Workload Characterization of a Personalized Web Site — And Its Implications for Dynamic Content Caching

Weisong Shi[†], Randy Wright[¶], Eli Collins[†] and Vijay Karamcheti[†]

[†] Department of Computer Science
Courant Institute of Mathematical Sciences
New York University
{*weisong,vijayk*}*@cs.nyu.edu*

[¶] NYUHome Team
Information Technology Services
New York University
randy.wright@nyu.edu

Abstract

Requests for dynamic and personalized content have become an important part of current-day Internet traffic; however, traditional caching architectures are not well-suited to cache such content. Several recently proposed techniques, which exploit reuse at the sub-document level, promise to address this shortcoming, but require a better understanding of the workloads seen on web sites that serve such content.

In this paper, we study the characteristics of a medium-sized personalized web site, *NYUHome*, which is a customizable portal used by approximately 44,000 users from the New York University community. Our study leverages detailed server-side traces of client activity over a two-week period in February 2002, obtained by instrumenting the NYUHome server. The paper presents statistics on document composition, personalization behavior, server-side overheads, and client-perceived request latencies. We then use these statistics to derive general implications for efficient caching and edge generation of dynamic content in the context of our ongoing CONCA project. Our study verifies both the need for and likely benefit from caching content at sub-document granularity, and points to additional opportunities for reducing client-perceived latency using prefetching, access prediction, content transcoding, and migrating channel generation functionality to the edge.

1 Introduction

The growing popularity of personalized Internet services, ranging from news portals to other "utility" services, has resulted in requests for dynamic and personalized content increasingly becoming an important part of current-day Internet traffic. Unfortunately, traditional solutions such as web caches and content distribution networks (CDNs) developed to improve delivery of static content do not yield the same benefits for dynamic content [18, 26].

More promising are recently proposed *object composition* approaches [3, 11, 12, 17, 20, 29, 35, 36, 39], which observe that despite multiple requests for the same site resulting in different content at document granularity, there exists substantial opportunity for reuse at the sub-document level (at the granularity of individual objects making up the overall document). Two recent studies have shown that approximately 60% of the bytes in dynamic responses from a set of popular web sites could in fact be reused from a previous retrieval of the page [34, 39].

Although encouraging, the above proposals and studies need to be supplemented with characterizations of the actual workload encountered on sites that serve dynamic and personalized content. These characterizations serve two roles: first, they provide evidence for whether or not object composition techniques are in fact required and if they are likely to be beneficial (given the specific client and content characteristics), and second, they can lend new insights into further improving delivery of dynamic and personalized content. Two recent studies [5, 30] have characterized the workloads seen by dynamic web sites, which characterize themselves by either generating the content dynamically for each request or updating their content very frequently; however, we have not seen any public literature on the characterization of personalized web sites. The latter, which allow users to customize their web pages by choosing amongst different *channels* (also called *modules*), impose additional challenges for content delivery because of the need to serve different content across the client population.[1]

In this paper, we address the above omission by studying the characteristics of a medium-sized personalized web site, *NYUHome*. NYUHome is a customizable portal to many web-based services and tools for the students, faculty, and staff of New York University, and is being widely used by the NYU community (more than 44,000 registered users). Although NYUHome is smaller and exhibits less diversity in its client population than some commercial personalized web sites (e.g., MyYahoo!), we believe that its personalization and workload characteristics are likely to demonstrate similar trends and should therefore be of interest to our research community.

Our study leverages an instrumented version of the NYUHome server, working with detailed server-side traces of client activity over a two-week period from February 13 to February 28, 2002.

Access to the server code enables us to collect information at a finer granularity than normally present in web server logs. In particular, the instrumented logs allow us to characterize, from both a server-side and client-side perspective, document composition (number, type, and TTL of channels), personalization behaviors, server-side overheads for document generation, and client-perceived request latencies. Our results show that: (1) a considerable fraction of NYUHome users do personalize their view of the site; (2) a significant fraction of document bytes are for content that is “sharable”; (3) non-sharable or personalized content is important enough for users to generate a large number of otherwise redundant requests; and (4) clients perceive request latencies over a wide range, determined primarily by their network connectivity.

We then use the above observations to derive general implications for efficient caching and edge generation of dynamic and personalized content. These implications are drawn in the specific context of our proposed CONCA architecture [35], which exploits knowledge of document structure and user access behavior to improve content delivery, but apply to other object composition techniques as well. We find that:

- Substantial benefits are likely from applying object composition techniques for personalized content, i.e., reusing content of “sharable” channels to serve subsequent requests. Additional improvements are possible by shifting channel generation functionality downstream as well.
- Both server load and client-perceived latencies can be further reduced by prefetching the content of a small number of personalized (non-shared) channels and pushing these eagerly towards the clients.
- The above optimization can be achieved in a practical fashion by identifying only a small group of clients because of the Zipf-like distribution of client popularity and personalization behaviors.
- Client-perceived request latencies can be made more uniform by specializing the document template and content, using transcoding, to the network connection employed by the client.

Note that the above optimizations only represent *opportunity* for improving performance and in practice, one needs to trade off realization costs against potential benefits. However, our analysis of the NYUHome site indicates that current response times offer sufficient leeway for most of these optimizations to be profitably realized.

The rest of this paper is organized as follows. In Section 2, we describe the structure of NYUHome and overview the CONCA architecture to provide a concrete setting for the use of object composition. Section 3 discusses the method of trace gathering, including the information gathered and the format of the logs. A detailed analysis of these logs, in terms of request distribution, user behavior, and server performance, is described in Section 4. The implications for CONCA-like architectures are presented in Section 5. Section 6 discusses related work and we summarize in Section 7.

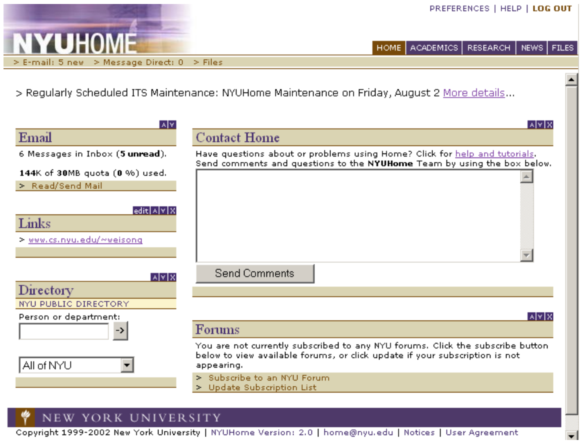
2 Background

2.1 NYUHome

NYUHome (`home.nyu.edu`) is a web portal for the students, faculty and staff of New York University

[1] In this paper, ‘personalized web sites’ refer to those sites whose content is governed by explicit user selection of channel or module preferences. We contrast this with sites that may customize or specialize their content based on implicit knowledge of user interests or access history.

(NYU) to obtain news and stock information, access their e-mail, register for courses, participate in web forums, access class pages, research tools, and more. The NYUHome screen is customizable at the granularity of *channels*, and can be personalized by different users in terms of both which channels are selected for display and their layout on the screen. The current version of NYUHome (Version 2.0) categorizes 20 channels into five tabs: `HOME, ACADEMICS, RESEARCH, NEWS`, and `FILES`. Figure 1 shows a screen snapshot of a dynamically generated and personalized NYUHome page; The default two-column template of channels in each tab is displayed in Figure 2.



> Regularly Scheduled ITS Maintenance: NYUHome Maintenance on Friday, August 2 More details...

6 Messages in Inbox (**5 unread**).

144K of **30**MB quota (**0** %) used.

Person or department:

Have questions about or problems using Home? Click for help and tutorials. Send comments and questions to the **NYUHome** Team by using the box below.

