

THESIS
1
2001



This is to certify that the

thesis entitled

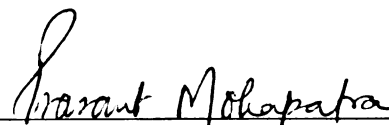
CHARACTERIZATION OF E-COMMERCE TRAFFIC

presented by

Udaykiran Vallamsetty

has been accepted towards fulfillment
of the requirements for

Master's degree in Computer Science
& Engineering


Major professor

Date July 26, 2001

PLACE IN RETURN BOX to remove this checkout from your record.
 TO AVOID FINES return on or before date due.
 MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE
NOV 28 2002		
JAN 15 2004		
AUG 25 2005 0207 06		

CHARACTERIZATION OF E-COMMERCE TRAFFIC

By

Udaykiran Vallamsetty

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Computer Science and Engineering

2001

Abstract

The world wide web has achieved immense popularity in the business world. Businesses are experiencing heavy bursts of traffic, causing a depreciation in revenues. It is thus essential to characterize the traffic behavior at these sites, a study that will facilitate the design and development of high-performance, reliable e-commerce servers. This thesis makes an effort in this direction.

Aggregated traffic arriving at a Business-to-Business (B2B) and a Business-to-Consumer (B2C) e-commerce site was collected and analyzed. High degree of self-similarity was found in the traffic (higher than that observed in web-environment) Heavy-tailed behavior of transfer times was established at both the sites. Traditionally this behavior has been attributed to the distribution of transfer sizes, which was not the case in B2C space. This implies that the heavy-tailed transfer times are actually caused by the behavior of back-end service time. An approach to reduce the burstiness in back-end service time was proposed, which splits the buffer cache to hold files with a maximum size restriction.

In B2B space, transfer-sizes were found to be heavy-tailed. Further study will be needed to split such a buffer cache with heavy-tailed arrivals. However, the effect of buffer cache on the service time was found to be negligible in B2B space, since a very high cache read hit ratio (99%) was seen. This workload characterization is a starting point for further studies. Inferences of this study can be further analyzed to aid in the design of high-performance e-commerce servers.

to my parents

ACKNOWLEDGMENTS

I would like to thank Dr. Prasant Mohapatra for his continual guidance and support throughout this work. I would also like to thank Dr. Lionel Ni and Dr. Sandeep Kulkarni for serving on my thesis committee. I would also like to thank Dr. Krishna Kant, Intel Corp. for his guidance throughout the course of this project.

Thanks to Teju, Jignesh, Singh and Ancha for their time, suggestions, support and coffee respectively. Finally I would like to thank my parents for everything.

TABLE OF CONTENTS

LIST OF FIGURES	vii
1 Introduction	1
1.1 Motivation	2
1.2 Objectives of the Thesis	3
1.3 Methodology and Inferences	4
1.4 Organization of the Thesis	5
2 Preliminaries	6
2.1 Definitions	6
2.1.1 Covariance	7
2.1.2 Auto-Covariance, Auto-Correlation	7
2.1.3 Stationarity	8
2.1.4 Long-range Dependence	8
2.1.5 Heavy-Tailed Distribution	9
2.1.6 Self-Similarity	9
2.2 Measuring H-parameter	10
2.2.1 The R/S Method	11
2.2.2 Variance-Time Plot	12
3 Related Work	14
3.1 Modeling Web Traffic	15
3.2 Capacity Planning	15
3.3 Invariants / Workload characterization	16
3.4 E-commerce Traffic Characterization	17
4 E-Commerce Architecture	19
4.1 Classification	20
4.1.1 Business-to-Business	22
4.1.2 Business-to-Consumer	22
4.2 System Configuration	23
4.2.1 Front-End	23
4.2.2 Back-end	25
4.3 Configuration: B2C	26
4.4 Configuration: B2B	28

5	Workload Characterization	30
5.1	Workload	30
5.2	Principal characteristics of the Workload	31
5.3	Data Collection	32
5.3.1	Types of Data Collected	32
5.3.2	Duration of Data	34
5.4	Characteristics of the Workload	34
5.4.1	OLTP Transactions	34
5.4.2	Secure Transactions	35
5.4.3	Dynamic Transactions	35
6	Front-End Characterization	37
6.1	Arrival Process	38
6.2	Processor Utilization	42
6.3	Response Time	43
6.4	Request/Response file sizes	47
6.5	Performance Implications	49
6.6	Variation in H-parameter	51
6.7	Summary	52
7	Back-End Characterization	53
7.1	Processor Utilization	54
7.2	Disk Accesses	57
7.3	Back-end Activity	59
7.4	Summary	60
8	Conclusion	61

LIST OF FIGURES

2.1	Hurst estimation for Poisson process with $\lambda = 15$ req/sec	12
4.1	Simple E-Commerce Site	21
4.2	Simplified configuration of the B2C site	27
4.3	Simplified configuration of the B2B site	29
6.1	Poisson arrival process with 15 req/sec, 1 sec granularity	38
6.2	Poisson arrival process with 15 req/sec, 10 sec granularity	38
6.3	Arrival process at B2C site, 6 sec granularity	39
6.4	Arrival process at B2B site, 6 sec granularity	39
6.5	Arrival process at 60 sec granularity	40
6.6	Arrival process at 600 sec granularity	40
6.7	A-V estimator test for self-similarity for arrival process	41
6.8	R/S plot test for self-similarity for arrival process at B2C site	42
6.9	R/S plot test for self-similarity for arrival process at B2B site	42
6.10	% Utilization at the front-end web server (4P), B2C	43
6.11	% Utilization at the front-end web server (2P), B2B	43
6.12	AV estimator for the front-end B2C web server(4P)	44
6.13	R/S estimator for front-end B2B server (2P)	44
6.14	Aggregated response time at the front-end web server (4P)	44
6.15	R/S test for estimating h-parameter for response time ($h = 0.56$)	44
6.16	AV estimator for the front-end web server response time (B2C)	46
6.17	LLCD of response-time distribution at the front-end	47
6.18	Estimated tail weight for the response-time distribution	47
6.19	Request size distribution over time	48
6.20	Response size distribution over time	48
6.21	LLCD of request size distribution	48
6.22	Estimated tail weight of request-size distribution	48
6.23	LLCD of response size distribution	49
6.24	Estimated tail weight of response-size distribution	49
6.25	Arrival Process at the Front-end server	51
6.26	Variation in H-parameter over 24 hours	51
7.1	Processor utilization of Catalog Server (5secs)	54
7.2	Processor utilization of the Main D/B server (5secs)	54
7.3	Estimation of H-parameter for Catalog Server ($H = 0.77$)	55
7.4	Estimation of H-parameter for Main D/B server ($H = 0.87$)	55
7.5	% Utilization of the B2B back-end server	56
7.6	H-parameter for the B2B database server ($H = 0.72$)	56
7.7	Queuing model for a simplified e-commerce request execution	57
7.8	File Operations per second from Main DB server (5sec)	58

7.9 Disk Queue Length at Main DB server (5sec) 58

7.10 Bytes transmitted per second from Main DB server (5sec) 59

7.11 Bytes transmitted per second from Catalog server (5sec) 59

Chapter 1

Introduction

The explosive popularity of Internet has propelled its usage in several commercial avenues. E-commerce, the usage of Internet for buying and selling products, has found a major presence in today's economy. E-commerce sites provide up-to-date information and services about products to users, and businesses alike. Services ranging from personalized shopping to automated interaction between corporations are provided by these web-sites. It has been reported that e-commerce sites generated \$132 Billion in 2000, more than double of the \$58 Billion reported in 1999 [1]. Even though the power of the servers hosting e-commerce sites has been increasing, e-commerce sites have been unable to improve their level of service provided to the users. It has been reported that around \$420 Million has been lost [2] in revenues due to slow processing of the transactions in 1999. Thus it is desirable and necessary to focus on the performance of the servers used in these environments.

There are two main classes of e-commerce sites, Business-to-Business(B2B) and Business-to-Consumer(B2C), providing services to corporations and individual users respectively. Web sites like Delphi, which provide services to corporations like General Motors come under B2B sites, whereas sites like Amazon.com providing services to general users come under B2C sites. Most of the revenue generated by B2C sites will

be during holiday season when the load on the system is maximum. During such periods, aggregate arrival rates orders of magnitude higher than normal rates will be observed at servers. This results in response-times orders of magnitude higher than in normal conditions, causing users to leave the site. Similarly, new products or other events may trigger surges in the B2B environment.

1.1 Motivation

Considering the revenues involved in e-business, availability and performance of the e-commerce servers become the two most important issues to be taken care. Service agreements with corporations or individual users have to be honored to gain or retain customers. Server overload can seriously compromise the availability and performance of the servers. To avoid these unwarranted situations, admission and overload control schemes have to be implemented at the server. Additionally load balancing is a popular approach for enhancing the performance. For these techniques to be effectively implemented, a good understanding of the workload is required. The techniques have to be tested under synthetic traffic and their effectiveness needs to be measured before deploying in the real-world environment. Due to the complex nature of e-commerce traffic and the access restrictions, there has been limited work reported where e-commerce traffic is modeled for synthetic generation. As a first step towards this goal, aggregate traffic arriving at the server has to be analyzed to understand its characteristics, high and low traffic periods and its effects on the server. This effort would require a complete characterization of the real workload seen at typical e-commerce sites. Also, the behavior of the server under different traffic and load conditions has to be understood to foresee overload of the server.

Due to limitations in access to live e-commerce sites, the studies reported on workload characterization have been limited, which restricted our understanding of

the behavior of e-commerce sites under different load conditions. Characteristics of this traffic have to be compared with web-traffic before applying any techniques devised for general web-traffic. Response-time predictions for QoS estimation have been done using controlled traffic [3], but similar behavior might not extrapolate to real traffic at e-commerce sites. Admission control work reported in [4] have been successful in web-environment but might not be effective in e-commerce environment. Our preliminary work has indicated significant differences in the traffic characteristics of e-commerce and general web servers.

1.2 Objectives of the Thesis

The objective of the study is to analyze the characteristics of e-commerce traffic and propose techniques for reducing the burstiness in the response-times. Traffic from a B2C and a B2B site is being used for the study. Front-end traffic has been previously studied in [5], but the impact of burstiness on the overall response-time of the system under different load conditions is the main concentration of this study. The response-time distribution is found to be heavy-tailed. This has been previously attributed to the heavy-tailed nature of request and response file-sizes. But the behavior of transfer sizes is not heavy-tailed, unlike the web-environment. The reasons for this behavior are investigated to develop techniques to reduce the burstiness in response-times. The back-end servers in an e-commerce site form the major components which effect the performance of the site [6]. The traffic arriving at these servers is characterized to obtain similar statistics about the impact of burstiness on the system. A correlation is drawn between the behavior of the front-end and the back-end servers under different load conditions. Performance implications from the results of the above experiments will give valuable information for improving e-commerce server performance.

1.3 Methodology and Inferences

Web traffic has been proven to be self-similar in nature [7, 8]. This *fractal nature* of web-traffic implies that, aggregate traffic does not smooth out as the number of sources increases (a Poisson-like arrival process would). In fact it has been shown that increasing the number of sources (users) increases the degree of burstiness. In this study, the workload is initially inspected for understanding the diurnal nature of the traffic. Different load periods were identified for both the B2C and B2B environments. These have been found to be complimentary in nature, which may be intuitive. A set of parameters were chosen for each site for each component which would impact the performance of the system to the maximum extent. Statistical tests are then used to prove the self-similar nature of the traffic at different scales. Two different tests are used for validating the results for each of the parameters. It has been observed that the arrival traffic is highly bursty in nature, much more than the burstiness seen in normal web-traffic [8]. Also preliminary tests have shown that the back-end utilization is more bursty than the front-end server utilization, the reasons for which are explained later. The transfer times are then studied. It is observed that the transfer times are heavy-tailed in nature. These are modeled using *Pareto distribution* [9].

Our preliminary studies have shown that response times cannot be predicted based on the file-sizes as the impact of queuing time will increase with the increase in the burstyness. Providing Quality of Service (QoS) is an important factor in B2B environment, and studies have been reported [10, 11] in normal web-environment for QoS support. In e-commerce environment, with the increase in burstiness at higher scales and the lack of understanding of behavior in high load periods, such studies have not been reported to effectively provide QoS support under any conditions. The inferences made in this study will provide a starting point for modeling e-commerce traffic. This study will also provide a basis to validate or refute the application of techniques used for web-traffic to the subset of e-commerce traffic.

Note

For this study data from two popular sites (one B2C and the other B2B) was used. Due to a non-disclosure agreement (NDA), the identity of these sites is not revealed. Throughout this thesis the two sites are identified as B2C site and B2B site. Without the NDA, we would not have been able to acquire the data for the study.

1.4 Organization of the Thesis

Discussion on work reported in the area is provided in Chapter 3. Chapter 2 discusses the basic mathematics involved in the study along with the implications of self-similarity. Chapter 4 discusses the configuration of e-commerce sites along with a discussion on the configuration of the sites used for this study. Chapter 5 discusses the characteristics of the workload used in the study. Preliminary observations on the workload used are discussed with appropriate explanations. Chapter 6 talks about the characterization of the traffic seen at the front-end of the two sites along with the results and analysis of the study. Chapter 7 talks about back-end traffic characterization along with a correlation study between the front-end and the back-end followed by a chapter discussing the conclusions with directions for future work.

Chapter 2

Preliminaries

In this chapter, a brief discussion of self-similarity is given. The different tests that are used to prove the presence of self-similarity in a time-series are described briefly, though the proofs of these are not given. For a detailed description of the properties and tests of self-similarity please refer [8, 12, 13] The discussion in the chapter closely follows the information obtained from these sources.

In Section 2.1 some basic mathematics used is presented, leading up to the definition of self-similarity. Section 2.2 deals with the methods used for estimating the Hurst parameter that describes the degree of self-similarity.

2.1 Definitions

A *time-series* [9] is a set of observations x_t , each one being recorded at a specified time t . In a *discrete-time series* the set of times in which observations are made is a discrete set, as is the case when observations are made at fixed time intervals. *Continuous-time series* are obtained when observations are made continuously over some time interval. The observations x_t are often supposed to be instances of a random variable X , and the time series is modeled as a stochastic process. A *stochastic process* is

a family of random variables $\{ X(t), t \in T \}$ with the same range. If T is an interval of real numbers then the process is said to have continuous time, and if T is a sequence of integers it is said to have *discrete time*. The term time series is often used to mean both the data and the process that is generating the series.

