Proc. 1998 Computer Measurement Group Conference (CMG'98), Anaheim, CA, Dec. 6-11, 1998.

# PREFETCHING RESULTS OF WEB SEARCHES

Harry Foxwell
Sun Microsystems
7900 Westpark Dr.
McLean, VA 22102
Harry.Fowell@sun.com

Daniel A. Menascé
Dept. of Computer Science, MS 4A5
George Mason University
Fairfax, VA 22030
menasce@cne.gmu.edu

*Users of WWW search sites receive Web pages that list documents that match query keywords. This paper examines the behavior of users of Web search sites. The paper shows that for some sites and user populations, the probability that a user will access a document is high enough to justify prefetching it. A formula for the average hit ratio as a function of the number of prefetched documents and the distribution of access to documents returned by search engines is derived. The optimal number of documents to prefetch is computed using the formula and measurement data.*

## 1. Introduction

Users of WWW search sites, such as Yahoo and Lycos, receive dynamically generated Web pages that list URL "hits", or document links, whose contents match query keywords. Users then click on some or all of these links to inspect the document, at which time the server for the link transmits the file. This paper examines the behavior of users of Web search sites. Usage patterns are described for both simple and complex search queries; metrics are suggested for determining which search query results to prefetch in order to reduce network delays as seen by the client.

Measurements suggest that for some search sites and user populations, the probability that a user will access a document is high enough to justify prefetching it. An expression to compute the average hit ratio as a function of the number T of prefetched documents and of the distribution of access to documents returned by search engines is derived. The optimal number of documents to prefetch is computed using the formula and measurement data for Lycos and Yahoo queries. It is also shown that user-perceived delays can be reduced as a result of prefetching.

### *1.1 Searching the Web*

Of the 10 most popular web sites for the week of March 17, 1998, five were Web search engine sites [HOT98]. Yahoo, AltaVista, Excite, Lycos, and Infoseek each serviced several million queries per day during that typical week. Special events, such as the Olympics, can increase Web server access rates to nearly 100 million HTTP operations per day [COCK97]. Although there are some differences among the various search engine sites, in general they provide the same services. A user wishes to locate Web sites that match query keywords. A query term or phrase is entered into a text field on the search site's main page, and the search engine returns dynamically generated HTML pages containing the "hits", or results, of the query. To create this page, the search engine examines an index of words and URLs that are obtained by a network agent called a "webcrawler" that connects to URLs and records their contents. AltaVista, for example, uses a webcrawler that examines as many as 6 million URLs per day, producing an index containing more than 200 GB of words and phrases [ALTA98]. Several multiprocessor servers are used to search this index.

The query result page contains a list of URLs along with additional information. Typically, the page includes the number of hits, the title and description of each URL, and the size of each document. Some search sites, like Excite and Infoseek, return a "relevance rating" for each URL based on the probability of finding the query terms in the document [EXC98]. Lycos and Infoseek return additional information, such as related topic links [LYC98]. Result pages may also contain advertising images and text.

A search for a common term such as "dinosaur" may return tens of thousands of hits; the initial result page usually contains 10 to 20 hits of the highest relevance. The user's browser may provide additional control over the number of links displayed per page.

Users may refine or qualify their query in several ways, seeking an exact phrase match, requiring or disallowing terms, or specifying logical operators (AND, OR). Because of the enormous number of

URLs that a query might return, there have been commercial and research efforts to simplify, organize, and classify the query results. These are mentioned in Section 1.3.

### *1.2 Performance Issues*

The contents of a URL may include text, images, audio and video streams, structured documents, and executable programs. Because some of these components may be relatively large, both the Web user and the local network administrator want to minimize the delivery time for this content while at the same time minimizing the local network load.

Several well-known technologies are used to reduce delays perceived by the user. First, the client's use of the HTTP protocol has evolved and improved. Earlier implementations of HTTP set up and shut down a complete TCP session for each GET operation, creating significant network overhead. The newer HTTP 1.1 protocol supports the "keepalive" feature, which keeps the connection to the Web server open for multiple GET requests, reducing overhead network traffic and user-perceived latency [COCK97,MENA98]. Browsers have also been enhanced to open simultaneous network connections to the same server, hoping to improve response time by overlapping transfers, although this reduces the benefit of the keepalive feature.

