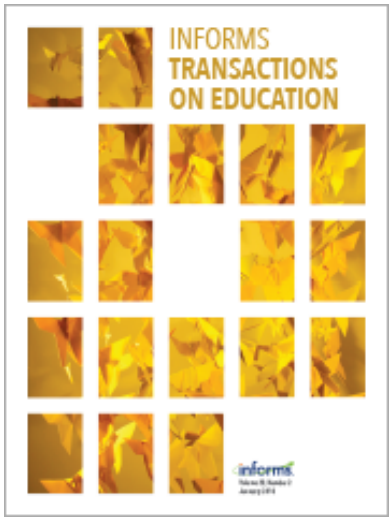


INFORMS Transactions on Education

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Simpler Spreadsheet Simulation of Multi-Server Queues

Thin-Yin Leong,

To cite this article:
Thin-Yin Leong, (2007) Simpler Spreadsheet Simulation of Multi-Server Queues. INFORMS Transactions on Education 7(2):172-177. https://doi.org/10.1287/ited.7.2.172

Vol. 7, No. 2, January 2007, pp. 172–177
ISSN 1532-0545 | 07 | 0702 | 0172
INFORMS Transactions on Education
DOI 10.1287/ited.7.2.172

Simpler Spreadsheet Simulation of Multi-Server Queues

Thin-Yin Leong
School of Information Systems, Singapore Management University, 80 Stamford Road, Singapore S178902, tyleong@smu.edu.sg

Process-driven spreadsheet queuing simulation provides a clear and intuitive approach for students in business modeling courses to learn about queue behavior. With basic spreadsheet skill, some guidance on the generation of random variates and simple assumptions like "service in arrival order," it is not difficult to construct a single server spreadsheet simulation model. However to extend the model to multiple servers, add-ins, macros, or multiple templates (each for a given fixed number of servers) are needed. Hora (2003) showed that array formulas can be used instead, to create the G/G/c spreadsheet model with parametrically adjustable number of servers. This is a great improvement, but array formulations are still inherently complex and therefore not easy for students to comprehend and apply. In this paper, I show that non-array formulations can achieve the same effect, effectively simplifying the model.

1. Introduction

Spreadsheets are very user-friendly and yet powerful tool for learning how to model real business problems (Winston 2004). In particular, building process-driven queuing simulation models (Grossman 1999) have been found to be good end-user modeling exercises to help students understand about queues. The process-driven approach can present queue activity clearly and intuitively. (e.g. graphical representation in Ingolfsson and Grossman 2002.)

