Web traffic characterization: an assessment of the impact of caching documents from NCSA's web server [*]

Hans-Werner Braun
Kimberly Claffy
Applied Network Research
San Diego Supercomputer Center
P.O. Box 85608
La Jolla, CA 92186-9784

phone: (619) 534 5035 email: `hwb,kc@sdsc.edu`
www: `http://www.sdsc.edu/0/SDSC/Research/ANR/`

September 23, 1994

We analyze two days of queries to the popular NCSA Mosaic server to assess the geographic distribution of transaction requests. The wide geographic diversity of query sources and popularity of a relatively small portion of the web server file set present a strong case for deployment of geographically distributed caching mechanisms to improve server and network efficiency.

The NCSA web server consists of four servers in a cluster. We show time series of bandwidth and transaction demands for the server cluster, and break these demands down into components according to geographical source of the query. We analyze the impact of caching the results of queries within the geographic zone from which the request was sourced, in terms of reduction of transactions with and bandwidth volume from the main server. We find that a cache document timeout even as low as 1024 seconds (about 17 minutes) during the two days that we analyzed would have saved between 40% and 70% of the bytes transferred from the central server. We investigate a range of timeouts for flushing documents from the cache, outlining the tradeoff between bandwidth

---

[*] This research is supported by a grant of the National Science Foundation (NCR-9119473), and other support from the National Science Foundation for studying engineering and architectural issues of the National Research and Education Network

savings and memory/cache management costs. We discuss the implications of this tradeoff in the face of possible future usage-based pricing of backbone services that may connect several cache sites.

We also discuss other issues that caching inevitably poses, such as how to redirect queries initially destined for a central server to a preferred cache site. The preference of a cache site may be a function of not only geographic proximity, but also current load on nearby servers or network links. Such refinements in the web architecture will be essential to the stability of the network as the web continues to grow, and operational geographic analysis of queries to archive and library servers will be fundamental to its effective evolution.

keywords: traffic analysis, geographic distribution, server workload, caching, accounting

1 motivation

Since February 1993 the National Center for Supercomputing Applications (NCSA) has operated a digital library server accessed by several hundred thousand users worldwide using the NCSA Mosaic software package. NCSA Mosaic is able to interface with several other information resource discovery service (IRDS) protocols in addition to the WorldWideWeb (*www*), or *http* protocol; it also includes support for *gopher*, *wais*, *ftp*, *nntp* (news), and even *telnet* and *finger*. During the past 18 months, since widespread introduction of NCSA Mosaic into the Internet, usage of the server has doubled roughly every 6-8 weeks, reaching 2.54 million transactions per week in early September 1994.

In the face of the overwhelming growth, NCSA has already implemented a round-robin scheme to distribute the *www* workload across a cluster of servers. Other major information servers, such as CERN and the UIUC Weather Server, are also beginning to suffer under the rapidly increasing load. At least a dozen *www* server sites are experiencing similar growth to that of NCSA, increasing the urgency for a more comprehensive and systematic approach to the analysis of both the requirements of server architectures and the impact of this rapid growth on the Internet today and in the future.

Figure 1 shows the growth of traffic from several IRDS applications on the NSFNET backbone, comparing it to the overall growth of NSFNET backbone traffic. Although the figure illustrates the exceptional growth of *www* traffic, simple metrics such as packet or byte counts do not reveal the burden that *www* traffic imposes on individual servers or how long before the demand reaches a threshold which the server and/or network cannot support. A rapid growth in information provisioning has occurred without an overall architecture to address models of information resources and their deployment strategies, e.g., optimized interaction between applications and network protocols, or overall network efficiency considerations such as file caching [1]. As a result, measurements that would indicate performance thresholds are not widely available.

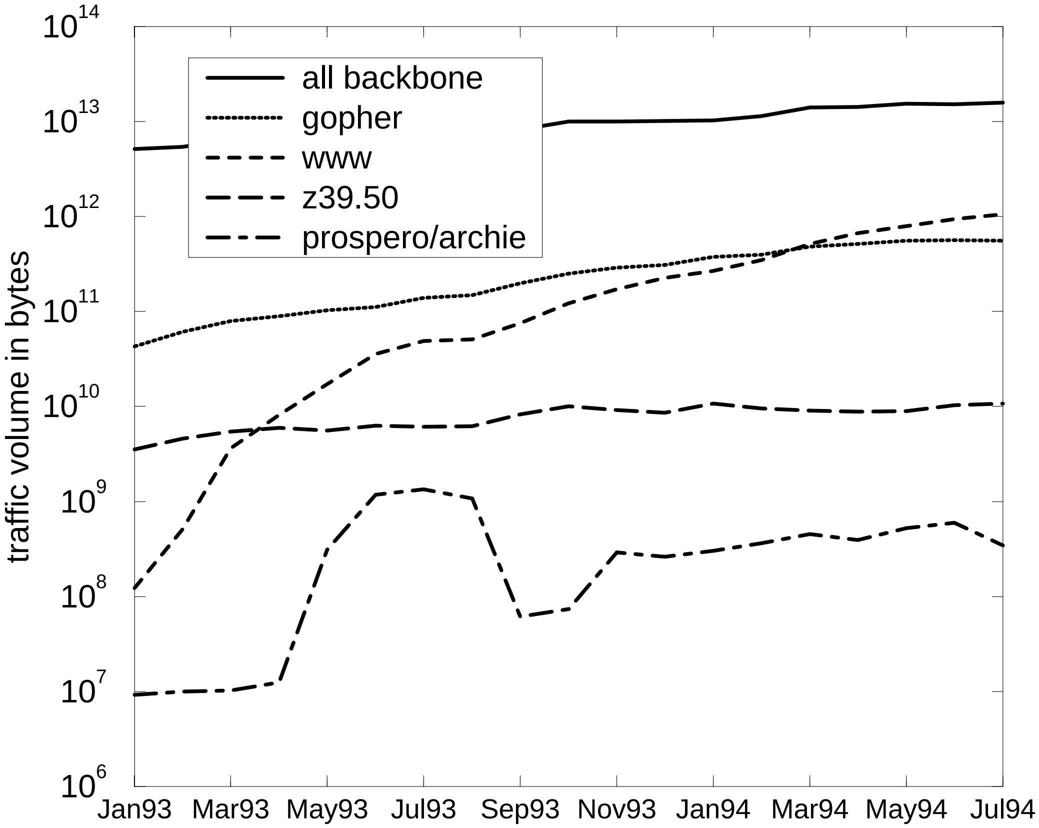

figure 1: Growth of IRDS traffic relative to overall traffic on NSFNET backbone (log scale) *(Data source: Merit, Inc. nis.nsf.net:nsfnet/statistics)*

Indeed, tracking *www* statistics on an Internet systems scale poses difficulty. A more rational architecture for *www* statistics collection at an Internet systems level would help, even more so in the face of a shift from hypertext to images in information server content which will accelerate traffic growth and likely change the characteristics of the resulting load. As such services become widespread on the Internet, we will need models of IRDS traffic load and server ability to cope. Prerequisite to such models is an understanding of the nature of the traffic from these new applications. Characterizing the service impact, including different types of server content, will require more complex metrics than simple packet, byte, and transaction counts.