Another widely used Web technology for reducing latency and network usage is the proxy cache server. These servers on the user's local network maintain copies of previously referenced files and URLs; when the file is referenced again by any user, it is loaded from the proxy server rather than from the originating server. This can significantly reduce client latency, the time to load and display the file as perceived by the user. Extensive studies of Web servers have typically shown that 30% to 50% of file requests may be handled by a proxy cache server once the cache has been populated [ABRA95, BRAU94]. Additional studies of user access patterns have shown that relatively few servers are responsible for most of the Web traffic. Abdulla and Fox found that 25% of the Web servers accessed were responsible for nearly 90% of the network traffic [ABDU97]. They also found that for some groups of users, less than half of the users were responsible for more than 90% of the Web accesses.

### *1.3 Users' Access Patterns*

Web users tend to be predictable in their interests as well as in their access frequency and duration. Not surprisingly, there are seasonal, monthly, weekly, and daily periodicities in Web accesses, especially among relatively homogeneous populations such as students and corporate workers [ABDU97]. These patterns may be exploited to deliver user-specific content, or to reduce latency and network load. For example, Yan and Jacobsen examined access pattern clusters to classify Web site users, and then dynamically generated suggested URLs based on user profiles [YAN96]. A commercial implementation of this concept, Learn Sesame, tracks and learns individual users' Web access patterns and generates personalized content for each user [RAPO98].

Web server page requests have been shown to approximate a Zipf distribution, sometimes referred to as "Zipf's Law". This distribution describes frequency of occurrence as a function of rank order, and is linear when plotted on a log-log scale [ZIPF35]. Event collections that follow Zipf's Law have a few elements that occur with great frequency and many elements that occur with low frequency. For Web use, this indicates that there are a few sites that many users access and many sites that very few users access. A study of Web use at Sun Microsystems confirms that site accesses closely approximate a Zipf distribution [NEIL97].

### *1.4 Prefetching Web Search Results*

When a search engine user receives a list of URLs, she may follow some of the links and ignore others. This paper investigates user behavior in selecting which links to follow, and suggests answers to the following questions:

- Is the probability that a user selects a link a function of the link's sequence number on the query result page?
- Is there a higher probability of selecting links generated by complex queries than those generated by simple queries?
- Can client latency be reduced by prefetching some of the documents to the proxy cache server?
- What should the decision criteria be for prefetching links?
- How does the hit ratio at the proxy cache server change with the number of documents prefetched?

For the purposes of this study, a *simple* query is defined as a one-word query, and a *complex* query will be defined as any multi-word query.

### *1.5 Organization of This Paper*

The remainder of this paper is organized as follows. Section 2 discusses related research on prefetching

Web data. Section 3 describes the data collection and analysis methods and the system architecture used to run the trace-driven simulations. Section 4 reports the results of this study, and Section 5 discusses the limitations of the design and data. Section 6 discusses the conclusions of this study, and suggests areas of future research on prefetching Web search results.

## 2.0 Related Work on Prefetching

Because of the enormous number of users and interacting systems on the World Wide Web, performance issues have been extensively studied. Many studies focus on predicting user behavior, then prefetching files in anticipation of user access. Cunha and Jaccoud used Random Walk and Digital Signal Processing models to characterize Web users into two groups: *net surfers*, who don't revisit URLs, and *conservative* users, who spend much of their time reexamining previously fetched documents [CUNH97].

Padmanabhan found that prefetching files for Web users decreased client latency, but increased network load and degraded service to other network users [PADM95]. Crovella and Barford suggested that by moderating the rate at which files are prefetched, the additional network load can be minimized [CROV97].

