Workload Characterization for an E-commerce Web Site

Qing Wang
Department of Computer Science
University of Saskatchewan
Saskatoon, SK S7N 5A9
Canada
Email: qiw237@mail.usask.ca

Dwight Makaroff
Department of Computer Science
University of Saskatchewan
Saskatoon, SK S7N 5A9
Canada
Email: makaroff@cs.usask.ca

H. Keith Edwards
Department of Computer Science
University of Western Ontario
London, ON N6A 5B7
Canada
Email: hkedward@uwo.ca

Ryan Thompson
SaskNow Technologies
Saskatoon, SK, S7K 1Y4
Canada
Email: ryan@sasknow.ca

Abstract

Electronic commerce servers have a significant presence in today's Internet. Corporations require good performance for their business processes. To date, little empirical evidence has been discovered which identifies the types requests that users make of E-commerce systems. In this paper, we examine the request level characteristics of the web site of a multinational car-rental company based on a 24 hour web server log. Our main conclusions are: i) An E-commerce web page typically contains many small image files and some popular image files are always requested together; ii) The percentage of requests for each service tends to be stable throughout the day when the time scale is large enough (10 minutes in this case); iii) Significant proportions of the requests are for dynamic pages and require the Secure Socket Layer protocol (SSL); and iv) most web objects are either requested primarily through SSL or primarily through non-SSL.

One of the performance implications with respect to the image request patterns is that these image files should be bundled to reduce the number of requests a client issues as well as the server overhead to transfer these small image files separately. The server should arrange its resource allocation taking the request mix into account in order to improve performance. Finally, the use of SSL with respect to the various objects suggests that further study is needed to determine for which pages it is appropriate to use these security measures as they have both performance and security implications.

Keywords: electronic commerce, performance evaluation, workload characterization

1 Introduction

For E-commerce web sites, and web sites in general, understanding the characteristics of the workload is the basis upon which to improve server performance, to perform capacity planning and to provide Quality of Service (QoS) to customers. The composition of an E-commerce workload can be examined from both the request-level and the session-level. The atomic unit of workload presented to an E-commerce server is a **request**, and the workload is a stream of requests. However, E-commerce workloads also naturally consist of **sessions**. A session is defined as a sequence of

requests issued by a single customer during a visit to an E-commerce site. E-commerce workload can be characterized at both the request-level and session-level. In this paper, we consider request-level characterization.

Much work has been done on characterizing web workloads (for example [1, 3, 5, 6, 7, 10, 13, 14], among others too numerous to mention), and there have been many different perspectives taken. Traces were taken from web servers, web proxies, and web browsers. What makes an E-commerce workload distinct from other workloads is the emphasis on a goal-oriented session that involves transactions, a significant amount of database activity and regular third party interactions (i.e. payment servers). E-commerce servers differ from traditional Web servers as there are additional requirements for transactional support, state maintenance and persistent and reliable storage [8]. The details will be described further in subsequent sections of the paper as necessary.

It is these differences which suggest that request-level workload characterization for E-commerce servers should be performed in a way different from that for other web servers. Some aspects of customer behaviour, such as the mix of requests for different services, should be explored. Factors associated with the resource usage of the system, such as percentage of requests going through SSL, should be characterized. We can obtain some ideas on how the resources of the system should be allocated, with respect to the goals of the customers and ultimately the vendor, by providing a characterization of these factors.

Previous research [4, 11, 12, 15, 17] has provided highly valuable results and in-depth analysis of E-commerce workloads. However, the number of published studies on E-commerce workload is still quite small due to difficulties in obtaining real workload raw data. Given the diversity and the rapid development in both the services and technology associated with E-commerce systems, more up-to-date workload studies are necessary.

This paper characterizes the workload for an E-rental business based on a server side web trace. The key results obtained are:

1. The popularity of image files does not follow the Zipf distribution. An e-commerce web page typically contains many small image files and some popular image files are always requested together. The request pattern is partly responsible for the deviation from the Zipf distribution. The performance implication is that image files that are always requested together should be bundled to reduce the number of requests a client have to issue and the overhead to transfer these small image files separately.

2. The percentage of number of requests for each service tends to be stable when the time scale is large enough (10 minutes in this case). This indicates customers are looking for similar services throughout the day. The performance implication is that the server should arrange its resource allocation taking the request mix into account in order to improve performance without concern for time of day effects.

3. Most web objects are either requested primarily through SSL or primarily not through SSL; only a very small percentage are requested with about the same probability through SSL and non-SSL, indicating a clear distinction among types of pages in the site. Intuitively, it would seem that SSL objects are associated with the most important business transactions, thus are requested more frequently, consuming more server resources.

There are also some other results which are consistent with previous research.

1. Services to customers are all delivered through dynamic web pages. The requests for dynamic pages in an E-commerce workload is much higher than that for other types of web application workloads.

2. Although client caches and web proxies have reduced the number of requests for images, there are still up to 88% of requests arriving at the server for images.

3. The popularity of dynamic pages follows the Zipf distribution for the most popular pages that account for the top 93% of

the requests. When accesses from robots are discarded, the percentage of requests accounted for by the Zipf distribution is 94%.

The remainder of this paper is organized as follows. Section 2 contains the related work and motivation in more detail, while Section 3 describes the web server logs used in this study. The characterization of the workload at the request level is given in Section 4, discussing the popularity of web objects, request arrival process, request mix and requests through SSL. Section 5 discusses the implication of the workload characteristics to server performance and resource management. We conclude in Section 6 and discuss future work.

2 Related Work

There are many types of web sites in existence today. The predefined bookmark structure of most web browsers attempts to categorize these sites for their users in an attempt to ease navigation through the entire web. Some of the categories of web sites include:

- News/Weather (updated regularly, static pages),
- Government (updated irregularly, archival content),
- Software distribution sites (static pages, but dynamic content of files),
- Web front ends to Email servers,
- Business to Consumer (B2C) E-commerce,
- Business to Business (B2B) E-commerce,
- Consumer to Consumer (C2C) E-commerce (E-bay, auctions),
- University web sites,
- Search engines.