In this paper we offer an example of how web workload characterization can suggest refinements to the web architecture. Specifically, we provide a geographic characterization of queries of a well-known, widely and heavily used server. Several people have written log file analysis programs for web servers [2, 3, 4, 5] but to our knowledge none has broken down the requests by geographic source within the United States.

2 approach

We examine the *www* access log files for the NCSA Mosaic server for two days: 2-3 August 1994. Although NCSA is only one of thousands of web servers, and two days only a brief snapshot of its usage, the high volume of requests that it services renders it an ideal location to investigate the potential of caching strategies and methodologies that could benefit this as well as many other servers.

The NCSA web server consists of four servers in a cluster. During this two-day period, the server cluster received 837,046 requests, for a total of approximately 14.3 gigabytes leaving the server, although from among 8,949 unique documents (approximately 400 megabytes). The wide geographic diversity of query sources present a strong case for deployment of geographically distributed caching mechanisms to improve server and network efficiency.

We first aggregate the logs from all four servers, and depict a time series of bandwidth and transaction demands for the cluster. We then break down the transaction load by geographic zones. We analyze the impact of caching the results of queries from the same zones, in terms of reduction of transactions and bandwidth from the main server. We investigate multiple timeouts for cached documents, and discuss implications for cache maintainers regarding optimal timeouts or cache configurations.

We also discuss future questions that caching inevitably poses, such as how to redirect queries initially destined for a central server to a preferred cache site. Preference of a cache site may be a function of not only geographic proximity, but also current load on nearby servers or network links. Such refinements in the web architecture will be essential to the stability of the network as the web continues to grow, and operational geographic analysis of queries to information resource servers will be fundamental to its effective evolution.

3 server statistics for 2-3 August 1994

table 1: top 25 documents from NCSA mosaic server for 2-3 August 1994

rank	requests	bytes	document
0	129917	2802	/SDG/Software/Mosaic/NCSAMosaicHome.html
1	99113	22719	/SDG/Software/Mosaic/mosaic.gif
2	41231	13612	/SDG/Software/Mosaic/NetworkStartingPoints.html
3	25442	59373	/SDG/Software/Mosaic/Docs/whats-new.html
4	22576	3115	/SDG/Software/WinMosaic/HomePage.html
5	19562	314	/SDG/Software/WinMosaic/gif/BLUEBALL.GIF
6	19110	16421	/SDG/Software/WinMosaic/gif/mosaic.gif
7	18797	11462	/SDG/Software/Mosaic/MetaIndex.html
8	18603	198	/SDG/Software/WinMosaic/gif/WHBALL.GIF
9	9132	39962	/demoweb/demo.html
10	8395	5443	/SDG/Software/MacMosaic/MacMosaicHome.html
11	7430	1028	/demoweb/sound.xbm
12	7052	7558	/demoweb/smarr-small.gif
13	6883	6169	/demoweb/al-small.gif
14	5924	1211	/SDG/Software/MacMosaic/GIFS/Mosaic.GIF
15	5807	1157	/SDG/Software/MacMosaic/GIFS/Arrow.GIF
16	5774	3964	/SDG/Software/WinMosaic/General.html
17	5530	5425	/SDG/Software/MacMosaic/GIFS/MacMosaicLogo.GIF
18	5425	2906	/SDG/Software/Mosaic/Docs/help-about.html
19	4974	3566	/SDG/Software/MacMosaic/GIFS/JurassicLogo.GIF
20	4920	7725	/General/Icons/NCSAHead.gif
21	4919	39962	/SDG/Experimental/demoweb/demo.html
22	4840	1668	/General/NCSAHome.html
23	4735	1034	/General/Icons/NewsIcon.gif
24	4633	1318	/General/Icons/InfoIcon.gif
25	4572	1921	/General/Icons/ExhibitsIcon.gif

Table 1 shows the top 25 documents from the NCSA mosaic server for 2-3 August 1994, the number of requests for each document, and the total number of bytes representing each document that the server sent. A total of 8,949 unique documents were requested during the two-day period, each between 1 and 129,917 times. The 8,949 documents together compose 398 Megabytes, the size of the "active document set" for this period. The top 25 documents were responsible for 59% of the documents requested and 45% of the bytes requested during the two-day period. These top 25 documents together compose 260112

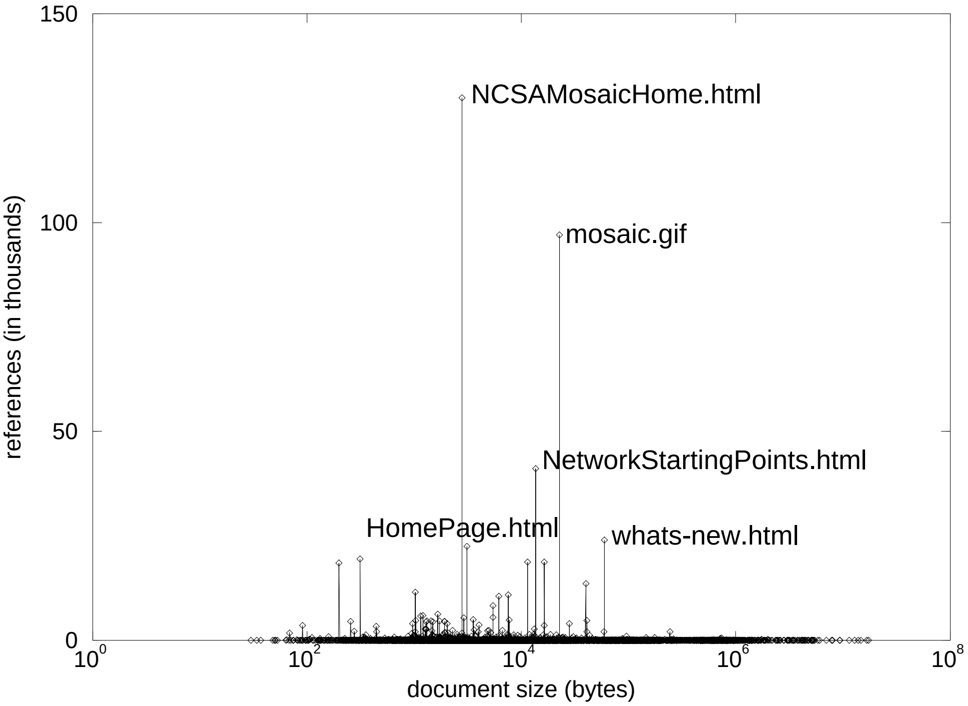

figure 2: histogram of document sizes requested from NCSA server cluster

bytes, .065% of the bytes that compose all files requested. and .0019% of the total bytes sent from the server during the two-day period. These initial statistics suggest that caching a very small portion of the NCSA server at regional sites could save a significant portion of the overall traffic leaving the NCSA server.

Figure 2 shows the distribution of retrieved document sizes. We note that the mean document size, 17 kilobytes, is not very representative since a few very large documents requested skew the mean of the distribution. (The maximum document size is over 17MB, requested only once; the 95th percentile only 57 kilobytes; and the median only 3 kilobytes.) The top 25 documents range between 198 bytes and 59 kilobytes; interestingly enough the smallest of these 25 documents is a gif image (often used as an icon) and the largest of them is the *NCSA What's New Page*. Table 1 indicates that most of the very popular files are under 10 kilobytes, fairly small relative to the sizes of other NCSA files.[1]

To assess geographic source of queries within the United States, we divide the continental US into eight zones, and use a ninth zone for Alaska and Hawaii,

---

[1] The URL of the third file in Table 1 is:
"http://ncsa.uiuc.edu/SDG/Software/Mosaic/StartingPoints/NetworkStartingPoints.html".