Chinen and Yamaguchi developed a prefetching proxy server, and studied the effects of retrieving the first N links within *any* Web page viewed by a user [CHIN96]. They found that the maximum cache hit rate was about 70% higher for the prefetching proxy than for a non-prefetching proxy, and that latency was significantly reduced. However, they also observed that the network traffic to the prefetching proxy server was nearly *triple* that of the non-prefetching server.

Prefetching Web pages can provide significant benefit to the client user, but places an additional burden on the proxy server, its connection to the Internet, and local network resources by generating unnecessary traffic when the prefetched documents are not accessed by the user. Successful prediction of a user's next access reduces unnecessary traffic.

The following section describes the method used to collect data for our analysis of prefetching documents returned by queries to search engines.

## 3.0 Method and Architecture

### *3.1 Data Used for this Study*

Sun Microsystems's internal network, SWAN (Sun Wide Area Network), supports nearly 25 thousand employees around the world; it provides distributed file and program services over TCP/IP networks to UNIX workstations and Network Computers. Sun employees have unrestricted access to the World Wide Web, and use it extensively for research and information retrieval. Employees maintain several hundred web sites for both internal and public use. SWAN is partitioned into several geographic domains. The data for this study comes from the East.Sun.COM domain's proxy cache server, *webcache.east.sun.com*.

The data represent a typical week of Web traffic from within SWAN to external web sites, April 12, 1998 through April 18, 1998, Sunday through Saturday. Atypical weeks might include corporate holidays, or weeks at the end of fiscal year quarters, when external Web traffic is purposely constrained in order to reserve network bandwidth for critical business needs. During the week studied, webcache.east logged more than 2.5 million HTTP operations.

### *3.2 System Architecture*

Sun employees access SWAN primarily from their desktop workstations, which are almost always some type of SPARCstation running Solaris, Sun's version of UNIX System V. Workstations connect to the network using 10baseT or 100baseT Ethernet. Users run either Netscape or HotJava browsers to access SWAN-based Web servers and external servers. All external access to the Internet goes through the corporate firewall systems. See Figure 1 for details. Each geographic area is assigned its own network domain, and each domain has a proxy cache server. Data for this study came from the East domain's server, covering primarily Eastern United States users, although users can set their browser to use any proxy cache server on SWAN.

The trace-driven regeneration of Web search pages was run from the author's workstation, an UltraSPARC 170 running Solaris 2.6. Data search and analysis programs were run on this workstation and on several multiprocessor UNIX servers available on the network. Programs were primarily Korn Shell scripts; the URL regenerator was written in Java.

### *3.3 Data Analysis*

After obtaining the *webcache.east* log files for the week of April 12-18, queries to *www.yahoo.com* and to *www.lycos.com* were selected into separate files. User identifiers (workstation IP addresses) were extracted. For each unique user query, a Java program accessed the search engine server, regenerated the result page, and extracted and sequenced the returned URLs. Then, for each URL in each user's regenerated result page, the log file was searched to determine if the user had accessed the URL; if so, the sequence number was recorded. The relative frequencies of each

sequence number up to 20 was computed as an estimate of the probability $P_i$ that users will select the *i*-th URL presented by the search engine.

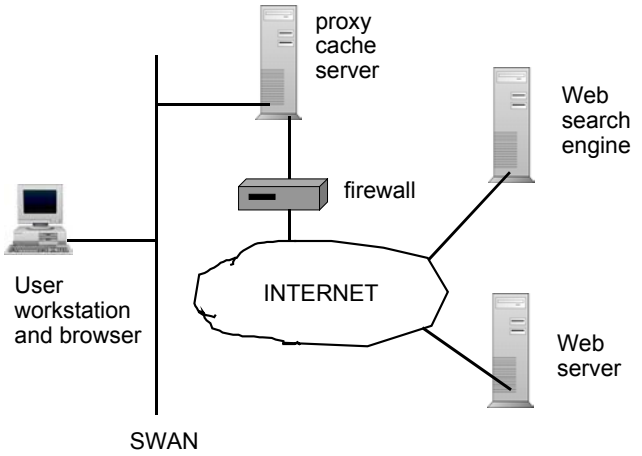


**FIGURE 1 User Access to Internet and Web Search Engines**.

### 4.0 Results

During the week studied, there were 3,081 unique users in the log file, 620 of which (approximately 20%) used search engines (Yahoo, AltaVista, infoseek, Excite, Lycos or HotBot). Table 1 shows the number of users who accessed each search engine, the total number of queries submitted to each search engine, and the average number of queries per user to each search engine. It should be noted that some users accessed more than one search engine.

