CHAPTER 6

EXPERIMENTAL RESULTS WITH E-COMMERCE WEB APPLICATION

6.1 Introduction

Keeping E-Customers of any E-Commerce Web site satisfied has always been a task of a highest priority for its management. Dissatisfaction can lead to bad reputation, loss of both current and potential E-Customers which result in substantial financial loses. Service unavailability and delayed response time are the main generators of E-Customers dissatisfaction [46]. According to [15] user satisfaction is inversely related to response time, which could be the single most important variable when it comes to user satisfaction. 'Faster is better' is the simplest equation in the entire field of Internet strategy [16]. In order to assure relevant QoS levels of the vital performance metrics, including the response time, a capacity planning methodology, based on the usage of relevant predictive models, has to be applied continually, as a crucial part of the E-Commerce Web site deployment. The requirements of E-Commerce applications were identified and the ability of the models discussed in the previous chapters to accommodate these is studied in detail in this chapter. The detailed study clearly brought out the applicability of the model in deriving relevant performance metrics.

6.2 Workload Characterization

When modeling workload one has to be aware of two facts: E-Customers are not mutually identical. E-Customers access the Web site and invoke specific E-Commerce functions in an unpredictable and stochastic manner. A mixture of various service categories of E-Customers is specified by defining a random variable along with its Probability Density Function (PDF). If there are *k* disjoint service categories of E-Customers identified, e.g. (t_1, t_2, …,t_k), then each of them can be assigned a corresponding probability (p_1, p_2, …, p_k), based on its particular participation within the workload mixture. We model four service categories of E-Customers based on the intensity of buying. These may be termed as *New shoppers, Rare shoppers, Ordinary shoppers, and Frequent shoppers*; service category–4, service category–3, service

category–2, and service category–1 as used in Chapter 4 may represent these in the same order. The specifics of their online shopping behavior can be defined through specification of the probabilities within the UBMG. A typical UBMG [41] for an E-Commerce SR, is shown in Figure 6.1. Each category of customer will have its own UBMG. The *Frequent shopper* will have a high probability for the 'Add' to 'Pay' transition and a *New shopper* will have a high probability for the 'Browse' to 'Quit' transition; similar differences will be present in the other transition probabilities as well. The focus here being the study of applicability of the models proposed earlier to E-Commerce applications, this disparity in the UBMGs has not been taken into consideration. One UBMG is used for all the four types of customers. The workload characteristics with a selected set of parameters have been used to illustrate the applicability of the model here.

6.2.1 Service Architecture and User Behavior Modeling

UBMG-based evaluations can generate virtual users to simulate real user visits to the Web applications with the aim of reproducing the performance of the tested object realistically. A total of six stages one corresponding to each of the states as shown in Figure 6.1, was used for modeling the service architecture. The UBMG (typical E-Commerce users) in Figure 6.1 was used to generate virtual users which are a group of status sequences represented by access operations to the application. For the transitions emanating from any source state to destination state in the UBMG, a random selection 'from a bag of balls with distributions confirming to respective probabilities' was used (Vide Section 4.3.2). A series of simulation runs have been carried out in order to estimate the values of various performance metrics. Each run was made long enough to get steady *state behavior* vis-à-vis the estimated parameters. The metric of interest of SR is user satisfaction in terms of service availability and response time. The metrics of interest of SP are maximum system utilization and minimum resource commitment [5, 54].

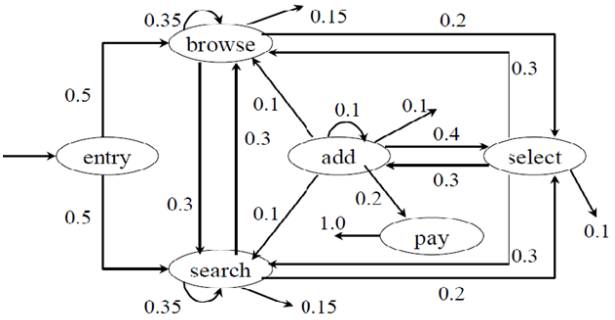

Figure 6.1: User Behavior Model Graph (mostly '*Browser*')

