Assessment of the Performance for E-Commerce Architecture vis-à-vis Advance Queuing Topology Algorithm (AQTA)

Riktesh Srivastava, Ph.D
Associate Professor
Information Systems
Skyline University College
University City of Sharjah
Sharjah, UAE

ABSTRACT
In this paper, we describe the formatting guidelines for IJCA Journal Submission. Impulsive stresses posed by unanticipated drifts of requests from users on Internet have made it exceptionally perplexing for E-Commerce to offer projected enactment of the architecture. In order to provide stable service, it is important for electronic commerce architecture to recognize the performances in terms of response time for requests, delay in accessing the services offered by servers implemented in E-Commerce architecture, and the number of requests waiting in the queue and thereby identifying the strategies to service the requests promptly. The correct and swift algorithm can prominently aid to build and operate electronic commerce architecture more proficiently.

In this paper, design and implementation of an algorithm named, AQTA is proposed, which uses the queuing models to prioritize the requests from different classes of users. The proposed algorithm takes the requests from clients, and uses three different set of algorithms simultaneously, namely, Classifier Algorithm, Priority Queue Algorithm and Final Queue Algorithm, to improve the performance of architecture. The comprehensive performance of the AQTA Algorithm is depicted using various Queuing models and the results are shown accordingly. The queue models which are implemented using the simulation study include the comparative study of the AQTA models using M/M/n/K and G/G/n/K models.

Keywords
AQTA, Classifier Algorithm, Priority Queue Algorithm, Final Queue Algorithm, M/M/n/K, G/G/n/K.

1. INTRODUCTION
The performance of any electronic commerce applications plays an indispensable part in alluring and holding consumers. The workload experienced by these applications tends to vary in a very dynamic way. The complication of the Electronic Commerce architecture collective with the massive deviations of the workload calls for the study for EC architecture. This paper introduces an algorithm, AQTA, to appraise the performance of EC architecture under ample load to animatedly monitor the performance levels of the system. The complete AQTA algorithm implemented using Java programming is actually combination of Classifier Algorithm (CA), Priority Queue Algorithm (PQA) and Final Queue Algorithm (FQA). These algorithms execute in concurrency for each requests being acknowledged and outlines the class of queue. By doing so, different classes of request are generated, which forms the base of AQTA algorithm.

The Classifier Algorithm (CA) is based on classification of customers' requests rather that the text classification of customer text. The CA designed followed the functional point analysis technique. PQA is combination of providing the web requests to the appropriate queue and then allocating the CDC for each class of queue. FQA is implementation of processing the web requests using either M/M/n/K or G/G/n/K algorithm. FQA founds its base on the study performed by [1]-[4].

2. ARCHITECTURE FOR AQTA IMPLEMENTATION
The complete architecture of the Electronic Commerce architecture for implementing the proposed AQTA algorithm includes 4 layers:

1) Web Request Management Server (WRMS)
2) Three Data Centers (DC) with 5 Virtual Machines (VM)
3) Application Servers(AS)
4) Database Servers (DS)

The network diagram demonstrated below intricate the structure for any Electronic Commerce architecture. The point worth noting here is that Web Request Management Server and Data Centers are implemented in the private cloud whereas Application Servers and Data Base Servers are implemented in the public cloud. The Web Request Management Server is the Web Server, implemented using Apache Tomcat, whereas, Data center includes VMware.