You are not currently subscribed to any NYU forums. Click the subscribe button below to view available forums, or click update if your subscription is not appearing.

Copyright 1999-2002 New York University | NYUHome Version: 2.0 | home@nyu.edu | Notices | User Agreement

Figure 1. A screen snapshot of a personalized NYUHome page (`NEWS` tab).

Implementation NYUHome is implemented using 150,000 lines of object-oriented perl and currently runs in a mod_perl environment within an Apache 1.3 server on a 12 processor domain of a 399 MHz Sun E10000 with 12 GB RAM. The system runs Solaris 2.6 in a clustered, failover environment.

Figure 3(a) shows the basic flow sequence of how NYUHome serves a client request. Authentication happens only at the beginning of each session, with subsequent requests relying on a session key. After authentication, the personal preferences are fetched to construct the layout with the appropriate channels. NYUHome relies on two object-level caches: a *shared cache* for sharable channels, and a *non-shared cache*, which stores user preferences and individual personal content. Depending on the channel TTL, content is obtained either from the caches or generated in a channel-specific fashion. For content that can be gathered periodically (e.g., `News`, `Events`), a program runs at regular intervals to populate the cache.

2.2 CONCA Architecture

CONCA (**CO**nsistent **N**omadic **C**ontent **A**ccess) [35] is a proposed edge architecture for the efficient caching and delivery of dynamic and personalized content to users who access the content using diverse devices and connection technologies. CONCA attempts to exploit reuse at the granularity of individual objects making up a document, improving user experience by combining caching, prefetching, and transcoding operations as appropriate.

To achieve its goals, CONCA relies on additional information from both servers and users. All content supplied by servers in CONCA architecture is assumed to be associated with a “document template” which can be expressed by formatting languages such as XSL-FO [41] or edge-side include (ESI) [36]. Given this information, CONCA node can efficiently cache dynamic and personalized content by storing quasi-static document templates and reusing sharable objects among multiple users. Moreover, based on the preference information provided by users, a CONCA cache node delivers the same content to different users in a variety of formats using transcoding and reformating.

3 Trace Gathering

This study leverages detailed server logs collected over a two-week period spanning February 13 to February 28, 2002. To ensure that our instrumentation did not produce unintended side effects, fewer than 10 lines of code were added at two locations: when a request was admitted into the system (`Accept Request`) and after the server was done processing a request (`Send Out`). At these points, we logged the following pieces of information:

- (at entry) The arrival time of a request, the document being requested; the source IP address; and the user ID or any session keys;
- (at end) The departure time of the response; the size of the document; for each channel present in the response, the channel ID, size, and the SHA-1 hash [1] of its contents; and for each column, the

HOME		ACADEMICS		RESEARCH		NEWS		FILES	
Email Directory Links	Contact Home Forums	Albert	Bookstore Classes	Directory Search	Library	Events	Finance Horoscope Movies News Sports Weather	Web Page	Files

Figure 2. Default channels and their layout in NYUHome.

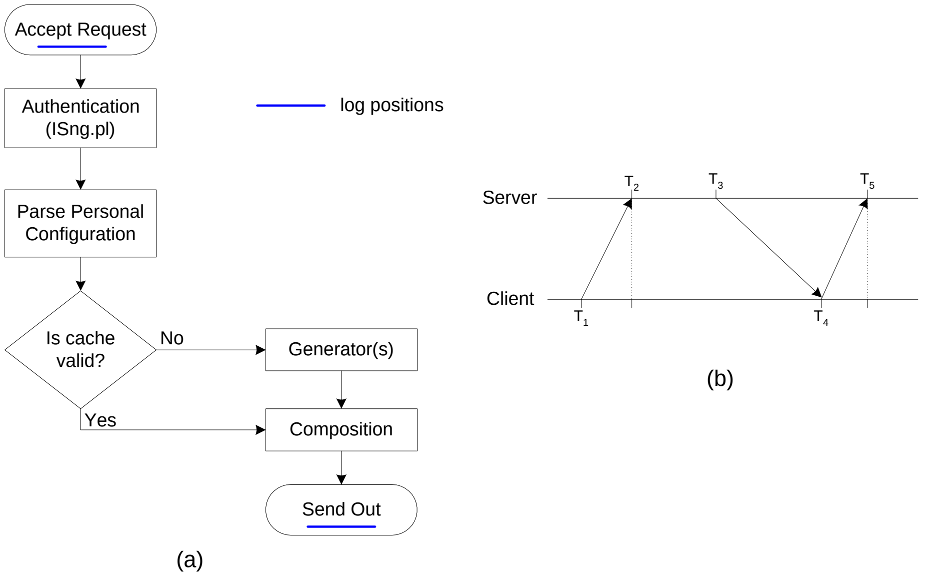


Figure 3. (a) The general flow chart to serve a request at NYUHome, and (b) time sequence of a request between client and server.

SHA-1 hash of the string obtained by concatenating the hash values of channels making up the column.

Only requests for the NYUHome main pages (served by ISng.pl) were logged. NYUHome main pages are served by NYUHome server, which is responsible for all five of the tabs, content for most of the channels, and response assembly. Requests for embedded images are served by another machine, as is the content for two hot channels: `Email` and `Albert`.

The SHA-1 hash values permit efficient computation of the change frequency of each channel and each tab. The overhead of computing the hash itself (0.30 milliseconds on the production web site) is three orders of magnitude less than the user-perceived request latency (seconds).

To estimate client-perceived request latency, we augmented each response to add a link to a blank pixel at the end of each document. When the client receives this reply, its browser will send out another request for this pixel. Recording the arrival time of this follow-on request and distinguishing it from the first request gives us three timestamps: T_2, the time the first request arrives at the server; T_3, the time the response leaves the server; and T_5, the time the follow-on request arrives. Figure 3 shows these timestamps in the context of the overall request-response timeline between the client and server. Assuming that the TCP connection establishment overheads are similar for both requests,[2]the collected timestamps allow us to estimate client-perceived latency, $T_4 - T_1$, as the time interval $T_5 - T_2$. Additionally, this interval can be divided into two components: *server processing time* ($T_p = T_3 - T_2$) and *network transfer time* ($T_n = T_5 - T_3$). Assuming that the response time dominates network costs, this decomposition allows us to correlate document content characteristics with observed costs.

[2] This assumption is true for NYUHome, which disables HTTP 1.1 persistent connections for reasons explained later in the paper.

In this estimation procedure, there are two likely sources of error. First, browsers may not wait until the entire document is received before sending out a request for the blank pixel. To assess the magnitude of error resulting from this assumption, we added another blank pixel at the head of each document and recorded the time interval between the requests for these two pixels. We found that the browser parsing time on average contributes less than 10% to the network transfer time. In the rest of the paper, the network transfer time is used to refer the network latency and the browser parsing time.

The second source of error is that client-side perceived latency ($T_4 - T_1$) may not be the same as server side latency ($T_5 - T_2$). To understand the range of error resulting from our estimation technique, we performed the following experiment. At the client side, we use `tcpdump` to record the timestamps when a user is browsing NYUHome, and calculate the client-side latency based on these timestamps. We then compare it with the server-side latency computed from corresponding logs at the server side, and find that on average the difference between these two values is less than 5%. Therefore, we believe that our estimation technique is a reasonable server-side approach to measure user-perceived latency.

The combination of user ID information (instead of inferring it from client IP address), detailed information about document composition, and the above estimation of client-perceived latency distinguish our log format from traditional web server logs. Additionally, timestamps are at microsecond resolution, which is more accurate than any publicly available server log formats, such as CLF or ECLF [24].

4 Trace Analysis

We start by presenting the overall characteristics of the trace and then analyze it from three perspectives: document composition, personalization and user behavior, and request processing overhead and latency.

