RC22567 (W0209-083) September 23, 2002
Computer Science

IBM Research Report

Traffic Profiling, Clustering and Classification for Commercial Web Sites

Zhen Liu, Mark S. Squillante, Cathy Honghui Xia, Shun-Zheng Yu, Li Zhang
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

Traffic Profiling, Clustering and Classification for Commercial Web Sites

Zhen Liu, Mark S. Squillante, Cathy H. Xia,
Shun-Zheng Yu, Li Zhang

IBM T. J. Watson Research Center
Yorktown Heights, NY 10598, USA
{zhenl,mss,cathyx,shuzheng,zhangli}@us.ibm.com

Abstract

The problems of workload characterization, performance modeling, workload and performance forecasting, and capacity planning are fundamental to the growth of Web services and applications. Previous studies have primarily focused on the complexity of Web traffic at the level of object-hits or page-views. In contrast, our study focuses on higher-level characteristics, and introduces techniques for profiling, clustering and classification of Web site traffic. In particular, we devise novel techniques for efficient and automated extraction of Web traffic patterns from access logs, for efficient and automated clustering of such traffic patterns, and for efficient and automated classification of Web traffic based on the extraction and clustering of traffic templates. Our approach has been applied to more than 25 existing commercial Web sites. Moreover, it has been demonstrated that our approaches can accurately capture and characterize the complexities of Web traffic in commercial Web sites. These methods provide new solutions to solve the challenging problems such as workload and performance prediction, and short-term and long-term capacity planning.

1 Introduction

In the most general sense, there are many different aspects of Web traffic, including diverse user request patterns, different types of Web content (e.g., static, dynamic, continuous media, etc), different types of resource requirements and usage (e.g., network bandwidth, CPU, memory/cache storage, I/O bandwidth, disk storage, etc), and different types of user requests (e.g., browsing, database transactions, secure transactions, etc). Each of these aspects of Web traffic are complex in their own right, as demonstrated by previous work in the research literature that has primarily focused on lower-level (or micro-level) behaviors [1, 4, 2, 5, 7, 10]. An additional cause of complexity is due to the time-varying interactions among these different Web traffic components, which exhibit complex patterns at macro-levels resulting from hourly effects, daily effects, weekly effects, monthly effects, seasonal effects, various trends, and other higher-level effects.

Another set of problems that are critical to the successful deployment of Web servers concerns the high-level issues of workload characterization, performance modeling, workload and performance forecasting, and capacity planning. The complexity of Web traffic at both micro-levels and macro-levels further compound the difficulty of these important issues. Moreover, as the Internet continues to grow, a large number of Web sites will experience an increase in their overall access rates, possibly due to more material being posted on the site, or greater functionality provided by the Web site, or more people accessing the Web site, and so on. It is therefore crucial to provide techniques that make it possible to manage this growth by accurately predicting the traffic, by judiciously planning the capacity, and by optimally deploying the available resources. To obtain effective solutions to these many problems, a very important first step is to identify

some high-level structures that can capture and characterize the fundamental elements amidst the various levels and sources of Web traffic complexity.

In this paper, we seek to identify high-level structures and invariants in Web traffic that can be exploited to help solve the problems of workload characterization, performance modeling, workload and performance forecasting, and capacity planning. This includes identifying four common macro-level patterns that appear to exist in the time-varying workloads of a wide variety of commercial Web sites. We then exploit these macro-level behaviors to develop new techniques for efficient and automated profiling, clustering and classification of Web traffic. In particular, we devise methods for efficient and automated extraction of Web traffic patterns from empirical data, for efficient and automated clustering of such traffic patterns, and for efficient and automated classification of Web traffic based on the extraction and clustering of traffic templates. Our techniques have been applied to data from 26 different commercial Web sites from a broad range of industries. We also consider the ways in which our methods can be used to help solve the fundamental problems of workload and performance prediction, capacity planning, hot-spot detection, dynamic offloading and Web site co-location. This work also can be exploited to examine the individual components of Web traffic either in isolation or in various combinations, where such components include user request patterns, different types of Web content, different types of resource requirements and usage, and different types of user requests.

The remainder of this paper is organized as follows. We shall first present in Section 2 some preliminary analyses of the traffic data from 26 commercial Web sites. We then propose in Section 3 a technique to cluster the Web sites according to their traffic patterns. In Section 4 we propose a method to determine the profiles of the Web sites based on the clustering results. These profiles are then used for the classification of new Web sites, as discussed in Section 5. In Section 6 we shall present various applications of our results. Concluding remarks are finally presented in Section 7.

2 Web Data Description and Preliminary Analysis

Our study includes an extensive analysis of data from 26 different commercial Web sites, which span a wide range of industrial sectors. For each commercial Web site, we consider a diverse set of measures extracted from the corresponding server access logs. These measures include number of hits, bytes, page views, visits, hits per visit, page views per visit, seconds per page view, and seconds per visit. The data is indexed by time, and thus each measure is computed at different time scales over a long interval. We consider a wide range of different time scales including monthly, weekly, daily and hourly time scales. The data used in our study spans a relatively long time interval covering at least 4 months, and in some cases as much as 17 months, of recent activity at the various commercial Web sites. In this section, we present a representative sample of the results of our analysis, which is then exploited in the next section to identify some clusters of Web sites with common features based on some of the characteristics and trends revealed by our analysis.

2.1 Empirical Patterns

A large number of different patterns distinguish the request processes for various commercial Web sites. Some of these patterns have more prominent visual effects, such as fluctuations of request volume over time or shifts in the request volume from one point to another, and so on. We refer to these types of characteristics as empirical patterns because they can be identified and examined by a nonmathematical analysis of the

time-series data [3, 9] (even though many of these characteristics can also be further quantified by statistical measures). Moreover, some of these empirical request patterns can be closely tied to societal events and/or human behaviors at the levels of hourly effects, daily effects, weekly effects, monthly effects, seasonal effects, various trends, and other higher-level effects.

We initially consider some of the daily patterns at the various commercial Web sites over a representative week. To do so, the measure of number of hits per hour is used to characterize the incoming request patterns over each day of the week. The corresponding time-series plots are provided in Figure 1 for 18 of the 26 different Web sites, where we note that this set of plots covers the range of daily patterns observed as part of our analysis of the complete set of Web sites. Furthermore, our focus on a single week is due to the fact that the hourly plots for each week at most of the Web sites are sufficiently consistent that the last week of data is often illustrative of the typical behavior for longer durations.

We first observe that a large number of Web sites exhibit daily access patterns for which there is a considerable drop in the request rate (both hits and pages) over Saturdays and Sundays relative to the other days of the week. (Such weekend effects also can be clearly seen in the corresponding autocorrelation function of each time series, as will be shown below.) Some of the Web sites across different industries often have clear weekend patterns with significant drops in the request rate over the weekend (see the first plot from the bottom on the left in Figure 1). A weaker yet still prevalent weekend pattern exists for some other Web sites, again across different industries, with less significant drops in the request rate over the weekend (see the fourth plot from the top on the left in Figure 1). Finally, some Web sites from certain industries do not exhibit any weekend patterns (see the second plot from the bottom on the left in Figure 1). We further observe that when the Web site has television coverage associated with it, then one typically finds some form of weekend patterns in the user requests for the Web site, which also appears to hold true for sporting event Web sites for similar reasons.

We now turn to consider some of the coarser time scales (i.e., weekly and monthly) and some of the other measures of request patterns in addition to hits (i.e., number of bytes, page views, visits, hits per visit, page views per visit, seconds per page view, and seconds per visit). In Figures 2 through 4 we provide the corresponding set of time-series plots for 3 different commercial Web sites. Once again we find weekend effects of varying degrees for many of the Web sites, which also are clearly apparent for the other measures of requests. Certain commercial Web sites with weekend patterns also exhibit very consistent week-to-week behavior in which request measures do not change much from one week to the next. Due to the nature of their business, some commercial Web sites incur a surge of requests during special seasons such as the Christmas rush or the season for a particular sport; e.g., see Figures 3 and 4.

