

QSOLVE : AN OBJECT-ORIENTED SOFTWARE TOOL FOR TEACHING QUEUEING THEORY

Fernando C. Castaño Mariño e Paulo R. L. Gondim
System Engineering Department - Instituto Militar de Engenharia - IME
(Rio de Janeiro - RJ)

Abstract - *Queueing systems are present in an enormous quantity of areas and situations of the real life, such as communication, computing, banking and transportation systems. Traditionally, queueing theory teaching is done by the development of formulations used for resolution of different kinds of queueing systems. Through these formulas we can have important information for the analysis of these queues and evaluate the influence of several parameters of interest such as the average number of users and average waiting time in the system. Depending on the adopted model, the related formulations can be somewhat complicated, and the student can't have adequate qualitative and quantitative notions about the influence of each parameter. In this work we describe the development and the use of a software tool (QSOLVE) that facilitates the learning of queueing theory. This software implements formulas from traditional queueing theory, speeds up the final results and makes easy the analysis of the influence of each parameter in the final result, thus giving better notion of the process behavior. There is also a module that allows the user to see some graphics for variables behavior analysis. During the construction of QSOLVE we've used an object-oriented language, allowing the use of principles such as inheritance, polymorphism and modularity. The system has been successfully used for the study and dimensioning of computer communications networks and transportation systems.*

1. Introduction

Queueing theory teaching has been done, frequently, by the use of traditional mathematical developments. With that theory we can obtain formulas for the performance analysis of different kinds of queues. The development of these formulas is very important for apprenticeship, once the student can access the formulations used for queue resolution, develop logical argumentation over different theoretical aspects involved and then generate new results. Applying these formulas one can obtain a lot of information about the queueing system performance and thus evaluate the application of that kind of queue under real conditions. Whereby these formulas are generally complicated, it is difficult to obtain the results manually.

The development of a software tool (QSOLVE) that implement these formulas and speeds up the final results makes easy the analysis of the

influence of each parameter in the final result, giving better notion of the process behavior.

This work is organized as follows: in section 2 we present an overview of the queueing resolution system and describe the object-oriented approach used for its construction; in section 3 and 4, the modules for effective measure calculation and graphical exhibition are described; in section 5, the Erlang B and Erlang C formulas are used for the dimensioning of delay-based and loss-based systems, respectively; the section 6 outlines our main conclusions.

2. Queueing Resolution System

The queueing resolution system is composed by three modules (fig.1):

a) Module for effective measure calculation

The user can have some effective measure giving as input some necessary parameters depending on the kind of queue.

b) Module for graphical exhibition

Graphics are exhibited showing the behavior of some variables.

c) Module for system dimensioning based on Erlang B and Erlang C formulas.

Dimensioning of parameters given the blocking probabilities for M/M/c/c and M/M/c queues.

This system was implemented using DELPHI programming language, version 2.0, considering that such language presents components which facilitates the user interface implementation and allows object creation.

Several classes of objects were created, such that T_MM1, T_MMm, T_Mminfinito, T_MG1, T_GG1, T_MD1, T_MG1lcf, T_MM1K and T_MMCK implementing the M/M/1, M/M/m, M/M/ , M/G/1, G/G/1, M/D/1, M/G/1 LCFS, M/M/1/K, M/M/c/K queues respectively.

These classes were developed so that a instantiated object once initialized with input parameters, compute all effective measuring for that class. In this sense, all formulas are implemented inside the class and new formulas can be added.

The priority queues were implemented by specific modules.