4.1 Overall Characteristics

Table 1 lists the aggregate statistics from the data we collected during the two-week period. The number in the parenthesis refers to the data bytes, the rest can be viewed as carrying template information. The total number of users who accessed NYUHome during the two-week period (27,576) represent 62% of the registered users.

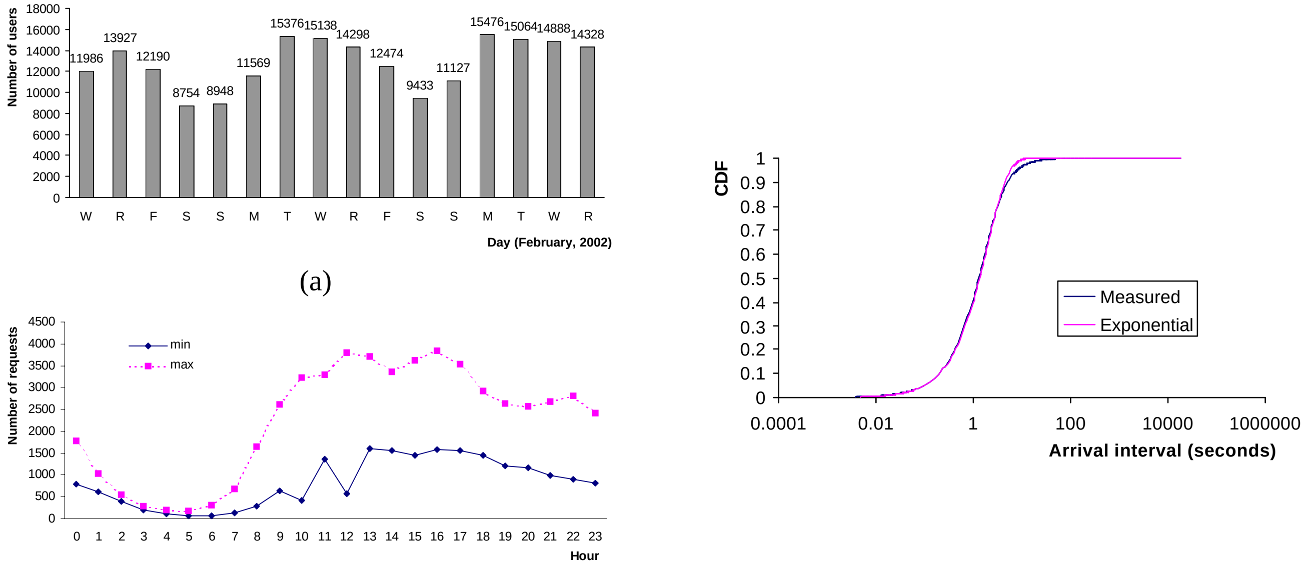
Figure 4(a) shows that, on average, NYUHome is accessed each day by 13,000 users during weekdays and 9,000 users during the weekend. On average, NYUHome received 1706 requests an hour: Figure 4(b) shows the minimum and maximum requests received during the same hour over the two-week period. Figure 4(c) shows the cumulative distribution of the inter-request arrival interval. Using the χ^2 method as the goodness-of-fit [15] measure,[3] we found that this distribution is captured very well by an *Exponential* distribution with $\lambda = 0.526$, suggesting a Poisson arrival process. This observation seemingly conflicts with that from previous studies of web servers [6] and telnet sessions [33], where it was found that the aggregate reference stream is not Poisson. Note however, that in terms of *busy documents* defined in [6] (a document referenced at least 50 times in a one-hour interval), the arrival process was indeed found to be Poisson. We ascribe our observation to the fact each document logged in the trace corresponds to a *busy* document.

To understand where these requests come from, we classified the source IP addresses into five categories: the NYU main campus, NYU Dialup (for phone modem connections), NYU-Resnet installed in student dormitories, NYU overseas campuses (at London and Florence), and other third-party ISPs. Figure 5 shows the distribution of IP addresses, and the corresponding number of requests that originate from each category. Although NYU machines contribute to only 17% of the IP addresses, they are responsible for 69% of all requests. The 60,688 IP addresses (83%) that fall outside NYU control represent varied connection options and correspond to administrative systems worldwide. Using the network-aware clustering technique proposed in [25], these IP addresses can be grouped into 4183 network clusters, where 109 clusters have more than 50 IP addresses and 60 clusters have more than 100 IP addresses. In Section 4.4, we correlate the measured client-perceived latency with the IP address category a request corresponds to.

[3] The Anderson-Darling (A^2) test was also used in our analysis, however, the results of the A^2 test showed no significance in terms of goodness of fit for large amounts of data, which is a common problem of A^2 [9, 32]. The χ^2 method is used as the goodness-of-fit measure in the rest of the paper.

Total # of requests	Total # of users	Total # of IP addresses	Total # of sessions	Total # of bytes (Mbytes)
643,853	27,576	73,119	520,408	6,533 (2,434)

Table 1. Aggregate statistics of the NYUHome log from 02/13/2002 to 02/28/2002.



(a)

(b)

Figure 4. (a) The number of users who access NYUHome each day; (b) the number of requests arriving each hour; and (c) the inter-request arrival interval distribution.

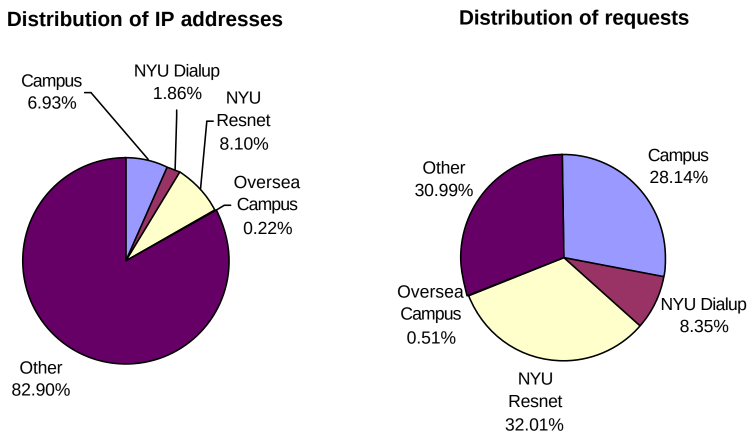


Figure 5. Distribution of IP addresses across five categories and the corresponding breakdown of request traffic.

4.2 Document Composition

To understand the characteristics of documents that were generated in response to client requests, we analyzed the properties of component tabs and individual channels.

Tabs Table 2 lists the number of requests to each tab, the number of users who accessed the tab, and the average number of requests per user (the ratio of the first two values). 90.1% of requests are for the default HOME tab. Of the requests to the other four tabs, 50% of requests from 7,148 users are for the ACADEMICS tab, which includes the course system Albert. On a per-user basis, after the HOME tab, the NEWS tab is the next popular. Table 2 also shows the average size of the document generated in response to a tab request, and the fraction of the response bytes taken up by the template (between 30% and 66%).

Channels Figure 6 shows, for each channel, the number of requests that involve the channel. The figures also show the percentage of total document bytes contributed to by the channel. Variations in the number of requests for channels that belong to the same tab, particularly prominent in the NEWS tab, are a direct result of personalization.

Figure 7(a) shows the cumulative distribution function (CDF) of the distribution of channel size. We find that the channel sizes, 99% of which are smaller than 3000 bytes, are best modeled using a *Weibull* distribution (with CDF $F(X) = 1 - e^{-(0.0012x)^{1.6}}$). This observation is in agreement with a previous study on six news and e-commerce web sites [34]. It is interesting to compare this distribution with the overall document-size distribution in Figure 7(b). The latter shows that 70% of the documents lie in a very small range between 9,725 and 10,688 bytes. The popularity of the HOME tab and the fact that the template accounts for a sizeable fraction of the overall document size explains this phenomenon.