A large number of Web sites exhibit periodic access patterns for which the request rate (both hits and pages) are roughly the same at around the same time of the day, and the same day of the week. This suggests a steady collection of people routinely accessing the Web site from day to day. Moreover, certain commercial Web sites have different access patterns during different intervals of time. We sometimes observe very regular request patterns followed by a sudden change to a different regime, followed by a return back to the regular request pattern. Similarly, certain commercial Web sites have different access patterns during different intervals of time in which we can observe a very regular request pattern followed by a sudden change to a different, yet regular request pattern; e.g., see Figure 3. This may be the result of a redesign of the Web site, or certain changes in the commercial business.

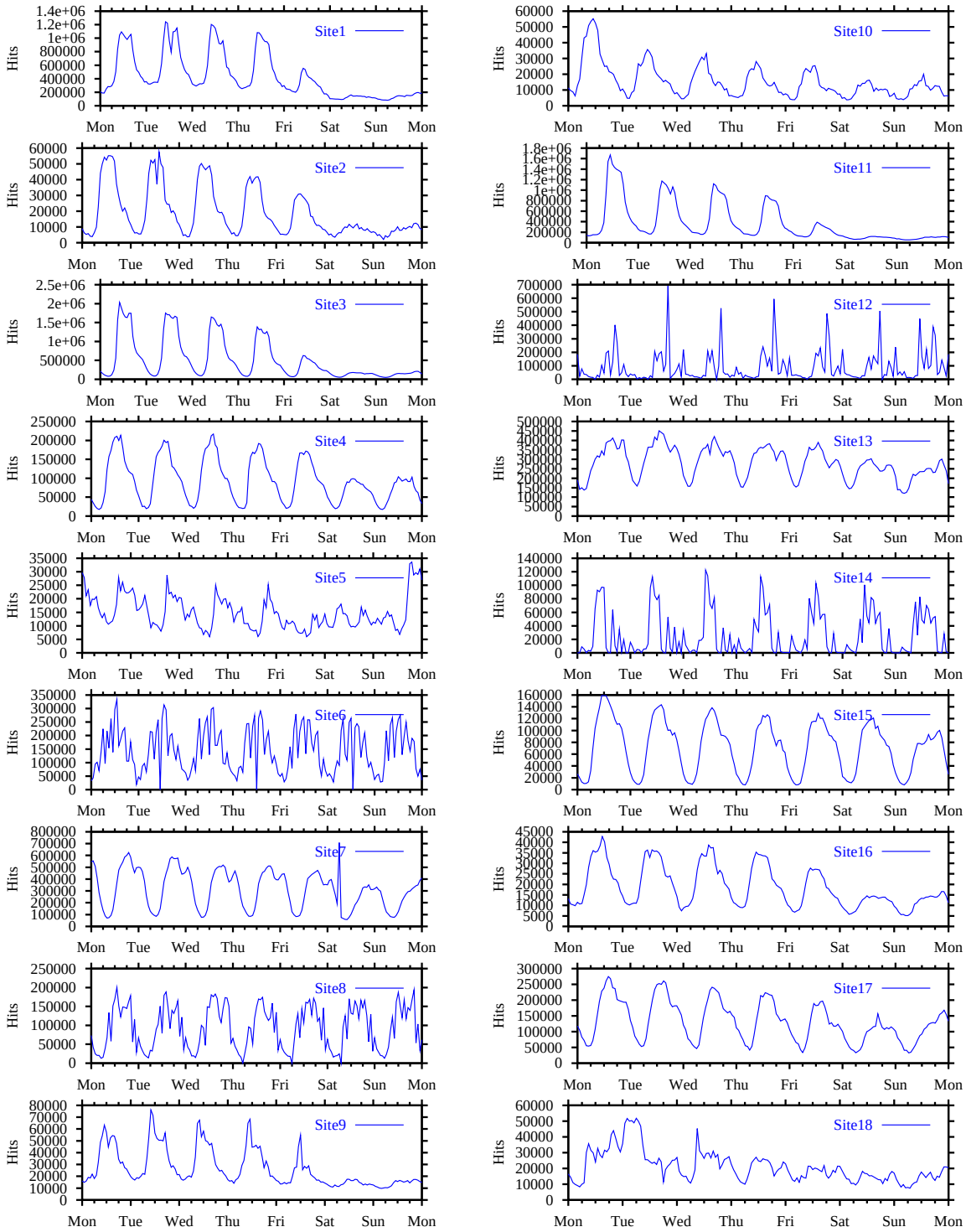

Figure 1: Hourly hits over a period of one week for 18 commercial Web sites.

2.2 Statistical Measures

In addition to the above empirical characteristics, some of the most complex characteristics of the user request patterns can be identified and examined through various statistical measures computed from the

time-series data for each of the various commercial Web sites. While a relatively large number of statistical measures have been considered as part of our study, we focus here are a few measures that provide some of the most interesting and representative sets of results. To simplify the presentation, we will use X_n to generically denote a given measure of interest (e.g., number of hits) at (discrete) time n for a given time scale (e.g., hours), $n = 1, 2, \ldots$.

Some useful aspects of the variability of the request time-series can be examined by the coefficient of variation (CV) of the overall request volume, i.e., the ratio of the standard deviation to the mean. More formally, we have

$$\mathrm{CV}[X_n] \;=\; \frac{\mathrm{Std}[X_n]}{\mathrm{E}[X_n]}$$

where $\mathrm{Std}[X_n] \;=\; \sqrt{\mathrm{Var}[X_n]}$. In Figure 5 we plot the coefficient of variation of various daily measures (the same as those considered in Figures 2 through 4) as a function of the mean request rate for 20 different commercial Web sites.

One rough measure of burstiness in the request patterns is the ratio of the peak to mean request rate over a certain interval of time. This measure quantifies the peak request volume relative to the average request volume, i.e.,

$$\mathrm{PM}[X_n] \;=\; \frac{\max\limits_{n} X_n}{\mathrm{E}[X_n]}.$$

In Figure 6 we plot this peak to mean ratio of various daily measures (the same as those considered in Figure 5) as a function of the mean request rate for 20 different Web sites.