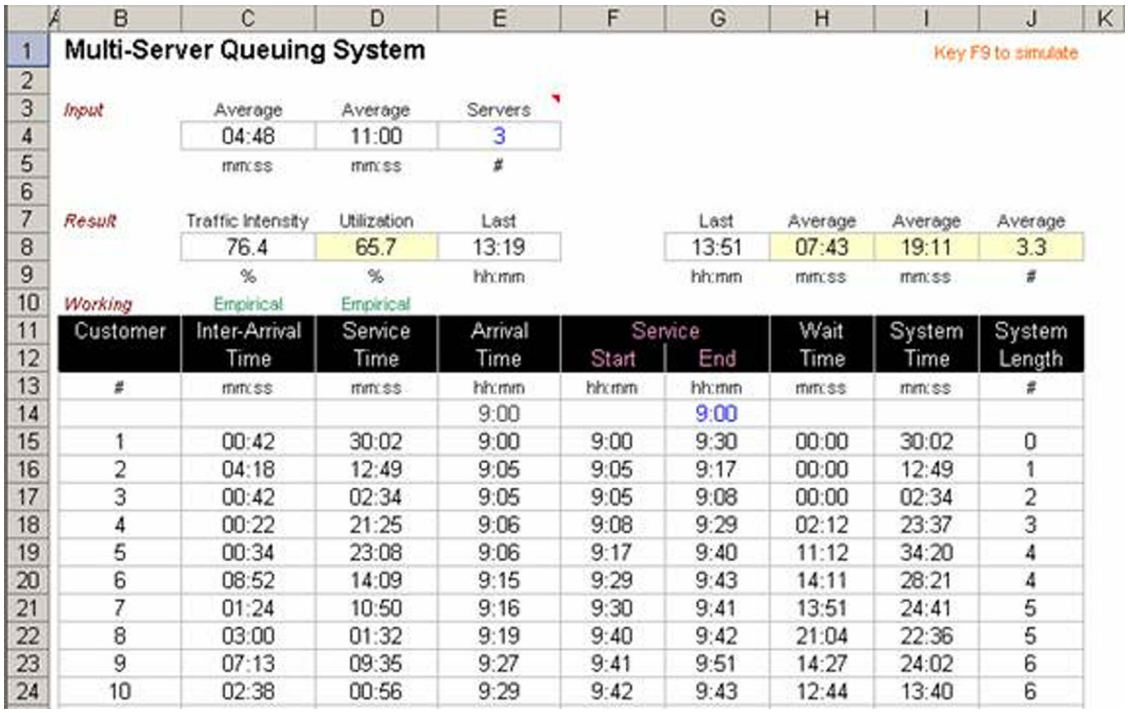
With the knowledge acquired from these and other pioneering works, building a single-server spreadsheet queue model under simple assumptions like "service in arrival order" and "no simultaneous events" is relatively easy. This is done (refer to Figure 1 below) by representing in each row of a spreadsheet table all time durations and time events related to a single customer: inter-arrival time, arrival time, service time, service-start time, and service-end time. Suitable random variates may be used to simulate the stochastic nature of queuing systems. Berger and Whitt (1992), making the same simplifying assumptions, presented (for queues with finite waiting room) a set of recursive equations, which are equivalent to those used in this paper. Whitt (2004) conducted simulation experiments exploiting these recursive formulas, using Splus and Fortran, to test his diffusion approximations.

To extend the single-server basic model for multiple servers, additional columns (corresponding to the service-start and service-end time events for each of the many servers) can be added. Such a spreadsheet once completed would only apply to the given fixed number of servers it is constructed for. That is, for each different number of servers, a separate spreadsheet has to be devised (examples in http://www.bus.ualberta.ca/aingolfsson/simulation/). The approach does have advantages in that each server may be modeled to have different service time distributions and one can examine their individual service performance properties. (e.g. duration of busy periods.)

Short of keeping a stock-pile of templates (one for a given fixed number of servers), macros may be employed to automate the effort of adding or removing columns and changing the formulas of affected cells when increasing or decreasing the number of servers. But these would be rather difficult to create and certainly not easily achieved in an interactive class-room, where students build models from scratch. Hora (2003) also highlighted other equally complex alternatives involving programming languages like Visual Basic or C++, Excel Add-ins, and specialized commercial simulation products such as Arena or ServiceModel, all of which requires non-trivial amounts of time to master before any model building can even begin.

To make the work less tedious, Hora (2003) proposed using array formulas instead. In the resulting G/G/c model, it is now possible to have adjustable c number of servers. While Hora's model is easy to use to examine implications of changes in the number

Figure 1 The Multi-Server Spreadsheet Model



	B	C	D	E	F	G	H	I	J
1	**Multi-Server Queuing System**							Key F9 to simulate	
3	*Input*	Average	Average	Servers					
4		04:48	11:00	3					
5		mm:ss	mm:ss	#					
7	*Result*	Traffic Intensity	Utilization	Last		Last	Average	Average	Average
8		76.4	65.7	13:19		13:51	07:43	19:11	3.3
9		%	%	hh:mm		hh:mm	mm:ss	mm:ss	#
10	*Working*	Empirical	Empirical						
11	Customer	Inter-Arrival Time	Service Time	Arrival Time	Service		Wait Time	System Time	System Length
12					Start	End			
13	#	mm:ss	mm:ss	hh:mm	hh:mm	hh:mm	mm:ss	mm:ss	#
14				9:00		9:00			
15	1	00:42	30:02	9:00	9:00	9:30	00:00	30:02	0
16	2	04:18	12:49	9:05	9:05	9:17	00:00	12:49	1
17	3	00:42	02:34	9:05	9:05	9:08	00:00	02:34	2
18	4	00:22	21:25	9:06	9:08	9:29	02:12	23:37	3
19	5	00:34	23:08	9:06	9:17	9:40	11:12	34:20	4
20	6	08:52	14:09	9:15	9:29	9:43	14:11	28:21	4
21	7	01:24	10:50	9:16	9:30	9:41	13:51	24:41	5
22	8	03:00	01:32	9:19	9:40	9:42	21:04	22:36	5
23	9	07:13	09:35	9:27	9:41	9:51	14:27	24:02	6
24	10	02:38	00:56	9:29	9:42	9:43	12:44	13:40	6