They vary in their purposes as well as the nature of the data that they deliver. Various technologies support the delivery of web content to clients and browsers. Recent trends in the use of web sites, however, and the functionality they provide has made the distinction between them via any observable measures increasingly difficult.

A recent study by He and Yang [9] isolates some major characteristics with respect to content on home pages for web sites. They note a high occurrence of embedded objects, while the home pages themselves were always less than 100 KB of HTML content. Most home pages were between 10 KB and 50 KB in size. While this does not indicate the relative use of bandwidth by the major object types, their study shows that dynamic web content provides a significant proportion of the content in a web site.

The further development of dynamic web content and ways in which to transmit it (.jsp, .asp, .cgi, etc.) has dramatically changed the interface for many web sites, making them appear more like canonical B2C E-commerce sites. For example, university web sites allow registration, tuition payment and library services via dynamic web page generation and secure transaction-based processing. One would be hard-pressed to call this an E-commerce server, yet the kind of traffic generated may be very similar. Databases are used heavily to retrieve selections based on query parameters, and third party financial institutions may be involved in the transactions. The use of automatic redirection, server clusters, multiple-level server architectures and content distribution networks further complicates any attempt at analysis.

For the purpose of this study, we consider E-commerce sites to be those owned by for-profit organizations that use their web site for both promotional purposes and the sale or exchange of goods and services. It is part of future work to compare the workload that is generated by more recent instantiations of the web sites of other types of institutions and organizations.

The traditional view of a web server is a site that serves HTML pages, which are static or change infrequently. The major sources of data for research on web behaviour have been local ISPs [16], university web sites [15], as well as the 1998 World Cup Soccer web site [3]. While these are actual sites with real datasets, it is not clear if they provide typical data for all types of web sites.

Most published studies on workload characterization on web servers use information servers in the most traditional sense [1, 3, 5, 6, 7, 10, 13, 14]. Characteristics considered important are: file size distribution, file popularity, request arrival process, etc. Some of the important results are as following: 1) HTML and image files account for 90-100% of requests; 2) File size distribution is Pareto; 3) File inter-request times are independent and follow exponential distribution; 4) Web server traffic is self-similar, and 5) file popularity follows a Zipf distribution. These results are based on information servers and are several years old, but are still helpful in understanding E-commerce workloads.

In recent years, there have been some publications on workload characterization for E-commerce servers [4, 12, 17]. Some common findings include: 1) arriving requests are bursty; 2) the portion of requests for dynamic objects is quite significant, in comparison to "traditional" Web server workloads; 3) the percentage of requests generated by robots are quite significant [4, 12]; and 4) file popularity follows Zipf-like distribution [12]. Results 1) and 4) confirmed similar results for information servers.

Some results are unique to a specific site, or have not been confirmed by other studies. Menascé et al.[12] found that more than 70% of the functions performed are product selection functions. Arlitt et al.[4] found that requests can be grouped into three classes: cachable, non-cachable and search, based on their demands on system resources. System scalability can be discussed based on this classification system.

Our characterization differs from previous research in several aspects:

- The types of E-businesses being studied in previous studies include: web-based shopping (i.e. retail, bookstore, etc.) [4, 12], and auction sites [12]. In this study, the business is a rental site. Since different E-businesses may attract different workload, it is important to characterize workload for different business types. In particular, we expect it would be rare to have multiple "items" in a shopping cart for a car-rental business, whereas at a bookstore, or hardware store, multiple items are often purchased in the same visit to the site.

- Instead of trying to cover every aspect of request-level workload characterization, we focus on a few interesting points, which are specific to E-commerce sites. When considering the file popularity, we separately examine the popularity of embedded images and objects explicitly requested by clients, as well as requests from robots. In this case, objects explicitly requested by clients are all dynamic objects and are not cached. They are of particular interest to server resource usage.

- We characterize the SSL usage. Vallamsetty et al. [17] mentioned that a large proportion of requests come in secure mode, but provided no characterization data.

- We also study the request mix in order to obtain a general idea on how customers used the site, which has not previously studied in an explicit manner.

If sufficient data can be obtained, the common attributes of the workloads can be generalized to other types of web servers. The use of a web interface for email is similar to a transaction for purchasing an item in an on-line store, as authentication must be performed and privacy must be ensured. As well, a service such as campus room bookings are subject to the same database concurrency constraints as airline ticket sales.

3 Web Server Logs

The data used in this study is composed of access logs collected from two servers used for a web site operated by a multinational car rental company. The data collected was from the central web servers for the entire corporation. The logs record the interaction between customers and the system in one day (24 hours), and contain just over 2 million entries and totalling 700 MBytes of data. This was the Sunday following the US Thanksgiving holiday in 2001, just 2 months after the WTC attacks. Since this is a world-wide site, it encompasses Monday in

Asia and Australia, though the primary market of this company is in the western hemisphere.

Unfortunately, this is the only trace we have been able to obtain. It is unclear whether the request arrival pattern is typical of all car rental companies or even this particular company. There is no evidence to indicate that access to the site was particularly abnormal that weekend, although the Thanksgiving weekend in the US has the heaviest travel of any weekend of the year. The day on which the data was collected was at the end of that weekend, however, and may not reflect the peak activity associated with Thanksgiving.

Table 1 lists the summary characteristics of the logs. The logs are combined in this study because they contain the traffic sent to the corporate web site, but served by two servers. This is confirmed by the fact that in many cases, both logs contain requests from the same IP address with the same cookie at approximately the same time.