Table 3 lists average, minimum, and maximum sizes of all the channels, by decreasing order of average size. Also shown is the number of distinct hash values generated for the channel content, both during the whole period and during the busiest day (February 25). The number of distinct hash values for a channel indicate how sharable a channel is, dependent both upon its time-to-live (TTL) and the nature of its content. This qualitative notion of sharability is shown in the last column of Table 3, which marks channels suitable for reuse with a 'Y' and others with an 'N'. For instance, channels such as Library, Search, and Bookstore are essentially unchanging over the entire duration of the trace; channels such as Events, Movies, Sports, and News are sharable and change only infrequently (at most a few times a day); channels such as Horoscope, Weather, and even Classes, are not sharable, but given reasonable-sized client populations are in fact shared. Finally, channels such as Email, ContactHome, Albert, and Directory are truly personalized (the last three because of per-user history) as reflected by their large number of distinct values.

4.3 User Behavior and Personalization

To understand the behavior of a particular user (associated with a particular user ID), we examined session statistics, client popularity, and personalization characteristics.

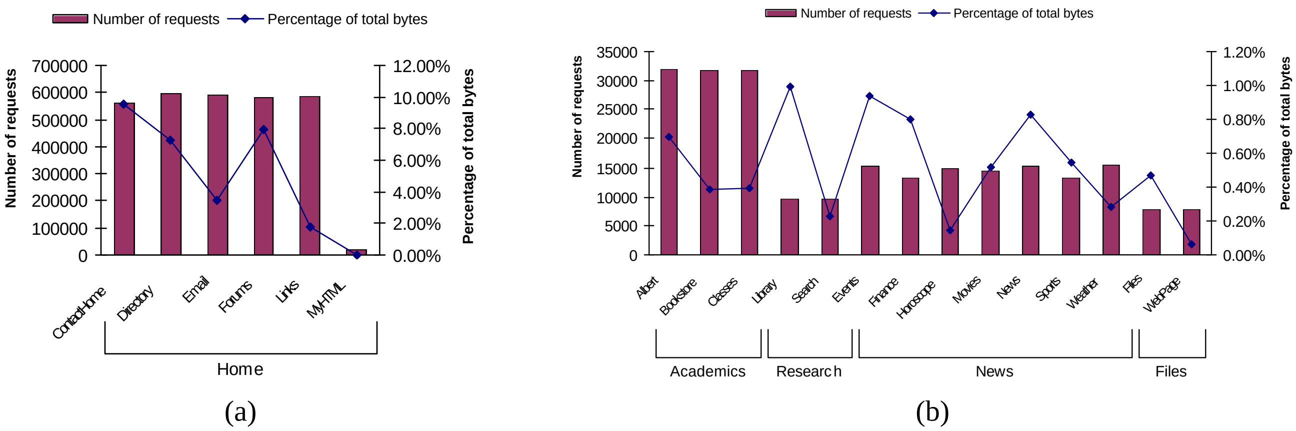
Sessions Figure 8(a) shows the cumulative distribution function of the number of requests per session (defined as the requests accompanied by the same session key). 82.85% of sessions contain one request only, for the default HOME tab. For sessions with multiple requests, Figure 8(b) shows the inter-request arrival interval. The mean inter-request time is 492.7 seconds, and the median is 92.9 seconds. Such relatively long inter-request intervals are the major reason NYUHome has disabled HTTP 1.1 persistent connections.

The concept of inter-request interval within a session is the single web site version of the “Inactive OFF” time between successive requests; however, our observation differs from previous studies that have characterized OFF times using a heavy-tailed Pareto-like distribution [9, 13]. In contrast, we find that the session inter-request intervals are captured best by a *Lognormal* distribution with $\mu = 4.5, \sigma = 2.2$, without a heavy tail. We ascribe this difference to the typical behavior of users in regards to a single personalized portal site, where a user may spend time on pop-up windows (such as checking e-mail), before going back to click other tabs.

Client Popularity Figure 9 shows the cumulative distribution function of the number of requests per user:

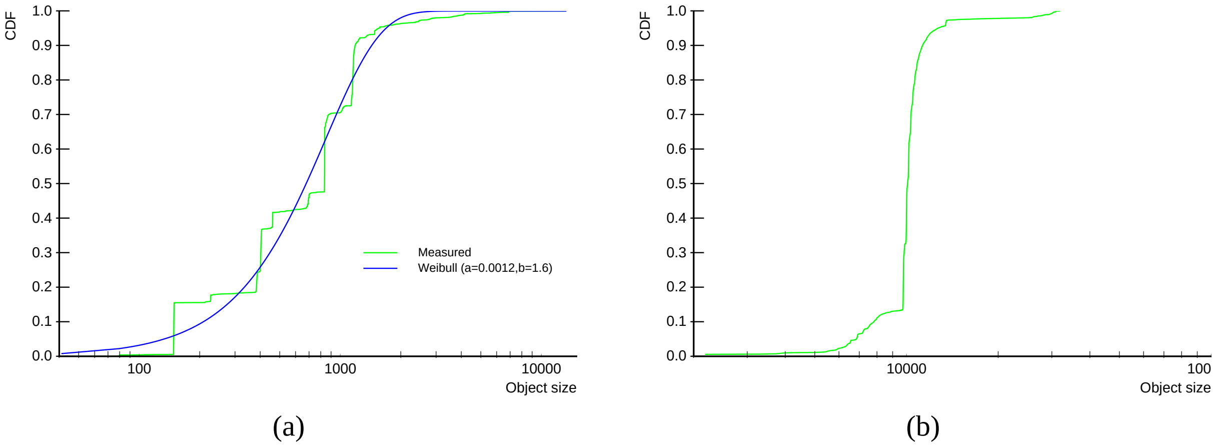
Tab	HOME	ACADEMICS	RESEARCH	NEWS	FILES
Number of requests	598,585 (90.1%)	32,229	9,873	15,595	7,927
Number of users	27,576	7,148	3,584	2,988	2,200
Requests per user	21.70	4.50	2.75	**5.22**	3.60
Average tab size	10024.65	7048.05	13374.59	26810.22	7052.34
Average template size	6611.18	3938.73	4169.09	9080.06	2506.19

Table 2. The number of requests to different tabs and corresponding number of users, and the average tab size and template size for each tab .



(a) (b)

Figure 6. Distribution of the number of requests to different channels in the (a) HOME tab; (b) other tabs.



(a) (b)

Figure 7. The cumulative distribution function of: (a) channel sizes, and estimated Weibull distribution, and (b) document sizes.

Channel	Average size (bytes)	Minimum size (bytes)	Maximum size (bytes)	# of distinct hashes	# of distinct hashes (02/25)	Sharability
Library	6860	2672	6887	3	2	Y
Events	4255	217	6391	19	1	Y
Files	4075	63	26064	7309	508	N
Finance	4062	68	4267	59	7	Y
News	3649	849	3965	406	28	Y
Sports	2777	68	2940	256	20	Y
Movies	2432	68	2524	41	5	Y
Search	1573	1565	1573	7	4	Y
Albert	1482	1482	1482	30020	1879	N
Weather	1251	41	3814	467	32	Y
ContactHome	1159	1102	2010	163808	20527	N
Forums	934	140	36778	10038	5631	N
Classes	845	214	11204	2135	641	Y
Directory	836	833	838	27273	14055	N
Bookstore	830	800	845	4	1	Y
Horoscope	664	366	884	221	19	Y
WebPage	523	497	540	2240	275	N
Email	398	303	840	396892	32343	N
Links	206	79	3860	1999	1078	N
MyHTML	129	80	2393	346	174	N

Table 3. The size information of channels, including average, minimum, and maximum sizes, and the number of distinct values of channels during the whole period and the busiest day (02/25/2002).