of servers, the array formulation proposed is quite difficult for business undergraduates to comprehend and construct (requiring a 3-finger [Ctrl + Shift + Enter] entry step). Therefore, it is not towel suited for end-user modeling-based teaching.

In this paper, I provide alternative non-array formulations to Hora's array formulation. This simplifies the required work drastically, enabling undergraduates to make their own multi-server spreadsheet simulations. The rest of the paper is organized as follows: the next section explains the layout of the base model and the single-server's service-start time formula, §3 recapitulates Hora's formulation, the new formulation is presented next in §4, followed by a discussion of how I conduct the class teaching it, and lastly a section with some final comments.

2. The Base Model

In Figure 1, row 15 holds all the simulated time durations and time events related to customer 1, row 16 for customer 2 and so on. In each row, from row 15 onwards, column B indicates the customer's order of arrival, column C has the inter-arrival time of this customer (defined as time interval between the arrival time of this customer and that of the preceding customer), column D has the time to service this customer, column E has this customer's time of arrival (computed from the preceding customer's arrival time plus this customer's inter-arrival time), column F has the service-start time of this customer, column G has the service-end time of this customer (which is equal to service-start time plus service time), column H has this customer's wait time (computed by subtracting arrival time from service-start time), column I has this customer's system time (which is the time spent from arrival to end of service) and finally column J has the system length (defined as the number of customers in service plus in queue just before this customer arrives).

For cells above row 15, the two input values in cells E4 and G14 are the number of servers and service end-time of customer 0 respectively. The system being studied is assumed to be opened at this time and the arrival time of customer 1 is obtained by adding this to the first inter-arrival time. Cells C4 and D4 hold the averages of two sets of raw data that I will discuss later, one for inter-arrival times and the other for service durations. For this model, I resample data from these raw data sets using the PERCENTILE and RAND functions. Alternatively, one can assume exponential inter-arrival time and service time distributions, and use the appropriate inverse functions.

Cell C8 computes the traffic intensity (defined as the ratio of the average arrival rate to the average service rate), cell D8 computes the utilization for this simulation run (by dividing the sum of performed service times by total available server time), cell E8 gives the value for the last customer's arrival time, and cell G8 gives the value for the last customer's service-end (or departure) time, which is needed for

computing utilization. The remaining cells compute the average statistics for their respective columns.

For the single-server case (which shares the common spreadsheet layout shown in Figure 1), the service-start time of any customer can be computed by the following argument: service cannot start unless the customer has arrived and also cannot start before the server has completed serving the preceding customer. Therefore, for customer 6 (taken as our representative customer hereon) the service-start time (in cell F20) would be computed by taking the larger of customer 6's arrival time and customer 5's service-end time, as follows:

$$=\text{MAX(E20,G19)} \qquad (1)$$

3. Array Formulation

While keeping the rest of the worksheet the same, Hora innovatively applied an array formula, used in place of formula (1), to compute the service-start time when there are c ($\geq$1) number of servers, thereby converting the G/G/1 model into a parametric G/G/c model. This is a major breakthrough, which permits easy comparisons of the various options with different number of servers.

To understand how this is achieved, consider the situation captured in Figure 1. When customer 6 arrives, customer 3 had already departed leaving behind four customers in the system. Of the four, customer 2 (with the smallest service-end time) will be next to leave, and after that customer 4 (with the second smallest service-end) will leave. Since there are 3 servers, the departure of customer 4 leaves behind an available server, who can immediately start service on the waiting customer 6.