2.1.1 Covariance

Covariance and Correlation co-efficient are measures used to observe the dependence between random variables. Given two random variables, X and Y , with means μ_X and μ_Y , their covariance is

$$Cov(X, Y) = E[(X - \mu_X)(Y - \mu_Y)]$$

For independent variables, the covariance is zero. But the reverse is not true, it is possible for two variables to be dependent and still have zero covariance. The correlation co-efficient of the two random variables is

$$\rho(X, Y) = \frac{Cov(X, Y)}{\sigma_X * \sigma_Y}$$

The correlation co-efficient is always between -1 and 1.

2.1.2 Auto-Covariance, Auto-Correlation

The *Auto-Covariance* function of a process $\{ X(t), t \in T \}$ is defined by

$$\gamma_X(r, s) = Covariance(X(r), X(s)) \quad r, s \in T$$

and the *Auto-correlation* function of X_t can be defined as

$$\rho_x(h) = \text{Cor}(X_{t+h}, X_t).$$

2.1.3 Stationarity

A discrete time-series is said to be stationary if the Expected value of X_t is finite ($E(X_t) < \infty$) and is equal for all t , and it holds that

$$\gamma_X(r, s) = \gamma_X(r + t, s + t) \quad \forall r, s \text{ and } t$$

Stationarity implies that the mean of the series is same as the series progresses and that the dependence between equally spaced entries is same throughout the series.

2.1.4 Long-range Dependence

A stationary time-series is $\{X_i ; i = 1, 2, 3, \dots, n\}$ is *Long-range dependent* if its auto-correlation function $\rho_X(h)$ is non-summable over increasing lags i.e,

$$\sum_{h=1}^{\infty} \rho_X(h) = \infty$$

Auto-correlation function gives the dependence of a series to itself in the future. A long-range dependent process will have long memory, i.e dependence between variables will not decay even as the distance between the variables increases. The decay is actually hyperbolic in nature, which is slower than exponential decay.

2.1.5 Heavy-Tailed Distribution

A distribution is said to be *heavy-tailed* if

$$P[X \geq x] \sim x^{-\alpha}, \text{ as } x \rightarrow \infty, 0 < \alpha < 2.$$

This implies that regardless of the behavior of the distribution, for small values of random variable, the asymptotic shape of the distribution is hyperbolic. *Pareto* distribution is a simple heavy-tailed distribution used to model web transfer times. This distribution is used in this study for modeling the response time behavior of the servers. The probability mass function of this distribution is

$$p(x) = \alpha k^{-\alpha} x^{-\alpha-1}, \quad \alpha, k > 0, x \geq k.$$

and its cumulative distribution function is given by

$$F(x) = P[X \leq x] = 1 - (k/x)^\alpha$$

2.1.6 Self-Similarity

Given a stationary time-series $\{X_i ; i = 1, 2, 3, \dots, n\}$, let $X^{(m)}(k)$ be defined as

$$X^{(m)}(k) = (1/m) \sum_{i=(k-1)m+1}^{km} X(i) \quad k = 1, 2, \dots$$

be the time series aggregated at a level of m . This series is obtained by dividing the original series into blocks of size m and averaging over each block. *The time-series X is self-similar if it has the same auto-correlation function as that of the m -aggregated time-series $X^{(m)}(k)$ for all m .*

This means that the distribution of the original series is same as that of the aggregated series, i.e. changing the scale will not change the distribution. One of the main implications of this behavior is that, bursts in the series will not smooth out even at very high or all time-scales.

It should also be noted that self-similarity typically refers to the scaling behavior of a continuous or discrete time process, while long-range dependence defines the tail behavior of the auto-correlation function of a stationary time-series. A process with long-range dependence has an autocorrelation function $\rho_X(k) \approx k^{-\beta}$ as $k \rightarrow \infty$. Thus the autocorrelation function of such a series decays hyperbolically, which is slower than exponential decay. Hence $\beta < 1$. For self-similar time-series with long-range dependence, we have $H = 1 - \beta/2$, where H represents the degree of decay of the series' autocorrelation function or the degree of self-similarity. Since $\beta < 1$ and $\beta > 0$, $1/2 < H < 1$, and as $H \rightarrow 1$ the degree of both self-similarity and long-range dependence increases.

2.2 Measuring H-parameter

It is difficult to use the definition of self-similarity to measure the value of H or to prove the self-similar nature of a given finite traffic trace. So different features of self-similarity such as slowly varying variances are exploited in order to estimate the Hurst parameter. It should be noted that this value will be 0.5 for a smooth Poisson traffic as shown in Figure 2.1 and increases with the increase in the degree of self-similarity. The H-parameter can be used both as test for self-similarity, as described in previous section, and also as a measure of the degree of self-similarity present in a time-series. In this section we will discuss two statistical methods for calculating the Hurst-parameter in a finite time-series.

2.2.1 The R/S Method

Given the time-series, $\{X_i; i = 1, 2, 3, \dots, n\}$, we can define the m-aggregated time-series, $X^m(t)$. For an arrival time-series, this would represent the number of arrivals during the interval $n(t-1)$ to nt . The R/S statistic or the rescaled adjusted range is defined by the ratio

$$R/S = \frac{R(t, k)}{S(t, k)}$$

where

$$R(t, k) = \max_{0 \leq i \leq k} [Y_{t+i} - Y_t - \frac{i}{k}(Y_{t+k} - Y_t)] - \min_{0 \leq i \leq k} [Y_{t+i} - Y_t - \frac{i}{k}(Y_{t+k} - Y_t)]$$

is called the rescaled range and

$$S(t, k) = \sqrt{k^{-1} \sum_{i=t+1}^{t+k} (X_i - \bar{X}_{t,k})^2} \quad \text{where}$$

$$\bar{X}_{t,k} = k^{-1} \sum_{i=t+1}^{t+k} (X_i)$$

$S(t, k)$ makes it possible to study the properties that are independent of scale. R/S value is calculated for different values of t and k and the values of $\log(R/S)$ is plotted against $\log(k)$. The slope of the line fitted through, $\log(R/S)$ and $\log(k)$, will give an estimation of the Hurst parameter. The ratio of R/S cannot be calculated for all the possible values of t and k . Since we are using Log-Log plots, logarithmically spaced values of k for every t would give a good estimate of the Hurst parameter. In this study, the R/S value is calculated for every t , giving random number of equally spaced values for k , such that $t + k \leq n$.

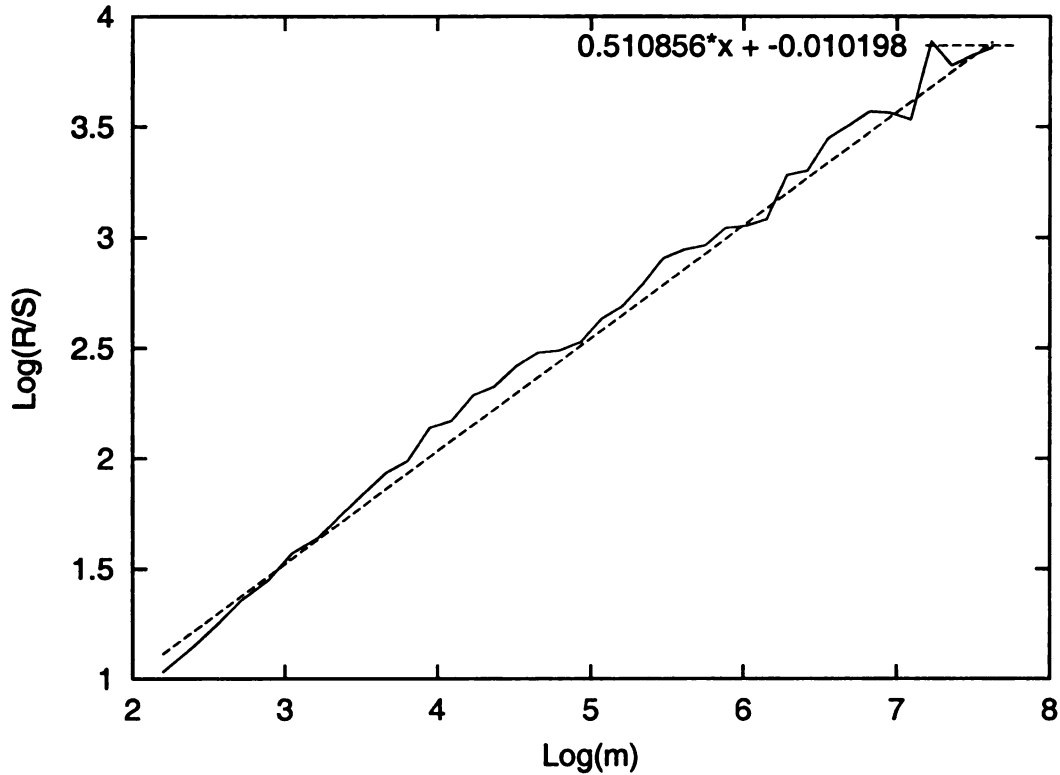


Figure 2.1: Hurst estimation for Poisson process with $\lambda = 15$ req/sec

2.2.2 Variance-Time Plot

Let $X^m(t)$ be a time-series representing the number of entries in each interval (bin) of size m . If for example the bin size has been chosen to be 100ms then $X^m(1)$ is the number of packets that arrived in the first 100 ms. Characteristic of long-range dependent processes is that the variance of the sample mean converges slower to zero than $1/n$ (the reciprocal of the sample size). It can be shown that

$$\text{var}(\bar{X}_n) \approx cn^{2H-2} \text{ where } c > 0$$

In practice, the mean of each pair of consecutive, non-overlapping bins is calculated and then the variance of these means is calculated. The 2-logarithm of the variance is plotted against the logarithm of the block size i.e 1. Then the same thing is done for blocks of size 4,8,16,...,length(X)/2 bins. The parameter H can be estimated by

fitting a simple least squares line through the resulting points and using the relation $\text{slope} = 2H - 2$. The values for the smallest and largest block sizes are usually not included when estimating H .

For both the tests described, linear regression was used to obtain the best fit line. Also the bucket size and time-scale were made less than the non-stationarity time-scale found in [5] for e-commerce traffic. It has been observed that the NST was around 900 secs for e-commerce traffic. So all the tests were performed within this time-scale only. The tests described in this section are used to observe the behavior and characteristics of the traffic at different components of the e-commerce server. A discussion on the general configuration of e-commerce servers, followed by a description of the two e-commerce sites analyzed in this thesis is given in the next chapter.

Chapter 3

Related Work

Workload characterization studies on Internet traffic can be classified into the following categories.

- Modeling traffic for synthetic generation
- Building a capacity planning model to obtain the bandwidth utilization and load on each component in the server by observing the client behavior.
- Building a set of invariants for performance analysis of web servers
- E-commerce traffic characterization
- Understanding the characteristics of traffic under varying load conditions and time-scales.

In the following sections each of the above categories of work will be discussed with emphasis on the relevances to this study.

3.1 Modeling Web Traffic

The notion of Self-Similarity in Ethernet traffic has been studied previously in [7, 8, 14]. In [7] the authors used the LAN traffic collected by Leland and Wilson [15] at Bellcore Morristown Research and Engineering Center. The authors demonstrated that Ethernet LAN traffic is statistically self-similar in nature. Traditional models like Pure-Poisson [16] or Poisson related models such as Poisson-batch or Markov-Modulated Poisson processes etc. were proven useless in modeling Ethernet LAN traffic [17]. This paper has also refuted the notion that aggregated Ethernet traffic has similarities to telephone traffic.

The authors bring out some of the major differences between models for self-similar traffic and the standard models used for packet traffic currently considered in the literature. They refute the conventional notion of Poisson like behavior of aggregate LAN traffic, that traffic does not smooth out by increasing the number of sources. In fact the authors have shown that the degree of self-similarity, which will be explained in a later section, increases with the increase in the number of traffic sources. Leland et al. help in establishing the inherent nature of the traffic that is observed, and provide a starting point to analyze the traffic and compare with the characteristics of the workload.

3.2 Capacity Planning

In [18] the authors have provided a methodology for determining the bandwidth for various components of a WWW Server. They used data from SPECweb96 benchmark, a proxy server and a dynamic server workload to build a capacity planning model for a web server. A correlation between the traffic arriving at a server and the architectural behavior of the server was made. Work has been reported in [14, 19, 20] about the performance implications of self-similar traffic on network performance, queuing delay

and packet loss. Previous studies on web traffic characterization [14, 19, 20, 21] have not looked at this aspect.

A few studies have been reported on the characterization of e-commerce traffic based on the client behavior. In [5] the authors have built a model for describing the user behavior based on the transactions carried out by a typical user upon entering an e-commerce site. The state of the system was modeled based on the transaction being performed at any instant. In [22] the authors have developed a resource utilization model for a server which represents the behavior of groups of users based on their usage of the site. This model represents the usage of each resource in the web server. The states of the model can be defined as a specific operation performed by a client at the server. The operations performed by the client can be strictly divided based on the resources used by that operation. This division would generate a model giving information about different resources utilized by the client at any instant for a given session. Resource Utilization Model for server will have an average usage information over all the clients. This kind of model would help in comparing the performance of two servers given two users at different load conditions. Most of the work done in this area has not looked at the actual traffic from e-commerce servers. An exception is the work reported in [5] where the authors have studied the traffic collected at a B2B and B2C e-commerce site.

3.3 Invariants / Workload characterization

In [23] the authors do a workload characterization of web traffic. The authors arrive at a set of parameters, called *invariants*. These represent the set of results which will hold true for any generic Internet server. (For example an invariant is, the file sizes transferred on an average is less than 21 Kilobytes). These invariants are used for doing further performance analysis of web caching based on the results about the file sizes. These parameters can also be used for comparing different sites.

These invariants or parameters were used in this work as a basis for studying the system behavior under different load conditions. It will be shown that most of the invariants are true even after the rapid change in the configuration of traffic arriving at servers. These parameters were also used to measure and correlate the effect of burstiness on the overall system performance. The impact of burstiness on these parameters has also been checked.

3.4 E-commerce Traffic Characterization

In [5] the authors have studied traffic arriving at e-commerce sites. In this paper the authors do a two-layer characterization of front-end e-commerce traffic. They study the traffic arriving at Business to Business (B2B) and Business to Customer (B2C) sites. Characteristics of the traffic have shown a distinct seasonality in the traffic over a single day. The authors find busy periods which are almost complimentary for the B2B and B2C environments. They show the presence of non-stationarity in the request arrival process to the server. Assuming that Non-Stationarity of the traffic is seen only at certain time-scales, the authors present a technique to analyze non-stationarity and long-range dependence properties in e-commerce traffic.