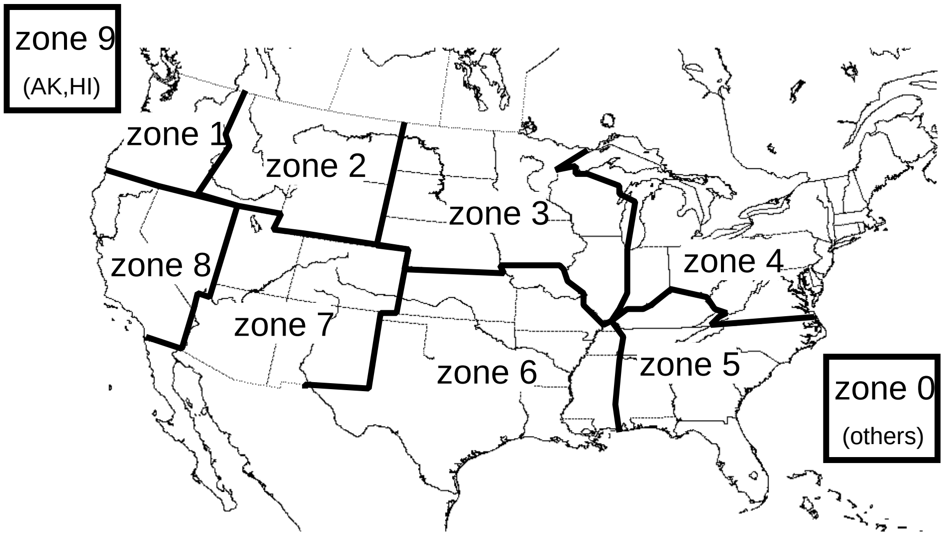

figure 3: division of United States into zones used for this study

and a tenth zone for all non-US and unknown sites,[2] according to figure 3. Table 2 shows the use of the NCSA mosaic server for 2-3 August 1994 by these zones; in the appendix we include a table of use by state.

4 trace-driven analysis

We used the two days of traces to simulate the results of caching documents locally upon their first reference to the NCSA server. Our trace-based analysis assumes that each zone maintained its own local cache during those two days. The local cache acts as an intermediary between the client and the central server; upon the first request for a document, the cache agent retrieves it from the central server and keeps a local copy. Future requests for that document within a certain timeout value, the effect of whose range we also investigate, would not require the query or document to traverse the national backbone to reach the central server. We then compute the savings such a policy offers in terms of backbone bandwidth and transactions to the central server.

We note that the benefit of a cache may not just be for bandwidth savings but for savings in demanded workload on the information server trying to process

---

[2] Individual state boundaries would be another reasonable candidate; we chose the coarser zones as more manageable for this study.

the queries, depending on the indivdual network situation. Our results apply to conservation of both resources.

A parameter of particular interest is the cache timeout, or how long a document can remain in the cache without being referenced before it is removed. Using cache timeouts of 2 to 10000 seconds, we track how many requests would require intervention from the central server, and how many the local cache could handle. We also keep track of how much memory the local cache would require to hold all documents that have not timed out yet.

We do not simulate cache replacement algorithms. Our goal is rather to demonstrate the benefits of caching and determine minimal reasonable cache size requirements. We assume that local disk for the cache is not an issue, i.e., disk storage is relatively cheap and more a matter of administrative planning. Cache contention is only an issue if the cached resource is expensive. Since one can purchase a gigabyte hard disk today for under $500, and our measurements show that much smaller caches can yield significant benefits, we did not focus on contention of this resource. As a temporary measure one can always reduce the cache timeout until additional disk space is procured. We expect however, that the timeout parameter will likely be important not for reasons of cache contention but rather to prevent documents from getting stale as new versions are placed on the original server without cache updates.

4.1 requests per minute with caches

Figure 4 shows the number of document requests per minute to the NCSA server cluster; figure 5 divides them according to the source zone of the request using the zones in figure 3. Figure 6 shows the the number of document requests per minute to the NCSA server for zone 8 given a local cache with a 4096 second timeout. The top line in the graph reflects requests for documents in the local cache, which include those requested for the first time from the central server as they are placed in the cache. The bottom line tracks this latter subset of queries, the set of documents that require transmission from the central server. The difference between these two lines illustrates the savings offered by the cache. Cache behavior for the other zones appears similar; we do not include their graphs here.

4.2 cache document timeout

We now investigate the effect of the cache timeout on the size of the cache needed and on the bandwidth savings. Figure 7 shows as a function of the cache document timeout for zone 0 the byte volume requested from the central server and the maximum size of the cache (in MB) needed for any one minute of the two-day period. The figure shows the tradeoff that the timeout poses between the need for documents from the central server and the local cache disk space requirements. Selecting an appropriate timeout involves balancing