57% of the users send less than 15 requests during the two-week period (one request/day on average), however, 5% of users send more than 90 requests (six requests/day on average). As in [5], we studied the relationship between the rank of users (based on the number of requests he or she issues) and the corresponding number of requests. Users who issue the most number of requests are assigned rank 1. If client popularity follows a Zipf-like distribution, the log scale plot should appear linear with a slope near $-\beta$ [8]. Figure 9(b) shows that the popularity of clients does follow a Zipf-like distribution for the top 2000 users with $\beta = 0.35$, but does not fit as well for users who issue fewer than 50 requests over the two-week period.

Personalization To understand how many users personalize their NYUHome pages and how, we calculated the distinct channel combinations (including both channel selection and layout options) for each tab, and then counted the distinct number of users who used a particular channel combination. The relative statistics are shown in Figure 10. The pie graphs in Figure 10(a) show the comparison between the percentage of users who use the default channel selection and layout and those who personalize one and/or the other. With the exception of the `ACADEMICS` tag, there was significant customization. The numbers are also likely to have been biased towards the lower end by the fact that a significant number of users likely use the `HOME` tab only to check their e-mail.

Figure 10(b) shows that the number of requests that are targeted to these different channel combinations are compatible with the user fractions. More interesting is the observation that in the four tabs where personalization occurs, a considerable percentage (about 30%) of the requests are for a channel combination that differs from the default only in its layout, not in the set of channels. Another statistic that does not come across in the graphs is that users are interested in other forms of personalization as well: 19% of the requests for the `NEWS` tab involve 40 channel combinations different from either the default or the simple layout exchange.

An additional observation pertains to how frequently users change their personalization preferences. In our two-week trace, this number was relatively small: only

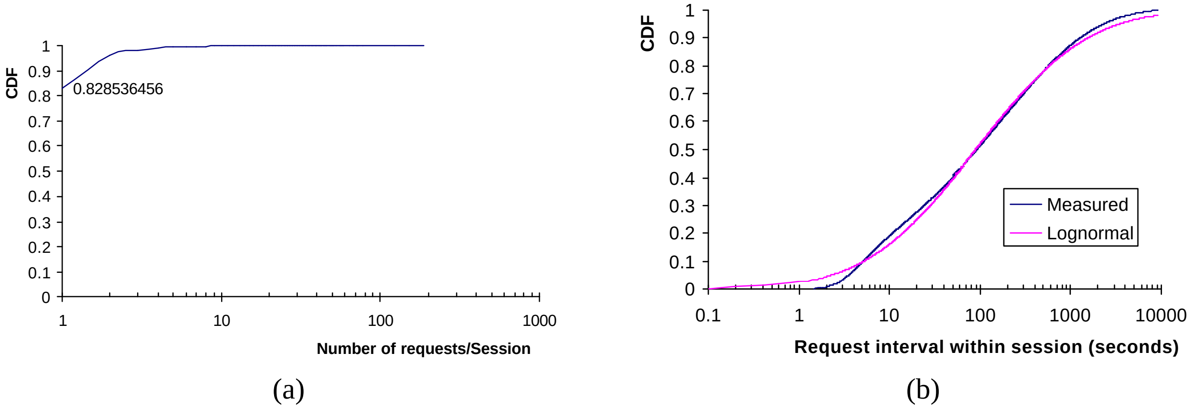


Figure 8. The cumulative distribution function of (a) the number of requests per session, and (b) the inter-request interval within a session.

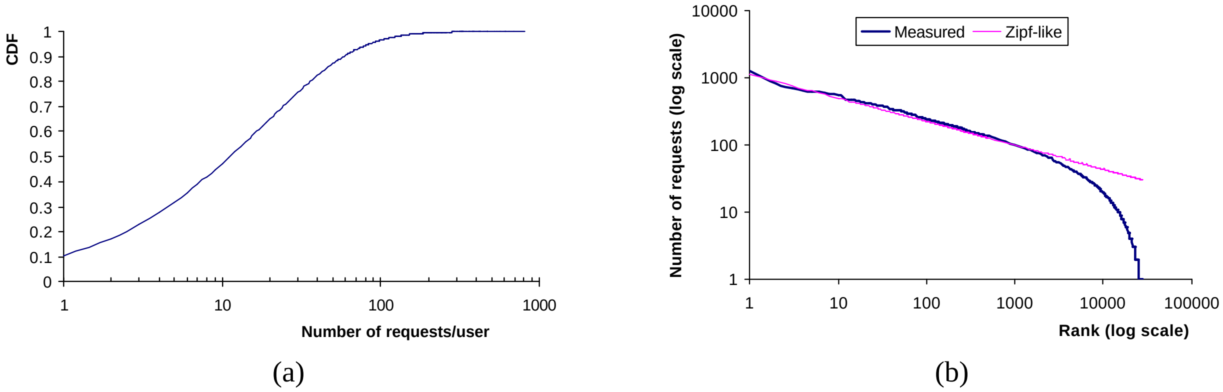


Figure 9. Concentration of the number of requests from client perspective: (a) cumulative distribution function of the number of requests per user; (b) client popularity (request count vs. rank).

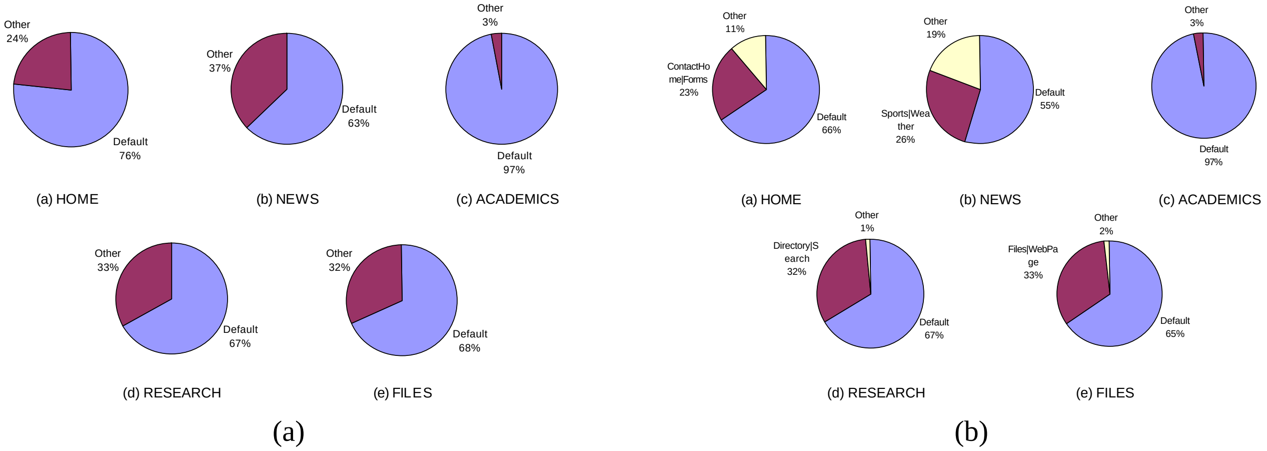


Figure 10. Effect of personalization: (a) the percentage of users who use personalization functionality; (b) the distribution of the number of requests to different channel combinations of each tab.

1,090 of the 27,576 users changed their preferences at least once.

4.4 Request Processing Cost and Latency

Table 4 shows, for the overall trace as well as for individual tabs, the average, minimum, and maximum values of the per-request processing time, T_p, and network transfer time, T_n, computed as described in Section 3. The lower half of Table 4 lists the network transfer times for the five categories of IP addresses identified in Section 4.1.