The above assumption was used in this study to analyze the traffic at time-scales less than the Non-stationary time-scale. This would also enable measurement of the burstiness factor using the R/S and Variance-Time plot techniques proven to be very reliable for stationary time-series [24, 8].

Some studies on E-commerce sites [25, 26, 27] have looked at various features of emerging technologies for providing increased security and accessibility to users. In this environment, the revenue generated has the highest importance and hence most of these studies have concentrated on increasing revenue generation by introducing new applications or technologies which would increase the performance of E-commerce sites based on the revenues generated per transaction.

As we can see above, not much work has been attempted in e-commerce traffic characterization. The main reason for this shortcoming is the unavailability of representative data. E-commerce sites have highly secure information in the traces and access logs. Due to the security implications e-commerce sites are reluctant to divulge this information for research purposes. Due to this, studies in this field are still in the preliminary stages. People are looking at the workload, trying to understand the characteristics of the traffic and its impact on the behavior of the system.

It should also be noted that most of the work reported on e-commerce traffic has been done on the front-end servers and to the best of our knowledge nothing has been reported on the back-end servers. The back-end servers are the ones which experience the maximum load in an e-commerce environment [6]. We would like to characterize the load on the back-end servers along with a study of the system characteristics collected from system logs in E-commerce sites. This would give an opportunity to study the correlation between the traffic arriving at the front-end and back-end of e-commerce servers. Further such a study would also provide an understanding of the impact of the front-end on the traffic arriving at the back-end. One of the main difference between normal web traffic and e-commerce traffic is the heavy percentage of *https* and *dynamic* requests. The impact of these kinds of requests on the overall traffic is explained in the next chapter.

Chapter 4

E-Commerce Architecture

In this chapter, a brief discussion on the architecture and design of a typical e-commerce site is given. The various building blocks in an e-commerce server are described right from the network to the database services. Detailed descriptions of the system architecture of the different servers being studied are given towards the end of the chapter. Additional details can be obtained from [6]. Section 4.2 gives a description of a typical e-commerce site. The sites that are used for this study, referred to as B2C site and B2B site, are discussed in the next sections.

A typical e-commerce site will comprise of the following components:

- A network connection to the Internet
- Web applications, act as web-servers or service providers for the users
- Database applications, provide the data and security for the information provided by the user
- Server operating system, the central point for all the above blocks.

A good e-commerce server can only be run through an architecture that meets these requirements across the network, web applications, database, and server operating

system. Any successful e-commerce implementation must also address these key characteristics:

- High availability
- Scalability
- Security and
- Performance.

Due to the highly critical nature of the applications, availability of the site becomes an important issue. Increasing growth in Internet usage and the growth in the speeds available to the users require e-commerce sites to be highly scalable for any changes in the environment. Financial and other personal information involved in most transactions require the site to be highly secure from intruders. Cost-performance of online stores is inherent in the business model of e-commerce and has to be maintained for profitability. Some of the issues involved in maintaining these characteristics are discussed in this chapter.

4.1 Classification

E-commerce sites can be broadly classified into two different categories.

- Business to Business (B2B)
- Business to Consumer (B2C)

The main difference between the above two categories of sites lies in the user population accessing these sites. Business-to-Business e-commerce sites serve transactions between different businesses whereas Business-to-Consumer sites serve general users over the Internet.

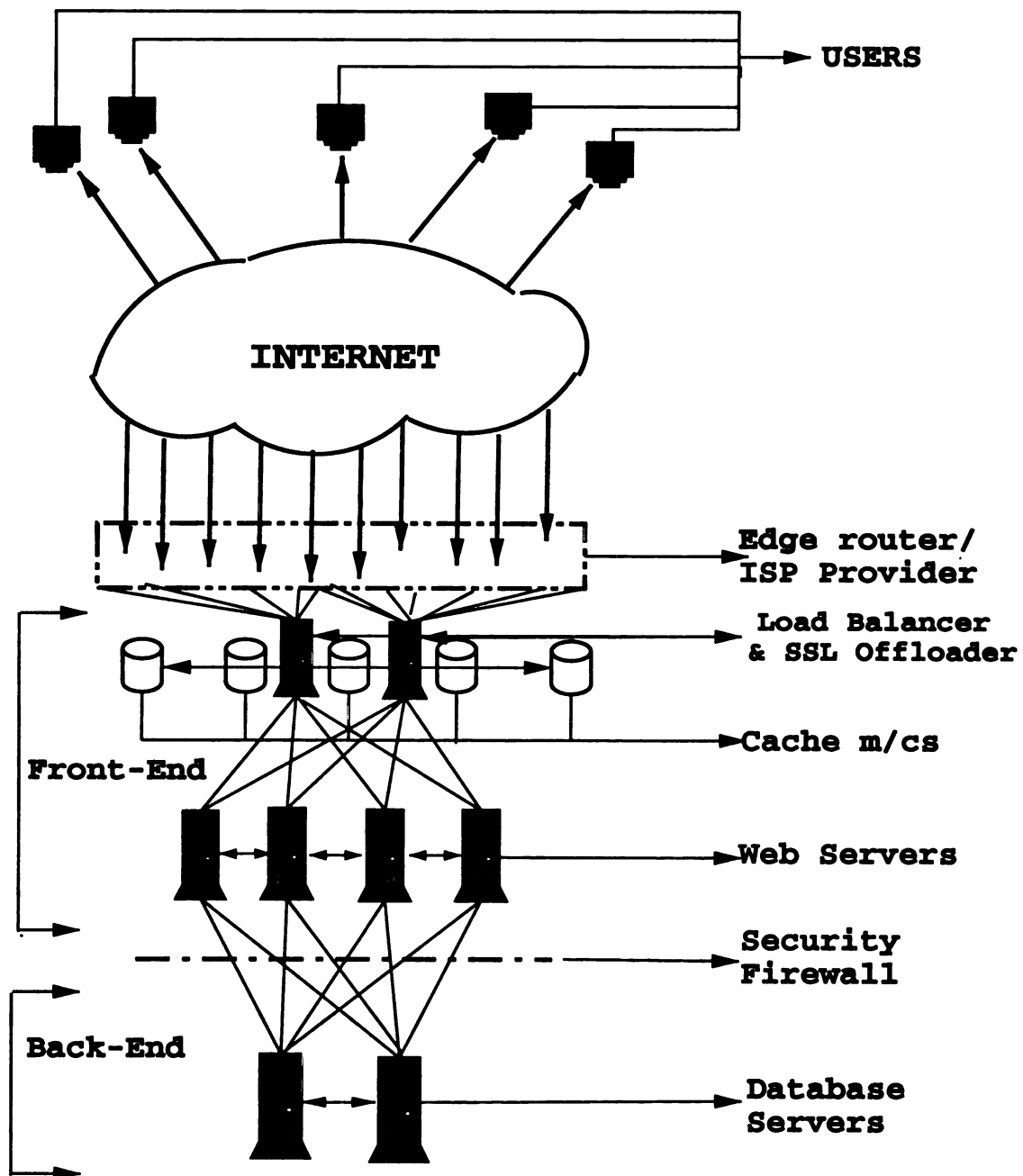


Figure 4.1: Simple E-Commerce Site

4.1.1 Business-to-Business

One of the main characteristics of this category of sites is the regularity in the arrival traffic [5]. In [5] it was observed that heavy traffic comes between 9am to 5pm, normal business hours. Regularity does not imply the lack of heavy spikes in the traffic. There will be sustained load on the system either due to seasonal effects or due to the availability of different services at the site. These sites can be categorized by the high amount of buying taking place in them. It has been observed that the percentage of transactions resulting in buying are very high compared to those in B2C environment.

Our preliminary analysis have revealed some very important features of B2B e-commerce sites. In B2B space, the population of users accessing a specific set of servers is known a-priori, along with the kind of transactions that will be taking place. This enables the designers to customize these sites to specific users, for specific transactions. With this information, the response time can be improved keeping the load on the system balanced evenly among all the different servers in the site.

4.1.2 Business-to-Consumer

B2C servers are the normal e-commerce sites where any user can get service. The security involved in B2C site is only restricted to any financial transactions involved, whereas in a B2B environment all the transactions are normally done in secure mode. One implication of this is that increased buying in a B2C environment can throttle the system since the designed system does not expect high percentage of buy transactions. Another important characteristic of a B2C site is the very low tolerance to delayed responses. This increases the need to make Quality of Service more important than providing absolute security for all the transactions, hence security is reserved for transactions involving buying.

4.2 System Configuration

In this section we will discuss the configuration of a typical e-commerce site. Since the general configuration of both B2B and B2C servers is similar, a typical configuration is described. Any differences observed between the two types of sites will be noted as the discussion progresses. Any e-commerce architecture consists of two main parts, the front-end and the back-end network. The front-end consists of web and application services accessible by the users over the Internet. The network devices that connect the front-end services to the Internet comprise of edge-routers, content-caching devices, load-balancers and security devices for client authorization and identification. Back-end consists of the security fire-walls and the database servers.

4.2.1 Front-End

Typically the front-end servers are comprised of the web server, application server, server load balancer and the SSL off-loader.

Web servers host the actual site content that clients see on their Web browsers. Web servers serve the different requests from the clients, comprising of static content, graphics, or dynamic content. These are the only systems in direct contact with the end user. In addition, Web servers are the only authorized hosts able to access the back-end database and application services as necessary. Majority of e-commerce sites address their scalability and high availability requirements by scaling out their Web servers. This is an easy approach due to the ease of scaling web server, with an added advantage that the increased speed at the front-end server will reduce the latency of requests arriving at the back-end servers.

The application servers are responsible for the business logic services. The application server will be the most heavily loaded server in the Business-to-Customer environment. This is due to the heavy traffic of dynamic and secure requests arriving at the server. In a large scale e-commerce site, there will be dedicated application

servers, alternatively these servers can be combined with the Web Servers or the Database servers. The decision is based on how the web server, business logic, and database services communicate. If the web servers make many small requests to the business servers then it probably makes sense to move the services closer together. Conversely, if the business servers process lots of data into small results then you can move the business logic closer to data. Additionally, the placement of application servers influences scalability, high availability, and security. However, because of the ease of scaling out and the low cost of Web servers, many e-commerce sites place application servers onto Web servers. This means the application services simply and efficiently inherit the scalability, high availability, and security of the Web servers. In a B2B environment the application server is separately maintained, both for scalability and security reasons.

Due to the heavy traffic seen by e-commerce servers and also due to the availability requirements, there will be a network of web servers instead of a single monolithic server at the front-end. This basically improves the scalability and fault-tolerance of the server to any bursts of busy traffic. Load balancers help increase the scalability of an e-commerce site. Load balancing works by distributing user requests among a group of servers that appear as single virtual server to the end user. Its main function is to forward user traffic to the most available or the "best" server that can provide a response to the user. Load balancers use sophisticated mechanisms to detect the best server. These mechanisms include finding the server with the least connections, the least load, or the fastest response times. They can also detect failed servers and automatically redirect users to the active servers. Ultimately, server load balancing helps maximize the use of servers and improves the response times to end users.

SSL, a user authentication protocol developed by Netscape using RSA Data Security's encryption technology. Many commerce transaction-oriented web sites that request credit card or personal information use SSL. The SSL off-loader typically decrypts all *https* requests arriving at the server. It should be noted that the link

between the front-end and the back-end servers is fully secure. So all the secure transactions are decoded by the security machine at the front-end before being sent to the back-end servers.

4.2.2 Back-end

The back-end servers mainly comprise of the database servers and the firewall which would protect sensitive data from being accessed by unauthorized clients. These firewalls provide security services through connection control. They are predominantly used when protecting mission-critical or sensitive data is of the utmost importance.

Because firewalls protect the most sensitive data, they play an important role in reaching the servers. Thus, firewalls are often implemented in pairs, whereby one is the active unit and the other is the standby unit. In the event of a failure of the active unit, the standby unit becomes operational. To ensure that connections to the application and database servers are maintained in the event of a failure of the firewall, firewalls must be able to perform stateful fail-over.

The database servers reside in the back-end of the network and house the data for e-commerce transactions as well as sensitive customer information. This is commonly referred to as the data services. Although Internet-based clients do not directly connect to these servers, the front-end Web servers initiate connections to these servers when a client conducts a series of actions such as logging in, checking inventory, or placing an order. Most e-commerce sites scale up their database servers for scalability and implement fail-over clustering for high availability. Partitioned databases, where segments of data are stored on separate database servers, are also used to enhance scalability and high availability in a scale-out fashion.

These servers can be designed by partitioning the customer information into a separate database and the catalog information into a different database. This kind of design decision will be made based on the amount of buying done at the site and

the overall response time experienced by the user. This division will also increase the scalability of the servers, if caching schemes for dynamic requests are adapted.

4.3 Configuration: B2C

A simplified configuration of the B2C site being used for the study is given in Figure 4.2. The site comprises ten web servers, each one powered by a Intel Quad P-III systems with a 512MB of RAM. The web servers run IIS 4.0 HTTP server. This cluster of web servers is supported by three image servers, each one powered by a Dual P-II system. As can be seen from the figure, the image servers serve both the database servers and the front-end web servers. For the purpose of our study, the image servers were considered to be in the back-end system. The product catalog server, connected to both the front-end and the back-end runs an NT 4.0 providing backup and SMTP services to the back-end servers. The LDAP server, connected to the back-end.

There are three different types of database servers present at the back-end, the customer database, the membership database and the catalog database. Only the customer database and catalog database are being used for this study. There is very minimal traffic coming to the membership database hence this was not used. Each of the databases, have NT 4.0, running SQL Ent. 7.0 SP1.

There are some other component of the site which are not shown in the figure. These are the components which will be used by specific hosts or are used for security or scalability of the site. It has to be observed that the B2C site is very similar to the generic e-commerce site model shown in 4.1. Since the user population that demand services from a B2C site is not well defined, the site is designed to sustain a variety of load conditions and user behaviors.

