INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

Web Application Queuing Model

Jonathan Chukwuemerie Uzoh
Ph.D. Student, Department of Computer Science, University of Nigeria, Nsukka, Nigeria
Henry Onyebuchi Ossamulu
Ph.D. Student, Pan African Christian Theological University, Abuja, Nigeria
Hyacinth Chibueze Inyiama
Professor & Supervisor, Department of Electronic and Computer Engineering, Nnamdi Azikiwe, Nigeria

Abstract:
Web application down-times and delays are unpleasant experience undergone by the web users every day. There is need therefore to find a way of streamlining the process of entering and exiting the web environment and here web application queuing model (WebAPPnet) come in handy. There is need to continuously evaluate and reevaluate the bandwidth usage by the web users to avoid crashing of the system and avoiding unnecessary delays. When there is enough and constant availability of bandwidth carrying capacity in the system, the average waiting time will be greatly reduced. How to achieve this objective is what this paper is all about.

Keywords: *Web application, Down-time, Queuing model, Bandwidth and waiting-time*

1. Introduction

The importance of queuing model for the analysis of web application downtimes cannot be over emphasized, especially when the application is of national or global merit. The queuing specialist or analyst will design and implement good parameters to avoid any jam in the system due to in coming requests and processing of requests.

Queuing deals with one of the most unpleasant experiences of life. Queuing is quite common in many fields, for example, web base application, in telephone exchange, in a supermarket, at a petrol station, at computer systems, and so on. In web base application, if the bandwidth is less than what can carry the numbers of client, there is a very large tendency of creating unusual traffic or even crash of the application.

Considering the population potential tendency of clients waiting on a web server to be served, we may assume this potential to be finite or infinite. The application users' potential is finite if the rate of arrivals depends on the number of clients being served and waiting, but if the rate of arrival is not affected by the number of clients then it is infinite. Every web base system has a capacity which constitute the number of clients the server may be able to process. For an efficient web system, its mechanism should have a capacity to handle unlimited clients with successive inter-arrival times.