6.3 Simulation with 'BBAM-I'

A series of simulation runs were done with a workload mix of {49%, 24%, 11%, and 15%}, the four customer categories–*New shoppers*–4, *Rare shoppers*–3, *Ordinary shoppers*–2, and *Frequent shoppers*–1–with probabilities 0.49, 0.24, 0.11, and 0.15 respectively. The UBMG in Figure 6.1 was used with the arrival rate as a stationary random process with $\{\mu=26, \sigma^2=7\}$. The simulation was done with different system capacities (ie., number of threads). Further SR requirements(for an agreed period as mentioned in SLA) were taken as:

1. Desired user satisfaction level >= 95%.
2. Customer satisfaction to be computed using metrics service availability and response time with equal weightage for either of them.

The results of simulation are summarized in Appendix A.3.1.

A close examination of the same reveals the following:

1. The user satisfaction derived from response time metric is always 100% as a thread once allocated remains committed until the session terminates.
2. With increase in number of threads, drop rate decreases and becomes 0 for number of threads (T) exceeding 180. This implies steady increase in service availability (reflected as user satisfaction) with its full value (100%) being realized for 180 threads.
3. As the system capacity represented by the number of thread increases from 100 through 120, 150, and 180 the drop rate reduces from 38.14% through 25.32%, and 7.85% to 0%. In other words customer satisfaction improves steadily with increase in system capacity. Further the system utilization reduces from 100%. These two together clearly bring out the need for

improving concurrency support without arbitrary (too much) increase in system capacity.

6.3.1 Effect of Peak Load

The workload represented by a stationary random arrival characterization as done earlier was used as a base. A random additional load from 150th to 200th timeslot– representing the peak load – was superimposed on the base load as shown in Figure 6.2.

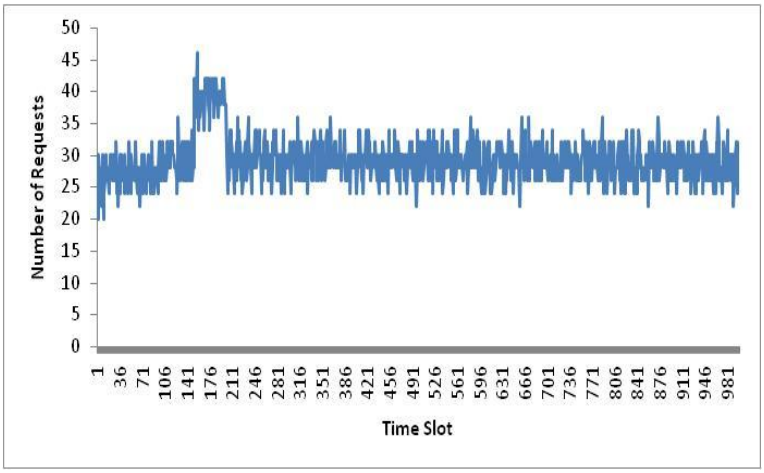

Figure 6.2: Workload Characterization – (with Superposed Load Fluctuations)

The simulation was carried out for 1000 timeslots on a system with 180 threads as capacity. Figures 6.3a and 6.3b show the variation of drop rate and system utilization. One can clearly see that the drop rate remains conspicuously high during the peak load with system utilization remaining at peak throughout this period.

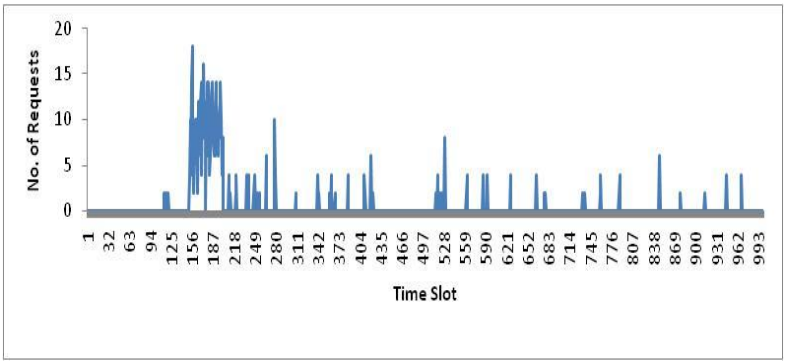

Figure 6.3a: Performance Evaluation with **'BBAM-I'**- Drop Rate of Requests (@Peak Load)