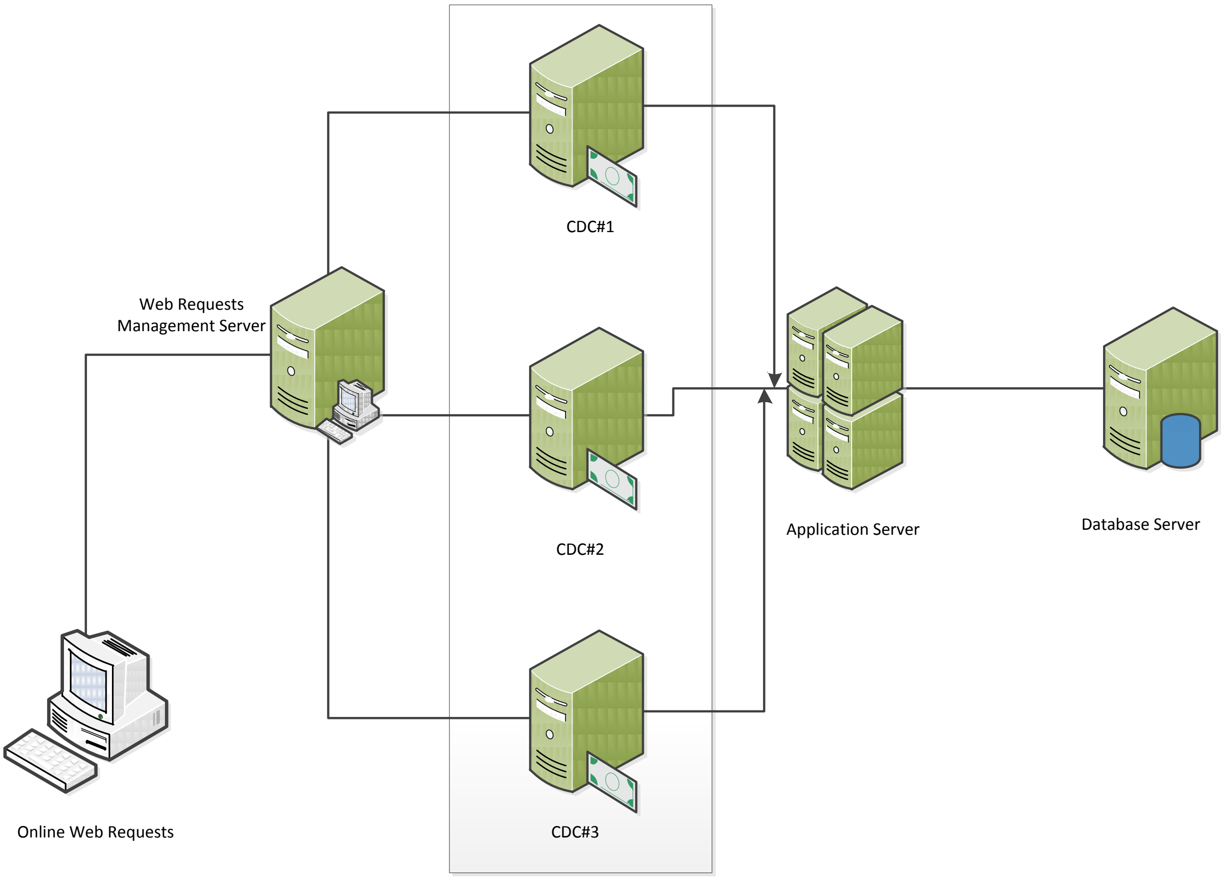


Fig 1: E-Commerce architecture for AQTA Algorithm

2.1 Web Request Management Algorithm

Web Request Management Algorithm executes at WRMS and is responsible for transferring the requests to CDC based on the processing time. The classification of the request is based on grain size. In AQTA algorithm, the three grain sizes are:

1. *Fine Grain:* If the request to be processed takes less than 5 seconds, based on the request type.
2. *Medium Grain:* If the request to be processed takes between 5 to 10 seconds, based on the request type.
3. *Coarse Grain:* If the request to be processed takes more than 10 seconds, based on the request type.

2.2 CDC Algorithm

CDC Algorithm functions at different CDC and for the study three different types of job scheduling mechanism are used, specifically:

CDC#1: As the job size necessitates less than 5 seconds, FCFS is used.

CDC#2: SSTF Algorithm is used here. For the enactment of the algorithm, the requests are organized based on job size from lowest to highest and then executed by the CDC.

CDC#3: SCAN algorithm is used in the Data Center 3. The algorithm is based on the concept that the maximum request size is taken into consideration and goes to the minimum level and again starts from smallest to largest request size.

2.3 Priority Queue Algorithm

The Priority Queue Algorithm is designed using Weighted Round Robin (WRR) Technique. The WRR algorithm uses the advantage of RR in eradicating the problem of starvation. It then apparatuses a priority-based performance of the requests. The configuration for the WRR is analogous to the RR in that it is merely a FCFS queue. A complete job queue cycle is once all the requests are processed in turn. In the situation that all requests are at the identical weight, FCFS is instigated. All requests in the queue are given time for processing in the method of a time slice, thereby jettisoning apprehension for starvation. This means that with WRR no requests can hold the Server for extended periods of time. Also short requests that get re-queued often will not have to wait for longer higher priority requests since all requests in the queue will be granted time.

The step-by-step procedure for PQA using WRR are as follows:

Step 1: All new requests gets entered into the queue.

Step 2: All requests are evaluated if an increase in weight is needed. This step is called Aging.

Step 3: New requests are assigned a time slice value according to weight.

Step 4: Job requests are reordered with higher weight at front

Step 5: All requests are given time on server

Step 6: Return to Step 1.