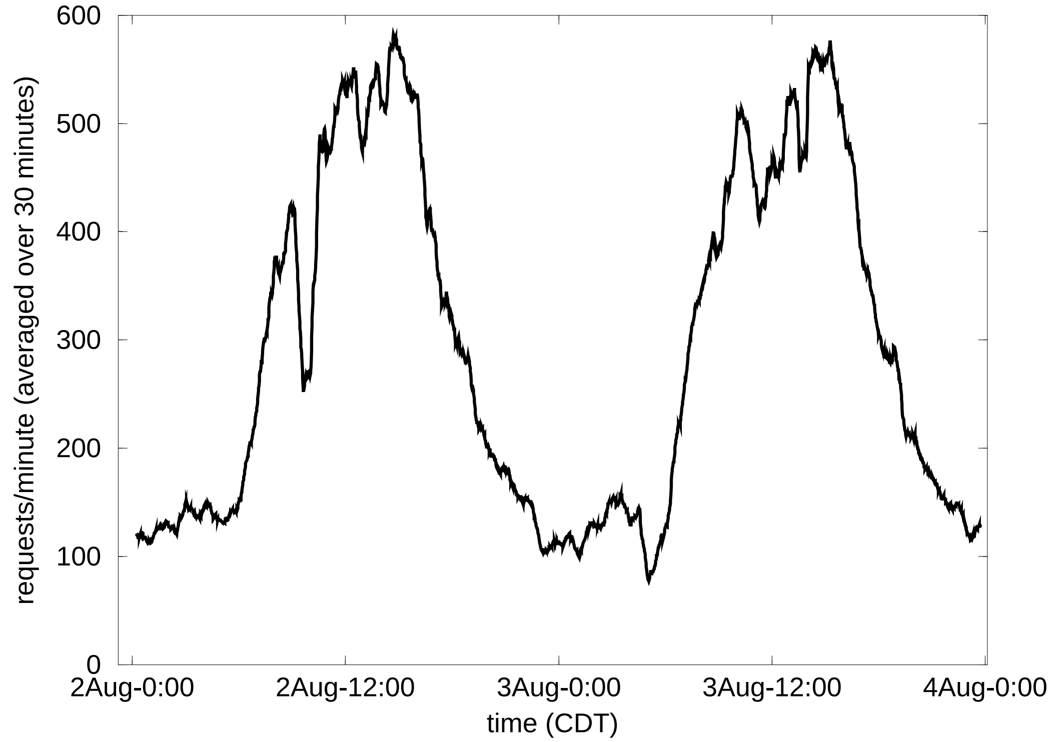

figure 4: document requests per minute to the NCSA server cluster

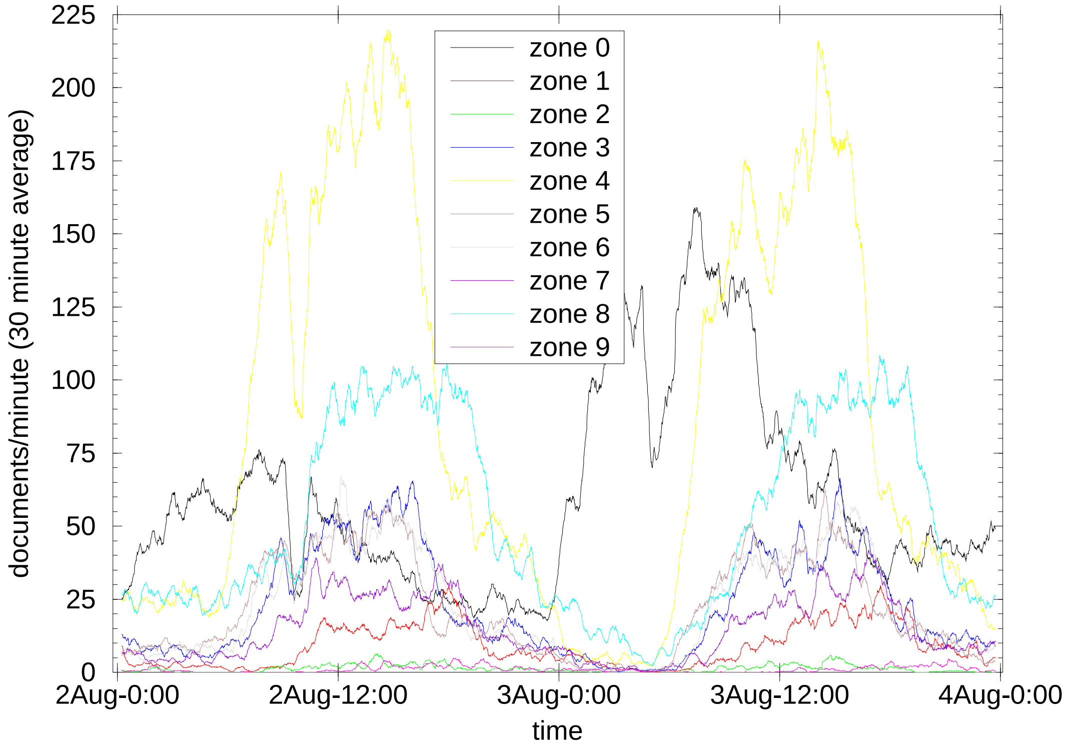

figure 5: document requests per minute from each zone to the NCSA server cluster

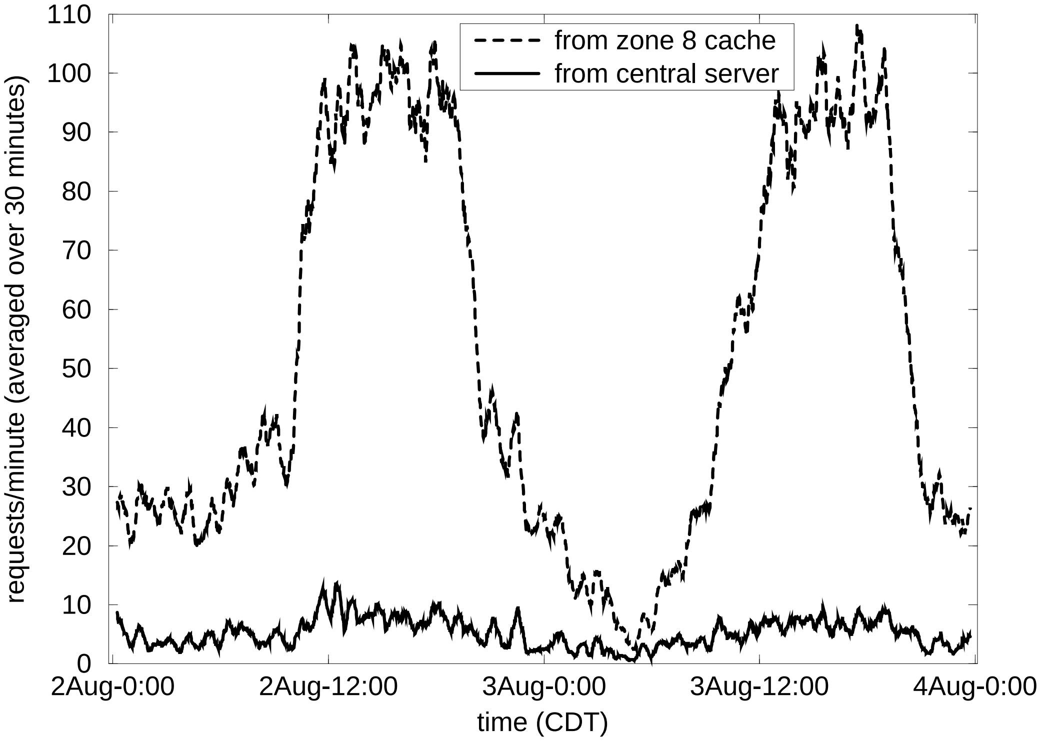

figure 6: document requests per minute for zone 8 to the local cache and the central server

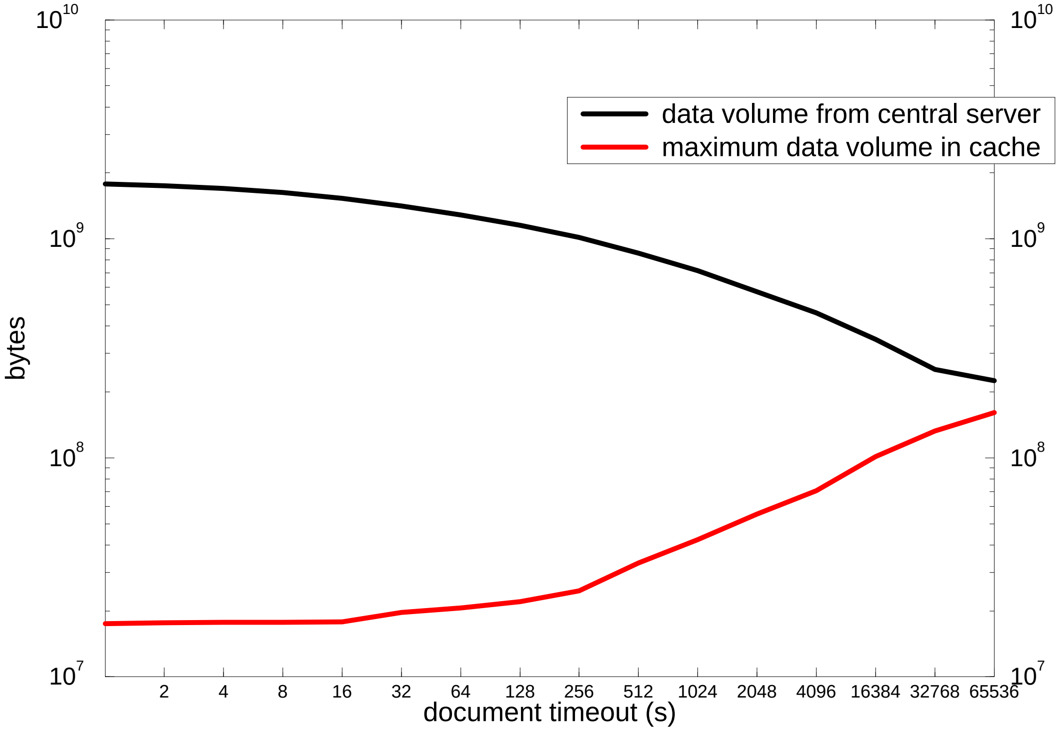

figure 7: maximum cache size and data volume from central server needed at zone 0 for various cache timeouts