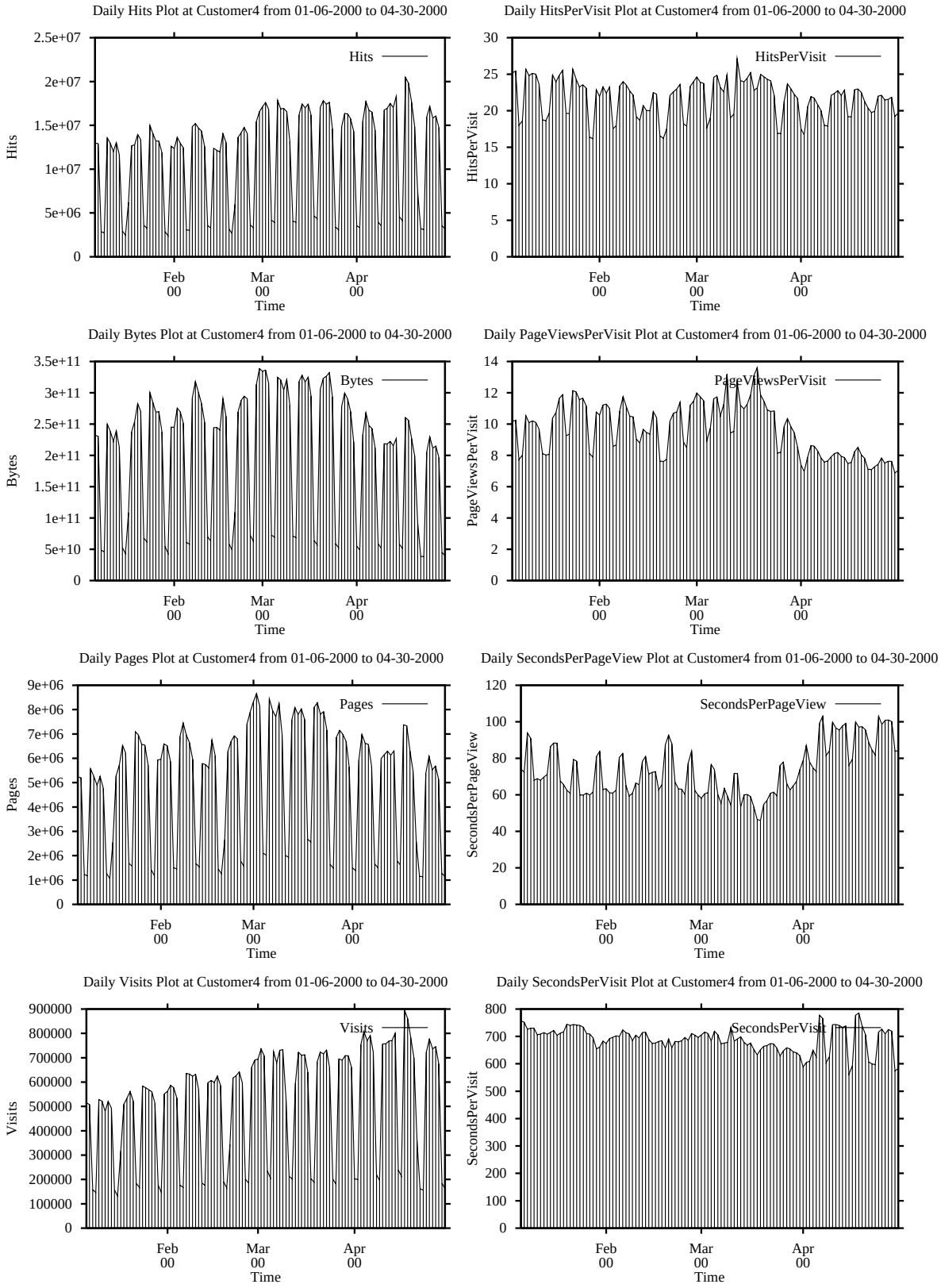

Figure 2: Daily time-series plots for one of the commercial Web sites.

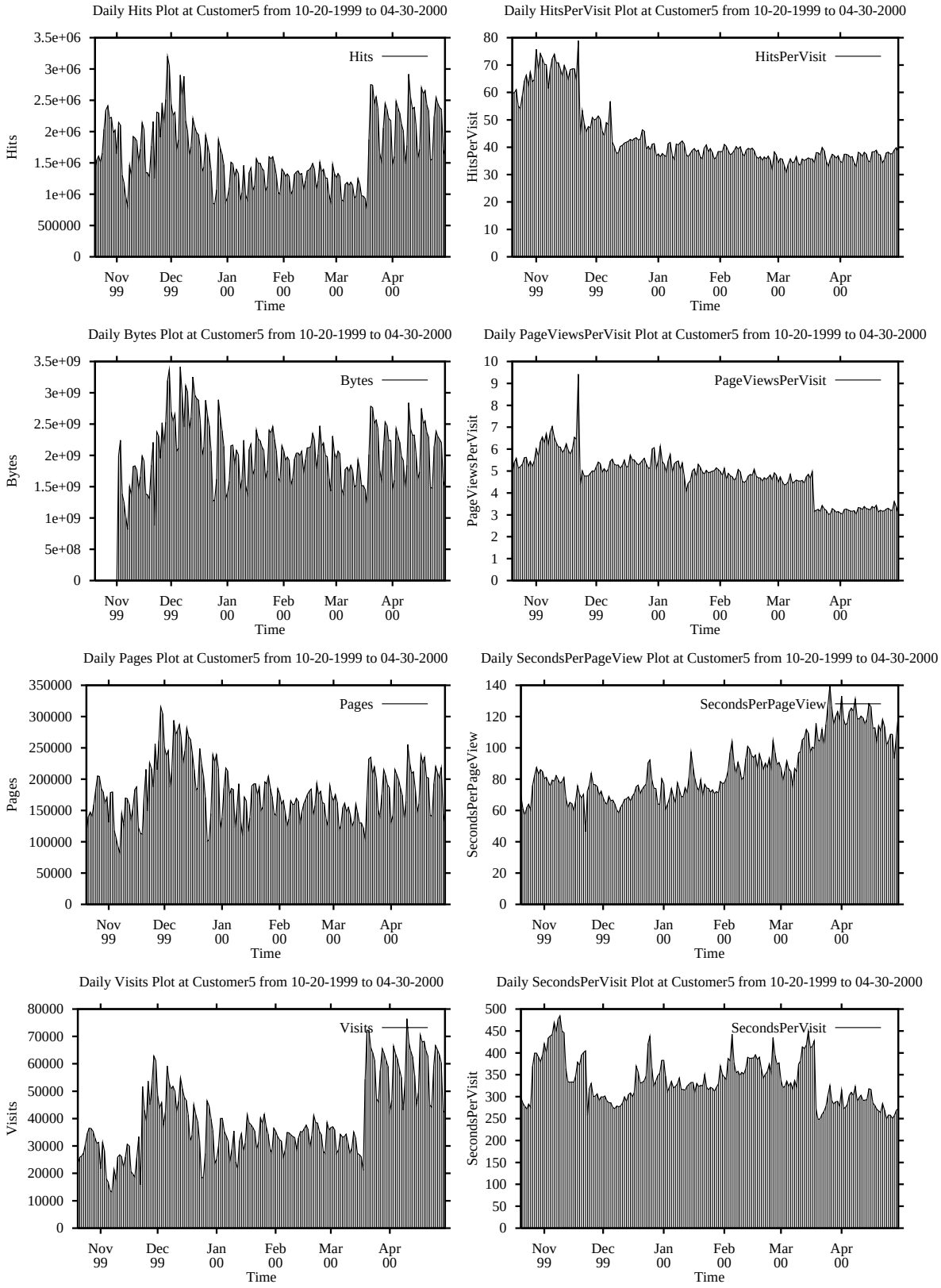

Figure 3: Daily time-series plots for one of the commercial Web sites.

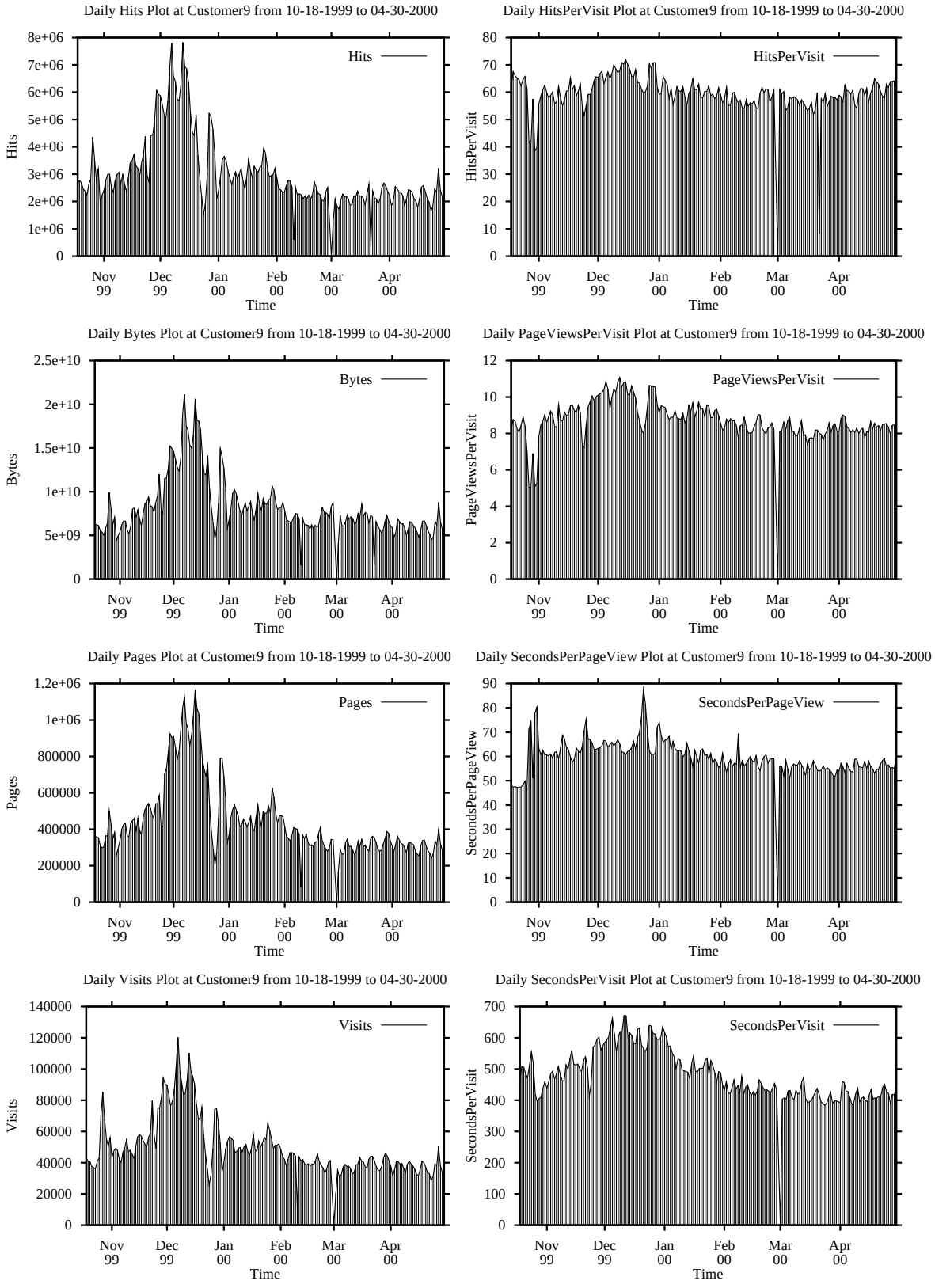

Figure 4: Daily time-series plots for one of the commercial Web sites.