The web server is Microsoft Internet Information Server 5.0 (IIS). IIS can write to multiple logs using standard and extended W3C's log formats and can even support custom logging. The log used in this research is in W3C extended log format. An entry in the log has 21 fields and represents one atomic request from a browser application.

Table 2 lists the breakdown of requests by type of requested web objects. The type of an object is determined by the URL file extension. The main types of requests found in this log are: Images (.gif and .jpg), Dynamic pages (.asp), JavaScript (.js), Cascading Style Sheets (.css), and HTML. A very small number of requests were for text files and limited numbers of other types of files.

Requests to web servers are divided into two categories: the first is requests issued explicitly by end-users for web pages containing services they want and the second is the requests issued automatically by web browsers for embedded objects in the web pages requested by users.

It is not hard to tell whether a web object is embedded. In web pages, embedded objects are referenced by special tags. Embedded objects can thus be identified by examining web pages. However, this is a tedious method. The main components of embedded objects are image (.gif, .jpg), javascript (.js), and cascading style sheet (.css). Thus, it is sufficient to assume that all image files, js, and css are embedded. In this case, requests for these objects make up about 94% of the workload. HTML files may also be embedded files.

There are only a few HTML files in the logs. By manual examination, it is clear that they are also embedded files. Most requests for ASP files were issued explicitly by customers. The service to customers is delivered by corresponding ASP pages. However, by examining the functionality of all ASP pages, it was found that there are also a small percentage of ASP requests for images and frames, thus believed to be embedded objects as well.

Dealing with requests for images as a part of workload characterization has been problematic in many respects. In most analyses they are ignored [4, 11, 12, 15]. Requests for embedded objects should be ignored when the objective of workload characterization is to study the customer behaviour. If the objective is to study the resource demand on the web server, however, requests for images should not be ignored, since these requests are an important part of the workload. Requests for images made up about 80-90% of the all requests, potentially consuming a large portion of the bandwidth (disk I/O) and RAM only. The CPU is not heavily used for these types of requests, since file operations can be cached and require no server processing. In some cases, there are separate image servers to handle requests for images. Thus, there is no need to consider these requests from the server point of view, but they do contribute to network traffic. In this study, there are no dedicated image servers, and we consider all objects requested from the server, except where indicated.

For the web site being studied, services to customers are all delivered through dynamic web pages. However, requests for images still made up about 88% of the all requests, which is still very high (close to the range described above for information servers [5]). Two main versions of the logs are used in the analysis: the original log and the reduced log after filtering. The filtering operation removes all embedded objects, which are mainly images (Table 2).

Table 1: Characteristic of the Logs

File Name	Start Time	End Time	Size (k Byte)	Number of Entries
log-b	2001-11-25 05:59:59	2001-11-26 05:59:59	295,193	838,195
log-q	2001-11-25 06:00:00	2001-11-26 05:37:46	358,920	1,182,527

Table 2: Breakdown of Requests by Types

	Log b					Log q				
Type	Files	Requests	%Request	MB	% MB	Files	Request	%Request	MB	% MB
Total	1,139	836,195	100	1380.9	100	1,942	1,182,527	100	2026.8	100
Image	814	736,644	88.09	652.9	47.25	828	1,054,046	88.14	921.3	45.46
Asp	303	56,181	6.72	491.4	35.59	290	88,847	7.51	786.0	38.78
Html	6	18,080	2.16	56.2	4.07	4	1,307	0.11	52.1	2.57
Js	1	8,080	0.97	152.8	11.06	1	12,259	1.04	227.9	11.25
Css	1	16,819	2.01	21.9	1.59	1	25,584	2.16	33.1	1.63
Other	14	391	0.05	5.7	0.41	18	484	0.04	6.3	0.31
Embedded	840	783,245	93.67	890.8	64.50	856	1,098,614	92.90	1242.6	61.31
Web pages	299	52,950	6.33	490.2	35.50	286	83,913	7.10	784.2	38.69

Also, requests from robots are filtered to isolate the behaviour of software agents from human users. This filtering was done manually by examining the log for client names that indicated the browser was a robot, or the fact that the file "robots.txt" was accessed [2]. Relatively few requests were from robots (0.1%), compared with other research [12] which attributed as much as 16% of the requests to robots.

4 Request Level Workload Characterization

An E-commerce workload has many characteristics that can be observed and/or calculated at the request level. Only some characteristics, which we think are specific to E-commerce sites, are discussed in this section, including the popularity of web objects, request arrival process, request mix and requests through SSL. While the popularity of web objects and the request arrival process have been discussed in many studies, discussions on request mix and requests through SSL are rare. Certain characteristics of web workload are difficult to analyze due to the length of the trace period. For example, self-similarity of the web traffic is more suited to analysis across time scales of days, hours, minutes and seconds, whereas only a 24-hour trace is available.

We noticed that a significant number of the requests had values of 0 for log entries for which 0 seems an unlikely value. In particular, 33% of the dynamic requests had 0 for the number of bytes sent from the server to the client (sc-bytes). As well, 10% of the requests had the value 0 for time-taken. Such a large percentage of zero values seemed strange. All the accesses for static pages have a reasonable number of bytes transferred, but the dynamic pages exhibit the zero-byte phenomenon. Most of the requests for zero time taken are for small images.

How could this be possible? We have some conjectures that remain to be verified, which could be possible causes for this logging behaviour. Searching through various Internet sources provided some clues as to the reason for this phenomenon. Special logging procedures are done for certain types of pages for reasons that are not explained in the web server documentation.

1. In some cases, the bytes could be listed as zero, because the request is for an ASP that must generate a page of response. The logging features of IIS set the value of sc-bytes to 0 if buffering of ASP pages is enabled. This is the default setting, and presumably the setting that was enabled during trace data collection. Since the re-

sponse is generated a line at a time, and buffered at the server before being sent, for some reason, IIS does not calculate the total number of bytes sent back as the response.

