Simulating a Single-Server Queue using the Q – Simulator

Irene K. Amponsah, Bennony K. Gordor, Francis Dogbey

Abstract—This paper introduces a technique for simulating a single-server exponential queuing system. The technique called the Q-Simulator is a computer program which can simulate the effect of traffic intensity on all system average quantities given the arrival and/or service rates. The Q-Simulator has three phases namely: the formula based method, the uncontrolled simulation, and the controlled simulation. The Q-Simulator generates graphs (crystal solutions) for all results of the simulation or calculation and can be used to estimate desirable average quantities such as waiting times, queue lengths, etc.

Keywords—Automation system-Simulator, Simulation, Single-server exponential system

I. INTRODUCTION

ESTIMATION of waiting times and other steady-state average quantities at service points such as banks, airports, restaurants, theatres, post offices, and traffic lights has always been an objective in the study and analysis of queues. Based on models developed, statistical simulation can be performed. Statistical simulation is driven by a stream of randomly generated instructions, based on statistics collected during a single detailed simulation. This method can give accurate performance estimates within minutes, allowing a large design space to be simulated quickly. And of course, real data is always used to validate the results of the simulation.

A common approach to estimating these quantities is to develop statistical models to estimate and write computer programs to stimulate queues in order to find optimal solutions. S. Nussbaum and J. E. Smith (2002),[4] did prior work that applied this technique to superscalar processors. They evaluated the extension of statistical simulation to Symmetric Multiprocessing (SMP) systems. Key program parameters are identified, and program statistics are collected during detailed simulations for both multi-programmed workloads (SpecInt) and parallel scientific workload (Splash-2). The accuracy of statistical simulation is evaluated at different levels of model detail, and it is shown that for multi-programmed workloads a 10% average error can be achieved, and for parallel benchmark programs 15% average error can be achieved. K. Salah et al (2007), [5] also developed a performance analysis and comparison of interrupt –handling schemes in gigabit net.

Irene K. Amponsah(nee Vorsah) is with the Department of Mathematics and Statistics, School of Physical Science, University of Cape Coast-Ghana (e-mail: Iamponsah@ucc.edu.gh).

Prof Bennony K. Gordor (Rtd) was with the Department of Mathematics and Statistics, School of Physical Science, University of Cape Coast-Ghana, (e-mail: benkgordor@yahoo.co.uk).

Mr. Francis Dogbey is with the Ghana-India Kofi Annan Centre for Excellence on ICT- Osu Accra, Ghana (e-mail: fdoggy2002@yahoo.co.uk).

Their analysis and simulation work gave insight into predicting the system performance and behavior when employing a certain interrupt-handling scheme. Similarly, Shaul K. Bar-Lev et al (2007), [6] conducted a research on the applications of bulk queues to group testing models with incomplete identification. They discussed some assumptions that gave rise to the bulk queuing model M/G $(m, M)/1$, where m and $M > m$ are the decision variables where each batch size can be between m and M. They computed the optimal values of the decision variables (m, M) that maximize the expected profit. Abolnikov et al (1993), [1] also in a related publication, researched on a multilevel control bulk queuing system with vacationing server. This article studied a Markov chain describing the evolution of the queue in a general single-server bulk queuing system with continuously operating (or, equivalently vacationing) server, semi-Markov modulated compound Poisson input, queue length dependent service time, and multilevel control server capacity. The studies established a necessary and sufficient criterion for ergodicity of this Markov chain and found its stationary distribution. As an example a single-server queuing system was considered, and some explicit results were obtained in this case. Kumar et al (2002),[3] carried out a research concerned with the analysis of a single-server queue with Bernoulli vacation schedules and general retrial times. It was assumed that the customers, who found the server busy, were queued in the orbit in accordance with an FCFS (first-come-first-served) discipline and only the customer at the head of the queue was allowed access to the server. They first presented the necessary and sufficient condition for the system to be stable and derived analytical results for the queue length distribution, as well as some performance measures of the system under steady-state condition. They showed that the general stochastic decomposition law for M/G/1 vacation models holds for the present system also. Some special cases were also studied. These and other related studies can be found on stochastic queues and their applications.