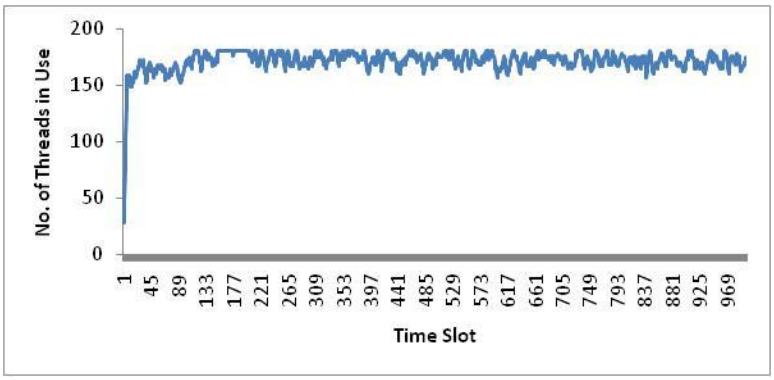

Figure 6.3b: Performance Evaluation **'BBAM-I'**- System Utilization (@Peak Load)

As the assigned thread is not withdrawn from the request until the session is completed, user satisfaction in terms of response time is 100%. In this case it can be seen that the user satisfaction is purely governed by service availability. With the total number of requests dropped being 632 (1.97%) user satisfactions in terms of service availability is 98.039%, the system utilization was 95.14% and the overall user satisfaction is 99%.

Findings: The service availability is limited by the system capacity (threads), improving this factor calls for increase in capacity. Increase in system capacity leads to poor system utilization. Finally, this approach as far as an E-Commerce application goes is extremely unfair to clients, as some clients will be able to establish connections quickly, while some others will experience service outage.

6.4 Simulation with 'BBAM-II'

The system capacity represented by the number of threads (*T*) used as 120 was chosen for analysis. All other parameters, simulation conditions and the SR requirements for scheme evaluation were identical to those with **'BBAM-I'**. Further *K* values-the '*Tokens-Basket*' sizes of 125, 130, 150, 160, and 180 were used. At the outset it is pertinent to mention that the execution time here also includes the queuing and de-queuing times associated with adding and removing requests from the '*Tokens-Basket*' and context switching (CPU switches between threads). Simulation results shown in Appendix A.3.2 reveal the following:

1. The system utilization is 95.8% for *K*=120; it rises to 99.84% at *K*=130 and remains so for all higher values of *K*. In fact in the first few time slots of simulation some threads remain free in all cases. This accounts for the

deficiency of 0.16% in system utilization. Thus in effect the system is fully utilized even with a marginal increase in the '*Tokens-Basket*' size above 120 – the number of threads.

2. The drop rate decreases from 22.74% through 19.67%, 7.852%, and 3.106% to 0% for *K* of 125 through 130,150,160, and 180 respectively. Thus the '*Tokens-Basket*' manifests as enhanced server availability.
3. Users enjoy 100% satisfaction on the service availability factor with *K* = 180. This is clear from the log as the drop rate is 0 with *K* = 180 which means the server has accepted every incoming request for service. Incidentally it also shows that *K* need not be increased further to serve this level of steady load.
4. The increase in number of tokens (*K*) reflects as a corresponding increase in service time. This is obvious from the fact that it is possible for a set of requests to be kept waiting with a token assigned to them due to unavailability of threads in that timeslot. This otherwise means that the introduction of the '*Tokens-Basket*' has reduced the service rate (computed as the ratio of the number of requests completed in that time slot to the number of total number of requests residing in the system) at each time slot whenever the number of requests in the system is greater the system capacity. As long as there were free threads in the system the service rate was on par with that of **'BBAM-I'** but once all threads were consumed and requests were kept waiting in the '*Tokens-Basket*' the service rate was adversely affected.
5. So long as the request execution time is less than the expected response time, the service quality is 'deemed' acceptable; but beyond this, the decrease in service quality is visible as a corresponding delay in response time.
6. For values of *K* from 125 through 130,150, and 160 to 180 the response time increases by 4.12%, 6.19%, 10.83%, and 13.59% to 15.44%. In fact this delayed service corresponds to those requests in **'BBAM-I'** which were denied system resources during different time slots due to non availability of threads.
7. The introduction of '*Tokens-Basket*' achieves two purpose:
   a. Serving additional requests without degraded quality as long as the execution time is less than the expected response time. That is, free threads available in thread pool at different timeslots can exploit this to serve additional requests without sacrifice of response time.

b. Serving additional requests with a slight delay in response time. The acceptable limit set by the SR on response time can be used to decide on how many of such additions can be accepted.