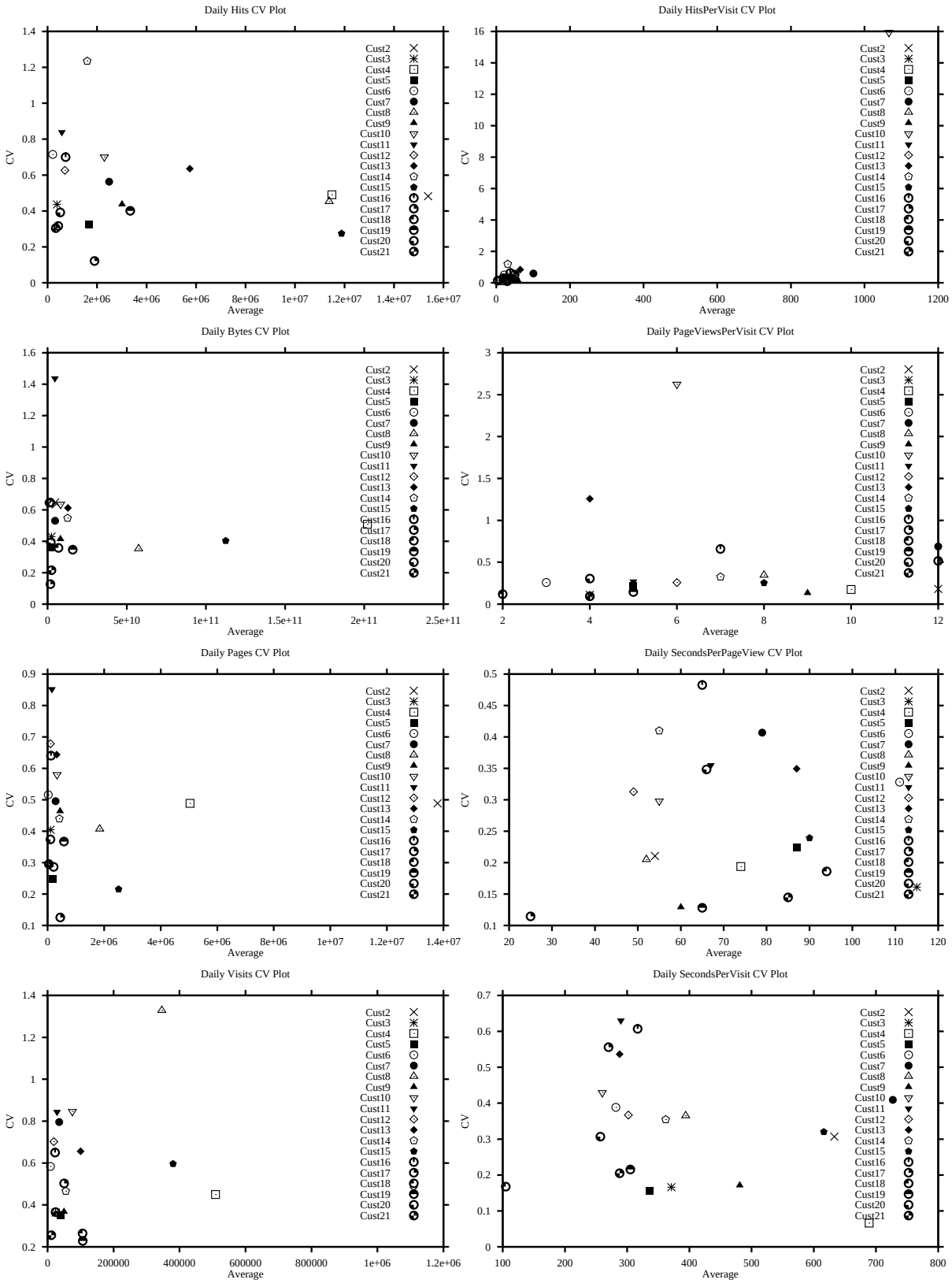

Figure 5: Daily coefficient of variation as a function of the daily average for different measures encountered at 20 commercial Web sites.

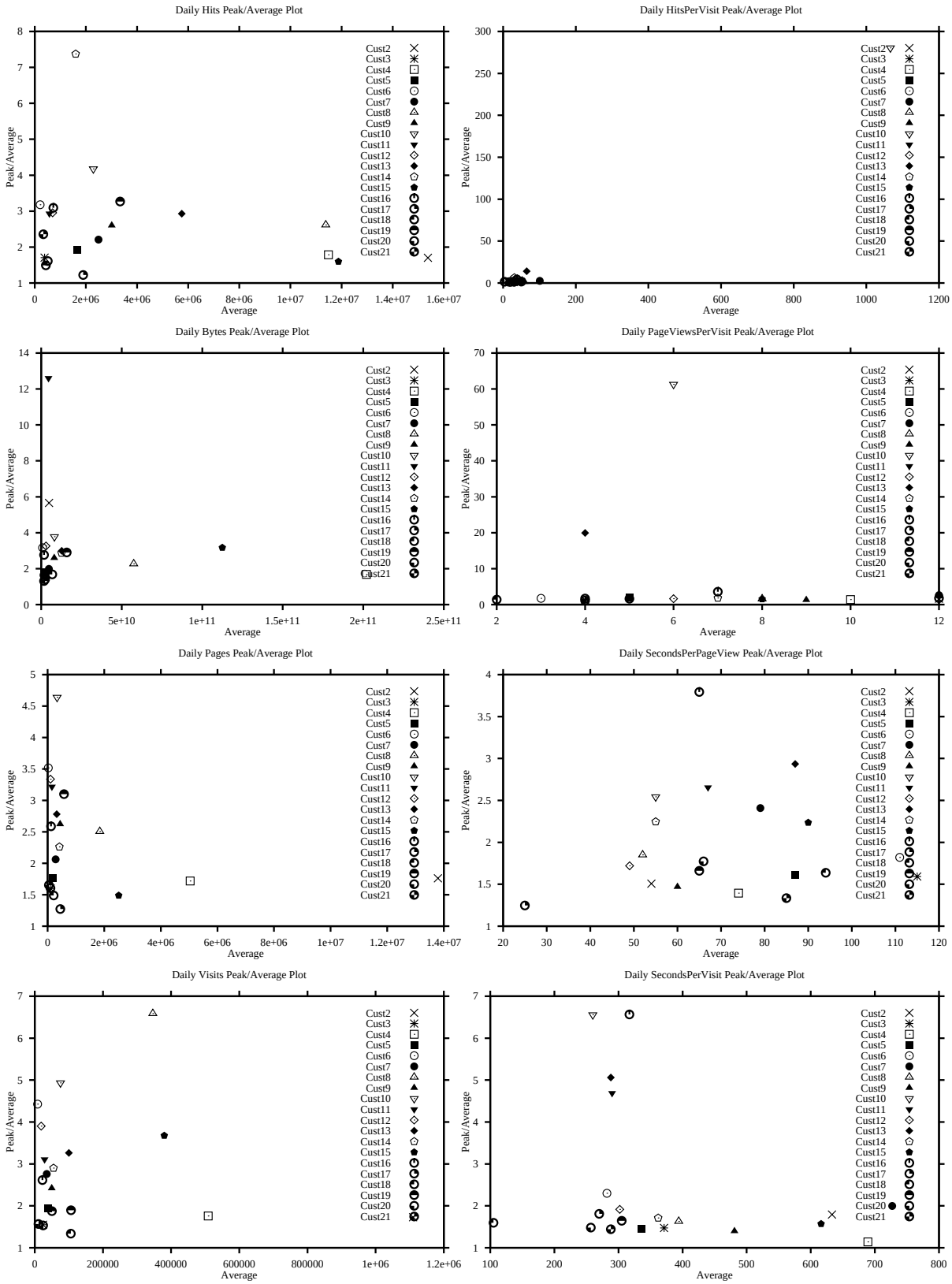

Figure 6: Daily peak to mean ratio as a function of the daily average for different measures encountered at 20 commercial Web sites.