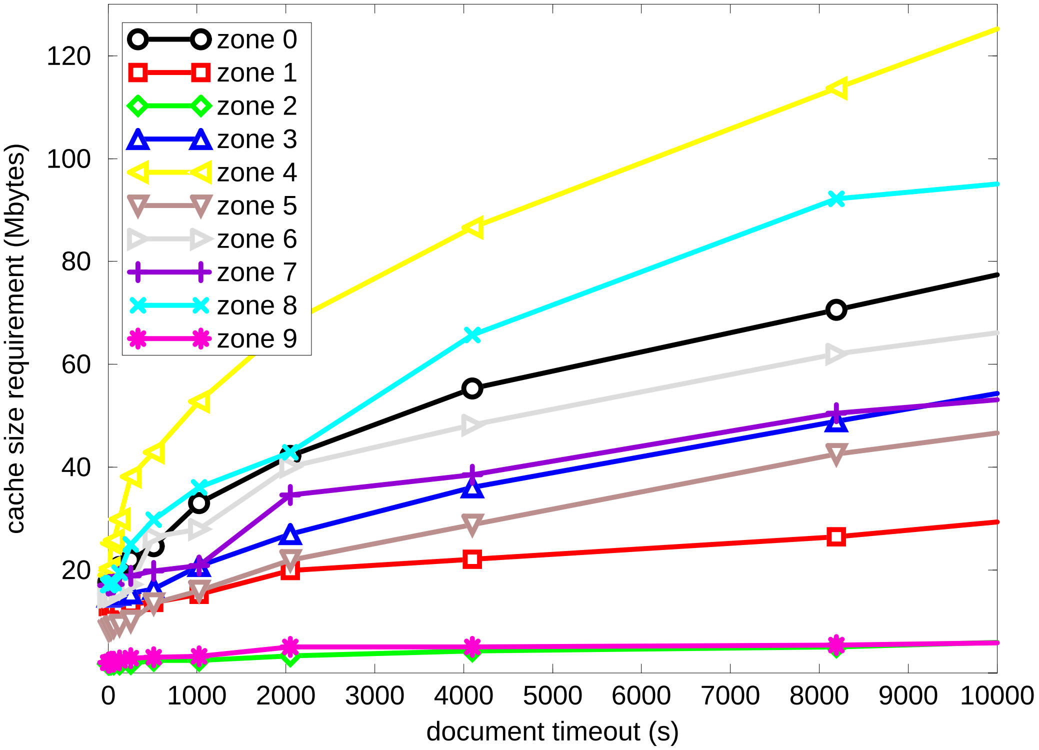

figure 8: maximum cache size needed for each zone for various cache timeouts

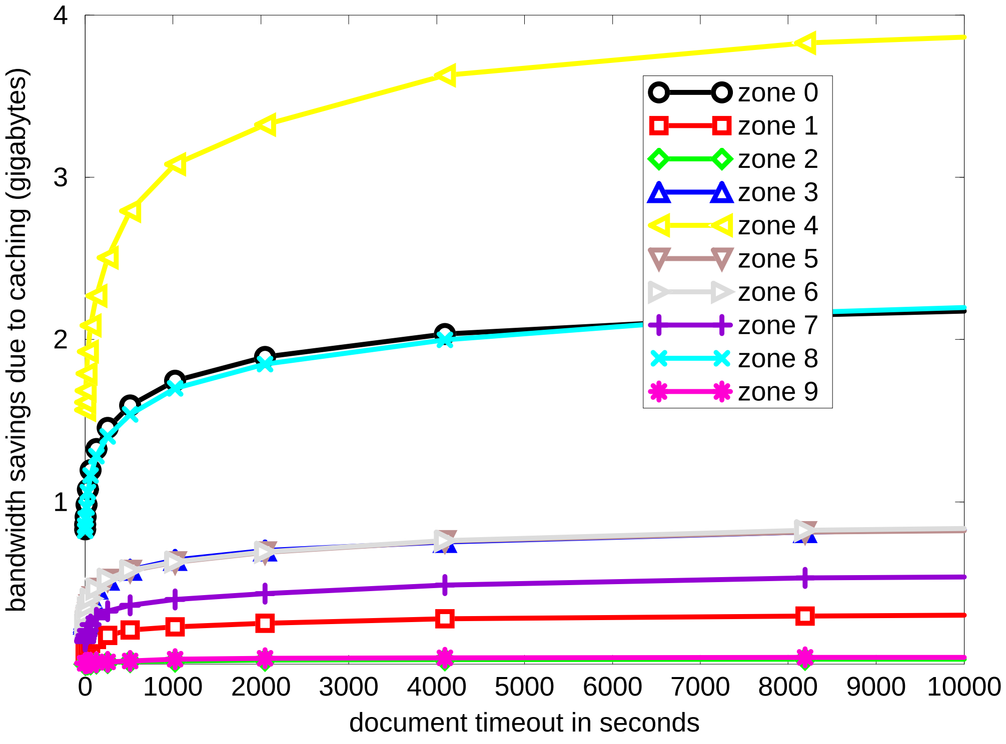

figure 9: bandwidth transmission savings due to local caching for all zones for various cache timeouts

cache management costs with the cost of using the central server. This latter cost may have several components, including delay, the cost of accessing servers that charge for usage, or the cost of using network service providers who charge for transmission of requested documents.

Figure 8 shows for all zones the maximum size of the cache (in MB) needed for any one minute of the two-day period. For this specific data set a cache of at most a few hundred megabytes would suffice for any of the zone caches, even with a timeout of several hours. In fact, increasing the timeout beyond an hour has little effect on the required cache size. In this example, the zone with the largest cache size requirement would need about 125MB of caching space. As figure 9 illustrates, these 125MB would have offered almost 4GB of savings in transmissions from the central server. Even small caches and small timeouts yield a significant benefit; the marginal benefit of larger timeouts is quite diminished. Figure 10 shows the percent of bandwidth saved through the use of caching in each zone for cache timeouts under 3000 seconds, assuming an unlimited cache size. Caching even with a document timeout as low as 1024 seconds (about 17 minutes) would have saved between 40% and 70% of the bytes transferred from the central server, depending on the zone; a 512 second timeout saves between 30% and 63% of the bytes transferred remotely. The savings occurs in not only bandwidth but processing by the central server; figure 11 graphs the analogous data as in figure 10 for transactions. A timeout as low as 512 seconds saves between 38% and 82% of the transactions to the central server. Timeouts as low as 64 seconds would have saved between 9% and 47% of the bytes and between 12% and 65% of the transactions to the centeral server.