2. None of the time values in the log are smaller than 15 msec. It could be possible that any request returning in fewer than 15 msec is recorded as 0. Since most of these requests with 0 time are for small image files, it is possible that they are cached in the server RAM, and could be sent very quickly. Since they take at most one TCP packet, this could well be less than 15 msec.

3. Most of the images requested through SSL have 0 for the time taken. Nearly all of the images not through SSL have non-zero time values. Perhaps there is something in the port setting that changes the logging procedure.

It appears that the value recorded for scbytes is meaningless for most of the dynamic pages. Thus, cannot make any intelligent inferences about the actual memory and/or bandwidth required at the server to generate the responses. Buffering would need to be turned off at the server to get more realistic values. Unfortunately, this would also reduce server performance as more packets, each of smaller size, would be used to send the response back to the client on a per-line basis. Thus, the high occurrence of zero values do not have any significant impact on the results we present.

4.1 Popularity of web objects

In general, the popularity of web requests has been shown to follow the Zipf distribution [5]. When the rank popularity of objects is graphed against frequency of requests on a log-log scale, the result should be a straight line if the relationship has a Zipf-like distribution. The slope of the line indicates the parameter α of the Zipf distribution.

Figure 1 (a) shows, however, that the popularity of image files (requests received at the server) does not follow Zipf's law. Instead of being a straight line, the popularity curve for image file has many "steps" (Figure 1 (a)). A "step" on the curve is formed by groups of image files which have almost the same popularity. For example, the image file ranking from the 3rd to about 20th have almost the same popularity and thus form a step on the curve. A close examination of these image files revealed that these image files are actually embedded in the same web page and thus are always requested together.

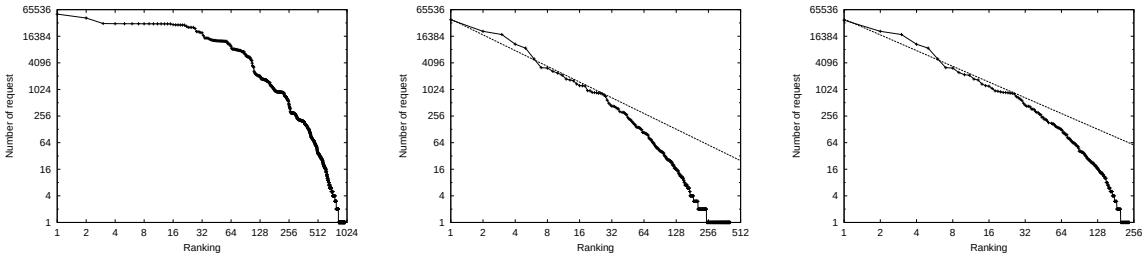
Another possible cause for the popularity of image files not showing a Zipf-like distribution is that a significant portion of requests for images may have been satisfied by client or proxy caches, reducing requests for images. Further analysis confirmed this conjecture. The log files show that 25.8% of the requests for images generate a response with the HTTP status code of 304. When a client requests an object which is still in the cache but has an expired timeout value, the cache will issue a conditional GET request to the server to check whether the object has been modified. If the object has not been modified, it will answer the conditional GET request with a status code 304, meaning that the object has not been modified and proxies can send their copy to the client to satisfy the request.

The fact that 25.8% of requests for images have the HTTP status code of 304 indicates that the overhead to maintain cache consistency is high in this case. The time-to-live parameters for object being cached should be adjusted.

In spite of the effectiveness of client or proxy caches, a large percentage of incoming requests are still for images since there are many embedded images in web pages. The performance implication is that the server cache is still very necessary, in addition to effective client and proxy caches.

Figure 1 (b) shows the popularity of dynamic pages; it is Zipf-like for most of the data points. There are 408 unique uri-stems. The top 32 pages account for 93% of the requests, and this follows the Zipf distribution. The remaining 7% of the requests are referenced in a pattern that is still under investigation.

This is somewhat consistent with Arlitt et al. [4]. Since dynamic pages were not cached and every request for dynamic page



(a) Embedded Objects

(b) Dynamic Pages

(c) Dynamic Pages - no robots

Figure 1: Popularity of Pages

has to reach the web server, this reflects the real popularity. The performance implication is that caching a small number of dynamic pages can be very beneficial. However, caching dynamic web objects is difficult. Most popular dynamic objects are requested through SSL, which makes them almost inherently non-cachable.

The vast number of pages that were requested only one time as well as those pages requested fewer than 10 times accounts for a large number of the web pages, but a very small percentage of the actual requests. This may be a result of the short length of the trace. 40% of the unique uri-stems were requested only once and 65% were requested 10 or fewer times.

When requests from robots are filtered out, the shape of the popularity graph remains the same (Figure 1 (c)), but many pages are no longer referenced. There are only 235 unique uri-stems and only 14% of them are one-timers, while 37% are requested 10 or fewer times. This is somewhat different than previous results which indicate a large number of one time requests for general information servers [5], who noted that up to 33% of the requests were one-timers in a large trace.

A significant proportion of the robot accesses were to areas of the site that were not popular. Filtering the robot accesses removed 175 one time requests. For example, the Employment Opportunities area had many one time requests definitely issued by robots, and only a very small number of requests that appeared to be accessed by humans. Other areas of the site visited by robots were weather forecasts, and some special promotions. Although these requests take up server resources, it is clear that robot requests should be given lower priority, as they do access unpopular pages in a sequential fashion and quality of service is not important for these clients.

4.2 Request Arrivals

Figure 2 (a) shows the number of requests arriving at each time period over the day. In order to focus on customer behaviour, embedded objects were filtered and the two trace files (logs b and q) were merged. The scale of the time slot is 1 minute. This traffic shows a slight time of day fluctuations with a long peak period between 800 to 1300 minutes. During these times, the traffic is substantially bursty. On average, the number of requests per minute is about 40 when the server is the least busy (around time slot 300), in contrast to about 135 when the server was the most busy (around time slot 1200).