6.4.1 Effect of Peak Load

A random additional load from 150th to 200th timeslot - representing the peak load - was superimposed on the base load as shown in Figure 6.2. The simulation was carried out for 1000 timeslots on a system with 180 threads as capacity and number of tokens=200 to study the effect of peak load on service availability. Figures 6.4a and 6.4b show the variation of drop rate and system utilization. The system utilization was found to be 95.47% compared to 95.14% with **'BBAM-I'**, i.e., the system was fully utilized for more number of timeslots. Comparing the drop rates shown in Figure 6.3a and 6.4a during timeslots from 150 to 200, it can be seen that although requests are rejected in **'BBAM-II'**, the drop rate with **'BBAM-II'** remains less than that of **'BBAM-I'** and is practically 0 at other time slots. From the recorded logs the request drop during peak load was found to be 292 requests (0.9%). In contrast it was 470 with **'BBAM-I'**. This clearly brings out enhanced service availability in contrast to **'BBAM-I'**. It is important to note that this increase happens only as long as free tokens are available; beyond that the system becomes saturated and starts rejecting requests as was the case with **'BBAM-I'**. This can be seen from the graph in Figure 6.4a. Rejection rate is 0 in **'BBAM-II'** only if unbounded token basket is chosen. The number of requests served with extended response time was 4250 (1.45%). The % increase in the average response time was 0.38%. User satisfaction computed based on service availability and response time (as evaluated in **'BBAM-I'**) was found to be (98.82%).

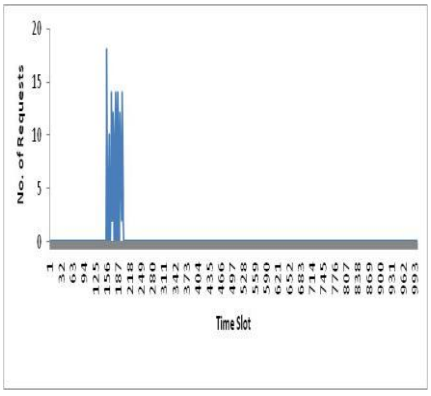

Figure 6.4a: Performance Evaluation **'BBAM-II'**- Request Drop (@Peak Load)

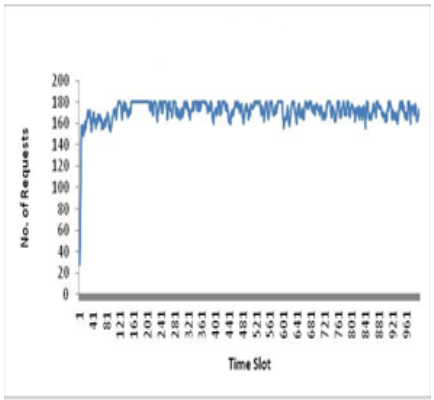

Figure 6.4b: Performance Evaluation **'BBAM-II'**- System Utilization (@Peak Load)

6.4.1.1 Effect of Increase in Average Workload

'BBAM-II' is flexible enough to be customized for achieving different levels of service performance based on requirements of SR and SP. A number of load conditioning policies could be employed by the PL to deal with increase in load; for e.g.

- SP can use PL to reduce the bottlenecks for shared resources. Arranging requests based on their CPU and I/O intensiveness can enable the SP to increase resource utilization.
- SR can use PL to offer differentiated service for site visitors.

Table 6.1 summarizes the simulation results with an assumption of increase in popularity of the site (average workload increase from μ=30 to μ=35) number of tokens (K)=500 and system capacity as 180(enough to handle μ of 30) and PL characterized by:

- Prioritization of requests based on service category.
- No limit set on the overall session time.

