Workload Characterization and Customer Interaction at E-commerce Web Servers

A Thesis Submitted to the College of

Graduate Studies and Research

in Partial Fulfillment of the Requirements

For the Degree of Master of Science

in the Department of Computer Science

University of Saskatchewan

Saskatoon, Saskatchewan

by

Qing Wang

Permission To Use

In presenting this thesis in partial fulfillment of the requirements for a Post-graduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
University of Saskatchewan
Saskatoon, Saskatchewan, Canada
S7N 5A9

Abstract

Electronic commerce servers have a significant presence in today's Internet. Corporations want to maintain high availability, sufficient capacity, and satisfactory performance for their E-commerce Web systems, and want to provide satisfactory services to customers. Workload characterization and the analysis of customers' interactions with Web sites are the bases upon which to analyze server performance, plan system capacity, manage system resources, and personalize services at the Web site. To date, little empirical evidence has been discovered that identifies the characteristics for Web workloads of E-commerce systems and the behaviours of customers.

This thesis analyzes the Web access logs at public Web sites for three organizations: a car rental company, an IT company, and the Computer Science department of the University of Saskatchewan. In these case studies, the characteristics of Web workloads are explored at the request level, function level, resource level, and session level; customers' interactions with Web sites are analyzed by identifying and characterizing session groups.

The main E-commerce Web workload characteristics and performance implications are: i) The requests for dynamic Web objects are an important part of the workload. These requests should be characterized separately since the system processes them differently; ii) Some popular image files, which are embedded in the same Web page, are always requested together. If these files are requested and sent in a bundle, a system will greatly reduce the overheads in processing requests for these files; iii) The percentage of requests for each Web page category tends to be stable in the workload when the time scale is large enough. This observation is helpful in forecasting workload composition; iv) the Secure Socket Layer protocol (SSL) is heavily used and most Web objects are either requested primarily through SSL or primarily not through SSL; and v) Session groups of different characteristics are identified for all logs. The analysis of session groups may be helpful in improving system performance, maximizing revenue throughput of the system, providing better services to customers, and managing and planning system resources.

A hybrid clustering algorithm, which is a combination of the minimum spanning tree method and k-means clustering algorithm, is proposed to identify session clusters. Session clusters obtained using the three session representations Pages Requested*,* Navigation Pattern*, and* Resource Usage *are similar enough so that it is possible to use different session representations interchangeably to produce similar groupings. The grouping based on one session representation is believed to be sufficient to answer questions in server performance, resource management, capacity planning and Web site personalization, which previously would have required multiple different groupings. Grouping by* Pages Requested *is recommended since it is the simplest and data on Web pages requested is relatively easy to obtain in HTTP logs.*

Acknowledgements

I am indebted to a large number of people who have contributed to the successful completion of my Master's program. First, I would like to thank my supervisor, Professor Dwight Makaroff, for his guidance and valuable suggestions. In particular, I highly appreciate the great amount of time he spent for my program and his high availability to students. I have learned a lot from his passion, dedication, and enthusiasm to research. My thanks also go to Professor Graham Links (external examiner), Professor Rick Bunt, and Professor Jim Carter for serving as members of my thesis defense committee. Their comments and suggestions have gone a long way to improve the quality of the thesis.

Special thanks to Dr. Keith Edwards (University of Western Ontario) and Shane Doucette (University of Saskatchewan) for help in obtaining Web server access logs which were analyzed in my study. Also thanks to the organizations that own these logs for willing to provide them for free to support this study.

Thanks to Brian Gallaway (system Administrator) for providing technical supports whenever I needed it. Thanks to Jan Thompson (Graduate Correspondent) for her effective administrative helps. Thanks to my fellow graduate students for making my time in the program memorable.

Finally, I would like to express my profound gratitude to wife, and my parents who are living in China, for their love and incredible support.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ASP	Active Server Pages
B2B	Business to Business
B2C	Business to Customer
C2C	Customer to Customer
CBMG	Customer Behavior Model Graph
CDF	Cumulative Density Function
CGI	Common Gateway Interface
CPU	Central Processor Unit
CSS	Cascading Style Sheet
EDI	Electronic Data Interchange
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IIS	Microsoft Internet Information Server
IP	Internet Protocol
JSP	Java Server Pages
LLCD	Log-Log Complementary Distribution
Perl	Practical Extraction and Report Language
PHP	Hypertext Preprocessor
QoS	Quality of Service
RAM	Random Access Memory
SRT	System Response Time
SSL	Secure Socket Layer
SQL	Structured Query Language
TCP	Transport Control Protocol
URL	Universal Resource Locator
W3C	World Wide Web Consortium

XML Extensible Markup Language

Chapter 1

Introduction

With the explosive development of the Internet, E-commerce has become an important part of today's economy. Most major corporations and organizations now have public Web sites for E-commerce-related activities. Corporations want to maintain high availability, sufficient capacity, and satisfactory performance for their E-commerce Web systems, and want to provide satisfactory services to the users of their systems.

For a Web system, understanding the characteristics of workload is the basis upon which to i) improve server performance, ii) perform capacity planning, iii) manage system resource, and iv) provide personalized services to users. Synthetic workloads are typically used in a simulation or emulation study of system performance, and synthetic workload generators are built based on workload characterization results. Capacity planning involves forecasting workload. To manage system resource effectively, one must understand how it is consumed. To provide personalized services, the interaction between users and the Web site must be understood.

Workload characterizations [1, 4, 6, 8, 12, 19, 38, 48, 49] have been performed on Web systems. However, these studies were done prior to the year 2000 and the systems involved were traditional Web servers, which are information-oriented Web servers in the nineties. Those servers supported only simple functions to provide information to users. Web servers have changed greatly since then. To support current E-commerce functions, Web systems have to provide transaction support, state maintenance, and persistent and reliable storage [33]. A goal-oriented session that involves transactions is often related to a significant amount of database activities, Secure Socket Layer (SSL), and regular third party interactions (i.e. payment

servers) [54]. It is these changes that suggest that Web workload characteristics have also changed, since changes in the functions and the technologies used in Web systems will inevitably cause changes in Web workload. The understandings in Web workload characteristics need to be updated.

Since the year 2000, studies of workload characterization for E-commerce Web systems have been reported [5, 43, 45, 47, 52, 54]. These studies have provided highly valuable results and in-depth analyses of E-commerce workloads. However, the number of published studies is still quite limited due to difficulties in obtaining real workload raw data. Given the diversity and the rapid development in both the functions and technologies in E-commerce systems, more up-to-date workload studies are necessary.

In the study of workload characterization of traditional Web information servers, a request is the basic unit for analysis. Workload to a Web server is viewed as a stream of requests. Although the analysis of requests is still an important part of the characterization of workload for E-commerce Web systems, the analysis of sessions is equally or even more useful. A session is a unit of activities by a single user. E-commerce workload is transaction-oriented and the interaction between users and the Web system can be better understood at the session level. Session features can be used to improve server performance. For example, as pointed out by Chen and Mohapatra [14]: i) session integrity requires that once a session begins, it should be allowed to complete; and ii) session affinity requires that requests belonging to the same session should be handled by the same front-end server for security and locality reasons.

An important part of E-commerce workload characterization is to identify and characterize session groups. Grouping sessions provide insights into customers' interests, navigation patterns, and resource usages [5, 44]. Recognizing and adapting to session groups can be used to improve server performance (in terms of either throughput or revenue), implement admission control, perform capacity planning, and provide personalized services.

In the related literature, sessions have been grouped in different ways in order

to discuss different problems. Menascé *et al.* [44] identified session groups based on navigation patterns for the purpose of improving server resource management and optimizing revenue. Arlitt *et al.* [5] grouped sessions based on resource usage in order to discuss scalability issues. Discovery of patterns of users' interactions with Web sites is also a hot research topic in the Web usage mining area [7, 11, 17, 31, 37]. The general goal of Web usage mining is to find usage patterns mainly for implementing personalized services.

Previous studies of session groups in the areas of both session level workload characterization and Web usage mining provide in-depth understanding of the methods used to identify session groups, the characteristics of session groups, and the usage of session groups to analyze related issues. However, it is difficult to compare these approaches since they differ from each other in i) session representations, ii) the clustering algorithms, iii) the functions of Web sites involved in the studies, and iv) the problems discussed.

A particular weakness of previous studies is that while many session attributes have been chosen to represent a session, the relationship among the session grouping results by different session representations have not been explored. One case study shows that a particular session representation worked well for a performance-related problem, while another case study shows that another session representation worked well for another performance-related problem For example, Menascé *et al.* [44] optimized revenue without considering the impact on server resource usage by representing sessions with navigation patterns, while Arlitt *et al.* [5] discuss server scalability without considering revenue by representing sessions with resource usages. The limitation in these approaches is that only one performance-related problem is analyzed at a time. In order to manage an E-commerce server well, one must analyze several related problems in the same context. For example, when optimizing server performance, one should also consider the impact on resource usage. In order to do that, the relationships between different session representations need to be understood.

Another weakness of previous studies is the k-means clustering algorithm used

for session grouping [5, 44]. The key of the algorithm is to select the right centroids for grouping, which is difficult to do since this involves predicting session groups to form. Not enough details with regard to how to select centroids for session grouping were given in previous studies; thus it is not clear whether the centroids were correctly selected. Techniques in selecting right centroids must be develop.

1.1 Motivation and Thesis Goals

The composition of the Web server workload for a Web site is largely influenced by factors such as the functions/contents of the Web site, the technologies used to organize and deliver the contents, and the usage of the Web site. Since these factors change over time, the workload changes and hence there is a need to update the understanding of Web workload.

For E-commerce Web sites, customers are the source of revenue. To serve customers better, one must understand their interests and the patterns in their actions. There is a need to study these interactions between customers and E-commerce Web sites.

What are the characteristics of E-commerce Web workload? How do customers interact with an E-commerce Web site? These two general questions form the basis of this thesis research. Unfortunately, there is not a complete answer to these questions. There are a great number of E-commerce Web sites and they differ from each other in many ways such as in functions/contents and in technologies used. It is impossible to cover all Web sites and obtain a complete picture of E-commerce workload in a single study. Normally, only a few Web site are sampled in a study. The understanding in Web workload characteristics has been built up and updated through the collection of case studies.

The motivation of this thesis is to characterize E-commerce workload and analyze the interactions between between customers and E-commerce Web sites through the use of 3 case studies. The general goal is to provide as much insight as possible based on the available data. In particular, the goals of this study are:

- To report characteristics of E-commerce workload that have significant performance implications with regard to capacity planning and server performance.
- To investigate customers' interactions with E-commerce Web sites by identifying and characterizing session groups, in particular: i) How can session groups be identified? What session representations and clustering algorithms should be used to identify session groups? ii) What are the characteristics of session groups? iii) How do different session representations compare when used for the purpose of clustering sessions?
- To report changes observed in the Web server workload characteristics in recent years, such as changes in the file types, distribution of file size, and popularity of files.

1.2 Thesis Overview

This study analyzes the Web access logs at public Web sites for three organizations: a car rental company, an IT company, and the Computer Science (CS) department at the University of Saskatchewan. The characteristics of Web workloads are explored at various levels to obtain a comprehensive understanding. The performance implications of workload characteristics observed in this thesis are discussed so that the results of this study can be incorporated into practice. The customers' interactions with Web sites are analyzed on the basis of session groups, focusing on Web pages requested, navigation patterns and resource usages.

The first part of the thesis characterizes Web workloads of the Web sites studied at request level, function level, resource level and session level. Web workload characteristics such as file types, distribution of file size, and popularity of files are analyzed and compared with previous studies. In addition, special attention is paid to the characteristics that distinguish E-commerce workload from traditional information Web workload, such as the much heavier use of dynamically generated Web pages and SSL, the mix of request types, and customer sessions.

The second part of the thesis analyzes sessions. Three session representations (Web pages requested, navigation pattern, and resource usage) are used independently, with a hybrid clustering algorithm, to obtain session clusters for a Web site. Session groups are identified from the session clusters and are characterized. For each Web site, session clusters obtained using different session representations are compared in order to discuss the relationships among them.

Observations on the recent E-commerce Web workload include the high percentage of dynamic Web objects, batch referencing of embedded Web objects, non-Zipf-conformity of the popularity of static Web objects, the stable mix of request types in the workload, and the heavy dependence of SSL in E-commerce. A hybrid clustering algorithm, which is a combination of the minimum spanning tree method and k-means clustering algorithm, was proposed to identify session groups. The session groups obtained at each Web site each have their own characteristics, which can be used to improve system performance and resource management. The session clustering results by different session representations is similar enough to be used in identifying session groups.

The remainder of this thesis is organized as follows. Chapter 2 introduces the background for the study. Chapter 3 reviews the related work. Chapter 4 describes the logs used in this thesis and the methodology used in analyzing requests and sessions. Chapter 5 presents the results for workload characterization. Chapter 6 presents the analysis of session groups. Chapter 7 summarizes the conclusions and proposes future work.

Chapter 2

Background

The dramatic growth of E-commerce activities on Web systems has stimulated research interest in studying changes in Web workload and performance-related issues in Web systems. This chapter provides some background into E-commerce, E-commerce Web systems, and the E-commerce activities studied in this thesis.

2.1 Definitions and Terminology

2.1.1 E-commerce

As pointed out by the Organization for Economic Cooperation and Development (OECD) [24], there are many definitions of electronic commerce (E-commerce) made by businesses, researchers, policy-makers, and statisticians, at both national and international levels. The existing definitions meet different needs and differ in three key elements, namely: activities/transactions, communication applications, and communication networks.

According to a 1997 OECD report [23], E-commerce "refers generally to all forms of commercial transactions involving both organizations and individuals, that are based upon the electronic processing and transmission of data, including text, sound and visual images." This is a broad definition which specifies little or no limitation on the three key elements. By this definition, the range of activities/transactions includes most layers of economic activities, covering commerce, finance, services, retail, health, government, education, and many other sectors. This definition does not specify the types of communication applications or communication networks. All communication applications (e.g., the Web, EDI (Electronic Data Interchange),

Minitel, fax, etc.) and all electronic communication networks (e.g., the Internet, intranets, etc.) are included.

2.1.2 Business Transactions

According to the ISO [27], a *business transaction* may involve one or more of the five types of fundamental business activities as follows:

- Business transaction planning: activities that bring sellers and buyers together, such as market research and the production or viewing of advertising/promotions.
- Business transaction identification: the buyer and the seller exchange information about goods, services, and participants. For example, the customer selects the colour of the car to buy, or the seller performs a credit check on the potential buyer.
- Business transaction negotiation: the buyer and the seller negotiate and reach an agreement on the price, terms, and conditions for the actualization of the transaction. In some cases, financial negotiations with third parties such as banks may be required.
- Business transaction actualization: the buyer and the seller formally commit to a transaction. The exchange of payments for goods/services occurs. This stage may also involve a third party such as a financial institution.
- Business transaction post-actualization: activities which take place after the exchange of payment and goods/services are delivered. For example, there could be warranty services or planned upgrades. Activities such as regular maintenance would be considered separate transactions unless specified as part of the original negotiation or contract.

These activities are events which comprise transactions, though not all activities will necessarily result in a completed exchange. All these activities are E-commerce

activities when the data related to the activities is processed and transmitted electronically.

2.1.3 Recent Developments

The OECD 1997 definition of E-commerce is too broad for use in the discussion of some business and government purposes. As a result, businesses, policy-makers, and statisticians have developed more specific definitions for their needs. Limitations are specified in these definitions with respect to the key elements. For example, some consider only committed transactions to be E-commerce activities [13, 26]; some include only the Web and/or EDI as communication applications; some specify the communication network to be computer-mediated or even more specifically the Internet [13, 26].

E-commerce can occur within and between three participant groups – businesses, individuals and government. The inclusion and explicit consideration of this aspect are consistent in most definitions of E-commerce. Most E-commerce activities fit into one of the following categories [25]:

- Business-to-Business (B2B),
- Business-to-Customer (B2C),
- Customer-to-Customer (C2C), and
- Government to Customer (G2C).

According to the OECD [25], in dollar value, B2B E-commerce accounted for 70% to 85% of all electronic sales in 1999. Many B2B transactions are still performed over closed EDI networks, but the trend is to use the Internet. Most B2C, C2C, and G2C E-commerce activities are performed over the Internet and this trend has grown rapidly.

2.1.4 E-commerce Web Systems

2.1.4.1 The 3-Tiered Architecture

The architecture of an E-commerce Web system (Figure 2.1) typically consists of three layers: Web servers, application servers, and database servers [42]. These are the components that are almost always in an organization's control. In many instances, there are also third-party services (e.g., payment services). Large E-commerce systems also contain other components such as load balancers, firewalls, image servers, and proxy servers.

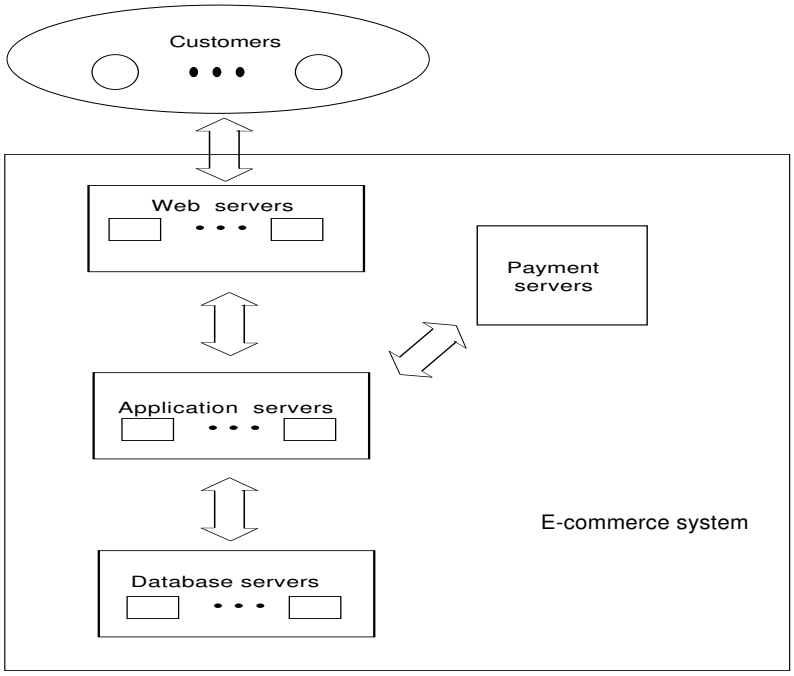


Figure 2.1: The 3-Tiered Architecture for E-commerce Web Systems

Web servers are the front ends of an E-commerce system. A Web server listens for requests coming from client applications over the network and sends responses back to clients. The communication between a Web server and a client is based on the HyperText Transfer Protocol (HTTP). Web servers and client browsers implement HTTP and they communicate by exchanging HTTP messages.

A Web server responds to a client's request with a Web page, which consists of

Web objects. A *Web object* is simply a file addressable by a single URL. A Web page is either static or dynamic. A *static page* is a presentation of static information and its contents are the same for all users. A *dynamic object* is dynamically generated and its contents are usually related to the input information from the client.

Dynamic content can be provided in several ways. Scripting languages are widely used to generate dynamic pages. Scripting languages are different from compiled languages (e.g., C, C++, Java) in that the source code is converted to machine instructions at run time, rather than at compile time. Thus, scripts can be directly embedded within the HTML Web pages and be processed either at the server or at the client. Examples of server-side scripting languages are CGI (Common Gateway Interface), Perl (Practical Extraction and Report Language), JSP (Java-Server Pages), PHP (Hypertext Preprocessor), and ASP (Active Server Pages). Examples of client-side scripting languages include JavaScript and VB Script.

If a client requests a static Web page, the Web server simply takes that page and sends it to the client. If the client clicks on a link that contains dynamic content, however, the Web server will pass the Web page to the interpreter, which is typically on the application server. The interpreter executes the commands in the Web page and returns an HTML document to the Web server that then returns the HTML page to the user.

An application server is the middle tier of an E-commerce system. It is responsible for implementing application logic and interacting with database servers. It queries the back-end database and uses the query result to generate a Web page dynamically. The dynamic page is then propagated to the client via the Web server.

A database server is the back-end of an E-commerce system. It provides data services to the application. Persistent data is managed by mechanisms that guarantee reliability, stability, and availability.

A third-party server brings services provided by independent institutions such as credit card authorization and payment. The payment gateway provides a secure, encrypted connection to transmit payment information between the Web site and any banks involved in the transaction. When an on-line purchase is made, funds are

transferred from the purchaser's credit card account to a merchant bank account. Finally, after the transaction has cleared, funds are transferred from the merchant account to the seller's checking or savings account. Payment may be made off-line and, in this case, the third-party computer system is not electronically connected to the organization's system.

2.1.4.2 Performance-related Issues

Performance Metrics

Traditionally, metrics including response time, throughput, reliability, and availability have been used to measure the performance of a Web system [39]. Response time can be measured at either the server side or the client side. Throughput can be measured in requests/second or transactions/second. These metrics measure the efficiency of a Web system in processing requests or transactions.

In addition to processing efficiency, the efficiency in generating revenue is also important for E-commerce Web systems. The performance of an E-commerce system should also be measured in an economic way. Menascé *et al.* [44] proposed a metric called *revenue throughput*, which is measured in dollars/second. Although the metric has not been widely accepted, the economic performance of an E-commerce system is considered important by organizations. One reason that this has not been widely used is that the data collection mechanism to capture revenue is separate from the Web server data. The logs used in this thesis do not contain any revenue data, so this metric cannot be used.

Customer/User's Experience and System Performance

An E-commerce system must provide customers/users with pleasurable experiences when using the system, in order to maintain that customer group and/or attract new customers [5]. Improving the customer's experience with the system is the key to improving the economic performance of the system, as customers are the source of revenue.

A customer's experience with an E-commerce system is influenced by many fac-

tors, including:

- Security: The customer's privacy and information must be protected.
- Response time: Customers should not have to wait too long for responses from the system.
- Availability: Whenever a customer wants to use the system, it should be in operation and the customer should not be denied access to it.
- Reliability: The system must have the ability to avoid failures, to maintain a reasonable level of performance even when users make mistakes, and to recover data and performance after failure in an acceptable amount of time.
- Functionality: The system should have the goods/services for which a customer is looking.
- Usability: The system should be easy to use; users should easily be able to find what they are looking for.

A system must maintain a high level of system performance in response time, in availability, and in reliability in order to provide customers with a pleasurable shopping experience and achieve high revenue throughput. Short response times and a high degree of availability and reliability are the essential elements of a pleasurable customer experience. These factors are directly related to system performance. Other factors such as security, functionality, and ease of use are also important [34]. Security has a negative impact on performance while functionality and ease of use do not substantially affect performance.

Scalability and Capacity Planning

If the system capacity is inadequate, system performance will degrade, leading to a loss of customers. The goal of capacity planning is to predict the future demand on the system and plan its capacity upgrade to match the anticipated growth in workload. Capacity planning often involves the whole system, including front-end,

middle-ware, and back-end systems. To perform capacity planning, one needs to predict the workload.

Overload Control

The incoming request stream to an E-commerce system is bursty [5, 45, 54]. An E-commerce site may experience heavy workload (several times more than regular volume) on special occasions (e.g., holidays), or due to a successful sale promotion. Overloading is common to E-commerce systems. Server overload control techniques are necessary in order to maintain or optimize system performance. These techniques require knowledge of Web workloads [15, 16, 56].

Personalization

In recent years, E-commerce Web systems have been designed to provide personalized services to customers/users [20, 52, 55]. A personalized service is a service designed to meet an individual customer's interests and needs. For an E-commerce Web site to provide personalized services, the first step is to gather and analyze information about customers. One of the most common techniques for the gathering of information to build a profile of a customer is to track that customer's interactions with the Web site. Once information is gathered, data mining technologies are used to analyze customer's behaviour patterns. Based on the analysis results, recommendations can be made to provide personalized services.

2.2 E-commerce Activities Involved in this Study

In this thesis, the broad OECD E-commerce definition is adopted. However, the data used in this study imposes restrictions with regard to the three key elements of E-commerce. The E-commerce activities being studied herein are limited to E-commerce activities recorded on HTTP logs for the Internet Web servers of organizations' public Web sites. These E-commerce activities are performed over the Internet.

This study only captures a subset of activities from the available logs of the specific organizations that were willing to provide data. Thus, this is a limited snapshot of Web-based E-commerce activities. The main E-commerce activities performed by customers at the Internet Web sites involved in the this thesis are as follows:

- *View-ad*, viewing advertising or promotions (business transaction planning).
- *Browse*, browsing corporation information and goods/services categories (business transaction identification).
- *Search*, searching for information and goods/services (business transaction identification).
- *Select*, selecting goods/services (business transaction identification).
- *Buy*, placing orders/reservations for goods/services, which involves business transaction actualization.
- *Deliver*, arranging the delivery of goods/services, which involves business transaction actualization.

These are the activities observed in the collected HTTP logs. A Web server HTTP log records information about the requests from client Web browsers and the responses by the Web server during the period of time when the log is collected. An HTTP log may not capture all activities supported by a Web site, as the logging duration may not be long enough and there may be many Web servers in a system. Also, an HTTP log is not capable of capturing all user activities, as there is no way of knowing what other off-line activities related to a transaction may have occurred.

A complete record of activities related to an E-commerce transaction or a customer is not available to the analyst; therefore, it is not reliable for the analyst to infer consumers' intentions based on Web server logs. However, although the data in HTTP logs are incomplete, they can still provide some insights into E-commerce Web server workload, customers' interaction with the servers, request patterns as

seen by the Web server, and resource requirements of the system. Such insights can be used in site design, performance modeling, capacity planning, overload control, and the personalization of the Web site.

2.3 Summary

This chapter presents some background on the definition of E-commerce and describes the limitations on E-commerce activities studied in this thesis. This information will be helpful in defining the scope of this thesis. This chapter also introduces the architecture and performance-related issues of an E-commerce Web system.

Chapter 3

Related Work

Workload characterization is the basis for studies on server performance. There are many publications on Web server workload characterization and most address traditional information Web servers. In recent years, there have been some studies focused on E-commerce servers. However, the number of E-commerce workload studies is still relatively small due to the difficulties in obtaining real-world workload traces. This chapter reviews previous studies on Web server workload characterization and studies that identify and characterize customer groups by clustering algorithms. Identifying and characterizing customer groups is an important way to understand customers' interactions with E-commerce Web sites.

3.1 Web Server Workload Characterization

Web server workload can be analyzed at different levels. This section reviews approaches that have been taken in Web workload characterization and the main results obtained in previous studies.

3.1.1 The Approaches

Web server workload characterization can be performed at many levels including request level, function level, session level, and resource level. At the request level, request arrival process, file type, file popularity, file size distribution, and other aggregated workload features are characterized. The request level characterization is very basic to Web server workload characterization.

At the function level, the functions provided at a Web site are analyzed. Analysis at this level is seldom performed on workload of traditional Web servers for information providers, since there were not many functions available. For E-commerce Web sites, however, the functionality is much more important since it affects the customer experience with the site.

At the resource level, the usage of the system resources is analyzed. There are many resources in a system, such as CPUs, memory, disks, caches, I/O, and network bandwidth. The resource usage is closely connected to the performance of the system. If a piece of resource is over-committed, there will be a bottleneck in the system.

At the session level, the client sessions are identified and characterized. A *session* consists of a sequence of requests from the same client during a visit to the Web site. A session is complete if, after receiving a request from a client, the server does not receive any more requests from the same client within a threshold time (this is called the session timeout threshold). The concept of sessions has been used in workload characterization and server performance analysis for Web servers (both E-commerce and non-E-commerce). For example, Kotsis [41] included sessions in a workload characterization model, and Cherkasova [16] introduced the notion of a session in order to implement an admission control mechanism guaranteeing the completion of any accepted session. However, session level workload characterization was not performed in these studies. The purpose of the basic session level workload characterization is to perform statistical analyses of sessions. The session attributes that are characterized include (but are not limited to):

- Session length: the number of requests that are explicitly issued by the client in a session.
- Session duration: the period of time a session lasts, which is the time between the arrival of the first request and the response of the last request.
- Session inter-arrival time: the period of time between the arrival of the previous session and the current session.

- Customer think-time: the time period between the completion of a request which is explicitly issued by the client, and the arrival of the following request.

3.1.2 Previous Studies

Most published studies on workload characterization for Web servers are at the request level [49]. Some characteristics are found common to many sites and are considered important in realizing potential improvements in Web server caching, prefetching, and overall performance. Some of these characteristics include file type, file size distribution, file popularity, request arrival process, and user reference pattern. Arlitt and Williamson [6] performed a relatively comprehensive study by analyzing six Web server access logs collected in 1994 and 1995 from academic environments, scientific research institutions, and commercial Internet providers, within periods ranging from one week to one year. Ten workload invariants, which are observations across all the data set studied, were reported. This study is well known and many results from this study can be considered to be common request level characteristics of Web workloads for traditional information Web servers. The main results are as follows:

- File type: Requests for HTML and image files account for 90% to 100% of the traffic, indicating that the Web sites rarely used dynamic pages.