1.1. Web APP Queuing Model

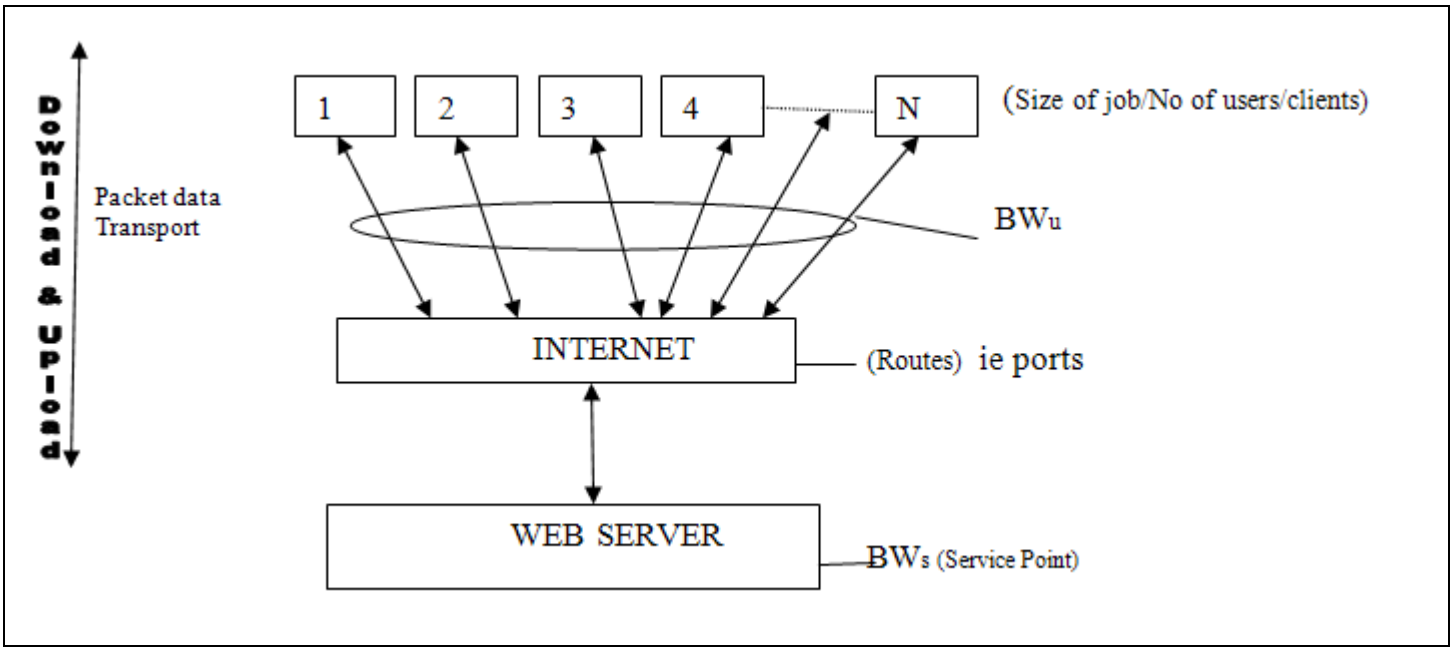

Figure 1: Network Queuing model for Web Base Application

BWu = Bandwidth used
BWs = Server available Bandwidth
Bandwidth Ratio = BWr = BWu/BWs
If BWr = 1, Network has Maximum Workload
If BWr < 1, Network is under utilized
If BWr > 1, Network traffic jam occurs
In a web queue, we should consider:
Time (T) between arrivals to the network queue
Size (U) of jobs/Number of users
Number (S) of servers
Considering the WebAPPnet project, there will be Many Users, Many arrivals Time and a single Server. Therefore, we present this as Mt/Mu/1s queuing model for WebAPPnet project (Figure 1, adopted from [1])
Where
Mt = Many arrival time
Mu = Many users
1s = One server
The web server has equal processor sharing capacity to all users or jobs in the WebAPPnet server
If the total number of clients in a Network remains constant, the Network is said to be a closed Network, but in the case of WebAPPnet, the web server is accessible to the general public so the size of job cannot be constant. So for WebAPPnet, it is called an open Network.
Web Traffic Model is a model of the data that is sent or received by a user's web browser.
Considering WebAPPnet, we adopt Poisson Traffic model. Poisson process is a model for circuit switch data as well as packet data. Looking at the many users with the WebAPPnet, the process will be a continuous Time Counting Process (TCP):

$$\{N(t), t \geq 0\}$$

1.2. Conditions to use Poisson Process

- N(0) = 0
- Number of occurrence counted in disjoint intervals are independent of each other (Independent Increment)
- The probability distribution of the number of occurrence counted in any time interval only depends on the length of time (Stationary Increment)
- The probability of distribution N(t) is the Poisson distribution with rate, λ and parameter λt
- No counted occurrence are simultaneous

Considering the conditions of the Poisson Process, the probability distribution of the waiting time until the next occurrence is an exponential distribution. The occurrences are distributed uniformly on any time interval.

1.3. Web Data Packet Queue/Traffic Mathematical Model

Inhomogeneous Poisson Process counts data transportation at a variable rate. Although the rate parameter may change over time.
Rate function is given as λ(t).

Expected Number of event between time *a* & *b* is

$$N a, b = \int_a^b \lambda(t)\, dt$$

Number (No.) of arrivals in the time interval [a, b] is given as N(b) –N(a)

P [N(b) – N(a) = K] = $\frac{e^{-N a,b}(N a,b)^k}{k!}$

K = 0, 1. -------

Rate Function = λ(t). This is workable in WebAPPnet because WebAPPnet is an open network and not a closed system..

1.4. Web Data in a Closed Network

The Poisson process of a closed Network is called Homogeneous Poisson model
The Rate Parameter = λ
Number of events in time interval = (t, t +٢)
Poisson distribution with associated parameter = λ٢

P [N(t + ٢) – N(t) = K] = $\frac{e^{-\lambda 2}(\lambda 2)^k}{k!}$

K = 0, 1. -------

Where N(t + ٢) – N(t) = K is the Number of events in time interval (t, t + ٢). This is not meant for our WebAPPnet network because it is for specified number of users at different time interval.