However, there are a number of few direct queue simulators. For example, an applet (University of Malta, 2010), [7] simulates a queuing situation where it starts with a newly opened service with no one being served or waiting. The arrival rate and service rates are fixed and once you start the simulation, an indicator shows how many customers are currently in the system, updated once each simulated second. Another example of a software named Simulator #2 (Baker, 2000), [2] requires that you provide the distribution of the Arrival Intervals, the discipline, the distribution of the service duration and the simulation controls.

This paper presents a simulator that simulates the simplest of queues that is found around us, i.e. the M/1/1. This simulator is named the Q-Simulator. What sets this simulator and the others apart is its ability to find the characteristics that is optimal for setting up an effective single-server queuing system. This is achieved by the simulator's ability to examine the effects of traffic intensity on steady-state average quantities, given the arrival rate/or the service rate.

II. STRUCTURE OF THE Q-SIMULATOR

The Q-Simulator, software developed for this research is to specifically automate single-server queuing process. The software allows the user to specify the arrival rate and the service rate either in seconds, minutes, or hours. However, these quantities are recalculated in minutes; this is because it is more practicable where exponential and poison distributions are concerned. The Q-Simulator has three options; namely: formula based method, uncontrolled simulation and the controlled simulation.

III. OPERATING THE Q-SIMULATOR

- The Q-Simulator Installation (*optional*)

Follow the instruction in the readme file on the storage media containing the installation file.

- On the task bar, Click on Start then Program Files then Q-Simulator (This will display a screen shown in Figure 1).
- From this screen, specify the time unit of the computation; which may be seconds, minutes, or hours.
- Select the *Task* menu to display the screen shown in Figure 2.
- Select appropriate options (e.g. Simulation, Controlled).
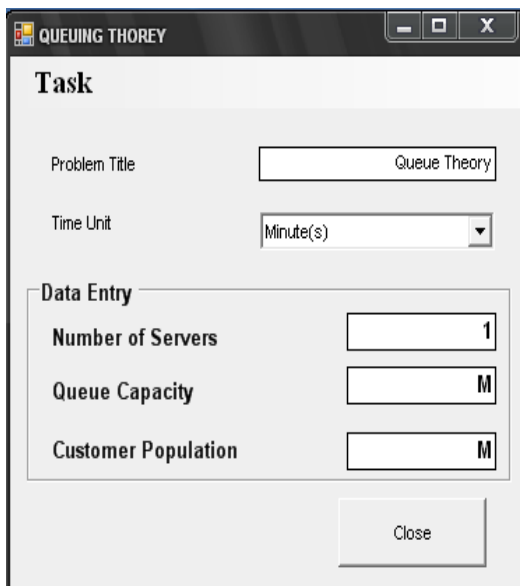
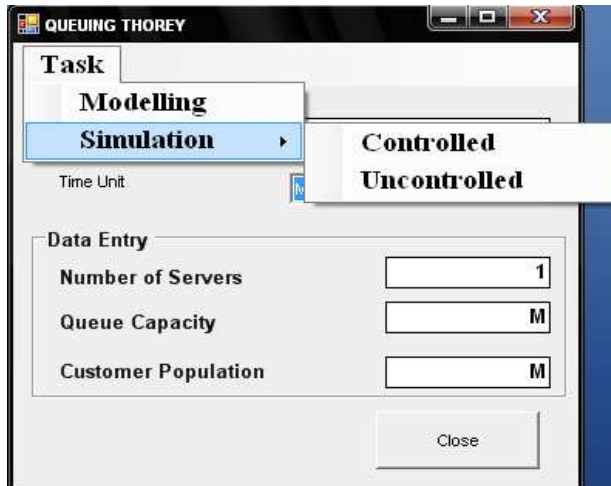

