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.

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.
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 appurtenances 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.

![Figure 2: Final Queue Algorithm Specification](image)

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 of 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:

\[
FQA_{RT} = FQA_{WRMS} + FQA_{CDC} + FQA_{PQA}
\]

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:

\[
R_{M/M/n/K} \text{ (for arrival of requests)} = \text{RAND (Requests)}
\]

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

\[
R_{M/M/n/K} \text{ (for departure of responses)} = \text{RND (Responses)}
\]

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

\[
FQA_{RT} = FQA_{M/M/n/K} \text{ (for arrival of requests)} + \text{RND (Responses)}
\]

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

\[
R_{G/G/n/K} \text{ (for arrival of requests)} = \frac{1}{3}(GA+HA+NEA)
\]

\[
R_{G/G/n/K} \text{ (for departure of responses)} = \frac{1}{3}(GA+HA+NEA)
\]

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:

<table>
<thead>
<tr>
<th>CDC#1</th>
<th>CDC#2</th>
<th>CDC#3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID: 1</td>
<td>ID: 1</td>
<td>ID: 2</td>
</tr>
<tr>
<td>Number of Processes: 10</td>
<td>Number of Processors: 10</td>
<td>Number of Processors: 10</td>
</tr>
<tr>
<td>TIME_SHARED</td>
<td>TIME_SHARED</td>
<td>TIME_SHARED</td>
</tr>
</tbody>
</table>

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

Table 1: Experiment Outcomes

<table>
<thead>
<tr>
<th>Number of Requests arriving per minute</th>
<th>M/M/n/K Queuing Model</th>
<th>G/G/n/K Queuing Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing time at DC1 /minute</td>
<td>Processing time at DC2/minute</td>
</tr>
<tr>
<td>250</td>
<td>299.143</td>
<td>299.614</td>
</tr>
<tr>
<td>300</td>
<td>301.316</td>
<td>298.424</td>
</tr>
<tr>
<td>350</td>
<td>299.391</td>
<td>300.918</td>
</tr>
<tr>
<td>400</td>
<td>299.366</td>
<td>300.509</td>
</tr>
<tr>
<td>450</td>
<td>301.408</td>
<td>299.85</td>
</tr>
<tr>
<td>500</td>
<td>299.962</td>
<td>301.545</td>
</tr>
<tr>
<td>550</td>
<td>303.333</td>
<td>299.937</td>
</tr>
<tr>
<td>600</td>
<td>307.562</td>
<td>300.281</td>
</tr>
</tbody>
</table>

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