4.3 tradeoff between cache memory and bandwidth

Figure 12 extends the results of the previous two figures, dividing the marginal savings in bytes transferred from the central server by the cache size. Minimizing this ratio for most zones given these two days of queries implies an optimal timeout is between 64 and 512 seconds.

Other considerations will also affect the preferred timeout. In particular the timeout may likely be a per-document parameter rather than a per-cache parameter, since some documents have a very low mean time between modification (MTBM) and thus caching them for more than a few seconds, or perhaps even at all, would be imprudent. These are parameters which will require considerable site-specific empirical investigation from studies such as this one. Rather than the results of this particular set of logs, we emphasize in this study the methodology and importance of web log characterization to tune caching for system efficiency.

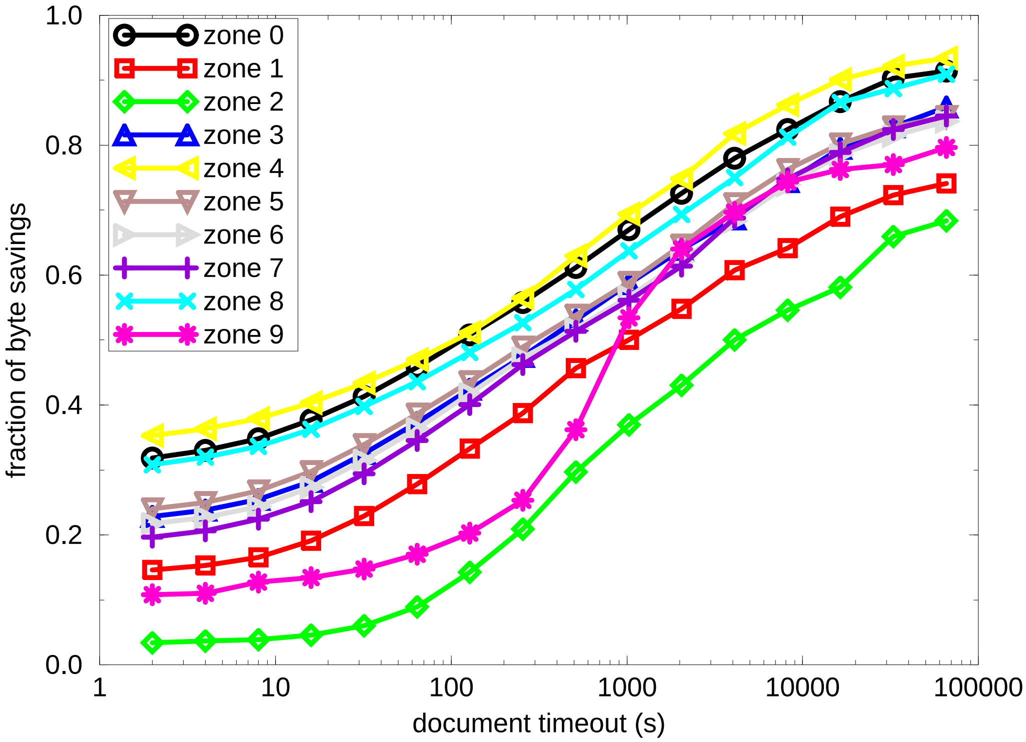

figure 10: percent of bandwidth saved through use of caches for all zones for various cache timeouts

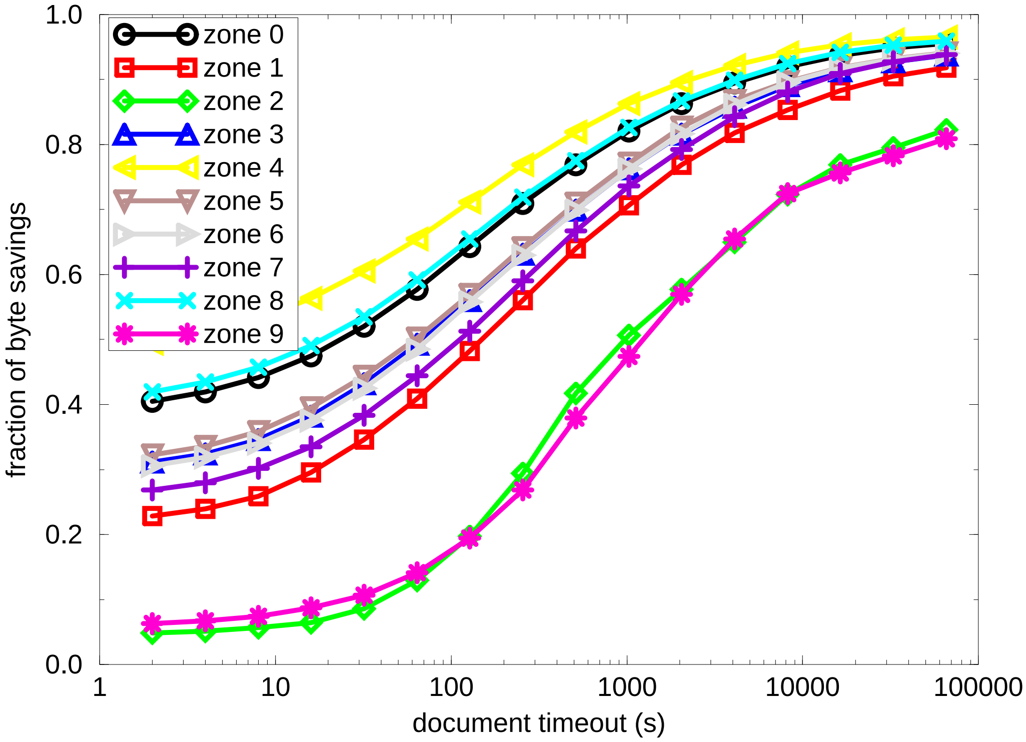

figure 11: percent of transactions saved through use of caches for all zones for various cache timeouts

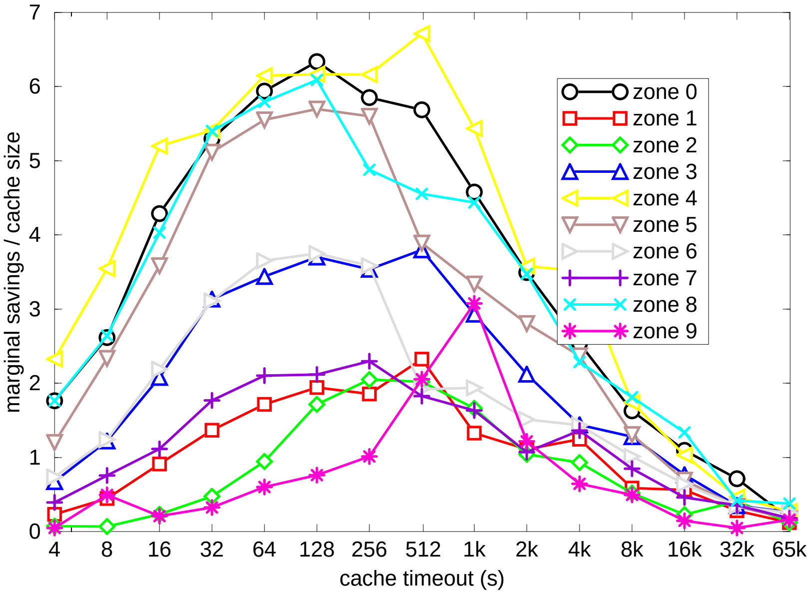

figure 12: marginal savings in bytes transferred over size of cache at each zone for various cache timeouts

5 future work

Caching is only one essential aspect of an effective *www* environment. In fact caching naturally implies several other areas of investigation that should be encouraged in development and refinement of the *www* infrastructure. Protocol and server optimization, cache consistency, security, and accounting mechanisms are all areas that benefit from insight into web traffic behavior that traffic characterization offers.

