

A Simple Yet Robust Caching Algorithm Based on Dynamic Access Patterns

*James E. Pitkow
Margaret M. Recker*

Graphics, Visualization, & Usability Center
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280 U.S.A.
Email {pitkow, mimi}@cc.gatech.edu

ABSTRACT

The World-Wide Web continues its remarkable and seemingly unregulated growth. This growth has seen a corresponding increase in network loads and user response times. One common approach for improving the retrieval rate of large, distributed documents is via caching. In this paper, we present a caching algorithm that flexibly adapt its parameters to the hit rates and access patterns of users requesting documents. The algorithm is derived from an analysis of user accesses in a WWW database. In particular, the analysis is based upon a model from psychological research on human memory, which has long studied retrieval of memory items based on frequency and recency rates of past item occurrences. Results show that the model predicts document access with a high degree of accuracy. Furthermore, the model indicates that a caching algorithm based upon the recency rates of prior document access will reliably handle future document requests. The algorithm presented is simple, robust, and easily implementable.

INTRODUCTION

By all indications, the World-Wide Web, or WWW (Berners-Lee, et. al. 1992; Berners-Lee, et. al. 1994), continues its remarkable and seemingly unregulated growth. The rapid growth in the number of users and hosts, information servers (Beebee 1994; Gray, 1994), and network traffic (Merit, 1994) has seen a corresponding increase in network loads and user response times (Viles and French, 1994). Of course, these are hardly problems unique to WWW. Indeed, computer science research has proposed many solutions for improving the retrieval rate of large, distributed documents. One common method is via caching, where certain documents are stored either in main memory or in local disk space. Hence, when a request for a document in the cache is received, it can be processed much more rapidly than having to retrieve the document from the intended server.

To date, several caching algorithms specifically targeted for the WWW have been implemented (e.g., Glassman, 1994; Smith, 1994; Luotonen and Atlis, 1994). Surprisingly, many of these systems lack an empirical or mathematical basis for their algorithms¹. As a result, system administrators wishing to decrease document retrieval latency are forced to arbitrarily define certain important, hard-coded parameters. These parameters include, for example, when to perform garbage collection, the time-to-live (ttl) for cached files, and the maximum file size allowed in the cache. Additionally, system administrators are often forced to manually remove undesired files from the cache in order to improve the cache's efficiency. Thus, it is not surprising to find that such weakly defined systems perform sub-optimally, typically averaging hit rates below 55%.

1. Although Glassman (1994) does model document access rates as a Zipf distribution.

In contrast, we propose that an ideal caching algorithm should flexibly adapt its parameters. Thus, as hit rates and access patterns change, the number of documents, the size of the cache, and the actual documents in the cache should also change. This suggests that an empirical analysis of document access patterns may provide the basis for developing adaptive caching strategies.

In this paper, we present a caching algorithm that is derived from an analysis of user accesses in a WWW database. Our analysis of access patterns is based upon a model from psychological research on human memory, which has long studied retrieval of memory items based on frequency and recency rates of past item occurrences (Ebbinghaus, 1885/1964). It was our expectation that human memory, where many items are recalled on a daily basis from a large available store of memories, forms a useful starting point for understanding document access in large, distributed, heterogeneous information spaces, or what we have termed *dynamic information ecologies*.

In particular, we employ a model from Anderson & Schooler (1991) to estimate the probability of future document access using frequency and recency rates of prior document accesses. The model is applied to the log file of accesses made to the Georgia Institute of Technology WWW repository during a three-month period in 1994. At the time of our analysis, the repository contained more than 2000 multimedia documents, and averaged over 3300 document requests per day. As we will show, the model predicts document access with a high degree of accuracy. Furthermore, the model indicates that a caching algorithm based upon the recency rates of prior document access will reliably handle future document requests.

The remainder of the paper is organized as follows. In the next section, we describe the memory model that underlies our approach. We then show results from modeling document access in a WWW database. This is followed by a description of the caching algorithm derived from the empirical results. This algorithm is simple, robust, and easily implementable.

MEMORY MODEL

Human memory has a long tradition of research in the psychology literature (Ebbinghaus, 1885/1964). One focus of this research is on the relationship of the time delay between when an item is presented and subsequent performance on recall. A related focus is on the number of practice trials for items and subsequent performance on recall. As might be expected, the results show that shorter delays and higher amounts of practice lead to better recall performance.

