Analysis of Web Workload on QoS to Assist Capacity

K. Abirami, N. Harini, P. S. Vaidhyesh and Priyanka Kumar

Abstract Workload characterization is a well-established discipline, which finds its applications in performance evaluation of modern Internet services. With the high degree of popularity of the Internet, there is a huge variation in the intensity of workload and this opens up new challenging performance issues to be addressed. Internet Services are subject to huge variations in demand, with bursts coinciding with the times that the service has the most value. Apart from these flash crowds, sites are also subject to denial-of-service (DoS) attacks that can knock a service out of commission. The paper aims to study the effect of various workload distributions with the service architecture 'thread-per-connection' in use as a basis. The source model is structured as a sequence of activities with equal execution time requirement with an additional load time of page (loading embedded objects, images, etc.). The threads are allocated to the requests in the queue; leftover requests if any are denied service. The rejection rate is used as a criterion for evaluation of the performance of the system with a given capacity. The proposed model could form a basis for various system models to be integrated into the system and get its performance metrics (i.e. QoS) evaluated.

1 Introduction

Today Internet has emerged as the default platform for application development. Unfortunately, modern applications demand more complexity than traditional applications. As the Internet was not designed to suit the requirements of modern applications, the execution results in high frustration of users. This factor demands a research on how the existing infrastructure could be modified for efficient execution of modern web applications [1]. The complexity exhibited by applications are multifold (process, data, load, configuration, scale, etc.). With an intent to improve the

K. Abirami (✉) · N. Harini · P. S. Vaidhyesh · P. Kumar
Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India
e-mail: k_abirami@cb.amrita.edu

D. Pandian et al. (eds.), *Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)*, Lecture Notes in Computational Vision and Biomechanics 30,
https://doi.org/10.1007/978-3-030-00665-5_57

performance of web applications a study on workload characterization is compelling. Many researchers have focused their study on understanding the characteristics and intensity of workloads. In this work, we discuss the role of workload models for resource assignment in the scenario of the e-commerce application. The impact of the workload on system properties and behavior is analyzed using a capacity planning model. The proposed system evaluates the Quality of Service (Qos) and Quality of Experience (QoE) perceived by the users for different workload distributions. These observations could aid in framing security mechanisms, recommendation engines, data distribution policies, etc.

When the system is scaled, the work also presents major findings from experimentation indicating performance implications. The rest of the paper is organized as follows: Sect. 2 presents a comprehensive overview of the literature on different workload distributions. Section 3 summarizes the characterization methodologies and related measurement process. Section 4 presents the results and analysis of experimentation and finally, Sect. 5 presents concluding remarks.

2 Literature Review

Many research work addresses the black box approach for the assessment of performance based on workloads. Rejection rates have a huge impact on the performance of the system [2]. Recent rates have a huge impact on the performance of the system [2]. Recent studies have also considered performance measurements based on user behavior patterns and businesses [3, 4]. The response time metric has been chosen in most of the research work for performance evaluation [5]. The Zipf law's applicability of web workloads is addressed by Levene et al. [6] and Menasce et al. [7]. Mi et al. [8] and Harini and Padmanabhan [9] discuss the need for stationary of arrival processes to study and characterize web load. Harini and Padmanabhan [9, 10] addresses the issue of the presence of malicious request in the incoming lot which needs to be weeded out before the commencement of processing. Workload management is a process of effective workload distribution to achieve optimal performance and productivity levels. Modelling workload distributions would aid one to understand the performance and the scalability of the system. Workload model of an application depicts how the application would perform in the given infrastructure. The performance is usually assessed using Service-Level Agreements (SLA). Little Theorem gives a relationship between the average number of users, arrival rate and average time, an end user spends in the system.

The theorem state that

$$L = \lambda N$$

where λ is the arrival rate and L is the effective arrival rate.

The only prerequisite being system should not preempt and must be stable. The arrival pattern of the request can be modeled based on different probability distributions like exponential, normal, binomial, Poisson, Zipfian.