5.1 protocol and server design

IRDS client-server protocols such as HTTP (used by NCSA Mosaic and *www*) are rapidly evolving. Optimal design of servers, clients, and the protocols by which they communicate will require understanding characteristics of both the aggregate workload of many users as well as the workload of individual sessions. Other technology developments, such as distributed or mirrored servers, will also need coordination between traffic analysts and protocol designers to assess how the changes in information services will impact the Internet.

More detailed traffic characterization studies of *www* and related traffic will enable improvements in both server and caching design. A flow-based methodology for traffic characterization, discussed in [6], is particularly applicable to IRDS traffic, since the principal burden on the servers is no longer so much in terms of packet counts, but by transaction frequencies as well as the range in complexity of each transaction. Objectives in IRDS traffic characterization include finding answers to the following questions:

- what is the growth rate of IRDS traffic?
- what is the current level of intensity of IRDS traffic, both in the aggregate and from a single user?
- how bursty is IRDS traffic, both in the aggregate versus what a single user imposes?
- what is the potential impact of sessions on an IRDS server? what network resources are required to support *how many* of *what kind* of sessions?
- what is the probability of a request (These metrics require information from individual servers regarding the complexity of individual service requests.)
- what kind of instrumentation and statistics collection architecture would enable the tracking/understanding of statistics to maintain sufficient IRDS resources?

An additional issue is server performance capability. Current measurements indicate that NCSA Mosaic traffic is significantly rate-limited by the performance of the servers. More powerful hardware will handle transactions faster, but we also need to examine server behavior to determine suboptimalities. One example we discovered in this study is that the NCSA server resolves the hostname for each IP address from which it receives a request, for transaction logging purposes. The server could ease its own workload by not performing all *gethostbyaddr()* resolution system calls in real-time for each access in order to log the name of the requesting host, but rather queueing ten or twenty IP addresses and obtaining the mappings in a batch mode.

However, such advances have ominous implications. As the servers get more efficient, simultaneous with advanced client workstations that are able to request more transactions per second, the result on the system is a higher Internet bandwidth per transaction and a reduction in the mean flow interarrival rate to the servers. As a simple example, a Mosaic page with embedded images will produce a small text-only flow to a client without automatic image download options set. If the any of three independent components of the user environment: server power, client workstation power, or client network bandwidth, improves, the user will enable the automatic image download option, resulting in multiple back-to-back flows for each object that contains embedded images. Present client/server protocols such as *http* will satisfy such requests with several high-bandwidth flows back-to-back as separate TCP connections between client and server.

5.2 adaptive nameserver behavior

Automating selection of an appropriate cache will require that nameservers provide geographically oriented responses relative to the source of domain name server queries. That is, a modified Domain Name Service (DNS) should resolve an IP address destined for the "virtual" server (NCSA's in this case, although a similar architecture would apply to any cached file depositories, in fact any other distributed information service), and redirecting the source to the server/cache closest to the client. "Closest" may reflect metrics including but not limited to physical distance or number of hops from the query source.

We have already started work in creating a conversion table to map IP addresses, to the best of our ability, to geographic locations within the US; we plan to extend this to world zones. An error margin exists since not all real network locations are known, and network numbers may span large geographic areas. But such a methodology offers reasonably high probability of locating the information requestor, and thus conserving system resources by facilitating their use of a local facility.

As with cached document timeouts discussed in section 4.2, appropriate DNS timeouts for web caching may require empirical evaluation. Servers that offer real-time traffic or stock market reports may require different timeouts than servers of Shakespeare plays or Biblical verse, which presumably do not change as frequently. Integrating such intelligence about the web, and the ability to gather it, into client and management software can allow dynamic selection of optimal servers based on roundtrip time, server load, policy, or other metrics.

5.3 cache consistency

NCSA uses the Andrew File System (AFS) to maintain a single, consistent image of server information content across their server cluster. AFS provides a manageable, secure set of file namespace accessible to multiple client computers. These clients cache files locally, allowing a scalable sharing scheme that reduces network traffic and, in the case of high-performance clients, providing high-speed read/write access to file contents. Features of the management capability built into AFS could also prove useful in support of a wide-area distributed caching system.

5.4 security

Cross-authentication among caches, or multiple AFS cells, will require a security model and implementation of that model both within a site and across sites. Caches will also eventually have to accommodate more secure versions of Mosaic that selectively make documents available to authenticated clients, without exposing cached documents to unauthenticated clients.

5.5 usage-based accounting

We also expect a need to investigate how caching will affect pricing and charging methodologies, including multiple levels of service for different servers, documents, times of day, etc. [7] Such issues, and associated precedence-based pricing methodologies underlying information services, will become even more important over the next several years as commercial service providers expand their Internet presence.

6 conclusion

In this study we analyzed two days of queries to the popular NCSA Mosaic server to analyze the impact of caching the results of queries within the geographic zone from which the request was sourced, in terms of reduction of transactions with and bandwidth volume from the main server. We find that a cache document timeout as low as 1024 seconds (about 17 minutes) during the two days that we analyzed would have saved between 40% and 70% of the bytes transferred from the central server. We investigated a range of timeouts for flushing documents from the cache, outlining a tradeoff between bandwidth savings and memory/cache management costs. For the particular data that we studied, we found that balancing this tradeoff, in the form of maximizing the marginal benefit of cache memory, implied cache timeouts of between 64 and 512 seconds.

We also discuss other issues that caching inevitably poses, such as how to redirect queries initially destined for a central server to a preferred cache site. The preference of a cache site may be a function of not only geographic proximity, but also current load on nearby servers or network links. Such refinements in the web architecture will be essential to the stability of the network as the web continues to grow, and operational geographic analysis of queries to archive and library servers will be fundamental to its effective evolution.

The popularity of information resources on the Internet has already caused an explosion in bandwidth demand, leading to the need for more judicious design of server topology and information distribution mechanisms. A longer term approach may include deploying information servers/caches as well as statistics collectors for them at strategically selected locations (e.g., Network Access Points (NAP) [8]). Such colocation will offer tighter integration of network functionality and facilitate evaluation and improvement of efficiency of information servers and supporting caches. Statistics collected at cache points could also provide a basis from which to plan future cache locations, and perhaps even hierarchical information systems that can balance efficiencies among the network, the servers, and the users.

7 acknowledgements

We would like to express our appreciation for the cooperation of Nancy Yeager and Eric Katz at NCSA with obtaining the log files for our analysis.

references

[1] P. B. Danzig, R. S. Hall, and M. F. Schwartz, "A case for caching file objects inside internetworks," in *Proceedings of ACM SIGCOMM '93*, pp. 239–248, September 1993.

[2] M. Vore, "*gnlog*." http://hopf.math.nwu.edu:70/1/logging, May 1994.