Based on a review of the memory literature, Anderson and Schooler (1991) argue that the relationship between the time when an item is first presented and subsequent performance (retention) is a power function. Therefore, under a logarithmic transform, a linear relationship is found between the time and performance measures. Similarly, they argue that the relationship between the number of practice trials and performance is a power function. In order to determine how past usage of information predicts future usage, Anderson and Schooler (1991) developed an algorithm for computing and estimating the occurrences of human originated environmental events², based upon event frequency and recency rates.

We used their algorithm in order to determine the relationship between the number of document requests during a period (called the window) and the probability of access on a subsequent day (called the pane). This analysis can be viewed as a parallel to the practice function in human memory research. In this case, given the frequency of past document requests, we are interested in determining the probability of new requests. Following the algorithm described in Anderson and Schooler (1991), we computed the frequency of all document accesses during each 7-day window in an access log file, and measured the probability of access during the next day (i.e., day 8). We selected a window of 7 days because we intuitively felt that this window would encompass the typical fluctuations inherent in the calendar week.

We illustrate how the frequencies and probabilities are calculated with the following example. During Window 1 (day 1 through 7), we find that documents A and B are accessed 6 times. We then find on Pane 1 (day 8) that A is accessed but B is not. Therefore, for the first window and pane, the probability of access for the frequency of value 6 is the sum of accesses in the pane (1+0) divided by the number of accesses in the window (1+1), or .50. These probabilities are calculated for all frequency values.

2. Specifically, they examined the occurrences of words in New York Times headlines, words uttered to newborns (from the CHILDES dataset), and the senders of electronic mail to one recipient.

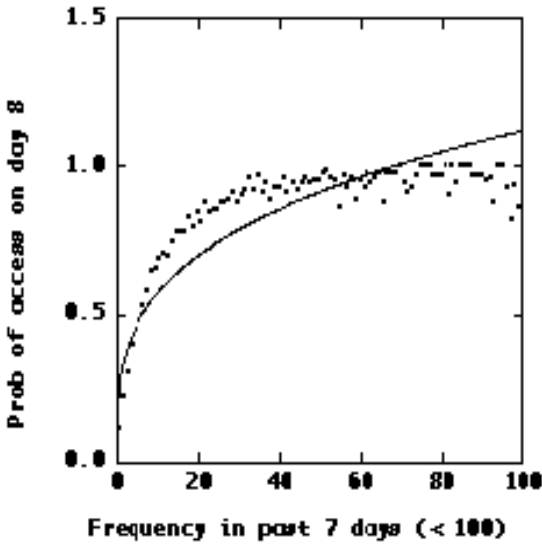


Figure 1: Probability of a document being accessed on Day 8 as a function of the number of times it was accessed in the previous 7 days (for frequencies < 100)

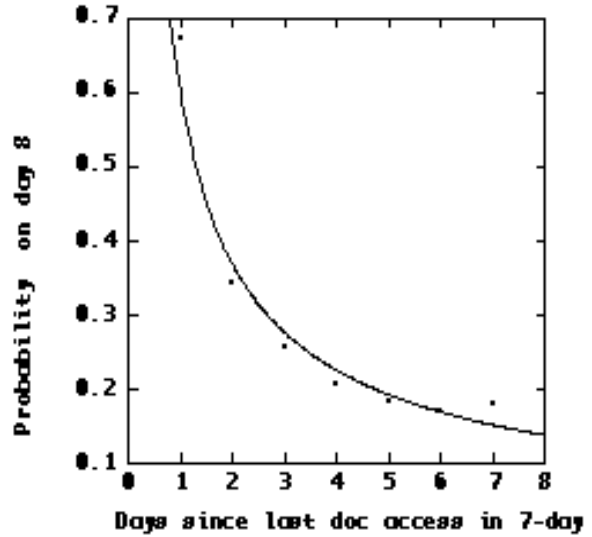


Figure 2: Probability of a document access on Day 8 as a function of how long it has been since the document was accessed in the previous 7 days.