2.3 Summary and Motivation

While the results of the above empirical and statistical analysis (including those not presented due to space limitations) illustrate and quantify various trends and behaviors found in commercial Web sites, most of these results are not easily exploitable as the basis for our approach to clustering, profiling and classification of Web traffic. In particular, the coefficient of variation and ratio of peak to mean statistical measures (among others) appear to provide different patterns for different request measures (e.g., number of bytes vs. seconds per page view) which, although quite interesting in their own right, makes their use for our clustering purposes somewhat limited. We therefore need a new approach to support the desired clustering, profiling and classification of Web traffic. This is considered in the next section, the results of which can be exploited in turn to address various application problems that are subsequently considered.

3 Clustering

We now consider the clustering of the web sites. Depending on the subset of measurement data used, different clustering results can be obtained. The web sites can be clustered according to the load/request patterns, user navigation patterns, or site hypertext structures. These clusterings are different in general. We can consider them as orthogonal so that they are in different dimensions.

In this paper, we present the clustering according to the load patterns. For this purpose, we use the number of hits per hour as representative data to characterize the incoming request patterns. Examples for the complete set of 26 different commercial Web sites are provided in Figure 7. For convenience, we will use hits and requests interchangibly.

Note that for 8 out of the 26 Web sites, only one-day traffic data is available to us. For the remaining 18 Web sites, we have data from at least 4 months: January to April 2000. Thus, we shall use these 18 Web sites for the study of clustering and profiling, and then use these 8 Web sites for the classification analysis.

It follows from the analysis of the previous section, that the daily traffic patterns for weekdays are quite regular for most Web sites, and they are quite different from those for weekends; see Figure 8. For weekdays, the averaging of workload patterns yields smoother and smoother curves. In other words, these traffic patterns exhibit some forms of ergodicity.

In our analysis, we shall focus on weekdays. We are interested in the *typical shapes*, or templates of these workloads. For this purpose, we consider the weighted average request pattern $\mu_i(h)$ representing the weighted mean of the hourly request pattern profile that occurs on Web server i, i.e.,

$$\mu_i(h) = \sum_{d=1}^{D} \frac{a_i(d)}{D} x_i(h,d), \tag{1}$$

where $x_i(h,d)$ denotes the number of requests from the empirical data that the i^{th} Web server receives in the h^{th} hour of day d, $h = 0, 1, \ldots, 23$, and where the weights $a_i(d)$ is the weight for day d of site i so that the workloads of different days are normalized to the same mean.

Moreover, we shall concentrate on the peak load regimes as they have a more significant impact on the Web server performance. We therefore normalize each pattern by its maximum value and define its peak

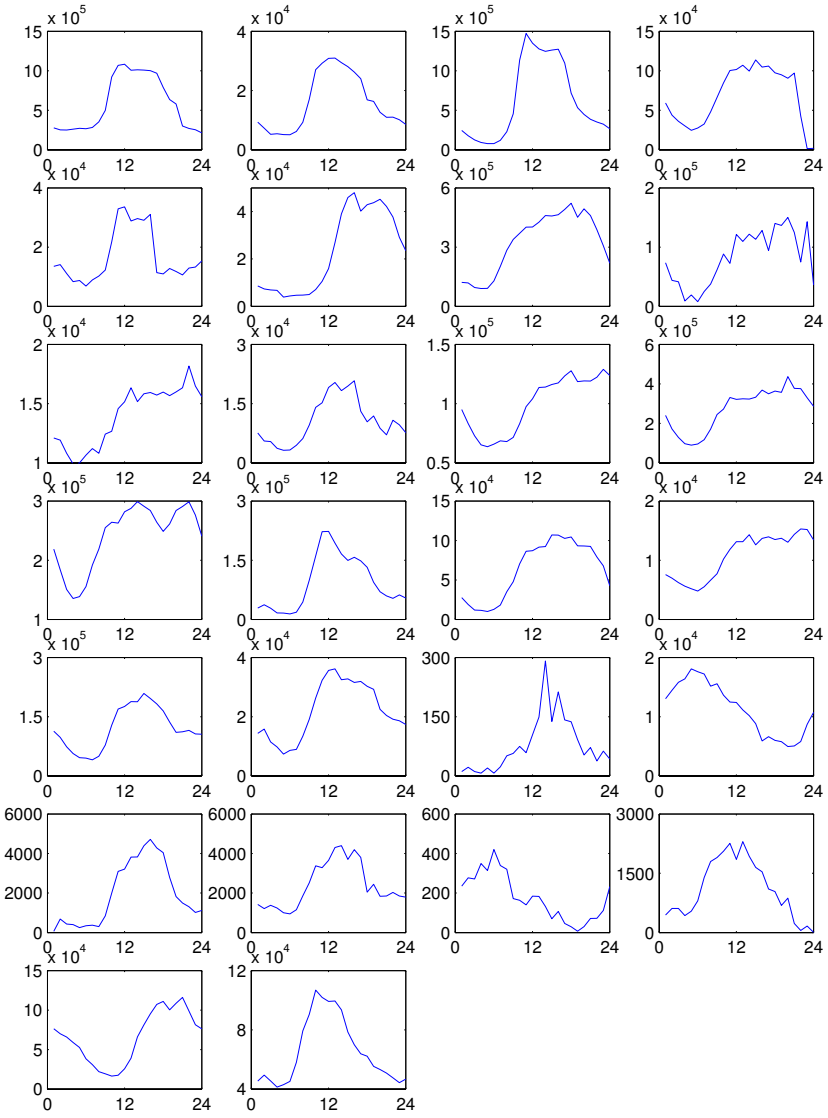

Figure 7: Requests per hour over one day collected from 26 different commercial Web sites

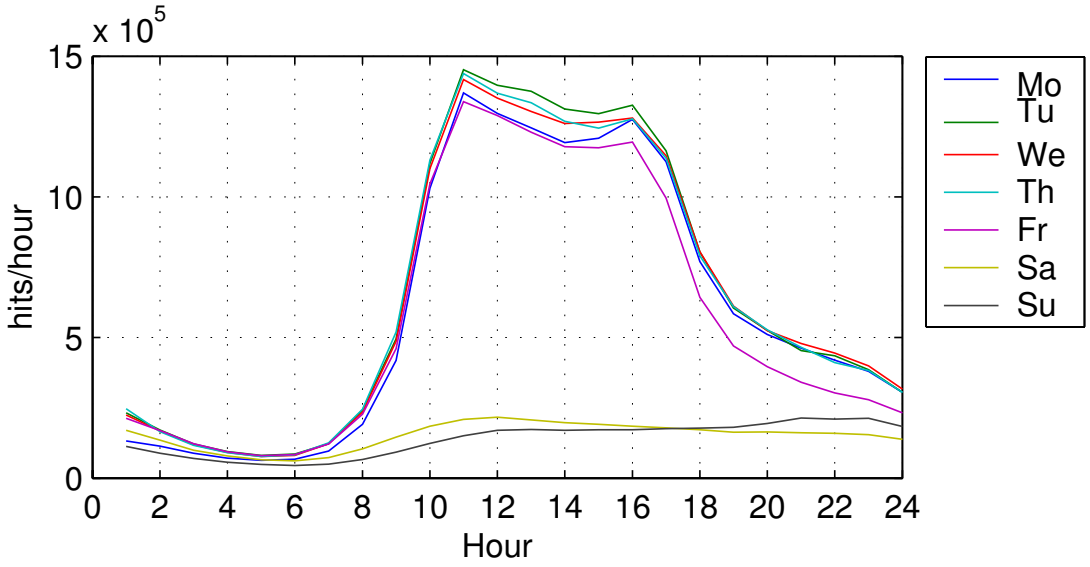

Figure 8: The average number of requests per hour over weekdays and weekends

hour traffic pattern as follows

$$\eta_i(h) = \begin{cases} \mu_i(h)/m_i, & \mu_i(h)/m_i > 0.5 \\ 0.5 & \mu_i(h)/m_i \leq 0.5 \end{cases}, \tag{2}$$

where $m_i = \max_h\{\mu_i(h)\}$. We then define a dissimilarity measure between the peak hour patterns as

$$\delta_{i,j} = \min_{h^s} \max_{h} \{ \mid \eta_i(h) - \eta_j(h + h^s \textbf{mod} 24) \mid \} \tag{3}$$

where h^s is used as the hourly shift needed when comparing two traffic patterns from different Web servers located in different time zones.