Table 6.1: Simulation Results with Increase in Workload using **'BBAM-II'**

Stages(S)=6, Simulation Timeslots=1000				
Threads(T)=180,Tokens(K)=500				
Workload Mix {47%,25%,13%,15%}				
μ=30, σ^2=7				
Service Category	**1**	**2**	**3**	**4**
Number of Requests	14255	7231	3336	4220
Average Response Time	4.2476635	8.508611	13.900862	16.858156
μ=35, σ^2=7				
Number of Requests	16699	8218	3968	5111
Average Response Time	5.246782	10.681522	15.72607	21.7837
Number of Additional Requests Served	2444	987	632	891

The inferences from the logs (Table 6.1) are as follows:

1. The service differentiation offered for different service categories is brought out in terms of average response time. It can be seen that with μ=30 and μ=35 the service category 1 customers are served with minimum response time in

that simulation run compared to other service categories. With μ=30 the response time for service category 1 is 4.247 compared to 8.508, 13.900, and 16.858 offered for service categories 2, 3, and 4 respectively; with μ=35, the respective values are 5.246, 10.68, 15.72, and 21.73.

2. With the increase in the average workload it can be seen that there is a corresponding increase in response time of different service categories. As seen in the logs it has increased from 4.247 to 5.246 for service category1, 8.508 to 10.68 for service category 2, 13.900 to 15.72 for service category 3 and 16.858 to 21.73 for service category 4. The number of requests additionally served under each category are 2444, 987, 632, and 891 respectively. If this increase in response time is not within the expected response time of the SR, it can serve as an indication for additional capacity requirement.

6.5 Profiling

A variety of additional information can be collected based on the simulation. For example all the state transitions of the customers during their visit to the E-Commerce site have been sifted out and shown in Table 6.2. The source and destination sites have been as given in the first column and the first row respectively. The following form a typical set of inferences from this:

- All the data in every row have been normalized with the respective total in the last column as the basis. The results are shown in Table 6.2. The entry to browse and entry to search transition fractions are 0.501884 (=13323/26546) and 0.498116 (=13223/26546) respectively. These are close enough to the respective Figures (0.5 each) in the UBMG of Figure 6.1. Similar closeness holds good for all other entries in Table 6.3 as well. In short the data in Table 6.3 conforms to the UBMG of Figure 6.1. We started with of course making allowance for statistical deviations. This clearly shows that the simulation duration is long enough to accommodate all the behavioral aspects of the customer set.

- With 10345 items added to the cart 1998 payments have been made. Thus every customer who has made a purchase has purchased 5.18 (=10345/1998)

items on the average.

- With a total of 26546 customers visiting the site the maximum number of transitions is from entry to browse at 13323 implying that on the average 50.1884% (=13323/26546) of customers have searched for an item but reverted to search itself (ideally the business would like every customer who comes, to make a purchase and also to make it available at the first '*hit*'). Similar inferences can be made on many other transitions too.

- The '*Buy-to-Visit*' ratio shows how many sessions out of the total number have terminated with buying something. Here its value is 8.5% (1998/23381).

- The percentage of sessions that have ended with an empty shopping cart is 80.7% (=1-1998/10345).

These performance measures can be used for estimating specific, business-oriented performance metrics, like revenue throughput and potential loss throughput. Thus the model allows one not only to assess a variety of performance parameters that can be utilized for capacity planning, but also to experiment and derive business metrics with various workloads and infrastructure modalities. Further at specific intervals the logs of the type in Table 6.2 can be collected separately for each category of customers. This information can be used to change the PL to enhance service differentiation. It is also useful in refreshing customer categorization dynamically. These can possibly enhance business returns conspicuously.

Table 6.2: User Behavior Model Graph Collected from the Logs for μ=30, σ^2=7 with K=500, T=180, and S=6.

	Browse	**Search**	**Select**	**Add**	**Pay**	**Exit**	**Total**
Entry	13323	13223					26546
Browse	23180	19474	13018			9706	65378
Search	19510	23068	13069			9655	65302
Select	8956	9031		9298		2917	30202
Add	968	1073	4141	1047	1998	1103	10330
Total	65937	65869	30228	10345	1998	23381	197758

6.5.1 Profiling at the Second Level of Abstraction

An E-Commerce Web application (available at URL www.offermama.com) was developed to test the performance and profiling capability of the model at the second level of abstraction. Seven Web pages (Figure 6.5) of the application were considered for performance evaluation. The load times of the individual Web pages from the local host and the average think time of users to switch between Web pages (taken as 5ms–as mentioned by TCP-W benchmark) were included to profile the response time. The obtained results for average response time with a single *stage* modeled as a FSM having 7 *stages* with each corresponding to a Web page and associated with the UBMG as shown in Figure 6.6 are summarized in Table 6.3.