- File size: The mean transfer size is less than 21 kilobytes, indicating that most files are small.

- File size distribution: The file size distribution is heavy-tailed. The existence of files with a very large size is related to the heavy-tailed property.

- File popularity: There is a concentration of requests on a small percentage of files; 10% of the files accessed account for 90% of server requests and 90% of bytes transferred. Many studies [1, 12, 48] confirmed that the file popularity follows Zipf-like distribution. This property can be informally interpreted to indicate that if the Web documents are ranked by popularity, the number of

references to a document will be inversely proportional to its rank. Thus, the n^{th} most popular document is exactly twice as likely to be requested as the $2n^{th}$ most popular document, indicating a concentration of requests to a few most popular documents. The Zipf-like distribution of Web files indicates that caching even a few documents can significantly improve server performance.

- One-time referencing: Approximately one-third of the files and bytes in the logs are accessed only once.

- Self-similarity: Arlitt and Williamson also discussed the self-similarity of World Wide Web traffic, which had been characterized earlier by Crovella and Bestavros [19]. Self-similarity refers to the fact that the statistical properties of a process are similar over different time scales.

- Inter-reference time: File inter-reference times follow exponential distribution.

These results are based on Web servers for information providers and are based mostly on HTTP logs collected in the early years of Web servers, the period from the early to the mid-1990s. With functional and technical developments in Web servers, workload characteristics change. Some more recent studies on Web workloads, in particular, on workloads to E-commerce servers, provide more updated characterization request, function, resource, and session levels.

Arlitt [3] carried out a session level workload characterization on the 1998 Football World Cup site. The session length, duration, and other factors were discussed. The Web clients were identified by IP addresses. Since session timeout threshold was not clear for the Web site, the characterization was based on assumptions for different session timeout threshold values, ranging from 0 to infinite. When the session timeout threshold was 1,000 seconds, about 10% of the sessions had only one request; over 70% had more than 16 requests and there were some very long sessions (i.e., session length $\geq$ 500); and over 90% of the sessions lasted less than 1,000 seconds. In this study, robots were not discussed since it was believed that most of the traffic was generated by human users.

Oke [47] studied a Web server access log collected in June 1998 from a busy commercial Internet Web site. At the request level, some workload characteristics were very close to those for Web sites of information providers [6], although the Web site was an E-commerce site. For example, most client requests (95.6%) were for either HTML or Image; the distribution of file size was heavy-tailed. A possible explanation is that in their early stages, such as in 1998 when dynamic pages were not yet popular, some E-commerce Web sites were not much different from information Web sites in terms of file types. There are also some workload characteristics which are different from those of the traditional workload. For example, the popularity of files did not conform with the Zipf-like distribution and the inter-reference time for popular files was close to Poisson distribution rather than to exponential distribution.

In Oke's study, at the session level, it was observed that there were both human and non-human clients and that these two types of clients had different behavioural characteristics with regard to, for example, session length and inter-reference time. A non-human client could be a Web proxy or a crawler. It was suggested that different resource management policies should be used for these two types of clients in order to maximize the performance of the Web server. For example, the timeout value for a non-human client session should be set shorter than that for the human client, since non-human clients can make good use of the persistent connections at the server at a very low timeout value. The two types of clients also differed in their phase-transition behaviours. Phase-transition is measured by the changes in the set of files requested in a session, against that for the previous session by the same client. The phase-transition behaviour of a human client is easy to predict while it is difficult to do so for a non-human client. This characteristic can be used to improve the caching or prefetching strategy. Similar to Arlitt [3], clients were identified by IP addresses and session timeout thresholds were not clear, which is a limitation for this study.

Menascé *et al.* [43] analyzed the logs from an online bookstore and an auction site, which were collected during August 1999 and April 2000, respectively. The

analysis was performed at the request, function, and session levels. At the request level, it was reported that the request arrival process was self-similar and that the popularity of search terms followed a Zipf-like distribution, which indicated that the popularity of some dynamic pages followed a Zipf-like distribution since search terms are associated only with requests for dynamic search. At the function level, it was found that 70% of the functions performed in the bookstore site were product selection functions (i.e., browse, search, and view) and the situation was similar in the auction site. The frequency of the function directly related to money spending is very low on both sites. On the same time scale, the request stream for a frequently executed function shows a pattern similar to that for the all requests. This behaviour was not observed for the infrequently requested functions. At the session level, it was found that most sessions were short in terms of both duration and length, which is measured by the number of requests in the session. Close to 90% of the sessions had fewer than 10 requests in length and most sessions lasted less than 1,000 seconds. Interestingly, the distribution of session length is heavy-tailed, especially for the bookstore site which is subject to requests generated by robots. Another observation on the session level is the activity of robots on the bookstore site. At least 16% of requests were generated by robots. The auction site, in comparison, is not used by robots.

Arlitt *et al.* [5] analyzed a five-day period of workload data collected in 2000 from a Web-based shopping system. The workload data included both the Web server log and the application server log, allowing analysis on the CPU usage of the system. The characterization of the Web server workload was at the request, session, and resource levels. At the request level, it was found that almost all requests (95%) were for dynamic resources. In this study, a request is considered to be dynamic if it contains a parameter list. The high proportion of requests for dynamic resources is one of the characteristics for E-commerce workload. It was also found that most popular files do follow a Zipf-like distribution. At the session level, the descriptive statistics for sessions were analyzed. The distributions for session length and inter-request times within a session were presented. A significant percentage of robot

sessions were observed. In this study, a session was identified as a robot session if the session length was longer than 30 requests. The inter-request times for robot sessions were typically shorter than those for human sessions. The resource level of characteristics of this study involves clustering session groups and will be reviewed later.

Vallamsetty *et al.* (2002) [54] characterized a B2C and a B2B E-commerce Web site mainly on the request and resource levels. The request arrival process was self-similar. The file size distribution is not heavy-tailed in nature, which is different than that for traditional Web workloads. It was believed that the lack of large image and video files removes the heavy-tailed nature of the traffic since images were handled with separated image servers and were not included in the analysis. This study also analyzed the processor utilization, disk access, and response times. The workload to the back-end servers was also characterized. The interactions among the workload to the front-end server and the workload to the back-end server were discussed.

These are relatively comprehensive studies on Web workload to E-commerce servers. There are many other recent studies [48, 38] which yielded somewhat similar results.

3.1.3 Summary of Previous Studies on Web Server Workload Characterization

Based on previous studies, some characteristics of E-commerce Web workload are different from those of the workload for traditional information Web servers. For example:

- File type and size: File types have changed significantly. Traditionally, most files are HTML and image files. More requests for other file types, dynamic files in particular, are observed in more recently characterizations. The file sizes have also changed since file sizes are associated with file types.

- File size distribution: The file size distribution is traditionally heavy-tailed.

Many more recent studies, however, do not observe this property. The existence of a small portion of files which are much larger in size than the others were not frequently observed at an E-commerce Web site.

- File popularity: File popularity follows Zipf-like distribution in traditional Web workloads. For E-commerce workloads, this distribution is most observed in requests for dynamic resources, rather than requests for general files [5, 45]. Some studies report non-conformity with Zipf-like distribution.

- Self-similarity: This self-similarity property of Web traffic is still reported in recent characterizations [45, 54].

The characterization of the functions provided at an E-commerce Web site is important in order to understand the workload. Different types of E-commerce Web sites provide different functionalities to their customers. On a high level, however, it seems that some functions such as *search, select*, and *buy* are common for most E-commerce Web sites.

The analysis of resource usage in an E-commerce system is not an easy task. The workload data on application servers and back-end database servers is more difficult to obtain than that on Web servers. There are many resources in an E-commerce system. The analysis of the usage of SSL is important for E-commerce workload characterization since the SSL-related activities are resource demanding.

Session level characterization plays a key role in the workload characterization for E-commerce since E-commerce activities are transactional in nature. Session characterization helps to understand customers' interactions with the Web site. For example, the activities of robots can be analyzed at this level.

3.2 Identifying and Characterizing Session Groups

A session has many attributes and can be represented in many ways. To group sessions, one must select one or more session attributes to represent a session and use an algorithm for clustering. The selection of session representations and clustering

algorithms has been mainly dependent on how the resulting session group would be applied to ensure a successful shopping experience for customers. In previous studies, session groups have been used either for performance analysis or for Web usage mining.

3.2.1 Clustering Sessions for Performance Analysis

Some studies clustered and characterized session groups in order to optimize server resource management, maximize revenue, and analyze the scalability of the system. These studies typically group sessions in a specific way for the purpose of discussing a specific performance problem.

Menascé *et al.* [43] characterized E-commerce workload based on navigation patterns. The action of E-commerce customers is captured using a Customer Behaviour Model Graph (CBMG). A CBMG is basically a first-order Markov chain [2] with states representing what types of services a customer may request. A state can be a single URL or a group of URLs providing similar services and having similar resource demands. Customers navigate from one state to another with measured probabilities.

A CBMG is specific in the sense that it represents only a specific group of users for a specific type of E-commerce. For example, a CBMG for a bookstore is different from that for an auction site since the two businesses provide different services to their customers and are likely to have different customer groups. In the bookstore example, the customer group can be divided into two groups: occasional buyers and heavy buyers. The CBMGs for these two groups are different in terms of the transition probability between states, although the two CBMGs will have the same states and navigation patterns. This characteristic calls for extensive research on CBMGs for different types of E-commerce. However, at this time only a few CBMGs have been published.

The advantage of CBMG characterization is that it allows us to understand the interactions between customers and the Web site and to identify session groups, pro-

viding a way to implement personalized service and a basis for priority-based server resource management [44]. However, CBMGs characterize E-commerce workloads based only on customer activities at the Web site, without considering the impact on server resources.

Using the Markov model to represent a Web user's navigation pattern has been reported in many studies. Chen and Mohapatra [14] obtained a state transition matrix (a form of CBMG) for an online retailer site, without grouping sessions. Some approaches used more complicated Markov models and analyzed the navigation patterns from the data mining points of view [58].

In Menascé *et al.* [43], sessions were grouped by navigation patterns (i.e., CBMG). A session is represented by an $n \times n$ matrix of transition counts between states i and j, $[c_{i,j}]$. The k-means clustering algorithm was applied to group sessions. In this algorithm, i) a session is considered as a point in a virtual space; ii) k points in the space are selected as estimated centroids of the k clusters; and iii) the remaining points are grouped to the cluster with the nearest centroid.

The grouping results are largely dependent on the selected centroids. This study, however, did not report the details regarding how to select the k centroids in the clustering algorithm. It is difficult to choose the right centroids since selecting group centroids requires prediction of groups that may exist. These details are important since, with them, one can evaluate the session grouping process and perform further analysis of session grouping results. The details can also be useful in similar studies of session groups. Due to the lack of this piece of information, it is not clear whether the centroids were correctly selected in this study.

The process of obtaining CBMGs is presented as a process of grouping sessions by navigation patterns. A CBMG obtained in this way represents a group of sessions with similar navigation patterns. However, the process of session grouping can be separated from that of obtaining CBMGs. A CBMG is a state machine to describe a group of sessions. One can derive a CBMG for a session group of any characteristic.

Arlitt *et al.* [5] characterized E-commerce workload based on the level of demand on resources. The motivation was to study the scalability. Requests were classified

into roughly three classes: cacheable, non-cacheable, and search. Each class of request has a distinct characteristic in CPU demand. Cacheable requests require few CPU cycles if there is a cache hit. However, the CPU demand will increase a few hundred times if a cache miss happens. Requests for personalized files are also counted as cache misses in this research.

Non-cacheable and search requests have to be answered with dynamically generated Web pages, and thus have a high CPU demand. Logically, search requests should also belong to the non-cacheable class. However, they were grouped separately since they are distinct from other non-cacheable requests in the demand on CPU.

A session is then represented by a vector of three attributes: (a_1, a_2, a_3), where a_1, a_2 and a_3 are the number of cacheable, noncacheable, and search requests over the total number of requests in the session, respectively. The k-means clustering algorithm was applied to group sessions, resulting in four session groups: heavy cacheable, moderate cacheable, search, and non-cacheable. These session classes are also distinct in CPU demand. The details of selecting centroids for grouping were also not reported in this study.

Based on obtained request and session classes, Arilitt's paper demonstrates that the system scalability is sensitive to the request class mix, request cache hit rate, and the degree of personalization of service. If the percentage of cacheable requests decreases by 5%, the demands on the capacity of application server increase by 17%. If the decrease in the percentage of cacheable requests is combined with the increase in the cache miss rate, the impact is quite significant.

3.2.2 Clustering Sessions for Web Usage Mining

Many studies have been done on mining Web usage in order to understand the user's interaction with a Web site and to discover Web usage patterns [10, 11, 17, 18, 21, 50]. Web usage mining is very relevant to session level workload characterization. One of the main goals of workload characterization is to extract workload properties and

use them to construct a workload model, while the purpose of Web usage mining is to discover Web usage patterns to better serve customers. However, both session level workload characterization and Web usage mining follow similar procedures to process data and use the same clustering techniques to group sessions.

The process of Web usage mining consists of three phases: preprocessing, pattern discovery, and pattern analysis [17, 53]. Preprocessing is necessary in order to convert the collected data (which is in forms such as Web usage logs, user registries and Web site topology) into the data abstractions necessary for pattern discovery. In this phase, data is filtered to remove irrelevant items, and users and sessions are identified.

Algorithms from several research areas such as data mining, machine learning, pattern recognition, and statistics are adopted for Web pattern discovery. Some of the most popular Web pattern discovery techniques are statistical analysis, association rules, clustering, classification, sequential pattern, and dependency modeling. The inputs for pattern discovery are user session files and transaction files; the outputs can be usage statistics, association rules, session clusters, page clusters, or sequential patterns.

The outputs from pattern discovery are analyzed in the pattern analysis phase. Uninteresting patterns and rules are filtered out so that the remaining results can be used for Web site personalization or other uses. The results from Web usage mining can be applied in roughly two ways [40]: one is learning user profile in order to build adaptive (or personalized) servers [28, 46], and the other is learning user navigation patterns for system improvement or site reorganization or modification [32].

In this thesis, the interest is to apply session clustering techniques to discover patterns in workload. The previous clustering approaches in mining Web usage differ from each other in the data abstraction of a user session, in the definition of similarity, and in the clustering algorithm used.

Shahabi *et al.* [51] considered page-viewing time as a primary feature to describe a session and clustered sessions using the k-means clustering algorithm. The error

rate for this method is high. Banerjee [7] improved this method by representing a session with a sequence of pages visited and calculating the similarity between two sequences based on their longest common subsequence and page-viewing time. A graph partitioning method was used to cluster the sessions. It seems reasonable to use page-viewing time as an indication of a user's interest on the page; however, this approach is somewhat application dependent.

Heer and Chi [35, 36, 37] described a method that utilizes multiple modalities of information to group similar user profiles into significant user categories. A user profile represents a significant surfing path which is extracted using the Longest Repeating Subsequence (LRS) method [50]. Thus, a user profile is essentially a path or a sequence of Web pages. A single page is further represented as a multi-modal vector with four modalities: page content, URLs, in-links, and out-links. Each modality is weighted using Term Frequency by Inverse Document Frequency weighting schemes. Each user profile is represented by a multi-modal vector and is clustered using the Wavefront Clustering technique, which is a variant of the k-means clustering algorithm. This technique has been implemented and tested with a real-world Web site; the result shows that it provides accurate profiles of site usage. However, since this method models sessions in a finer degree of granularity, there is a potential scalability problem. Fu *et al.* [29, 30, 31] grouped pages with the same prefix in their URLs to reduce the number of different pages in a session, before applying the clustering algorithm. The problem in that approach is that page categorization should be determined ahead of time manually for best results. Estivill-Castro and Yang [22] pointed out that most clustering algorithms in the literature are difficult to use for grouping sessions by navigation, since the similarity between two navigation paths is a high-dimension problem. This is especially true when more data features are considered to compare two paths. They presented a randomized, iterative algorithm to solve the problem.

Cooley [17, 18] did systematic and in-depth research on Web usage mining, covering the procedures of preprocessing, pattern discovery, and pattern analysis. A Web usage mining framework named webSIFT (Web Site Information Filter System) was

built. The webSIFT integrates the procedures of preprocessing, pattern discovery, and pattern analysis. Many existing pattern discovery techniques are used in this system, including clustering techniques.

Xiao *et al.* [57] proposed a measurement of similarity among sessions based on a chosen session attribute, including page-view, frequency of viewing a page, time spent in viewing a page, or viewing order. An $n \times n$ similarity matrix containing the similarity measurement among all n sessions is then computed. Clustering users with similar interests is performed by permutation of the similarity matrix. This approach is unique in its clustering algorithm, but has a scalability problem.

3.2.3 Two Weaknesses in Previous Studies of Identifying Session Groups

Previous studies in identifying session groups have shown that it is difficult to select an appropriate session attribute to represent a session. A session has many attributes and the importance of a session attribute is application dependent in some cases. It is difficult to compare previous approaches, since the different approaches use different session representations and clustering algorithms. The issue of session representation has not been addressed, which is a weakness of previous studies.

Previous studies on session clustering typically selected a session representation for a specific problem. For example, Menascé [43] characterized the workload based on customers' navigation patterns, ignoring the resource demand. Menascé's system of classification can be used for optimizing the server in terms of revenue without considering the impact on the capacity and scalability of the server and server response time. In comparison, Arlitt [5] characterized the workload session based on demands on resource, ignoring the customer behaviour aspect of the requests. Arlitt's classification system focused on the capacity and scalability of the server and server response time but without considering the revenue management of the server. Obviously, a well-managed server should consider resource management, resource usage, and other issues comprehensively. In order to do that, the relationship among

session groups obtained using different session representations must be understood.

Another weakness of previous studies is with regard to the k-means clustering algorithm used for session grouping [5, 44]. Insufficient details were given in how the k centroids in this algorithm were selected for session grouping. These details are the key to the algorithm since once centroids are selected, session clusters will form around them. It is necessary to know enough about how the k centroids are selected that one can evaluate the session grouping process and perform further analysis of session grouping results. Techniques in selecting right centroids are needed.

3.3 Comparison to Current Study

In this thesis, Web access logs from E-commerce Web sites are characterized. The objective is to enlarge the understanding on recent E-commerce workload at the request, function, resource, and session levels, and to characterize the customer's interaction with a Web site. This study differs from previous studies in several aspects:

- Different types of E-commerce Web sites have different workloads. The difference in functions provided at a Web site results in a difference customer usage pattern, which in turn results in a different workload. For example, it would be rare to have multiple "items" in a shopping cart for a car-rental business, whereas at a bookstore or a hardware store, multiple items are often purchased in the same visit to the site. It is important to characterize workload for different business types. The types of E-commerce studied in previous studies include Web-based shopping (i.e., retail, bookstore, etc.) [5, 45] and auction sites [45]. In this study, Web sites for a car rental company, an IT company, and the CS department of the University of Saskatchewan are involved.

- For request level workload characterization, Web objects (files) are categorized into stand-alone objects and embedded objects. The stand-alone objects are explicitly requested by Web clients; embedded objects are embedded in stand-

alone objects and are requested automatically by Web browsers. In the logs used for this thesis, stand-alone objects are mostly dynamic objects. Dynamic objects are not cached and thus are of particular interest to server resource usage. It is necessary to characterize stand-alone objects and embedded objects separately. This is the approach taken in this thesis.

- At the resource level, the usage of SSL is characterized in this thesis. Vallamsetty *et al.* [54] mentioned that a large proportion of requests come in secure mode, but provided no quantitative characterization.

- At the functional level, the mix of request types is studied in order to obtain a general idea regarding how customers use the site and to predict workload composition. Previous study [45] noticed that the request stream for a frequently executed function shows a pattern similar to that for the all requests. The mix of request types, however, has not been studied in an explicit manner.

- At the session level, the distributions of session length, duration, and session inter-arrival times are analyzed.

In addition to Web workload characterization, an important part of this thesis is to independently identify and characterize session groups using three session representations: Web pages requested, navigation patterns, and resource usages. The relationships among these session representations are analyzed.

Chapter 4

Log Descriptions and Research Methodology

This chapter describes the Web server logs used in this thesis, the procedures used to prepare raw data for analysis, and the methodologies used in analyzing requests and sessions. In section 4.1, the functionalities of the Web sites from which the logs were collected are briefly described. The general characteristics of the logs and the composition of an entry in a log are introduced to provide background information on the raw data. The logs then are filtered to remove errors and information that is not useful for analysis. Section 4.2 describes how requests are analyzed to characterize workload. In section 4.3, session level analysis is presented. Methods for identifying sessions, clustering sessions, and identifying and characterizing session groups are described and discussed.

4.1 Web Server Logs

This section describes the Web sites being studied and the Web server logs used. There are many kinds of E-commerce Web sites and they provide different functions such as auction, rental, retail, and service. For a complete study of E-commerce workload, Web logs from all kinds of sites should be analyzed, as well as logs from application servers and database servers in those respective sites. In reality, it is difficult to collect HTTP logs since companies do not want to release sensitive corporate information. Logs collected from three Web sites are available for this study. The first Web site is that of a multinational car rental company which provides online car rental services. The second is a site for a multinational IT company which sells hardware/software and provides IT services. The third is the Web site for

the Computer Science (CS) department of the University of Saskatchewan, which provides services and information to students, faculty members and other users. By the broad OECD E-commerce definition, these are all E-commerce Web sites. Activities on CS department Web site, such as booking equipment and submitting assignments, also involve authentication, transaction, database interaction, and use of SSL, similar to E-commerce activities at other Web sites.

Table 4.1: Description of the Logs

Log Name	Start Time	End Time	Size (MB)	Number of Entries
Car-Rental-Log	25/Nov/2001 05:59:59	26/Nov/2001 05:59:59	654	2,020,722
IT-Company-Log	01/Sep/2003 01:58:59	01/Sep/2003 23:59:15	209	196,459
Univ-Log-Oct03	01/Oct/2003 00:01:41	01/Nov/2003 00:01:31	480	4,310,682

Web server logs contain information only from Web servers. The activities on application servers and database servers are not available. Thus, this study is limited to activities at Web servers.

4.1.1 Log Descriptions

4.1.1.1 Car-Rental-Log

This Web site is operated by a multinational car rental company. The primary function of the Web site is car rental service. A customer can get a quote by filling in a short form to specify what kind of car he/she wants and where, when, and for how long he/she wants it. When the customer is satisfied the quote does not exceed his/her budget, the customer can then proceed to make a reservation. During the rate-checking process (i.e., getting the quote), the customer may search to select car types and make other decisions. The Web site also provides some other travel-related services, such as weather information and maps.

Access logs were collected from two separate Web servers for the site. Unfortunately, no information on the exact architecture of the Web site was available. Requests from the same user could be received by either server. 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. Therefore, the logs from two Web servers were combined into a log named *Car-Rental-Log*.

The logs were collected on November 25, 2001 and recorded the interactions between customers and the system in one-day period (24 hours). This was the Sunday following the US Thanksgiving holiday, 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. Although the Thanksgiving weekend in the US may have the heaviest travel of any weekend of the year, the log may not reflect the peak activity associated with Thanksgiving since data was collected was at the end of that weekend. The logs contain just over 2 million entries (Table 4.1).

It should also be noticed that because customers make travel plans in advance, the potential dates of activity are in the future. The analysis of the pick-up and return days indicates that most customers plan their car-renting weeks or even months ahead. For example, the number of customers who reserved or checked rates for their car-rental plans reached a peak at Christmas time (Figure 4.1). Intuitively, reserving a car in advance seems to be normal behaviour for most people. Based on that, it does appear that the request arrival pattern shown in the log used for this study is normal for a car rental company.

4.1.1.2 IT-Company-Log

The second log is from a IT company Web site which provides customers with product sales (e.g., hardware and software, retail and wholesale distribution), services (e.g. business solutions, consulting, and training), customer support, and other information. The company has many locations around the world and each branch location has an individual home page linked to the company's general Web site.

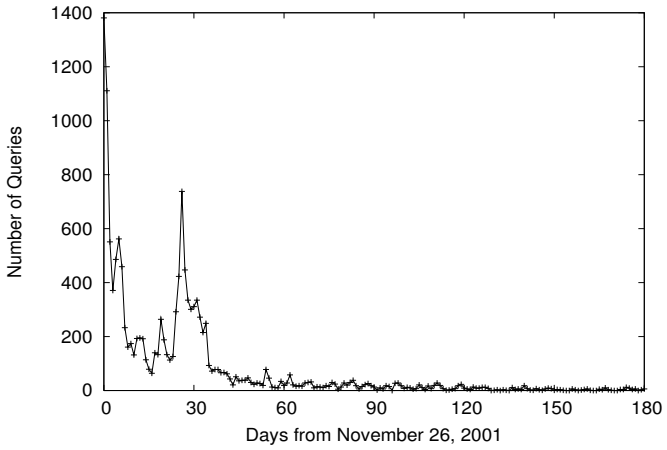


Figure 4.1: The Distribution of Pick-up Days Shown in Car-Rental-Log

The logs obtained for this study were collected at one location in Canada on September 01, 2003. The length of log collection time was 24 hours. No information about the configuration of this Web site is available. The logs were collected from four Web proxy servers. For each proxy server, a set of four logs are available: a proxy log for port 443, a proxy log for port 80, a cache log for port 443, and a cache log for port 80. A total of 16 logs were collected. These logs were combined into one log, which is named *IT-Company-Log*. It contains only 196,459 entries and has a total size of 209 MB (Table 4.1).

4.1.1.3 Univ-Log-Oct03

The final site from which data was collected is the Web site for the CS department of the University of Saskatchewan. This site was initially designed to provide users with information relevant to the department such as course descriptions, course contents, and department news. Over the years, the functions of the site have changed. While still serving as an information-oriented site, the Web site has increasingly provided services to users. Examples of these services are: E-handin (i.e., electronic assignment hand-ins), employment (i.e., student job position applications), and room/equipment booking. These services are provided mainly for faculty members and students. Users have to log on the system to use the services and some

services involve database interaction. The site also hosts hundreds of personal Web pages for faculty members and students, which creates a wide diversity in the contents at the site.

The HTTP log *Univ-Log-Oct03* covers one month of activity, October 2003. The log has 4,310,682 entries and a size of 480 MB (Table 4.1).

4.1.2 Log Formats

The format of an HTTP log depends mainly on the Web server at which the log was collected. A Web server may support several log formats and system administrators can select and customize the format for access logs. The Web server used at the Web site for the car rental company is Microsoft Internet Information Server 5.0 (IIS). IIS can write to multiple logs using standard and extended W3C log formats, which are defined by W3C (World Wide Web Consortium). Car-Rental-Log is in W3C extended log format. An entry in the log has 21 fields (Table 4.2) and represents one request from a browser application. The field cs-username (client user name) is normally turned off in this log for confidentiality purposes.

The Apache Web server is used for the CS department Web site and the Web access log is in NCSA Common Log Formats. An entry in Univ-Log-Oct03 has 9 fields, which are the top 9 fields in Table 4.3. The Apache Web server is also used at the IT company Web site. The Web access logs at the IT company Web site, however, are in NCSA Combined Log Formats, which is an extension of the NCSA Common Log Formats. A log entry in the NCSA Combined Log Formats has 12 fields, with 9 of them exactly the same as those in the NCSA Common Log Formats (Table 4.3).

Some pieces of information are available in all three log formats, although their names may be somewhat different in different log formats. The common fields in all logs are client IP address, time stamp, URL stem, URL query, HTTP version, HTTP method, HTTP status code, and the size of the response message. The common fields in all three formats of logs record information that is essential for

Table 4.2: Fields in the W3C Extended Log Format

Field	**Meaning**
date	Date at which transaction completed
time	Time at which transaction completed
c-ip	Client IP address
cs-username	Client username
s-sitename	Server site name
s-computername	Server computer name
s-ip	Server IP
s-port	Server port number
cs-method	HTTP method
cs-uri-stem	Stem portion alone of URL (omitting query)
cs-uri-query	Query portion alone of URL
sc-status	HTTP status code
sc-win32-status	Server to client: win32-status
sc-bytes	Bytes server sent to client
cs-bytes	Bytes client sent to server
time-taken	Time taken (milliseconds) for the server to reply the request
cs-version	HTTP version used on the client side
cs-host	Host of the client machine
cs(User-Agent)	Browser at client side
cs(Cookie)	Cookie
cs(Referer)	Referer

Web workload characterization.