2.1 Distributions

2.1.1 Exponential Distributions

Exponential distribution is a well-known concept in the theory of probability and statics. The distribution denotes the time between two events in processes where the events are continuous and occur independently. The key property of the distribution is memorylessness. This general exponential distribution is given by

$$f(x;\lambda) = \left\{ \begin{array}{cc} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{array} \right\}$$

where λ greater than 0 is the rate parameter. The distribution is well supported in the interval 0 to infinity. This distribution is mainly used to model service times rather than arrival patterns. These can have a strong effect on performance evaluation results.

2.1.2 Normal Distribution

Normal distribution is a very commonly used distribution to determine whether an observation falls between two extreme limits. This distribution is used to model random variables in natural and social sciences. The normal distribution is used in real-valued random variables, where the distributions are not available. The general normal distribution is given by

$$f\left(x|\mu, \sigma^2\right) = \frac{1}{\sigma}\psi\left(\frac{x-\mu}{\sigma}\right)$$

The standard normal deviate is given by Z where

$$Z = \frac{(X-\mu)}{\sigma}$$

These could be used to model the peak of arrivals and at a more concrete level, it can help one to identify results of random effects on workloads.

2.1.3 Poisson Distribution

The application of Poisson distribution in traffic problems is not new. A Poisson distribution is a probability distribution of a discrete random variable that represents the number of statistically independent events occurring within a unit of time or space. Time-based Poisson variables are more popular. Given the expected Value μ of the Poisson variable x the probability function is defined as the probability of observing k events in an interval is given by the equation

$$P(k \text{ events in interval}) = \mathrm{e}^{-\lambda}\frac{\lambda^k}{k!}$$

where the average number of events per interval e is the number 2.71828 … (Euler's number) the base of the natural logarithms k is any natural number, $k! = k$ (k 1) (k 2) 2 1 is the factorial of k. This could be used to model the rate of arrival of request patterns and it could be also used to measure the performance when requests are queued in the system.

2.1.4 Zipf Distribution

A Zipf distribution is sometimes referred to as zeta distribution. This is particularly used for modeling rare events. The probability density function for Zipf distribution is the nth raw moment is defined as the expected value of X_n:

$$m_n = E\left(X^n\right) = \frac{1}{\zeta(s)}\sum_{k=1}^{\infty}\frac{1}{k^{s-n}}.$$

The series on the right is just a series representation of the Riemann zeta function, but it only converges for values of *s-n* that are greater than unity. Thus:

$$m_n = \begin{cases} \frac{\zeta(s-n)}{\zeta(s)} & \text{for } n < s - 1 \\ \infty & \text{for } x \geq 0 \end{cases}$$

Note that the ratio of the zeta functions is well defined, even for $n > s-1$ because the series representation of the zeta function can be analytically continued. The Zipf distribution turns out to better describe varied human activities. It is a good model for popularity distribution. This does not change the fact that the moments are specified by the series itself, and are therefore undefined for large n.

2.1.5 Binomial Distribution

The Binomial distributions will have two outcomes, success or failure. The experiment can have n number of trials and the outcomes are independent. The general equation of the distribution is given by the following:

$$b(x; n, p) = \binom{n}{r} p^x (1 - p)^x$$

where,

n represents the number of trials, x represents the number of successes, p represents the probability of success in an individual trial.

The distribution could be used for modeling random arrival patterns, study effects of peak load, perform resource assignments, etc.

2.2 Summary of Findings

A special case of performance evaluation that deserves individual attention is capacity planning. Many research works propose different methodologies for setting up configurations that would provide desired performance. The required system capacity obviously depends on workload intensity, i.e., one needs more capacity to do more work. The relationship between capacity and workload is often not linear. Researchers have also stated that burst is an important attribute that contribute to capacity planning. Burst refers to large fluctuations in workload intensity.

A good characterization technique thus requires a clear understanding of burst characteristics. A combination of system model with workload characterization can enhance the performance of the system. Although individual schemes specific to Internet services have been explored in large, to the best of our knowledge a comprehensive study based on multiple workload distribution based analysis with its performance assessment has not been addressed to a greater extent.

