Web-CAM: Monitoring the dynamic Web to respond to Continual Queries

Shaveen Garg
IIT Bombay
shaveen@cse.iitb.ac.in

Krithi Ramamritham
IIT Bombay
krithi@cse.iitb.ac.in

Soumen Chakrabarti
IIT Bombay
soumen@cse.iitb.ac.in

1. WHY WEB-CAM?

While the flexibility and autonomy of information production and sharing on the Web is phenomenal, it is daunting to navigate, collect, process, and track data in this dynamic and open information space. The problem is aggravated when the sources of information change continually and unpredictably, and the queries are *Continual Queries*, i.e., queries for which responses given to users must be continually updated, as the sources of interest get updated. We need to frequently visit the sites of interest and fuse the newly updated information to track the changes to determine if they are relevant to a user's information needs.

We seek to relieve the user from maintaining personal Web-polling schedules by building Web-CAM, a Web-based Continuous Adaptive Monitoring system. It continuously *push*es the updates to the user while they are still fresh. The term *monitoring* is used explicitly to account for the differences from the classical crawling technique. A *monitoring task* fetches a web page, much like a crawler does, but with the goal of fetching new information relevant to one or more queries, while a *crawl* is not done with any specific user request in mind.

2. HOW WEB-CAM WORKS

Web-CAM [1] follows a multiphase approach as depicted in Figure 1.
(a) Based on a query, specified by a user as a set of keywords, pages relevant to this query are identified.
(b) In the tracking phase, the identified pages are visited at frequent intervals and changes to these pages are tracked. The update statistics are collected and from the observed change characteristics of the pages, a probabilistic model of their change behaviour is formulated and weights are assigned to pages to denote their importance for the current queries. This model yields the probabilities with which the pages are expected to undergo a change of relevance in each time interval (called *epoch*).
(c) In the next (Resource Allocation) phase, we allocate the network resources needed to monitor a set of selected pages. As the resources available are limited, only a set of (predicted) update instances is to be selected for actual monitoring. Web-CAM does its monitoring in *epochs*, each epoch being of a duration which can be changed according to the change frequency of a page. The purpose of this phase is to decide how to allocate the limited network resources for an epoch among the sites to be monitored. Web-CAM's hypothesis is that minimizing the weighted importance of changes that are not reported to users, leads to optimal resource allocation. Thus, the expression for the expected number of lost changes for every page is mathematically minimized to obtain the optimal set of instances at which the pages will be monitored.
(d) Given these resource allocations, the Scheduling phase produces an optimal achievable scheduling of the monitoring tasks. The aim is to minimize the total delay between the ideal time instances and the actual scheduled time instances when a monitoring task is executed.
(e) Pages are monitored as per the designed schedule. Based on the results of the monitoring tasks, scheduling, resource allocations, change statistic computations, and page relevance are revised. A block level diagram of Web-CAM's architecture is shown in Figure 2. The steps involved are numbered in the diagram and their sequence should be self-explanatory.

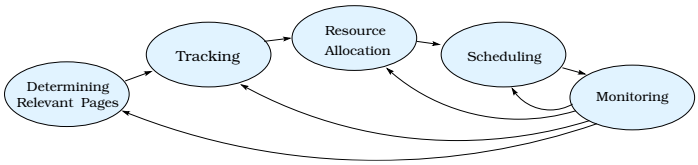

Figure 1: Different phases of Web-CAM approach

3. DEMONSTRATION DETAILS

A user should be able to submit queries and receive updates anytime while he is browsing the Web. Therefore, Web-CAM has been designed as a toolbar in the browser itself. This gives the user an option of submitting a query to Web-CAM as soon as he finds something interesting to monitor. An update notification can also be sent to the user as a message on the toolbar which is better than email notification as the user will be able to access the updated page with just a click on the toolbar.

Our demonstration will showcase the continual query submission process and the update notification process both

SIGMOD 2004 June 13-18, 2004, Paris, France.

