A tool for solving *Ph/M/c* and *Ph/M/c/N* queues

Thomas Begin
LIP (UMR ENS Lyon-INRIA-CNRS-UCBL)
Université Claude Bernard Lyon 1
France

Alexandre Brandwajn
Baskin School of Engineering
University of California Santa Cruz
USA

***Abstract*—We have developed a free online tool for solving the steady-state behavior of a *Ph/M/c* queue. The tool has a simple graphical interface, is freely accessible on the Internet and is compatible with most current browsers (queueing-systems.ens-lyon.fr). Based on a recently published simple recurrence method, it provides customary performance measures (i.e., mean number in system, mean waiting time, system utilization, probability of loss, probability of waiting), as well as the complete steady-state distribution for the number of customers in the system. As of the end of 2011, our tool for solving the steady-state behavior of a *Ph/M/c* queue had close to 15 new visits per day. We plan to add new queues to this tool in the near future (e.g., *M/G/1* queue).**

I. Introduction

In spite of an abundant literature devoted to the solution of queues with multiple servers, researchers, practitioners and Ph.D. students can still find it challenging to obtain a fast and accurate solution of a model seemingly as ordinary as a *G/M/c* queue. Indeed, there is no closed-form expression for performance indices of a *G/M/c* queue. Thus, to obtain the desired results, one has to select one of the existing methods (e.g., matrix-geometric techniques, spectral expansion method) and implement it. This requires both theoretical skills to clearly understand the method's scientific basis and programming skills to implement it. There are some implementations available on the Internet but they are often written in a specific programming language, such as FORTRAN or COBOL, and are limited to specific cases that may not scale easily. We think that many scientists and perfromance practitioners do not have the time to go through these steps, and therefore either resort to simple but potentially inaccurate approximate solutions of the multi-server queue, or limit themselves to a simpler, but potentially inadequate, model such as the *M/M/c* queue.

Hence, we decided to provide a web-based tool with a graphic interface for our recently developed method for solving the steady-state behavior of a $Ph/M/c$ queue [1]. The tool was specifically designed with simplicty of use in mind, and can be run by any performance analyst with access to the Internet.

The remainder of this paper is organized as follows. Section II discusses the capabilities of our tool, and Section III provides guidelines regarding its use.

II. Tool overview

Our tool provides the exact solution to a *Ph/M/c* queue. A *Ph/M/c* queue is a queue with c homogeneous servers, each having exponentially distributed service times. The arrival process, which determines the time between customers arrivals, can follow any acyclic Phase-type distribution, or without loss of generality, any Coxian distribution. Note that Coxian distributions (or Phase-type distributions) can approximate arbitrarily closely any theoretical or empirical distribution [2].

To keep the tool as portable as possible, we implemented our tool as a web application. It includes (i) some PHP code for retrieving the submitted parameter values and for presenting the results, (ii) JavaScript functions (compatible with most popular browsers) to enhance the user interface and (iii) the use of the graphical library Protovis [3] to build a dynamic chart. The actual solution of the *Ph/M/c* queue itself is performed by a standard ANSI C program.

We now detail some of the features of our tool.

A. With or without a restricted buffer

Our tool can solve $Ph/M/c$ queues with unrestricted or restricted buffer (queueing room). In the case of a finite buffer, customers arriving when the buffer is full are assumed to be lost. Conversely, with an unrestricted queueing room, no incoming customers are lost, but the *Ph/M/c* queue possesses a steady-state solution only if the mean arrivals rate is less than c times the mean service rate (stability condition).

B. Two ways of defining the arrival process

The arrival process, which determines the rate and the pattern with which new requests (workload) arrive into the queue, can be defined in two ways. (i) The time between arrivals can be specified by its first two moments, or more specifically, by its mean and coefficient of variation, as is commonly done in queueing theory. The tool then finds a Coxian distribution matching the entered values. This first option allows a fast description of the workload but one should keep in mind that higher-order moments may have a significant influence on many steady-state performance metrics of the queue, and there is an infinite number of Coxian distributions withe the same first two moments. (ii) Another option is to explicitly specify the Coxian distribution. This latter solution allows more flexibility and accuracy for the choice of the inter-arrivals distribution. In the current version of our tool, we limit the number of stages (phases) of the Coxian distribution to 10.

C. Complete steady-state distribution

The solution delivered by our tool includes customary performance parameters (i.e., mean number in system, mean waiting time, system utilization, probability of loss, probability of waiting). Additionally, it provides the complete steady-state distribution for the number of customers in the system at arbitrary instants, which is displayed as a dynamic chart.

III. Guidelines

The tool consists of two web pages. One is the query page, the other being the results page.