2.3 Problem Statement

To build a system capable of characterizing the performance of a system model for varied workload distributions, which could aid in capacity planning, arriving at optimal configuration to improve QoS, assist in data movement with applied security features.

3 Proposed System

Though Internet services remain simple in the structure at the start they become more complex when functionalities of the service expand. The block diagram for service architecture used for experimentation is shown in Fig. 1.

3.1 Model Description

The basic service is taken as composed of n sequential activities. A single processor system with T threads is used as a basis for service. The service request is taken as a stationary random process having a selected distribution (Normal, Binomial, Exponential, Poisson, Zipf) with its associated parameters.

3.1.1 Request Arrival Pattern

A Service request is taken as a stationary random process with an associated mean and variance attributes. The proposed system considers discrete random arrival pattern. Each arrival is independent of the previous arrival. An arrival set is characterized with a number of incoming requests and a class type associated with it. The arrival capacity is not limited.

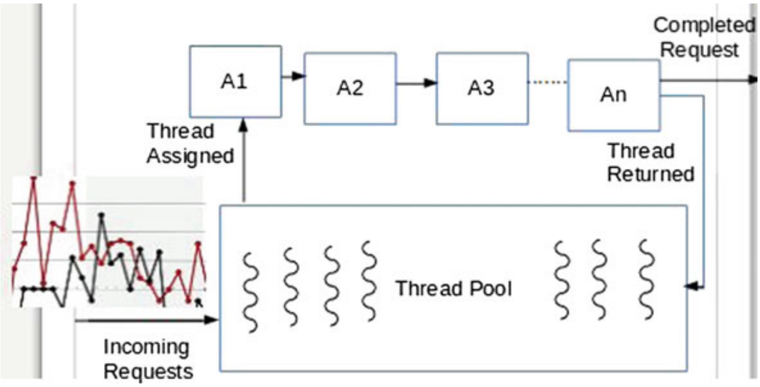


Fig. 1 Service architecture

3.1.2 Request Characteristics

A request is characterized by arrival time and service class. Each request is assumed to have the same number of tasks to be completed. Each request is assumed to have a load time that is dependent on the dynamic content of the webpage and an associated service time.

3.1.3 Service Scheme

At the beginning of every time slots, threads from the request that completed execution are returned to the free pool of threads. The incoming requests are assigned threads for until request list is exhausted or threads in the thread pool are exhausted. Once a thread is allocated to a request it remains associated with the request until completion of execution. There can be time slots when threads are free and those when request are dropped. Both do not happen simultaneously. An extensive simulation was carried out with different distributions. The variations in terms of request drops are modeled and presented in Sect. 4.

3.1.4 Service Scheme Algorithm

The algorithm for modeling arrival:

Step 1: Generate the total number of arrivals for each time slot based on the random number generated by the distribution parameters.
Step 2: Assign the arrivals randomly to n categories as category1, category2 …category$n$ with:

$$\sum_{k=1}^{n}(\text{category1} + \text{category2} + \cdots \text{category}n)$$

It should be equal to a number of arrivals in the time slot category1 = rand()% number of arrivals.

3.1.5 Resource Allocation

Step 1: Initialize index and rejections as 0, Initialize T as maximum number of threads
Step 2: For the arrivals in timeslot allocate thread from the thread pool
Step 3: Update thread counter in the thread pool
Step 4: If not enough threads for allocation.

```
for index in value:
    if value[index]$\geq$ rejection threshold:
        rejection threshold −=value[index]
    else
        rejections +=value[index]
return rejections
```