Processing Overhead The average server processing overhead across all of the requests is 1.41 seconds; 28% of requests incur overheads larger than this value. To understand whether there is a relationship between server load and request processing overhead, we looked at the average processing time seen by requests on the least busy (02/16) and the busiest day (02/25) in our trace. Figure 11 shows the processing overhead and number of requests received by the server on an hourly basis on these two days. We conclude that the average processing time is independent of load,[4]and reason that the NYUHome server is operating far below it's planned capacity most of the time. Consequently, the high server processing overheads represent the inherent overhead associated with dynamic generation of personalized content.

To understand the primary contributors for this processing overhead, we computed the correlation coefficient between the overhead and the number of channels in and the overall document size of the response. We found a strong correlation, 0.98, between the number of channels and the processing overhead, which explains the lower average T_p values for the `ACADEMICS` and `RESEARCH` tabs. On the other hand, the correlation coefficient for the relationship between document size and processing overhead achieved a value of 0.044, indicating the lack of any significant correlation between the two.

To understand the details of server processing overhead, we propose a simple model which involves the number of channels N and three types of per-channel cost — t_c for obtaining content from a cache, t_g for generating the content synchronously, and t_a for assembling the content into a document. Assuming the number of channels served from cache is n_c, and the number of channels generated by server dynamically is n_g, where $n_c + n_g = N$, the total processing overhead can be modeled as $T_p = n_c \times t_c + n_g \times t_g + N \times t_a$. Using the average T_p of different tabs, we end up with the following relationships: $t_g + t_a = 0.523s$ and $t_c + t_a = 0.329s$. Thus, we find that generating a channel synchronously incurs an additional average overhead of about 0.2 seconds.

Transfer Time and Throughput The average network transfer time was observed to be 2.45 seconds, with 27% of the requests resulting in larger times and 15% of the requests spending more than 5 seconds in the network. Looking at throughput, defined as the ratio of document size and the network transfer time, we find a mean value of 30 KB/s and a median value of 13 KB/s. Both network transfer time and throughput are captured well by the *Lognormal* distribution with $\mu = 0.005$, $\sigma = 1.55$ (transfer time) and $\mu = 9.35$, $\sigma = 1.6$ (throughput) respectively. Our finding of throughput coincides with earlier observations made by Balakrishnan et al. using traces from the 1996 Atlanta Summer Olympic Games web server [7].

To identify the primary contributor to network transfer time, we again computed the correlation coefficient between transfer times and document sizes. The result, -0.0031, reveals that in general there is no clear correlation between the two. A stronger correlation was observed when we separated out the network transfer time based upon the category of IP address a particular request belongs to. The five categories from Section 4.1 correspond to two (Campus and NYU-Resnet) with good LAN-like connectivity, one (Overseas) with WAN-like connectivity, one (NYU Dialup) with phone modem connectivity, and the remaining (Other ISPs) that correspond to varied connectivity options ranging from ADSL, cable modems, to phone modems. As one might expect, the faster connectivity options result in lower transfer times and better throughput, while the slower connectivity options see degraded performance. On average, users who access NYUHome using phone modems (NYU-Dialup) encounter five times the network transfer time and 1/20th the throughput of those accessing NYUHome from campus.

[4] The bursts at 3:00am on 02/16, 12:00pm on 02/25, and 1:00pm on 02/25 occur because of backup operations and a restart of the session manager respectively.

Tab	Average T_p	Min T_p	Max T_p	Average T_n	Min T_n	Max T_n[a]	Throughput (KB/sec)
Total	1.41	0.05	487.75	2.45	0.002	19.74	30.24
HOME	1.44	0.05	487.75	2.51	0.01	20.29	29.93
ACADEMICS	0.66	0.06	45.39	1.61	0.002	12.43	19.30
RESEARCH	0.48	0.06	23.31	2.07	0.04	12.21	31.76
NEWS	1.92	0.06	224.01	1.73	0.03	14.16	67.62
FILES	2.07	0.05	46.19	1.73	0.04	12.32	19.17
Campus	—	—	—	1.35	0.004	9.35	38.54
NYU Dialup	—	—	—	7.43	0.9	42.19	2.26
NYU-Resnet	—	—	—	1.02	0.002	8.15	43.02
Overseas	—	—	—	2.24	0.34	9.50	8.35
Others	—	—	—	3.92	0.01	28.51	17.38

[a] Max T_n refers to the 99th percentile value: a small fraction of requests involve file and web page upload/download and can incur transfer times of several minutes.

Table 4. The average, minimum, and maximum values of server processing overhead and network transfer time, and the average throughput.

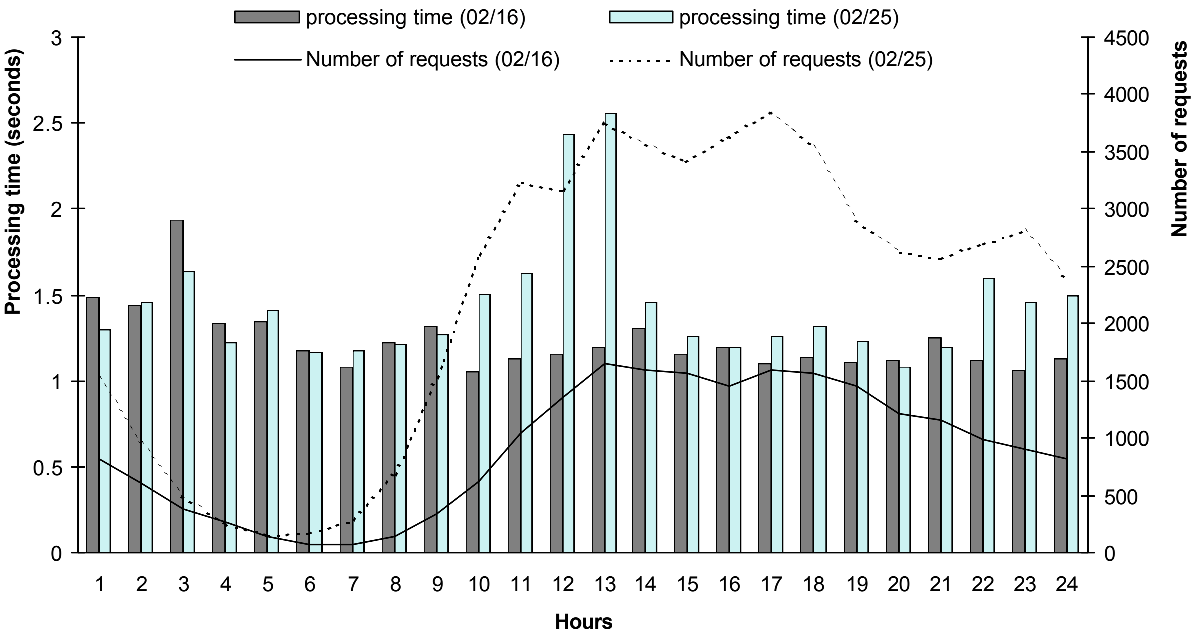


Figure 11. Processing overhead comparison between the least busy (02/16) and busiest (02/25) day.

5 Implications for Dynamic Content Caching and Delivery

The analysis of the NYUHome traces points to both the need to improve delivery of personalized content, and the opportunity for leveraging various solutions at the server-side and on surrogates or proxy caches to address this need. We discuss these implications below:

1. Need for efficient delivery of personalized content
Our study has shown that six years after the introduction of the concept of personalized web sites [27], a substantial fraction of users are using the concept — 30% in our case, and larger if one accounts for the fact that a large fraction of NYUHome users use it only to check email. However, this situation comes at the cost of increased server overheads that are several times larger than that seen for static content or even non-personalized dynamic content, and larger network transfer latencies. Together, these two factors contribute to client-perceived latencies of several seconds.