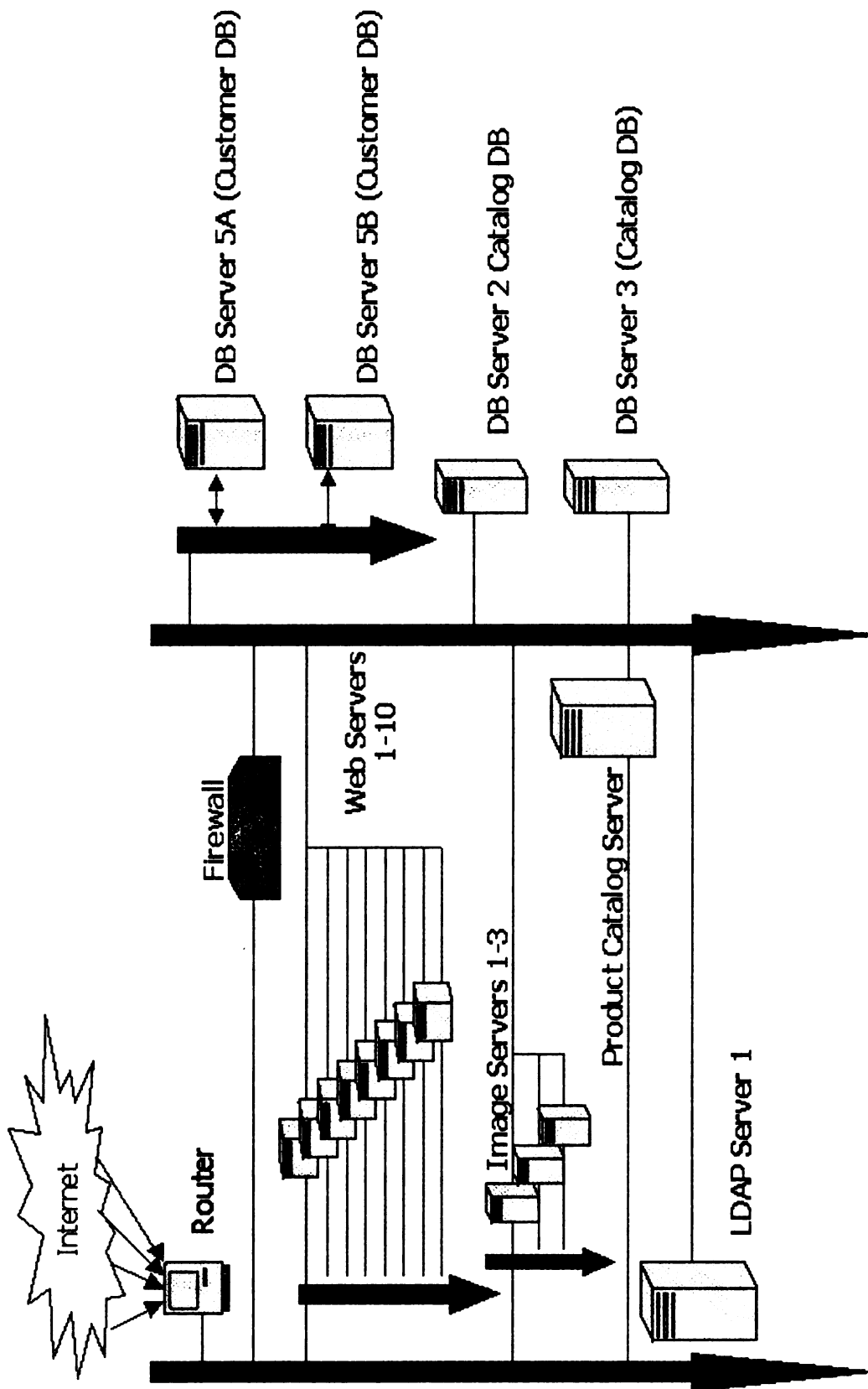


Figure 4.2: Simplified configuration of the B2C site

4.4 Configuration: B2B

In the B2B space, the design of e-commerce sites is completely different from their design in B2C space. Here the user population is known a-priori. The transactions being processed by each user arriving at the server is also known with reasonable bounds. B2B sites serve a limited population as opposed to B2C sites which aim at serving the entire Internet. These aspects enable the designers to customize the site to specific user requirements.

Scalability is one of the main issue that has to be taken care of when designing such customized system. So the design is done as a cluster of B2C sites, interconnected to form a large B2B portal. The interconnections between the individual B2C components in the site determine the user population to that site and also the services provided by that site. Figure 4.3 shows a simplified version of the B2B site being used for the study. Each of the web servers, can be individually used as a B2C site with its own database and network connection.

An important feature of B2B sites is the accessibility constraints on the users. Here the access to the web servers is restricted by Login/ Authentications machines which do load balancing and along with directing the traffic to the appropriate servers. Also in a typical corporation, there is a lot of internal traffic utilizing most of the services provided by their own B2B portal. These interactions can be allowed directly to enter the site without authentication, but instead of allocating any special resources, the interconnections are changed based on the load on different servers. Another important feature in a typical B2B site is the migration of the application servers from the web servers closer to the database servers. Because of independent authentication machines, traffic to the application servers can be sent directly without loading the web servers. Even though this would depend on the services provided by the site, application servers are separated due to their usage internally for training etc. and for promotional usage to other corporations.

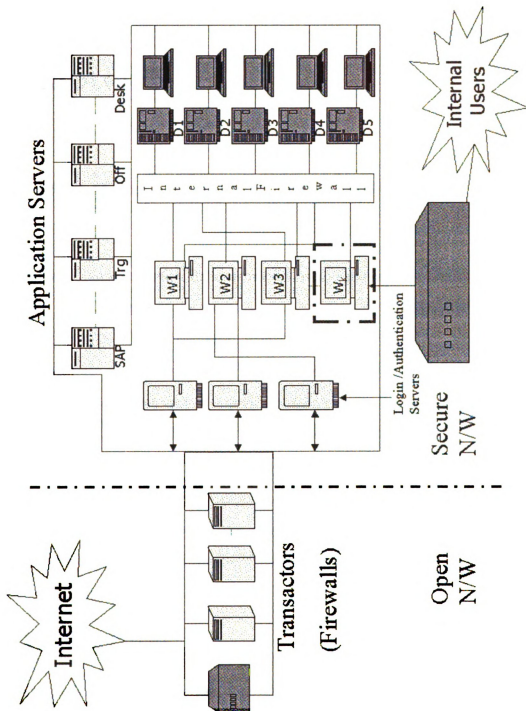


Figure 4.3: Simplified configuration of the B2B site

Chapter 5

Workload Characterization

Workload Characterization is the process of studying real-user environments, observing key characteristics of the workload and developing a workload model that can be used repeatedly. One of the most valuable benefits of workload characterization is the immediate perspective gained from a simple graph. Workload characterization provides the equivalent view of a network-scale drawing detailing the total bandwidth of the network, the significance of users, and the bandwidth requests of a particular workload. All this, displayed graphically, makes the transparent workload visible and meaningful, and can aid the process of system design.

In this chapter, the key features of Web workload are discussed along with main differences observed in E-commerce workloads over normal LAN or Server workloads.

5.1 Workload

Workload is defined as the set of all inputs the system receives from its environment. The composition of workload collected depends on the choice of the system boundaries, the goal for which the workload is being studied and the availability of representative data. A user process and the modules of operating system in charge

of resource management are generally considered part of the system. The system programs which assist the user upon request constitute the 'gray area' as their inclusion depends on the study being performed. A time frame is necessary to define a workload. Workload is defined over a day, month or any specific period of time. A simplifying assumption that the workload is insensitive to all changes in the system's performance is made. This assumption is made because influences on a user due to system modifications are unpredictable.

5.2 Principal characteristics of the Workload

In this section we describe the main characteristics of any good workload. Also we will see the relevance of the workload we are working with to the actual requirements for a workload.

- **Representativeness:** Workload for a full day period in a busy weekday is used, thereby increasing the representativeness of the data obtained. Representativeness again is another term used for accuracy, and we have made sure that we are looking at the data at a very high granularity thereby not losing any accuracy in representing the actual system.
- **Flexibility:** Possibility of easily and inexpensively modifying a model to reflect variations in the real workload. Even though we are not generating a model for the workload it has to be made sure that the tests are flexible for different kinds of workloads available.
- **Simplicity of construction:** The simplicity of construction of a workload-model implies that the cost and complexity of gathering information necessary to design and operate a model should be minimal. We have used time-tested statistical tests for proving the self-similar nature of the workload. The results can be repeated, using different workload under similar conditions.

- **Compactness:** It is related to the degree of detail and hence to the representativeness and usage costs of model. A very compact model is less detailed, less representative and cheaper to use than a less compact one. The workload is observed at different granularity to make sure we are not losing any information while going for a compact representation. It should be noted that most of the properties being tested are scaling properties and will vary with the scale at which we are looking at the workload.
- **Reproducibility:** Statistical tests are used for analyzing the behavior of the workload. Tests for self-similarity being used here(Chapter 2) are tested for similar workloads. Some of the assumptions in [5] are used to apply these tests to the new workload for further analysis. All these tests can be reproduced under similar conditions.

5.3 Data Collection

As described in Chapter 4, there are two major categories of e-commerce sites present, B2B and B2C. For this study data was collected from two different sites, one from each category. Data generated using synthetic techniques cannot be used to study behavior of e-commerce sites. Most of the established traffic generators like SURGE [28] generate representative web traffic, but there has not been any generators for e-commerce traffic. Due to this, data collected from e-commerce servers is used for the study. In this section different types of data collected, the duration of the observation period and the validity of the data collected are described.

5.3.1 Types of Data Collected

As discussed in Chapter 3, most of the work in this area has concentrated on the characteristics of the traffic arriving at the server. In this study we would like to

analyze the behavior of the server to changes in the behavior of the traffic. So data was collected at different levels in the system.

Access Logs

Web Server access logs from the the front-end and the back-end servers were collected. Data was collected at a granularity of 1 sec. This is an application level data giving the load on the httpd. This data will give the characteristics of the traffic arriving at the system. The average network bandwidth utilized and the file transfer rate can also be obtained from this data. Assuming a steady state operation of the system the network throughput of the system can also be calculated. Since the requests arriving at the servers are used as the basic unit, we cannot use the network bandwidth as a complete measure of the throughput of the system. But this measure is used to assess the load on the system.

Performance Monitor Logs

For the system level information, data was collected from the Performance logs [29] from all the servers present in the site. This data was collected at a granularity of 5 sec. This would give information about the I/O bandwidth used, the processor and disk utilization of the system et cetera. Since this is a software monitor being used for data collection the amount of information collected using this tool is very limited. Otherwise this tool itself would put load on the system thereby skewing the results obtained in the experiments. Data was collected at a constant rate of 5 sec intervals. So this data is at a higher scale than the logs from the web servers. But both the scales are below the Non-Stationarity time scale used for the analysis.

5.3.2 Duration of Data

Data was collected at the server and the performance monitor for an entire day. A weekday is used for data collection since this would represent normal traffic. Data for a five day period was used to study the average behavior of the traffic over a long period of time. But this data would not be helpful in looking at specific load periods by averaging. Due to the high variability in the arrival rate, we have to average the data over very short periods to obtain any meaningful information. So data collected over one day was used for analysis of the data under different burstyness levels. Diurnal nature of the data was studied for different levels of burstyness observed at the server. For this analysis, data was used from the access logs.

5.4 Characteristics of the Workload

In this section, the main characteristics of the workload being studied are described. [8, 23, 22] reported the characteristics of web workloads. In this work we are looking at the subset of web workload comprising of e-commerce traffic. The main differences between web and e-commerce workload are discussed below.

5.4.1 OLTP Transactions

Heavy presence of Online Transaction Processing activity is observed among the transactions taking place at the server. This is due to the heavy database transactions accruing for every request from the user. Due to security reasons most of the data is present in the database server which is protected by a secure firewall. This prevents the web server from responding to most of the requests without sending a query to the back-end server. In the B2C site that we were looking at, it is observed that all the requests had their responses coming from the back-end servers. Here even though the cache engines are present, the responses are sent from the back-end servers only.

In most of the sites, all the requests are converted into dynamic files with the filename as the argument. The actual file, present at the back-end database is obtained as a response to a query sent with the filename as the argument. This increased back-end activity makes the response times more dependent on the file sizes. One of the assumptions that can be made about e-commerce traffic is that the file size is a good representation of the response time. More about this will be discussed in later chapters.

5.4.2 Secure Transactions

Along with the database activity there will be a large percentage of requests coming in secure mode. Even though B2C traffic has lesser secure traffic, B2B sites experience almost complete secure traffic from users. This is due to the heavy security constraints present in industry to industry transactions. Increased amount of secure transactions implies heavy processing at the front-end server. Most of the sites have SSL off-loaders, which do encryption/decryption of requests to reduce the load on the system. This process adds to the response time. Aggregating these transactions with normal transactions increases the variability in the response times observed by the user. So the response time observed at any instance for a file even though it is a static html file will not be representative of the actual response time for that file.

5.4.3 Dynamic Transactions

The amount of Dynamic transactions has been increasing in the Internet traffic at a steady pace. SPECweb96 [30] did not have any dynamic transactions even though it was agreed to be a very good representation of web workload. SPECweb99 [31] had 35% dynamic requests. Both these commercial benchmarks did not consider the advent of e-commerce transactions in the WWW. TPC-W [32] is a transactional web

benchmark. The workload is performed in a controlled Internet commerce environment that simulates the activities of a business oriented transactional web server. This benchmark has almost all the requests generated dynamically by the server. Increased amount of dynamic requests reduce the overall characteristics of the traffic.

An understanding of the configuration of the sites being studied and the issues which make the traffic more complex than normal web space is provided in this chapter. The analysis done to find the reasons behind this complex behavior is given in the next chapter. Statistical tests discussed in Chapter 2 are used to investigate the characteristics of the different components of the server and its impact on the traffic as it passes through.

Chapter 6

Front-End Characterization

It is observed that traffic in e-commerce space is much more complex than the normal web traffic. The reasons for this are explained in Chapter 5. A visual inspection reveals the workload at e-commerce sites to be more bursty than normal web workload. To study this behavior a set of parameters were used, which would have the maximum impact on the behavior of the traffic. It would not be feasible to study all the parameters affecting the traffic, so the following parameters were considered for studying the behavior of the servers.

- Arrival process
- Utilization of the server
- Response time
- Request file sizes
- Response file sizes

Before designing any system a capacity planning study is done to decide the normal working parameters of the system [33]. The list above is a good set of parameters to

start with for designing an e-commerce server. The problem with a capacity planning study for an e-commerce site is that the normal parameters are orders of magnitude lower than the parameters observed at high load periods. For a Poisson arrival process this effect can be smoothed out by observing the system for longer periods of time, or by looking at the system at higher granularity. But as explained in the previous chapter, e-commerce workload is highly bursty in nature and cannot be smoothed out even at orders of magnitude of increase in granularity. In this chapter these effects will be investigated for the front-end web server. Each of the above parameters will be looked at individually at different scales to find their impact on the overall system.