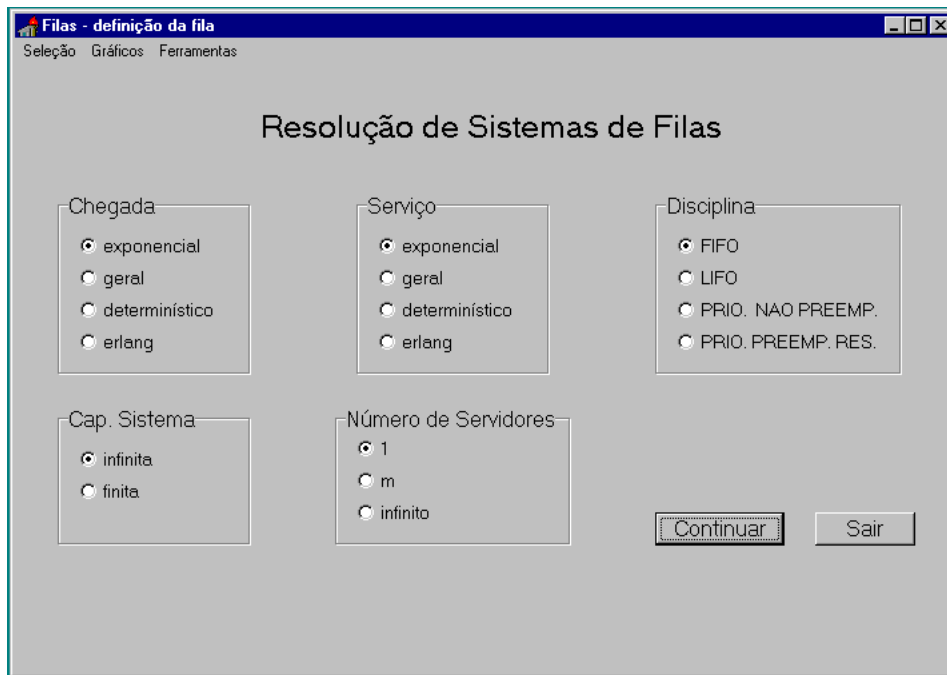


Figure 1. Tool presentation

3 - Module for effective measure calculation

The formulas for effective measure calculation were based on traditional queue literature [1], [2], [3],

[4], and for each case some parameters are necessary (fig 2).

In Table 1, we can see these parameters for each kind of queue, using Kendal's notation:

Table 1 . Parameters for each kind of queue

Queue	Parameters	Effective Measure
<u>M/D/1</u>	- Arrival Rate - Service Rate	- Average Number in the System - Average Time in the System - Average Number in the Queue - Average Time in the Queue
<u>M/M/1</u>	- Arrival Rate - Service Rate	- Average Number in the System - Average Time in the System - Average Number in the Queue - Average Time in the Queue
<u>M/M/m</u>	- Arrival Rate - Service Rate - Number of Servers	- Average Number in the System - Average Time in the System - Average Number in the Queue - Average Time in the Queue - Erlang C
<u>M/M/</u>	- Arrival Rate - Service Rate	- Average Number in the System - Average Time in the System

		- Average Number in the Queue - Average Time in the Queue
<u>M/M/1/k</u>	- Arrival Rate - Service Rate - Users in the System	- Average Number in the System - Average Time in the System - Average Number in the Queue - Average Time in the Queue
<u>M/M/c/K</u>	- Arrival Rate - Service Rate - Users in the System - Number of Servers	- Average Number in the System - Average Time in the System - Average Number in the Queue - Average Time in the Queue - Erlang B (for $c = K$)
<u>M/G/1</u>	- Arrival Rate - Average Service Time - Second Moment of Service Time	- Average Number in the System - Average Time in the System - Average Number in the Queue - Average Time in the Queue
<u>M/G/1 LIFO</u>	- Arrival Rate - Average Service Time - Second Moment of Service Time	- Average Number in the System - Average Time in the System - Average Number in the Queue - Average Time in the Queue
<u>M/G/1 Non preemptive priorities</u>	- Arrival Rate (for each priority) - Service Rate (for each priority)	- Average Number in the System (for each priority) - Average Time in the System (for each priority) - Average Number in the Queue (for each priority) - Average Time in the Queue (for each priority)
<u>M/G/1 Preemptive resume priorities</u>	- Arrival Rate (for each priority) - Service Rate (for each priority)	- Average Number in the System (for each priority)
<u>G/G/1</u>	- Average Arrival Time - Second Moment of Arrival Time - Average Service Time - Second Moment of Service Time	- Upper Limit For Average Time in the Queue

