] ⇒ [access_hour=20]
Looking for gasoline stand on Sunday ?
[from=address][shop_cond=gasoline][shop_web=townpage] ⇒ [access_week=Sun]
Search for hotels often from station if at Kanagawa-ken
[address=Kanagawa-ken,][shop_cond=hotel] ⇒ [from=station]
People at Osaka start searching convenience stores from ZIP number !
[address=Osaka,][shop_cond=conveni] ⇒ [from=zip]
People at Hokkaido always search convenience stores from address
[address=Hokkaido,][shop_cond=conveni] [shop_web=townpage] ⇒ [from=address]

Table 4. Some results of MIS log mining with regard to time and location acquisition method

3.3. Sequential Rule Mining

The problem of mining sequential patterns in a large database of customer transactions was introduced in [3]. The transactions are ordered by the transaction time. A sequential pattern is an ordered list (sequence) of itemsets such as "5% of customers who buy both A and B also buy C in the next transaction".

We show some sequential patterns that might be interesting in Table 5. When we set *support* to 0.1 % we obtained 55000 sequence patterns. Here *support* is the percentage of user sessions containing the pattern.

Some patterns indicate the behavior of users that might be planning to do shopping. We can derive from second pattern that significant part of users check the weather forecast first, then they look for the shops in the yellow-pages service called "Townpage" then look again for additional information in the vicinity with *kokono Search* and finally they confirm the exact location in the map.

After finding a shop, check how to go there and the weather
[submit_shop=Shop Info] → [submit_rail=Search Train] → [submit_newspaper=Newspaper] ⇒ [submit_weather=Weather Forecast]
Or decide the plan after checking the weather first
[submit_weather=Weather Forecast] → [submit_shop=Shop Info] [shop_web=townpage] → [submit_kokono=Kokono Search] ⇒ [submit_map=Map]
Looking for shops after closing time
[submit_shop=Shop Info] [access_hour=22] [access_week=Fri] ⇒ [submit_map=Map] [access_hour=22] [access_week=Fri]

Table 5. Some results of sequential pattern mining

4. Mining Web Log using RDBMS

The ability to perform web mining using standard SQL queries is a next challenge for better integration of web and RDBMS. The integration is essential since better management of web data has become a necessity for large sites.

The performance issue was a major problem for data mining with RDBMS. Parallel execution of SQL can boost the execution of data mining queries. However the problems with this approach are the cost of the hardware and also the degradation of performance when more processing nodes are used. Our PC cluster has shown high cost performance ratio and good speed-up ratio for some decision making applications[15]. We will show that parallel platforms such as PC cluster could also handle the task of web mining with sufficient performance.

4.1. Association Rule Mining Based on SQL

A common strategy to mine association rule is:

1. Find all itemsets that have transaction support above minimum support, usually called large itemsets.
2. Generate the desired rules using large itemsets.

Since the first step consumes most of processing time, development of mining algorithm has been concentrated on this step.

In our experiment we employed ordinary standard SQL query that is similar to SETM algorithm[10]. It is shown in Figure 3.

Transaction data is normalized into the first normal form (transaction ID, item) . In the first pass we simply gather the count of each item. Items that satisfy the minimum support inserted into large itemsets table C_1 that takes form(item, item count). Then transaction data that match large itemsets stored in R_1.

In other passes for example pass k, we first generate all lexicographically ordered candidate itemsets of length k into table RTMP_k by joining k-1 length transaction data. Then we generate the count for those itemsets that meet minimum support and include them into large itemset table C_k. Finally transaction data R_k of length k generated by matching items in candidate itemset table RTMP_k with items in large itemsets.

Original SETM algorithm assumes execution using sort-merge join. Inside database server on our system, relational joins are executed using hash joins and tables are partitioned over nodes by hashing. As the result, parallelization efficiency is much improved. This approach is very effective for large scale data mining.

4.2. Parallel Execution Environment and Performance Evaluation

The experiment is conducted on a PC cluster developed at Institute of Industrial Science, The University of Tokyo. This pilot system consists of one hundred commodity PCs connected by ATM network named NEDO-100. We have also developed DBKernel database server for query processing on this system. Each PC has Intel Pentium Pro 200MHz CPU, 4.3GB SCSI hard disk and 64 MB RAM.