A cookie takes the form of "KEY = VALUE" and allows sessions to be identified from logs. A cookie is piece of important information for Web workload characterization, but not all logs record cookies. In Car-Rental-Log and IT-Company-Log, most requests contain cookies. Univ-Log-Oct03 does not contain cookies, providing no absolute ways to identify sessions. However, there are other techniques which provide good approximations for session identification. Methods to identify sessions are presented in Section 4.3.

Car-Rental-Log has some pieces of information that are not available in the other two logs; for example, the time taken for the server to respond to a request. Thus, more analysis can be done on Car-Rental-Log since it has more data, as explained in more detail later in this thesis.

Table 4.3: Fields in the NCSA Log Format

Field	**Meaning**
Host	The IP address of the client (remote host) who made the request to the server
rfc931	The identifier used to identify the client making the HTTP request
username	The username (or user ID) used by the client for authentication.
date:time timezone	The time that the server finished processing the request. The format is: [day/month/year:hour:minute:second zone]
HTTP method	The HTTP methods
URL	URL of the resource requested, including the URL stem and URL query
HTTP version	protocol version used by the client
HTTP status code	Codes indicating the success or failure of a HTTP request
server-sent bytes	The size of the object returned to the client by the server, not including the response HTTP headers
referrer	The URL which linked the user to your site
user_agent	The Web browser and platform used by the visitor to your site
cookies	HTTP cookies

4.1.3 Reduced Logs

Some entries in a raw Web server log contain information that is not useful for the purpose of workload characterization. A Web server access log with unwanted entries removed is referred to as a *Reduced Log*. Whether a log entry is useful for workload characterization is determined by its HTTP status code, which indicates the success or failure of a request. Some common HTTP status codes which appeared in the logs are described in Table 4.4. Table 4.5 shows the percentage of requests for each HTTP status code in the logs.

For the discussion in this section, a request with a specific HTTP status code "XXX" is referred as an XXX-request. For example, a request with response code "200" is referred as a 200-request. The treatments of requests with different HTTP status codes are discussed as follows:

- The 200-requests are the successful requests and thus are all kept in the reduced log. The 200-requests are the main part of a log, containing most of the Web usage information on a Web site. The 200-requests make up 73%, 84.9%, and

Table 4.4: Status Codes in HTTP Response Message

Status Code	**Description**
200	Success (the request has succeeded)
206	Partial Content(The server has fulfilled the partial GET request for the resource)
301	Removed Permanently
302	Redirection (The requested resource resides temporarily under a different URL)
304	Not Modified (The client has performed a conditional GET request and access is allowed, but the document has not been modified)
4xx	Client Error
5xx	Server Error

Table 4.5: Breakdown Requests by HTTP Response Codes

Response Code	Percentage of Requests (%)		
	Car-Rental-Log	IT-Company-Log	Univ-Log-Oct03
200	73.0	84.9	86.4
206	0.06	0	0.23
301	0	0	2.18
302	1.7	6.0	0.62
304	25.0	7.4	8.22
4xx5xx	0.18	1.7	2.35
Other	0.06	0	0
Total	100.00	100.00	100.00

86.4% of Car-Rental-Log, IT-Company-Log, and Univ-Log-Oct03 (Table 4.5), respectively.

- A small portion (0.23%) of the requests in Univ-Log-Oct03 are 206-requests. The response code "206" means that the server sent partial contents of the object requested to the client (Table 4.4). To get partial content, a request must have included a range header field indicating the desired range, and may have included an If-Range header field to make the request conditional. It usually takes a series of 206-requests to obtain the complete content of the requested object. It was observed that most 206-request sequences followed a

request with HTTP response code 200 to the same object. When analyzing the popularity of objects, the 206-request sequences are removed since the sequence of requests can only be counted as one reference to the requested object. When analyzing the file transfer sizes, however, all the 206-requests are kept.

The 206-requests are for large static files. A few specific files are related to most of the 206-requests in Univ-Log-Oct03. For example, 53% of the 206-requests are for two files posted on two students' personal Web pages. There are no 206-requests in Car-Rental-Log and IT-Company-Log. The distribution of 206-requests indicates that they were not an important part of the traffic in the logs we analyzed.

- All 301-requests are removed since the code "301" means that the file requested has been removed permanently. A total of 2.18% of requests in Univ-Log-Oct03 are 301-requests, indicating some recent changes made to the Web site. There are no 301-requests in Car-Rental-Log and IT-Company-Log.

- The HTTP status code "302" means that the requested file is under a different URL and redirection has been done. All 302-requests are also removed since the real URL is not available, which is a piece of key information required for workload characterization. All three logs have some 302-requests. The Web site for the IT Company performed a large amountt of redirection (6.0%). Further investigation shows that most of these redirections were requests to log on the secure session. The removal of 302-requests does not affect the workload characterization. In the other two logs, the percentage of 302-requests is much smaller.

- The HTTP status code "304" means "Not Modified." When a client requests an object that 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, the server will answer

the conditional GET request with the status code 304, meaning that the object has not been modified and proxies can send their copies to the client to satisfy the request. The existence of 304-requests indicates the success of caching in the network or client side.

The 304-requests are a large portion of the incoming request stream in all logs used in this thesis. The percentage of 304-requests is as high as 25% for Car-Rental-Log, indicating a high overhead to maintain cache consistency in this case. A possible explanation is that the time-to-live parameters for objects being cached are not set properly. IT-Company-Log and Univ-Log-Oct03 have 7.4% and 8.2% of requests with code "304", respectively.

For file popularity, 304-requests are included since the request for an object does arrive at the Web site. The analysis of file popularity is more accurate with the inclusion of 304-requests. The real popularity of a file, however, is not known since even more requests for popular pages may be satisfied at the caches from which there is no information. For file size and transfer size analysis, however, 304-requests are excluded since they contain no message body. The 304-requests consume very few resources on the server; therefore, they are not important when analyzing the resource usage.

- All 4xx-requests and 5xx-requests are requests with errors and thus are removed. The unsuccessful request rate was as high as 5.3% for Univ-Log-Oct03, and 1.7% for IT-Company-Logs.

After removing unwanted entries, about 98%, 92.3%, and 94.6% of the requests in Car-Rental-Log, IT-Company-Log and Univ-Log-Oct03 (respectively) are kept. The remaining requests are almost all 200-requests and 304-requests.

4.1.4 Filtered Logs

Requests to Web servers are divided into two categories: the first are requests issued explicitly by users for Web pages containing information on goods/services

in which they are interested; the second are the requests issued automatically by Web browsers for Web objects embedded in Web pages. It is important to remove embedded objects in order to study the interactions between users and Web sites. Thereafter, a reduced HTTP log with embedded objects removed is referred to as a *filtered log*.

Image objects (.gif, .jpg, etc.), Javascript objects (.js), and Cascading Style Sheet objects (.css) are normally embedded and thus could be removed without further examination. Requests for Image objects, Javascript objects, and Cascading Style Sheet objects make up about 93%, 84%, and 68% of Car-Rental-Log, IT-Company-Log, and Univ-Log-Oct03, respectively. After removing these requests, most requests for embedded objects are gone.

Caution should be taken in deciding whether a HTML object is embedded. There are only 6 and 36 HTML files in Car-Rental-Log and IT-Company-Log, respectively. The percentages of requests for these files are also relatively small (less than 1.5%). By manual examination, it is clear that they are embedded. However, there are 17,881 HTML files in Univ-Log-Oct03. Requests for these HTML files are close to 15% of all requests in the log. Many HTML files in Univ-Log-Oct03 are personal Web pages, but some are embedded. It is hard to examine every HTML file manually due to their sheer number. It is assumed that all HTML objects are not embedded.

Most requests for dynamic objects are issued explicitly by users since dynamic pages are designed for a Web site to respond to user inputs. However, there are exceptions. There are a few dynamic objects in both Car-Rental-Log and Univ-Log-Oct03 which are embedded objects as well. Requests for these objects are removed. Some JSP pages at the CS department Web site are embedded. Requests for these pages are removed from Univ-Log-Oct03.

After removing embedded objects, 5.5%, 16%, and 31% of the requests in the original logs remained for Car-Rental-Log, IT-Company-Log, and Univ-Log-Oct03, respectively. All the remaining requests in Car-Rental-Log and IT-Company-Log are for dynamic pages. For Univ-Log-Oct03, about half of the remaining requests are for dynamic pages and the other half are for HTML files.

4.1.5 Log Inaccuracies

A significant number of the requests in Car-Rental-Log have values of 0 for log entries for which 0 seems an unlikely value. In particular, 33% of the dynamic requests have 0 for the number of bytes sent from the server to the client (sc-bytes). As well, 10% of the requests have the value 0 for time-taken. Such a large percentage of zero values seems unlikely. All the accesses for static pages have reasonable numbers of bytes transferred, but the dynamic pages exhibit the zero-byte phenomenon. Most of the requests for zero time taken are for small images. This phenomenon may be explained by the following reasons:

1. In some cases, the number of bytes could be listed as 0 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 response 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 in 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.

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, but this has not been confirmed.

It appears that the value recorded for sc-bytes is meaningless for most of the dynamic pages. Thus, it is inappropriate 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 in order to get more realistic

values. Unfortunately, this would also reduce server performance, as more packets 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 does not have any significant impact on the results we present, though it does restrict the scope of the analysis.

About 9.3% and 7.1% of requests have 0 for the number of bytes sent by Web servers in both IT-Company-Log and Univ-Log-Oct03, respectively. Restricted log format and Web server setting information prevents further analysis.

4.2 Analyzing Requests

Based on the requests in a Web server log, analysis of Web workload can be performed in many aspects such as file types, file sizes, file popularity, request arrival process, functions of the Web site, request mix, and the usage of SSL. Requests for static and dynamic Web objects were analyzed together in previous studies [47, 54]. They are analyzed separately in this study because requests for dynamic pages have become the most important part of the Web workload for many E-commerce systems. The caching of dynamic objects is difficult and needs to be handled separately from the caching of static objects. While the methodology is described here, the results for analyzing requests in the three logs are given in Chapter 5.

File Types and Sizes

The files requested in a log are categorized by file name extensions. The characteristics of each file type are analyzed. The percentage of requests for each file type is analyzed and compared with previous studies. The distribution of file size and transfer size are analyzed, in order to see whether they follow the heavy-tailed distribution as reported by previous studies [6, 54].

Popularity of Web Objects

The popularity of a Web object is measured by the number of times that it is requested in the log. Information of Web object popularity is useful in Web site design

and caching. Previous studies have reported that the popularity of Web objects follows a Zipf distribution [1, 4, 5, 6]. The popularity of Web objects in this study are compared with previous studies. Some observations such as objects with close popularity and objects that are requested only once in the log are discussed.

Request Arrival Process

The characteristics of request arrival process are useful in predicting workload volume and managing server resource. The characteristics can be obtained by analyzing the number of requests arriving at the Web server versus time. The association between system response time and the volume of incoming requests is also analyzed to discuss whether the system is stressed.

Categorizing Web Pages by Functionalities

Web pages are the interfaces of an E-commerce Web site. Details of goods/services which an E-commerce Web site provides are presented in Web pages. Customers visit these selected Web pages for desired goods/services and related information. For an E-commerce site, there may be hundreds of Web pages. Web pages may be categorized to reduce the complexity in the analysis of requests from customers. The filtered logs are used in categorizing Web pages, as only the requests that are explicitly issued by customers are considered.

One categorizing method examines the URL of a Web page. If the URL prefixes for two Web pages are the same, they belong to the same category. The categorizing process is easy for a Web site which has a directory structure that matches the functionality of the pages. If the organization of the Web pages on the site has a tree structure, one only needs to decide what level of nodes on the tree structure to pick as the grouping granularity. The categorizing can be done manually with little effort. An automatic classification of pages into categories, however, is possible.

This type of classification of pages is highly dependent on the design of the Web site. Some Web sites do not use, or do not strictly follow, a site category system with a tree structure. For example, different filenames may refer to the same Web

page though the path prefix may be different, as when symbolic links are used. Even for Web sites which use simple tree structures, the categorization may not be permanent since Web sites tend to change from time to time. For these Web sites, the method of merging Web pages is not so straightforward and requires a reasonable amount of effort.

Mix of Requests

The percentage of requests for Web objects that belong to a Web page category, which is defined as the *mix of requests* in this study, provides some details of request composition and information regarding customer activity at the Web site. Such information is useful for server resource management and performance optimization. The explicit study of the mix of requests, however, has not been reported in literature on Web server workload characterization.

To obtain the percentage of requests for Web objects belonging to a Web page category, the logging period is divided into time slots of a specific length. For each time slot, the number of requests for Web objects belonging to a Web page category is compared with the total number of requests.

Requests through SSL

Secure Socket Layer (SSL) communication is an important component of an E-commerce server. A Web system processes requests through SSL in a way different from the processing of requests which are not through SSL. SSL works by using a private key to encrypt data that is transferred over the Internet. Since the data encryption process is resource-demanding, requests through SSL pose a higher demand on system resources. In E-commerce workload characterization, it is important to take into account the SSL usage.

The analysis is performed in two aspects: i) how SSL is used in general, which is indicated by the percentage of requests through SSL; ii) how each Web object is related to SSL. Web objects are categorized by the percentage of requests for this object which are processed through SSL. It is part of future work to analyze the

consequences of different SSL uses for Web objects.

4.3 Analyzing Sessions

As defined in Chapter 3, a session is a sequence of requests from the same user during one logical visit to the Web site. The interactions between a Web server and Web users can be analyzed using sessions. The analysis of sessions includes the statistical characteristics of sessions, the arrival process of sessions, and session groups. A session group is a collection of sessions which have common characteristics. The key to identifying session groups is to define and measure the similarity among sessions. There are two basic steps involved: the first is to select an appropriate representation for a session; the second is to design an algorithm to cluster sessions. The results of session analysis are presented in Chapter 6.

4.3.1 Identifying Sessions

To analyze activities at a Web site, sessions must be identified from the HTTP server logs. The filtered logs are used for the purpose of analyzing a user's interactions with a Web server since embedded objects are not explicitly requested by users. To identify sessions from HTTP logs, the first step is to identify requests from the same user. The second step is to determine which requests belong to the same session, since a user may initiate many sessions during the period when the logs were collected. Two consecutive sessions from the same user are separated by a period of inactivity. If the time difference between the current request and the most recent request of an ongoing session is less than a session timeout threshold, the current request belongs to that session. Otherwise, the request belongs to a new session.

A reliable and efficient way to determine whether requests are from the same user is to use cookies. Requests with the same cookie are from the same user. Typically, the first request from a user does not contain a cookie; the server will then assign a cookie to the request stream from that IP address and the cookie will be used for the

session. Hence, it is difficult to identify the first request for a session. If cookies are not available for an HTTP log, IP addresses can be used to identify users. Requests from the same IP address are assumed to be issued by the same user. This is a rough assumption since an IP address could represent many different users. For example, an IP address may represent a machine that is used by many users, or that may belong to a proxy which represents many, rather than only one specific, Web users.

Some Web access technologies have specific policies which govern session timeout threshold. By default, an ASP session ends if a client does not send a request in 20 minutes[1]. This session timeout threshold can be customized if necessary. The default session timeout for Apache is 30 minutes. The default session timeout setting can be used when there is no evidence of customization, or when the customized timeout value is unknown. If users are identified by cookies and the percentage of distinct cookies is high, most of the sessions will be correctly identified regardless of the correctness of the session timeout threshold.

The warm-up and cool-down effects for an HTTP log must be taken into account to avoid session fragments. The session timeout threshold is used as the warm-up and cool-down times. Sessions that end in the first session timeout threshold time and sessions that begin in the last session timeout threshold time are removed from the identified sessions.

4.3.2 Characterizing Sessions

Workload Traffic in Sessions

The workload traffic measured in the unit of sessions provides information such as the session arrival rate, the distribution of arriving sessions, and the distribution of the number of active sessions. The information is useful in workload modelling and server resource management. The following aspects of sessions are analyzed:

- Number of Session Arrivals: sessions whose first request is received by the server in the chosen time period.

[1] http://www.w3schools.com/asp/asp_sessions.asp

- Number of on-going sessions: sessions that are being processed by the server in the chosen time period.

- Session inter-arrival time: defined in Section 3.1.1.

Session Length and Session Duration

A session has many characteristics that are analyzed in session level workload characterization [3, 43, 47]. Session length and session duration are characteristics which are basic to a session. These are defined in Section 3.1.1.

4.3.3 Selecting Session Attributes to Represent a Session

A session has many attributes (or data features). Some key session attributes are as follows:

- *Pages Requested*: the set of Web pages requested in the session. *Pages Requested* can be used to analyze what a customer requests from the Web site. A session can be represented by the set of *Pages Requested* in the session.

- *Navigation Pattern*: the order in which a customer moves between pages in the Web site. *Navigation Pattern* can be used to observe customer behaviour involved in the shopping experience. A session can be represented by the set of *moves* made in the session. A *move* is a single action of moving from one Web page to another.

- *Resource Usage*: the resources (such as CPU, I/O bandwidth, and memory) consumed in the session. The usage analysis of resource is useful in capacity planning and scalability analysis. The usage of a specific or a combination of resources can be used to represent a session. For example, a session can be represented by the amount of particular resource (e.g., CPU) consumed in the session.

- *Page View*: a single request for a Web page. *Page Views* are often used in online advertising, where advertisers use the number of *Page Views* a site

receives to determine where and how to advertise. A session can be represented by a set of *Page Views* in the session, as opposed to *Pages Requested*.

- *Page Link*: a link that allows users to browse from one Web page to another. *Page Links* can be further categorized into *In-links* and *Out-links*. An *In-link* of a Web page is a link to this page; an *Out-link* of a Web page is a link from this page. The *In-links/Out-links* provide information on all possible browsing directions at the Web site. A session can be represented as a vector of *In-links* and/or *Out-link*.

- *Page Viewing Time*: the time that a customer spends in viewing a page. It can be used as an indicator of how much interest a customer has in that page. A session can be represented by a vector with *Page Viewing Time* on each page requested in the session as elements. One problem with measuring this attribute is that if a user temporarily leaves the browser to do another activity, the *Page Viewing Time* will be inaccurate.

- *Page Contents*: the contents of a page. *Page Contents* can be measured by key words found on the page and can be used to analyze how important a Web page is. A page is represented as a vector of key words and a session is represented by a vector of pages.

All these attributes can be selected to represent a session. Different attributes have been selected to describe a session in previous studies [5, 7, 29, 35, 43, 51].

Three sets of attributes have been selected for the purposes of this study: *Pages Requested*, *Navigation Pattern*, and *Resource Usage*. The attributes *Page View*, *Page Viewing Time*, *Page Links*, and *Page Contents*, were not used in this study since data on these attributes were not available. Sessions represented by the selected attributes can be used to analyze customers' interests, the usage of the Web site, the customers' interactions with the Web site, and resources required to support the Web site. The analysis of these issues is related to the Web server performance (in terms of both revenue and throughput), to server resource management, and to

capacity planning. Data representing these attributes is available from the logs used in this study. Data on *Pages Requested* and *Navigation Pattern* is available in all three logs. The *Resource Usage* can be estimated only for Car-Rental-Log.

Analyzing a customer's interests based on HTTP logs is very speculative since the information in HTTP logs is incomplete. Customers may perform some activities off-line, or may visit more than one Web site at the same time. However, information in HTTP logs is still useful in analyzing E-commerce activities at an E-commerce Web site.

4.3.3.1 Representing a Session by *Pages Requested*

The number of distinct Web pages at a Web site can be in the order of hundreds or even thousands. In order to represent a session by *Pages Requested*, Web pages are categorized to reduce complexity and to attempt to match the desired functionality of the page. The method for merging of Web pages has been described previously in this chapter.

A session is viewed as a point in a coordinate space with each distinct Web page category as a dimension. If there are n Web page categories identified at the Web site, the coordinate space will have n-dimensions. To represent a specific session, the dimensions of the coordinate space are first determined. Each distinct Web page category at the Web site is a dimension for the space. Then, the coordinates for the session can be obtained. If the session requests a Web page category at least one time, its coordinate in this dimension is 1, otherwise; it is 0.

For example, if there are two Web page categories at a Web site, and these are represented by *P1* and *P2*, the dimension vector for the site will be (*P1*, *P2*). If *P1* is visited once and *P2* is visited 3 times in a session, the session can be represented as (1, 1).

4.3.3.2 Representing a Session by *Navigation Pattern*

To capture a customer's navigation pattern, the basic unit of browsing is represented by a *move*. A *move* is the transition between a pair of consecutive requests. A session

can be represented by a set of distinct *move*s and is a point in the space defined with all distinct *moves* as dimensions. If a session makes a *move*, then its coordinate at the corresponding dimension is the number of times that this particular *move* was made. A session's coordinate for a dimension is 0 if it did not make the corresponding *move*.

For example, if there are two Web page categories at a Web site, and these are represented by *P1* and *P2*, the *move* vector will be (Enter → *P1*, Enter → *P2*, *P1* → *P1*, *P1* → *P2*, *P1* → Exit, *P2* → *P1*, *P2* → *P2*, *P2* → Exit). If the route a customer browses in a session is (Enter → *P2* → *P2* → *P2* → *P1* → Exit), the vector representing the session will be (0, 1, 0, 0, 1, 1, 2, 0).

4.3.3.3 Representing a Session by *Resource Usage*

The System Response Time (SRT) for a request is the period from the time when the system receives the request to the time when the system sends the response to the request. The processing of an E-commerce request may involve the E-commerce system, third-party systems, and the networks connecting them. Third-party systems provide third-party services such as payment services and credit card authorization. No requests for third-party services are observed in the logs used in this study; thus the SRT for a request is the response time of the E-commerce system.

According to general queuing rules [39], the response time for a job in a queuing system is equal to the sum of the service time and waiting time for the job. An E-commerce system is a queuing system. The service time for a request is the time that request is receiving service in the E-commerce system. The service time is also referred to as *Resource Time* in this thesis, as it is the time for the request to use the resources of the system. The resources in an E-commerce system are mainly CPUs and disks at Web servers, application servers, and database servers. If these servers are distributed, the time involved in internal network communication between these servers is also included. The waiting time for a request is the time that the request is waiting for its turn to use various resources.

Statistically, the mean response time is equal to the sum of the mean waiting

time and the mean resource time [39], i.e.,

$$E[SRT] = E[WT] + E[RT] \tag{4.1}$$

where $E[SRT]$, $E[WT]$, and $E[RT]$ are the average SRT, waiting time, and resource time, respectively. The resource time for a request depends on how the request is processed in the system, given that the processing power of the system is a constant. The processing power of a system stays the same unless changes in software, hardware, and configuration are made to the system. If two requests ask for the same good/service, their resource time will be close to the same since the system processes them the same way (i.e., same or similar operations at Web servers, application servers, and database servers). $E[RT]$ is, therefore, dependent on the mix of requests arriving at the system. If the mix of requests is approximately the same, $E[RT]$ will be approximately the same.

$E[WT]$ depends on the number of requests in queues and the length of time each request is queued. If there are not many requests in queues, $E[WT]$ is close to 0. If the request arrival rate is low enough so that almost all incoming requests can be served immediately, $E[WT]$ will be close to 0. In this case, according to equation 4.1, $E[SRT]$ is equal to $E[RT]$. $E[SRT]$ will be stable if $E[RT]$ is stable. $E[SRT]$ can be obtained from HTTP logs and $E[WT]$ and $E[RT]$ are unknown. If a stable $E[SRT]$ is observed and the mix of requests is also stable in a log, it can be inferred that $E[WT]$ is close to 0 during the time the log was collected. This is the assumption made in this thesis, though it limits the accuracy of the measurements since $E[WT]$ is not precisely 0.

When $E[WT]$ is negligible, one can approximate the resource time with SRT and represent a session by resource time or *Resource Usage*. An SRT value is first selected to be the resource time for each Web page category. It is assumed that the resource times for requests for the same Web page category are statistically stable since Web pages in the same category are related to goods/services of the same category. The SRTs for all requests for the same Web page category are sorted to

find the median, which is used as the resource time for that Web page category. The median, rather than the average of the SRTs, is used since the average may be skewed because of outliers.

Once the resource times for the Web page categories are determined, a session can be represented by a vector of two elements, (T, t), where T is the total resource time and t is the average resource time per request. The total resource time for a session is the sum of the resource times for all Web page categories requested in the session. Range normalization is applied to the vector to prepare it for its use in the clustering algorithm.

There are several assumptions involved in representing a session by *Resource Usage*. This is a very rough method and the accuracy is hard to evaluate. However, we can at least represent a session by *Resource Usage* and further analysis based on that data can provide some insights into the user's resource usage.

4.3.4 Clustering Algorithm

4.3.4.1 The Data Structure for a Session

A session can be viewed as a point in a virtual space with dimensions of $(x_1, x_2, ..., x_n)$, where $x_1, x_2, ..., x_n$ are the chosen session attributes to represent a session. A one-dimensional array is the natural choice for the data structure to store the coordinates in all dimensions for a session. The length of the array is the same as the number of dimensions so that the coordinates for all dimensions can be stored in the array. This data structure may require extensive computer memory since the number of dimensions can be hundreds or even thousands. This is especially true when using the *Navigation Pattern* attribute. For example, if there are N Web page categories at a Web site, the dimensions will be the N Web page categories when choosing *Pages Requested* to represent a session. The dimensions will be $(N^2 + 2N)$ possible *moves* if the chosen session attribute is *Navigation Pattern*. The number of sessions being clustered may also be large (i.e., in a scale of 10,000 or greater), depending on the number of visitors to the Web site and the duration of the period when HTTP

logs were collected. In addition to the use of much main memory, the computational complexity will be high since the amount of data to process is very large.

If a session is short and the number of dimensions is large, the coordinates in most dimensions are 0. If these 0 entries do not have to be stored in the data structure for a session, the memory required for the data structure will be greatly reduced. The value of a non-zero coordinate and the name of the corresponding dimension can be stored as a value-name pair. A session can be represented as a set (or list) of value-name pairs. In this way, dimensions with 0 coordinates can be ignored and memory space can be saved.

4.3.4.2 Clusters of Sessions

A cluster is defined as a set of sessions that are close to each other within the dimensions of the particular space being considered. The centre of a cluster (hereafter called a **centroid**) is represented by a point whose coordinate at each dimension is calculated by averaging the coordinates at that dimension for all sessions belonging to the cluster. The size of a cluster is defined as the number of sessions that belong to it.

The distance between two clusters is the distance between their centroids. It can be calculated by the Euclidean Distance Formula [39], which is used to calculate the distance between two points in a coordinate space. If, for example, the coordinates for two points in an n dimensional coordinate space are $(x_{i1}, x_{i2}, ..., x_{in})$ and $(x_{j1}, x_{j2}, ..., x_{jn})$, then the distance d between these two points is defined as $d = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}$.

In the implementation of the algorithm proposed in this thesis, a point (i.e., a session) is represented by a set of value-name pairs. The Euclidean Distance Formula is not used directly. The distance between two clusters can be calculated using the algorithm *findDistance*. The core of the algorithm, however, is still the Euclidean Distance Formula (Figure 4.2).

When two clusters are merged into one, the size of the new cluster is the sum of the sizes for the two clusters. The coordinates for centroid of the new cluster are

Table 4.6: Description of Two Clusters

Clusters	A	B
Number of sessions in the cluster	S_a	S_b
List of non-zero dimensions for the centroid	$(x_{a_1}, x_{a_2}, ..., x_{a_{m1}})$	$(x_{b_1}, x_{b_2}, ..., x_{b_{m2}})$
List of non-zero coordinates for the centroid	$(c_{x_{a_1}}, c_{x_{a_2}}, ..., c_{x_{a_{m1}}})$	$(c_{x_{b_1}}, c_{x_{b_2}}, ..., c_{x_{b_{m2}}})$

Algorithm *findDistance*:

Input: cluster A, cluster B (described in Table 4.6)
Output: d, the distance between A and B

(1) Mark all non-zero dimensions in B un-visited, $d \leftarrow 0$ (distance = 0)
(2) For each non-zero dimension in A, x_{a_i}, ($i \leq m1$)
    if there is a non-zero dimension in B, x_{b_j} ($j \leq m2$), which is the same as x_{a_i}
        $d \leftarrow d + \sqrt{(c_{x_{a_i}} - c_{x_{b_j}})^2}$
        mark the non-zero dimension x_{b_j} in B visited
    else
        $d \leftarrow d + \sqrt{(c_{x_{a_i}} - 0)^2}$
(3) For each un-visited non-zero dimension in B, x_{b_i}
        $d \leftarrow d + \sqrt{(c_{x_{b_i}} - 0)^2}$
(4) Return d as the distance between A and B

Figure 4.2: Algorithm *findDistance*

calculated by averaging the coordinates for the centroids of the two merged clusters. The algorithm *mergeCluster* describes how the coordinates for the centroid of the new cluster are calculated (Figure 4.3).