[3] T. Boutell, "*wusage*." http://siva.cshl.org/wusage.html, National Institutes of Health, Grant P41-RR02188, 1994.

[4] R. Fielding, "*wwwstat*." http://www.ics.uci.edu/WebSoft/wwwstat/, July 1994.

[5] K. Hughes, "*getstats*." http://www.eit.com/software/getstats/getstats.html, May 1994.

[6] K. Claffy, *Internet workload characterization*. PhD thesis, UC, San Diego, June 1994.

[7] R. Bohn, H.-W. Braun, K. Claffy, and S. Wolff, "Mitigating the coming Internet crunch: multiple service levels via precedence," *Journal of High Speed Networks*, forthcoming 1994. available for anon-ftp at `ftp.sdsc.edu:pub/sdsc/anr/papers/`.

[8] H.-W. Braun, "Redesign and extension of multi-agency interconnection points," tech. rep., SDSC, 1992. internal document for NSF's NREN Engineering Group.

table 2: use of NCSA mosaic server for 2-3 August 1994 by zone (rank: rank in transactions with server; rankb: rank in bytes from server; total transactions includes error transactions which do not count for any bytes; zone abbreviations: **w**est, **n**orth, **c**entral, **s**outh, **e**ast, **a**laska, **h**awaii

rank	**rankb**	**zone**	**transactions**	**MB**	mean size (bytes)
total:			837046	14286	
2	3	0 (non-US)	174183	2609	14980.33
8	8	1 (wn)	26043	463	17771.03
9	10	2 (nw)	3702	58	15597.49
4	5	3 (nc)	65649	1097	16705.75
1	1	4 (ne)	244138	4439	18182.72
6	6	5 (se)	63853	1068	16730.16
5	4	6 (sc)	65102	1117	17159.08
7	7	7 (ws)	42858	710	16568.47
3	2	8 (sw)	148505	2666	17955.21
10	9	9 (ah)	3004	58	19377.45

table 3: Use of NCSA mosaic server by state/country for 2-3 August 1994

rank	**cntry**	**state**	**zone**	**docs**	**bytes**
total				837046	2147483647
1	US	CA	8	145598	2621510418
2	NA	NA	0	62252	894330476
3	US	TX	6	38925	680838262
4	US	MA	4	37717	634709347
5	US	NY	4	33381	696532822
6	US	IL	3	30332	522966957
7	US	MD	4	29721	505382000
8	US	VA	4	22851	393806738
9	US	OH	4	22851	444286542
10	US	PA	4	21476	400809313
11	CA	0	0	20382	304423183
12	US	CO	7	19111	310859261
13	US	WA	1	18335	333029825
14	US	MI	4	18009	344861584
15	US	MN	3	17692	274638632
16	US	DC	4	17381	280037564
17	US	NJ	4	16270	315045969
18	GB	0	0	15763	222711746
19	US	NC	5	13877	240078643
20	US	FL	5	13503	231911667
21	US	GA	5	12277	192537809
22	DE	0	0	10545	183008881
23	US	TN	5	10354	176007868
24	JP	0	0	10249	165921270
25	US	AZ	7	10107	163980120
26	US	WI	3	8593	145737128
27	US	MO	6	8418	131344736
28	AU	0	0	7493	107012296
29	US	AL	5	7455	124223661
30	US	OR	1	7404	124964129
31	US	NM	7	7324	134285714
32	US	IN	4	6744	109036100
33	US	UT	7	5894	94772223
34	US	CT	4	4587	70678284
35	IT	0	0	4464	64038078
36	NL	0	0	4278	62489298
37	US	KS	6	4241	82722799
38	US	NE	3	4238	70730202
39	CH	0	0	4165	72455350
40	US	SC	5	4154	67754630
41	FR	0	0	4087	76568933
42	US	OK	6	4057	72588545
43	US	LA	6	3965	64819565
44	US	RI	4	3747	65934914
45	SE	0	0	3410	50762360
46	US	MS	6	3273	54424639
47	FI	0	0	2991	53707872
48	NO	0	0	2979	39502839
49	US	DE	4	2649	65546944
50	US	ID	2	2629	42996735

table 4: Use of NCSA mosaic server by state/country for 2-3 August 1994 (continued)

rank	**cntry**	**state**	**zone**	**docs**	**bytes**
total				837046	2147483647
51	US	IA	3	2538	50623601
52	NA	NA	4	2463	38266499
53	US	HI	9	2189	38503353
54	KR	0	0	2043	11500017
55	US	NH	4	1953	31565864
56	US	AR	6	1552	19730814
57	US	KY	5	1545	26643593
58	NA	NA	8	1544	20745664
59	NZ	0	0	1528	19720119
60	SG	0	0	1492	21935222
61	US	VT	4	1469	28716615
62	US	NV	8	1363	24181781
63	IL	0	0	1359	19178537
64	AT	0	0	1323	15785004
65	TW	0	0	1115	13162151
66	BE	0	0	1088	15442134
67	MX	0	0	1050	14881967
68	ES	0	0	966	25170483
69	US	SD	3	957	12556026
70	US	WV	0	922	15916157
71	US	ME	4	869	13875845
72	IE	0	0	858	13024287
73	DK	0	0	790	9472628
74	US	AK	9	781	19466401
75	BR	0	0	772	9767507
76	NA	NA	3	705	10775217
77	NA	NA	5	688	9113164
78	NA	NA	6	671	10620744
79	US	ND	3	594	8688197
80	HK	0	0	570	5604672
81	ZA	0	0	551	19203646
82	CL	0	0	544	6049164
83	US	MT	2	537	6699578
84	US	WY	2	503	7654628
85	NA	NA	7	422	6194101
86	PL	0	0	337	3536398
87	CZ	0	0	333	4428339
88	NA	NA	1	304	4816913
89	HU	0	0	300	9318328
90	PT	0	0	255	2985816
91	RU	0	0	230	2632574
92	TH	0	0	203	3713699
93	GR	0	0	202	8526723
94	US	AE	0	189	1786375
95	US	AP	0	187	2800382
96	TR	0	0	171	4148226
97	MY	0	0	157	1503424
98	PR	0	0	139	1155487
99	IS	0	0	137	2251322
100	CO	0	0	121	11468524

table 5: Use of NCSA mosaic server by state/country for 2-3 August 1994 (continued)

rank	**cntry**	**state**	**zone**	**docs**	**bytes**
total				837046	2147483647
101	SI	0	0	116	877606
102	PH	0	0	116	1396951
103	IN	0	0	102	1565076
104	CN	0	0	94	2489674
105	LU	0	0	76	1066322
106	HR	0	0	74	1612613
107	KW	0	0	73	652688
108	ID	0	0	69	1053343
109	EC	0	0	62	540937
110	PE	0	0	60	603109
111	SK	0	0	58	502771
112	EE	0	0	55	345213
113	CR	0	0	52	645797
114	US	AA	0	43	404357
115	NA	NA	9	34	240119
116	NA	NA	2	33	390953
117	BG	0	0	28	418784
118	PA	0	0	26	1044251
119	GU	0	0	26	281927
120	AR	0	0	17	203845
121	VE	0	0	14	138786
122	LV	0	0	14	204552
123	UY	0	0	5	24820
124	VI	0	0	5	80064
125	LT	0	0	4	122656