Table 6.3: Total Average Response Time with **'BBAM-II'** for μ=30, σ^2=7 with K=500,T=180, and S=6

	Index	**Page1**	**Page2**	**Page3**	**Page4**	**Page5**	**Page6**	**Total (ms)**
Index		7.4691				5.8636		7.1262136
Page1			4.8019	7.8302	7.0568			6.1227974
Page2	4.356	7.5935		7.6181				6.9493177
Page3			4.8455		6.2545		5.38	5.4165049
Page4		7.4061						7.4061135
Page5							4.667	4.6666667
Total (ms)	4.356	7.4796	4.8065	7.7264	6.8658	5.8636	5.345	6.3093054

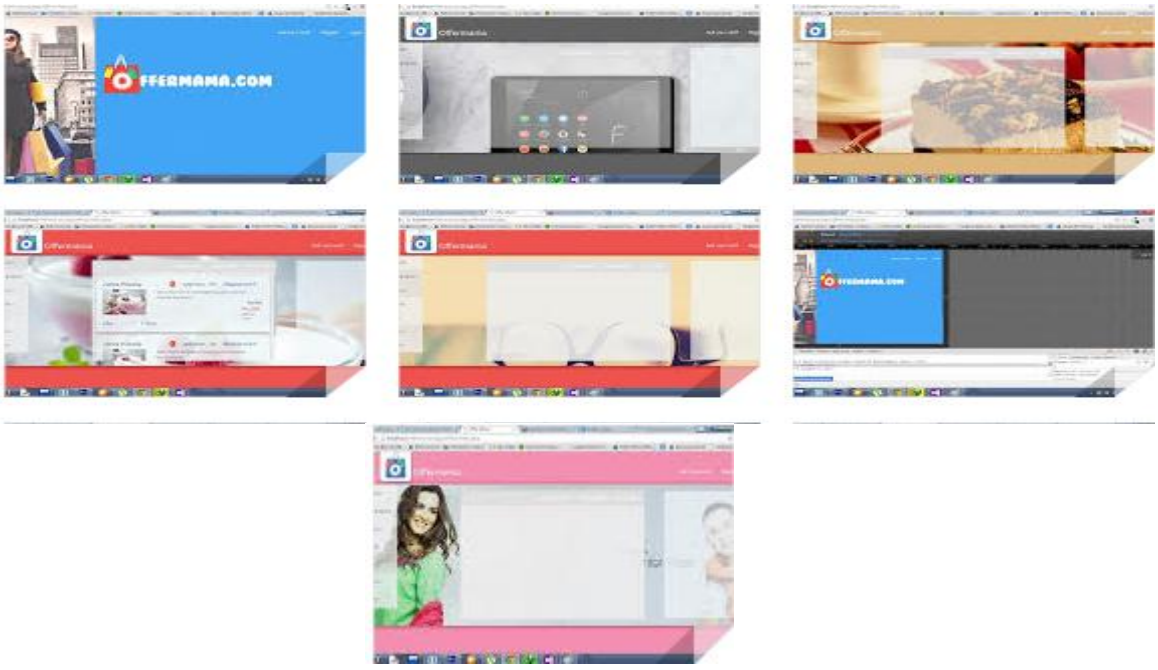

Figure 6.5: Web Pages used in Performance Evaluation [*Courtesy*: *www.offermama.com*]

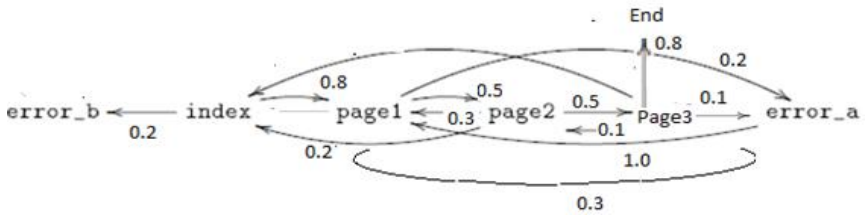

Figure 6.6: Second Level of Abstraction – User Behavior Model Graph for '*Page-Visit*' in a *Stage*