6.1 Arrival Process

In Figure 6.1 a Poisson arrival process is shown with an average arrival rate of 15 reqs/sec. The data is collected over a period of 1000 seconds. Figure 6.2 shows the same arrival process but at a higher amount of granularity. This process is aggregated at 10 second intervals. It can be seen that the arrival process smooths out after increasing the scale by 10. The maximum variance observed comes down to 2.5 from 20, observed in the original time-series. This luxury of traffic smoothing out will not occur if the arrival process is a combination of long-range dependent on/off processes.

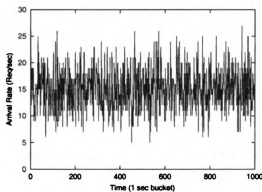


Figure 6.1: Poisson arrival process with 15 req/sec, 1 sec granularity

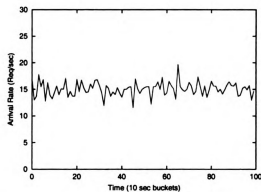


Figure 6.2: Poisson arrival process with 15 req/sec, 10 sec granularity

Figures 6.3, 6.4 show the arrival process at the B2C and B2B e-commerce sites. The data shows traffic on a normal weekday with an average arrival rate of 0.65 requests/sec at the front-end web server for the B2C site and a around 1 req/sec arrival rate at one of the web servers in the B2B site. A visual inspection reveals the burstiness in the arrival process. The B2C server is a 4P system with an average processor utilization of 6% per processor and disk utilization of 2% during the period starting from 9.00am till 6.00pm. The low utilization is typical of e-commerce sites since they are designed for much higher load and sustain a very minimal load during normal working periods. It is the high load periods showing bursts of orders of magnitude more than normal operating parameters which cause concern for better capacity planning and performance analysis of these systems.

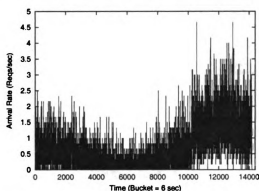


Figure 6.3: Arrival process at B2C site, 6 sec granularity

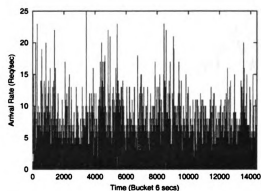


Figure 6.4: Arrival process at B2B site, 6 sec granularity

Figures 6.3, 6.4 show that the sites have distinct high and low load periods during the course of a day. For the B2C site, busy period starts around 6:00pm in the evening and ends at around 11:00pm in the night. Since this is a B2C site serving general consumers, the traffic is heavy during the after-office periods. Distinctive low periods during the morning between 7:30am to 11:30am can also be observed. In case of the B2B site, the traffic concentration lies mostly during normal office hours, between 9:00am to 8:00pm, which is intuitive. It should be noticed that the graphs show aggregated arrival traffic for the B2B site and the averaged arrival process for the B2C site.

In Figures 6.5, 6.6 the arrival process at the B2C site is shown at higher granularities. In this study higher granularity means that the granularity is more coarse. This figure plots $X^m(t)$ for $m = 60, 600$ where $X(t)$ is the original time-series of arrivals. As the value of m increases, the granularity at which the data is looked at increases. It can be observed that the traffic burstiness does not reduce even when the granularity is increased by three orders of magnitude. The maximum variability observed remains almost a constant (≈ 14) for the different scales. This shows unmistakable presence of self-similarity in the arrival process. In [5] the authors conjectured the presence of non-stationarity at higher time-scales (900 secs) for e-commerce traffic. So the arrival process is tested for self-similarity at much lower time-scales.

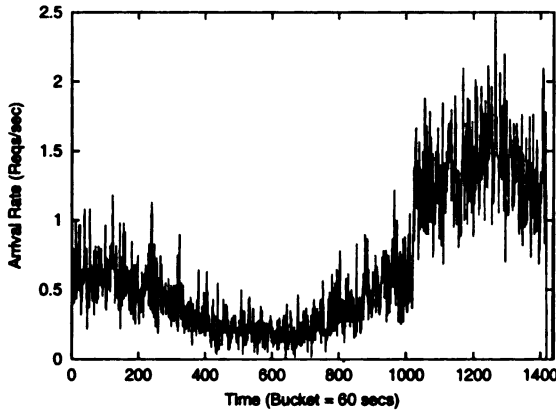


Figure 6.5: Arrival process at 60 sec granularity

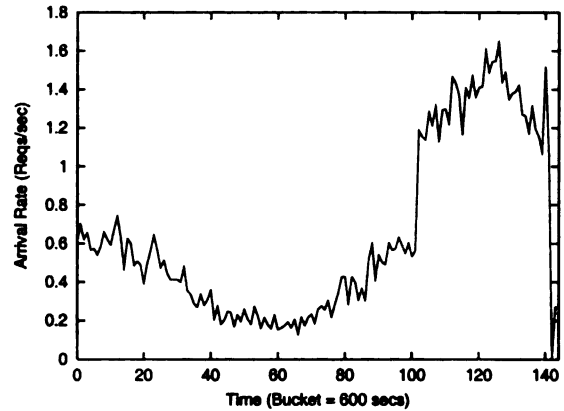


Figure 6.6: Arrival process at 600 sec granularity

The Arby-Veitch (AV) [34] estimator test was used for estimating Hurst-parameter for the arrival time-series. This is known to be a reliable test for workloads with busy periods showing a non-stationary behavior. E-commerce workload is influenced by busy periods caused by different sales promotions and seasonality. In [35] the authors show that this test works well for such a workload. For a detailed discussion on the AV-wavelet based estimator, please refer [34]. Hurst parameter is also calculated using the R/S plot¹ test. Reliability of this test under low time-scales for e-commerce

¹Please refer Chapter 2 for details

traffic is tested by comparing the H-parameters obtained using the two methods. Figure 6.7 used the arrival time-series at the B2C site with a granularity of 3 sec.

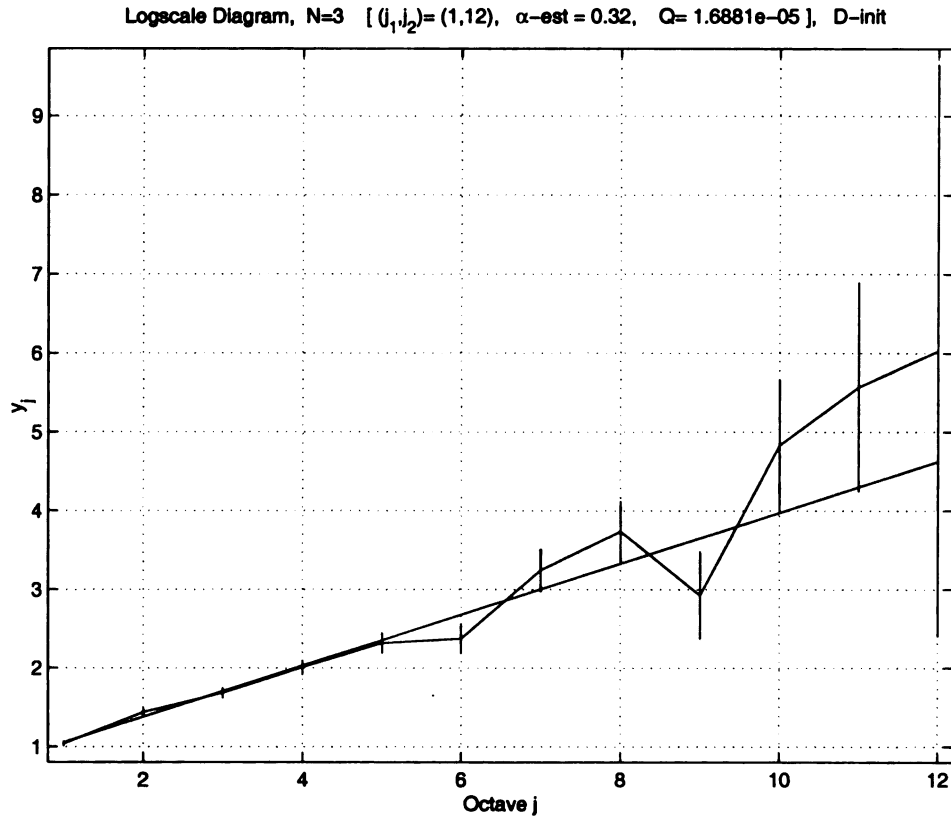


Figure 6.7: A-V estimator test for self-similarity for arrival process

The Hurst parameter is estimated to be 0.662. This shows that the arrival process at the B2C site is *self-similar* in nature.

In figures 6.8, 6.9 the log-log plot of the rescaled range ratio is shown for the B2C and B2B arrival traffic respectively. The Hurst parameter is estimated to be 0.662 using a linear-regression line through the R/S points for the B2C site, which matches the estimation made by the AV-estimator. Similar test was done for the arrival traffic at the B2B site. Using the AV-estimator the H-parameter was estimated at 0.69, whereas the R/S plot gave an estimate of 0.70 for the H-parameter. Here also it is observed that the R/S test gives a good approximation of the H-parameter. This shows that the Rescaled Range test gives a good approximation of the Hurst-parameter at lower time-scales. From the above results it can be conjectured that

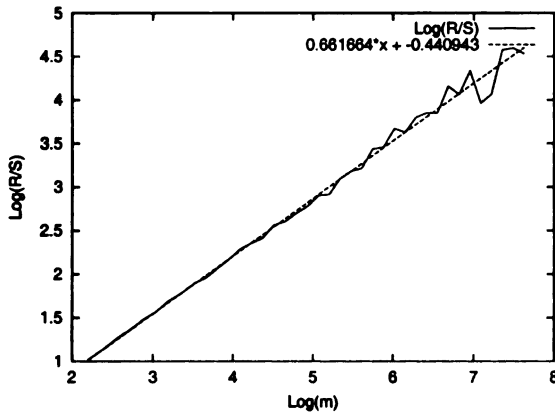


Figure 6.8: R/S plot test for self-similarity for arrival process at B2C site

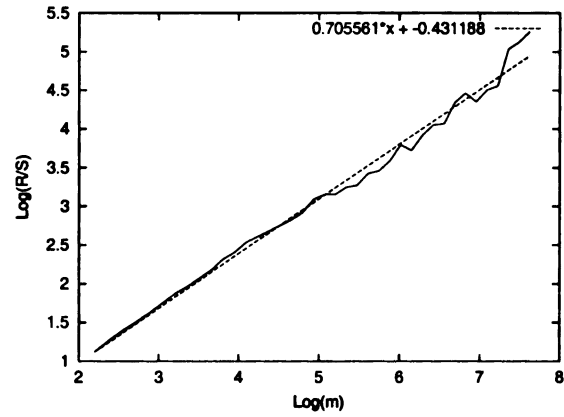


Figure 6.9: R/S plot test for self-similarity for arrival process at B2B site

the arrival series does not show any non-stationary behavior at lower time-scales. Also the R/S statistic can be used for the other parameters for estimating the Hurst parameter.

6.2 Processor Utilization

Figures 6.10, 6.11 show the %Utilization of the front-end web server for the B2C and B2B sites respectively. As explained earlier, the data is collected between 9:00am till 5:00pm at a granularity of 5 secs for the B2C site. For the B2B site the data represents the activity between 10:00 am in the morning till 9:30 am the next day morning. The B2C server sustains a constant load throughout the day, with an average load of 7% on each of the four processors. High and low load periods can be observed on the B2C server during the course of the day. This behavior is absent in the B2B server. This is due to the a-priori knowledge of the transactions and load from users in the B2B space. B2B sites are customized for specific traffic patterns and a normal traffic would not affect the load on the system to a higher degree. Thus the load on the system appears almost constant even though there is a variation in the arrival rate at the server. The time-series obtained from the utilization was also tested for self-similar

behavior. The AV-wavelet based test and the R/S plot test are used for estimating the h-parameter. The estimated Hurst parameter is 0.755 using the AV estimator, and 0.77 using the R/S plot test for the B2C site. In the B2B space, the load on the system did not have a high degree of self-similarity. The H-parameter is estimated to be 0.66 using both the AV-estimator and the R/S plot test. Due to a balanced load on the B2B system throughout the duration, the degree of self-similarity is very low. The effect of the arrival process is not seen in the overall load sustained by the B2B server. A higher H-parameter implies an increased degree of self-similarity. Utilization is a factor of the response-time and the arrival process. The inherent burstiness in the arrival process is already established in the previous section. So the service time distribution is observed. Since the h-parameter for the utilization is more than that for the arrival process, it is assumed that the service-time is long-range dependent with a heavy tailed distribution.

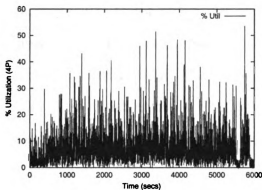


Figure 6.10: % Utilization at the front-end web server (4P), B2C

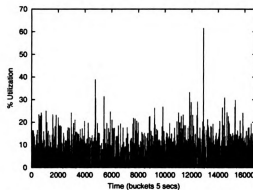


Figure 6.11: % Utilization at the front-end web server (2P), B2B

6.3 Response Time

In figure 6.14 the response time observed by the users over the entire day period is shown for the B2C site. Previous studies [8, 36] have concentrated on the study

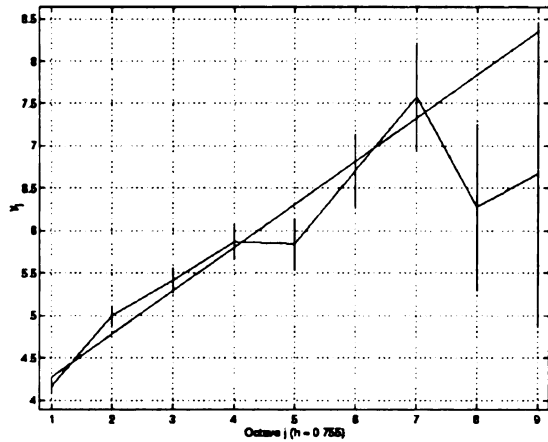


Figure 6.12: AV estimator for the front-end B2C web server(4P)

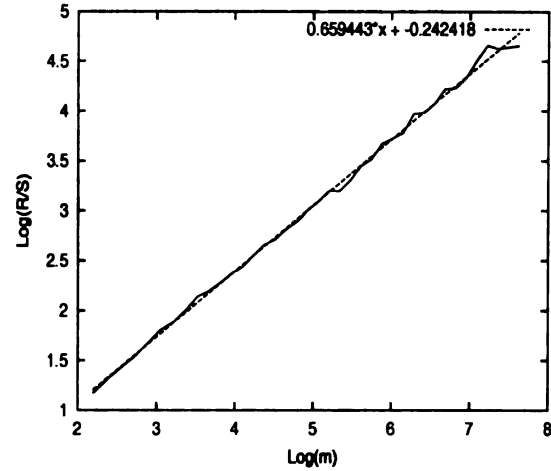


Figure 6.13: R/S estimator for front-end B2B server (2P)

of the heavy-tailed behavior of web response times. In this work the response-time distribution is converted into a time-series by aggregating the response-times seen for non-overlapping intervals of 5 secs. Even though the times seen are not the actual response times observed by the user, they can be used for time-series analysis. Only a multiplicative factor of 1/5 will be required to get the actual response-times. The time-series obtained is checked for self-similarity and any non-stationary behavior. The AV test and R/S plot test are used for estimating the h-parameter. As explained earlier, a good estimation of H-parameter is obtained using R/S test only when the time-series is stationary. So both the tests are used for estimating the h-parameter.