Continuing our example, we find in Window 2 (day 2 through 8) that documents C and D are accessed 6 times. We find on Pane 2 (day 9) that neither document is accessed. Our new probability of access for the frequency of value 6 is the sum of accesses in the two panes (1+0+0+0) divided by the number of accesses in the two windows (1+1+1+1), or .25. In this way, probabilities are computed for all frequency values, for all windows and panes in the dataset. This form of analysis can be viewed as similar to computing the conditional probabilities for a poisson distribution of frequencies. In order to remove the inherent upper and lower bounds from the analysis, Anderson & Schooler (1991) compute the “need odds” for each frequency, which is simply $P/(1-P)$, where P is the probability as computed above.

Similarly, we applied their algorithm in order to determine the relationship between how recently documents are requested during a period and the probability of access on a subsequent day. This analysis parallels the retention function in human memory research. In this case, we are looking at the probability of document access on the eighth day (the pane) based on how many days have elapsed since the document was last requested in the window (still 7 days). The recency probabilities are computed in the same fashion as the frequency probabilities, with recency values begin used instead of frequency values, i.e. the X-axis in Figures 1 and 2.

In the next sections, we describe our test dataset, and show how we applied the model and algorithm to compute and analyze document access rates in the Georgia Tech WWW database in terms of frequency and recency.

Dataset

The dataset used in our analysis was the log file of accesses to the Georgia Tech WWW repository during a three month period, January 1 through March 31, 1994³. At the time of our analysis, the repository contained more than 2000 multimedia documents. From the log file, we removed all accesses made by Georgia Tech machines. We felt that these accesses may not accurately represent the average user to the data because they often represent users testing new documents or default document accesses made by client programs.

The trimmed log file comprised 35 megabytes of data, with a mean record length of 100 bytes and totaling roughly 305,000 requests. The number of requests ranged from 300 to 12,000 document per day, with a mean of 3379 accesses per day over the three month period. Some individual documents in the database were accessed over 4000 times per week.

3. Due to a system failure, data from three days in March are absent.

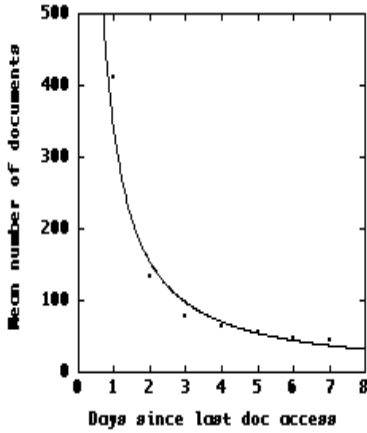


Figure 3: The mean number of documents for each day across windows as a function of how long it has been since the document was accessed in the previous 7 days.

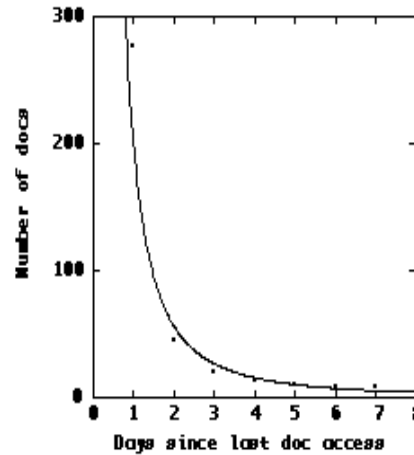


Figure 4: The number of documents in cache on the target day as a function of how long it has been since the document was last accessed in the previous 7 days.

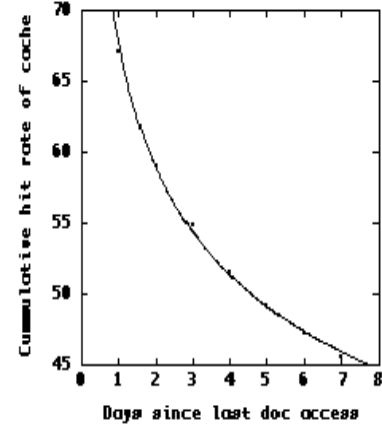


Figure 5: The cumulative hit rate of documents in the cache as a function of how long it has been since the document was last accessed in the previous 7 days.