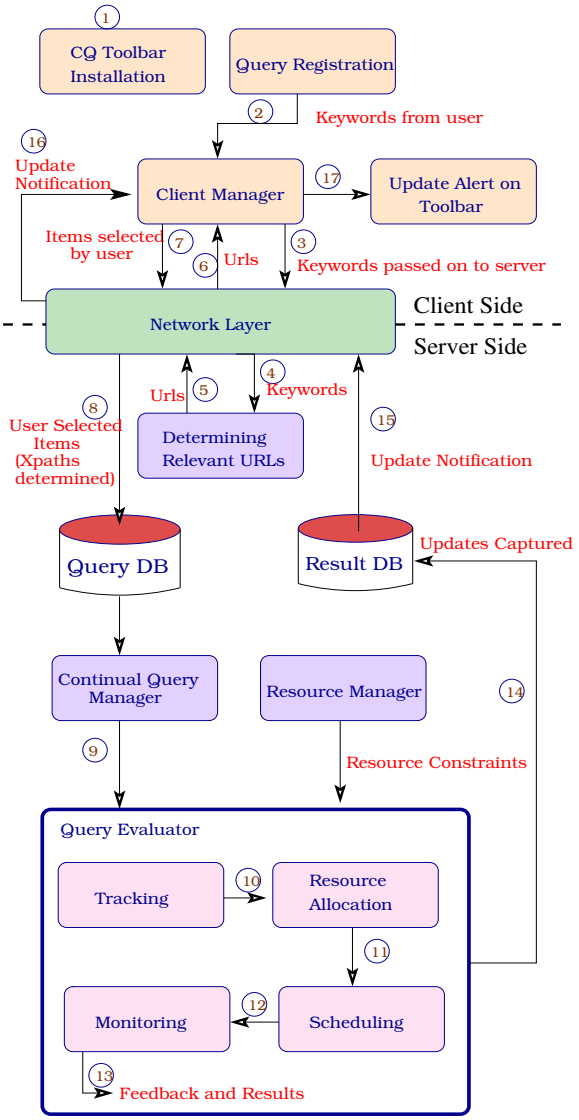

Figure 2: Web-CAM Architecture
(The numbers along the links indicate the step-by-step process from query submission to first update notification)

using a client side toolbar GUI. We will also demonstrate multiple query monitoring by the server side system, the optimal resource allocation technique of the Web-CAM approach, and performance comparison with alternative monitoring and resource allocation techniques.

For the demonstration, we will run our experiments on user queries relating to the domain of **auctions**. Consider a user who is looking for a Sony digital camera for an appropriate price. There are hundreds of auction sites available (e.g. auctions.yahoo.com, auctions.amazon.com, www.ebay.com, auctions.msn.com, etc.). An interesting offer can appear in any of those sites at any time. It is very hard for the user to keep visiting these sites for checking for a good deal. Web-CAM can help him track the sites easily. He just has to submit a query to the Web-CAM once and the system will automatically notify him if there is any new update in the prices of the products of his interest. Similarly, Web-CAM

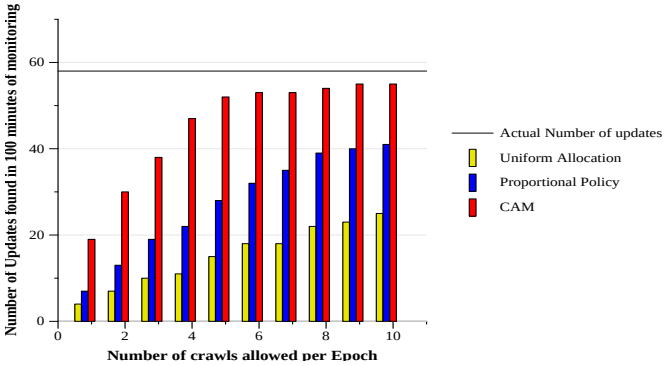

Figure 3: Monitoring performance over a 100-min run.

can also help a user who wants to sell his product on some auction site and he wants to be updated whenever someone increases the bid or someone offers the same product for a lower price.

Figure 3 shows the results for a sample query run on Web-CAM. With search keyword as "digital camera", Web-CAM fetched 35 relevant pages from different auction sites. Two random items were picked from each page to be monitored and fed to the system. The initial epoch time was kept as 5 minutes and resource constraints were varied from 1 to 10 crawls per epoch. The pages were monitored for 100 minutes (after initial tracking of 1 day for building statistics) using the Web-CAM approach and were compared against classical techniques of uniform and proportional allocation. It can be seen that by crawling just 10% of the all the pages, Web-CAM is able to return 80% of the changed information to the users.

The results can be further improved by:
(a) more intelligent monitoring. It is observed that the bid price of an item changes more frequently as the closing time of the auction approaches. Also the number of bids made so far on the tracked item is an indication of how active the auction is. We use this information to assign weights to the pages which help get better results. More of such features like difference of auction price from product's actual market-price, etc. can also be incorporated.
(b) using a better technique for prediction. Web-CAM uses Markov Model to predict the change behaviour of a page based on its past change history. The technique gives good results but still there might be better ways to forecast the behaviour of a page by using standard forecasting time-series techniques like Fourier series, average smoothing, etc. By plugging in these techniques to Web-CAM, its efficiency can be increased even more.

4. ADDITIONAL AUTHORS

Additional authors: Mehul Wagle (IIT Bombay), T. Siva Kumar (IIT Bombay) and Sandeep Pandey (CMU, USA).

5. REFERENCES

[1] S. Pandey, K. Ramamritham, and S. Chakrabarti. CAM: Monitoring the dynamic web inorder to respond to continuous queries. *World Wide Web (WWW)*, 2003.