4.3.4.3 The Minimum Spanning Tree Method and k-means Method

To develop the algorithm used in this thesis, two popular clustering techniques, the minimum spanning tree method and the k-means method [39], are studied. The minimum spanning tree method starts with N clusters and then merges clusters with the shortest distance until the desired number of clusters are left. In the case of this study, one problem in using the minimum spanning tree method is the demand on

Algorithm *mergeClusters* (cluster A, cluster B, cluster M):

Input: cluster A, cluster B (described in Table 4.6)
Output: M, the new cluster formed by merging A and B

(1) Mark all non-zero dimensions in B un-visited
(2) For each non-zero dimension in A, x_{a_i}
put x_{a_i} into the list of non-zero dimensions for M
if there is a non-zero dimension in B, x_{b_j}, which is the same as x_{a_i}
$c_{x_{a_i}} \leftarrow (S_a \times c_{x_{a_i}} + S_b \times c_{x_{b_j}})/(S_a + S_b)$
mark the non-zero dimension x_{b_j} in B visited
else
$c_{x_{a_i}} \leftarrow (S_a \times c_{x_{a_i}})/(S_a + S_b)$
(3) For each un-visited non-zero dimension in B, x_{b_i}
put x_{b_i} into the set of non-zero dimensions for M
$c_{x_{b_i}} \leftarrow (S_c \times c_{x_{b_i}})/(S_a + S_b)$
(4) Update the number of sessions in M to be: $S_m = S_a + S_b$
(5) Update the list of non-zero dimensions for M to be $(x_{a_1}, x_{a_2}, ...x_{a_{m1}}, x_{b_i}...x_{b_j})$,
where $i, j \leq m2$
(6) Update the list of non-zero coordinates for M to be
$(c_{x_{a_1}}, c_{x_{a_2}}...c_{x_{a_{m1}}}, c_{x_{b_i}}...c_{x_{b_j}})$,
where $i, j \leq m2$

Figure 4.3: Algorithm *mergeCluster*

main memory and computational power. The computational complexity for this method is $O(N^2)$. N could be quite large, making it difficult to use this method. Another problem encountered using this method is that it tends to result in a very small number of large groups due to susceptibility to the influence of a small number of outliers. It is very difficult to remove these outliers beforehand, and as a result, it is difficult to obtain the desired number of session groups.

The k-means method selects k centroids and then merges clusters to the nearest centroid. It has a computational complexity of $O(N)$ and makes much less demand on main memory than does the minimum spanning tree method. Thus, this method is more suitable when N is large. Previous research has used this method to create session groups [5, 43]. The problem with this algorithm is that it is difficult to

choose the right centroids and the right value for k. The grouping results are largely dependent on the centroids chosen since a selected centroid is supposed to be the centre for a group. The selection of group centroids is difficult since it is somewhat reliant on the prediction of groups that may exist. There are not many details in previous research on the selection of group centroids.

4.3.4.4 The Proposed Hybrid Clustering Algorithm

The clustering algorithm proposed in this research, *groupSession*, combines both the minimum spanning tree method and the k-means method (Figure 4.4) in a two-stage process with manual intervention. First, the minimum spanning tree method is applied to obtain an intermediate result. Based on the result, centroids are manually selected for the k-means method. The k-means method is then applied to the intermediate result to obtain the final session groups.

As is mentioned in Section 4.3.4.3, the computational complexity for the minimum spanning tree method is $O(N^2)$. The initial number of sessions to be clustered, N, may be very large since the algorithm starts with each session as a cluster. Sessions which have the same set of non-zero dimensions belong to the same group since the distance between them is 0. Before applying the minimum spanning tree method, sessions which have the same set of non-zero dimensions are merged to reduce the computational complexity. After these sessions are merged, N are reduced by 20% to 60% in most cases.

The minimum spanning tree method is then applied to the clusters obtained from the previous step to perform the first stage of merging. The purpose of this stage is to identify clusters which can be used as centroids for session grouping in the next stage. This method iteratively merges clusters which are close to each other in the coordinate space being considered. In each round of merging, distances between all clusters are computed and the clusters with the minimum distances are merged. Many rounds of processing are required in order to get a desirable result.

The key in the first stage is to control the number of rounds of merging, R. If R is too small, clusters that can be selected as centroids for the next stage would

Algorithm *groupSession*:

Input: N sessions
Output: data representing k session groups
Let:
$(x_1, x_2, ..., x_n)$ be the completed set of selected dimensions representing a session
$(x_{m_1}, x_{m_2}, ...x_{m_m})$ be the set of non-zero dimensions for a specific session, and
$(c_{x_{m_1}}, c_{x_{m_2}}, ..., c_{x_{m_m}})$ is the non-zero coordinate for a specific session
where:
$m \leq n$
$(x_{m_1}, x_{m_2}, ...x_{m_m}) \subseteq (x_1, x_2, ..., x_n)$
$c_{x_{m_i}}$ is the co-ordinate at the dimension of x_{m_i} (i = 1, 2, ..., m)
(1) Represent each session with: $(x_{m_1}, x_{m_2}, ...x_{m_m})$ and $(c_{x_{m_1}}, c_{x_{m_2}}, ..., c_{x_{m_m}})$
(2) Initiate each session to be a cluster, mark all clusters to be active
(3) For each active cluster i
check all remaining active clusters j
if cluster j has the same set of non-zero dimensions as cluster i
merge cluster j to cluster i and mark it as inactive
(4) Repeat until a set of relatively large clusters appear
From all active clusters
find the pair of active clusters with the smallest distance
merge them to get a new cluster and mark one of them as inactive
(5) Manually select a few biggest clusters as the centroids
(6) For each unattached cluster
group the clusters to the nearest chosen centroids

Figure 4.4: Algorithm *groupSession*

not appear. On the other hand, if R is too large, clusters of different characteristics would be merged due to outlier influence. In either case, the result is not desirable for the purpose of selecting centroids for session groups.

One method for acquiring a desirable result is to start with $R = 1$, then increase R by 1 until R is too large. R becomes too large when session groups of different characteristics are forced to merge by outliers. In this case, very large session groups will appear. In extreme cases, over 90% of sessions will belong to the same group.

By examining the results for each R value in ascending order, one can observe the formation of a number of clusters which are much larger than the others. The sizes and characteristics for the set of large clusters are relatively stable when the

value of R is within a narrow range. If any R value in that range is selected, the result for the next stage will be sufficiently stable.

A group of clusters are selected from the results of the minimum spanning tree method to be candidates for session grouping centroids. The K largest clusters can be selected. It is suggested that the value of K be two or three times the size of the set of large clusters. From empirical experience, in most cases it is sufficient to pick the top 20 to 50 largest clusters to be the candidates. The K value can be larger or smaller, but the set of large clusters observed at the selected R value must be included.

The second stage of the algorithm is to apply the k-means method to the clusters obtained from the first stage. The key issue for this hybrid algorithm is how many centroids for grouping are chosen and how to select them. The number of session groups is the same as the number of centroids selected for k-means clustering algorithm, as a group will form around each centroid. Choosing a right number of session groups is important in order to obtain a desirable result. Having fewer session groups means that some groups are not identified and thus provides no appreciable insight. More groups, on the other hand, means that at least two of the groups have quite similar characteristics. The number of session groups that are chosen is a matter of subjective evaluation, since the actual number of customer groups is unknown from the data obtained.

An iterative selection process is suggested in order to obtain desirable results. List the candidate clusters by sizes in descending order and examine them one by one, starting from the largest cluster. The largest cluster is selected as the first centroid. For the remaining clusters, if the size of a cluster is relatively large and the characteristics of the cluster are different from those that have already been selected as centroids, the cluster will also be selected as a centroid. When *Pages Requested* or *Navigation Pattern* are used to represent a session, the characteristics of a cluster can be evaluated by examining the coordinates of its centroid. If the centroids for two clusters have a similar set of non-zero dimensions, the two clusters have the same characteristics. When *Resource Usage* is used to represent a session,

all clusters have the same set of dimensions. In this case, two clusters are considered to have similar characteristics if the coordinates for their centroids are close to each other.

After centroids are selected, each unattached cluster is grouped to the nearest chosen centroid and session groups are obtained. However, this is only one result in the iterative process, as the choices of centroids may not be right and the session group result may not be desirable. The session group result is not desirable if two groups have similar characteristics, or if a group contains sub-groups of sessions with very different characteristics. Analyze the result if it is not desirable, select different centroids, and repeat the process until a satisfactory result is obtained.

4.3.5 Identifying and Characterizing Session Groups

A session group is a collection of sessions of similar characteristics or behaviour. A set of several session clusters can be obtained by selecting a session representation and applying the clustering algorithm to those sessions. A cluster is supposed to be a session group since the clustering algorithm is designed to organize sessions of similar characteristics into the same cluster. The results of clustering, however, may not be as good as intended since these results are influenced by many factors. In a set of session clusters, some clusters may consist of sessions with similar characteristics while some may contain sessions of different characteristics, depending on how successful the clustering algorithm works. A session cluster is selected to represent a session group only if the majority of the sessions in this cluster have similar characteristics.

Using the same clustering algorithm but selecting different session representations, several sets of session clusters are obtained for a single Web server log. All available sets of session clusters are used in order to identify session groups in a log. Session groups of interests may not be well revealed in a single set of session clusters, since a set of session clusters is only one of the ways of dividing sessions in a log. A session group may be easily identified in one set of clusters, but not so

in another. Even if multiple sets of session clusters are used, there is no guarantee that all session groups can be identified. Using all available session cluster sets is a strategy in achieving the best possible result.

The characteristics for a session group can be revealed by analyzing the following:

- Web pages requested, navigation patterns, and resource usages by the sessions in the group.
- Statistical data of the group such as average session length, session duration, and resource time.
- The distribution of session length and duration.

4.3.6 Comparing Session Representations

Three session attributes, namely *Pages Requested*, *Navigation Pattern*, and *Resource Usage*, are selected to independently represent a session for the purpose of session grouping. The same clustering algorithm is used for all session representations in this study, so session representations can be compared in the same context (i.e., the same data set and the same clustering algorithm). The comparison will provide insights into the consequences of different session representations with respect to identifying session groups. Such a comparison has not been done in previous studies.

The comparison of the two session representations is performed by comparing the session clusters obtained by using each representation independently. If the session cluster sets by two session representation are similar, the two session representations are inter-changeable for the use of identifying session groups. Otherwise, they are not inter-changeable.

Suppose $R1$ is a session representation and $S1$ is the session cluster set obtained using $R1$; $R2$ is another session representation and $S2$ is the session cluster set obtained using $R2$.

An indirect comparison of $R1$ and $R2$ is to check whether similar session groups can be identified from both $S1$ and $S2$. If so, using $R1$ and $R2$ should yield similar

results in session clustering.

A direct comparison of $R1$ and $R2$ is to check whether clusters in $S1$ and $S2$ have a roughly one-to-one correspondence. If, for any cluster Ci in $S1$, one can find a cluster Cj in $S2$ so that Ci and Cj have a high degree of overlap, $S1$ and $S2$ are considered similar. The degree of overlap between two session clusters is measured by the percentage of sessions which are in both session clusters.

4.3.7 Limitation of Session Analysis

A session does not capture all activities of the customer for the duration of the session. Some requests are satisfied by browser caches, Web caches, and Web proxies, and some activities are off-line. A single session should not be used to represent a customer since that session may or may not be typical of the customer. A customer may visit a Web site many times for different purposes during the period when a Web access log is collected, resulting in multiple sessions of different characteristics. A collection of sessions from the same customer would be required in order to study a specific customer's usage of a Web site. From a Web server's point of view, however, activities performed by customers are organized in the unit of sessions. The characterization of sessions is useful in improving the performance of Web servers, which is the ultimate goal of of this study.

4.4 Summary

This chapter first describes the three Web server logs used in this thesis and how the raw data is prepared for analysis. The procedure for raw data treatment is much the same as that used in previous studies. This chapter then presents methods for analyzing requests and sessions. Most ideas behind these methods are not new. However, some aspects such as the method to identify session groups, especially the hybrid clustering algorithm, are developed in this thesis based on previous studies. This method has several advantages and can be used in similar studies for other E-commerce workload analysis.

Chapter 5

Web Workload Characterization

This chapter presents the results of the E-commerce Web workload characterization. Characteristics such as observed file types, file size distribution, the popularity of Web objects, and request arrival processes are analyzed and compared to the results of previous studies. In addition, the request mix and requests through SSL are analyzed, as these also are workload characteristics associated with user behaviour and server resource usage.

5.1 File Category and Characterization

The types of files at a Web site are closely related to the technology used to construct the Web site. Analysis regarding file category and characterization is helpful in understanding the composition of Web workload.

5.1.1 File Category

By file name extensions, most of the files in the logs fall into one of the following categories: Image, HTML, XML, Cascading Style Sheet, JavaScript, Dynamic, Audio, Video, Formatted, Compressed, and Program. This file category system is based on that used in previous studies [6, 47], with addition of three new categories: XML, Cascading Style Sheet, and JavaScript. These new categories are added into the category system because these files types are relatively new and are present in the logs used. The description of these file categories and the rationale to divide these categories are as follows:

- Image: This category includes all formats of image files which appeared in the logs. There are 17 types of image files observed, but over 95% of the image files are in either Graphics Interchange Format (.gif) or Joint Photographic Expert Group (.jpeg, .jpg).

- HTML: This category includes .html, .htm, and .shtml files.

- XML: This category includes all XML-related files (.xml, .xsl, .xls, .dtd, .xsd, .xslt, .sty, and .xlt). XML is a Web technology that has become popular in recent years.

- Cascading Style Sheet: This category includes files with the extension of .css. The Cascading Style Sheet is also a part of HTML technology, but is grouped separately because the number of requests for this file type is high in HTTP logs.

- JavaScript: This category has the file name extension of .js. JavaScript has been widely used to create interactive Web pages. It normally runs on the client side.

- Dynamic: This category includes files used at the server side to create dynamic pages. The execution of these files may involve interaction with back-end servers. Files with extensions such as .jsp, .asp, .cgi, .pl, .php, and .tcl are dynamic.

- Audio: This category includes .mp3, .wav, and .wma files.

- Video: This category includes .avi, .asf, .asx, .rm, .mpeg, .mpg, .wmv, and .mov files.

- Formatted: This category includes files formatted for human reading such as .txt, .dat, .doc, .pdf, .ppt, .ps, .tex, .bib, .bbl, and others

- Program: This category includes source files (.java, .c, .cpp, .sql, .jav, .lisp, and .clp), executable files (.class, .exe, .dll, and .bat), system configuration

files (.profile, .xinitrc, .fvwm2rc, .login, .xdefaults, and .policy), and others.

5.1.2 File Type Characterization

Table 5.1 lists the requests by type of requested Web objects. There are many more files on Univ-Log-Oct03 than appear on the other two logs when considering both the number of file types and the number of distinct files. Except for XML, Audio, and Video, all other file types account for more than 4% each of either requests or bytes. In comparison, IT-Company-Log and Car-Rental-Log contain mainly Image, HTML, Cascading Style Sheet, JavaScript, and Dynamic. Requests for file types such as Audio, Video, Formatted, Compressed, and Program are few, or even absent, in these latter two logs. There are 86,525 unique files identified in Univ-Log-Oct03. In comparison, the number of unique files for IT-Company-Log and Car-Rental-Log are 731 and 1,281, respectively. The large number of unique files in Univ-Log-Oct03 can be explained by the diversity of course contents and the hundreds of personal home pages that the site hosts for faculty members and students. The IT-Company and Car-Rental sites have fewer file types since they are more tightly organized around designated functionalities.

Previous studies [6, 47] showed that Image and HTML files accounted for 90% to 100% of requests. In these logs, the percentage has decreased to 70% to 90%, although these are still the most popular file types for all logs. There are two conflicting factors affecting the percentage of requests for images, particularly for E-commerce-oriented Web sites. On one hand, E-commerce-oriented sites tend to use more small images for quick visual display. The average size for image files in Car-Rental-Log is only slightly larger than 1 kilo-bytes (the median is only 443 bytes). The average transfer size for image files in IT-Company-Log is also small. When the transfer size for images files decreases, more image files tend to be published on a Web page, increasing the percentage of requests for images. About 90% of the requests in Car-Rental-Log are for images. On the other hand, however, static HTML pages are used less and less due to the increased use of dynamic pages, which

Table 5.1: Statistical Analysis on File Types and Transfer Sizes

File Types	# of Requests	% of Requests	% of Bytes	Mean of Size (bytes)	Median Size (bytes)	CoV of Size (bytes)	# of files
Car-Rental-Log							
Images	1789819	90.29	46.34	882	317	2.1	894
Html	19386	0.98	3.18	5584	249	3.0	6
Cascading Style Sheet	42393	2.14	1.62	1302	139	1.1	1
JavaScript	20231	1.02	11.16	18847	32365	0.8	1
Dynamic	109817	5.53	37.35	11546	22963	0.6	374
Formatted	77	0.004	0.001	437	441	0.1	2
Programs	556	0.028	0.34	20735	21902	0.3	3
Total	1982278	100	100	-	-	-	1281
IT-Company-Log							
Images	125063	68.88	19.43	3435	1431	1.5	634
HTML	2774	1.52	3.04	22712	25131	0.3	36
Cascading Style Sheet	115	0.063	0.001	190	316	0.7	2
JavaScript	29196	16.08	7.52	5772	3540	0.9	25
Dynamic	24406	13.44	70.0	57582	46645	1.1	34
Total	181554	100	100	-	-	-	731
Univ-Log-03Oct							
Images	2347057	59.40	38.68	8759	1572	14.1	24797
HTML	583428	14.76	10.56	8781	3270	4.9	43948
XML	1400	0.035	0.018	7067	3419	1.6	598
Cascading Style Sheet	155544	3.94	0.41	1245	1148	0.7	216
JavaScript	185098	4.68	4.57	11875	3788	1.0	104
Dynamic	572095	14.48	7.49	6288	2686	15.7	765
Audio	29	0.0008	0.017	275701	60626	2.1	23
Video	364	0.009	1.09	1444110	284696	2.0	52
Formatted	63871	1.62	24.32	184504	32768	2.8	6352
Compressed	4048	0.10	7.44	922216	43388	3.5	613
Programs	37316	0.94	5.08	66792	2044	10.4	9057
Others	958	0.024	0.29	148068	145188	1.0	-
Total	3951208	100	100	-	-	-	86525

can respond to the input from users. The percentage of requests for dynamic pages ranged from 6% to 15% in all the logs. The rise in the percentage of dynamic pages has changed the relative proportions of images and static HTML objects. In IT-Company-Log and Car-Rental-Log, less than 2% of requests are for HTML objects. In Univ-Log-Oct03, 14.8% of requests are for HTML objects, but one function of this Web site, namely the hosting of personal home pages, does not exist on the other two Web sites.

Cascading Style Sheets are widely used since they can provide a consistent style for all Web pages on a site and can reduce the work required to implement Web pages. In Univ-Log-Oct03, 5.21% of all requests are for Cascading Style Sheet files. JavaScript is also popular since it can run on the client side and create interactive Web pages. In IT-Company-Log, the requests for JavaScript reach 15% of the total number of requests. Cascading Style Sheets and JavaScript have not been characterized in previous research [6, 47]. Logs analyzed in these studies were collected in 1995 [6] and 1998 [47], and these technologies were not popular at that time. The analysis in this study demonstrates the changes in the technologies used to implement a Web site.

Requests for dynamic pages are the most important part of the Web workload in the logs. Dynamic pages are necessary to make a database-driven Web site work since they are used to query databases and return the results to Web users. Most of the important E-commerce activities on a Web site, such as searching, selecting, and ordering goods/services, are related to dynamic pages. Car-Rental-Log shows that dynamic pages accounted for 5.5% of requests and 38% of the bytes transferred; IT-Company-Log records that as high as 15% of requests and 71.7 % of bytes transferred were related to dynamic pages. Although the Web site for the CS department is more information-oriented, it too receives a large percentage of requests for dynamic pages (14.5%). Dynamic pages on this site support services to students and faculty members, such as on-line assignment submission and room/equipment booking. Although the percentage of requests for dynamic pages is 14.5%, only 7.5% of bytes are related to dynamic pages in the Univ-Log-Oct03, which is much less than

the percentages for the other two logs.

Changes in the technologies for building a dynamic page are also observed. CGI and Perl were the most popular technologies for implementing a dynamic Web page in the late 1990s. But in the logs used in this study, Servlet, ASP, and JSP are the most popular file types. Each log used in this study has a dominant dynamic file type. About 98% of dynamic files in Univ-Log-Oct03 are .jsp files; in Car-Rental-Log, all of them are .asp files; and the dynamic files are exclusively Servlets for IT-Company-Log. This variety is not a surprise as there is much technology that supports dynamic pages. It is sufficient for a Web server to use only one of the technologies to fulfill most functions.

There is only a small percentage (1.8%) of requests for Audio, Video, Formatted, and compressed files in Univ-Log-Oct03. The number of bytes related to these requests, however, is 32.9%. Files of these types are usually large.

5.1.3 File Size and Transfer Size Distribution

The sizes of the files that are available at a Web server and the total file sizes that are successfully transferred from the Web server to Web clients are important Web workload characteristics. The file size distribution is helpful in understanding the nature of files at the Web server. The transfer size distribution is a factor that influences Web traffic. When modeling Web workload, the distribution of file size and transfer size must be taken into account.

Previous studies have revealed that the distribution of file transfer sizes follows a Pareto distribution, which has a heavy-tailed property [4, 6, 9, 19]. The Cumulative Distribution Function (CDF) for a Pareto distribution [39] is

$$F(x) = P[X \leq x] = 1 - (k/x)^{\alpha}, \quad for \quad x \geq k. \tag{5.1}$$

In a Log-Log Complementary Distribution (LLCD), a Pareto distribution is indicated by a straight line. The heavy-tailed property of file size distribution implies that a significant percentage of the transfer size is associated with a small set of

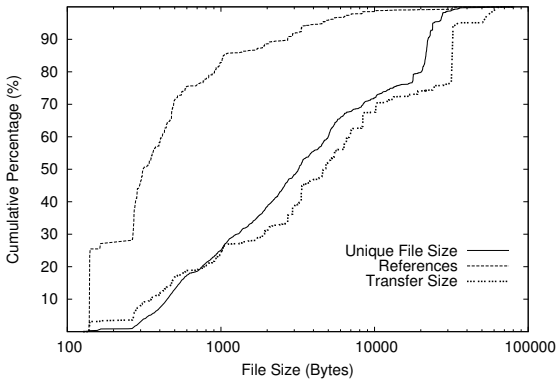
large files.

Figure 5.1 shows the CDFs of file references, unique file size, and transfer size, with file size on the horizontal axis and cumulative percentage on the vertical axis, for all three logs. The distribution of unique file size and file references is helpful in understanding the distribution of file transfer size since the file transfer size is related to the size of files and the reference pattern.

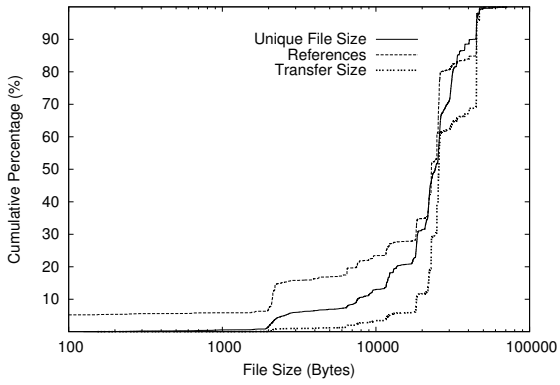
The CDFs for both Car-Rental-Log and IT-Company-Log exhibit no tails (Figure 5.1(a1), (a2), (b1), (b2), as there are almost no requests for large files in the distributions. The largest static file in Car-Rental-Log is less than 100,000 bytes, and the largest dynamic file in that log is less than 80,000 bytes. Over 99% of the static files referenced are less than 10,000 bytes. In IT-Company-Log, the largest static file is less than 60,000 bytes and only a few dynamic files are larger than 400,000 bytes.

In Univ-Log-Oct03, about 5% of the static files are larger than 1,000,000 bytes, accounting for less than 1% of the referenced files. This small set of files, however, is responsible for about 42% of transferred bytes (Figure 5.1(c1)). These large files are mostly Compressed, Formatted, and Video types (Table 5.1). Although the CDF for the file size distribution appears to be heavy-tailed, the LLCD is not a straight line 5.2(a1). Thus the distribution is not heavy-tailed.

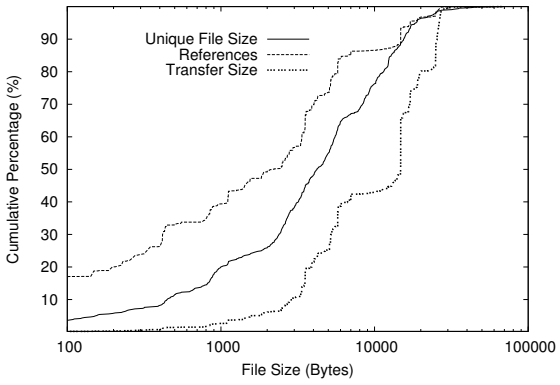
There are also some large dynamic files in Univ-Log-Oct03, which are mostly files transferring students' assignments to markers. Only 0.3% of dynamic files are larger than 1,000,000 bytes, accounting for about 6% of transfer size (Figure 5.1(c2)). A file about course information, with a size of 47,700 bytes, was requested 16,973 times during the logging period. The number of bytes transferred in total for this file accounts for close to 30% of the total transfer size. The LLCD 5.2(a2) shows that the distribution of transfer size for dynamic files in Univ-Log-Oct03 is heavy-tailed.