**Table 1 - Search Engine Statistics.**

| Search Engine | Number of Users | Number of QueRies | Queries per User |
|---|---|---|---|
| Yahoo | 576 | 1320 | 2.29 |
| AltaVista | 290 | 906 | 3.12 |
| infoseek | 221 | 673 | 3.05 |
| Excite | 209 | 407 | 1.95 |
| Lycos | 131 | 294 | 2.24 |
| HotBot | 45 | 221 | 4.91 |

Yahoo users in this study had a low probability of accessing the returned query URLs. $P_1$ was 28% overall, 26% for simple queries, and 37% for complex queries. Hit frequencies for URL sequence numbers 2 and higher were 20% or less. Figure 2 shows the estimated probability that URLs returned by Yahoo queries will be selected by a user.

Users of the Lycos search engine were assumed to be more knowledgeable about the Web and search tools because Lycos is a lesser known search engine [HOT98]. These users were therefore expected to be more interested in the results of their queries. This expectation was confirmed; the Lycos users had much higher hit frequencies for both types of queries, about 86% for $P_1$. Figures 3 shows the hit frequencies for URLs returned by Lycos queries. These results indicate that Web searchers are likely to access the first few results of their queries, and are unlikely to access URLs beyond the first browser page. Highly experienced Web users familiar with alternate search engines are very likely to access returned URLs, and for such users, prefetching these documents may be of value.

To estimate the benefits of prefetching URLs resulting from queries to search engines we compute the hit ratio of prefetched URLs as a function of the number *T*, the threshold, of results to be prefetched. Let,

- $P_i$: estimated probability that the *i*-th URL will be accessed,
- *T:* threshold, i.e., number of URLs to prefetch. The first *T* URLs will always be prefetched.
- *N*: maximum number of URLs resulting from a query to be retrieved by a user. In our experiments we assumed *N* to be equal to 20 since as shown in Figures 2 and 3 the probability of access for the 20-th document is already very low.

We define the *HitRatio* as the ratio of the average number of documents found in the cache per search and the average number of documents accessed. The average number of documents found in the cache is given by

$$\sum_{i=1}^{T} p_i \qquad (1)$$

since all *T* URLs will be found in the cache provided they are accessed. The average number of documents retrieved per search is

$$1 + T + \sum_{i=T+1}^{N} p_i \qquad (2)$$

since the result of the query, a dynamically generated document, is always retrieved, all *T* prefetched documents are retrieved, irrespective of whether they are accessed or not, and the remaining (*N*-*T*) documents are only retrieved if they are accessed.

So, the average hit ratio as a result of prefetching is given by

$$HitRatio = \frac{\sum_{i=1}^{T} p_i}{1 + T + \sum_{i=T+1}^{N} p_i} \qquad (3)$$

Figures 4 and 5 illustrate the variation of *HitRatio* as a function of the threshold *T* for Yahoo and Lycos queries, respectively. As it can be seen from the figures, the average hit ratio increases initially with the threshold *T*, achieves a maximum for an optimal threshold value *T**, and then decreases again. This indicates that a threshold value greater than *T** does not bring benefits in terms of higher hit ratios due to a small probability of the URLs being accessed.