2. Effectiveness of server-side fragment caches

Although NYUHome relies on simple fragment-based caches, our observations show that these by themselves are not sufficient to reduce per-request processing overheads. Accessing the cache and assembling the content incurs per-channel overheads of about 0.33 seconds (see Section 4.4). Generating channel content incurs an additional per-channel overhead of 0.2 seconds.

More efficient server-side caching schemes such as DUP [11, 12] are likely to yield better performance, as are schemes which cache partial responses in addition to per-channel content and can use these to incrementally construct the full response. The latter are particularly well-suited for requests that refer to one or more default selection and layout of channels. As we found in Section 4.3, a significant fraction of all personalization takes the relatively simple form of only layout modification.

3. Potential for and likely benefits from using the object composition technique
Object composition techniques as in our CONCA architecture, advocate caching of channel content at surrogates and proxy caches; requests from clients are forwarded to servers only to download missing channels, which are then assembled into a response sent back to the client.

We observe that among the eleven NYUHome channels with average length larger than 1 KB (see Table 3), six of them — `Library`, `Events`, `News`, `Sports`, `Movies`, and `Search` — are completely sharable; one — `Finance` — has a large portion that is sharable; and one more — `Weather` — although not completely sharable can be effectively shared amongst users that share interest in the same zip code. Combined with the fact that over 60% of requests refer to the default layout of a tab (see Figure 10), and that the tab layout template, which can be cached as well in CONCA-like architectures, contributes to anywhere from 30% to 66% of the transmitted bytes (see Table 2), the object composition technique can yield significant bandwidth savings and reduction in server processor overheads.

Taking the `HOME` tab as an example, let's examine the potential bandwidth savings by downstream proxy caches. The average size of this page is 10,000 bytes, and it includes 5 channels: `Email` (398), `Contact Home` (1158), `Directory` (835), `Forums` (934), and `Links` (206). The number in the parenthesis refers to the average size of each channel in bytes and is taken from Table 3. The size of the layout template is 6,469 bytes (calculated by subtracting the sum of channel sizes from the total page size). Assuming a request is received and authenticated at a proxy cache, which has already cached the template and four channels except the `Email` channel, what the proxy cache needs to do is send a request for the latter to the server. As such, the number of bytes that must be transmitted between the NYUHome server and the proxy cache reduces from 10,000 to 398 (96% are saved).

Additional savings are possible by redesigning tab layouts so as to separate out sharable channels from those that are truly personalized. From tables in Section 4.2, the large number of `HOME` tab requests and its correspondence with the number of distinct hash values seen for the `Email` and `ContactHome` channels seems to suggest that users may be loading the tab primarily to track changes in the contents of the personalized channels. Modifying the tab layouts can help avoid the need to transmit unnecessary channels.

4. Benefits from proxy prefetching and/or server pushing

Although several NYUHome channels are sharable, a sizeable fraction (40%) do refer to truly personalized content, and therefore do not benefit from caching of channel content at surrogates or proxy caches. More suitable solutions for channels such as `Email`, `ContactHome`, `Links`, and `Forums`, involve either the proxy cache prefetching the content from the server or the server pushing the content upon detecting an update.

Such eager propagation of content can avoid unnecessary downloads — a large fraction of the difference between the total number of requests involving the `HOME` tab and the number of distinct hash values for the `Email` channel likely fall into this category. Additionally, as seen in Figure 8, the interval between successive requests in a session is large enough (on the order of several minutes) to permit use of sophisticated prefetching policies.

5. Benefits from predicting access patterns
To allow prefetching schemes such as the ones described above to be practically employed in personalized web sites, the conflicting demands of personalization and prediction need to be reconciled. In other words, for prefetching to be successful, we need to predict access patterns of individual users, which is likely to result in prohibitive space and time overhead. Fortunately, the Zipf-like distribution of client popularity (see Figure 9(b)), which indicates that a small number of users are responsible for most of the requests, suggests a solution to this problem. By focusing on predicting the access patterns of only the users who make the most requests, overheads of collecting and exploiting access pattern information can be made manageable.

6. Need for migrating channel generation functionality to edge servers
Server overheads for request processing, observed to be strongly correlated to the number of channels in the document, can be reduced by offloading channel content caching and content assembly to proxy caches. However, as we find in Section 4.4, generating channel content incurs an additional per-channel overhead of 0.2 seconds, implying that additional improvements are possible by shifting channel generation functionality downstream as well. Clearly, this choice needs to be traded off against the cost of maintaining consistency between the state at the server and that at edge servers. For channels such as `Classes` and `Forums`, generated from read-mostly data, migration of the channel generation code may be an attractive option.

7. Need for customizing content based on network connection characteristics
The network transfer times reported in Table 4 show that there are wide variations in the latencies seen by different groups of IP addresses based on their connectivity characteristics. To provide a uniform user experience across multiple device types and network connections, one might imagine defining different default layouts and channel content for each class of device or network. Taking the example of the `NEWS` tab and `NYU-Dialup` users, the latency perceived by clients can be lowered by reducing the number of channels in the tab, by reducing the amount of content in each channel (e.g., the `News` channel can incorporate fewer headlines), and by changing the formatting to reduce the fraction of document bytes devoted to defining the tab layout template.

6 Related Work

Web workload characterization has been extensively studied in the past five years from the perspective of proxies [10, 16, 38, 40], client browsers [2, 8, 13, 14, 22], and servers [5, 6, 27, 28]. Many of these previous research results accurately capture the characteristics of static web content. However, for dynamic and personalized web content which introduces the notion of channels, many of these characteristics need to be revisited. Moreover, personalized content necessitates understanding of new characteristics, such as the change frequency of user preferences, the number and sizes of channels making up a document, the freshness times of these channels. To the best of our knowledge, the work described in this paper is one of the first efforts trying to model these latter characteristics for a personalized web site. Our work also distinguishes itself from previous work in that we are looking for characteristics at the sub-document granularity and work with an instrumented server as opposed to existing server logs.

The first user experience analysis of a personalized web site was done by Manber et al. on the Yahoo! site [27]. In that paper only general information and some high level implications for the design of personalized web sites were presented. In contrast, our study examines detailed quantitative characteristics of personalized web sites and additionally proposes several optimizations to improve performance of dynamic content caching.

Recent studies of the MSNBC news site [30] and a large shopping site [5] are closely related to our effort. In [30], Padmanabhan and Qiu analyzed the dynamics of both the server content and client accesses made to the MSNBC news server by analyzing the standard HTTP logs from the web site. More recent work by Arlitt et al. [5] focuses on characterizing the scalability of a large web-based shopping system. Although our study shares a similar motivation with these two previous works, it complements these efforts by focusing on the characteristics of personalized web sites.

Finally, our work is also related to several previous studies of web server performance [4, 19, 21, 23, 31, 37], but differs from them in that the overhead of dynamic web content generation and related network transfer time are studied.

7 Summary and Future Work

In this paper, we have presented the analysis of a medium-sized personalized web site, NYUHome, using instrumented server logs. In addition to a detailed study of characteristics, we also present several implications derived from these observations. The main implications include: (1) Personalization functionality is increasingly being accepted, and traditional caching and prefetching schemes need to be revisited; (2) Substantial benefits are likely from applying object composition techniques for personalized content; (3) Both server load and client-perceived latencies can be further reduced by prefetching the content of a small number of personalized (non-shared) channels; (4) Client-perceived request latencies can be made more uniform by specializing the document layout and content, using transcoding, to the network connection employed by the client.

Our future work includes integrating these implications into a CONCA prototype, and cooperating with the NYUHome team to accelerate the delivery of their content.