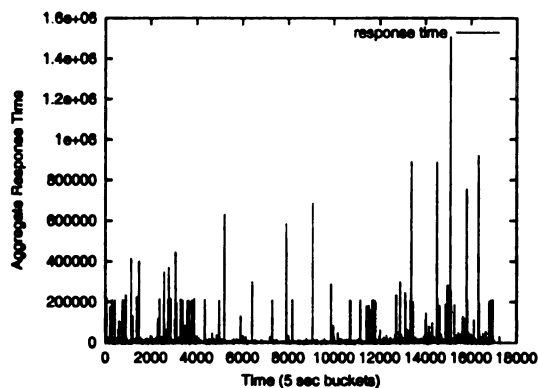


Figure 6.14: Aggregated response time at the front-end web server (4P)

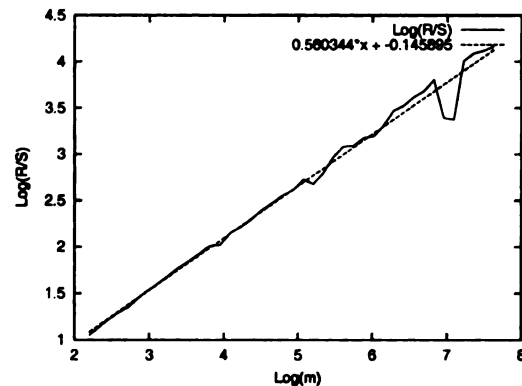


Figure 6.15: R/S test for estimating h-parameter for response time (h = 0.56)

Response time is one of the very important performance metrics in the design and analysis of any server system. High burstiness in the arrival traffic implies saturating server queues, leading to response times going up. Studies have shown that 90 percentile response-times can be used for predicting the mean response-time [3] for QoS predictions. This measure cannot be used with high burstiness present in the response-time distribution. In this study we would like to investigate techniques for reducing this burstiness in the response-time distribution. Figure 6.14 shows response times orders of magnitude higher during the high load periods in the evening. Comparing this graph with the arrival process shown in figure 6.3, unmistakable correlation can be found between the different load periods. Even though the utilization of the system does not get effected, buffer queue lengths increase thereby increasing the user perceived response times. Increased burstiness impacts the overall response time of the system to a higher extent than the arrival process. This burstiness in the response time is a factor of the back-end data retrieval time and the server processing time. So the increased burstiness can be injected either by a heavy-tailed distribution of file sizes causing a bursty transfer time or by the burstiness in the back-end service time. Further discussion on the back-end system will be continued in the next chapter.

The response time distribution is studied for the presence of heavy-tailed behavior. In [8] the authors established the heavy-tailed behavior of web response times. A random variable following a heavy-tailed distribution can take on extremely large values with non-negligible probability. Thus heavy-tailed distribution of web transmission times implies that the users can observe response times orders of magnitude higher than normal response times during a busy period. The authors in [8] used Log-Log cumulative distribution plots (LLCD) to estimate the tail weight of web transmission times. Similar method was used to model the response time behavior for the B2C site. In figure 6.17 the LLCD plot of the response-time distribution is shown. This figure shows that for values greater than 2, the distribution is nearly linear indicating

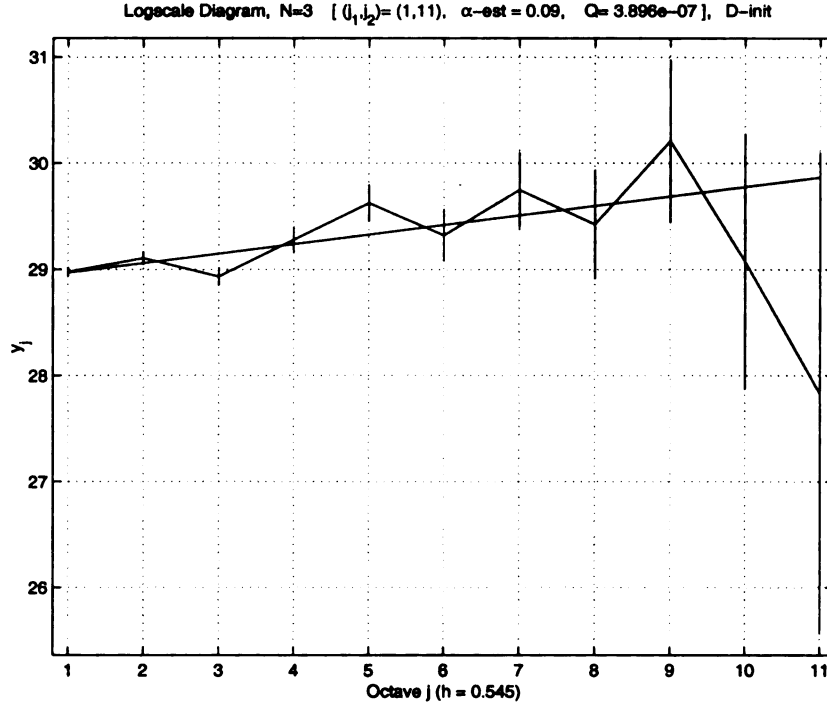


Figure 6.16: AV estimator for the front-end web server response time (B2C)

a hyperbolic tail. A least-squares fit was made for data points more than 2 giving a slope of -1.55^2 as shown in Figure 6.18. This indicates that $\alpha = 1.55$. This shows that the transmission times are in fact heavy-tailed and can be modeled using a Pareto distribution with $\alpha = 1.55$. Similar result is found with the distribution of response-times in the B2B space. The distribution is found to be heavy-tailed with an $\alpha = 1.58$, for file transfers greater than 1000 secs.

In [8] the authors showed that the distribution of web transfer times over different sets of data is heavy-tailed with $\alpha = 1.21$. The tail weight appears to be reduced in e-commerce environment. The reduction in tail weight could either be a characteristic of the dataset being used or the inherent behavior of e-commerce traffic. The distribution is found to be heavy-tailed for file transfers of more than 1000 secs. This means that the response-sizes are in controllable limits till around 1000 sec response times. Response-times more than 1000 secs in B2B environment may not result in

²The $R^2 \geq 0.95$ for all least-square fits

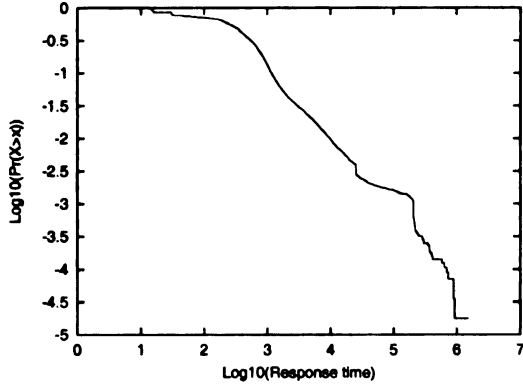


Figure 6.17: LLCD of response-time distribution at the front-end

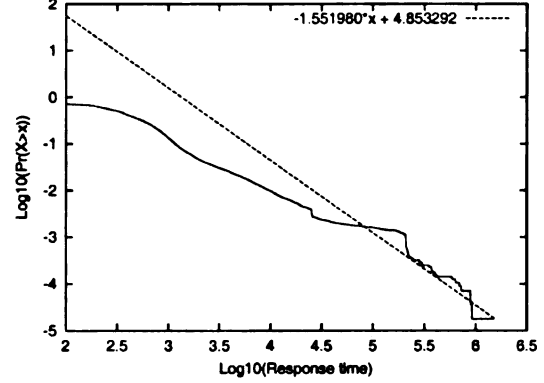


Figure 6.18: Estimated tail weight for the response-time distribution

high revenues so the response-time in B2B is assumed to be under controllable limits even at higher load periods.

The heavy-tailed behavior of response times in web-environment is attributed to the distribution of transfer sizes. This has been known to follow the distribution of file-sizes in UNIX environment. Request and response size distributions at the B2C server are investigated in the next sections. The response-time distribution will be discussed again after a discussion on the transfer sizes.

6.4 Request/Response file sizes

The request and response file sizes in UNIX [37, 38] and web environment [8] have been studied previously. It was observed that these distributions show a heavy-tailed behavior with a tail weight of approximately $\alpha = 1.06$ for file-sizes greater than 1000 bytes [8]. This was considered one of the main reasons for the heavy-tailed behavior of the web response times. In e-commerce environment, it has already been shown that transfer times have a heavy tailed behavior with $\alpha = 1.55$. In this section the behavior of transfer size distribution is studied. Figures 6.19, 6.20 show the request and response size distribution over the observation period at the B2C server.

It can be observed that the distribution of transfer sizes is fairly constant in the

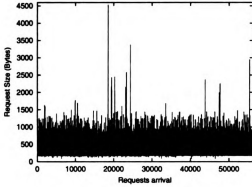


Figure 6.19: Request size distribution over time

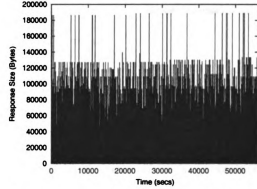


Figure 6.20: Response size distribution over time

B2C environment. A visual inspection rules out the possibility of heavy burstiness in the aggregated time-series obtained from the transfer sizes. The distribution of request sizes is further investigated for heavy-tailed behavior using LLCD plots.

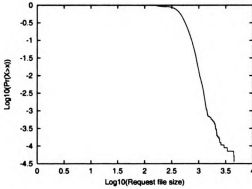


Figure 6.21: LLCD of request size distribution

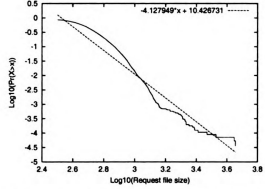


Figure 6.22: Estimated tail weight of request-size distribution

Figure 6.21 shows the log-scale plot of the cumulative probability function over the different request sizes observed. The plot appears linear after $x > 2.5$. A linear-regression fit to the points for requests more than 320 Bytes gives a line with slope $\alpha = -4.12$ ($R^2 = 0.947$). The linear fit can be seen in figure 6.22. This gives an estimate of $\alpha = 4.12$ thereby indicating that the request size distribution is not heavy-tailed in nature. This result refutes the previous results about web traffic. In [8] the authors found that the requests also follow a heavy tail distribution with $\alpha = 1.16$.

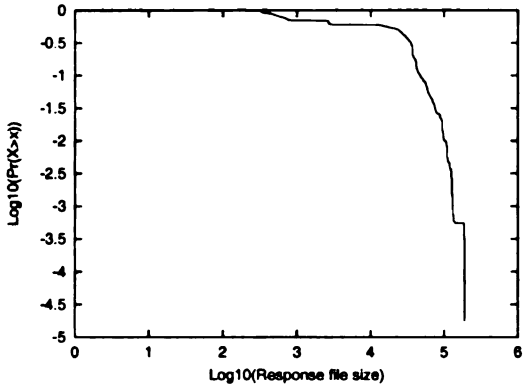


Figure 6.23: LLCD of response size distribution

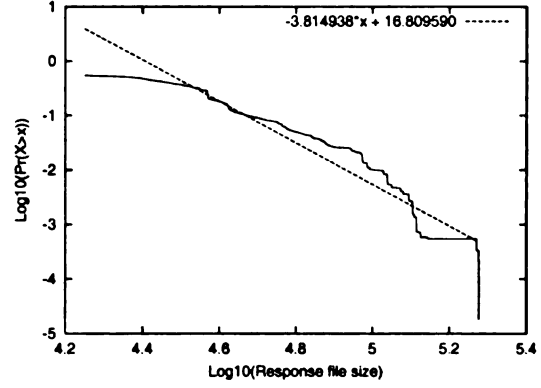


Figure 6.24: Estimated tail weight of response-size distribution

In Figures 6.23, 6.24 the LLCD plot of the response file sizes sent by the server can be seen along with the linear fit obtained for the LLCD plot. The plot appears linear after $x > 4.25$. A linear fit to this part of the plot gives an $\alpha = 3.81$ ($R^2 = 0.92$). This indicates that the response file sizes do not follow a heavy tailed distribution.

6.5 Performance Implications

Previous studies on web traffic and LAN traffic have attributed the self-similar behavior of network traffic to the aggregation of long-range dependent ON/OFF processes. In E-commerce space, the response-times are found to be heavy-tailed in nature even though the request and response file sizes are almost a constant. The heavy-tailed behavior of response-times in web environment was believed to be caused by the heavy-tailed behavior of the file transfer sizes in the web environment. Studies in UNIX file-systems [37, 38] have also indicated the same behavior even though the actual measurements were not made. In e-commerce environment, the transfer sizes do not follow a heavy-tailed distribution as shown earlier in this section. Heavy-tailed behavior of web transfer sizes are fundamentally caused by the inclusion of image and video files in the overall traffic. Since these files are minimized in e-commerce environment (for reducing the overhead in response times), the behavior of the transfer

sizes becomes somewhat intuitive. The lack of large image and video files removes the heavy-tailed nature of e-commerce traffic.