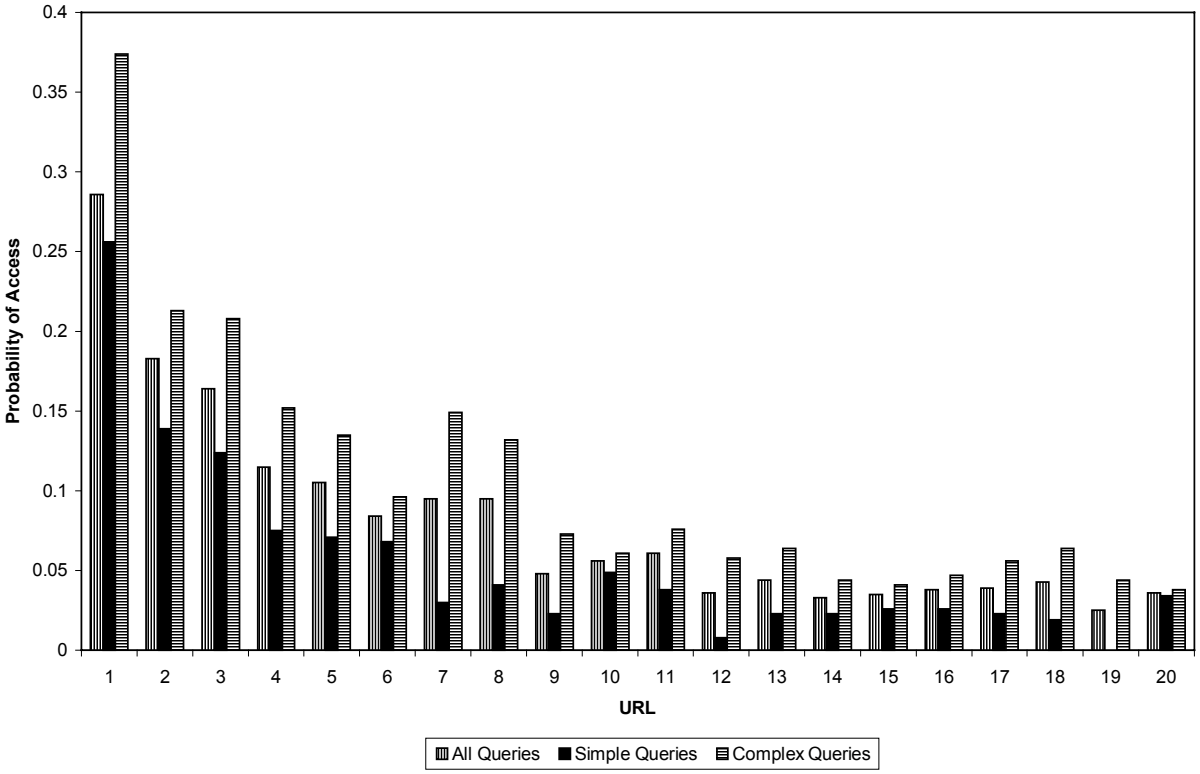


**Figure 2 - Estimated Probabilities of Accessing URLs from Yahoo Queries**

The maximum average hit ratio for Yahoo queries is around 8% and occurs for a threshold value of 3. The low value of the hit ratio for Yahoo queries is due to the low probability of access as shown in Figure 2. On the other hand, Lycos queries exhibit a maximum average hit ratio of 35% when the first eight URLs are prefetched. The value of prefetching for Lycos queries comes from the high access probabilities for the first eight URLs (see figure 3).

While there may be little value in prefetching URLs with low probabilility of access, the potential benefit for Lycos users' search results may be high. To estimate the potential improvement in user-perceived latency, we measured the average retrieval time for the URLs returned by Lycos' queries. Each document was retrieved twice: one directly from the originating server and another from the proxy cache server. Only the HTML code for each page was retrieved; no embedded references or images were retrieved. The mean retrieval time from the originating servers was 1.3 seconds. From the proxy server, the mean retrieval time was 0.8 seconds. So, if eight URLs are prefetched, the average user-perceived latency is 0.35 x 0.8 + (1 - 0.35) x 1.3 = 1.13 sec. This represents a 14% reduction in user-perceived latency.