A. Performance query

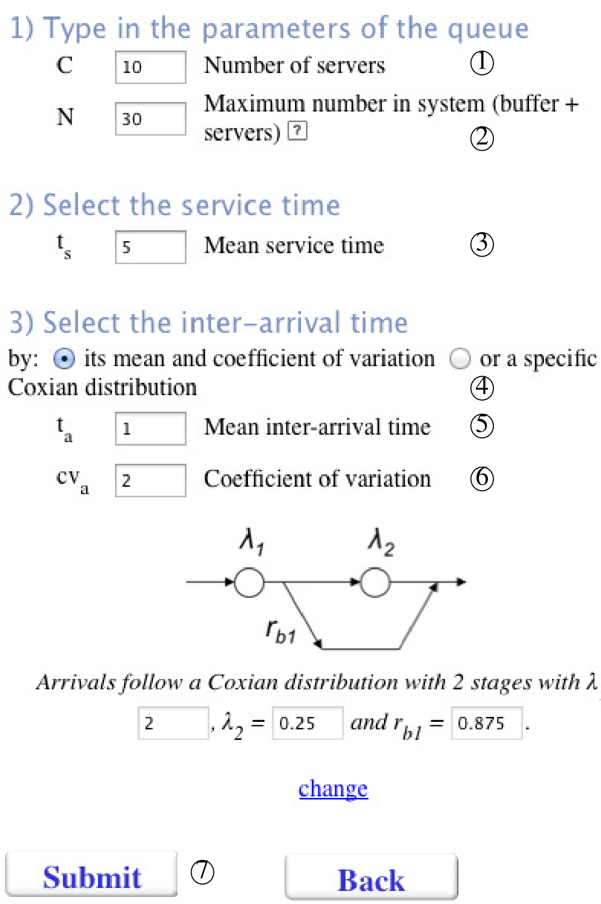


Fig. 1. Specify the parameters values of the *Ph/M/c* queue

The query page, as shown on Fig. 1, contains several fields and two radio buttons. Their function is as follows.

- ① indicates the number of servers,
- ② indicates the system capacity, namely the maximum number of customers allowed in the system including those in service. If this number is omitted, the capacity is assumed to be unlimited.
- ③ indicates the mean service time to process a customer at each server,
- ④ selects the way of defining the arrival process. If the first option is chosen, two additional fields (⑤ and ⑥) and a diagram depicting the corresponding distribution appear as a pop up. Otherwise, additional fields appear to allow the specification of the desired Coxian distribution,
- ⑤ indicates the mean time between arrivals,

Mean number in system $\bar{Q} = 5.2784$

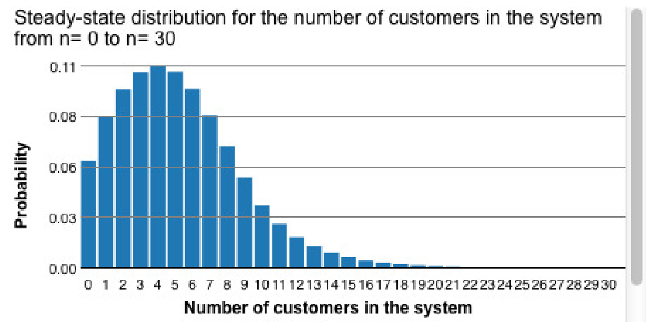
Mean waiting time $\bar{W} = 0.2787$

Mean throughput $\bar{X} = 0.99994$

System utilization (in %) $U = 50.00$

Probability of loss $P_{loss} = 0.00006$

Probability of waiting $P_{wait} = 0.16604$



Steady-state distribution for the number of customers in the system
from n= 0 to n= 30

n	p(n)
0	0.05993
1	0.08437
2	0.10003
3	0.10964
4	0.11311
5	0.10994

Fig. 2. Performance results for the *Ph/M/c* queue

- ⑥ indicates the coefficient of variation of the time between arrivals,
- ⑦ press the "Submit" button to access to the performance results page.

B. Performance results

Fig. 2 illustrates the performance results page. This page shows customary performance metrics, and it also includes the steady-state distributions for the number of customers in the system. The latter is displayed as a graph showing its overall shape and presented as a table giving the precise numeric values of state probabilities.

Acknowledgment

The authors thank Dominique Ponsard and Lucas Delobelle for their help.

References

[1] A. Brandwajn and T. Begin, "A recurrent solution of $Ph/M/c/N$-like and $Ph/M/c$-like queues," *Journal of Applied Probability*, vol. 49, no. 1, March 2012.

[2] A. O. Allen, *Probability, Statistics, and Queueing Theory with Computer Science Applications*. Orlando, FL, USA: Academic Press, Inc., 1978.

[3] M. Bostock and J. Heer, "Protovis: A graphical toolkit for visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, November 2009.