Figure 2 (b) shows that the burstiness of the incoming requests had only a small effect on the System Response Time (SRT) averaged over the same time interval. The SRT for a request is the period of time the system takes to respond to the request; it includes queuing time, CPU time and disk time at each of the web server, application server, database server and payment servers. During the early part of the day, when the request arrivals were very infrequent (between minutes 200 and 400), the range of response times is narrow and the average response time is at its minimum. When arrivals reach a certain threshold (approximately

40 requests per minute), the response times increase and the variability increases dramatically. It is worthy to note, however, that the burstiness of arrivals has no effect on average SRT, the range for SRT per request is quite stable for the day. This indicates that the resources available for a request are sufficient over the course of the day, i.e. the server did not reach its capacity even at the busiest moment. It is impossible to apportion the components of time-taken among the server applications and the database, but the stability indicates that at least the web server was not inserting delays that are linear with the arrival rate.

A further analysis of the response times produced some interesting and unexpected results. While a full statistical analysis has not been completed, some general trends can be observed. Figure 2 (c) shows an X-Y plot for the request arrival rate and response time averaged over 1 minute intervals. Some time intervals have very long average response times, even though very few requests were issued (the points in the top left corner). Graphs were constructed for 10 second intervals and 10 minute intervals and they show similar characteristics. They are not shown due to space considerations. We observed the following general characteristics of the graphs:

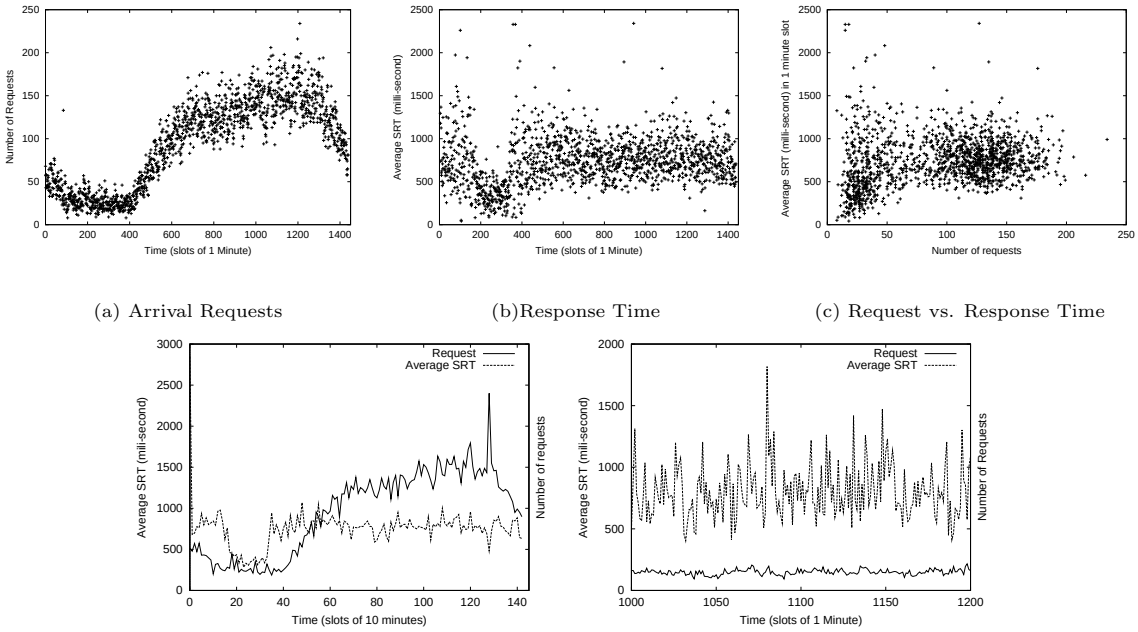
- Once a threshold of requests per minute is reached (in this case, 70 requests), the average response time never goes below approximately 300 msec. Thus, there is some minimum queuing time that is required that prevents an immediate response.

- With a request arrival level above the threshold, the range of average response times is very similar. The average response time for 100 requests per minute is very similar to the average with 170 requests per minute. If the server had been continuously busy, there would be a somewhat linear relationship between request rate and response time, due to uniformly longer queue lengths. Figure 2 (d) shows the relationship between response time and request arrival over time in intervals of 10 minutes. We can see that the request arrival rate is increasing over the last several hours of the trace, but the response time stays constant.

- When the server request rate is low, the range of response times is greater. This indicates that the response time likely has more to do with the types of requests and correlated burstiness than the volume. These requests will likely make different uses of the database or other application servers. This contributes more to the variance than queuing at the web server. Though we show later that the request mix is stable over long periods of time, it is bursty over short intervals and the long response time could be due to complicated requests being received in that time period, or transient queue buildups.

- There is some correlation between the request rate in one time period and the response time in the time periods that follow. This indicates that the response time is affected by the request arrival rate in previous time periods, where queues could have built up and need time to drain. This effect is shown in Figure 2 (e), which is an excerpt from time 1000 minutes to 1200 minutes. Other time periods show similar effects. In particular, when the request rate decreases substantially after a peak in request rate, the queue could have time to drain decreasing the response time. The effect when the rate stays high for several time periods is less clear. Over the day, there are enough periods of low activity to reduce queue lengths. Again it is important to note that attributing response time to queue length is speculative at best, due to the lack of logs from the database and application servers.

Overall, it seems that the response time does not show a strong correlation to the request arrival rate, indicating that the long response times are due to the burstiness of arrivals or the particular type of arrivals and that substantial periods of lower activity are found throughout the day.