1.5. Key Element in a Web Queuing

- SERVER
- CLEINTS
- INTERNET(Routes)

1.6. Queuing Laws Applicable to Web Systems

- FIFO - First In First Out: who comes earlier leaves earlier
- LIFO - Last Come First Out: who comes later leaves earlier
- RS - Random Service: the client is selected randomly
- SPT – Shortest processing time first
- P -Service according to priority

The major aim of investigating queuing is to know the performance of the system. Considering the web, most of the queuing laws are very applicable. These laws include FIFO, RS and SPT. Although where these laws comes into play depends on the communication routes which as well depends on the bandwidth and frequency of the clients and server. This directly affects the processing, uploading and downloading of the web application system. The server processes randomly because the processing capacity of the clients may not be the same. This builds up inter arrival time of different units and the client in the same application with lesser loads may reach the server faster.

1.7. Actions during Web Services

When you use a web service, you have a client and a server (Figure 1). If the server fails, the client must take responsibility to handle the error. When the server is working again the client is responsible of resending it. If the server gives a response to the call and the client fails, the operation is lost.
You don't have contention, that is, if million of clients call a web service on one server in a second, most probably your server will go down [3]. You can't expect an immediate response from the server, but you can handle asynchronous calls.

1.8. Analysis of a Queuing System

To analyze a queuing system, we have to identify the possible properties of the incoming flow of requests, service times and service disciplines. The arrival process can be characterized by the distribution of the inter-arrival times of the clients.

In queuing theory these inter-arrival times are usually assumed to be independent and identically distributed random variables [2]. The other random variable is the service time. Sometimes it is called service request or work.
The service times, and inter-arrival times are commonly supposed to be independent random variables.
The structure of service and service discipline tell us the number of servers, the capacity of the system, that is, the maximum number of clients or clients staying in the system including the ones being under service. The service discipline determines the rule according to how the next client is selected.

1.9. Average Time Spent in the System per Client

1.9.1. Average Time Spent = P

$$\acute{P} = \frac{1}{N}\sum_{i=1}^{N} pi$$

Where P1, P2, …. PN are individual processing time for each N client.
For a stable system, $\acute{P} \longrightarrow P$ as $N \longrightarrow \infty$
An illustration of the analysis of queuing model:
Assuming there are 60,000,000 users on a Yahoo mail application queuing on the same server at the same time. If the summation of the time spent on all the clients is 3.25 years. Estimate the average time per a client.

1.9.2. Solution
Number of users = 60,000,000
Application type = web based
Total time = $\sum_{i=1}^{N} pi = 3.25\ yrs$
1 year = 365 days
31/4 years = 1186.25 days
1 day = 24 hours
Therefore 1186.25 days = 28,470 hours
1 hour = 60 minutes
Therefore 28,470 hours = 1,708,200 minutes
1 minute = 60 seconds
Therefore 1,708,200 minutes = 102,492,000 seconds
Then
$\acute{P} = \frac{1}{N}\sum_{i=1}^{N} pi$ = 102,492,000/60,000,000 = 1.7082 seconds
Therefore, the average processing time per client is 1.71 seconds.
The above illustration of the analysis queuing model tells us that the system is an open system and can handle unlimited clients. Obviously, the system capacity and bandwidth may always be underutilized which makes it better efficient.

2. Conclusion
Looking at the web queuing model as illustrated in figure 1, it is advisable to consider the number of clients on the system before selecting the host of the application. Definitely it will be cost determinant, so the system analyst should estimate the growing population on the application over a range of time in order to avoid the system from crashing. When the network model is open and the numbers of clients can access the server at inter-arrivals time without hitches, this defines the unlimited capacity and efficiency of the system traffic thereby reducing average waiting time in the queuing model.

3. References

i. A Tool For People-Centred Information Resource Management for the public sector by Uzoh, Jonathan.C, a Ph.d work with Nigerian Copyright Commission Reg. No. LW0580, 2014
ii. Basic Queueing Theory, Dr. János Sztrik, 2012
iii. http://stackoverflow.com/questions/2383912/message-queue-vs-web-services