2.4 Final Queue Algorithm

The main experimentation for the proposed AQTA algorithm is the final queue algorithm. After the queue is formed to be processed, all the requests must be organized in the final queue. The aim of the final queue algorithm is that each request must be given significance to be processed. Amongst the various queue models available, the analytical study was the comparative study between M/M/n/K and G/G/n/K queue models. Figure 2 illustrates the specification of FQA in detail.

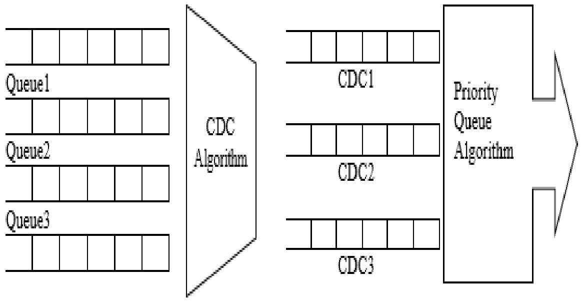


Fig 2: Final Queue Algorithm Specification

3. MATHEMATICAL REPRESENTATION OF SYSTEM

As discussed in section 2 of the paper, the comprehensive employment of the AQTA algorithm is tested using two different queuing models, namely, M/M/n/K and G/G/n/K. The performance of the system depends upon the response time from each the implemented queue models. The mathematical representation of the response time includes the response time for WRMS, CDC and PQA and thus can be mathematically represented as given in equation 1:

$$FQA_{RT} = t_{WRMS} + t_{CDC} + t_{PQA} \quad (1)$$

3.1 Response time for M/M/n/K queue model

Response time depends upon two factors to be analyzed, the response time generated for the requests arriving at WRMS and departing after being processed.

Based, on the study conducted by [5]-[11], the response time for M/M/n/K can be evaluated as:

$$R_{M/M/n/K \text{ (for arrival of requests)}} = RAND\ (Requests) \quad (2)$$

where RPT is the request processing time and b is any constant.

$$R_{M/M/n/K \text{ (for departure of responses)}} = RAND\ (Requests) \quad (3)$$

$$FQA_{M/M/n/K} = R_{M/M/n/K \text{ (for arrival of requests)}} + R_{M/M/n/K \text{ (for departure of responses)}}$$

$$FQA_{RT\ (M/M/n/K)} = RAND\ (Requests) + RAND\ (Responses) \quad (4)$$

3.2 Response time for G/G/n/K queue model

3 different algorithms for processing the requests for G/G/n/K models. These algorithms are:

a) Gaussian Algorithm
b) Equiprobable Algorithm
c) Negative-Exponential Algorithm

$$R_{G/G/n/K \text{ (for arrival of requests)}} = 1/3(t_{GA}+t_{EA}+t_{NEA}) \quad (5)$$

$$R_{G/G/n/K \text{ (for departure of responses)}} = 1/3(t_{GA}+t_{EA}+t_{NEA}) \quad (6)$$

$$FQA_{RT\ (G/G/n/K)} = R_{G/G/n/K \text{ (for arrival of requests)}} + R_{G/G/n/K \text{ (for departure of responses)}} \quad (7)$$

$$FQA_{RT\ (G/G/n/K)} = 2/3(t_{GA}+t_{EA}+t_{NEA}) \quad (8)$$

4. TECHNICAL SPECIFICATIONS OF THE ELECTRONIC COMMERCE SETUP

Web Request Management Server and three data centers forms the base of the complete experimental setup. For the experiment, Apache Tomcat 7 is used for Web Request Management Server. The challenging job for the study was to setup three Data Centers for the information flow. The specification for each of the Data Centers is:

CDC#1 ID: 0 Number of Processors: 10 VM Policy: TIME_SHARED	CDC#2 ID: 1 Number of Processors: 10 VM Policy: TIME_SHARED	CDC#3 ID: 2 Number of Processors: 10 VM Policy: TIME_SHARED

The public cloud (Data Center 1) was used to set up Application and Data Base Server and was based in US. The other two private clouds (Data Center 2 and Data Center 3) are based in Sharjah within the University premises. There are 5 VM used in two Data Centers 2 and 3, which handles 300 requests per minute and buffer size is 1000 Bytes for each request.

