Position Statement at ACM SIGCOMM'95 Workshop on Middleware, Cambridge, MA, Aug 1995.

Information Dissemination and Speculative Service

Two Candidate Functionalities for the Middleware Infrastructure*

Azer Bestavros
mailto:best@cs.bu.edu
http://cs-www.bu.edu/faculty/best

Computer Science Department
Boston University

April 29, 1995

Introduction

Recently, there has been a flurry of research on replication techniques to reduce network traffic and minimize the latency of distributed information retrieval systems. Most of these techniques have concentrated on client-based caching, whereby recently/frequently accessed information is cached at the client (or at a proxy thereof) in anticipation of future accesses to the same information. We believe that these local (myopic) solutions, focussing exclusively on a particular client or set of clients, are likely to have limited impact. In a number of recent pilot studies [1, 2], we have demonstrated that (at least for HTTP traffic) servers are in a much better position to make decisions regarding *what*, *where*, and *when* to replicate information—or at least influence the making of such decisions. In this position paper we describe two middleware components that we believe must be part of future network services: popularity-based information dissemination and speculative service.

Popularity-based dissemination of information to reduce traffic and balance load

In [1], we proposed a hierarchical demand-based replication strategy that *optimally* disseminates information from its producer to servers that are closer to its consumers. The basic idea of our protocol is to *off-load* popular servers by duplicating (on other servers) only a small percentage of the data that such servers provide. The extent of this duplication (how much, where, and on how many sites) depends on two factors: the popularity of the server and the expected reduction in traffic if dissemination is done in a particular direction. In other words, our protocol provides a mechanism whereby "popular" data is disseminated automatically and dynamically towards consumers—the more popular the data, the closer it gets to the clients. We used the logs of our departmental HTTP server as well as the logs of the more popular Rolling Stones HTTP server to validate an analytical model of server popularity and file access profiles. Using that model we showed that by disseminating the most popular documents on servers (proxies) closer to the clients, network traffic could be reduced considerably, while servers are load-balanced.

To measure the potential of our protocol, we performed trace simulations using the logs of the Rolling Stones HTTP server `http://www.stones.com` from November 1, 1994 to February 19, 1995. These logs reflect total accesses of over 1 GigaByte of multimedia information per day, serviced to over 100,000 clients. Figure 1(a) shows the percentage reduction in server load (measured in bytes serviced) if various block sizes of decreasing popularity are disseminated to and serviced at other servers. Of the 400 MBytes of information accessed at least once[1] during the analysis period, replicating and servicing only 21 MBytes (5.25%) at other servers resulted in a reduction of 85% in the load at the Rolling Stones server. Figure 1(b) shows the percentage reduction in network bandwidth (measured in bytes*hops) if various block sizes of decreasing popularity are dessiminated to and serviced at enough other servers[2] to reduce the average number of hops between any client and `www.stones.com` by 4 hops.

*This work has been partially supported by NSF (grant CCR-9308344).

[1] Notice that the total number of bytes *available* from that server is much larger than 400 MBytes.

[2] For `http://cs-www.bu.edu` the number of such servers was 8.