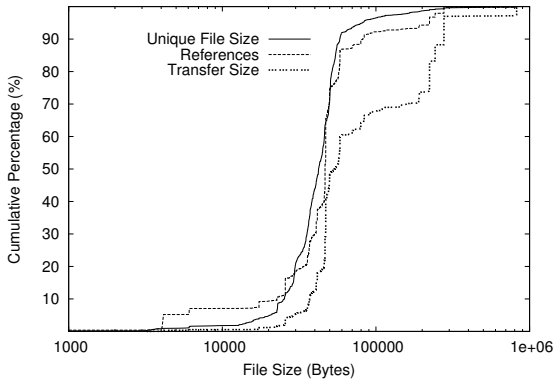
(a1) Car-Rental-Log (Static)



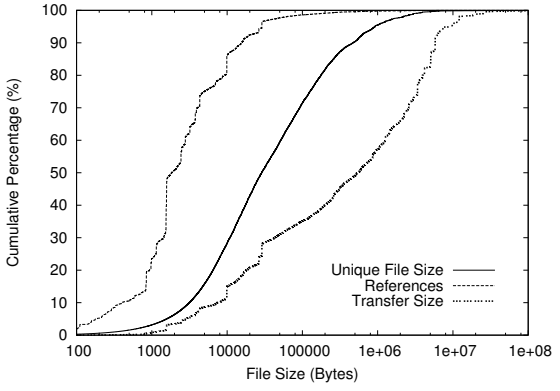
(a2) Car-Rental-Log (Dynamic)



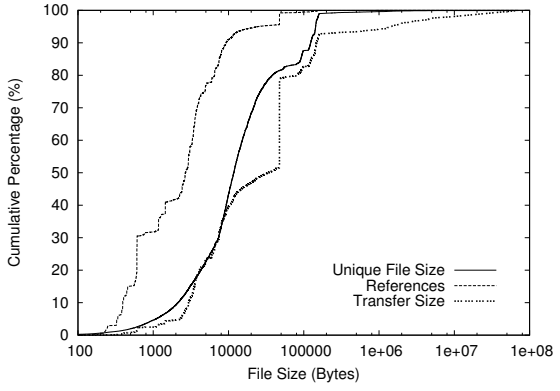
(b1) IT-Company-Log (Static)



(b2) IT-Company-Log (Dynamic)

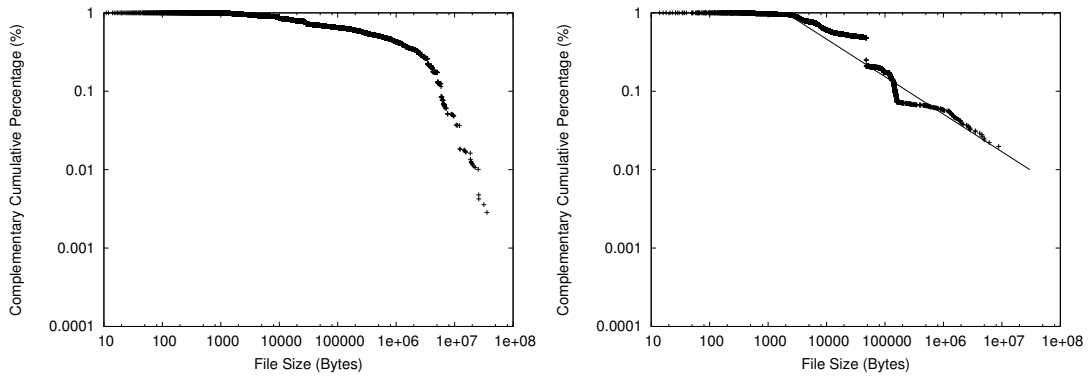


(c1) Univ-Log-Oct03 (Static)



(c2) Univ-Log-Oct03 (Dynamic)

Figure 5.1: CDFs for File Size Distribution



(a1) Univ-Log-Oct03 (Static)

(a2) Univ-Log-Oct03 (Dynamic)

Figure 5.2: LLCDs for File Size Distribution

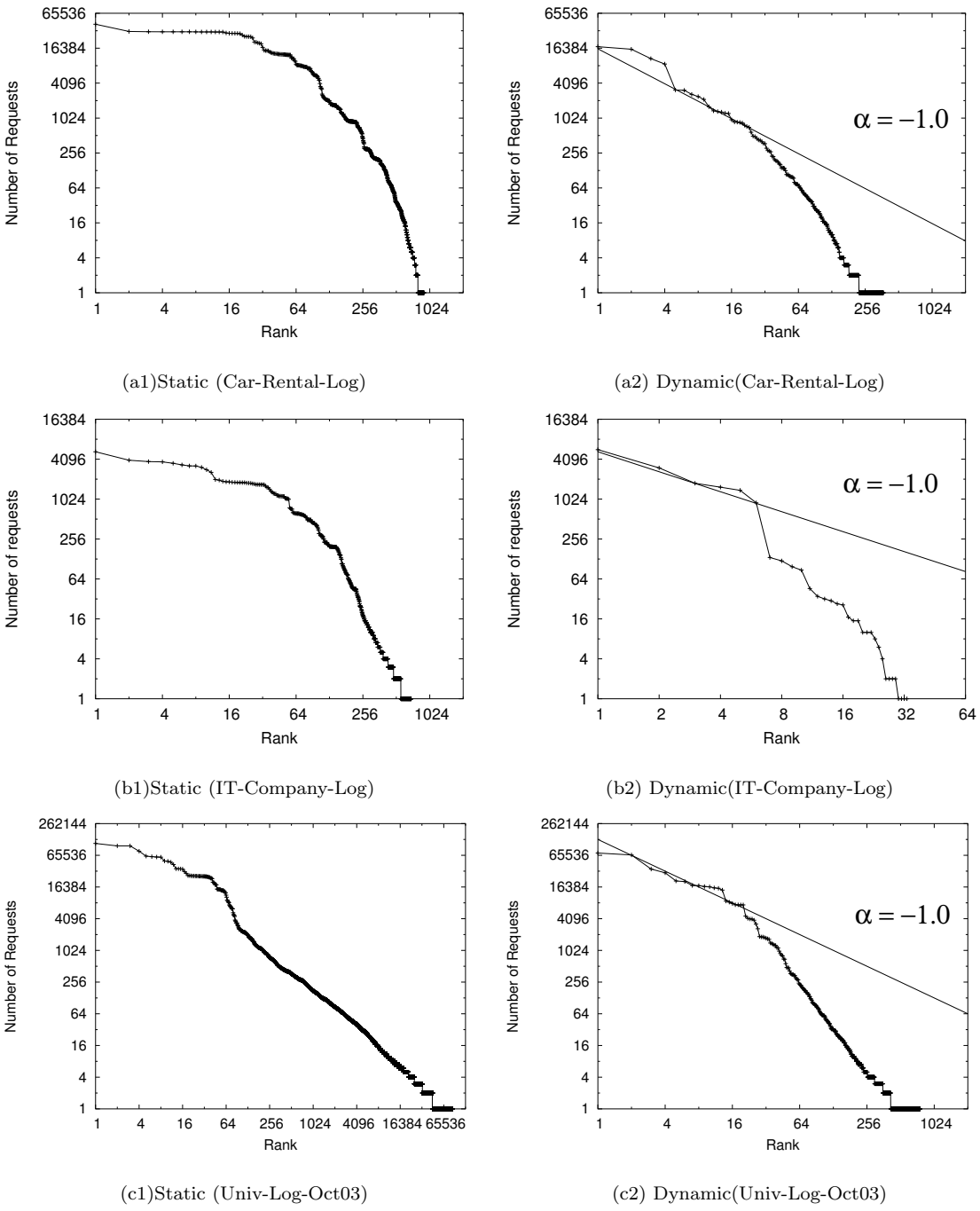
5.2 Popularity of Web Objects

The concentration of requests for a Web object reflects its popularity. Analysis of the popularity of Web objects is useful in improving server performance by caching. Figure 5.3 shows the popularity of Web objects that were requested in the reduced logs. Static objects and dynamic objects are studied separately since the caching mechanism handles these objects differently.

5.2.1 The Popularity of Static Web Objects

Zipf's law [59] is the observation that the frequency of occurrence of events, as a function of the rank when the rank is determined by the above frequency of occurrence, is a power-law function with the exponent, α, close to unity. For a Zipf distribution, when the ranked popularity of objects is graphed against frequency of requests on a log-log scale, the result should be a straight line with a slope of -1.0. If the slope is between -0.5 and -1.0, the distribution is described as Zipf-like.

Previous studies have found that, in general, the popularity of Web objects has followed the Zipf distribution [1, 4, 5, 6]. The results of this section provide some insight into what changes have taken place in the intervening years, since none of the tests can claim that they are representative. This is merely a new snapshot of Web traffic.



(a1)Static (Car-Rental-Log)

(a2) Dynamic(Car-Rental-Log)

(b1)Static (IT-Company-Log)

(b2) Dynamic(IT-Company-Log)

(c1)Static (Univ-Log-Oct03)

(c2) Dynamic(Univ-Log-Oct03)

Figure 5.3: File Popularity

The distributions of the static Web objects in IT-Company-Log and Car-Rental-Log clearly are not Zipf-like (Figure 5.3 (a1), (b1)), since the popularity curves are far from straight lines. The popularity of the static objects in Univ-Log-Oct03 is

also not a Zipf-like distribution (Figure 5.3 (c1)) since the popularity curve on the log-log scale for the top 100 objects is not close to a straight line. The observation of non-Zipf-conformity has also been reported elsewhere [47].

A possible cause for the popularity of static files not showing a Zipf-like distribution is that a portion of requests may have been satisfied by client or proxy caches, reducing requests arriving at the Web server. Further analysis confirmed this conjecture at least for Car-Rental-Log. The log files show that 25.8% of the requests for static files generate a response with the HTTP status code of "304," which indicates that the object being requested is cached somewhere in the network or at the client. It is a safe assumption that many more of these requests are actually generated at the client but are never seen by the server. Another main cause of the non-Zipf-conformity is the batch referencing behaviour, which is discussed in the following section.

5.2.2 Batch-Referenced Objects

Instead of being a straight line, the popularity curve for static objects in all three logs contains many "steps" (Figure 5.3 (a1, b1, c1)). A "step" on the curve is formed by groups of objects that have almost the same popularity. For example, the objects ranked from 3-22th, 24-26th, 42-48th, 49-52th for Car-Rental-Log have almost the same popularity and thus form a step on the curve. A close examination of these objects reveals that they are image files which are actually embedded in the same Web page and thus are always requested together. The same is true for the static objects in IT-Company-Log ranking from 3-5th, 9-11th, and 17-26th. The existence of these "steps" is possibly the main reason why the popularity of the static objects for these logs is not Zipf-like. There are also "steps" on the popularity distribution curve for Univ-Log-Oct03. For example, the objects ranking from 3-4th, 6-9th, 14-16th, and 19-39th all seem to be batch-referenced objects.

A Web page at an E-commerce site typically contains several images that are requested by client browsers individually. These embedded images are typically quite

small in file size, and the overhead involved in logging and setting up connections would be relatively high for such files. Persistent connections available with HTTP 1.1 would alleviate some of this particular problem, but 20% of the requests in Car-Rental-Log used non-persistent HTTP 1.0 at that point in time.

The batch-referenced objects have not been explicitly discussed in previous studies on Web objects. The popularity curves by Oke have "steps" [47], which indicates the possible existence of batch-referenced objects, but this issue was not explored.

5.2.3 One-timers

There are a large number of Web objects that are requested only one time in all three logs. A Web object is referred as a one-timer if it is requested only once during the period when the log is collected. The percentages of requests for one-timers are 12%, 19.3%, and 46.9% in the Car-Rental-Log, IT-Company-Log, and Univ-Log-Oct03, respectively. The high percentage of requests for one-timers is consistent with a previous study [6], which noted that the percentage of requests for one-timers is as high as 33% in a large trace.

The high percentage of one-timers in Univ-Log-Oct03 indicates that the vast number of Web pages at that site are rarely requested. In fact, most of the one-timers at that site are related to personal Web pages, which usually receive many fewer requests than do the general interest Web pages.

5.2.4 The Popularity of Dynamic Web Objects

Figure 5.3 (a2, b2, c2) shows the popularity of dynamic pages for all three logs. The beginning part of the distribution curve is a straight line and has a slope of -1.0 for every log. The popularity for the top 32 objects of Car-Rental-Log, the top 6 objects of IT-Company-Log, and the top 28 objects of Univ-Log-Oct03 follow the Zipf distribution. These top objects account for most of the requests. For example, the top 32 objects of Car-Rental-Log account for 93% of the requests, and this follows the Zipf distribution. Thus, Figure 5.3 (a2, b2, c2) indicates that the most

popular dynamic objects follow Zipf distribution. This is somewhat consistent with Arlitt *et al.* [5], who observed that most popular dynamic objects follow Zipf-like distribution.

5.3 Request Arrival Process

Understanding the patterns and characteristics of incoming request streams to a Web server is necessary for server resource management and for optimizing server performance. If the volume and composition of the incoming workload are estimated with reasonable accuracy, server resources can be arranged in a manner that is able to better serve the requests.

5.3.1 Volume and Composition of Request Arrivals

Figure 5.4 (a1), (b1), and (c1) shows the number of requests for all Web objects versus time for Car-Rental-Log, IT-Company-Log, and Univ-Log-Oct03, respectively. A data point in these figures represents the number of incoming requests in a time slot of 1 minute. Since the logging duration for Univ-Log-Oct03 is one month, only data collected in the first three days are shown in the figure.

For all logs, the traffic was not evenly distributed throughout the day. The number of incoming requests is consistently low in one period and consistently high in another period during the day. For example, in Car-Rental-Log, the average number of requests per minute is about 500 from time slot 0 to 500, in contrast to about 2000 from time slot 700 to 1300. The period when the volume of requests is much higher roughly corresponds to the business hours of the day. Similar distribution has been reported in many studies [6, 47, 49, 54].

Figure 5.4 (a2), (b2), and (c2) shows the number of arriving requests for dynamic Web objects for the three logs. For Car-Rental-Log and IT-Company-Log, the traffic with respect to requests for dynamic Web objects roughly correlated with the traffic with respect to requests for all Web objects. In any selected time slot, if the number of requests for all objects is high, the number of requests for dynamic objects is

also high, and vice versa. This correlation indicates that the ratio of requests for dynamic objects to requests for all objects is relatively stable. This correlation of traffic is largely related to Web site design. At the Web sites where Car-Rental-Log and IT-Company-Log were collected, most static objects are embedded into a Web page which is implemented as a dynamic object. When a user requests a dynamic page, the browser automatically requests all static pages embedded in it. Thus, the ratio of requests for dynamic objects to the requests for static objects tends to be stable.

This kind of correlation is not present in Univ-Log-Oct03. The time period in which the volume of requests for dynamic objects reaches its peak is not the same as that for requests for all objects. At this Web site, many static objects are Web pages by themselves and have no connections with any dynamic objects.

5.3.2 System Response Time versus Request Arrivals

System Response Time (SRT) for a request is the period of time the system takes to respond to the request. A potential area of interest is how SRT is affected by the volume of incoming requests. The analysis was done for only Car-Rental-Log since the other two logs contain no information about SRT.

Figure 5.5 (a) shows the SRT for requests for dynamic pages averaged over a granularity of an one-minute time slot. Over the day, the response times are roughly evenly distributed in the range from about 400 to about 1100 milli-seconds, although there are some outliers and an exception between minutes 200 and 400. The range of response times is from 50 to about 500 milli-seconds in the period between minutes 200 and 400, which may be related to the small number of requests for dynamic pages in the same period (Figure 5.4 (a2)).

The significance of Figure 5.5 (a) is that it shows that the distribution of response times stayed the same even though the number of requests had increased by a factor of 3 in the period between minutes 400 and 1200. The increase in the number of requests has little impact on the average response time. This is more directly shown

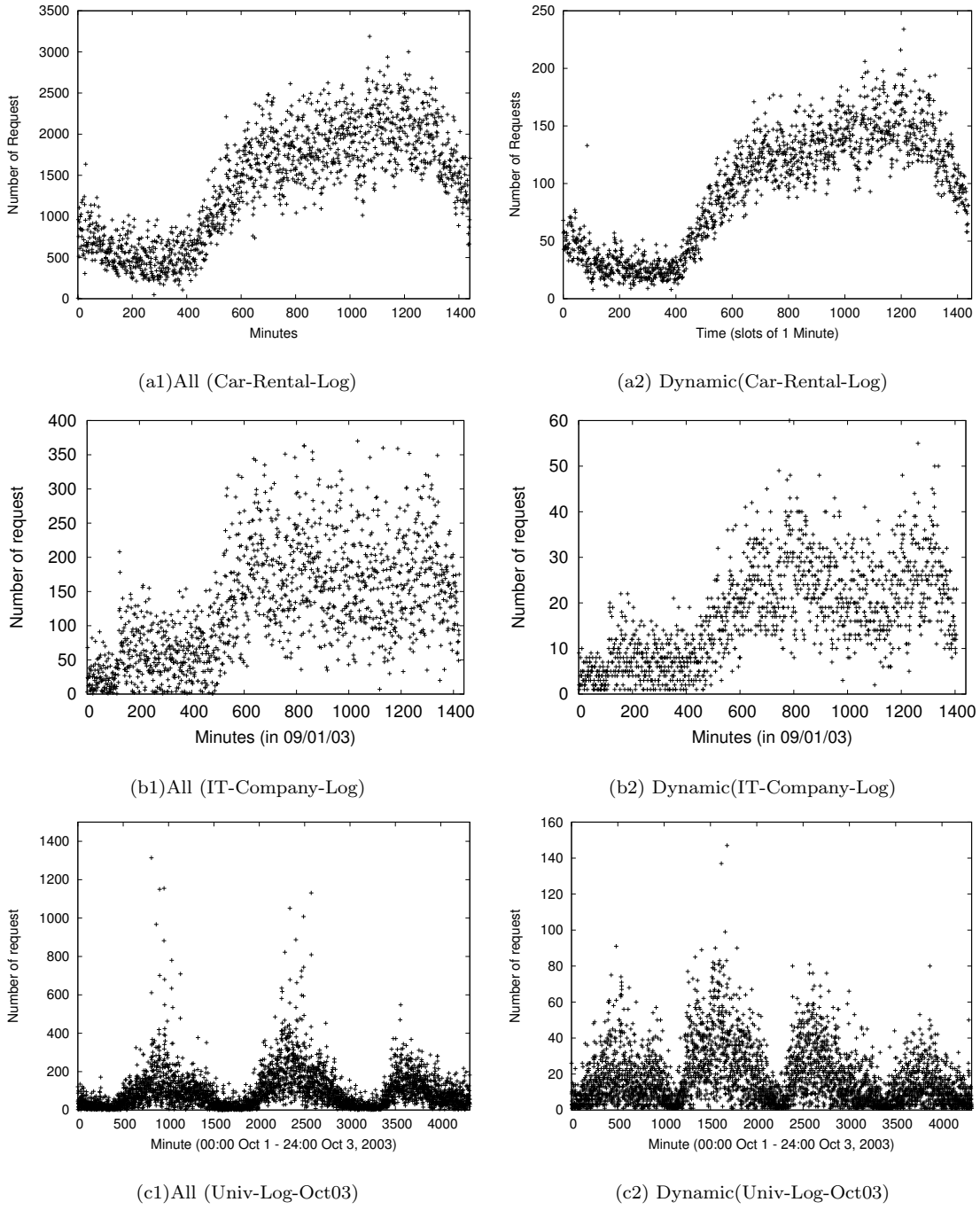
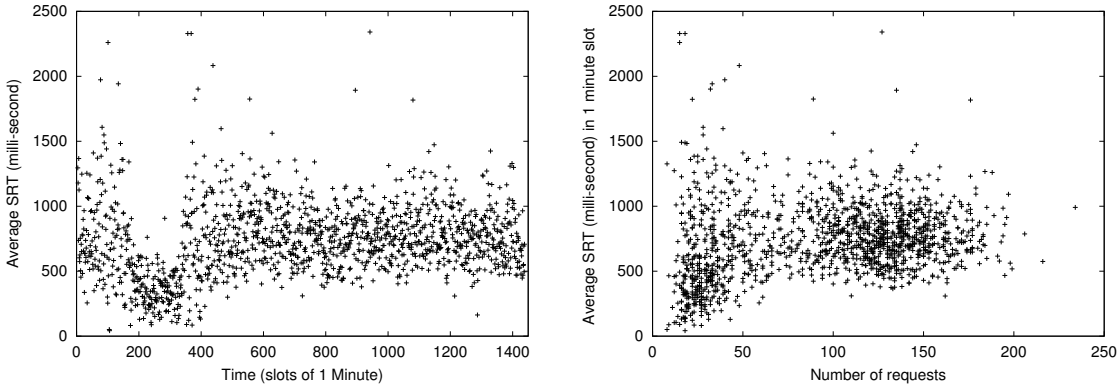


Figure 5.4: Request Arrival

in Figure 5.5 (b), where the range and average of response times stayed almost the same even though the number of requests increased from 50 to about 180 per minute.

Figures 5.5 (a) and (b) indicate 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. If the server had been over-committed, consistently high queuing delay would have resulted in longer SRTs as more requests arrived. As discussed in Chapter 4, in this case, the time for a request to wait to use resources is assumed to be 0, and the resource time for a request is assumed to be the SRT for the request.



(a) Response Time

(b)Request vs. Response Time

Figure 5.5: The Analysis of Response Time (Car-Rental-Log)

Some time intervals have very long average response times, even though very few requests were issued (the points in the top left corner of Figure 5.5(b)). This indicates that the response time likely has more to do with the types of requests and correlated burstiness than with the volume. These requests likely make different uses of the database or other application servers, and this is the main factor causing the variance of response time. 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 to transient queue buildups.

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

5.4 Web Page Categories

Using the method described in Chapter 4, the Web pages requested in the logs are manually placed into categories based on their URLs, in order to reduce complexity. After merging, 17, 8, and 9 Web page categories are obtained for Car-Rental-Log, IT-Company-Log, and Univ-Log-Oct03, respectively (Table 5.3). The types of E-commerce activities, which appear in Table 5.3, are defined in Section 2.2.

Table 5.3 provides some high-level information on what customers want, or more accurately what they do, when visiting the sites. Basically, a Web page category in both Car-Rental-Log and IT-Company-Log matches with a type of E-commerce activity. However, there are exceptions. In Car-Rental-Log, the categories *Express* and *Group* do not match with specific activities, because these categories cover all types of activities performed by specific groups of customers. In IT-Company-Log, the activity related to the category *Account* is not defined in Section 2.2, since the percentage of the activity is small. Table 5.2 shows the percentages of E-commerce activities involved in both logs. The percentage of requests related to the same type of activity can be quite different for different logs. For example, only 3.9% of requests in Car-Rental-Log were related to *View-ad*, while this number was 23.0% for IT-Company-Log.

Table 5.2: The Percentage of E-commerce Activities

Log	*View-ad*	*Browse*	*Search*	*Select*	*Buy*	*Deliver*
	%	%	%	%	%	%
Car-Rental-Log	3.9	27.3	15.6	14.4	37.1	0
IT-Company-Log	23.0	50.7	7.1	0.6	17.2	0.2

Table 5.3 shows that the Web site for the IT Company provides functions that are similar to those of on-line bookstores or of general shopping sites. These general functionalities include browsing, searching and selecting products, and placing orders. The Web site for the car rental company provides another model of customer interaction. In particular, there are natural restrictions on the interactions that oc-

Table 5.3: Web Page Categories

Abbr.	Web Pages	Description	Type of Activities	Number of Requests	Percentage of Requests
Car-Rental-Log					
P	*Promote*	promotions, special offers	*View-ad*	5323	3.9
H	*Home*	home page	*Browse*	17221	12.6
RH	*ResHome*	base page for reservation	*Browse*	23699	17.3
I	*Info*	info, help	*Browse*	7141	5.2
L	*Locations*	available locations	*Browse*	5847	4.3
T	*Travel*	travel information	*Browse*	1842	1.35
VH	*Vehicle*	vehicles to choose	*Browse*	4799	3.50
S	*Search*	pop up search info	*Search*	21420	15.6
CR	*CheckRate*	check rate	*Select*	19659	14.4
CA	*Cancel*	cancel reservation	*Buy*	845	0.62
MK	*MakeRes*	make reservations	*Buy*	819	0.59
MD	*ModRes*	modify reservations	*Buy*	1251	0.91
Q	*Quote*	reservation quote	*Buy*	20430	14.9
V	*View*	view reservations	*Buy*	3794	2.77
E	*Express*	express lane	all activities	741	0.54
G	*Group*	group service	all activities	948	0.69
O	*Other*	others	*Browse*	1101	0.80
IT-Company-Log					
	Promote	promotion	*View-ad*	5611	23.0
	Category	category browsing	*Browse*	10497	43.0
	Search	product search	*Search*	1729	7.1
	Cart	shopping cart related	*Select*	145	0.6
	Order	order related	*Buy*	4190	17.2
	Ship	shipping information	*Deliver*	29	0.2
	Account	account maintain		292	1.2
	Other	other operation	*Browse*	207	7.7
Univ-Log-03oct					
	Class	classes related	*Browse*	306485	30.0
	EHandin	E-Handin system	*Select/Buy*	210572	20.6
	I-help	student self-help system	*Browse*	10499	1.0
	Intranet	internal network	*Select/Buy*	82834	8.1
	Personal	personal web pages	*Browse*	182840	17.9
	Research	research related	*Browse*	45597	4.5
	Resource	resource, information	*Browse*	147630	14.5
	Robots	robot self-identification	*Browse*	6800	0.7
	Other	other operation	*Browse*	26907	2.6

cur. A customer is looking for one item when renting a vehicle, not for multiple items as may be typical when shopping in a bookstore. There are several parameters for that one item, such as car type, rental dates, pickup locations, payment terms, and other options (child seats, late arrival, airline information). In a bookstore, the item has only a few attributes. The only choice would be quantity, hardcover/softcover distinction, and payment/delivery options. There may be more browsing activities in the Web site for the car rental company.

At a higher level of abstraction, customer activities at a car rental site are similar to those at a bookstore. Customers who eventually purchase items from commercial sites all follow the search-select-buy step, though the search may be optional.

Functionalities provided by the university site are clearly different from those for a general shopping site. Most activities in that site are related to class information, assignment submission, and information on other resources. The search-select-buy step is not directly followed in this case since the main function of the Web site is to provide information, help, and services to students and faculty members. However, at a high level, activities at this site can be mapped to the list of E-commerce activities defined in Section 2.2. For example, viewing class information can be considered as the *Browse* activity, while submitting assignments electronically can be considered as a combination of *Select* and *Buy* activities (i.e., students pay for the marking of assignments), The high-level mapping of E-commerce activities is shown in Table 5.3.

Table 5.2 does not show the percentage of activities for Univ-Log-Oct03, since the Web page categories for Univ-Log-Oct03 are too general. For example, it is difficult to separate *Select* and *Buy* activities for the *EHandin* category. The level of granularity, however, is appropriate since there will be hundreds of Web page categories if one lower level of granularity is considered. Based on Table 5.3, A large percentage (71.2%) of activities in Univ-Log-Oct03 are *Browse*.

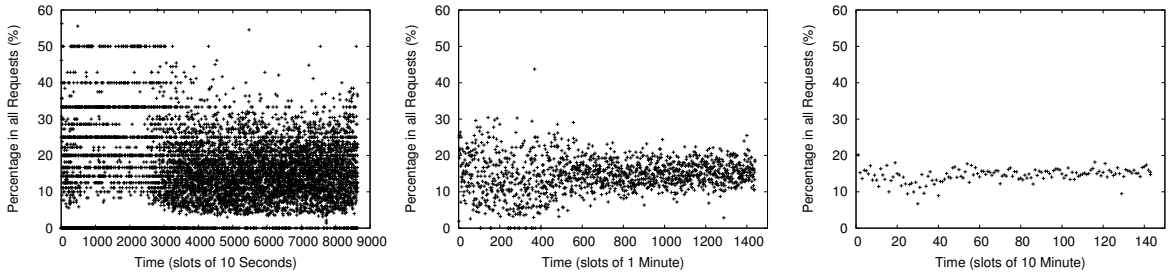
5.5 Mix of Requests by Types

Web objects are classified into Web page categories by their URLs. Thus, requests can be classified by the Web page categories they access. Requests for the same Web page category belong to the same request type. The mix of request types is the relative proportion (or percentage) of request types in the request stream. It provides some details of request composition and information on customers' activities at the Web site. The analysis regarding the mix of request types is useful for server resource management and performance optimization. The study of the mix of request types, however, has not been reported.

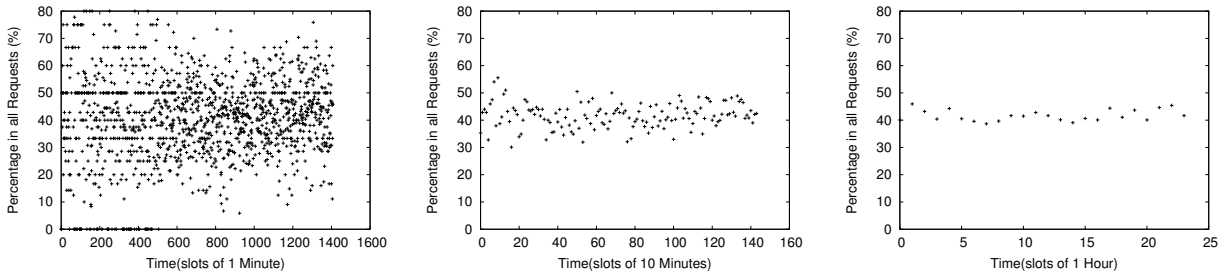
To study the mix of request types, we investigate the percentage of requests for each specific request type in all requests in the workload. Figure 5.6(a) shows this percentage for the request type *Quote*, which represents requests for reservation quotation in Car-Rental-Log (Table 5.3) in the time scales of 10 seconds, 1 minute and 10 minutes. When the time scale is 10 seconds, the percentage ranges between 5% and 30%. The range for the percentages narrows to from 10% to 20% as the time scale increases to 1 minute, and further down to 12% to 16% when the time scales up to 10 minutes. The percentage tends to be relatively stable when the time scale is large enough, which is about 10 minutes in this case. This trend is observed not only for the request type *Quote*, but also for all other request types for Car-Rental-Log.

The same trend is observed for IT-Company-Log. Figure 5.6(b) shows the percentage of requests for the request type *Category*. The range of the percentage narrows when the time scale increases. The time scale for a relatively stable request mix is about 1 hour for this log, which is longer than that for Car-Rental-Log. A relatively stable percentage of requests in all requests is observed for all other request types in this log, at a time scale of 1 hour.

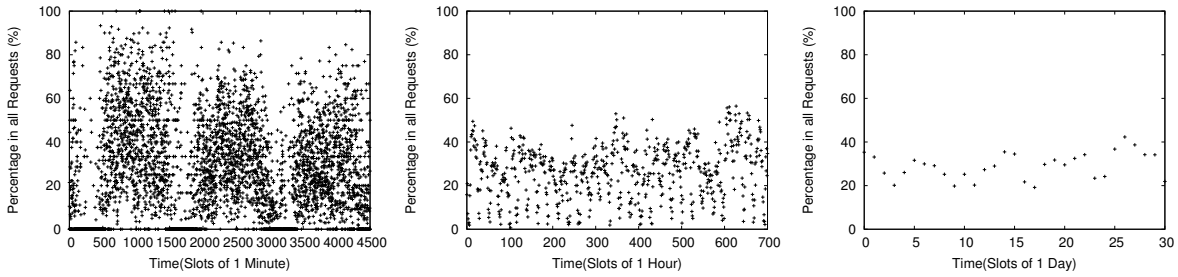
For Univ-Log-Oct03, the percentage of requests for a request type does not become relatively stable until the time scale is up to about a day (Figure 5.6(c)). Such a large time scale is not useful for the purpose of server resource management.