Generalizing for a system with c servers and n customers in the system at the time just before a customer arrives, this newly arrived customer's service-start time equals the (n$-$c$+$1)th smallest of the service-end times from among the customers then in the system. If however the number of customers in the system at that time is less than the number of servers, the arriving customer receives immediate service. With this logic, Hora's array formula for cell F20 is as follows:

$$\{=\text{IF(J20>=\$E\$4,PERCENTILE(IF(\$G\$14:G19>\$E20,\$G\$14:G19),(J20-\$E\$4)/(J20-1)),E20)}\} \qquad (2)$$

where {…}, depicting an array formula, is automatically generated by Excel after entering the formula with the distinctive 3-finger (on the Ctrl, Shift, and Enter keys) entry action. Array formulas typically perform multiple calculations on two or more set of values (referred to as array arguments) to return single or multiple (array) results.

The formulation in (2) needed the number of customers in the system, at the time point just before a customer arrives, to be computed. Hora explained that this can be done by counting the number of customers, from among those who had arrived earlier, whose service-end times are larger than the customer's arrival time. Therefore, the number of customers in the system just before the arrival of customer 6 (cell J20) can be computed as follows:

$$=\text{COUNTIF(\$G\$14:G19,">"\&E20)} \qquad (3)$$

The COUNTIF(range, criteria) function counts the number of cells that meet the specified criteria in the given range. In the above formulation, the criteria is "greater than arrival time of the reference customer." (See appendix for an explanation of the COUNTIF function and, &, the concatenation operator.)

4. New Formulations

Hora's approach is difficult because it uses the PERCENTILE function to find the (n $-$ c $+$ 1)th smallest of the service-end times from among the customers who are then in the system. This entails first identifying, from among all the customers who came before, the sub-set of customers who are in the system just before a customer arrives (therefore the need to invoke the array mode) and also converting the required (n$-$c$+$1)th rank position into the equivalent (n $-$ c)/(n $-$ 1) relative position for use as an argument in PERCENTILE. These are rather complex and non-standard actions.

Careful examination will yield the fact that this service-end time is also the cth largest of the service-end times of all the customers who came before. It is therefore not necessary to identify the above-mentioned subset of customers. I therefore propose the following simpler alternative formulation for computing the service-start time of the 6th customer (cell F20):

$$=\text{IF(J20>=\$E\$4,LARGE(\$G\$14:G19,\$E\$4),E20)} \qquad (4)$$

The LARGE(range, k) function finds the kth largest value from among all the values in the given range. In formulation (4), it finds the cth largest service-end time of all the customers who came before the reference customer. (See appendix for an explanation of the LARGE function.)

The logic of formulation (4) can be summarized as follows:

- If the number of customers in the system just before the reference customer arrives is less than the number of servers (i.e. J20<E4), this customer is immediately served upon arrival (setting F20 = E20).

- If not, service for this customer starts when, from among all customers who came before, the one with the cth largest service-end time departs the system.

Now there is an even more direct approach, one closer to the single-server formulation (1). This alternative formulation is as follows:

$$=\text{MAX(E20, LARGE(\$G\$14:G19,\$E\$4))} \qquad (5)$$

The explanation for formulation (5) is as in formulation (1), except now LARGE finds the service-end time of the next available server. This formula however would not work (for first few customers) when the customer's arrival order is less than c, since it is not possible to calculate cth largest when there are less than c terms to select from. To correct for this, the needed IF function is added to (5), yielding the final formulation for cell F20 as:

$$=\text{MAX(E20, IF(B20>\$E\$4,LARGE(\$G\$14:G19,\$E\$4),0))} \qquad (6)$$

Formulation (6) states that if the arrival order of the customer is less than the number of servers, the service-end time of the next available server is 0 and therefore the arrival time of the customer prevails.

Comparing the formulations, (6) is superior to (4) because it does not require the number of customers in the system as an input (although for independent interest that value may be computed). Formulation (4) in turn is superior to Hora's formulation (2), because the role of the PERCENTILE function, used in the more complex array mode and contorted way, is filled more directly and simply by the LARGE function. Formulas for all the key cells of the completed model are summarized in Figure 2.