4.1 Load Balancing Policy for the Data Center 2 and Data Center 3

Data Center 2 followed Round Robin technique, wherein, the controller allocates the requests to a list of VMs on a turning basis. The first request is assigned to a VM- selected randomly from the group and then the controller disperses the successive requests in a circular order. Once the VM is consigned the request, the VM is moved to the end of the list. In this technique; there is a better allocation concept known as Weighted Round Robin Allocation in which controller assigns a weight to each VM so that if one VM is capable of handling twice as much load as the other, the powerful server gets a weight of 2. In such cases, the controller assigns two requests to the powerful VM for each request assigned to a weaker one. The major issue in this allocation is this that it does not consider the advanced load balancing requirements such as processing times for each individual requests. Thus the choice is to use RRLB technique for Data Center 2.

Data center 3 uses Throttled Load Balancer (TLB) allocation policy. The TLB preserves a record of the state of each virtual machine (busy/ideal). If a request arrived regarding the allocation of virtual machine, the TLB sends the ID of ideal virtual machine to the data center controller and data center controller allocates the ideal virtual machine.

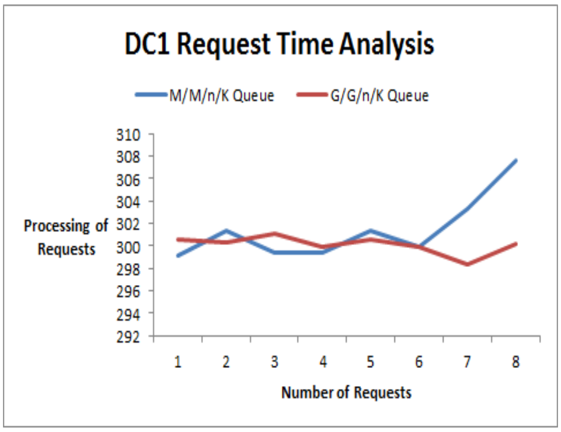
5. RESULTS AND ANALYSIS

Table 1 demonstrates the experiment outcomes for each of the Data centers w.r.t the number of requests arriving per minute. The table is illustrative in nature and is self-explanatory.

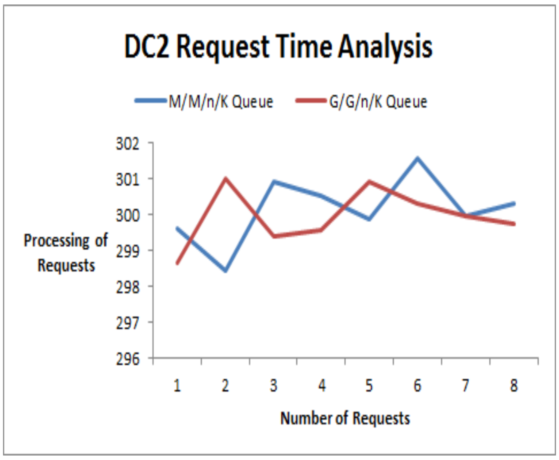
Table 1: Experiment Outcomes

Number of Requests arriving per minute	M/M/n/K Queuing Model			G/G/n/K Queuing Model		
	Processing time at DC1 /minute	Processing time at DC2/minute	Processing time at DC3/minute	Processing time at DC1 /minute	Processing time at DC2/minute	Processing time at DC3/minute
250	299.143	299.614	297.612	300.619	298.637	296.897
300	301.316	298.424	299.781	300.337	300.99	300.012
350	299.391	300.918	301.521	301.09	299.364	297.905
400	299.366	300.509	300.667	299.955	299.536	300.545
450	301.408	299.85	299.984	300.601	300.89	298.997
500	299.962	301.545	301.666	299.957	300.297	300.001
550	303.333	299.937	302.351	298.422	299.965	300.231
600	307.562	300.281	298.983	300.197	299.742	296.887

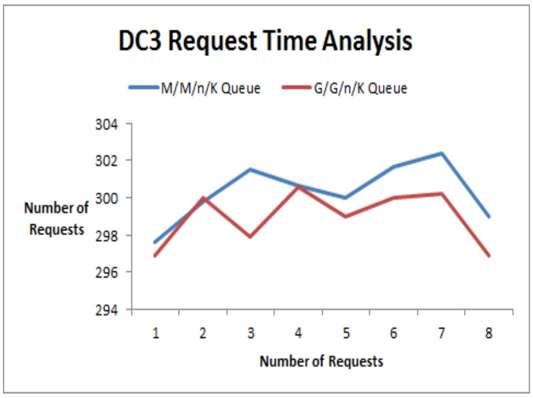
The analysis of the results is displayed in the following three charts of Figure 3.