The performance evaluation using TPC-D benchmark on 100 nodes cluster is reported[15]. The results showed it can achieve significantly higher performance especially for join intensive query such as query 9 compared to the current commercially available high end systems.

```
CREATE TABLE LOG (id int, item int);

– PASS 1

CREATE TABLE C_1 (item_1 int, cnt int);
CREATE TABLE R_1 (id int, item_1 int);

INSERT INTO C_1
     SELECT    item AS item_1, COUNT(*)
     FROM      LOG
     GROUP BY item
     HAVING COUNT(*) >= :min_support;

INSERT INTO R_1
     SELECT    p.id, p.item AS item_1
     FROM      LOG p, C_1 c
     WHERE      p.item = c.item_1;

– PASS k
CREATE TABLE RTMP_k (id int, item_1 int,
   item_2 int, ... , item_k int)
CREATE TABLE C_k       (item_1 int,
   item_2 int, ... , item_k int, cnt int)
CREATE TABLE R_k       (id int, item_1 int,
   item_2 int, ... , item_k int)

INSERT INTO RTMP_k
     SELECT    p.id, p.item_1, p.item_2, ... ,
                 p.item_k-1, q.item_k-1
     FROM      R_k-1 p, R_k-1 q

  WHERE      p.id       = q.id
     AND     p.item_1 = q.item_1
     AND     p.item_2 = q.item_2
                          .
                          .
                          .
     AND     p.item_k-2 = q.item_k-2
     AND     p.item_k-1 < q.item_k-1;

INSERT INTO C_k
     SELECT    item_1, item_2, ..., item_k,
                 COUNT(*)
     FROM      RTMP_k
     GROUP BY item_1, item_2, ..., item_k
     HAVING COUNT(*) >= :min_support;

INSERT INTO R_k
     SELECT    p.id, p.item_1, p.item_2, ...,
                 p.item_k
     FROM      RTMP_k p, C_k c
     WHERE      p.item_1 = c.item_1
        AND     p.item_2 = c.item_2
                          .
                          .
                          .
        AND     p.item_k = c.item_k;

DROP TABLE R_k-1;
DROP TABLE RTMP_k;
```

Figure 3. SQL query to mine association rule

We show execution time of SQL query for association rule mining with several minimum supports in Figure 4. The minimum supports are 0.2 %, 0.3 % and 0.4 %. We also varied the number of nodes : 1, 2, 5 and 10. The data used here is synthetic transaction data generated with program described in Apriori algorithm [2] to show that we can handle larger data with PC cluster. The number of transactions here is 200000. We also show the result of directly coded Apriori-based C program on single node for reference. The execution time for this sequential program are shown as dots on the down left of the figure.

The result is surprisingly well compared even with directly coded Apriori-based C program on single processing node. On average, we can achieve the same level of execution time by parallelizing SQL based mining with around 4 processing nodes.

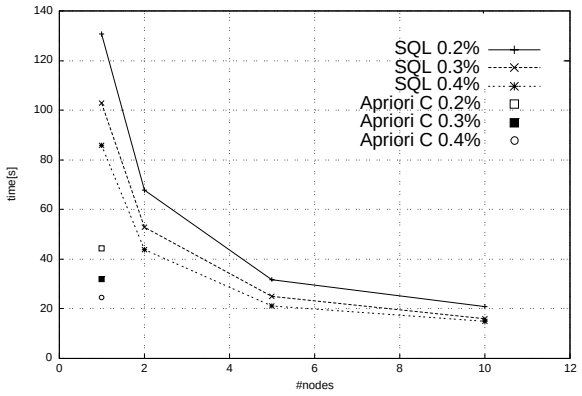

Figure 4. Execution time

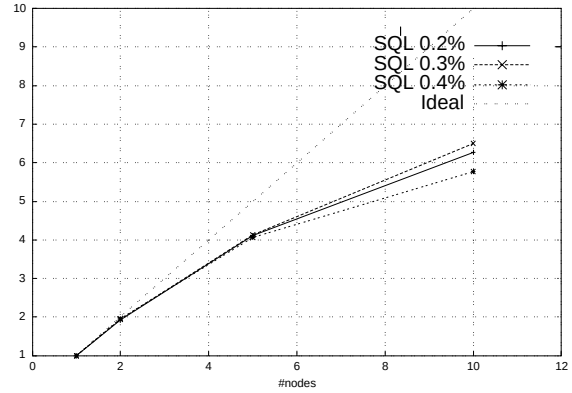

Figure 5. Speedup ratio

The speedup ratio shown in Figure 5 is also reasonably good, although the speedup seems to be saturated as the number of processing nodes increased. Ideal line here is the optimal speedup we can achieve, such as 10 times faster with 10 nodes. As the size of the dataset assigned to each node is getting smaller, processing overhead and also synchronizing cost that depends on the number of nodes cancel the gain.