Further modeling and measurements are needed to characterize latency and throughput for the SWAN network. In general, access to the Web through proxy cache servers may be modeled as a queuing network [MENA94, MENA98]. URL access requests to the proxy server are queued; residence time in the server varies according to the CPU and I/O capabilities of the server. Web servers queue requests from clients and from proxies, process them, and return them to the clients or to the proxies where they are queued for processing. The Internet itself, consisting of routers, bridges, and other network equipment, may also be

modeled as a queue. Each organization's network varies in throughput and server capacities; additional measurement and analysis of SWAN's capabilities would permit estimation of the effects of prefetching. This study did not estimate SWAN's throughput and latency.

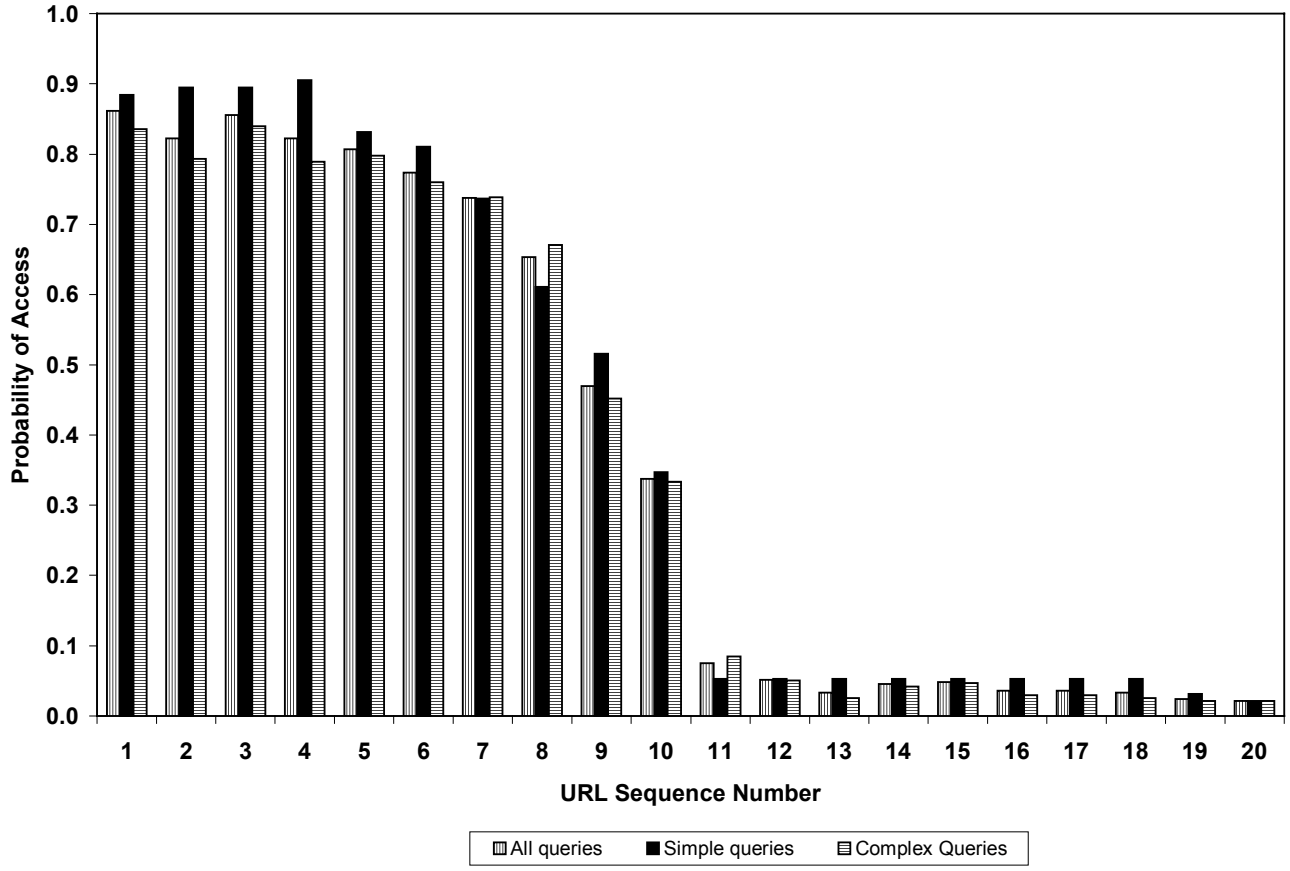


**Figure 3 - Estimated Probabilities of Accessing URLs from Lycos Queries**

In general, prefetching URL contents would reduce average client latency by the difference in delivery times between the originating web server and the proxy cache server. Proxy server load, however, would increase. The load from prefetching documents that *will* be accessed by the user may have to be moderated. Additionally, the proxy server load would be unnecessarily increased by the number and size of documents prefetched but not accessed by the user.

## 5.0 Issues

The measurements to estimate the access probabilities were run two weeks after the original user search data were generated. Because of the rapid change of the number and content of Web links, a page generated by the experiment would not be expected to exactly match the page originally generated by the user. This may cause the measurements to underestimate the user's probability of following a specific link.

The sample population studied consists of users quite familiar with the Web and with search tools. Less experienced populations may exhibit different use patterns.

Regenerating Web pages from log files is a slow and error-prone process. The experiments frequently failed to regenerate a search page due to network timeouts and uninterpretable URLs.

## 6.0 Conclusions and Future Research

For certain classes of Web sites and users, prediction of the user's next URL access is possible. Prefetching the contents of this URL reduces the client latency,

although it does increase proxy server and network load. Further research should examine capturing the hit data at the proxy server, and prefetching files whose hit probabilities are high. Search engines other than Yahoo and Lycos should be investigated: Excite, Infoseek, AltaVista, and corporate *intra*net servers. Other user populations' access patterns should be examined, as well as refining the definition of a complex query to increase the probability of successful prediction of user access. Modifying Web browsers to recognize usage patterns, to predict URL accesses, and to prefetch files might also be considered.