Acknowledgement

The authors would like to acknowledge the support from ITS of New York University for this study. We wish to thank Balachander Krishnamurthy from AT&T research Lab for helping us perform IP addresses clustering. We would also like to thank the anonymous referees for their many constructive comments. This research was sponsored by DARPA agreements F30602-99-1-0157, N66001-00-1-8920, and N66001-01-1-8929; by NSF grants CAREER:CCR-9876128 and CCR-9988176; and Microsoft. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of DARPA, Rome Labs, SPAWAR SYSCEN, or the U.S. Government.

References

[1] F. 180-1. *Secure Hash Standard*. U.S. Department of Commerce/N.I.S.T., National Technical INformation Service, Springfield, VA,, apr 1995.

[2] A. Adya, P. Bahl, and L. Qiu. Analyzing browse patterns of mobile clients. *SIGCOMM Internet Measurement Workshop*, Nov. 2001.

[3] Akamai Technologies Inc. Edgesuite services, `http://www.akamai.com/html/en/sv/edgesuite_over.html`.

[4] V. A. Almeida and M. A. Mendes. Analyzing the impact of dynamic pages on the performance of web servers. *Proceedings of the Computer Measurement Group Conference*, Dec. 1998.

[5] M. Arlitt, D. Krishnamurthy, and J. Rolia. Characterizing the scalability of a large web-based shopping system. *ACM Transactions on Internet Technology* 1(1):44–69, 2001.

[6] M. Arlitt and C. Williamson. Internet web servers: Workload characterization and performance implications. *IEEE/ACM Transactions on Networking*, 1997.

[7] H. Balakrishnan, S. Seshan, M. Stemm, and R. H. Katz. Analyzing stability in wide-area network performance. *Proc. of ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, June 1997.

[8] P. Barford, A. Bestavros, A. Bradley, and M. E. Crovella. Changes in web client access patterns: Characteristics and caching implications. *World Wide Web, Special Issue on Characterization and Performance Evaluation* 2:15–28, 1999.

[9] P. Barford and M. E. Crovella. Generating representative web workloads for network and server performance evaluation. *Proceedings of Performance '98/ACM SIGMETRICS '98*, July 1998.

[10] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and zipf-like distributions: Evidence and implications. *Proc. of the IEEE Conference on Computer Communications (INFOCOM'99)*, Mar. 1999.

[11] J. Challenger, A. Iyengar, and P. Dantzig. A scalable system for consistently caching dynamic web data. *Proceedings of Infocom'99*, Mar. 1999.

[12] J. Challenger, A. Iyengar, K. Witting, C. Ferstat, and P. Reed. A publishing system for efficiently creating dynamic web content. *Proc. of the IEEE Conference on Computer Communications (INFOCOM'00)*, Mar. 2000.

[13] M. E. Crovella and A. Bestavros. Self-similarity in world wide web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking* 5(6):835–846, 1997.

[14] C. Cunha, A. Bestavros, and M. E. Crovella. Characteristics of WWW client-based traces. Tech. Rep. BU-CS-95-010, Computer Science Department, Boston University, July 1995.

[15] R. B. D'Agostino and M. A. Stephens, editors. *Goodness-of-Fit Techniques*. Marcel Dekker, Inc, 1986.

[16] F. Douglis, A. Feldmann, B. Krishnamurthy, and J. Mogul. Rate of change and other metrics: a live study of the world wide web. *Proc. of the 1st USENIX Symposium on Internet Technologies and Systems (USITS'97)*, Dec. 1997.

[17] F. Douglis, A. Haro, and M. Rabinovich. HPP:HTML macro-pre-processing to support dynamic document caching. *Proc. of the 1st USENIX Symposium on Internet Technologies and Systems (USITS'97)*, Dec. 1997.

[18] R. Doyle, J. Chase, S. Gadde, and A. Vahdat. The trickle-down effect: Web caching and server request distribution. *Proc. of the 6th International Workshop on Web Caching and Content Distribution (WCW'01)*, June 2001.

[19] Y. Hu, A. Nanda, and Q. Yang. Measurement, analysis and performance improvement of apache web server. *Proceedings of the IEEE International Performance, Computing, and Communications Conference*, Feb. 1999.

[20] IBM Corp. Websphere platform, `http://www.ibm.com/websphere`.

[21] A. Iyengar, J. Challenger, D. Dias, and P. Dantzig. High-performance web site design techniques. *IEEE Internet Computing* 4(2), March/April 2000.

[22] T. Kelly. Thin-client web access patterns: Measurements for a cache busting proxy. *Proc. of the 6th International Workshop on Web Caching and Content Distribution (WCW'01)*, June 2001.

[23] B. Kothari and M. Claypool. Performance analysis of dynamic web page generation technologies. *Proceedings of the International Network Conference (INC)*, July 2000.

[24] B. Krishnamurthy and J. Rexford. *Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching and Traffic Measurement*. Addison-Wesley, Inc, 2001.

[25] B. Krishnamurthy and J. Wang. On network-aware clustering of web clients. *Proc. of the ACM SIGCOMM'00*, 2000.

[26] B. Krishnamurthy, C. Wills, and Y. Zhang. On the use and performance of content distribution networks. *Proceedings of SIGCOMM IMW 2001*, Nov. 2001.

[27] U. Manber, A. Patel, and J. Robison. Experience with personalization on Yahoo! *Communications of ACM* 43(8):35–39, Aug. 2000.

[28] S. Manley and M. Seltzer. Web facts and fantasy. *Proc. of the 1st USENIX Symposium on Internet Technologies and Systems (USITS'97)*, Dec. 1997.

[29] J. C. Mogul, F. Douglis, a. Feldmann, and B. Krishnamurthy. Potential Benefits of Delta-Encoding and Data Compression for HTTP. *Proc. of the 13th ACM SIGCOMM'97*, Sept. 1997.

[30] V. N. Padmanabhan and L. Qiu. The content and access dynamics of a busy web site: Findings and implications. *ACM SIGCOMM'2000*, 2000.

[31] V. Pai, P. Druschel, and W. Zwaenepoel. Flash: An efficient and portable web server. *Proceedings of the 1999 USENIX Annual Technical Conference*, June 1999.

[32] V. Paxson. Empirically-derived analytic models of wide area TCP connections. *IEEE/ACM Transactions on Networking*, 1994.

[33] V. Paxson and S. Floyd. Wide-area traffic: The failure of poisson modeling. *Proc. of ACM SIGCOMM'94*, pp. 257–268, Aug. 1994.

[34] W. Shi, E. Collins, and V. Karamcheti. Modeling object characteristics of dynamic web content. *IEEE Globecomm 2002 Conference*, Nov. 2002.

[35] W. Shi and V. Karamcheti. CONCA: An architecture for consistent nomadic content access. *Workshop on Cache, Coherence, and Consistency(WC3'01)*, June 2001.

[36] M. Tsimelzon, B. Weihl, and L. Jacobs. ESI language sepcification 1.0, 2000, `http://www.esi.org`.

[37] M. Welsh, D. Culler, and E. Brewer. SEDA: An architecture for well-conditioned, scalable internet services. *Proc. of the 18th ACM Symp. on Operating Systems Principles (SOSP-18)*, Oct. 2001.

[38] D. Wessels. *Web Caching*. O'Reilly Inc., 2001.

[39] C. E. Wills and M. Mikhailov. Studying the impact of more complete server information on web caching. *Proc. of the 5th International Workshop on Web Caching and Content Distribution (WCW'00)*, 2000.

[40] A. Wolman, G. M. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. M. Levy. On the scale and performance of cooperative web proxy caching. *Proc. of 17th ACM Symposium on Operating Systems Principles (SOSP)*, 1999.

[41] W3C XSL Working Group, `http://www.w3.org/Style/XSL/`.