Results

Results show that the model predicts document access with a remarkable degree of accuracy. Figure 1 plots the probability of access on Day 8 as a function of frequency of access during the previous 7-day window (for frequencies < 100). The plot shows a strong power relationship between frequency and probability of access. Interestingly, this relationship mirrors the power law of practice found in memory research. As expected, the regression analysis of the log-log transform reveals a linear relationship, accounting for 72% of the variability, $F(1, 94) = 246.53$; $p < .001$; $MS_E = 124.32$; $R^2 = .72$.

Similarly, Figure 2 plots the probability of access as a function of the recency of access. The plot shows the steep negative slope typically found in retention plots in memory research. The regression analysis of the log-log transform reveals a near-perfect linear relationship, and accounts for 92% of the variability, $F(1, 5) = 56.17$; $p = .001$; $MS_E = 3.71$; $R^2 = .92$.

These robust relationships were found despite the nonstandard, heterogeneous, and inherently chaotic nature that characterizes WWW repositories, i.e. dynamic information ecologies. Moreover, the frequency and recency relationships mirror those found in the human memory literature. In addition, regression analysis suggest that recency proved to be a much better predictor than frequency.

CACHING ALGORITHM

The results from the previous section indicate that recency and frequency of past document access are strong predictors of future document access. Overall, recency proved to be a stronger predictor than frequency. We note that these results were obtained without knowledge of the *contents* of the documents. Moreover, our results come from analyzing the access patterns of real users, with real information needs -- not users in experimental or laboratory studies. Finally, for the purposes of this paper, we define hit rates as the mean number of documents per day of recency multiplied by the probability that the document is needed on the target day.

Analyses of the results were performed to formulate the caching algorithm. In particular, we focused on deriving strategies for determining which items should be stored in the cache, and for what duration. These analyses are presented in the next section, along with a sketch of the resulting caching algorithm.

Results

The caching strategy we present was formulated from additional analyses of the log file data. Figure 3 shows the mean number of documents that occur for each recency day in all sampled windows. Again, we note that the changes in the number of document accesses decreases through time across days, following a power law relationship.

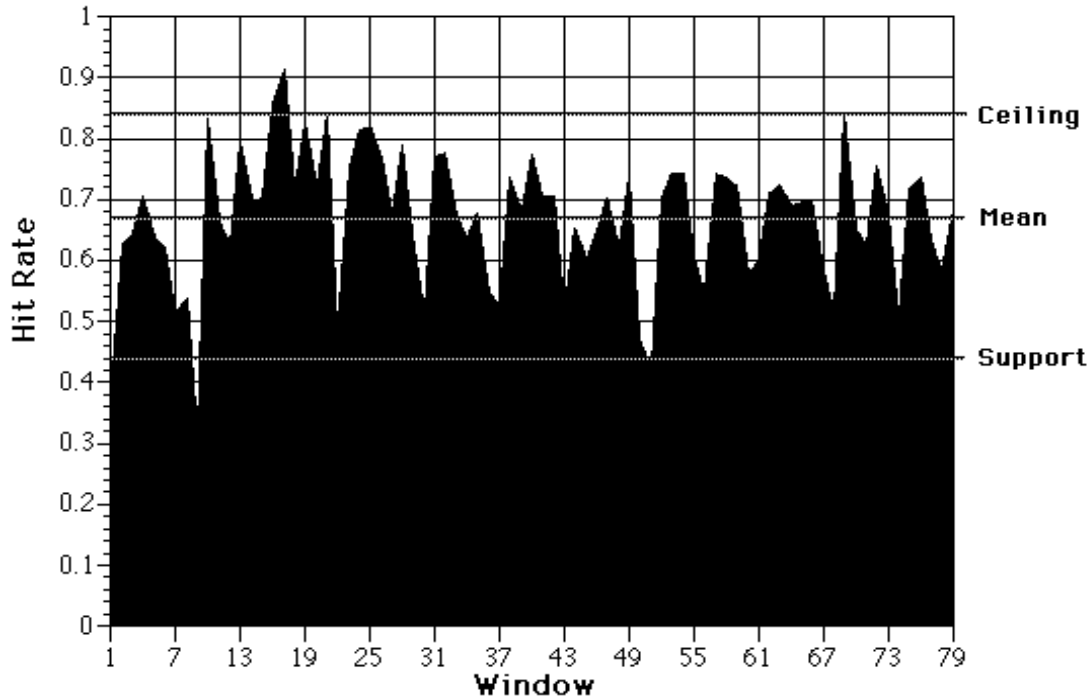


Figure 6: For all accesses with a recency of one day (i.e., the last day in the window) the Y-axis represents the hit rate while the X-axis represents all windows in the dataset.