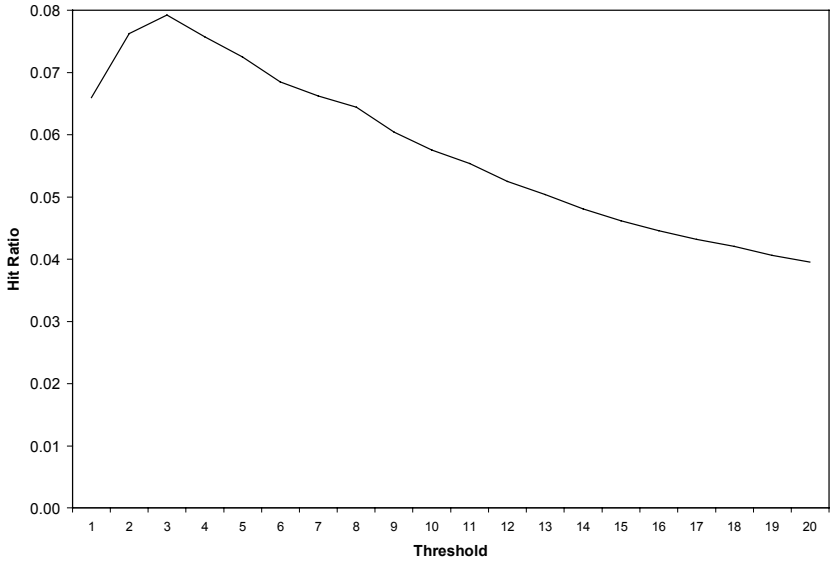


**Figure 4 - Hit Ratio vs. Threshold for Yahoo Queries.**

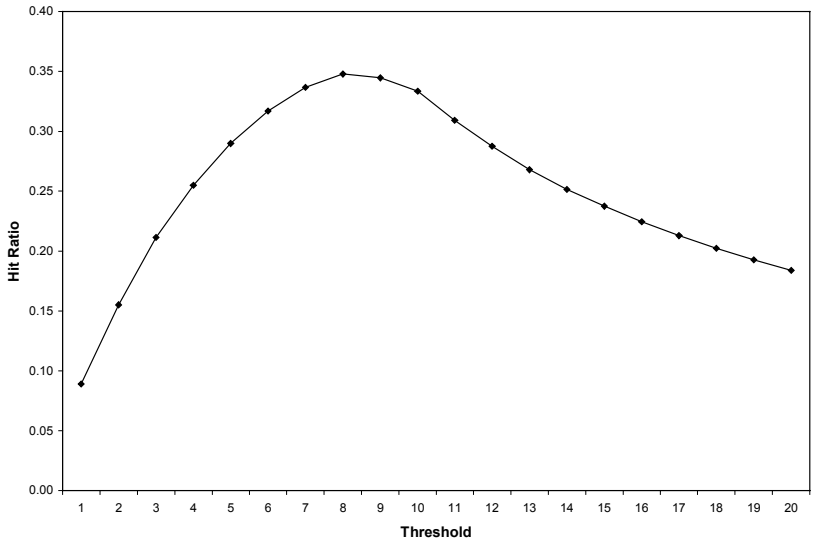


**Figure 5 - Hit Ratio vs. Threshold for Lycos Queries.**


## Acknowledgments

Software, hardware, and data for this study were provided by Harry Foxwell's employer, Sun Microsystems.


## References


[ABDU97] Abdulla, G., Fox, E., Abrams, M., and Williams, S. "WWW Proxy Traffic Characterization with Application to Caching", 1997.

[ABRA95] Abrams, M., Standridge, C., Abdulla, G., Williams, S., and Fox, E. "Caching Proxies: Limitations and Potentials", Computer Science Department, Virginia Tech, 1995.

[ALTA98] AltaVista, http://www.altavista.digital.com.

[BRAU94] Braun, H., and Claffy, K. "Web Traffic Characterization: An Assessment of the Impact of Caching Documents from NCSA's Web Server", *Proceedings of the 2nd International WWW Conference*, Chicago, 1994.

[CHIN96] Chinen, K., and Yamaguchi, S. "An Interactive Prefetching Proxy Server for Improvements of WWW Latency", Nara Institute of Science and Technology, 1996.

[COCK97] Cockcroft, A. "Java Server Sizing Guide", Sun Microsystems, 1997.

[CROV97] Crovella, M., and Barford, P. "The Network Effects of Prefetching", Computer Science Department, Boston University, 1997.