Fig. 1 Interface One

Fig. 2 Interface Two

A. Using Formula Based Modeling/ Analytical Method

- Select Modeling from Figure 2.
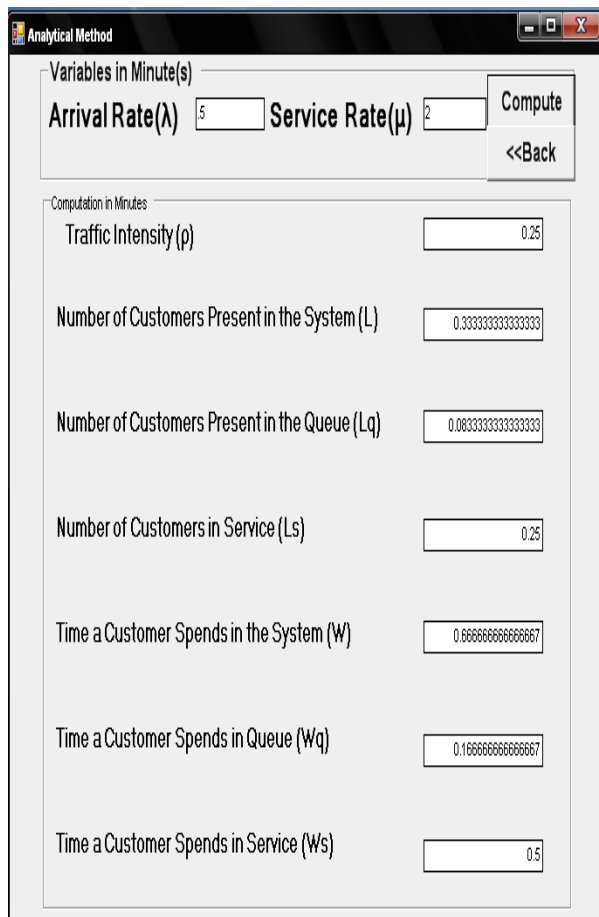 This will display the screen shown in Figure 3.

Fig. 3 Analytical Method Interface

• Enter values for Arrival Rate and Service Rate

• Click on *Compute* button to calculate the average quantities or *<<Back* to Cancel the effect.

The formula based method calculates the average quantities based on a known inter-arrival rate, λ and service rate μ. The process assumes that the researcher computes λ and the μ from the data. Then, by inputting these values the average quantities are obtained.

Fig. 4 Controlled Simulation Interface

B. Using Controlled Simulation

• Select Simulation and then Controlled from Figure 2. This will display the screen shown in Figure 4.

• Select the Variable to be controlled. i.e. Service Rate or Arrival Rate

• Supply the parameters for the scale of the Traffic Intensity and Controlled variable.

• Click on the *Simulate* button to generate the average quantities or *Back* button to leave this screen.

• Click on *Chart* button to display the chart corresponding to the set of values currently in the database. This will display the graph in Figure 5.
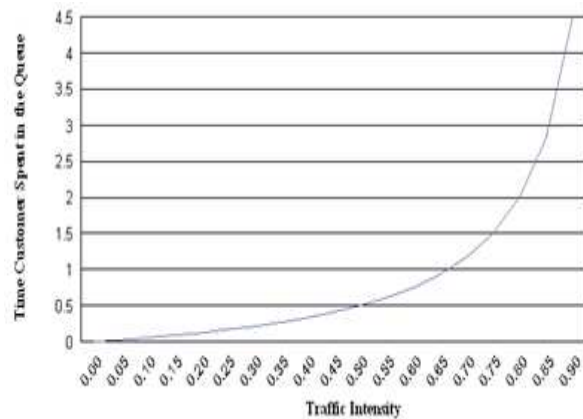

Fig. 5 Effect of Traffic Intensity on Time Customer Spends in Queue

It must be noted that all the system average quantities have their corresponding graphs plotted against traffic intensity. The researcher has the prerogative to choose from this list as regarding the interest at hand.

The controlled simulation observes most of the assumptions of a single server queuing system. The assumptions include $0 \leq \rho < 1$ and $\mu > \lambda$. In the controlled simulation, one is expected to provide the scale of ρ, indicating how ρ values are increased between [0, 1]. In addition, either μ or λ is specified which is used together with a ρ value to determine λ or μ respectively. The computed average system quantity therefore translates into the desired graphs.