(a)Web Page *Quote* (Car-Rental-Log)



(b) Web Page *Category* (IT-Company-Log)



(c) Web Page *Class* (Univ-Log-Oct03)

Figure 5.6: Percentage of Requests for a Request Type in all Requests

5.6 Requests through Secure Socket Layer (SSL)

This section presents the analysis of the usage of SSL. An entry in Car-Rental-Log contains an HTTP port number, which can be used to identify whether an object is requested through SSL. If the port number is 443, then the request is supposed to transfer through SSL. The entries in IT-Company-Log and Univ-Log-Oct03 do not contain HTTP port numbers. IT-Company-Log, however, consists of logs separated by port numbers. Thus the SSL behaviour can also be analyzed for this log.

The SSL usage for a Web object is evaluated by the percentage of requests for

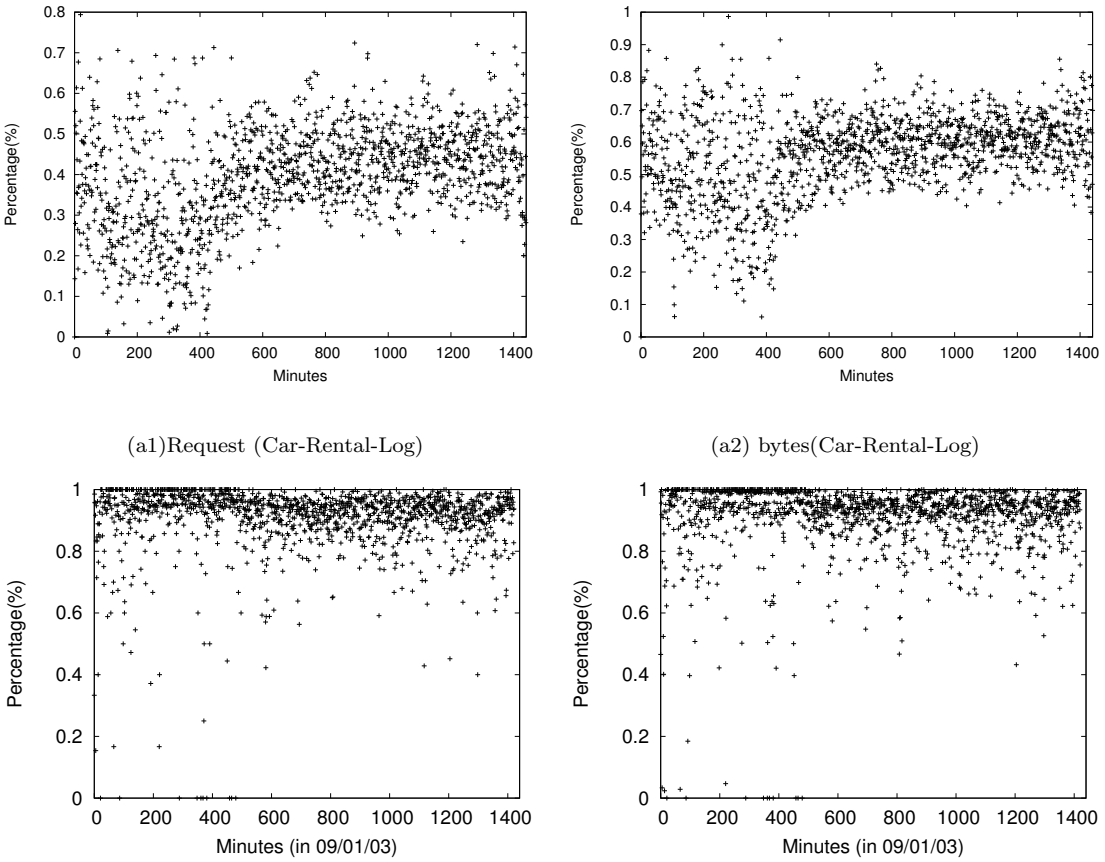
Table 5.4: Partition of Web Objects Based on SSL Usage

Percentage of requests (%) for an object through SSL (%)	0 (non-SSL-objects)	0 < 10 (90%-non-SSL-objects)	10 < 90	90 < 100 (90%-SSL-objects)	100 (SSL-objects)
Partition of Web objects (%) (Car-Rental-Log)	57	18	11	5	9
Partition of Web objects (%) (IT-Company-Log)	10.4	0.5	11.7	22.7	54.7

this object that are processed through SSL. Table 5.4 shows the partition of Web objects based on their SSL usages. 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 requests are not through SSL, then the object is referred to as a 90%-non-SSL-object, etc.

A key observation is that for each particular object, its requests have a strong probability either to use SSL for the primary access method or to not use SSL access at all (Table 5.4). Very few objects show similar frequencies of SSL and non-SSL requests. Of all objects in Car-Rental-Log (including embedded objects), only 9% are SSL-objects and 5% are 90%-SSL-objects, while 57% are non-SSL-objects and 18% are 90%-non-SSL objects. For this log, only a small percentage of Web objects are SSL-oriented. IT-Company-Log shows different partitions of Web objects based on SSL usage. About three-quarters of Web objects are either SSL-objects or 90%-SSL-objects, while non-SSL-objects accounted for only about 10% of all objects.

Figure 5.7 (a1) and (a2) shows that, on average, over 40% of requests and 50% of bytes in Car-Rental-Log are processed through SSL. Although only 9% of Web objects in the log are SSL-objects, these objects are mostly key components for supporting the server's functions and thus the relative frequency of access to these objects is high. Over 90% of requests and over 90% of bytes in IT-Company-Log are processed through SSL (Figure 5.7 (b1) and (b2)). In both cases, the use of SSL is an important workload characteristic.



(a1)Request (Car-Rental-Log)

(a2) bytes(Car-Rental-Log)

(b1)Request (IT-Company-Log)

(b2) bytes (IT-Company-Log)

Figure 5.7: Percentage of Requests through SSL

It is also observed that requests for 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 encrypt requires a fine level of granularity. In the case where images that do not need protection from unauthorized viewing are embedded in a Web page containing data that needs SSL protection, it is an option to present the entire page via SSL. In this way the system does not have to switch back and forth between SSL and non-SSL use, although there is overhead in encrypting these images.

Another option is to encrypt only data which needs security protection, but there is also overhead since both browser and server need to switch back and forth between

SSL and non-SSL use. The quantitative analysis of the consequences of using SSL or non-SSL was not done in this thesis due to insufficient information. This analysis is part of future work.

5.7 Summary

This chapter has presented the characteristics of the Web server workload for Car-Rental-Log, IT-Company-Log, and Univ-Log-Oct03. The results show that Web server workload has changed over time due to changes in the goods/services a Web site provides and due to new technologies used to implement the Web site. The structure of a Web site has become more complicated. The workload characteristics in file types, file size distribution, file popularity, the mix of request types, and the usage of SSL are observed in the logs used in this study. These characteristics have performance implications for E-commerce servers which will be useful in improving server performance, managing server resource, and performing capacity planning. A summary of workload characteristics and their performance implications are as follows:

- CSS and JavaScript files have become widely used. The percentages of requests for these types of files range from 3% to 16% in all logs.

- Almost all important functions of the Web sites where the logs were collected were implemented using dynamic files. The percentages of requests for dynamic files range from 5.5% to 15% in all logs. Due to the large numbers of requests for dynamic files, the characterization of dynamic files should be separated from that for static files.

- In spite of the effectiveness of client or proxy caches, a large percentage (60% to 90%) of incoming requests are still for images. The performance implication is that the server cache is still very necessary, in addition to effective client and proxy caches.

- Embedded images in the same Web page tend to be requested together. If the Web page is a popular one, such as the home page for a site, the amount of workload and overhead generated by requesting these images is high. The batched-referenced images in a popular Web page should be cached as a bundle so that a client needs only one request to receive all embedded images. Thus, the server can send all these images in one reply. In order to do this, the caching mechanism must have knowledge regarding the objects that need to be handled together.

- In general, the popularity of dynamic Web objects follows a Zipf-like distribution, indicating that caching would be beneficial for system performance.

- The incoming request stream is bursty. Knowing the composition of the incoming request stream is helpful for organizing the server resource. For example, at the peak of the requests for dynamic pages, the back-end database would receive the highest service demand. More resources should be assigned to it so that it does not become a bottleneck.

- The request mix is relatively stable when the time scale to measure it gets large enough. This stability indicates that customers are looking for similar services throughout the day. The request mix can be taken into account when allocating server resources to optimize performance. For example, jobs can be scheduled in such a way that each request type gets its share of resource based on request mix. The stable request mix is also useful in forecasting workload. For example, assuming sales will increase by 50%, the volume of a specific request type can be predicted based on request mix.

- For E-commerce servers, most pages related to revenue generation are requested through SSL. To optimize server resource management with respect to revenue-generating page requests, priority should be granted to requests using SSL.

- It is observed that some images were processed through SSL to save the cost

of moving back and forth between SSL and non-SSL use. This practice should be careful evaluated since the cost of SSL processing is high. The performance effect has not been quantified because the detailed data is not available.

Chapter 6

Session Group Identification and Characterization

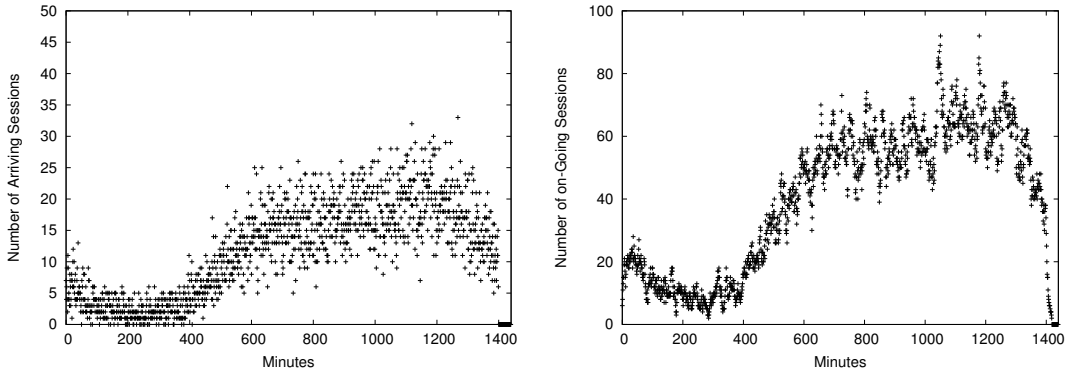
This chapter presents the results of analyzing the collected Web traces with respect to session behaviour. Sessions were identified and characterized. Session groups were identified from the session clusters obtained using the hybrid clustering algorithm proposed in this study. Three session attributes were selected to represent session characteristics: *Pages Requested*, *Navigation Pattern*, and *Resource Usage*. The composition of the session groups and their salient features are compared and analyzed.

6.1 Session Arrivals and Session Characteristics

In this section, the numbers of arriving sessions and concurrent sessions, the distribution of session inter-arrival times, session length, and session duration are analyzed. This analysis is useful for server resource management and workload modelling.

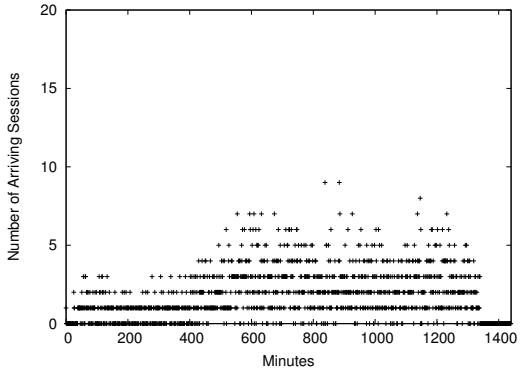
6.1.1 Identifying Sessions

Sessions are identified using the methodology described in Chapter 4. Requests in Car-Rental-Log and IT-Company-Log have cookies, but this information is not available for requests in Univ-Log-Oct03. Thus, sessions in Car-Rental-Log and IT-Company-Log are identified by cookies and sessions in Univ-Log-Oct03 are identified based on IP addresses. There are 16,511, 2476, and 139,680 sessions identified in Car-Rental-Log, IT-Company-Log, and Univ-Log-Oct03, respectively.

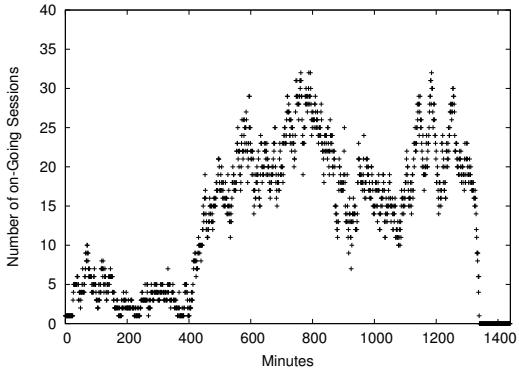


(a1)Arriving Sessions (Car-Rental-Log)

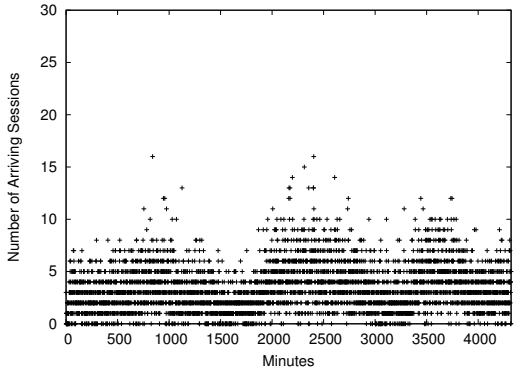
(a2) Concurrent Sessions (Car-Rental-Log)



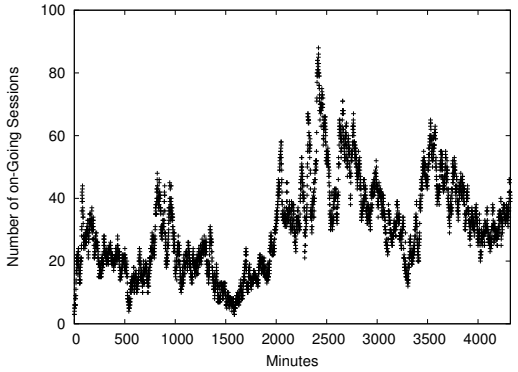
(b1)Arriving Sessions(IT-Company-Log)



(b2) Concurrent Sessions (IT-Company-Log)



(c1)Arriving Sessions (Univ-Log-OCT03)



(c2) Concurrent Sessions (Univ-Log-OCT03)

Figure 6.1: Session Arrival Rates

6.1.2 Session Arrivals

The distributions of the number of session arriving at a Web server and the number of concurrent sessions the Web server is serving are useful for server resource management. This information can also be used in building workload models. Figure

6.1 shows the number of arriving sessions and concurrent sessions per minute. The start time in GMT were 05:59:59-25/Nov/2001, 01:58:59-01/Sep/2003, and 00:01:41-01/Oct/2003, for Car-Rental-Log, IT-Company-Log, and Univ-Log-Oct03, respectively. Only the data collected in the first three days are shown for Univ-Log-Oct03, which has a logging duration of one month. In any minute, the numbers of arriving sessions are smaller than 35, 10, and 18 for Car-Rental-Log, IT-Company-Log, and Univ-Log-Oct03, respectively. The number of concurrent sessions are fewer than 100 for both Car-Rental-Log and Univ-Log-Oct03, and fewer than 35 for IT-Company-Log.

In general, the distributions of arriving sessions and concurrent sessions have patterns similar to those of distributions of arriving requests, as presented in Figure 5.4. With a lower request arrival rate, the number of new sessions is fewer, as is the number of concurrent sessions.

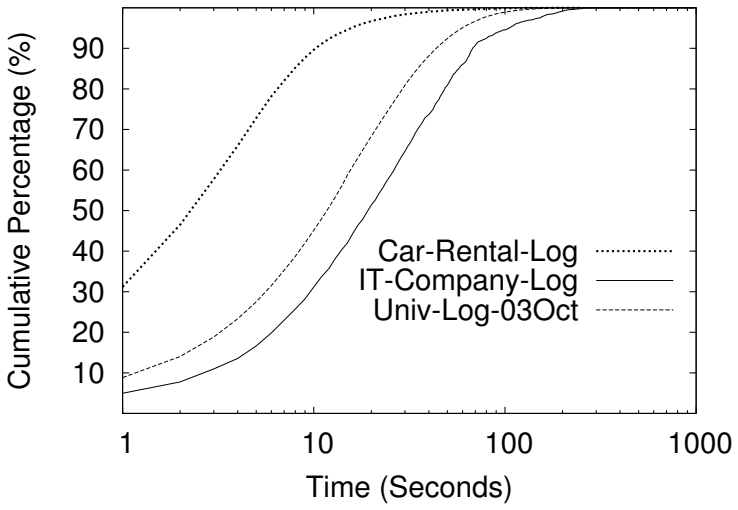


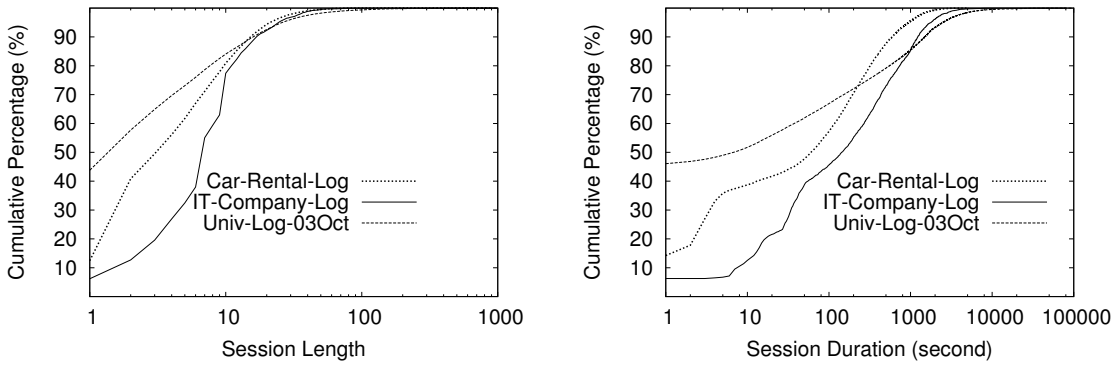
Figure 6.2: The Distribution of Session Inter-arrival Times

The session inter-arrival times are quite different between the three logs. The time granularity in the logs is one second, which limits the precision of the analysis. If two sessions start within the same second, the inter-arrival time between them is 0. Car-Rental-Log shows a much shorter session inter-arrival time than those of the other two logs. The percentages of sessions which have 0 inter-arrival times from

previous sessions are 31%, 9.3%, and 6.3% for Car-Rental-Log, Univ-Log-Oct03 and IT-Company-Log, respectively (Figure 6.2). In Car-Rental-Log, only 10% of sessions start later than 10 seconds from the starting time for the previous session. In comparison, this number is 55% and 70% for Univ-Log-Oct03 and IT-Company-Log, respectively.

6.1.3 Session Length and Duration

The session length distribution (Figure 6.3(a)) indicates that nearly half (43%) of the sessions in Univ-Log-Oct03 have only one request. These one-request sessions are mostly related to visits to personal Web pages owned by students and faculty members of the CS department. On the other hand, there are some extremely long sessions that are longer than 100 requests. A few of these long sessions are as long as 1000 requests. These are believed to be robot sessions. There are only 12% and 7% of sessions with only one request for Car-Rental-Log and IT-Company-Log, respectively. There are almost no sessions that are longer than 100 requests in these two logs.



(a) Length

(b) Duration

Figure 6.3: CDFs for Session Length and Duration

Most of the sessions (96% for Car-Rental-Log, 87% for the other two logs) last less than 1000 seconds (Figure 6.3 (b)). This observation is consistent with Menascé *et al.* [45]. The sessions in Car-Rental-Log have shorter durations in comparison to

those of the other two logs.

6.2 Obtaining Session Clusters

Session clusters were obtained using the methodology described in Chapter 4. A set of session clusters is obtained by selecting a session representation and applying the hybrid clustering algorithm to sessions in a log. Several sets of session clusters can be independently obtained from a log by applying the hybrid clustering algorithm to different session representations.

Three session attributes are selected to represent a session in Car-Rental-Log: *Pages Requested*, *Navigation Pattern*, and *Resource Usage*. Accordingly, 3 separate sets of session clusters are obtained. The hybrid clustering algorithm identifies 5 clusters for each session cluster set. In total, 15 session clusters are obtained for Car-Rental-Log.

For IT-Company-Log and Univ-Log-Oct03, the *Pages Requested* and *Navigation Pattern* are used to represent a session. The *Resource Usage* is not used since related information is not available in these logs. Two separate sets of session clusters are obtained for each of these logs. The hybrid clustering algorithm identifies 2 and 8 session clusters for each session representation in IT-Company-Log and Univ-Log-Oct03, respectively. In total, 4 and 16 session clusters are obtained for IT-Company-Log and Univ-Log-Oct03, respectively.

Session clusters of the same set are named with a common prefix which indicates the session attribute used for clustering. The prefixes pagC, nvgC, and resC correspond to the session attributes of *Pages Requested*, *Navigation Pattern*, and *Resource Usage*, respectively. For example, the first and second session clusters obtained by using *Pages Requested* are named pagC-1 and pagC-2, and the the first and second session clusters obtained by using *Navigation Pattern* are named nvgC-1 and nvgC-2, respectively. The numbers in this naming scheme indicate only the order in which a cluster appears in the clustering result. The names pagC-1 and nvgC-1 do not indicate that any logical connection exists between the two clusters,

although the numbers in both names are the same.

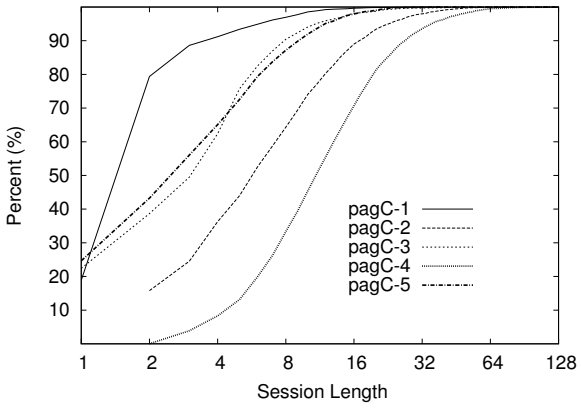
6.3 Session Groups for Car-Rental-Log

The session groups in Car-Rental-Log are identified and analyzed in this section. Three sets and a total of 15 session clusters were identified for this log. The characteristics for a session cluster are analyzed based on the statistical properties (Table 6.1), the average frequency for a session in a cluster to request a Web page category (Table 6.2), and the CDFs for session length and duration (Figure 6.4). By analyzing these clusters, 5 session groups are identified. Their characteristics are described as follows:

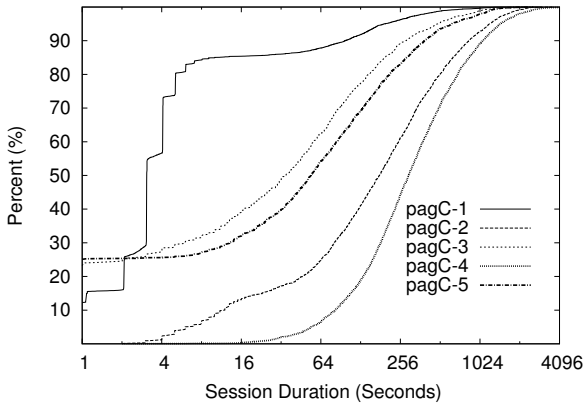
Table 6.1: Characteristics of Session Clusters (Car-Rental-Log)

Sessions			Avg. Length	Avg. Duration (Seconds)		Avg. SRT (Seconds)		Avg. scByte (bytes)		Avg. csByte (bytes)	
Clusters	Number	Percent (%)	per session	per session	per request	per session	per request	per session	per request	per session	per request
pagC-1	5730	34.7	2.5	34.8	13.9	3.7	1.5	54226.1	21664.7	1713.4	684.6
pagC-2	3100	8.8	8.5	326.1	38.2	11.6	1.4	96895.8	11348.3	6974.9	816.9
pagC-3	1349	8.2	4.5	108.9	24.0	0.9	0.2	18904.4	4163.6	2651.0	583.9
pagC-4	3601	21.8	14.5	467.5	32.3	10.6	0.7	153678.8	10624.3	9673.3	668.7
pagC-5	2731	16.5	4.7	145.8	31.1	0.8	0.2	11675.3	2488.3	2617.2	557.8
nvgC-1	6056	36.7	3.3	67.3	20.4	4.4	1.3	67007.1	20357.9	2302.2	699.5
nvgC-2	2127	12.9	8.1	282.4	35.0	13.0	1.6	96903.3	12018.3	6979.7	865.7
nvgC-3	4734	28.7	6.6	221.6	33.7	3.2	0.5	53702.1	8165.3	4001.7	608.5
nvgC-4	1522	9.2	9.4	356.1	38.0	9.2	1.0	83278.2	8882.2	6668.3	711.2
nvgC-5	2072	15.5	14.2	405.2	28.6	7.3	0.5	110518.3	7802.2	9226.0	651.3
resC-1	3930	23.8	1.9	8.6	4.5	3.6	1.9	47738.6	24852.6	1173.7	611.0
resC-2	3080	18.7	9.1	307.6	33.6	5.9	0.6	89897.1	9833.2	5924.1	648.0
resC-3	4467	27.0	4.8	150.7	31.2	0.5	0.1	11741.7	2427.8	2499.5	516.8
resC-4	957	5.8	4.9	127.1	25.9	11.3	2.3	78167.2	15939.9	4526.1	923.0
resC-5	4077	24.7	12.2	407.9	33.4	13.1	1.1	154556.1	12644.5	9477.9	775.4

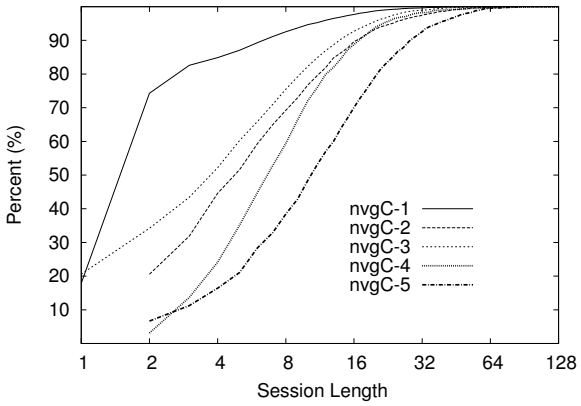
1. Rate-Checkers. This group is best represented by the session cluster resC-1. The actions taken by customers in this session group were basically to do a quick rate-checking and leave, visiting mainly two Web page categories, *CheckRate* (checking rate) and *Quote* (getting the result). Very little attention was paid to other relevant information (Table 6.2). Thus sessions were very short. About 88% of all sessions in this group were exactly 2 requests in



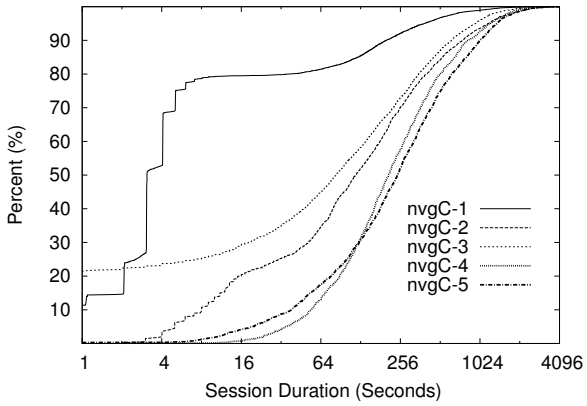
(a1) Length(Web Pages)



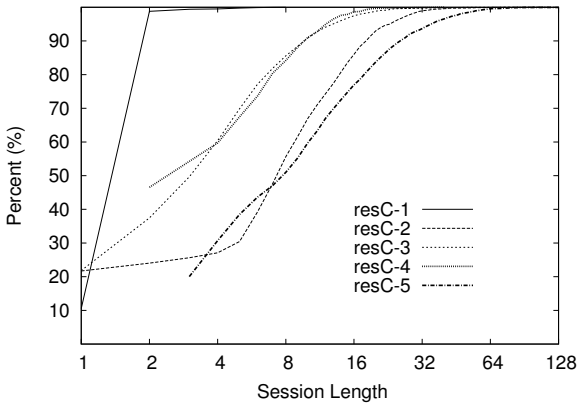
(a2) Duration (Web Pages)



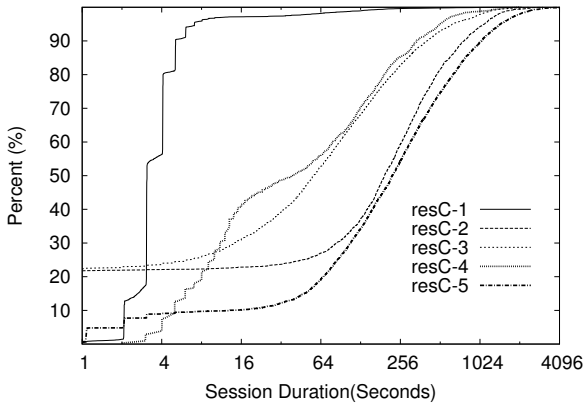
(b1) Length(Navigation)



(b2) Duration (Navigation)



(c1) Length(Resource)



(c2) Duration (Resource)

Figure 6.4: CDFs for Session Cluster Characteristics (Car-Rental-Log)

Table 6.2: Web Page Categories Requested in a Session Cluster (Car-Rental-Log)

List	P	H	RH	I	L	T	VH	S	CR	CA	MK	MD	Q	V	E	G	O
pagC-1	0.0	0.0	0.07	0.05	0.01	0.01	0.01	0.11	**1.18**	0.0	0.0	0.0	**1.03**	0.02	-	0.0	0.0
pagC-2	0.4	0.56	1.94	**0.86**	0.11	0.15	0.22	**1.78**	0.66	0.09	0.08	0.12	**1.78**	0.58	0.06	0.09	0.05
pagC-3	0.07	0.1	1.39	0.09	**1.56**	0.01	0.14	0.05	0.24	0.23	0.08	0.14	0.01	0.35	0.01	0.03	0.05
pagC-4	0.69	0.58	2.25	0.48	0.49	0.2	**0.81**	**3.43**	**2.67**	0.02	**0.11**	0.06	**2.19**	0.25	0.05	0.11	0.07
pagC-5	0.22	0.4	0.32	**0.63**	0.13	0.09	0.25	**2.0**	0.05	0.06	0.02	0.15	0.01	0.14	0.11	0.03	0.1
nvgC-1	0.07	0.02	0.18	0.09	0.01	0.03	0.06	0.14	**1.42**	0.01	0.01	0.01	**1.18**	0.04	0.0	0.02	0.01
nvgC-2	0.3	0.49	**2.24**	0.62	0.04	0.15	0.28	0.28	0.61	0.04	0.04	0.07	**2.32**	0.42	0.06	0.07	0.05
nvgC-3	0.44	0.35	1.15	0.66	**0.75**	0.13	0.36	0.5	**0.82**	0.13	0.08	0.19	0.39	0.35	0.09	0.08	0.1
nvgC-4	0.31	0.86	1.76	0.53	0.06	0.13	0.56	**2.36**	0.71	0.03	0.04	0.03	**1.56**	0.31	0.04	0.06	0.03
nvgC-5	0.37	0.47	1.56	0.35	0.39	0.1	0.51	**6.53**	**1.93**	0.03	**0.11**	0.05	**1.49**	0.2	0.03	0.04	0.03
resC-1	0.0	0.0	0.0	0.0	-	0.0	-	-	**0.89**	-	-	-	**1.03**	-	-	-	-
resC-2	0.44	0.42	1.29	0.62	0.34	0.19	0.44	**2.04**	**1.38**	0.19	**0.14**	0.16	0.75	0.52	0.07	0.09	0.08
resC-3	0.3	0.33	0.88	0.18	0.6	0.06	0.3	**1.66**	0.31	0.0	0.02	0.0	-	0.01	0.07	0.05	0.07
resC-4	0.03	0.04	**1.37**	0.11	0.01	0.04	0.04	0.13	0.75	-	0.01	-	**2.33**	0.04	-	0.0	0.0
resC-5	0.41	0.56	1.97	**0.91**	0.21	0.15	0.44	**1.74**	**2.21**	0.06	0.07	0.18	**2.65**	0.49	0.03	0.08	0.05

length (average length: 1.9) (Figure 6.4 (a3), Table 6.1). Most sessions (94%) were less than 9 seconds in duration (Figure 6.4 (b3)). Customers took quick actions; the time spent per request was on average only 4.5 seconds (Table 6.1). A careful examination of this group revealed that the same organization was responsible for nearly all of the sessions. This organization is a search facility, which examines many similar sites to compare prices for the same item. These sessions did not request images from the server and entered at the reservations home page, which is step 4 of the 5-step process in this rental site.