5. Teaching Note

The class on simulation modeling of queues is part of an undergraduate course (entitled "Computer as an Analysis Tool"), which explores business modeling using spreadsheets. Though I have conducted the course in a class room fully equipped with desktop computers, my current classes are all in regular 45-seat seminar rooms. In this course, students bring their own laptop computers to every lesson. On our campus, which has excellent wireless network coverage, students are all expected to have their own personal laptops.

The class is conducted, exploring the concept of waiting lines, using an Excel workbook (SimplerMultiServerQueues.xls http://archive.ite.journal.informs.org/Vol7No2/Leong/SimplerMultiServerQueues.xls) as the starting point. The workbook contains the following worksheets: Home, Data, Scratch, Proto, 1-Server, and c-Server. Worksheet "Home" contains introductory information and a legend, which serves as a common "look-and-feel" guide for all worksheets provided and built in the course. Upon release to students, the last two

Figure 2 **Formulas of Key Cells in the Spreadsheet Model**

Documentation

Average Inter-Arrival Time	C4	=AVERAGE(Data!B13:K27)
Average Service Time	D4	=AVERAGE(Data!B30:K44)
Number of Servers	E4	<Input>, validated as whole numbers > 0
Opening Time	G14	<Input>, also Customer 0 "Service-End" Time
Traffic Intensity	C8	=(D4/E4)/C4*100
Utilization	D8	=SUM(D15:D999)/(E4*(G8-G14))*100
Last Customer Arrival Time	E8	=MAX(E14:E999)
Last Customer Service-End Time	G8	=MAX(G14:G999)
Average Time Customer waited	H8	=AVERAGE(H15:H999)
Ave Time Customer in System	I8	=AVERAGE(I15:I999)
Average # Customer in System	J8	=SUMPRODUCT(((J15:J998+1)+J16:J999)/2,C16:C999)/(E8-E14)
Inter-Arrival Time	C20	=PERCENTILE(Data!$B$13:$K$27, RAND())
Service Time	D20	=PERCENTILE(Data!$B$30:$K$44,RAND())
Customer Arrival Time	E20	=E19+C20
Service-Start Time	F20	=MAX(E20, IF(B20>=$E$4,LARGE($G$14:G14,$E$4),0))
Service-End Time	G20	=F20+D20
Customer Wait Time	H20	=F20-E20
Customer in System Time	I20	=G20-E20
# of Customer in System	J20	=COUNTIF($G$14:G19,">"&E20)

worksheets are usually hidden, using Excel's worksheet hide feature (by activating Format/Sheet/Hide in the menu).

Students are to use worksheet "Scratch" (which is just a blank worksheet) to start placing basic information and laying out the table headers. The desired result of this is as shown in worksheet "Proto." Setting aside their various attempts, the class then continues on using the "Proto" worksheet provided. I work interactively with students to develop the formulas to fill the cells for customer 1, taking care to alter the cell referencing (to absolute, relative or mixed as needed) so that the customer 1's row can later be copied and pasted down to complete the simulation model.

Some explanation on the generation of random variates is needed before the inter-arrival time and service time cells can be filled. The top portion of worksheet "Data" provides the necessary notes and formulas. Two simple methods are presented there: re-sampling and inverse function. These would have been taught in an earlier class that covers random variates and Monte-Carlo simulation. I usually use worksheet "Data" to help review the concepts and reinforce learning.

The first model to complete is the single-server model. After evaluating the effect of different traffic intensities on the queue performance, I proceed to show how to change the formula in the service-start cell to make the single-server model into a parametric multi-server model. I usually take a row of students in class to role play the queue and explain why the logic of the service-start time formula is correct. The students, which should number more than servers, are assumed to be arranged in order of their arrival. Students who are supposedly being served would, one-at-a-time but not necessarily in arrival order, be asked to leave the system, to emulate the end of their service. Taking the role of an incoming new arrival, I would then talk through with the class to establish the time point at which service should start for me.

Once completed, students are required to tidy up their work to our prescribed common layout and format (as shown in Figure 1) and complete the documentation of the model (as shown in Figure 2) to improve readability and reduce risk of spreadsheet errors. The result is a high quality professional workable "Decision-Support System." Students can finally unhide the last two worksheets and evaluate their solutions against them.