Figure 4 plots the predicted number of documents that would be in the cache and requested on the target day (day 8). This plot essentially combines Figures 2 and 3. The Y-axis of this graph represents the mean number of documents occurring on each day multiplied by the probability of the document being requested on the target day. The rapid drop off in the number of documents needed in the cache suggests that decreasing the size of the sampling window might lead to a more gradual drop off, though we have yet to examine this possibility.

Figure 5 plots the cumulative hit rate from Figure 4, as a function of the number of days documents are stored in the cache. This plot shows the net effect of implementing a cache with different number of days of recency kept in the cache. Again, we note the presence of the power law relation in examining the predicted hit rates for each day. For example, from this figure, we see that if the cache keeps documents for up to four days since their last access, the cache will contain the requested item on the next day (day 8) 51.5% of the time. Similarly, on day seven, if all requested documents were cached during the sampling window, there is only a 46% chance of a document in the cache being requested on the target day.

Figure 4 and 5 thus strongly suggest that the most important data point for a caching algorithm is the pool of documents accessed one day ago. Figure 6 shows the hit rate of documents stored in the cache with a recency of one day as a function of all windows (from January 1 through March 31). This provides us with a window by window analysis of the model's predictive abilities. Closer inspection of this plot reveals a support line of roughly 44%, a ceiling of 84%, with a mean of 67%.

Finally, upon inspecting the day-by-day predictive abilities of the model as shown in Figure 6, we note that the hit rates decrease during weekends. We believe that this results from the presence of documents requested on Fridays that were not requested on Saturdays. Thus, the observed drops seem natural when taking into account the decreased usage of computer resources on weekends. Furthermore, this suggests that the overall predictive abilities of the model might be increased by excluding weekends from the analysis, thus pushing the support, mean, and ceiling lines up higher than their current values.

Figure 7 shows the results of excluding weekends from the analysis. Fourth degree polynomial curve fitting was used instead of a linear model to explicate inter-window progressions. The removal of weekend data when forming windows results in an overall increase in the predictive ability of the model. Most notably, the support level increases to 54% and the mean and ceiling increase marginally (69% and 85% respectively), though these interpolated lines were not included in the figure to avoid cluttering. However, it is interesting to note that the removal of weekend data from the analysis resulted in the maximum hit rate dropping from 92% to 88%, though the reason for this is unclear.