We next use the complete linkage (or furthest neighbor) algorithm [6, 8] to cluster the request patterns based on our dissimilarity measure. In the first step, each pattern represents its own cluster, and the distances between these patterns are defined by the dissimilarity measure given by Equation (3). Then the two patterns with the smallest distances are linked together. The distances between this new cluster and the other clusters (or individual patterns) are defined by the greatest distance between any two patterns in the respective clusters (i.e., by the "furthest neighbors"). As a result, the algorithm proceeds in subsequent steps to link more and more patterns together and to aggregate larger and larger clusters. Using a cutoff of 1 in the algorithm (where the cutoff must be between 0 and 2, exclusively), the request patterns shown in Figure 7 are clustered into 4 distinct classes, which are illustrated in Figure 11. We can also use the K-means clustering algorithm [6, 8] to obtain the same clustering results if we specify that the algorithm forms exactly 4 clusters which are to be as distinct as possible.

From Figure 11 we observe that:

i. For the Class 1 patterns, the request traffic load increases to the peak level by noon and then goes down significantly in the afternoon. This suggests that users' interests for such Web sites are more

instantaneous, just like checking the weather report every day. The Web sites are popular primarily in their local areas.

ii. For the Class 2 patterns, the request traffic load increases to the peak level at noon and continues to remain high in the afternoon, but it becomes very low in the evening. This suggests that users show their interests for these Web sites primarily during working hours.

iii. The request traffic load for Class 3 increases to the peak level somewhat after noon and remains high throughout much of the afternoon. In the evening users continue to show some interest for these Web sites.

iv. For Class 4 patterns, the request traffic load remains at a high level over a long period, from before noon well into the evening. This suggest that most of the users visit these Web sites either during working hours or in their spare time, and that the users are probably spread over the country and even the world.

4 Profiling

In this section, we turn to consider the profiling of Web sites. This involves first determining a metric upon which the traffic profiles will be based. One possibility is to use the average, or the weighted average, load $\mu_i(h)$ as the profile for the Web site.

We discuss here the profiles in terms of Web site clusters. After employing the above clustering methods, we proceed to find the templates of the four classes of request patterns. We define the template for a class to be the request pattern that is most similar to all of the members of the class. A naive approach is based on simply averaging all the members in the class. Obviously, this is not a good choice because an outer member may be far from this template but close to the templates of the other classes. Another underlying consideration is that the templates defined for different classes should be far from each other. Therefore, we define the template so that it minimizes the maximum difference to all members in the class.

Let G_k denote the set of request patterns that belong to class k. Though similar in shape, the members within the same class are actually lying in different time zones. To define the template, we therefore first need to identify the *correct* shift biases $\{h_i^s, i \in G_k\}$, so that upon the shift all members have the closest shapes. Such shift biases can be solved via the following mathematical expression:

$$\min_{\{h_i^s, h_j^s,\, i,j \in G_k\}} \left\{ \max_{\{i,j \in G_k\}} \max_h \left[\mid \eta_i(h + h_i^s \mathbf{mod} 24) - \eta_j(h + h_j^s \mathbf{mod} 24) \mid \right] \right\}, \tag{4}$$

That is, upon the shift $\{h_i^s, i \in G_k\}$, the maximum difference between any two members of the class should be minimized.

Such a problem can be solved using exhaustive search with a time complexity of $O(H^{|G_k|})$, where $H \equiv 24$ is the number of hours. We shall use the following algorithm with the time complexity of $O(H|G_k|)$.

[Shifting Algorithm]

Step 0. Initialization: Set $\{h_i^s, i \in G_k\}$ to be arbitrary integers between 0 and 23.

Step 1. For each $i \in G_k$, update h_i^s so that

$$h_i^s = \arg\min_{h_i^s} \left\{ \max_{j \in G_k} \max_h \left[\mid \eta_i(h + h_i^s \mathbf{mod} 24) - \eta_j(h + h_j^s \mathbf{mod} 24) \mid \right] \right\} \tag{5}$$

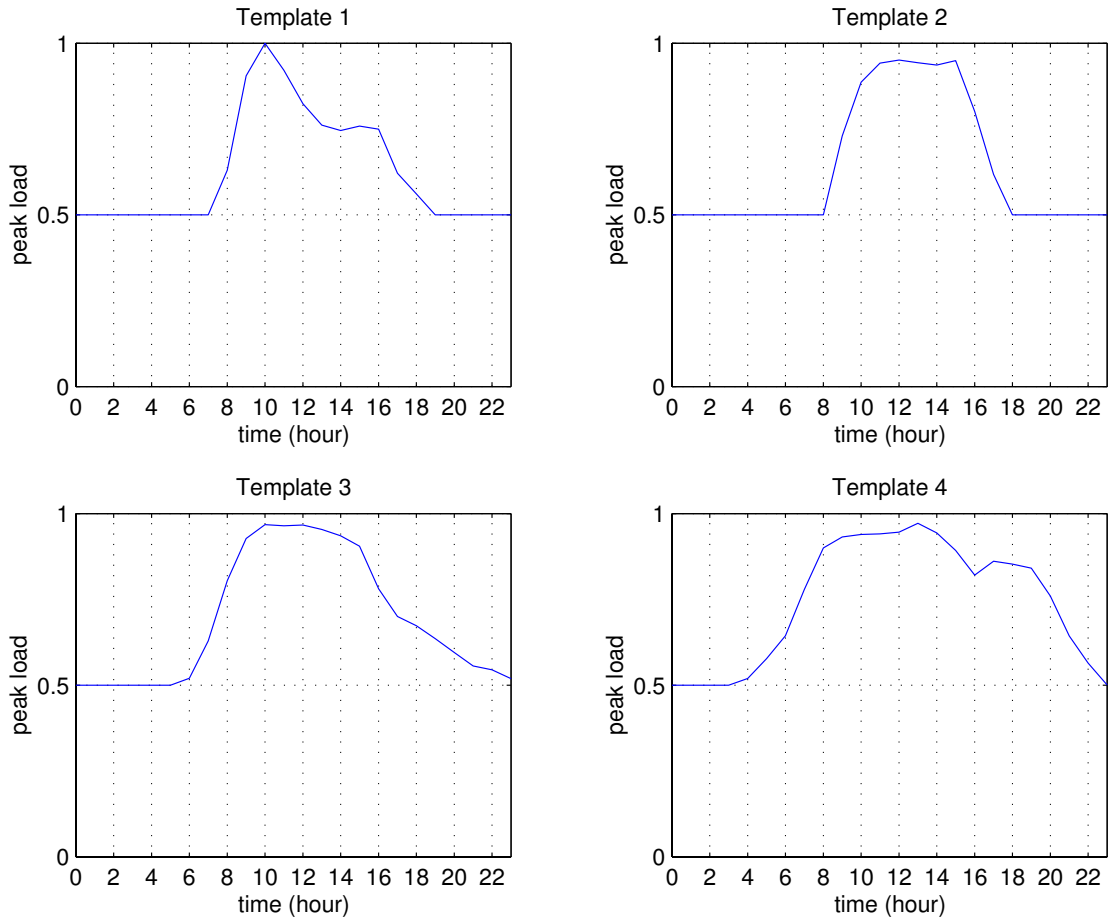

Figure 9: Templates of the four classes of request patterns

Step 2. Repeat Step 1 until $\{h_i^s, i \in G_k\}$ converge (to possibly a local optimum).

Thus, for each $i \in G_k$, h_i^s is chosen so that upon the shift, member i is close to all other members (under the current shifting) of the class. The procedure is then repeated iteratively until no further improvement can be obtained.