The logs (Table 6.4) collected by the model at the second level of abstraction on *'Page-Visit'* patterns of users can aid a SR to evaluate metrics like '*Page-Visit-Ratio*' which could form the basis for recommender systems to evaluate the contents and the design of Web pages. A detailed profiling of the individual pages based on their content as shown in Figure 6.7 can aid SR to decide on the amount of reduction on dynamic contents of the page at times of peak loads on server. This reduction in the quality of the Web page delivered can reduce the delay in response time. Figure 6.7 shows that a reduction of size of image by about 50%, 75%, and 100% can minimize the delay in the response time by 0.5s, 1s, and 1.5s respectively.

Table 6.4: User '*Page-Visit*' Metrics Collected from the Logs
for μ=30, σ^2=7 with K=500, T=180, and S=6

	Index	**Page1**	**Page2**	**Page3**	**Page4**	**Page5**	**Page6**	**Total**
Index		783				189		972
Page1			4265	2457	1650			8372
Page2	972	1453		2297				4722
Page3			462		475		3812	4749
Page4		2124						2124
Page5							188	188
Total	972	4360	4727	4754	2125	189	4000	21127

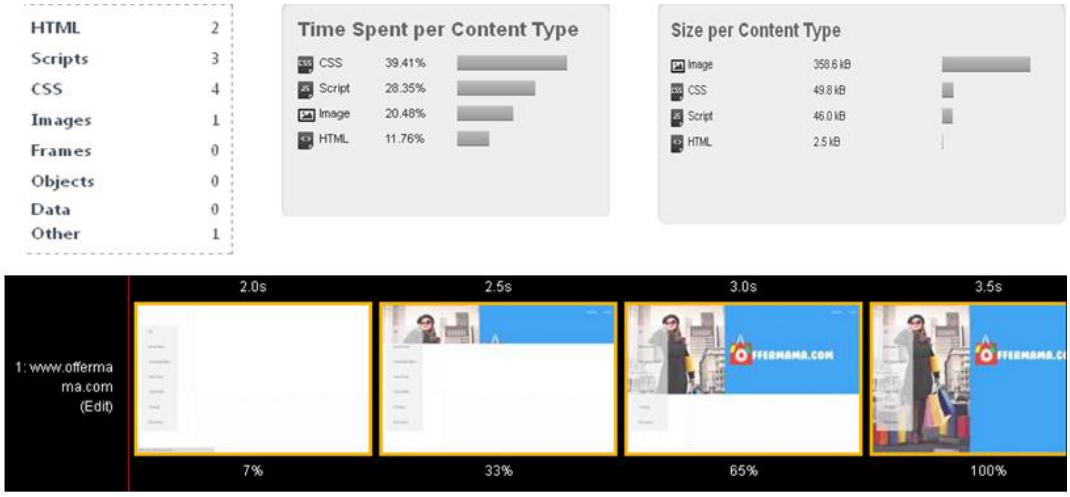

Figure 6.7: Profiling of Web Page Load Time
[*Courtesy: www.offermama.com*]

6.6 Deriving Limits for Service Level Agreement Parameters

As mentioned in Chapter 5 (Ref: Table 5.2) the simulation results can aid the SP in deciding on the commitments to be made in the SLA.

6.6.1 Service Level Agreement Template

Purpose : E-Commerce Application

Signatory : ----------

Valid From : ---------- to ----------

Service Description : An online shopping application with the following functionalities (Login, Browse, Search, Select, Add Cart, Pay).

Requirements:

- The Web application for online shopping is to be hosted in dedicated server.
- The application's performance is to be modelled based on the available UBMG.
- Service differentiation to be offered for 4 types of customers.
- Strong profiling of requests regarding session information, arrival time and response time.
- Objective is to maximize the service availability and minimize user frustration.