6. Final Comments

Though the original intent of this paper was to provide simpler alternative formulations to Hora's G/G/c formulation, the model constructed is not restricted to modeling systems with independent identical distributions for the arrival and service processes. For example, the model could be used for trace-driven simulations, where historical inter-arrival and service times recorded from real systems are used. In Berger and Whitt (1992)'s notation, the model presented here can be characterized as the more general A/A/c/8 queue: arbitrary arrival process, arbitrary service process, c servers, and infinite waiting buffer.

Multiple replications of the simulation model can be done using either a spreadsheet simulation add-in (e.g. XLSim http://www.analycorp.com/, Crystal Ball http://www.decisioneering.com/, and @Risk http://www.palisade.com/) or Excel's DataTable feature. I prefer to do the latter as it does not incur additional cost for students. Queue performance results (like average wait-time, probability distribution of system length etc.) can be collected for the numerous simulation runs, under different traffic intensities and number of servers. In interpreting the results of multiple replications one needs to keep in mind that each replication terminates when some prespecified number of customers has completed service, as opposed to when a certain amount of time has passed.

With the simpler proposed formulation, my colleagues and I have for several academic terms conducted classes on spreadsheet queuing simulation (in an interactive manner, along with modeling of other business processes) for second-year business, accountancy, economics, information systems, and social science undergraduates. Using only native Excel features, these undergraduates can build reasonably sophisticated process-driven spreadsheet queuing models, learn to interpret implications of simulation results, apply them to concrete real-world situations and have fun doing it.

Acknowledgments

I have benefited much and the paper is vastly improved by the many constructive comments and suggestions given by the associate editor and anonymous referees. Their effort in providing "developmental review" of this paper is much appreciated.

Appendix

LARGE

LARGE(range, k) returns the kth largest value in the data set specified by the range. For example, LARGE(A1:A20, 1) returns the largest value in the data set contained in cells A1 through A20 (same result as MAX(A1:A20)), LARGE(A1:A20, 2) returns the second largest value, LARGE(A1:A20, 3) returns the third largest value, and so on. When used against a column with running serial numbers 1, 2, 3,.., it can be used to sort a set of numbers in descending order.

CONCATENATION

To concatenate is to combine end-to-end two strings of text. The operator provided in Excel for this purpose is &. Therefore, "ABC" & "DEF" will yield "ABCDEF." This can be done to text or cells with values. For example, ">" & E20 yields "> 5" when cell E20 contains the value 5. Excel automatically converts the number 5 into text before completing the concatenation.

COUNTIF

COUNTIF(range, criteria) counts the number of values that satisfies the criteria in the data set specified by the range. The criteria, which is *not* a logical expression, must be expressed in quotes to denote text, for example ">5." So COUNTIF(A1:A20, ">5") counts the number of values that are greater than 5 in cells A1 through A20. Now using the & concatenation operator and setting the value in cell E20 to 5, COUNTIF(A1:A20, ">"&E20) becomes equivalent to COUNTIF(A1:A20, ">5"). The advantage of the former is that the value of E20 can be arbitrarily changed to other values to suit our purpose.

References

Berger, A., W. Whitt. 1992. Comparisons of multi-server queues with finite waiting rooms. *Stochastic Models* **8** 719–732.

Grossman, T. A., Jr. 1999. Spreadsheet modeling and simulation improves understanding of queues. *Interfaces* **29**(3) 88–103.

Hora, S. C. 2003. Spreadsheet modeling of the G/G/c queuing system without macros or add-ins. *INFORMS Trans. Ed.* **3**(3), http://ite.pubs.informs.org/Vol3No3/Hora/.

Grossman, T. A., A. Ingolfsson. 2002. Graphical spreadsheet simulation of queues. *INFORMS Trans. Ed.* **2**(2), http://ite.pubs.informs.org/Vol2No2/IngolfssonGrossman/.

Winston. 2004. *Microsoft Excel: Data Analysis and Business Modeling*. Microsoft Press.

Whitt, W. 2004. A diffusion approximation for the G/GI/n/m queue. *Oper. Res.* **52** 922–941.