But it is observed that the response time is still showing a heavy-tailed behavior in both B2C and B2B space. As explained earlier this implies that the user perceived response-time can increase by orders of magnitude under load conditions. Due to the critical nature of e-commerce applications and also the business model (increasing criticality with the increase in load), it is imperative that the response-times are kept under normal bounds even in high load conditions. In e-commerce environment response-time is dependent on the processing time and the transfer time. Since the file-sizes do not follow a heavy-tailed distribution, it can be safely assumed that the transfer time does not contribute to the variation in the response-time. This shows that the characteristic of the processing time is affecting the response-time to a higher extent than the response size. Also the effect of file-sizes appears to be negligible on the end-end response-times observed. This result contradicts the behavior of response-times for normal web traffic where the response-size of files can be assumed as a good approximation of the response-time. The difference is that, in web environment the transfer times consumes most portion of the response-time which is not the case in e-commerce environment due to the different composition of requests. Due to the presence of OLTP type of transactions, it can be assumed that the processing time consumes the major portion of the response time. This is decreasing the dependence of the file sizes on the response time.

With the absence of a major impact in response-time due to the file sizes, the variation in the response-time is attributed to the variation in the response-time seen at the back-end servers. This aspect will be further investigated as part of the back-end analyses.

6.6 Variation in H-parameter

Data for a full day period is being used for this study. As described earlier, there are different load periods during the day. These load periods are caused by the user behavior, thereby showing different values of burstiness in the traffic arriving at the server. These levels of burstiness are studied along with their impact on the overall system response-time. The results obtained from this analysis are described in this section.

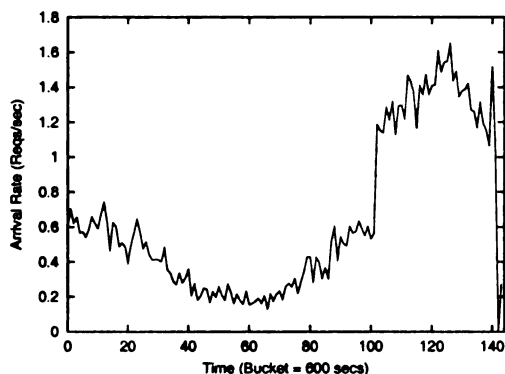


Figure 6.25: Arrival Process at the Front-end server

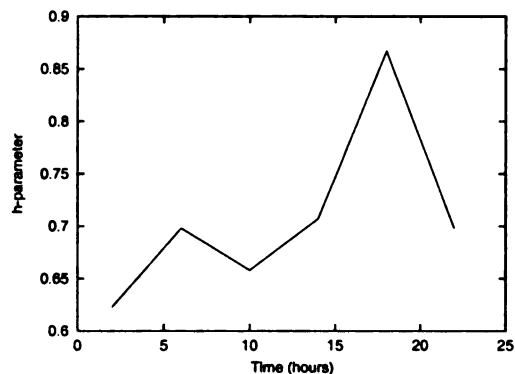


Figure 6.26: Variation in H-parameter over 24 hours

Figure 6.26 shows the variation of H-parameter over the entire day period. The data was divided into blocks of size 4 hours. It can be observed that the variation in H-parameter closely follows the variation in the arrival rate of the requests shown in figure 6.25. Correlation of this variation with the average response time was seen for the same period. Though the average value of response-time will not be representative for highly bursty periods, that value can be used for low time scales with shorter durations. The average response-time did not show any correlation with the variation in the H-parameter. This lack of correlation with the response-time shows that the impact of burstiness does not reduce even at low traffic demand. This has been a concern in previous studies on web environment. High queuing times occurring in low load periods might not effect the over all response-time, but this behavior will

move the bottleneck in system from the processors to the Queues, which are difficult to manage.

6.7 Summary

Traffic arriving at the front-end of two sites one B2C and the other a B2B e-commerce site is characterized and the results are shown in this chapter. The presence of self-similarity in the arrival process, processor utilization, response-time is shown using an estimate of the H-parameter. The H-parameter was obtained using the AV-wavelet based estimator. The R/S test, used at lower time-scales, is also shown to be a good estimate of the H-parameter in comparison to the AV-wavelet based estimator, only it can be used reliably at low time-scales. So the presence of Non-Stationarity is eliminated at lower time-scales. Implications of the presence of self-similarity in the different parameters was discussed. The heavy-tailed nature of the response-time was established for e-commerce environment, but it was also shown that the transfer sizes do not follow a heavy-tailed distribution as observed in web environment. This raises questions about the reasons for heavy-tailed behavior of the response-times, which were attributed to the variation in the service-time requirements of the different queries at the server. Diurnal variation in the h-parameter (level of burstiness) was studied for the B2C front-end data. It is observed that the level of burstiness does not effect the average response-time seen and the impact of burstiness on buffering is not reduced under low traffic demand conditions.

Chapter 7

Back-End Characterization

The most important and sensitive information in E-commerce servers is kept in the back-end servers. It is the back-end servers that perform the business logic for the e-commerce site and are hence the most crucial components of any e-commerce server. In this chapter the characterization of the behavior of the back-end servers is discussed. The parameters used for doing the characterization depend mostly on the configuration of the site and the purpose of the individual components [39] in the back-end. As described earlier, the composition of back-end servers is closely dictated by the business model of the site. So different parameters might be interesting for different sites. In this study the following parameters are used for studying the characteristics of the two sites.

- Processor utilization
- Disk accesses

In the B2C site there are four different servers at the back-end¹. These are:

- Main database server

¹Please refer the chapter on E-commerce for a complete discussion on the configuration of the B2C site

- Customer database server
- Image server
- LDAP server

The image server and LDAP server are not heavily loaded during the observation period. There is a single burst of traffic to and from these servers when the data is updated daily. This burst is also seen in other back-end databases and will be discussed in detail later in this chapter. The only servers that experience a sustained load throughout the day are the customer database and the main database. These two servers are used for studying the characteristics of the back-end system.

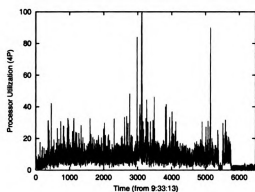


Figure 7.1: Processor utilization of Catalog Server (5secs)

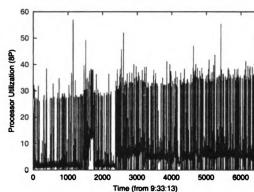


Figure 7.2: Processor utilization of the Main D/B server (5secs)

7.1 Processor Utilization

In figures 7.1, 7.2 the processor utilization of the two back-end servers in the B2C site is shown. It can be observed that the back-end server experiences a sustained load of 10% on average over the entire period. There is a visible peak of almost 100% utilization of the catalog server. This will be discussed later in the section. For the Main D/B server, the utilization remains at around 30% for most of the observation period. This shows that the load on back-end servers is higher than on the front-end

servers, when compared with figure 6.10. Previous studies have speculated that the load on the back-end servers is more regulated due to the presence of the front-end server. One of the reasons for this speculation is the service time of the front-end server. This either causes a delay or reduces peak of any burst reaching the back-end servers. This behavior of the back-end servers is investigated by looking at the time-series obtained from the utilization of the servers. H-parameter values of 0.87

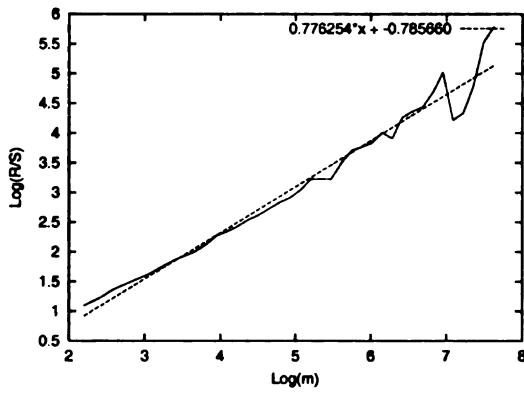


Figure 7.3: Estimation of H-parameter for Catalog Server ($H = 0.77$)

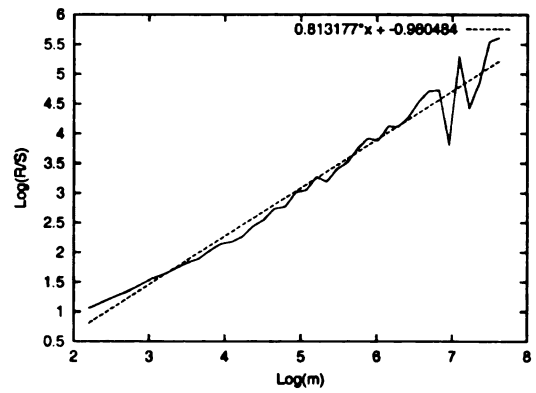


Figure 7.4: Estimation of H-parameter for Main D/B server ($H = 0.87$)

and 0.77 were obtained for the utilization of the main database server and the catalog server respectively. The burstiness observed at the back-end servers is more than the front-end servers ($H = 0.77$). Similar results have been observed in the B2B space also. The utilization of the database server of the B2B site is shown in Figure 7.5. It can be observed that the load on the system reaches 100% around the 4000th bucket. This is the updation activity which takes place in most e-commerce sites. The actual time when this takes place is around 1.00pm in the night. Similar activity can be seen in the other back-end servers, but nothing can be observed at the front-end servers, as the bulk of the data which needs any maintenance is present in the back-end servers only. Figure 7.6 shows the Hurst parameter estimation for the utilization time-series of the database server. The back-end server in B2B space is also found to be more bursty than the front-end traffic. This contradicts previous assumptions

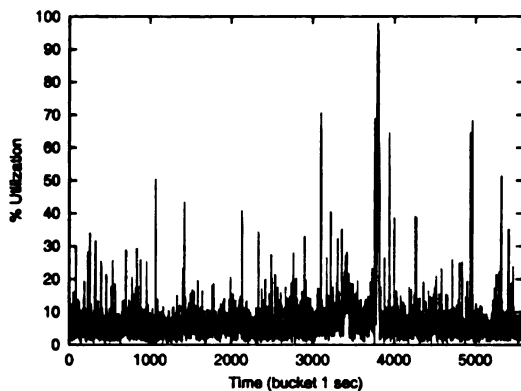


Figure 7.5: % Utilization of the B2B back-end server

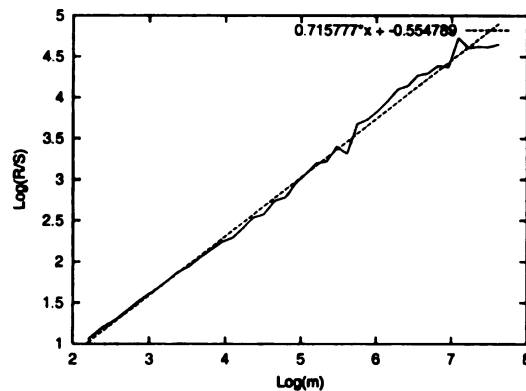


Figure 7.6: H-parameter for the B2B database server ($H = 0.72$)

about burstiness at the back-end servers in web environment. This aspect is further investigated.

Utilization is a factor of arrival process and the service requirements of the requests [40]. Figure 7.7 shows a queuing model shoeing simplified execution of a request at an e-commerce server. Obtaining data for the arrival process at the back-end servers is difficult since data of this nature is not collected normally in commercial databases. So the arrival process at the back-end is approximated from the front-end. Arrival process at the back-end, Q_2 , is a factor of the arrival process at the front-end and the service requirements at the front-end, S_1 . The front-end service time is difficult to estimate from the access logs. The service-time information in the logs will indicate the time taken at the entire site. It can be assumed that the service-time for requests at the front-end will be a constant, causing a delay in the requests before being transmitted to the back-end. There is no processing involved at the front-end for database queries. Since most of the requests arriving at the site are database queries we can assume that this will incorporate a delay in the requests arriving at the front-end. So the arrival process at the main back-end database will only be a delayed process of the front-end arrivals.

In the above speculation the effect of embedded requests from the front-end has not been considered. Without the effect of embedded traffic from the front-end it can

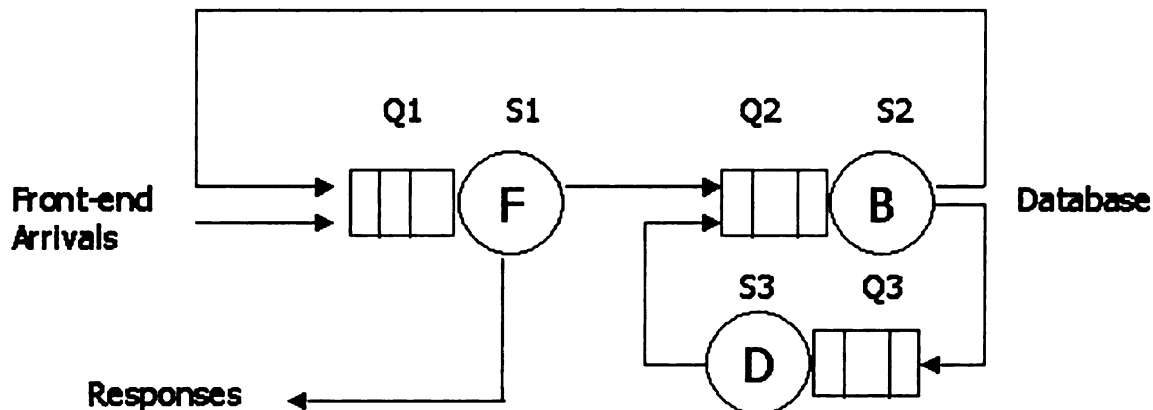


Figure 7.7: Queuing model for a simplified e-commerce request execution

be assumed that the back-end arrival is a delayed process of the front-end arrival. Given this the self-similar behavior of the back-end utilization can be explained to some extent. But, since the H-parameter is more at the back-end, the service time at the back-end, S2 should also be fractal in nature, rendering highly bursty utilization. Since the requests and response sizes do not follow a heavy-tailed distribution, the service time at the back-end server is a factor of the fractal nature of the query processing time. The most dominant component of simple query processing done at e-commerce servers will be the access to the files being requested. This would involve buffer access or disk access depending on the availability of the files in the cache. In the next section, the access pattern of requests to the buffer cache and the disk are discussed.

7.2 Disk Accesses

The accesses to the disk are filtered by the buffer cache [40]. Caching at the back-end servers is very important since it reduces the overall response-time seen by the users. In this section the effectiveness of the buffer cache in B2C site will be discussed. The B2C site has four disks for the Main DB system. Disk accesses are used for the study instead of disk utilization. Reliable data could not be obtained for the disk utilization

due to the presence of a cluster of four disks.