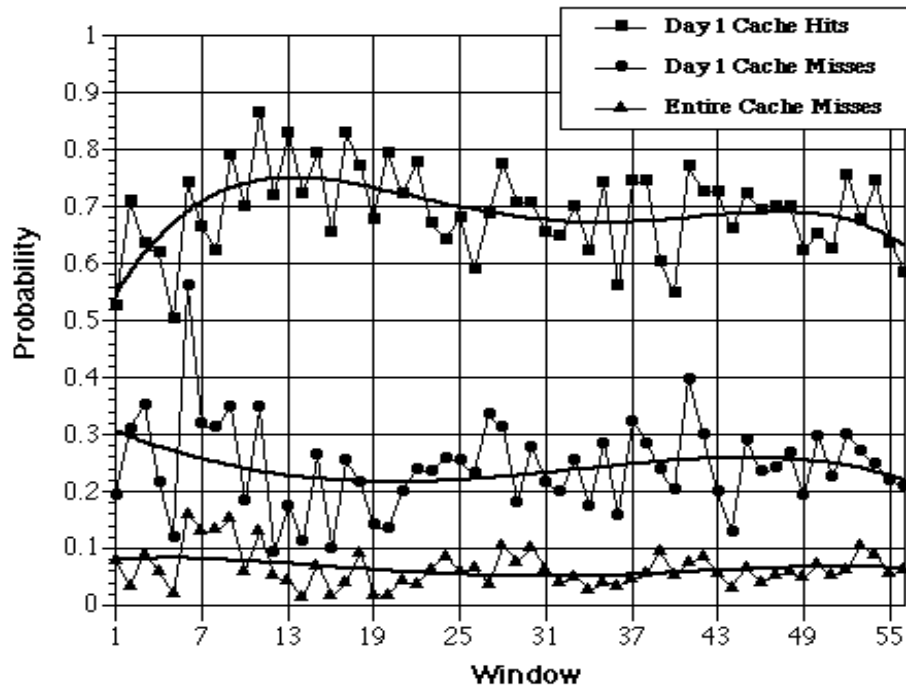


Figure 7: The above graph displays the cache hit and miss ratios for a caching policy of one day and the cache miss ratio for a caching policy of all days in the window. The X-axis represents all windows excluding weekends in the dataset.

To further supplement that analysis, we computed the probability of cache misses, i.e. requests for documents not in the cache on the target day, for a caching policy based on a recency of one day and a caching policy of all the days in the window. That is, the graph displays the probability of cache misses for a cache that stores only documents that were accessed one day ago and for a cache that contains documents requested up to seven days ago. The one day cache had a mean of .24 with a standard deviation of .08 while the entire caching policy had a mean of .06 with a standard deviation of .03. These results support the data from Figure 5 and the existence of a direct relation between cache hits and cache misses in that a cumulative caching policy translates to a decrease in cache hits and misses while a one day old caching policy results in higher cache hits and miss ratios. In sum, these results do suggest that turning off the cache selection algorithm during weekends improves the cache's overall performance, though trade-offs exist.

Caching Algorithm

The above results and analysis provide the motivation for simple, least-recently used (LRU) (Tanenbaum, 1992), caching and garbage collection algorithms. Basically, the cache traps all document requests. If the desired item currently exists in cache, a HEAD request is sent and the last modified time of the file is checked against the cached file.

```

if (cache is full) or (end of day) then
    day = window size
    while (cache is above comfort level) and (day > 1) do
        remove files with recency == day
        day = day-1
    end while
    if day == 1 then
        while (cache is above comfort level) do
            remove largest file
        end while
    end if
end if

```

Figure 8: Sketch of the garbage collection algorithm.

If the cached file is older than the file currently offered from the server, the cache retrieves the document from the server and updates the cache with the latest version⁴. This will ensure that the most recent version of the document is returned to the user. Naturally, if the requested file does not exist in the cache, the cache will retrieve the file and add it to the cache. Of course, special considerations need to be taken into account by the algorithm, for example, checking if the requested URL is actually a Common Gateway Interface (CGI), or contains server-side includes, etc.

A sketch of the garbage collection algorithm is presented in Figure 8. It is important to note that the comfort level (this can be defined as the total size of the cache minus the observed average growth of the database) and window size are adaptable parameters in the algorithm. Based upon the size of the database and the number of accesses to the database, the recency curves can be periodically updated to reflect current access patterns. Hence, rather than having the database's system administrator manually clean the cache and arbitrarily define hard-coded parameters, the caching software can maintain itself by flexibly adapting its parameters to the behavior of users. The parameterization and relationship between parameters is currently being investigated. In addition, while the proposed garbage collection algorithm removes the largest files from the cache only when the most recent documents remain and the cache is above the comfort level, alternative schemes exist. For example, the caching software could compute the frequency rates of each of the remaining documents, similar to the analyses shown in Figure 4, and then remove from the cache the least frequently used documents. Additionally, the software might combine file size information with frequency information to derive a weighted metric. These alternative schemes have yet to be examined thoroughly.

DISCUSSION

In this paper, we have presented a caching algorithm based on an post-hoc analysis of user access patterns in a WWW database. The analysis was based upon a model from psychological research on human memory retrieval. Our analysis of access patterns suggest that recency rates for document access are stronger indicators of future document requests than frequency indicators. These results formed the motivation for a simple, yet robust caching algorithm that can be integrated into existing caching/proxy server software without much difficulty. Furthermore, this algorithm is sensitive to the users accesses of the target repository and hence suggests the possibility of adaptable caching policies.