CDC#1: Request Time Analysis



CDC#2: Request Time Analysis



CDC#3: Request Time Analysis

6. CONCLUSION

Based on the outcomes cited in section 5 of the research paper, for CDC#1 and CDC#2, the response time is practically the same for the lower number of requests received at the web request management server. However, G/G/n/K model was found to be more appropriate for greater quantity of requests arriving at the architecture. For CDC#3, both the models were found correspondingly apposite for the number of requests arriving. As a commendation of AQTA algorithm, the employment of AQTA algorithm using G/G/n/K was found to be more seemly under the given conditions.

7. REFERENCES

[1] Riktesh Srivastava, "Memory Estimation of Internet Server via M/G/1, G/M/1, G/G/1 Queuing Models: A Comparative Perspective ",6th Annual ISOne World Conference, Emerging Academia and Enterprise Agendas, ISOne World, Las Vegas,USA., 2007.

[2] Riktesh Srivastava, "Design and Implementation of G/G/1 Queuing Model Algorithm w.r.t. it's a applicability for Internet Gateway Server, ", International Arab Journal of Information Technology (IAJIT), pp.367-374, 2007.

[3] Riktesh Srivastava, "Estimation of Web Proxy Server Cache Size using G/G/1 Queuing Model", International

Journal of Applied Research on Information Technology and Computing (IJARITC). pp.46-58, 2010.

[4] Riktesh Srivastava, “Analysis Of Job Scheduling Algorithm For An E-Business Model In A Cloud Computing Environment Via GI/G/3/N/K Queuing Model”, International Journal of Advancements in Technology, pp. 215-229, 2012.

[5] Riktesh Srivastava, “Estimation of Web Proxy Server Cache Size using G/G/1 Queuing Model”, International Journal of Applied Research on Information Technology and Computing (IJARITC).

[6] Riktesh Srivastava, “Mathematical Analysis of Electronic Commerce Architecture using Queuing Theory”, December 31 2010-Januray 4 2011, International Conference on Mathematics of Date, Allahabad, INDIA.

[7] Riktesh Srivastava, “Memory Estimation of Internet Server via M/G/1, G/M/1, G/G/1 Queuing Models: A Comparative Perspective “,6th Annual ISOne World Conference, Emerging Academia and Enterprise Agendas, ISOne World, Las Vegas,USA., 2007.

[8] Riktesh Srivastava, “Analysis Of Job Scheduling Algorithm For An E-Business Model In A Cloud Computing Environment Via GI/G/3/N/K Queuing Model”, International Journal of Advancements in Technology, pp. 215-229, 2012.

[9] Riktesh Srivastava, “Analysis of Mobile Agent based e-Supply Chain Management system using Queuing Theory: A comparative study between M/M/1 and M/D/1 models”, International Journal of Research in Computer Application and Management, (IJRCAM).

[10] Riktesh Srivastava, “Analysis of job scheduling algorithm for an e-Business model in a Cloud Computing Environment via GI/G/3/n/k Queuing Model”, November 4-5, 2011 International Conference on Business and Technology, Organized jointly by IAMS and FRI University, INDIA.

[11] Riktesh Srivastava, “Mathematical model using BI for improving E-Commerce Businesses applicability’s”, The International Research Journal of IT and Science Management (IRJITSM).

[12] Riktesh Srivastava, Implementation of Job Scheduling Algorithms in a Cloud Computing Environment for B2C Electronic Commerce architecture using GI/G/1 Queuing Model, International Journal of Computer Applications (IJCA).

[13] Riktesh Srivastava, Evaluation of Response Time using Gang Scheduling Algorithm for B2C Electronic Commerce architecture implemented in Cloud Computing Environment by Queuing Models, International Journal of Future Computer and Communication (IJFCC).

[14] J.P.Buzzen, A Queueing network model of MVS, ACM Computing surveys, Sept 1978, Vol. 10, No 3, pp-319-331.

[15] Anderson, Darrell et. al.(1999), “A Case for Buffer Servers”, pp. 82-88, IEEE Seventh Workshop on Hot Topics in Operating Systems.

[16] Dimitri Bertsekas, Robert Gallager, Data Networks, Second Edition, Prentice Hall of India Private Limited, 1997.

[17] M. Arlitt and C. Williamson, "Internet Web Servers: Workload Characterization And Performance Implications", IEEE/ACMTransactions on Networking, Vol. 5, No. 5, pp. 631-645, Oct. 1997.

[18] A. Iyengar, et al., "High-Performance Web Site Design Techniques", IEEE Internet Computing, Vol. 4, No. 2, pp. 17-26, 2000.

[19] Kishor S.Trivedi, Probability & Statistics with Reliability, Queuing, andComputer Science Applications. Prentice-Hall of India Private Limited. New Delhi-110 001 2004.