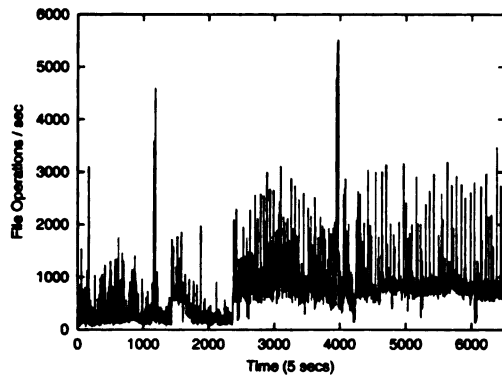


Figure 7.8: File Operations per second from Main DB server (5sec)

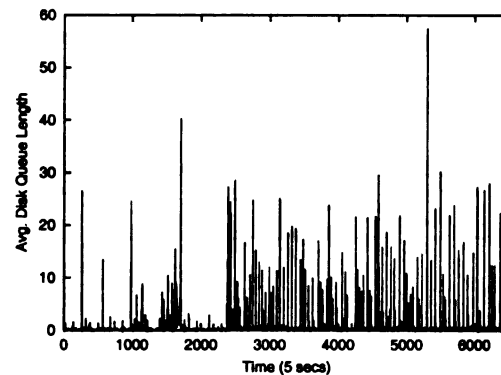


Figure 7.9: Disk Queue Length at Main DB server (5sec)

Figures 7.8, shows the distribution of the file request rate at the Main DB server. This shows the arrival rate of file requests seen by the four hard disks. Figure 7.9 shows the average queue length seen by the hard disks at the Main DB server. The average queue length is found to be self-similar in nature with $H = 0.77$. This would result in a heavy-tailed behavior in the average response-time of the hard disk. The reason for the burstiness in the queue length can be attributed to the arrival of file transfers at the hard disk. This rate is also found to be bursty in nature with $H = 0.83$. The buffer cache does not appear to be effective since the hard disk is experiencing requests at this level of burstiness.

In the previous chapter, the response-time at the front-end is found to be heavy-tailed in nature even though the request and response size did not follow this distribution. The burstiness in the service time at the back-end was attributed to this behavior. Here it can be seen that the heavy-tailed distribution of response-time at the back-end is due to the bursty arrival process to the hard disks, causing the queue length to be bursty. This high burstiness in queue length will remove the effect file sizes may have on the transfer times. This conclusion also supports the previous speculation that file-sizes were not a good representation of response-times in e-commerce environment.

7.3 Back-end Activity

Data at the back-end being highly volatile in nature has to be updated with the data at the cache machines and the front-end servers. This is done in a short period of time, such that there is no effect on the actual user population visiting the site.

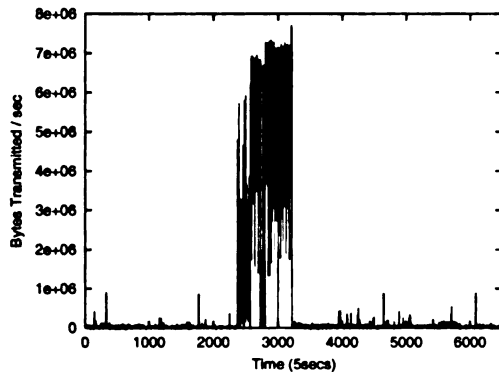


Figure 7.10: Bytes transmitted per second from Main DB server (5sec)

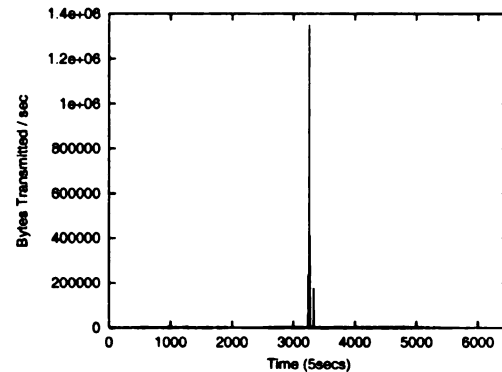


Figure 7.11: Bytes transmitted per second from Catalog server (5sec)

Figures 7.10, 7.11 show the bytes transmitted by the back-end servers per second. A high burst of traffic can be observed between 1pm to 2pm. The front-end traffic did not show any signs of this activity going out of the server. In the absence of any traffic going out of the server, it is assumed that this is an update or logging operation occurring internally in the server. It has to be noticed that the database server did not show any significant change in the processor utilization during this period. This supports the speculation that this is an I/O based traffic. The catalog server however showed a sustained peak in processor utilization indicating that this traffic effected the processor load on the catalog server. Similar behavior is also noticed in the B2B site, but the effect of this on the system performance is found to be negligible. From the above discussion, it can be noticed that the effect of load on the database servers, differs with the services provided by the servers.

7.4 Summary

The traffic at the two back-end database servers is characterized using the processor utilization and the access pattern to the disks. The utilization at the back-end is found to be more bursty than the process observed at the front-end for both the B2C and B2B servers. This refutes previous assumption that back-end servers see a smoothed traffic from the front-end. For the main database it is observed that the disk request process is self-similar in nature. This causes a bursty queue length resulting in a heavy-tailed distribution of response-time seen at the front-end². For the B2B environment, this will not be having a major impact as the response-sizes are found to be heavy-tailed in nature. Also regular update activity is seen in all the back-end servers between 1pm to 2pm for the B2C servers and between 12:00 am to 1:00 am for the B2B servers. This did not show any significant changes in the processor utilization as the activity is mainly IO based.

²Please refer Chapter 6 for a further discussion

Chapter 8

Conclusion

Aggregated traffic arriving at an e-commerce servers is characterized in this thesis. Live traffic was collected from two different e-commerce sites. One is a B2B site and the other is a B2C site. The data were collected at three different levels. Access logs from the web servers is collected for application level information, Microsoft performance logs were collected for system level information and processor counters were collected for architectural information like cache hit ratio etc. Information from this data was used to understand the load behavior of the traffic for a normal weekday. Only a specific set of parameters (arrival process, utilization, response-time, transfer sizes etc.) which would impact the system to the maximum extent were used for characterization of the workload.

Self-similar nature of the traffic was established using Hurst-parameter as a measure of degree of self-similarity. Two different tests were used for measuring the Hurst-parameter. The AV-estimator and the R/S plot were used. It was found that, at lower time-scales, R/S plot gave a good approximation of Hurst-parameter, implying that Non-Stationarity does not come into picture at lower time-scales. It was observed that the load behavior of the two sites was complimentary in nature with traffic load shifting from one type of e-commerce site to the other during the later part

of the day. Unlike previous speculation, the back-end server was found more bursty than the front-end server, this was attributed to the fractal nature of the service time at the back-end.

In both the sites, the response-times were found to be heavy-tailed in nature, complying to the results found in web environment. But in the B2C environment, highly bursty arrival of file requests was seen at the disks. It was found that this arrival process is causing high queuing delays at the disk reducing the impact of disk transfer time as compared to the queuing time. This increased the burstiness in the overall response-time seen at the front-end server. An approach was proposed which would reduce the effect of peaks at the disks. This scheme works by splitting the cache into two parts, sending requests to the disk from two different buffer caches. Since the locality of files is random in nature and the file-sizes do not follow a heavy-tailed distribution, the combination of the two request streams to the disk will smooth out over a period of time [7]. This would reduce the burstiness in the back-end service time and also the behavior of the overall response-times seen by the users.

The response-times in both the sites were modeled using Pareto-distribution. In B2B space, the response-sizes were also found to be heavy-tailed in nature. But since the buffer cache hit ratio was very high in these servers, the disk accesses were negligible, rendering the back-end service time very regular.

This work provides an understanding of the complexity of the traffic arriving at e-commerce sites. A first step is made to characterize the traffic. The behavior of the servers at different load periods is studied, along with the behavior of the traffic at different periods of the day. The next step would be to model the traffic arriving at the servers for synthetic generation. But due to the long-range dependent properties of the traffic, much more data will be needed for any such study.

Bibliography

- [1] “Real numbers behind ’net profits.” ActivMedia, June 2000.
- [2] “E-commerce statistics.” <http://www.zdnet.com/ecommerce/stories/main/0,10475,2636088,00.html>.
- [3] D. Krishnamurthy and J. Rolia, “Predicting the performance of an e-commerce server: Those mean percentiles,” in *Proc. First Workshop on Internet Server Performance, ACM SIGMETRICS ’98*, June 1998.
- [4] X. Chen, H. Chen, and P. Mohapatra, “An admission control scheme for predictable server response time for web accesses,” in *Proceedings of the 10th World Wide Web Conference*, 2001.
- [5] K. Kant and M. Venkatachalam, “Two-layer characterization of e-commerce traffic,” tech. rep., Intel Corp. OR, 2000.
- [6] “Cisco and microsoft e-commerce framework architecture.” <http://www.microsoft.com/technet/ecommerce/ciscomef.asp>.
- [7] W. E. Leland, M. Taqqu, W. Willinger, and D. Wilson, “On the self-similar nature of ethernet traffic,” in *Proceedings of SIGCOMM*, pp. 183–193, September 1993.

- [8] M. Crovella and A. Bestavros, "Self-similarity in world-wide traffic : Evidence and possible causes," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 835–846, December 1997.
- [9] P. J. Brockwell and R. A. Davis, *Introduction to Time-series and Forecasting*. New York, USA: Springer-Verlag Inc., 1996.
- [10] L. Cherkasova and P. Phaal, "Session based admission control: A mechanism for improving the performance of an overloaded web server," tech. rep., HP Labs, 1998.
- [11] R. Pandey, J. Barnes, and R. Olsson, "Supporting quality of service in http servers," *Proc. 17th Annual SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, 1998.
- [12] J. Beran, "Statistics for long-memory processes," in *Monographs on Statistics and Applied Probability*, (New York, NY), Chapman and Hall, 1994.
- [13] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson, "Self-similarity through high-variability: Statistical analysis of ethernet lan traffic at source level," in *Proceedings of SIGCOMM*, (Boston, MA), pp. 100–113, 1995.
- [14] G. K. K. Park and M. Crovella, "On the relation between file sizes, transport protocols and self-similar network traffic," *Proc. IEEE Int'l Conf. on Network Protocols*, pp. 171–180, October 1996.
- [15] W. E. Leland and D. Wilson, "High time-resolution measurement and analysis of lan traffic: Implication for lan interconnection," in *Proceedings of IEEE INFOCOM*, pp. 1360–1366, 1991.
- [16] K. Kant, *Introduction to computer system performance evaluation*. New York, NY: Mc Graw-Hill, Inc, 1992.

- [17] V. Paxson and S. Floyd, "Wide area traffic: failure of poisson modelling," *Proc. of SIGCOMM*, 1994.
- [18] K. Kant and Y. Won, "Server capacity planning for web traffic workload," *IEEE trans. on knowledge and data engineering*, pp. 731 – 747, October 1999.
- [19] K. Park, G. Kim, and M. Crovella, "On the effect of traffic self-similarity on network performance," *Proc. SPIE Int'l Conf. Performance and COntrol of Network Sys.*, pp. 296–310, 1997.
- [20] P. R. Mourin, *The Impact of Self-Similarity on Network Performace analysis*. PhD thesis, Carleton Univ., December 1995.
- [21] P. Mohapatra, H. Thanthry, and K. Kant, "Characterization of bus transactions for specweb96 benchmark," *2nd Workshop on Workload Characterization (WWC)*, October 1999.
- [22] D. A. Menasce, A. A. F. Almeida, R. Fonseca, and M. A. Mendes, "Resource management policies for e-commerce servers," *2nd Workshop on Internet Server Performance*, May 1999.
- [23] M. F. Arlitt and C. L. Williamson, "Web server workload characterization: The search for invariants," *ACM SIGMETRICS Conf.*, May 1996.
- [24] D. P. Heyman and T. V. Lakshman, "What are the implications of long-range dependence for vbr-video traffic engineering?," *IEEE Trans. Networking*, pp. 301–317, June 1996.
- [25] R. J. Glushko, J. M. Tenenbaum, and B. Meltzer, "An xml framework for agent-based e-commerce," *Communications of ACM*, vol. 42, p. 106, March 1999.
- [26] M. Ma, "Agents in e-commerce," *Communications of ACM*, vol. 42, pp. 78 – 80, March 1999.

- [27] W. Diffie, "E-commerce and security," Tech. Rep. 3, Sun Microsystems, Palo Alto CA, September 1998.
- [28] P. Bradford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," in *Proc. 1998 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, Madison, July 1998.
- [29] "Microsoft management console : Performance." <http://www.microsoft.com/windows2000/techinfo/howitworks/management/mmcover.asp>.
- [30] "An explanation of the specweb96 benchmark." <http://www.specbench.org/osg/web96/webpaper.html>, 2000.
- [31] "Specweb99 design document." <http://www.specbench.org/osg/web99/docs/whitepaper.html>.
- [32] "Tpc benchmark w standard specification, version 1.4." <http://www.tpc.org/tpcw/default.asp>.
- [33] R. Jain, *The Art of Computer system performace analysis*. John Wiley & sons Inc., 1991.
- [34] D. Veitch and P. Abry, "A wavelet based joint estimator for the parameters of lrd," *Special issue on Multiscale Statistical Signal Analysis and its Applications, IEEE Trans. Info. Th.*, vol. 45, April 1999.
- [35] M. Rougham and D. Veitch, "Measuring long-range dependence under changing traffic conditions," *Proc. of INFOCOM*, pp. 1513–1521, 1999.
- [36] Z. Sahinoglu and S. Tekinay, "On multimedia networks: Self-similar traffic and network performance," *IEEE Communications Magazine*, January 1999.

- [37] R. A. Floyd, "Short-term file reference patterns in a unix environment," tech. rep., Computer Science Dept. Univ. of Rochester, 1986.
- [38] J. K. Ousterhout, H. D. Costa, D. Harrison, J. A. Kunze, M. Kupfer, and J. G. Thompson, "A trace driven analysis of the unix 4.2bsd file system," tech. rep., Dept. of Computer Science, Univ. of California at Berkley, 1985.
- [39] D. A. Menasce and A. A. F. Almeida, *Capacity Planning for Web Performance: Metrics, Models and Methods*. Upper Saddle River, NJ: Prentice Hall, 1998.
- [40] J. Hennessey and D. Patterson, *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, 1996.

MICHIGAN STATE LIBRARIES



3 1293 02177 8224