C. Using Uncontrolled Simulation

• Select Simulation and then Uncontrolled from Figure 2, this will display Figure 6.

• Specify the range of values for Service Rate or Arrival Rate

• Click on the *Simulate* button to generate the average quantities or *Back* button to leave this screen

• Click on *Chart* button to display the chart corresponding to the set of values currently in the database. This will display a screen similar to Figure 7.
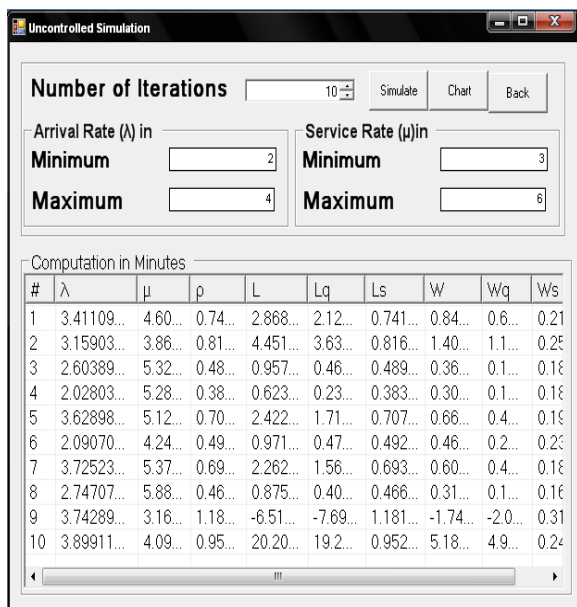
Fig. 6 Uncontrolled Simulation Interface

The uncontrolled simulation randomly generates λ and μ within a specified range for a specified number of iterations. These randomly generated λ and μ values are then used to compute the rest of the system average quantities.

This process is useful in simulating real life situations where there is the need to study the effects of uncontrolled λ and μ on other system average quantities. It is worth noting that, this process randomly estimates λ and μ purely based on the specified range and as a result, most of the basic assumptions of a queuing system are not observed. The essence is to provide an opportunity to verify the basic assumptions of queuing theory for single server exponential queues. Although many of the average quantity computations are reasonable, the associated graph drawn based on these quantities are somewhat unrealistic.

A review of this paper in future will highlight queues in complex systems using the Q – Simulator.

REFERENCES

[1] L. M. Abolnikov, J. H. Dshalalow and A. M. Duhovny "A multilevel control bulk queuing system with vacationing server," in *Operations Research Letters*, vol.13.3, 1993, pp. 183–188.
[2] S. L. Bakers, "Queue simulation demo",2000,(Assessed 9/17/2010) http://hspm.sph.sc.edu/COURSES/J716/demos/StudentLines/Queues.html1
[3] Kumar, B. Krishna, D. Arivudainambi, A. Vijayakumar. The M(x)/G/1 retrial queue with Bernoulli schedule and general retrial times, Pacific Journal of Operational Research, 2002 , http://findarticles.com/p/articles/mi_qa5473/is_200211/ai_n21323383/
[4] S. Nussbaum, J. E. Smith "Statistical Simulation of Symmetric Multiprocessor Systems" 35th Annual Simulation Symposium http://www.ece.wise.edu/
[5] K. Salah, K. El-Badawi and F. Haidari , "Performance analysis and comparison of interrupt-handling schemes in gigabit net. Computer Simulation, vol. 30.17, 2007, pp. 3425–3441.
[6] Shaul, K. Bar-Lev, M. Parlar, Perry, D, Stadje, W. Van Der Duyn, F. A. Schouten, "Applications of bulk queues to group testing models with incomplete identification " *European Journal of Operational Research,* vol.183.1, 2007, pp.226–237
[7] University of Malta / Department of Statistics and OR Tutorial. 2010. (Assessed9/17/2010) http://staff.um.edu.mt/jsk11/simweb/sq2/sq2.htm1