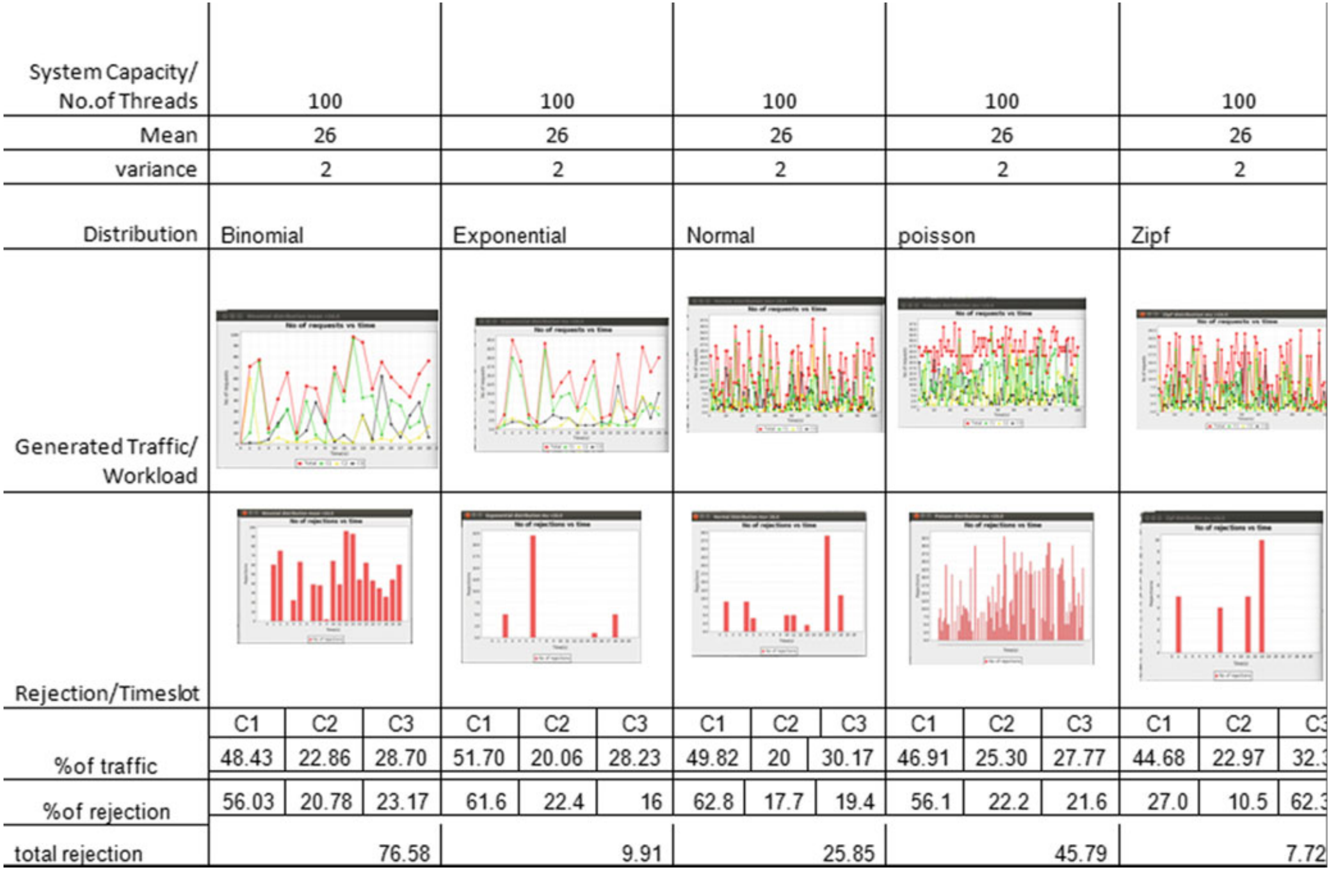
4 Results and Discussions

The effect of workloads based on different distributions was studied through extensive simulation process. The representative results for thread pool capacity 100 are presented in Fig. 2. The simulation run duration in each case was selected in such a way that all possible service request values appeared enough number of times to bring out all behavioral characteristics. The rejection of requests for different distributions is also presented in Fig. 2. To facilitate service differentiation, three categories of arrivals were considered (this could be used for priority scheduling). Automated service history collection which enables culling intelligent information out of it is also collected by the system.