Guarantees:

Guarantees expected by SR from SP

- User Frustration : $minUserFrustration
- Request Rejection : $minRequestRejection
- Throughput : $maxThroughput
- Service Availability : $maxServiceAvailability
- Resource Commitments
    - Threads : $minThreads
    - Tokens : $maxTokens

Guarantees expected by SP from SR

- Payment for hosting service : ----------
- Workload : $max average workload
- UBMG : Probability of transitions from one state to another
- Functionalities (FSM States): $MaxStates

Evaluation Parameters

(i) Metrics
    a. Throughput - $minThroughput
    b. Response time - $maxResponseTime
    c. User Frustration - $maxUser Frustration
    d. Drop rate - $maxDropRate

(ii) Additional Functionalities from SP
    a. Load Conditioning.
    b. Fast Authentication service.

(iii) Function – PL Rule for identifying service categories and offering service differentiation.
    Equations indicating dependencies of metrics on parameters (e.g., user frustration is to be quantified by *response time* and *request drop*).

Subject to conditions:

1.	Average Workload expected	:	30
2.	Number of Service Categories	:	4
3.	Type of Service	:	Dedicated hosting
4.	Number of States to be processed in the application	:	6

Response time: Minimum commitment of SP should be the maximum average response time of service category 4(16.85). Referring to the results presented in table 6.1 for μ=30 we can see that this commitment was satisfied for all the service categories and for μ=35 it was satisfied for service categories 1, 2, and 3 only. The deviation with service category 4 is this case is due to increase in the workload (In SLA workload agreed for service was only for μ=30), which is an indication for SP to renegotiate the SLA.

Throughput: Maximum commitment=T/S=180/6=30. From the Table 6.1 it can be seen that the commitment is met with μ=30 i.e., 29040/1000=29.040 is the throughput at each timeslot. This is the throughput assured without any degradation in service quality. For μ=35 the *throughput*=33996/1000=34 requests per time slot but with corresponding increase in response time. In other words if the SP is offering a *throughput* $>T/S$ there will be a corresponding deviation in response time. SP not being able to meet minimum response time guarantee and maximum throughput agreed is an indication for SLA renegotiation.

User Frustration: It can be observed from the results presented in the Table 6.1 that in each simulation run the user frustrations as a measure of response time increases from service category 1 to service category 4. The user satisfaction measured in terms of request rejection is 100% as K (number of tokens) =500 were sufficient enough to accept every incoming request.

6.7 Model Evaluation

The suitability of the schemes **'BBAM-I'** and **'BBAM-II'** in terms for E-Commerce can be summarized as follows:

- Offers better service availability at times of increase in work load. This is very important because user becomes more frustrated when the service becomes totally unavailable and it leads to business loss.

- Offers flexibility to support service differentiation (different service levels for different types of E-Customers).
- Offers support for strong profiling. Collected logs and derived performance metric allow assessment of the business requirements and capacity. These can aid SR and SP to find if any SLA re-negotiation is required, SR in studying behavior patterns of online customers on their site and improve business strategies, SP in indicating resource bottlenecks and under utilization.
- Facilitates easy deployment by shielding application programmers from many of the details of scheduling and resource management.
- Supports Internet Service to analyze the request stream and to adapt behavior to changing load conditions. For example, the system should be able to prioritize and filter requests to support degraded service under heavy load.

6.8 Conclusions

The chapter has presented a systematic way for performance evaluation in E-Commerce applications using the proposed operational schemes presented in Chapter 3 and Chapter 4. The approach is based on a workload characterization generated by use of UBMGs that allow describing the behaviors of typical users on a Web site. Performance evaluation allows the contributions to be clearly justified. It was verified that the model can be also enhanced at ease by introducing new classes of E-Customers, new performance metrics, as well as several other characteristic operating scenarios. The study reported here clearly brings out the potential to enhance business performance through dynamic service categorization. Of course, this warrants more detailed investigation. Two aspects that could affect the service availability were identified at the end of the study of the modeling of the E-Commerce application:

1. Possible delay in system throughput, increased response time at the ‘entry’ stage (home- page) due to malicious requests in the workload.
2. Possible service unavailability (time out of sessions/delayed response time) at the ‘payment’ stage (Web page for making payment) because of the dependency on an external mobile network service (SMS-OTP) for authentication.

Solutions to such cases are explored in Chapter 7.