2. Confirmers. This group is best represented by the session cluster resC-4, which is a small cluster that contains 5.8% of all sessions. About 45% of the sessions had exactly 2 requests (Figure 6.4 (c1)). Further examination shows that these sessions identically requested the Web page groups *ResHome* and *Quote*. The objective of these sessions seemed to be confirming car reservations that had already been made.

3. Browsers. This group is best represented by the session cluster pagC-5. Actions taken were mostly searching, checking out relevant information (Web page categories: *Search, Locations, Info*, etc.)(Table 6.2). There were many

fewer rate-checking activities in comparison to those of the other session groups. A session in this session group visited the Web page category *CheckRate* only 0.05 times on average. About 40% of the sessions in this group were not longer than 2 requests (Figure 6.4 (a1)), but 10% of the sessions were longer than 9 requests (average session length: 4.6).

4. Location-Finders. This group is best represented by the session cluster pagC-3. The main activity of this group is to search for information related to car rental locations. This group is small and only 8.2% of sessions belong to the cluster pagC-3.

5. Buyers. This group is best represented by the session cluster pagC-4. The actions taken in this group included getting detailed information, searching, checking rates and making reservations. The most visited Web page categories were *Info, Promote, Group, CheckRate, Search*, etc. (Table 6.2). Many sessions were relatively long in both length and duration. The median for session length was about 11; the average session length was 14.5 and average session duration was 467.5 seconds (Table 6.1). The percentage of sessions making a reservation is much higher than those of other session groups, with an average frequency of 0.11 visiting the Web page category *MakeRes* in a session, as shown in Table 6.2.

Three session groups (Browsers, Location-Finders, and Buyers) are identified from the clusters by *Pages Requested*, and two (Rate-Checkers and Confirmers) are identified from the clusters by *Resource Usage*. In this case, using more than one cluster set does help to identify more session groups.

The analysis of session groups provides insights into the usage of the Web site, which are useful for improving server performance, managing system resources and optimizing revenue throughput. For example, the Buyers group is the group that is most likely to bring revenue to the Web site; thus it should be granted a higher priority in using system resources. Activities by this group involve intense database usage, and so it is the Confirmer group.

The Rate-Checker group is generated by robots or other programs used by other organizations, which consume a lot of system resources. It appears that other organizations are using the Web site for free for their benefits. Further investigation is necessary to determine if existence of these groups is also beneficial to the Web site. In the case that such activities are good for both parties, establishing a B2B data exchange channel for them should be considered since these are interactions between businesses. In this way, activities between businesses can be better controlled with respect to ensuring performance and to charging the other party. Since the percentage of requests from robots can be large, directing them to the B2B channels will reduce the regular B2C workload so that B2C requests also can be served better.

6.4 Session Groups for IT-Company-Log

For this log, 2 sets and a total of 4 session clusters were identified. Most customer activities are related to searching product categories, checking out promotions, and ordering goods. Category searching is the most common activity on the Web site. Almost all session groups spent close to half of the session in category search. Thus, category search is not a Web page category that gives a session its distinguishing characteristic. Two session groups are identified and they are described as follows:

- Browsers (pagC-1 and nvgC-1): This group performed category and product searches and checked out promotions, but did very little order-related activities (Table 6.4). The average session length is close to 7, which is relatively short (Table 6.3).

- Buyer (pagC-2 and nvgC-2): This group also performed category and product searches and checked out promotions, as did the Browser group. However, most order-related activities are associated with this group (Table 6.4). About 70% of sessions had a length in a narrow range from 6 to 9 requests (Figure 6.5 (a1, a2)). The average session length, however, is skewed up to 12.8 and 14.6 for pagC-2 and nvgC-2, respectively, by some long sessions (Table 6.3). The

Table 6.3: Characteristics of Session Clusters (IT-Company-Log)

				Avg. duration (Seconds)		Avg. scByte (bytes)	
Cluster	Size	Percent (%)	Avg. session length	per session	per request	per session	per request
pagC-1	1215	49.8	6.5	335.5	51.6	512	79.5
pagC-2	1261	50.2	12.8	596	46.6	1089	84.6
nvgC-1	1576	63.4	6.9	408	59.1	543	78.4
nvgC-2	907	46.6	14.6	572	39.2	1261	86.5

Table 6.4: Web Page Categories Requested in a Session Cluster (IT-Company-Log)

List	Promote	Category	Search	Cart	Order	Ship	Account	Other
pagC-1	2.4	2.98	0.61	0.06	0.13	0.01	0.17	0.12
pagC-2	2.14	5.41	0.79	0.06	3.16	0.02	0.07	1.22
nvgC-1	2.61	2.66	0.72	0.07	0.26	0.01	0.15	0.46
nvgC-2	1.69	6.90	0.67	0.04	4.14	0.02	0.07	1.07

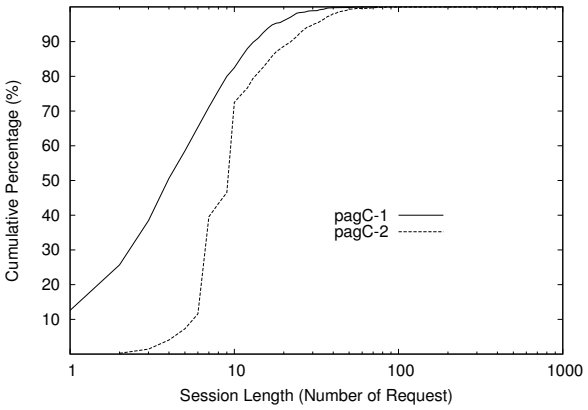
per-request duration is around 85 seconds, which is much longer than that of the Browser group.

In this case, all session groups (Browsers and Buyers) can be identified from the clusters either by *Pages Requested* or *Navigation Pattern*. Clusters by the two session representations can be used interchangeably for identifying session groups.

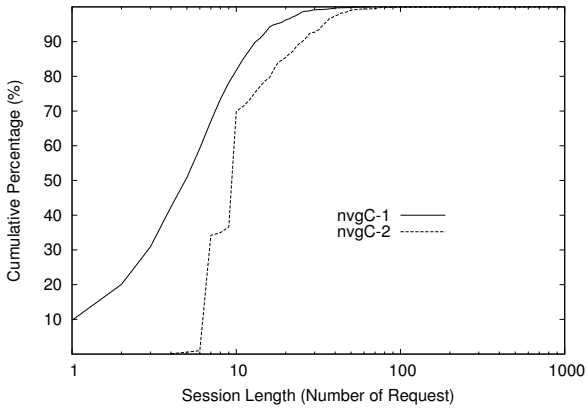
6.5 Session Groups for Univ-Log-Oct03

A total of 16 session clusters were obtained for this log (Table 6.5) and 8 session groups were identified as follows:

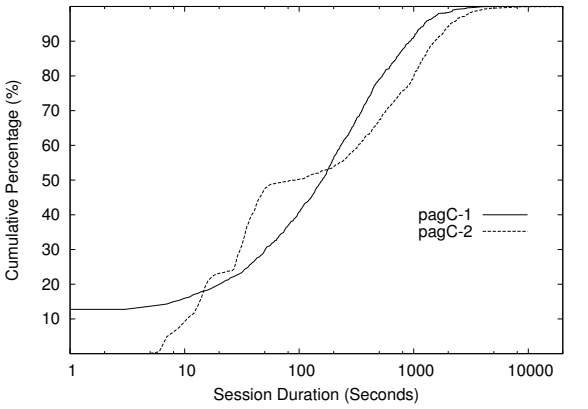
- Personal (pagC-1 or nvgC-1): Most requests were for Web pages on the personal Web pages that the department Web site hosts. The sessions were also short. About 60% of the sessions had only one request (Figure 6.6).



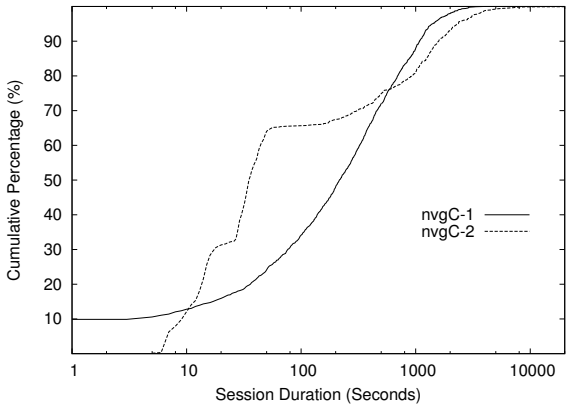
(a1) Length(Pages Requested)



(a2) Length(Navigation Pattern)



(b1) Duration(Pages Requested)



(b2) Duration(Navigation Pattern)

Figure 6.5: CDFs for Session Cluster Characteristics (IT-Company-Log)

Table 6.5: Characteristics of Session Clusters (Univ-Log-Oct03)

				Avg. duration (Seconds)		Avg. scByte (bytes)	
Session Cluster	Size	Percent (%)	Avg. session length	per session	per request	per session	per request
pagC-1	64418	46.1	3.1	461.1	150.6	35718	11665
pagC-2	8416	6.03	4.4	177.0	40.1	10871	2461
pagC-3	15007	10.7	8.0	482.7	60.4	12302	1539
pagC-4	16867	12.1	8.2	716.5	87.2	28283	3441
pagC-5	2111	1.51	1.2	119.8	103.8	355	307
pagC-6	5307	3.80	18.5	1207.4	65.2	15330	828
pagC-7	14515	10.4	5.5	317.3	57.5	50164	9085
pagC-8	13039	9.3	29.3	848.7	28.9	11058	377
nvgC-0	62866	45.0	3.8	519.8	135.6	36196	9442
nvgC-1	8631	6.18	4.4	276.5	62.7	11072	2511
nvgC-2	13843	9.91	9.2	398.2	43.1	11243	1216
nvgC-3	10007	7.16	2.5	309.3	121.9	21390	8427
nvgC-4	3805	2.72	2.8	312.0	110.8	4302	1527
nvgC-5	18469	13.2	15.5	1025.8	66.4	44050	2849
nvgC-6	4789	3.43	4.4	322.4	73.0	9394	2126
nvgC-7	17270	12.4	17.7	434.7	24.5	22743	1282

Table 6.6: Web Pages Categories Requested in a Session Cluster (Univ-Log-Oct03)

List	Class	EHandin	I-help	Intranet	Personal	Research	Resource	Robots	Other
pagC-1	0.06	0.00	0.00	0.00	**2.34**	0.09	0.4	0.05	0.12
pagC-2	0.02	0.00	0.00	0.00	0.00	**3.76**	0.55	0.02	0.07
pagC-3	0.85	0.10	0.03	0.22	**1.13**	0.16	**4.95**	0.03	0.54
pagC-4	**4.43**	0.03	0.03	0.38	0.67	0.29	0.48	0.03	**1.93**
pagC-5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1.15**	0.00
pagC-6	**11.9**	0.36	**1.38**	0.34	0.67	0.16	**2.46**	0.00	**1.31**
pagC-7	**5.39**	0.04	0.00	0.09	0.00	0.00	0.00	0.00	0.00
pagC-8	**5.66**	15.8	0.18	**5.37**	0.09	0.01	**1.66**	0.00	0.55
nvgC-1	0.48	0.01	0.02	0.01	**2.57**	0.07	0.48	0.04	0.16
nvgC-2	0.02	0.00	0.00	0.00	0.23	**3.95**	0.10	0.01	0.09
nvgC-3	0.22	0.18	0.03	0.24	0.47	0.21	**7.37**	0.01	0.49
nvgC-4	0.08	0.01	0.01	0.05	0.15	0.07	0.12	0.04	**2.01**
nvgC-5	0.04	0.07	0.01	**0.97**	0.30	0.09	0.19	**0.98**	0.16
nvgC-6	**11.5**	0.89	0.17	0.73	0.49	0.16	0.47	0.00	1.01
nvgC-7	**2.01**	0.29	**0.89**	0.48	0.11	0.01	0.24	0.00	0.39
nvgC-8	**2.85**	10.9	0.09	**3.41**	0.03	0.00	0.15	0.00	0.25

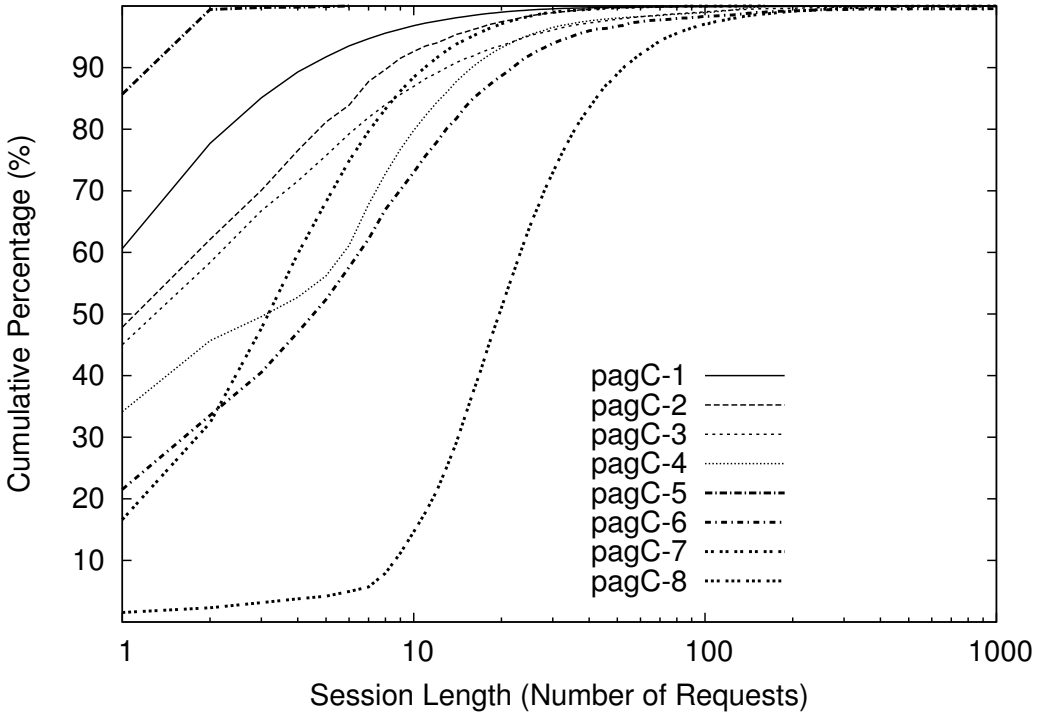
- Research (pagC-2 or nvgC-2): This group mainly looked for research-related information, such as research groups in the department, research projects, etc. Sessions were short, with average session length of 4.4. About 45% of sessions had only one request (Figure 6.6).

- Resource (pagC-3 or nvgC-3): This group visited the Web site for resources; for example, online tutorials, department events, programs available, employment opportunities in the department, etc. About 45% of sessions had only one request.

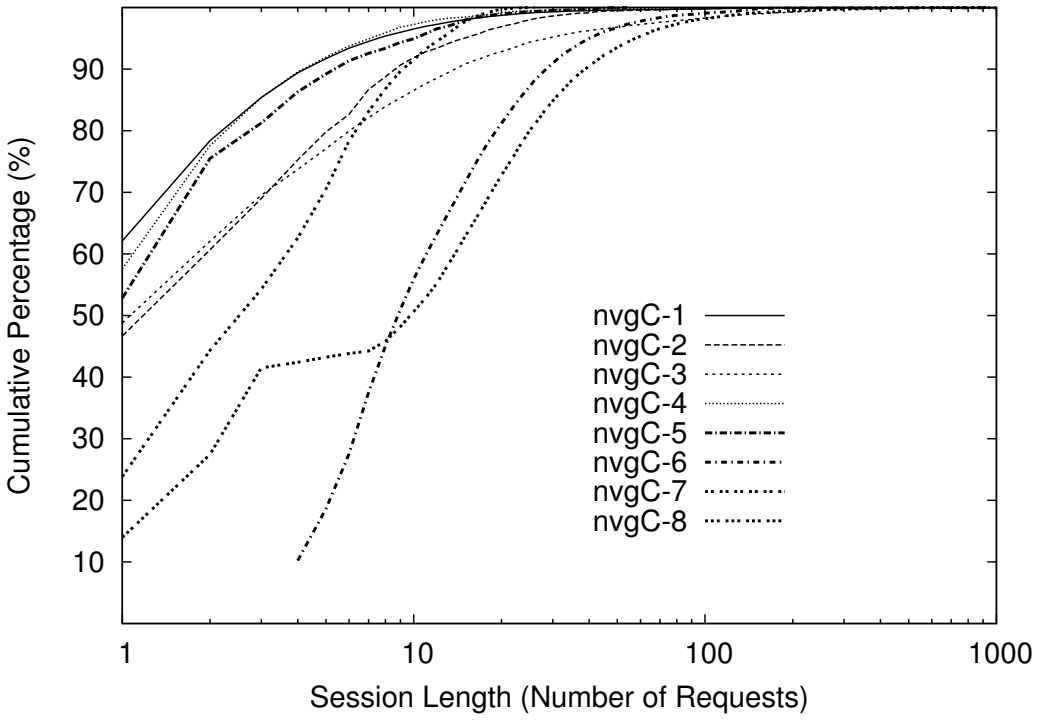
- Robots (pagC-5 or nvgC-5): This group comprised sessions that identified themselves as robots by requesting for the file, robot.txt. Typically, this is the first request a robot makes to a Web site. This group was not intended to cover all robot sessions. More than 85% of sessions had only one request. Many robot sessions were very long and had requests to many different Web page categories; thus they were more likely be grouped into other groups.

- I-help (pagC-6 or nvgC-7): This group of sessions formed around the centroid that was related to requests for *I-help*, which is a system that allows students to help each other with course-related questions. The frequency of requests for *I-help* in this group was 1.38, which was much higher than that of the others. The Web page category *Class* was also requested with a very high frequency in this group.

- Classes (pagC-7 or nvgC-6): These sessions looked mainly for class-related information. The Web page category *Class* was the most popular Web page category for Univ-Log-Oct03. About 30% of requests to the Web site requested *Class*. This is not a surprise since this is a Web site for the CS department of the University of Saskatchewan. One of the main functionalities of the Web site is to provide class-related information. High percentages of requests for *Class* were also observed in several session groups. This group was identified as the "Classes" group since *Class* was almost the only Web page category

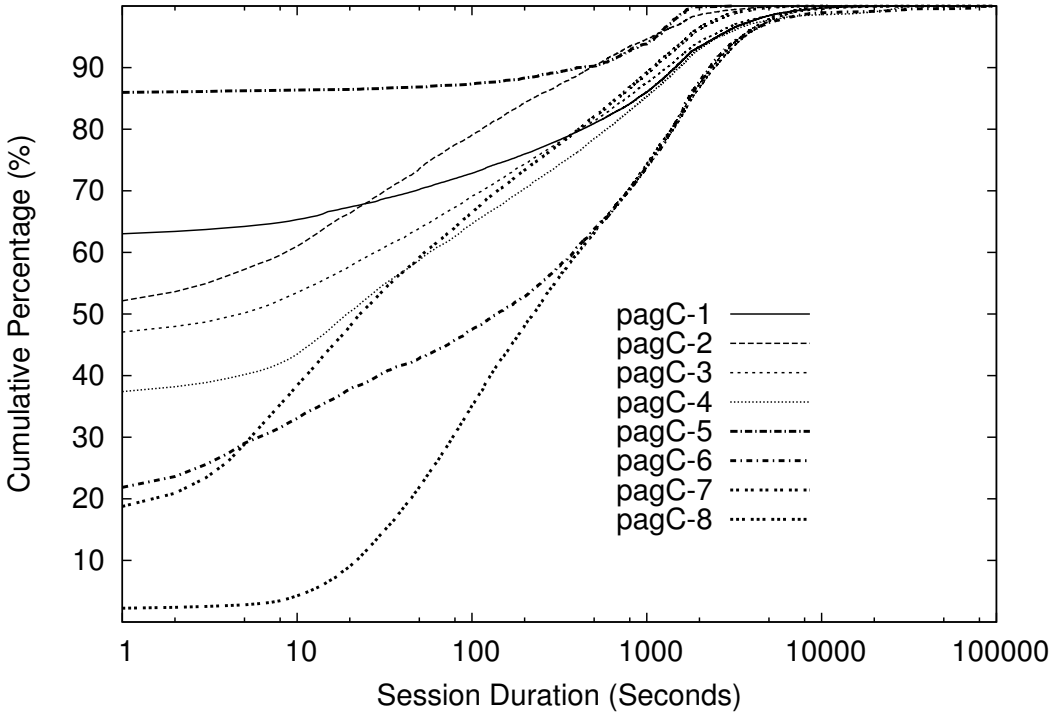


a) Session Length (Pages Requested)

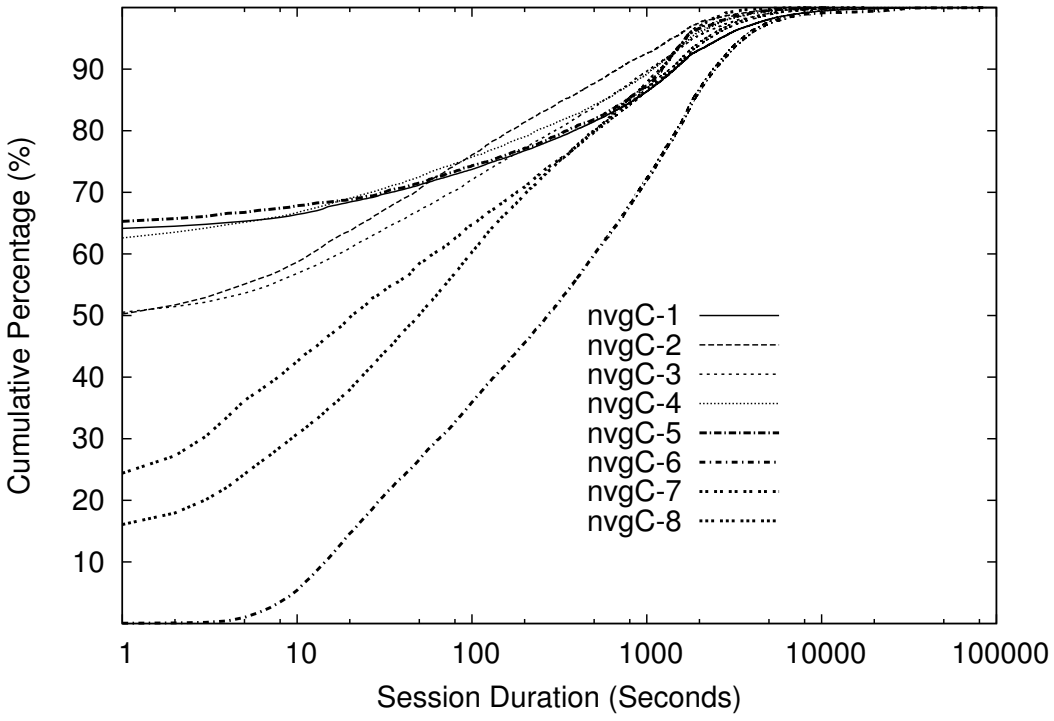


b) Session Length (Navigation Pattern)

Figure 6.6: CDFs for Session Clusters (Session Length)



a) Session Duration (Pages Requested)



b) Session Duration (Navigation Pattern)

Figure 6.7: CDF for Session Clusters (Session Duration)

requested by the group.

- EHandin/Intranet(pagC-8 or pagC-8): Sessions in this group were relatively long. The frequency in requesting *EHandin* was very high. The Web page category *EHandin* is related to a system called "EHandin", a system used by students to submit their assignments electronically. An average session in the group issued more than 10 requests for *EHandin*. In addition to *EHandin*, *Intranet* was also requested at a relatively high frequency in this group. *Intranet* is related to requests through the intranet for the CS department. Requests for the *EHandin* are transfered through the intranet.

- Others (pagC-4 or nvgC-4): Sessions in this group visit the Web site for Web page categories other than those mentioned above.

Similar to the case for IT-Company-Log, all session groups in Univ-Log-Oct03 can be identified from the clusters either by *Pages Requested* or *Navigation Pattern*. Clusters by the two session representations can be used interchangeably for identifying session groups.