(a) Arrival Requests (b)Response Time (c) Request vs. Response Time

(d) Response Time (10 minute intervals)

(e)Response Time (1 minute intervals)

Figure 2: Request Patterns

4.3 Request mix

For an E-commerce site, there are hundreds and even thousands of web pages. To reduce complexity, web pages are grouped by functionality. If two web pages share the same URI (Uniform Resource Indicator) prefix, they have the same general functionality and can be grouped together via a manual classification of URI prefixes. There are 408 unique web pages (uri-stems) in the reduced log, which can be grouped manually into about 17 services (Table 3).

Figure 3 shows the percentages of the number of the requests for page "rQut" in the total number of requests for all pages over time, calculated on time slots of 10 second, 1 minute and 10 minutes, respectively. The range for the percentages become smaller as the time scale increases. Obviously, the trend is that the percentage tends to be stable over time when the time slot is large enough, which is close to be 10 minutes in this case. In fact, this trend is observed not only for the request type "rQut", but also for all other request types shown in Table 3. In summary, the request mix is stable over time, regardless of the total number of incoming requests. This stability indicates that customers are looking for similar services throughout the day. This is consistent with Menascé et al. [11]. When the time scale is relatively small, requests looks random. Once the time scale is large enough, the random nature of individual requests will provide a uniform distribution for the request types.

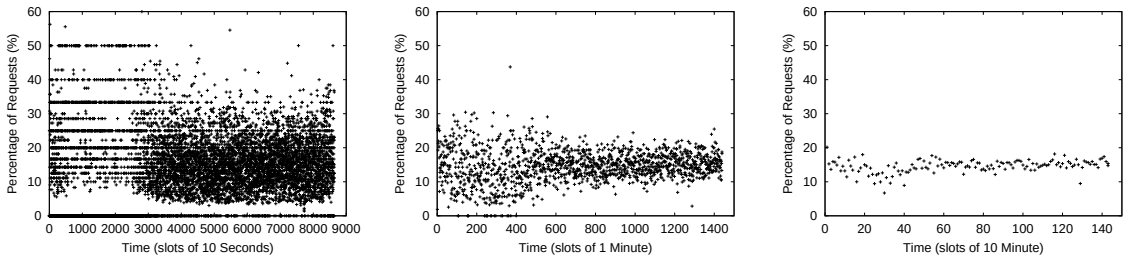
4.4 Requests through Secure Socket Layer (SSL)

Secure Socket Layer (SSL) is an important component of an E-commerce server. The way a request is processed with SSL is different than without SSL. In general, requests through SSL post a higher demand on system resources. In E-commerce workload characterization, it is important to take into account the SSL.

A key observation is that for each particular object, its requests have a strong probability

Table 3: Grouped web pages for the site

Web Pages	Abreviation	# of requests	% of request	% of request through SSL
hom	home page	17221	12.6	4.24
expL	express lane	741	0.54	26.2
gSrv	group service	948	0.69	10.8
info	info, help	7141	5.2	46.0
loc	available locations	5847	4.3	0
othr	others	1101	0.80	4.6
prmt	promotions, special offers	5323	3.9	2.1
rChR	check rate	19659	14.4	100
rCnl	cancel reservation	845	0.62	100
rHom	base page for reservation	23699	17.3	5.1
rMkR	make reservations	819	0.59	100
rMod	modify reservations	1251	0.91	100
rPpU	popup search info	21420	15.6	37.6
rQut	reservation quote	20430	14.9	80.9
rVew	view reservations	3794	2.77	100
trvl	travel information	1842	1.35	0.7
vhcl	vehicles to choose	4799	3.50	23.6



(a) (b) (c)

Figure 3: Percentage requests for web page "rQut"

to either use SSL for the primary access method or to not use SSL access (Table 4). Very few objects show similar frequency of SSL and non-SSL requests. In Table 4, a web object is classified by the percentage of requests for this object through SSL. If none of the requests for a web object are through SSL, this object is referred to as a non-SSL-object; if over 90% of the requests are not through SSL, then the object is referred to as a 90%-non-SSL-object, etc. Of all objects at the web site being studied (including embedded objects), only 9% are SSL-objects and 5% are 90%-SSL-objects, 57% are non-SSL-objects and 18% are 90%-non-SSL objects. Obviously, only a small percentage of web objects are SSL-oriented.

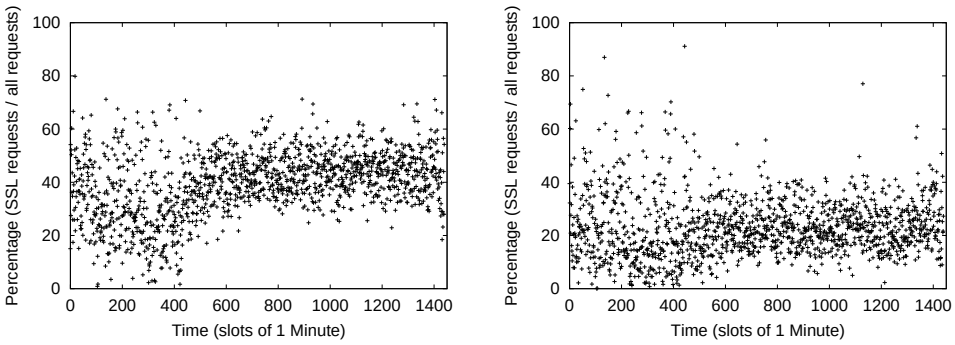
The non-SSL objects do not necessarily provide as large a proportion of the requests that 57% might imply. Many of these objects are requested only once. For the SSL-objects, a large proportion of them were images. 90% of the SSL objects requests were for images, for a total of 8% of all objects.