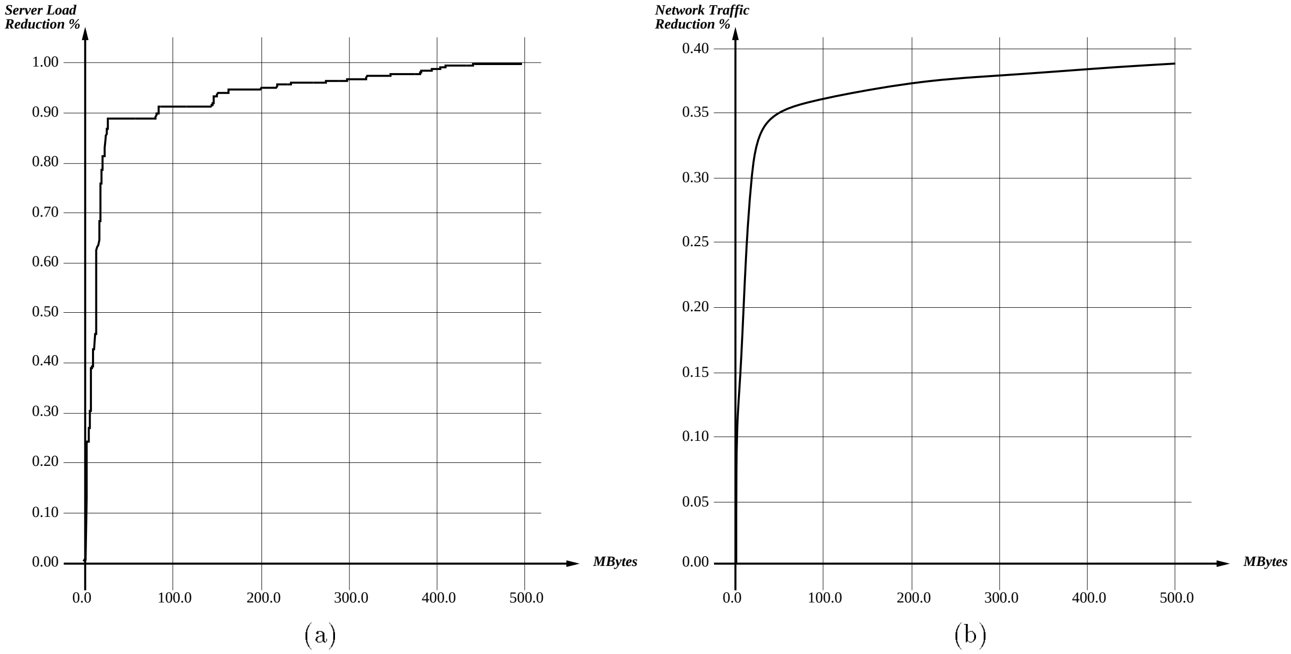

Figure 1: Projected reduction in server load (a) and network traffic (b) as a result of dissemination.

Speculative service to reduce service time and server load

Speculative service implies that a client's request for a document is serviced by sending, in addition to the document requested, a number of other documents (or pointers thereto) that the server (or proxies thereof) *speculates* will be requested by the client in the near future. This speculation is based on statistical information that the server maintains for each document it serves. The notion of speculative service is analogous to prefetching, which is used to improve cache performance in distributed/parallel shared memory systems, with the exception that servers (not clients) control when and what to prefetch. In [2], and using extensive trace simulations we showed that *both* server load and service time could be reduced considerably, if *speculative service* is used. This is *above and beyond* what is currently achievable using client-side caching [3] and server-side dissemination [1].

Given that a client has requested a particular document (say $\mathcal{D}_i$), what is the likelihood that it will request another document (say $\mathcal{D}_j$) in the *near* future? Let $p[i,j]$ denote the conditional probability that document $\mathcal{D}_j$ will be requested within a limited window of time T_w, given that document $\mathcal{D}_i$ has been already requested. Let P denote the square matrix representing $p[i,j]$, for all possible documents $0 \geq i, j \leq N$. We define P^* to be the closure of P, where $P^* = P^N$. Obviously, $p^*[i,j]$ denotes the probability that there will be a sequence of requests starting with document $\mathcal{D}_i$ and ending with document $\mathcal{D}_j$, in which every request is separated by at most T_w units of time from the previous request in that sequence.

We used the logs of our departmental HTTP server `http://cs-www.bu.edu` (over 200,000 accesses from almost 8,500 clients, representing almost 19,000 sessions) to drive a trace simulation to study the potential of speculative service. Figure 2(a) shows the reduction in server load, the reduction in service time, and the reduction in client caching miss rate for various levels of speculative service (T_p). The figure also shows the increase in traffic as a result of speculative service. Figure 2(b) shows the reduction in server load, service time, and client caching miss rate as a function of the percentage increase in traffic (as a result of speculation). These results suggest that a significant improvement in performance could be achieved for a miniscule increase in traffic. In particular, using only 5% extra bandwidth results in a whopping 30% reduction in server load, a 23% reduction in service time, and a 18% reduction in client miss-rate. Using 10% extra bandwidth results in a reduction of 35%, 27%, and 23% in these metrics, respectively.

In a separate study [4], we are evaluating the merits of client-initiated prefetching based on user access patterns of WWW documents. In that study, extensive user logs [5] are analyzed to obtain a per-user relationship similar to the P and P^* relationships (i.e. a user profile). Such a relationship is used to initiate document