6.6 Comparing Session Representations

6.6.1 Car-Rental-Log

The indirect comparison of session clusters by using session representations *Pages Requested*, *Navigation Pattern*, and *Resource Usage* is performed by checking how well the identified session groups are matched in each cluster set. The analysis of how a cluster matches a session group is mainly based on Table 6.1 and Table 6.2:

- pagC-1: shows the characteristics of Rate-Checker group. The main activities of this group are checking rates (*CheckRate*) and requesting quotations (*Quote*). It seems to be a typical Rate-Checker group since the frequency of requesting Web page categories other than *CheckRate* and *Quote* is very

low. However, this cluster only partially matches with Rate-Checker, since it is about 1.5 times larger than resC-1, which represents Rate-Checker. The cluster pagC-1 contains almost all sessions in resC-1, which were almost all generated by a robot. In addition, pagC-1 also contains sessions that are believed to be human rate-checkers. Thus, pagC-1 has a longer session duration and higher frequency in requesting *CheckRate*. Robot rate-checkers can do the checking in a few seconds and only check it once in a session, while human rate-checkers take much more time and may repeat the same type of requests in a session. The cluster pagC-1 represents a broader rate-checker group than that of resC-1.

- pagC-2: has the characteristics of Confirmer group. The main activities in this cluster are *ResHome* (frequency: 1.94) and *Quote* (frequency: 1.78). However, this session cluster only partially matches Confirmer since it is more than twice of the size of resC-4, which represents Confirmer.

- pagC-3: represents the Location-Finder group.

- pagC-4: represents the Buyers group.

- pagC-5: represents the Browsers group.

- nvgC-1: has the same characteristics as pagC-1, representing a broader rate-checker group which includes both robot and human rate-checkers. About 5% of the sessions in nvgC-1 are longer than 10 requests (Figure 6.4 (b1)), which skews up the average session length and duration, making them longer than those for pagC-1. However, the average number of server-sent bytes for nvgC-1, pagC-1, and resG-1 are very close, indicating that similar activities are performed in these clusters.

- nvgC-2: is similar to pagC-2 and has the characteristics of Confirmer group. The main activities in this cluster are *ResHome* (frequency: 2.24) and *Quote* (frequency: 2.32).

- nvgC-3: is a combination of Location-Finder and Browsers groups. The frequency of requesting *Locations* is 0.75, which is much higher than those of other clusters by *Navigation Pattern*. However, this cluster also performs many other browsing activities.

- nvgC-4: has the characteristics of Browsers group, but also has some other activities.

- nvgC-5: matches the Buyer group. Like pagC-4, it has a long average session length and duration, intense activities (searching, checking rate, and information), and a higher possibility of making reservations. The frequency of requesting *MakeRes* is 0.11, which is much higher than that of other clusters in the same set.

- resC-1: represents the Rate-Checker group.

- resC-2: has the characteristics of Buyer group.

- resC-3: is a combination of Location-Finder and Browsers groups. It is similar to nvgC-3.

- resC-4: represents the Confirmer group.

- resC-5: has the characteristics of Buyer group, but it performs many more rate-checking activities.

Session clusters by *Pages Requested* match well with the five identified groups Rate-Checkers (pagC-1), Confirmers (pagC-2), Location-Finders (pagC-3), Buyers (pagC-4), and Browsers (pagC-5). In the cluster set by *Navigation Pattern*, matches are found for groups Rate-Checkers (nvgC-1), Confirmers (nvgC-2), and Buyers (nvgC-5). The Location-Finders group and part of the Browsers group are in the cluster nvgC-3, while another part of the Browsers group is found in the cluster nvgC-4. Similarly, In the clusters set by *Navigation Pattern*, matches are found for groups Rate-Checkers (resC-1), Confirmers (resC-4), and Buyers (resC-2). The

Location-Finders group and part of the Browsers group are in the cluster resC-3. The cluster resC-5 has the characteristics of the Buyer group.

In general, session clusters by all three session representations identified, to some extent, the five session groups. The session clusters by *Pages Requested* have the best match with the session groups. But results by all session representations are similar.

A direct comparison between two cluster sets can be made based on Table 6.7, which demonstrates the best matches among different sets of clusters. For cluster Ci in a cluster set $S1$, one can find its best match in another cluster set $S2$ by examining the number of overlap sessions between Ci and every cluster in $S2$. The cluster which has the largest number of overlap sessions with Ci is the best match. If two clusters are the best match to each other, they are a pair of matching clusters.

Table 6.7 shows that most clusters overlap 50% with their best matches, and that there are many pairs of matching clusters. From the clusters obtained by using *Pages Requested* and *Navigation Pattern*, 4 pairs of clusters match each other (pagC-1 and nvgC-1, pagC-2 and nvgC-2, pagC-4 and nvgC-5, and pagC-5 and nvgC-3). Three matched pairs are found for clusters by *Pages Requested* and *Resource Usage*: pagC-1 and resC-1, pagC-4 and resC-2, and pagC-5 and resC-3. Three match pairs are also found for clusters by *Navigation Pattern* and *Resource Usage*: nvgC-1 and resC-1, nvgC-2 and resC-5, and nvgC-3 and resC-3. Thus, the matches indicate that the three session representations are related. On the other hand, however, the matches between clusters by different session representations are not strong. As shown in Table 6.7, many matching pairs have very different sizes.

6.6.2 IT-Company-Log

Session groups Browsers and Buyers can be identified from the clusters obtained by using session representations of both *Pages Requested* and *Navigation Pattern*. Table 6.8 shows there is a high degree of overlap between the cluster pairs pagC-1 and nvgC-1, and pagC-2 and nvgC-2. Thus, both indirect and direct comparisons

Table 6.7: Overlaps among Clusters (Car-Rental-Log)

Pages Requested		*Navigation Pattern*		
Cluster	Cluster Size	Best Matching Cluster	Cluster Size	Overlap Sessions
pagC-1	**5730**	nvgC-1	**6056**	**5464**
pagC-2	3100	nvgC-2	**2127**	**1700**
pagC-3	**1349**	nvgC-3	4734	**1330**
pagC-4	3601	nvgC-5	**2072**	**1057**
pagC-5	**2731**	nvgC-3	4734	**1695**
Navigation Pattern		*Pages Requested*		
Cluster	Cluster Size	Best Matching Cluster	Cluster Size	Overlap Sessions
nvgC-1	**6056**	pagC-1	**5730**	**5464**
nvgC-2	**2127**	pagC-2	3100	**1700**
nvgC-3	4734	pagC-5	**2731**	**1695**
nvgC-4	**1522**	pagC-4	3601	**826**
nvgC-5	**2072**	pagC-4	3601	**1057**
Pages Requested		*Resource Usage*		
Cluster	Cluster Size	Best Matching Cluster	Cluster Size	Overlap Sessions
pagC-1	5730	resC-1	**3930**	**3913**
pagC-2	**3100**	resC-5	4077	**1532**
pagC-3	**1349**	resC-3	4467	**1125**
pagC-4	3601	resC-2	**3080**	**1283**
pagC-5	**2731**	resC-3	4467	**2486**
Resource Usage		*Pages Requested*		
Cluster	Cluster Size	Best Matching Cluster	Cluster Size	Overlap Sessions
resC-1	**3930**	pagC-1	5730	**3913**
resC-2	**3080**	pagC-4	3601	**1283**
resC-3	4467	pagC-5	**2731**	**2486**
resC-4	**957**	pagC-2	3100	**667**
resC-5	4077	pagC-4	**3601**	**1643**
Navigation Pattern		*Resource Usage*		
Cluster	Cluster Size	Best Matching Cluster	Cluster Size	Overlap Sessions
nvgC-1	6056	resC-1	**3930**	**3863**
nvgC-2	**2127**	resC-5	4077	**1224**
nvgC-3	**4734**	resC-3	**4467**	**3144**
nvgC-4	**1522**	resC-5	4077	**841**
nvgC-5	**2072**	resC-3	4467	**1025**
Resource Usage		*Navigation Pattern*		
Cluster	Cluster Size	Best Matching Cluster	Cluster Size	Overlap Sessions
resC-1	**3930**	nvgC-1	6056	**3863**
resC-2	**3080**	nvgC-3	4734	**1116**
resC-3	**4467**	nvgC-3	**4734**	**3144**
resC-4	**957**	nvgC-2	2127	**743**
resC-5	4077	nvgC-2	**2127**	**1224**

Table 6.8: Overlaps among Clusters (IT-Company-Log)

Pages Requested		*Navigation Pattern*		
Cluster	Cluster Size	Best Matching Cluster	Cluster Size	Overlap Sessions
pagC-1	1215	nvgC-1	1576	1117
pagC-2	1261	nvgC-2	907	809

Table 6.9: Overlaps among Clusters (Univ-Log-Oct03)

Pages Requested		*Navigation Pattern*		
Cluster	Cluster Size	Best Matching Cluster	Cluster Size	Overlap Sessions
pagC-1	64418	nvgC-1	62866	61943
pagC-2	8416	nvgC-2	8631	7933
pagC-3	15007	nvgC-3	13843	12726
pagC-4	16867	nvgC-4	10007	8590
pagC-5	2111	nvgC-5	3805	2110
pagC-6	5307	nvgC-7	4789	3113
pagC-7	14515	nvgC-6	18469	7333
pagC-8	13039	nvgC-8	17270	9732

show that session clusters by *Pages Requested* and *Navigation Pattern* are similar.

6.6.3 Univ-Log-Oct03

As shown in the discussion for session groups in Univ-Log-Oct03, every cluster by *Pages Requested* matches a session group, and every cluster by *Navigation Pattern* also matches a session group. Table 6.9 shows that there is a roughly one-to-one correspondence between the session cluster obtained by *Pages Requested* and that by *Navigation Pattern*. For example, the cluster pairs pagC-0 versus nvgC-0, pagC-1 versus nvg-1, and pagC-2 versus nvgC-2 are almost the same. High degrees of overlap is also observed for other cluster pairs.

6.6.4 Summary of Session Representations

The analysis of all three logs shows that session clustering results obtained by both *Pages Requested* and *Navigation Pattern* are very similar. The session clustering result by *Resource Usage*, which is available only for Car-Rental-Log, shows less (but not substantially less) similarity to the results obtained by the other two session representations. In the case of Car-Rental-Log, using the three session representations together does provide more insight into session groups. However, if only a single session representation is used for session grouping, one can still obtain similar results.

6.7 Summary

The session-level characterization results indicate that session groups can be identified for a Web site. The characterization of the session groups can be used to improve server performance, implement Web site personalization, and improve resource management.

By indirect and direct comparison, it is observed that the session clustering results obtained by session representations *Pages Requested*, *Navigation Pattern*, and *Resource Usage* are similar. The performance implication is that these three representations can be used interchangeably to produce similar groupings.

Clustering by *Navigation Pattern* is complicated since there are too many coordinates involved (323 coordinates in the case of Car-Rental-Log). The computational complexity may result in complicated clustering results, since clustering is based on the distance between sessions and the distance is determined by coordinates.

The issue in grouping sessions by *Resource Usage* is that it is difficult to obtain resource usage data. The data are not available in IT-Company-Log and Univ-Log-Oct03. In case of the Car-Rental-Log, the resource usage data are based on approximations and assumptions.

Clustering by *Pages Requested* is much simpler than by *Navigation Pattern*, since there are many fewer coordinates involved. The information on Web pages requested is available in any HTTP log. Clustering by *Pages Requested* works well in identifying session groups in all cases.

Chapter 7

Contributions, Conclusions, and Future Work

7.1 Thesis Summary

This thesis is a study of three cases: two commercial Web sites and a CS department Web site. The main goals of the thesis are: (1) to characterize workload at the request, function, resource, and session levels; (2) to improve methodology for clustering sessions; and (3) to identify and characterize session groups.

This study consists of two logical parts. The first part presents the analysis of requests. Web workloads were analyzed at many levels. At the request level, workload characteristics such as file types, file size distribution, the popularity of Web objects, and the process of arrival requests were analyzed; at the function level, the functional composition of the Web site and the mix of request types in the workload were studied; at the resource level, the usage of SSL was quantified. The performance implications of the workload characteristics were also discussed.

The second part presented the analysis of sessions. Web user sessions were identified and characterized. Session representations were selected and a hybrid clustering algorithm based on k-means and minimum spanning tree methods was proposed for session grouping. Sessions were placed into groups by independently applying the hybrid clustering algorithm to three session representations: *Pages Requested*, *Navigation Pattern*, and *Resource Usage*. Session groups were then identified and characterized based on the session clusters obtained from the grouping processes. The session clustering results obtained by different session representations were compared and the association among them was discussed.

7.2 Contributions

The contribution of this thesis include the methods being proposed and used in the study and the facts being observed or confirmed. The methods may be used in other similar studies; the facts have performance implications and may be used to improve the performance of Web servers.

7.2.1 Contributions in Method

Methods to analyze requests and sessions are used in this thesis. The analysis of requests includes the characterization of file types, file sizes, file popularity, request arrival process, functions of the Web site, composition of the request stream, and usage of SSL. The analysis of sessions includes identifying and characterizing sessions, clustering sessions, and identifying session groups. Most ideas behind the analysis are taken from previous approaches, but some methods are new and some methods are enhanced, which are contributions of this study. These methods provide insight to understanding of the workload and may be applied to other similar studies. The contributions of the thesis in methods are as follows:

1. This study updated the file type category system that previously has been used for Web server workload characterization. In previous studies, file types being characterized include Image, HTML, Dynamic, Audio, Video, Formatted, Compressed, and Programs. Three new file types, namely Cascading Style Sheet, JavaScript, and XML, are added into the category system in this study. Files of these three types have been used more frequently in recent years and it is important to characterize them.

2. Dynamic files and static files were separately characterized. In previous Web server workload studies, all files identified on the Web server logs were characterized together and the aggregated characteristics were reported. In recent years, however, dynamic files have become key components in Web-based systems, since many functions of a Web site are implemented using dynamic files.

In the logs available for this study, almost all key functions are related to dynamic files. In addition, Web systems handle dynamic files and static files differently. Thus, they are separately characterized.

3. The mix of request types (request mix) was studied. The explicit characterization of request mix was not done in previous studies. A request type includes all requests for a function provided at a Web server. The request mix indicated the function composition of Web server workload. This function level characterization is useful in improving Web server design, understanding customers' usage of the Web site, and performing capacity planning.

4. New quantitative characterizations of SSL usage were presented. In particular, categorizing Web objects by the percentage of requests using SSL exceeds the analysis of the previous study [54]. The use of SSL is important in protecting sensitive data and ensuring security in an E-commerce transaction, but the processing through SSL is relatively expensive. Thus there is a trade-off between security and performance. This study is the first step to developing a method to handle the trade-off properly. The results indicate that this is worth pursuing since many objects that are requested via SSL do not appear to need privacy protection.

5. This thesis proposed a hybrid clustering algorithm to obtain session clusters. The k-means clustering algorithm has been used in many studies to identify session groups. To obtain a reasonable result with the k-means clustering algorithm, the key is to correctly select centroids for grouping. Insufficient details were given by previous studies on how centroids were chosen to cluster sessions. The proposed hybrid clustering algorithm combined the minimum spanning tree method and the k-means clustering algorithm, providing an empirical and iterative method to select centroids for grouping. This hybrid clustering algorithm was successful in effectively identifying session groups in this study.

6. In this study, three session representations – *Pages Requested*, *Navigation Pattern*, and *Resource Usage* – were compared for the purpose of identifying session groups. Using the three session representations together does provide more insight into session groups. However, session clustering results based on these session representations were similar enough so that it was possible to use different session representations interchangeably to produce similar groupings. The grouping based on one session representation was believed to be sufficient to answer questions in server performance, resource management, capacity planning and Web site personalization, which previously would have required multiple different groupings. Grouping by *Pages Requested* is recommended since it is the simplest and data on Web pages requested is easy to obtain in HTTP logs.

7.2.2 Contributions in Characterizing Workload and Session Groups

Based on three case studies, some workload characteristics which have been explicitly reported are observed in this study. These workload characteristics and related performance implications are as follows:

1. The batch referencing behaviour shows that some embedded images are always requested together. If the Web page is a popular one, such as the home page for a site, the amount of workload and overhead generated by requesting these batched embedded images is high. If a mechanism can be built to handle (i.e., requesting, sending, and caching) the batched embedded images in a Web page as a bundle, server performance can be improved.

2. The request mix was relatively stable when the time scale for measuring it gets large enough. This stability indicates that customers are looking for similar goods/services throughout the day. This observation may be used to improve server performance. For example, the arriving requests can be queued by the

types of functions they require, and be scheduled for execution by taking into account the request mix. The stable request mix may also be used to forecast workload. For example, assuming sales will increase by 50%, the volume of a specific request type can be predicted based on request mix.

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 objects in the site with respect to security protection.

4. Session groups of different characteristics were identified for all logs. The analysis of session groups is helpful in improving system performance, maximizing revenue throughput of the system, providing better services to customers, and managing and planning system resources.

5. Some session groups were more closely related to revenue-generating E-commerce activities such as selecting goods/services and placing orders. These groups could be granted a higher priority in using system resources than groups that are more focused on activities that are unlikely to bring revenue to the Web site. Identifying these session groups in real time requires further analysis of the logs and an algorithm to predict the user's future behaviour.

6. There are session groups in Car-Rental-Log that correspond to robots or programs that are used by other organizations, though the target of the Web site is individuals. Alternative architecture based on this observation to create separate B2B communication channels was suggested, but not evaluated.

7. Some groups make exclusive use of a particular function and thus have a high demand on a specific resource of the system. Such information is useful in organizing server resources.

7.3 Directions for Future Research

There are several directions that can be investigated in future research:

1. Analytical models can be established to analyze navigation patterns at a Web site. These models will provide a deeper insight into customers' interactions with the Web site.

2. A synthetic E-commerce workload generator based on both the request-level and session-level workload characterization results can be built. The synthetic workload should be session-based and match the general customer behaviour. The workload it generates would be more realistic and thus more helpful in server performance studies.

3. The workload characteristics and system behaviour at both the application server and database server levels should be explored. The workload and resource distributions among Web servers, application servers, and database servers should be analyzed. If workload is not reasonably balanced or matched with available resources at these servers, there will be a potential bottleneck in the system. There are many factors affecting workload distribution in the three tiers of an E-commerce system. Workload combinations and system architecture are among the most important factors. For example, changes in request types would result in changes in workload distribution in the system since different session groups have differences in resource usages. This research is complicated, but it can be initiated with simple experimental designs such as changing workload combinations under the same architecture.

4. Better data collection should be obtained so that more comprehensive analysis such as quantitative analysis of the consequences of different SSL uses and revenue throughput can be performed.

5. Since unintentional B2B behaviour was observed, system design alternative could be explored which isolate this traffic.

6. In order to improve server performance, research should be performed to develop and evaluate algorithms to identify session groups in real time.

References

[1] V. Almeida, A. Bestavros, M. Crovella, and A. Oliveira. Characterizing Reference Locality in the WWW. In *IEEE/ACM International Conference on Parallel and Distributed Systems (PDIS)*, pages 92–103, Miami Beach, FL, USA, December 1996.

[2] C. Anderson, P. Domingos, and D. Weld. Relational Markov Models and Their Application to Adaptive Web Navigation. In *Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 143–152, Edmonton, AB, Canada, 2002.

[3] M. Arlitt. Characterizing Web User Sessions. *ACM SIGMETRICS Performance Evaluation Review*, 28(2):50–63, 2000.

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

[5] M. Arlitt, D. Krishnamurthy, and J. Rolia. Characterizing the Scalability of a Large Web-based Shopping System. *ACM Transactions on Internet Technology (TOIT)*, 1(1):44–69, 2001.

[6] M. Arlitt and C. 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, Philadelphia, PA, USA, May 1996.

[7] A. Banerjee and J. Ghosh. Clickstream Clustering Using Weighted Longest Common Subsequences. In *Proceedings of the Web Mining Workshop at the 1st SIAM Conference on Data Mining*, pages 34–40, Chicago, IL, USA, April 2001.

[8] P. Barford. Web Server Performance Analysis, Tutorial at ACM SIGMETRICS, Atlanta, Georgia, USA, May 1999.

[9] P. Barford, A. Bestavros, A. Bradley, and M. Crovellar. Changes in Web Client Access Patterns: Characteristics and Caching Implications. *World Wide Web*, 2:15+, January 1999.

[10] P. Berkhin, J. Beche, and D. Randall. Interactive Path Analysis of Web Site Traffic. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 414–419, San Francisco, CA, USA, August 2001.

[11] J. Borger and M. Levene. Data Mining of User Navigation Patterns. In *Workshop on Web Usage Analysis and User Profiling*, pages 31–36, San Diego, CA, USA, August 1999.

[12] 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, USA, March 1999.

[13] Statistics Canada. A Reality Check to Defining eCommerce. Website, 1999. `http://www.statcan.ca`.

[14] H. M. Chen and P. Mohapatra. Session-Based Overload Control in QoS-aware Web Servers. In *IEEE INFOCOM*, New York, June 2002.

[15] X. P. Chen, P. Mohapatra, and H. M. Chen. An Admission Control Scheme for Predictable Server Response Time for Web Accesses. In *Proceedings of the Tenth International Conference on World Wide Web*, pages 545–554, Hong Kong, China, May 2001.

[16] L. Cherkasova and P. Phaal. Session Based Admission Control: A Mechanism for Improving the Performance of an Overloaded Web Server. Technical Report HPL-98-119, HP Lab Technical Reports, 1998.

[17] R. Cooley. Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data, Ph.D. Thesis, University of Minnesota, MN, USA, May 2000.

[18] R. Cooley. The Use of Web Structure and Content to Identify Subjectively Interesting Web Usage Patterns. *ACM Transactions on Internet Technology (TOIT)*, 3(2):93–116, 2003.

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

[20] A. Datta, K. Dutta, D. VanderMeer, K. Ramamritham, and S. B. Navathe. An Architecture to Support Scalable Online Personalization on the Web. *The VLDB Journal - The International Journal on Very Large Data Bases*, 10(1):104–117, 2001.

[21] K. Dutta, D. VanderMeer, A. Datta, and K. Ramamritham. Discovering Critical Edge Sequences in E-commerce Catalogs. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 65–74, Tampa, FL, USA, October 2001.

[22] V. Estivill-Castro and J. H. Yang. Categorizing Visitors Dynamically by Fast and Robust Clustering of Access Logs. *Lecture Notes in Computer Science*, 2198:498+, 2001.

[23] Organisation for Economic Co-operation and Development (OECD). Electronic Commerce: Opportunities and Challengers for Government (The Sacher Report). Website, 1997. `http://www.oecd.org`.

[24] Organization for Economic Cooperation and Development(OECD). Defining and Measuring E-Commerce: A Status Report, DSTI/ICCP/IIS(99)4/FINAL. Website, October 1999. `http://www.oecd.org`.

[25] Organization for Economic Cooperation and Development(OECD). E-commerce: Impacts and Policy Challenges (Economics Development Working Papers No.252). Website, June 2000. `http://www.oecd.org`.

[26] Organization for Economic Cooperation and Development(OECD). The OECD Definitions of Internet and Electronic Commerce Transactions. Website, April 2000. `http://www.oecd.org`.

[27] International Orgranization for Standardization. Information Technology – Business Agreement Semantic Descriptive Techniques – Part 1: Operational Aspects of Open-EDI for Implementation (ISO/IEC 15944-1:2002). Website, September 2002. `http://www.iso.org/`.

[28] Y. Fu, M. Creado, and M. Shih. Adaptive Web Sites by Web Usage Mining. In *International Conference on Internet Computing (IC'2001)*, volume 1, pages 28–34, Las Vegas, NV, USA, June 2001.

[29] Y. Fu, K. Sandhu, and M. Shih. Fast Clustering of Web Users Based on Navigation Patterns. In *World Multiconference on Systemics, Cybernetics and Informatics (SCI/ISAS'99)*, pages 560–567, Orlando, FL, USA, August 1999.

[30] Y. Fu, K. Sandhu, and M. Shih. A Generalization-Based Approach to Clustering of Web Usage Sessions. In *International Workshop on Web Usage Analysis and User Profiling (WEBKDD'99)*, pages 21–38, San Diego, CA, USA, August 1999.

[31] Y. Fu and M. Shih. A Framework for Personal Web Usage Mining. In *International Conference on Internet Computing (IC'2002)*, pages 595–600, Las Vegas, NV, USA, June 2002.

[32] Y. Fu, M. Shih, M. Creado, and C. Ju. Reorganizing Web Sites Based on User Access Patterns. *International Journal of Intelligent Systems in Accounting, Finance and Management*, 11(1), 2002.

[33] 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)*, volume 3, San Antonio, TX, USA, November 2001.

[34] A. K. Ghosh. *E-commerce Security*. John Wiley & Sons, Inc., 1998.

[35] J. Heer and E. Chi. Identification of Web User Traffic Composition Using Multi-Modal Clustering and Information Scent. In *Proceedings of the Workshop on Web Mining, SIAM Conference on Data Mining*, pages 51–58, Chicago, IL, USA, April 2001.

[36] J. Heer and E. Chi. Mining the Structure of User Activity using Cluster Stability. In *Proceedings of the Workshop on Web Analytics, SIAM Conference on Data Mining*, Arlington, VA, USA, April 2002.

[37] J. Heer and E. Chi. Separating the Swarm: Categorization Methods for User Sessions on the Web. In *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, pages 243–250, Minneapolis, MN, USA, April 2002.

[38] A. Iyengar, M. Squillante, and L. Zhang. Analysis and Characterization of Large-Scale Web Server Access Patterns and Performance. *World Wide Web*, 2(12):85–100, 1999.

[39] R. Jain. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, Inc, 1992.

[40] R. Kosala and H. Blockeel. Web mining research: A Survey. *SIGKDD: SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery and Data Mining, ACM*, 2:1–15, July 2000.

[41] G. Kotsis, K. Krithivasan, and S. Raghavan. A Workload Characterization Methodology for WWW Applications. In *Performance and Management of Complex Communication Networks*, pages 153–173. Chapman Hall, 1998.

[42] D. Menascé and V. Almeida. *Scaling for E-Business*. Prentice Hall, 2001.

[43] D. Menascé, V. Almeida, R. Fonseca, and M. Mendes. A Methodology for Workload Characterization of E-commerce Sites. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, pages 119–128, Denver, CO, USA, 1999.

[44] D. Menascé, V. Almeida, R. Fonseca, and M. Mendes. Business-Oriented Resource Management Policies for E-commerce Servers. *Performance Evaluation*, 42:223–239, May 2000.

[45] D. Menascé, V. Almeida, R. Riedi, F. Ribeiro, R. Fonseca, and W. Meira Jr. In Search of Invariants for E-business Workloads. In *Proceedings of the 2nd ACM conference on Electronic Commerce*, pages 56–65, Minneapolis, MN, USA, 2000.

[46] B. Mobasher, H. Dai, and M. Tao. Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization. *Data Mining and Knowledge Discovery*, 6:61–82, 2002.

[47] A. A. Oke. Workload Characterization for Resource Management at World Wide Web Servers, Msc. Thesis, University of Saskatchewan, Saskatoon, SK, Canada, April 2001.

[48] 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.

[49] J. Pitkow. Summary of WWW Characterizations. *World Wide Web*, 2:3+, 1999.

[50] J. Pitkow and P. Pirolli. Mining Longest Repeating Subsequences to Predict World Wide Web Surfing. In *USENIX Symposium on Internet Technologies and Systems (USITS'99)*, Boulder, CO, USA, October 1999.

[51] C. Shahabi, A. Zarkesh, J. Adibi, and V. Shah. Knowledge Discovery from Users Web-page Navigation. In *the IEEE International Workshop on Research Issues in Data Engineering (RIDE)*, pages 20–29, Birmingham, UK, 1997.

[52] 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.

[53] J. Srivastava, R. Cooley, M. Deshpande, and P. N. Tan. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations*, 1(2):12–23, 2000.

[54] 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, CA, USA, June 2002.

[55] D. VanderMeer, K. Dutta, A. Datta, K. Ramamritham, and S. B. Navanthe. Enabling Scalable Online Personalization on the Web. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, pages 185–196, Minneapolis, MN, USA, October 2000.

[56] T. Voigt, R. Tewari, D. Freimuth, and A. Mehra. Kernel Mechanisms for Service Differentiation in Overloaded Web Servers. In *USENIX Annual Technical Conference*, pages 189–202, Boston, MA, USA, June 2001.

[57] J. T. Xiao, Y. C. Zhang, X. H. Jia, and T. Z. Li. Measuring Similarity of Interests for Clustering Web-Users. In *Proceedings of the 12th Australasian conference on Database technologies*, pages 107–114, Queensland, Australia, 2001.

[58] A. Ypma and T. Heskes. Categorization of Web Pages and User Clustering with Wixtures of Hidden Markov Models. In *Workshop Notes of International Workshop on Web Knowledge Discovery and Data mining, WEBKDD'02*, pages 31 – 43, Edmonton, AB, Canada, July 2002.

[59] G. K. Zipf. *Human Behavior and the Principle of Least Effort.* Addison-Wesley, 1949.