Once the optimal shift biases $\{h_i^s, i \in G_k\}$ are identified, the template of class k is simply defined to be

$$T_k(h) = \frac{1}{2}\left\{\max_{j \in G_k}\left\{\eta_j(h + h_j^s \mathbf{mod} 24)\right\} + \min_{j' \in G_k}\left\{\eta_{j'}(h + h_{j'}^s \mathbf{mod} 24)\right\}\right\}, \tag{6}$$

for $h = 0, 1, \ldots, 23$. That is, the template is the most similar to all of the members of the class. The corresponding results are provided in Figure 9 and Table 1. Applying an analogous algorithm to the appropriate empirical data, we can determine the templates corresponding to different effects such as day of week, week of month, month of year, etc. As a specific example, the corresponding day of week patterns are shown in Figure 10.

Depending on the load in these two time intervals, we cluster the request patterns into the foregoing four profiles. From our analysis of the empirical data of the 18 commercial Web sites, we observe that

Time (h)	Template 1	Template 2	Template 3	Template 4
0	0.5000	0.5000	0.5000	0.5000
1	0.5000	0.5000	0.5000	0.5000
2	0.5000	0.5000	0.5000	0.5000
3	0.5000	0.5000	0.5000	0.5000
4	0.5000	0.5000	0.5000	0.5196
5	0.5000	0.5000	0.5000	0.5773
6	0.5000	0.5000	0.5199	0.6443
7	0.5000	0.5000	0.6288	0.7790
8	0.6299	0.5000	0.8043	0.9000
9	0.9049	0.7296	0.9273	0.9317
10	0.1000	0.8864	0.9676	0.9388
11	0.9209	0.9419	0.9642	0.9407
12	0.8236	0.9507	0.9666	0.9458
13	0.7613	0.9433	0.9536	0.9715
14	0.7458	0.9365	0.9354	0.9438
15	0.7583	0.9495	0.9051	0.8926
16	0.7494	0.8012	0.7808	0.8207
17	0.6211	0.6178	0.7001	0.8609
18	0.5609	0.5000	0.6735	0.8528
19	0.5000	0.5000	0.6354	0.8410
20	0.5000	0.5000	0.5951	0.7596
21	0.5000	0.5000	0.5559	0.6438
22	0.5000	0.5000	0.5448	0.5647
23	0.5000	0.5000	0.5188	0.5005

Table 1: Data of the four templates

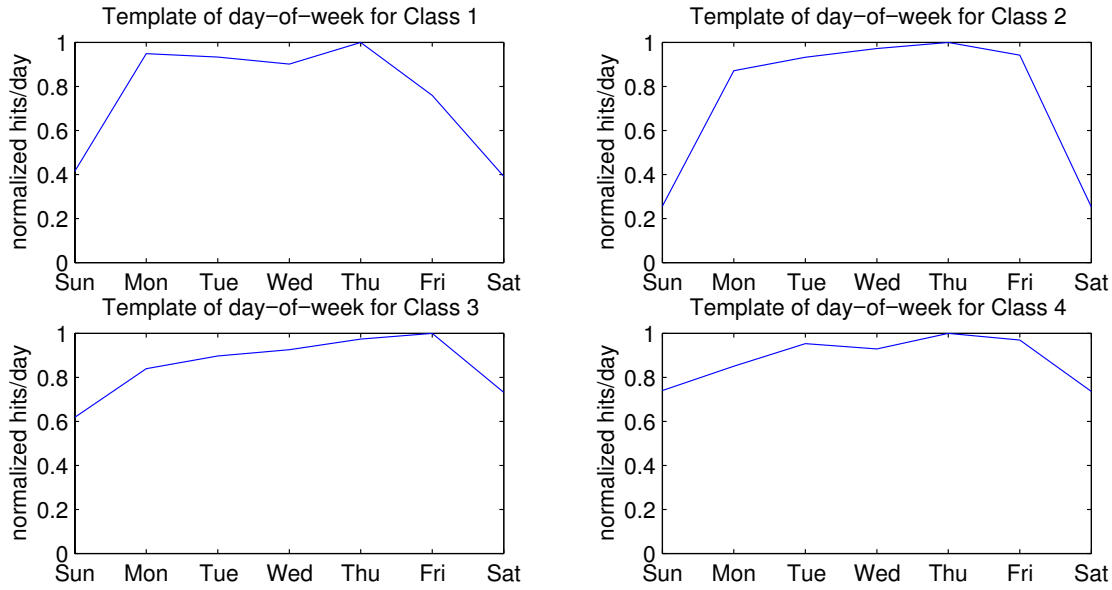

Figure 10: Templates of day-of-week for the four classes

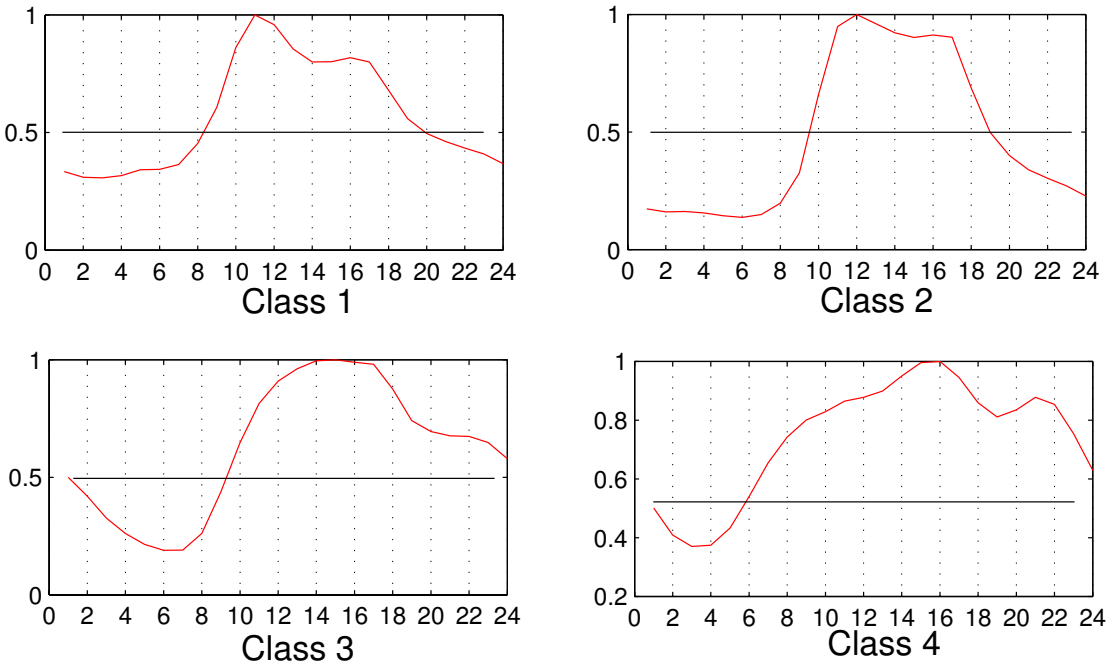

Figure 11: Four classes of request patterns

the request patterns basically fall within one of four common request pattern profiles, each of which are illustrated in Figure 11. This observation has been based on our devised algorithms for profiling, clustering and classification of Web traffic, which is next described in more detail.

Sample #	Template 1	Template 2	Template 3	Template 4
1	**0.0928**	0.2375	0.2440	0.3528
2	**0.0593**	0.1431	0.1735	0.3410
3	**0.0928**	0.1466	0.1735	0.3410
4	0.1665	**0.1014**	0.2001	0.3609
5	0.2446	**0.1025**	0.1735	0.3528
6	0.1463	**0.1135**	0.2329	0.3609
7	0.2076	**0.1135**	0.2063	0.2981
8	0.2343	0.1942	**0.0529**	0.1768
9	0.2199	0.2512	**0.0671**	0.2000
10	0.2251	0.1491	**0.0731**	0.2625
11	0.2272	0.2284	**0.0731**	0.2141
12	0.2386	0.2138	**0.0508**	0.2240
13	0.1764	0.1905	**0.0748**	0.1799
14	0.3536	0.3776	0.2585	**0.1103**
15	0.3655	0.4265	0.2529	**0.0914**
16	0.2948	0.3457	0.1972	**0.0830**
17	0.2556	0.2916	0.1232	**0.0983**
18	0.2726	0.3003	0.1437	**0.1103**

Table 2: Distance to the four templates

	Template 1	Template 2	Template 3	Template 4
Template 1	0	0.1911	0.1923	0.3410
Template 2	0.1911	0	0.2001	0.3609
Template 3	0.1923	0.2001	0	0.2055
Template 4	0.3410	0.3609	0.2055	0

Table 3: Distance between the four templates

5 Classification

The templates resulting from the above construction then can be used to recognize incoming request patterns. First we use the templates to recognize the existing samples and to check if the results are consistent with the clustering. The recognition results are shown in Table 2, which are identical to the clustering results. The distances between templates are shown in Table 3. Obviously the distances between the templates are significantly larger than the distances between any template and its members. For the last eight samples (i.e., sites) that contain only one day's worth of data, we use the similar pattern recognition methods for the classification analysis. The results are shown in Table 4. It shows that the pattern can be roughly recognized even from one-day raw data without smoothing over several weeks. This is useful for real-time collocation of workload among several web sites located in different time zones across the world.