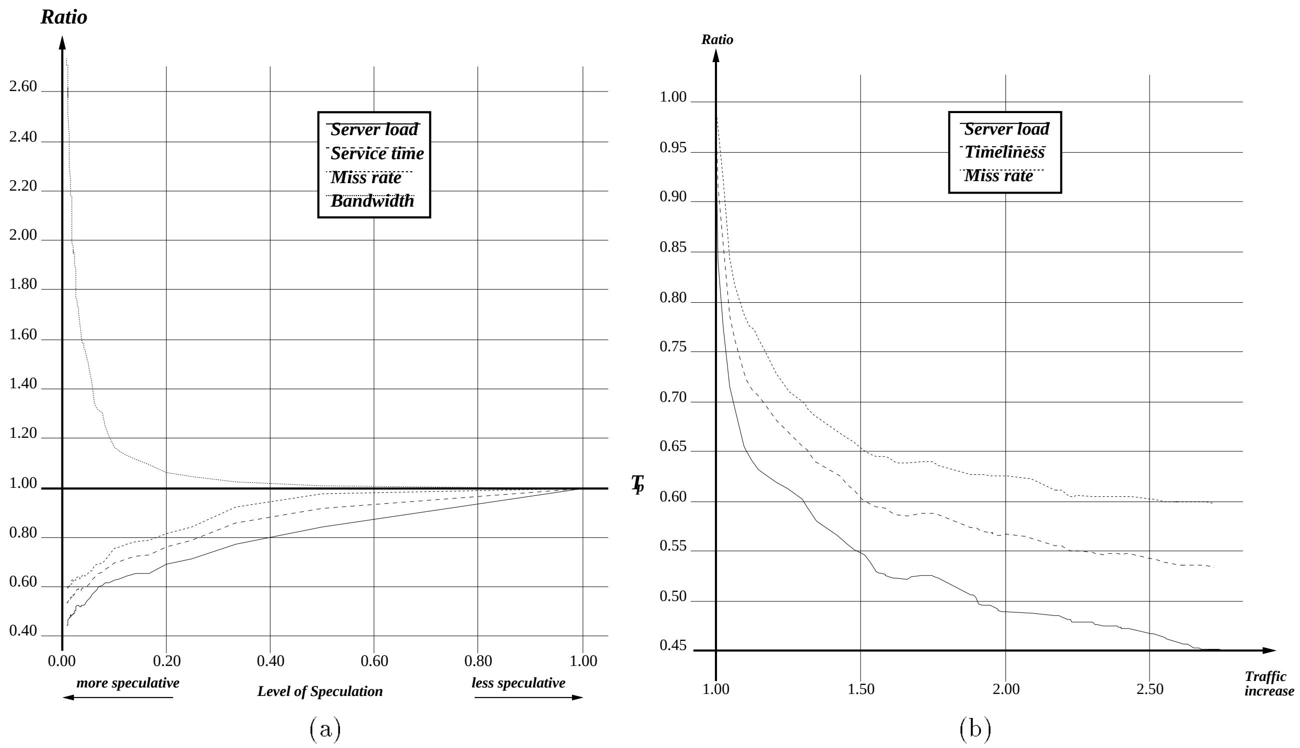

Figure 2: Effect of speculation on server load, service time, miss rate, and network bandwidth

prefetching. Preliminary results indicate that client-initiated prefetching is extremely effective for access patterns that involve frequently-traversed documents, but (obviously) not effective at all for access patterns that involve newly-traversed documents. For such access patterns, only speculative service could improve performance. Currently, we are investigating the incorporation of client-initiated prefetching (based on user access patterns) and server-initiated speculative service (based on server logs) into a single protocol.

Conclusion

We believe that involving servers into the caching, replication, and prefetching decisions is crucial, if current network protocols and services (such as the WWW) are to scale up. We have identified at least two functionalities of the middleware infrastructure that we believe must be incorporated in future scalable systems: popularity-based dissemination and speculative service.

References

[1] Azer Bestavros. Demand-based document dissemination for the world wide web. Technical Report TR-95-003, Boston University, CS Dept, Boston, MA 02215, February 1995. (submitted for publication).

[2] Azer Bestavros. Using speculation to reduce server load and service time on the www. Technical Report TR-95-006, Boston University, CS Dept, Boston, MA 02215, April 1995. (submitted for publication).

[3] Azer Bestavros, Robert Carter, Mark Crovella, Carlos Cunha, Abdelsalam Heddaya, and Suliman Mirdad. Application level document caching in the internet. In *IEEE SDNE'96: The Second International Workshop on Services in Distributed and Networked Environments*, Whistler, British Columbia, June 1995.

[4] Azer Bestavros and Carlos Cunha. A prefetching protocol using client speculation for the www. Technical Report TR-95-011, Boston University, CS Dept, Boston, MA 02215, April 1995.

[5] Carlos Cunha, Azer Bestavros, and Mark Crovella. Characteristics of www client-based traces. Technical Report TR-95-010, Boston University, CS Dept, Boston, MA 02215, April 1995.