[CUNH97] Cunha, C., and Jaccoud, C. "Determining WWW User's Next Access and Its Application to Prefetching", Computer Science Department, Boston University, 1997.

[EXCI98] Excite, http://www.excite.com.

[HOT98] Hot 100 Websites, http://www.hot100.com.

[INFO98] Infoseek, http://www.infoseek.com.

[LYCO98] Lycos, http://www.lycos.com.

[MENA94] Menascé, D. A., V. A. F. Almeida, and L. W. Dowdy, *Capacity Planning and Performance Modeling: from mainframes to client-server systems*, Prentice Hall, Upper Saddle River, NJ, 1994.

[MENA98] Menascé, D. A. and V. A. F. Almeida, *Capacity Planning for Web Performance: metrics, models, and methods*, Prentice Hall, Upper Saddle River, NJ, 1998.

[NEIL97] Neilsen, J. "Zipf Curves and Website Popularity", http://www.useit.com/alertbox/zipf.htm, 1997.

[PADM95] Padmanabhan, V., "Improving World Wide Web Latency", Computer Science Division, University of California at Berkeley, 1995.

[RAPO98] Rapoza, J., "Learn Sesame Gets More Personal", *PC Week*, March 16, 1998.

[YAH98] Yahoo, http://www.yahoo.com.

[YAN96] Yan, T., Jacobsen, M., Garcia-Molina, H., and Dayal, U., "From User Access Patterns to Dynamic Hypertext Linking", *Fifth International World Wide Web Conference*, Paris, France, 1996.

[ZIPF35] Zipf, G., "Selective Studies and the Principle of Relative Frequencies in Language", 1932.