Developing adaptable caching is desirable since it removes the need for intuitive adjustment of parameters and periodic human cleaning of the cache. Of course, a trade-off between cache optimization and human involvement exists, though the cache hit and miss ratios for the proposed algorithm provide support for continued exploration into the parameterization of the model. This model currently justifies a content-free caching policy based on document accesses one day ago for the studied database. Our current agenda is to explore the parameterization of the model by analyzing a variety of access logs of varying sizes, access rates, etc. and developing caching simulations to benchmark different caching policies. Furthermore, the effects of the parameterization will provide the basis for understanding the trade-offs associated with cache hits and cache misses.

While it is doubtful that the results will greatly effect server-side caching, large scale caches may benefit from this algorithm. In general, caching at the server side level is primarily dependent on the hardware and operating systems used. In contrast, larger scale caching (e.g., caching documents to avoid satellite/transocean communication costs) is typically constrained by the amount of available memory and efficient document removal, i.e. garbage collection, algorithms. While the suggested algorithm has not been tested against the usage pattern of such large-scale caches, it does provide the impetus for further research and cache simulation analysis.

ACKNOWLEDGEMENTS

M. Recker is partially supported by the Army Research Institute for the Behavioral and Social Sciences under Contract No. MDA-903-90-K-112. J. Pitkow is partially supported by Dr. Jorge Vanegas. The programs (in C) and scripts (for Unix) used in performing the analyses described in this paper are available from `URL:ftp://ftp.cc.gatech.edu/pub/gvu/www/pitkow/prediction_analyzer/prediction_analyzer.tar.Z`.

4. Of course, at this point the cache could also check the consistency level requested by the user (i.e. the user demands that the most recent version be returned, the user does not require the latest version of the file, etc.), and then choose the most appropriate action, though the current HTTP protocol does not allow for users to express the desired consistency level.

REFERENCES

- Anderson, J. and Schooler, L. (1991). Reflections of the Environment in Memory. *Psychological Science*, 2(6):192-210.
- Beebee, P. (1994). The SG-Scout Home Page. Available via URL: <http://www-swiss.ai.mit.edu/~ptbb/SG-Scout.html>
- Berners-Lee, T., Cailliau, R., Groff, J., and Pollermann, B. (1992). World-Wide Web: The information universe. *Electronic Networking: Research, Applications, and Policy*, 1(2):52--58.
- Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H., and Secret, A. (1994). The World-Wide Web. *Communications of the ACM*, 37(8):76--82.
- Ebbinghaus, H. (1885/1964). *Memory: A contribution to experimental psychology*. Mineola, NY: Dover Publications.
- Glassman, D. (1994). A Caching Relay for the World-Wide Web. In *Proceedings of the First International World Wide Web Conference*. Amsterdam: Elsevier.
- Grey, M. (1994). Growth of the World-Wide Web. Available via URL: <http://www.mit.edu:8001/afs/sipb/user/mkgray/ht/comprehensive.html>
- Luotonen, A. and Atlis, K. (1994). World-Wide Web Proxies. In *Proceedings of the First International World Wide Web Conference*. Amsterdam: Elsevier.
- Merit NIC. (1994). NSFNET Statistics. Available via URL: <gopher://nic.merit.edu:7043/11/nsfnet/statistics/1994>
- Smith, N. (1994). What can Archives off the World-Wide Web? In *Proceedings of the First International World Wide Web Conference*. Amsterdam: Elsevier.
- Tanenbaum, A (1992). *Modern Operation Systems*. Prentice-Hall, NJ.
- Viles, C. and French, J. (1994). Availability and Latency of World-Wide Web Information Servers. *University of Virginia Department of Computer Science Technical Report CS-94-36*.

AUTHOR INFORMATION

JAMES PITKOW received his B.A. in Computer Science Applications in Psychology from the University of Colorado Boulder in 1993. He is a graduate student in the College of Computing at Georgia Institute of Technology. His research interests include user modelling, adaptive interfaces, and usability. Email: pitkow@cc.gatech.edu.

MIMI RECKER received her Ph.D. from the University of California, Berkeley, in 1992. She is currently a Research Scientist in the College of Computing at the Georgia Institute of Technology. Her research interests include cognitive science; interactive learning environments; human-computer interaction; cognitive modeling; multimedia. Email: mimi@cc.gatech.edu.