5. Summary

The web has change some paradigms of information retrieval. The challenge to provide satisfying answers from web queries faces problems from the chaotic nature of web page generation, the phenomenal growth of information on the web and the complexity of web structure. The technology of web mining has showed promising results to improve the information retrieval quality from the web. More sophisticated and complex web mining techniques will be inevitable for the future of web. To support that we need powerful systems, yet with reasonable cost. We have shown that PC cluster

has possibilities to be the platform for large scale web mining.

We have introduced some raw results of data mining on a portal site with focus on mobile users. However we have not provided the evaluation of the quality of the results yet. We believe that the quality of web mining techniques depends on the application. The comparison with other techniques in certain applications such as prefetching and recommendation system will be one of our future work.

Acknowledgements

We would like to thank people from NTT Software, in particular Mr. Katsumi Takahashi and Dr. Atsuhiro Goto for providing the log file of MIS and helpful discussions.

References

[1] R. Agrawal, T. Imielinski, A. Swami. "Mining Association Rules between Sets of Items in Large Databases" In *Proc. of the ACM SIGMOD Conference on Management of Data*, 1993.

[2] R. Agrawal, R. Srikant. "Fast Algorithms for Mining Association Rules" In *Proc. of the VLDB Conference*, 1994.

[3] R. Agrawal, R. Srikant "Mining Sequential Patterns" 'In *Proceedings of Int. Conf. on Data Engineering*, March 1995.

[4] R. Srikant, R. Agrawal "Mining Sequential Patterns: Generalizations and performance improvements" 'In *Proceedings of 5th Int. Conf. on Extending Database Technology*, March 1996.

[5] G. O. Arocena, A. O. Mandelzon, G. A. Mihaila. "Applications of a Web Query Language" 'In *Proceedings of WWW6*, April 1997.

[6] S. Brin, L. Page "The Anatomy of a Large Scale Hypertextual Web Search Engine". In *Proceedings of WWW7*, May 1998.

[7] A. Buchner, M. D. Mulvenna. "Discovering internet marketing intelligence through online analytical Web usage mining" In *SIGMOD Record* (4)27, 1999.

[8] R. Cooley, B. Mobasher, J. Srivistava. "Data preparation for mining World Wide Web browsing patterns" In *Journal of Knowledge and Information Systems* (1)1, 1999.

[9] E. Spertus, L. A. Stein. "Squel: A Structured Query Language for the Web" In *Proceedings of WWW9*, May 2000.

[10] M. Houtsma, A. Swami. "Set-oriented Mining of Association Rules" In *Proc. of International Conference on Data Engineering*, March 1995.

[11] J. Kleinberg. "Authoritive sources in s hyperlinked environment". In *Proceedings of ACM-SIAM Symposium in Discrete Algorithm*, 1998.

[12] S. Lawrwence, L. Giles. "Accessibility of information on the web" In *Nature*, Vol. 400, pp. 107-109, 1999.

[13] M. Perkowitz, O. Etzioni. "Towards Adaptive Web Sites: Conceptual Framework and Case Study", In *Proceedings of WWW8*, May 1999.

[14] Katsumi Takahashi, Seiji Yokoji, Nobuyuki Miura "Location Oriented Integration of Internet Information - Mobile Info Search". In *Designing the Digital City*, Springer-Verlag, March 2000.

[15] Takayuki Tamura, Masato Oguchi, and Masaru Kitsuregawa "Parallel Database Processing on a 100 Node PC Cluster: Cases for Decision Support Query Processing and Data Mining". In *Proceedings of SC97: High Performance Networking and Computing(SuperComputing '97)*, November, 1997.

[16] S. Thomas, S. Sarawagi "Mining Generalized Association Rules and Sequential Patterns Using SQL Queries" 'In *Proceedings of Int. Conf. on Knowledge Discovery and Data Mining*, March 1998.

[17] T. Yan, M. Jacobsen, H. Garcia-Molina, U. Dayal. "From user access patterns to dynamic hypertext linking" In *Proceedings of WWW5*, May 1996.