On the time slot of 1 minute, the average percentage of requests through SSL is about 42% (ranging from from 30% to 50%) (Figure 4 (a)). Only about 20% of SRT was spent on SSL (Figure 4 (b)), which is unexpectedly low in comparison to proportion of SSL requests. This phenomenon occurs because of the fact that many of the image requests through SSL have 0 time values. When images are removed, the substantial time taken is removed from the non-SSL requests, but nearly no time is removed from the SSL requests, increasing the percentage of time spent in SSL significantly.

When considering the filtered log, the importance of SSL requests is more obvious.

Table 4: Partition of web objects based on SSL usage

Classification of web objects	non-SSL-object	90 % non-SSL-object	-	90 % SSL-object	SSL-object
% of requests through SSL	0	$0 < 10$	$10 < 90$	$90 < 100$	100
Partition of web objects (%)	57	18	11	5	9



(a) Request

(b) SRT

Figure 4: SSL Requests (before filtering)

Although the average percentage of requests through SSL is still about 42% (Figure 5 (a)), the average percentage of SRT is as high as 90% (Figure 5 (b)).

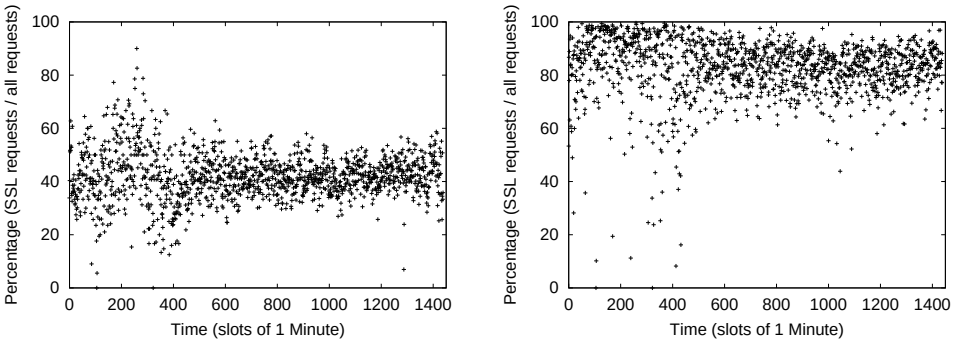
Although the number of web objects requiring SSL access is relatively small, these objects are key components of the server's functionality and thus the relative frequency of access to these objects is high. Requests associated with revenue generation are often through SSL. Almost all requests for services directly connecting to car reservation, including pages rChR, rCnl, rMkR, rMod, rQut, and rVew in Table 3, are through SSL. The performance implication is that, if the server is to provide service differentiation (or QoS), requests through SSL should be given a higher priority.

5 Implication for Server Management

The previous section has noted several characteristics regarding the requests that are presented to the E-commerce web server in this study. In this section, we briefly analyze some implications that these observations have for web site hosts and offer some suggestions with respect to providing enhanced performance. The limited trace period does not allow us to make strong conclusions about the behaviour over time. We note that, when averaged over significant time periods, the resources requested at the server are fairly consistent. Thus most of our conclusions relate to the distribution of requests over SSL and embedded object retrieval.

A group of image files embedded in the same web page tend to have the same popularity since clients always request them together. A web page on a E-commerce site typically contains several images. Clients request them individually. These embedded images are typically quite small in file size, and the overhead involved in logging and setting up connections would be significant for such files which are very small. Persistent connections available with HTTP 1.1 would alleviate this particular problem, but 20% of the requests in the logs use non-persistent HTTP 1.0. A more recent trace may have a substantially lower proportion of HTTP 1.0 requests.

If the web page is a popular one, such as the home page for a site, the amount of



(a) Request

(b) SRT

Figure 5: SSL Requests (after filtering)

workload and overhead generated by requesting these images is significant. To improve server performance, an "application-aware" caching technique should be introduced. The idea is that for a popular web page, its embedded images should be cached as a bundle so that a client needs only one request to get all embedded images. Thus, the server can send these images all in one reply. To do this, the server must be aware of the site's configuration and there must be a mechanism for the server to provide site configuration to caches and clients. Still, there is a request made to the server and the logging operation that has to occur for each image file, each of which represent unnecessary overhead.

In general, the popularity of dynamic web objects follows a Zipf-like distribution, indicating that caching would be beneficial for system performance. However, for E-commerce servers, caching dynamic pages is difficult not only because of the dynamic nature of the pages, but also due to the fact that a significant percentage of requests for dynamic pages are requested through SSL. The same URL is used to identify the page, but the query strings passed to the GET requests are different. These pages cannot be cached. The popular pages also tend to have a lot of POST methods used to access them. This is also non-cachable between various clients, as the information posted in the form would be different for every user, and almost every request.

Security is crucial in an E-commerce transaction. Customers are likely more concerned about security than QoS. It is also essential to maintain a certain level of QoS, however, for an E-business to be successful. For E-commerce servers, most pages related to revenue generation are requested through SSL. A significant number of the requests in the trace used SSL, and it is not clear whether the extra processing was required on these pages or not. To optimize server resource management with respect to revenue-generating page requests, priority should be granted to requests through SSL.

As well, providing some sort of "application-aware" approach to using the SSL protocol could increase performance. For example, some images were processed through SSL. It does not seem reasonable, at first glance, to encrypt the data for an image, which generally does not improve security. Some images may need to be protected, but deciding which data to protect should be done with a finer level of granularity. If images are located on the same page as data which needs SSL protection, security reasons suggest that the entire page be presented via SSL.

The request mix is somewhat random in a small time scale but tends to be relatively stable for a larger time scale. The performance implication is that server should arrange its resources taking the request mix into account in order to improve performance. For example, the web server can set up a queue for each type of request, then schedule jobs in such a way

that each request type gets its share of resource based on request mix. The stable request mix can also be used for capacity planning by forecasting the workload associated with different user traffic. For example, assuming sales will increase by 50%, we can get an idea how the workload will look according to the request mix.