4.1 Measuring Overhead in Dynamic Pages

Processing dynamic webpage requires additional page load time that includes loading time of images, audio video links etc., To understand the effect of this additional time a webpage with following specification (i.e.,) load time approximately 9.14 s etc., is presented in the table. Image loading time for a website is approximately 1–2 ms. As an example, the website "www.amrita.edu", the total no of requests is 151. The size of the page is 2.6 MB, load time is 9.14 s. Out of this 151 requests 94 (64.1%) requests are given for image. The download time for the images is approximately 1–2 ms each.

This paper clearly forms a useful contribution for assessing the impact of load on the web server for a selected system configuration. Our observations indicated that system model integration with the architecture can enable one to analyze the performance of an Internet service. High rejection rates indicate the need for an increase in the capacity of the system. While doing so, one should ensure that the resources are not underutilized. A low percentage of rejection rates with Zipf distribution indicate the identical behavior of incoming requests. The rejection rate under the Normal distribution shows the random occurrences of peak load in the traffic. The rejection rate under Binomial and Poisson distribution clearly demand scaling the system capacity.



System Capacity/ No.of Threads	100			100			100			100			100		
Mean	26			26			26			26			26		
variance	2			2			2			2			2		
Distribution	Binomial			Exponential			Normal			poisson			Zipf		
Generated Traffic/ Workload															
Rejection/Timeslot															
	C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3
%of traffic	48.43	22.86	28.70	51.70	20.06	28.23	49.82	20	30.17	46.91	25.30	27.77	44.68	22.97	32.
%of rejection	56.03	20.78	23.17	61.6	22.4	16	62.8	17.7	19.4	56.1	22.2	21.6	27.0	10.5	62.
total rejection	76.58			9.91			25.85			45.79			7.72		

Fig. 2 Table 1

5 Concluding Remarks

Modern web services have thrown up many unconventional challenges for monitoring QoS. Although ways for monitoring QoS parameters have been addressed extensively in the literature, the methodologies and techniques applied for creating workload models are strictly related to the objectives of the studies. With the aim of studying the effect of the movement of data in distributed systems in terms of response metric, the scheme proposed in the paper was implemented. The scheme enabled to understand the impact of different distributions on system performance (Completed vs. Rejected Services). Experimentation clearly revealed the effect of dynamic contracts in processing concurrent requests. Schemes like loading essential partial images rather than complete contents could be used to improve the QoS.

References

1. Calzarossa MC, Massari L, Tessera D (2016) Workload characterization: a survey. ACM Comput Surv (CSUR) 48(3):48
2. Galletta DF, Henry R, Mccoy S, Polak P (2004) Web site delays: how tolerant are users. J Assoc Inform Syst pp 1–28
3. Goncalves MA, Almeida JM, dos Santos LG, Laender AH, Almeida V (2010) On popularity in the blogosphere. IEEE Internet Comput 14(3):42–49
4. Gusella R (1991) Characterizing the variability of arrival processes with indexes of dispersion. IEEE J Sel Areas Commun 9(2):203–211
5. Gunther NJ (2001) Performance and scalability models for a hypergrowth e-commerce web site. In: Performance engineering, state of the art and current trends. Springer, London, UK, pp 267–282
6. Levene M, Borges J, Loizou G (2001) Zipfs law for web surfers. Knowl Inf Syst 3(1):120–129
7. Menasce D, Almeida V, Riedi R, Ribeiro F, Fonseca R, Meira W Jr (2000) In search of invariants for e-business workloads. In: EC 00: proceedings of the 2nd ACM conference on electronic commerce, New York, NY, USA. ACM, pp. 56–65
8. Mi N, Casale G, Cherkasova L, Smirni E (2008) Burstiness in multi-tier applications: symptoms, causes, and new models. In: Middleware 08Proceedings of the 9th ACM/IFIP/USENIX international conference on middleware, New York, NY, USA. Springer, New York, Inc., pp 265–286
9. Harini N, Padmanabhan TR (2012) A secured-concurrent available architecture for improving performance of web services. In: Communications in computer and information science, vol 292, no 1. Springer, pp 621–631
10. Harini N, Padmanabhan TR (2013) Admission control and request scheduling for secured-concurrent-available architecture. Int J Comput Appl 63(6):24–30