Performance Engineering - retake

Performance Engineering

We are going to study the performance of software system that executes web sessions from users from the Internet. The system offers a betting service to the users.

We have observed the running system during 1 regular week. During that week, we have seen that 806400 user sessions have been completed. Actually, we have seen that sessions arrived at a rate of 80 sessions per minute; and therefore, we can deduce that all sessions that arrived were completed.

The system is composed of four service centers: A *WebServer,* a *WinnerPaymentServer*, a *PlayerEngagementServer*, and a *BettingServer.* The description of the system and measured information is the following:

1. Each user's session starts executing in the *WebServer*. There is only one resource for the *WebServer*, whose service time is 50ms in average. After the *WebServer* executes, the user session can follow two different paths, depending on whether the user is the winner of a previous bet or not.
2. If the user is a winner of a previous bet, which happened 20% of times, the user executed in the *WinnerPaymentServer* and credit is transferred to the user's account. The utilization of the *WinnerPaymentServer* is 20% and the service time of a job is 300ms. After transferring the credit, the session continues in the *PlayerEngagementServer*, which offers new betting possibilities to the user in events that will happen in the next 30 minutes.
3. If the user is not a winner of a previous bet, which happened 80% of times (because the user lost the previous bet or because the user did not bet the last time), the user session executed directly in the *PlayerEngagementServer.* This server checks all the possible bettable events that will be resolved in the next 30 minutes and generates a list with the events that are most likely to captivate the user to do a new bet. This kind of "recommendation" service needs a lot of CPU to discover the preferences of the user, for which there are 3 servers working in parallel. The service time of a job in the *PlayerEngagementServer* are 500ms.
4. After executing in the *PlayerEngagementServer* , the user may have been convinced to bet on one of the elements in the list (which happens the 60% of times) or may finish his/her session and leave the system (the 40% of times). If he/she has been convinced to bet again, the session continues its execution in the *BettingServer*.
5. The *BettingServer* executes the bet of the user in the system. After the *BettingServer* execution, the user waits 15 minutes in average (the time until the finishing moment of the event in which the bet has been done) and the user session executes again in the *WebServer* following the description in item "1)". There is a single resource for executing the *BettingServer,* which has been found busy 40% of time. We have also observed that there were, in average, 0.6666 jobs in the *BettingServer;* and that, the average time between a job arrived to the *BettingServer* and its execution was completed (including the time it spent waiting for service in a queue) was 0.33333 seconds.

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

A) Use the operational laws to calculate the average number of visits V_k to the *WinnerPaymentServer* for each user session and the service time S_k of the *BettingServer.*

B) Model the System using Queueing Networks (in JMT or in your preferred Queueing Network simulation engine). Add screenshots of: the structure of the network and about all the information you add to each component (service times, routing probabilities, number of resources, etc.) Simulate the model to calculate the Utilization and Throughput of each of the four components in the system and show screenshots of the results.

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

Hint3: The part "the user waits 15 minutes in average" can be modeled in the Queueing Network as an *Infinite Server* (a.k.a. *Delay*).

C) Model the system in exercise "B)" with UML Diagrams profiled with MARTE. You are ONLY required to represent with MARTE profile the 3 next pieces of information: the workload of the system, the time that the *PlayerEngagementServer*needs to execute, and the number of resources for the *PlayerEngagementServer.* 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 (obviously, including in the comments where you have added the performance information using MARTE).

Good luck!

Submission status

Attempt number	This is attempt 1.	
Submission status	Submitted for grading	
Grading status	Graded	
Due date	Friday, 3 April 2020, 11:59 PM	
Time remaining	Assignment was submitted 10 mins 12 secs early	
Last modified	Friday, 3 April 2020, 11:48 PM	
File submissions, max 100 MB per file	rq222ah_2dv608.pdf Opt-out URKUND	3 April 2020, 11:48 PM
Submission comments	Comments (0)	

Feedback

Grade for assignment	43.5 (43.5 %)	
Graded on	Tuesday, 21 April 2020, 11:56 AM	
Graded by	 Diego Perez Palacin	
Feedback files	rq222ah_2dv608.pdf	21 April 2020, 11:56 AM

◄ Performance Engineering

Architecting and Design Forum ►