6 Conclusions

This paper analyzes the request-level workload for a car-rental business. Although we only have data from one site for one day, we believe the result reveals some common characteristics for E-commerce workloads. Further data is necessary to confirm these intuitions based on these initial observations. As expected, the use of dynamic pages and SSL are important characteristics for E-commerce sites. In this case, services to customers are all delivered through dynamic web pages, and almost half of the requests are through SSL. Caching for dynamic pages is hard in general and doing that for E-commerce sites is even more difficult due to the SSL factor. Since the revenue related transactions are mostly through SSL, a resource management policy optimizing revenue should give priority to SSL requests.

The request mix for an E-commerce site tends to be stable over time, averaging on a time scale in an order of 10 minutes. The request mix is related to the services the site provides and the behaviour of customers. Relatively stable request mix may indicate a relatively stable customer mix. It is possible to make use this characteristic for task scheduling in order to improve server performance. This characteristic can also used for capacity planning.

Request-level characterization is only the beginning of analyzing the workload for E-commerce web sites. Since the activity is primarily transaction-focused, consideration of session-level behaviour is necessary. As part of future work, the session-level characterization of the workload will be done. This will allow an analysis of customer behaviour from a functional point of view with respect to the impact on the server and network systems.

The difference in use of the SSL protocol suggests some interesting possible future work related to the performance of the web server. It is clear that some extra processing is necessary to encrypt and decrypt the web content sent via SSL. Future work will examine the differences between response times for similar requests with or without SSL protection with more precision.

About the Authors

Qing Wang is an M. Sc. student in Computer Science at the University of Saskatchewan. His research interests are in performance analysis of electronic commerce systems.
Dwight Makaroff is an Associate Professor in the Department of Computer Science at the University of Saskatchewan. His research interests are operating systems, multimedia server systems and performance of electronic commerce systems.
H. Keith Edwards is a Ph.D. student in the Department of Computer Science at the University of Western Ontario and a CAS student at IBM Toronto Lab. His research interests are queuing theory, management information systems, and logging for distributed applications.
Ryan Thompson is a recent graduate of the Department of Computer Science at the University of Saskatchewan. He owns and manages SaskNow Technologies, which provides web hosting and application services.

References

[1] V. Almeida, M. Crovella, A. Bestavros, and A. Oliveira. Characterizing Reference Locality in the WWW. In *IEEE/ACM International Conference on Parallel and Distributed System (PDIS)*, December 1996.

[2] Virgilio Almeida, Daniel A. Menascé, Rudolf H. Riedi, Flávia Peligrinelli, Rodrigo C. Fonseca, and Wagner Meira Jr. Analyzing Web Robots and Their Impact on Caching. In *Proceedings of the 6th Web Caching and Content Delivery Workshop*, June 2001.

[3] Martin Arlitt and T. Jin. Workload Characterization of the 1998 World Cup Web Site. *IEEE Network*, 14:30–37, May 2000.

[4] Martin Arlitt, Diwakar Krishnamurthy, and Jerome Rolia. Characterizing the Scalability of a Large web-based Shopping System. *ACM Transactions on Internet Technology (TOIT)*, 1(1):44–69, 2001.

[5] Martin Arlitt and Carey Williamson. Web Server Workload Characterization: The Search for Invariants. In *Proceedings of the 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 126–137. ACM Press, 1996.

[6] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *IEEE INFOCOM*, pages 126 – 134, New York, NY, March 1999.

[7] Mark E. Crovella and Azer Bestavros. Self-similarity in World Wide Web Traffic: Evidence and Possible Causes. *IEEE/ACM Transactions on Networking (TON)*, 5(6):835–846, 1997.

[8] G. Gama, W. Meira Jr., M. Carvalho, D. Guedes, and V. Almeida. Resource Placement in Distributed E-commerce Servers. In *The Evolving Global Communications Network (GLOBECOM 2001)*, San Antonio, Texas, nov 2001.

[9] X. He and Q. Yang. Characterizing the Home Pages. In *Proceedings of the 2nd International Conference on Internet Computing (IC-2001)*, pages 976–982, Las Vegas, Nevada, June 2001.

[10] Arun Iyengar, Mark S. Squillante, and Li Zhang. Analysis and Characterization of Large-Scale Web Server Access Patterns and Performance. *World Wide Web*, 2(12):85–100, 1999.

[11] Daniel Menascé, Virgilio A. F. Almeida, Rodrigo Fonseca, and Marco A. Mendes. A Methodology for Workload Characterization of E-commerce Sites. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, pages 119–128. ACM Press, 1999.

[12] Daniel Menascé, Virglio Almeida, Rudolf Riedi, Flávia Ribeiro, Rodrigo Fonseca, and Wagner Meira Jr. In Search of Invariants for E-business Workloads. In *Proceedings of the 2nd ACM conference on Electronic Commerce*, pages 56–65. ACM Press, 2000.

[13] V. Padmanabha and L. Qui. The Content and Access Dynamics of a Busy Web site: Findings and Implications. In *ACM SIGCOMM*, pages 111–123, Stockholm, Sweden, August 2000.

[14] J. Pitkow. Summary of WWW Characterizations. *World Wide Web*, 2, 1999.

[15] W. Shi, R. Wright, E. Collins, and V. Karamcheti. Workload Characterization of a Personalized Web Site – And its Implications for Dynamic Content Caching. Technical Report TR2002-829, New York University, 2002.

[16] F. D. Smith, F. H. Campos, K. Jeffay, and D. Ott. What TCP/IP Protocol Headers Can Tell Us About the Web. In *ACM SIGMETRICS 2001/Performance*, pages 245–256, Cambridge, MA, June 2001.

[17] U. Vallamsetty, K. Kant, and P. Mohapatra. Characterization of E-Commerce Traffic. In *Fourth IEEE International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems(WECWIS'02)*, pages 137–144, Newport Beach, California, June 2002.