Performance Engineering

We are going to upgrade a software system that executes web requests from users from the Internet. The system offers an imageboard service to the users.

We have observed the running system for 200 minutes. During these 200 minutes, we have seen that 18000 execution requests have been completed. Actually, we have seen that requests arrived at a rate of 90 requests per minute; and therefore, we can deduce that all requests that arrived were completed.

The system is composed of four service centers: A *WebServer,* a *ContentDeliveryServer*, a *ContentUpdoadServer*, and a *SecurityAndLoggingServer.* We are interested in knowing the average service demand D_k of each service center for a user request, but we could not directly measure it. However, we have been able to measure the following:

- Each user's request executes once in the *WebServer*. There is only one resource for the *WebServer.* We have seen that the *WebServer* has been busy 900 seconds. After the *WebServer* executes, a request can follow two different paths, depending on whether the user wanted to see a full page of images or wanted to upload a new image.
- If the requester wanted to see a full page of images, which happened 70% of times, the request executed in the *ContentDeliveryServer* iterating 11 times its execution in this server (so, an average page has 11 images). We also observed that there are two resources for executing the *ContentDeliveryServer* and that the average utilization of the server was 57.75%.
- If the requester wanted to upload an image, which happened the 30% of times, the request executed in a *ContentUpdoadServer*. The service time of the *ContentUpdoadServer* is 300ms. The *ContentDeliveryServer and ContentUpdoadServer* work independent of each other (thus, all the activities to synchronize the images between the upload and delivery are out of the scope of this assignment).
- Finally, each request (regardless of whether it was an imageboard download or upload request) needed to execute in the *SecurityAndLoggingServer,* where it has to pass a security check of the request and leave a trace of its execution and contents for possible future investigations by authorities or user preferences analysis. We measured that its average utilization was 90%.

You have to submit a PDF document with our answers to the following 3 exercises:

A) Use the operational laws to calculate the Service Time D_k of each of the four service centers.

B) We would like to upgrade our previous system to make it faster. We believe that the bottleneck of the system execution is in the *SecurityAndLoggingServer*. We want to increment the power of the server by adding 2 more resources to it (therefore, in total it will have 3 resources). Model the upgraded System using Queueing Networks (in JMT or in your preferred Queueing Network simulation engine). Add screenshots of: the structure of the network, all the information about the *ContentDeliveryServer* (this will need 2 screenshots) and the information about the *SecurityAndLoggingServer*. Simulate the model to calculate the System Response Time when the *SecurityAndLoggingServer* has 3 resources and show a screenshot of the response time result.

Hint1: In the cases that, from a service center (e.g., *WebServer*) a job can go to more than one service center, use Probabilistic Routing.

Hint2: When a request iterates X times in a group of service centers, you can model it adding an additional loopback arc (see slide 34 of the first Performance lesson), having the loopback arc probability (X-1)/X and the arc that leaves the service center probability 1/X.

Hint3: Use the exponential distribution for all times and rates (frequencies) you need to model.

C) Model the system in the previous exercise with a UML Activity Diagram profiled with MARTE. You are **ONLY** required to represent with MARTE profile the 4 next pieces of information: the workload of the system, the time that the *WebServer* needs to execute a request, the probability of going to the *ContentDeliveryServer,* and the probability of going to the *ContentUpdoadServer.* Write the Stereotypes you need to use, the Properties you use from them, and their Values as comments attached to the corresponding elements (like the examples in the slides that we saw during the lesson). You can use your favorite tool for modeling UML Activity Diagrams. Take a screenshot of the diagram and paste it as answer to the exercise (obviously, including in the comments where you have added the performance information using MARTE!)