The classification procedure consists of:

i. Using Equations (1) and (2) to compute the normalized peak hour request pattern for newly collected data.

Sample #	Template 1	Template 2	Template 3	Template 4
19	**0.3529**	0.4433	0.4536	0.4407
20	**0.2583**	0.3012	0.4051	0.4000
21	**0.1022**	0.2368	0.2669	0.3609
22	**0.1695**	0.9776	0.1702	0.3235
23	0.2386	**0.2059**	0.2468	0.3609
24	0.2386	**0.1375**	0.1959	0.3528
25	0.2149	0.2149	**0.1168**	0.2152
26	0.2386	0.1570	**0.1072**	0.2385

Table 4: Pattern recognition of one-day behavior

ii. Using Equation (3) to compute the distance measures with respect to the four templates given by Table 1.

iii. Recognizing the per-class request pattern template that is the closest match.

Since these four templates may not cover all possible request patterns, a new type of request pattern or an extraordinary request pattern may be far from all of the four templates (e.g., the minimum distance is greater than 0.19, in referring to Table 2 and Table 3). In such cases the new request pattern should be carefully checked and then added as a new template for this new class of request patterns.

6 Applications

There are a number of applications of the results presented above based on Web traffic profiling. It is beyond the scope of this paper to provide details of these applications which will appear in a companion paper. Rather, we briefly discuss some of these possible applications below.

Traffic prediction. One of the key issues in capacity planning is the prediction of workload behavior. Typically, the prediction mechanisms need to capture the characteristics of long-term trends, periodicity, dependency and variability. It is difficult to use a single technique to capture all these factors. It is therefore interesting to consider hybrid techniques pertaining to both macro and micro level statistics. While the long-term trends can quite easily be measured using linear regression methods, the periodicity at different time scales (e.g., monthly, weekly and daily) can be handled using the profiling technique presented in the paper. Thus, the high-level workload can be predicted in this way. Micro-level analysis can be further used for the detailed predictions.

Capacity planning. The clustering approach can greatly simplify the capacity planning task in, say, a Web server farm context. Indeed, we can first focus on the templates (or profiles) of different clusters, and analyze the capacity demand for each individual profile and the impact of the scaling factor as well as of the mixtures of the profiles. Then, when a new customer comes along, we can use the classification technique to determine the cluster to which it should belongs so as to adjust the capacity requirements, if necessary. In the same way, we can also easily adjust short-term capacity planning decisions if some Web sites cause the clusters to change because of special events or Web site redesign.

Hot-spot detection. With special events, e.g., holiday sales for e-commerce Web sites, some Web sites can be heavily loaded and thus require certain additional operations in order to fulfill the needed quality of service; examples of such operations include offloading (see discussions below) and adding new resources. The key question concerns how to detect these "hot-spots" in order to trigger the appropriate mechanisms at the right time. With the profiling approach, this kind of hot-spot can be detected once we observe that the current workload is deviating significantly from typical behavior which is described by the templates.

Dynamic off-loading. As mentioned in the previous paragraph, sometimes dynamic offloading operations are needed in order to alleviate server overload. One way to achieve this is to create new (or use different) versions of the Web pages with references of offloadable objects (such as images) to the servers onto which some of the extra load can be offloaded (such as Akamai servers). With the templates, it is easy to determine what is the threshold beyond which traffic should be offloaded. Even better, we can be proactive: as these offloading schemes have short or long lag times (the time between when the offloading scheme is in place and the time epoch when this has taken effect), we can start the offloading scheme before the server is saturated. Again, this new threshold can be determined from the template together with the lag time.

Web site co-location. The basic idea of co-location is to share resources (such as Web servers) among multiple entities (such as Web sites) so that peak load conditions for any given entity can be handled by borrowing resources from other entities. An important problem in this paradigm concerns the clustering of the Web sites for resource sharing. Based on the observed traffic templates, one can easily identify the shapes of the peak regimes for the different sites, as well as the different traffic peaks and valleys at different times and at different geographical time zones. Optimization tools can then be used to achieve load balancing across a number of Web sites in order to obtain the smoothest possible overall peak loads. The optimization problem can be considered as a general bin packing problem where the items are the templates. This also can be formulated as an integer programming problem.

Observe that these templates also can be obtained for measurement data other than the request process, such as the server utilization, bandwidth consumption, etc. In this way, the problems mentioned above can be addressed in a similar way with respect to these other workload processes.

7 Remarks and Conclusions

In this paper we developed techniques for efficient and automated profiling, clustering and classification of Web workload patterns based on macro-level behaviors. In contrast, previous research has focused on the complexities of Web traffic at micro-levels and individual macro-level behaviors. Our algorithms efficiently and automatically characterize Web traffic patterns from empirical data, for efficient and automated clustering of such traffic patterns based on the measurement of dissimilarity between different patterns, which characterizes the peak hour workloads and masks the effects of the off-peak hour workloads. We devised a novel algorithm for determining the templates of classes of traffic patterns, where four templates are identified from the data of more than 25 existing commercial Web sites. We then developed a method for recognizing Web traffic patterns by measuring the distance (or dissimilarity) between the workload and each of the templates.

In addition to the foregoing methods for efficient and automated profiling, clustering and classification

of Web traffic, our paper can be exploited in a wide variety of application areas including traffic prediction, dynamic offload, collocation of Web sites, short-term and long-term capacity planning, etc. The paper also can be exploited to examine the individual components of Web traffic either in isolation or in all possible combinations, where such components include (but are not limited to) user request patterns, different types of Web content (e.g., static, dynamic, continuous media, etc), different types of resource requirements and usage (e.g., network bandwidth, CPU, memory/cache storage, I/O bandwidth, disk storage, etc), different types of user requests (e.g., browsing, database transactions, secure transactions, etc).

References

[1] M. F. Arlitt and C. L. Williamson. Web Server Workload Characterization: The Search for Invariants. In Proceedings of the ACM SIGMETRICS '96 Conference on Measurement and Modeling of Computer Systems, pp. 126-137, Philadelphia, PA , May 1996. ACM.

[2] M. F. Arlitt and C. L. Williamson. Internet Web Server: Workload Characterization and performance implications. IEEE/ACM Transactions on Networking, 5(5):631-645, October 1996.

[3] P. J. Brockwell and R. A. Davis. Time Series: Theory and Methods. Springer Verlag, 1987.

[4] M. E. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. In Proceedings of the ACM SIGMETRICS '96 Conference on Measurement and Modeling of Computer Systems, pp. 160-169, Philadelphia, PA , May 1996. ACM.

[5] M. E. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. IEEE/ACM Transactions on Networking, 5(5):835-845, December 1997.

[6] J. A. Hartigan. Clustering algorithms. New York: Wiley (1975).

[7] A. K. Iyengar, M. S. Squillante, and Li Zhang. Analysis and Characterization of Large-Scale Web Server Access Patterns and Performance. World Wide Web, Vol. 2, June, 1999.

[8] A. K. Jain and R. C. Dubes. 1988. *Algorithms for Clustering Data*. Engelwood Cliffs, New Jersey: Prentice Hall.

[9] M.G. Kendall and J.K. Ord. Time Series. Oxford University Press, 1990.

[10] M. S. Squillante, David D. Yao, and Li Zhang. Web Traffic Modeling and Web Server Performance Analysis. In IEEE Conference on Decision and Control, December 1999.