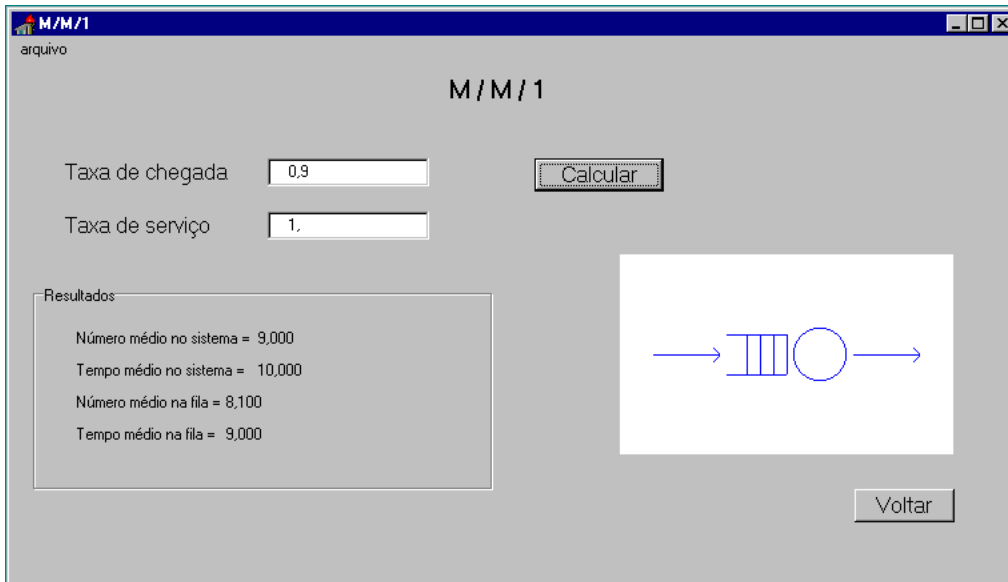


Figure 2. Parameters definition for M/M/1 queue

4 - Module For Graphical Exhibition

This module allows the user to see some graphics for analysis of variables. These graphics are generated by the defined objects for each queue. For each graphic the user can indicate the X and Y axis limits for a detailed study in a specific region of the graphic. This module allows the superposition of graphics in the same figure for behavior comparison of different queues or the same queue with different parameters (fig 3).

Table 2 shows the implemented graphics.

Table 2 . Implemented graphics for each kind of queue

Kind of queue	Graphic
M/D/1	Number in the System x Utilization Factor Time in the System x Utilization Factor Number in the Queue x Utilization Factor Time in the Queue x Utilization Factor
M/M/1	Number in the System x Utilization Factor Time in the System x Utilization

	Factor Number in the Queue x Utilization Factor Time in the Queue x Utilization Factor
M/M/m	Number in the System x Utilization Factor Time in the System x Utilization Factor Number in the Queue x Utilization Factor Time in the Queue x Utilization Factor
M/M/1/K	Number in the System x Utilization Factor Time in the System x Utilization Factor Number in the Queue x Utilization Factor Time in the Queue x Utilization Factor
M/M/c/K	Number in the System x Utilization Factor Time in the System x Utilization Factor Number in the Queue x Utilization Factor Time in the Queue x Utilization Factor

5 - Module for system dimensioning based on Erlang B and Erlang C formulas

This module is composed by tree sub-modules:

- 1) Sub-module for dimensioning based on Erlang B (M/M/c/c queue) formula
- 2) Sub-module for dimensioning based on Erlang C (M/M/c queue) formula
- 3) Sub-module for dimensioning based on a pre stored table with values for Erlang B (M/M/c/c queue) formula

5.1) Sub-module for dimensioning based on Erlang B (M/M/c/c queue) formulas

The Erlang B formula allows to compute the blocking probability based on the number of servers, service rate and arrival rate.

A problem occurs when someone wants to compute the number of servers, service rate or arrival rate based on a given blocking probability. For example, a common practical problem is: how many servers do we need to obtain a blocking probability under a specific value given the arrival rate and the service rate supported by each server? This problem occurs because it is difficult to invert the Erlang B formula to compute directly the others parameters based on blocking probability.

A solution to this problem is to compute the blocking probability for one server, if it not comply, compute the blocking probability for two servers, and successively until the blocking probability comply with the specified.

Given that the Erlang B formula is complex, without a computer this process is very slow and

subject to errors. Technicians usually have a table for pre defined values, and then they try to fit the real problem in the cases of the table. With a computer, the user can find accurate values and the process becomes very efficient.

5.1.1) Dimensioning the number of servers:

The program considers initially one server, computes the blocking probability given the arrival rate and the service rate (or utilization factor). If the obtained blocking probability is greater than the requested, the program makes the same operation for two servers, and successively, until the computed blocking probability comply with the requested.

Then, for dimensioning the number of servers, the user informs the arrival rate, the service rate (or utilization factor) and the upper limit of the blocking probability (fig 4).

5.1.2) Dimensioning the arrival rate

This problem is analogous to 5.1.1. In this case, the user informs the number of servers, service rate and blocking probability. The user also needs to inform to the system the arrival rate which will be considered by the system to initiate to processing and to incrementing the arrival rate to next attempt.

For dimensioning the arrival rate, the user informs the server number, the service rate, initial arrival rate (upper limit), the step and the upper limit of the blocking probability.

5.1.3) Dimensioning the service rate

This process is analogous to the above described in 5.1.1.

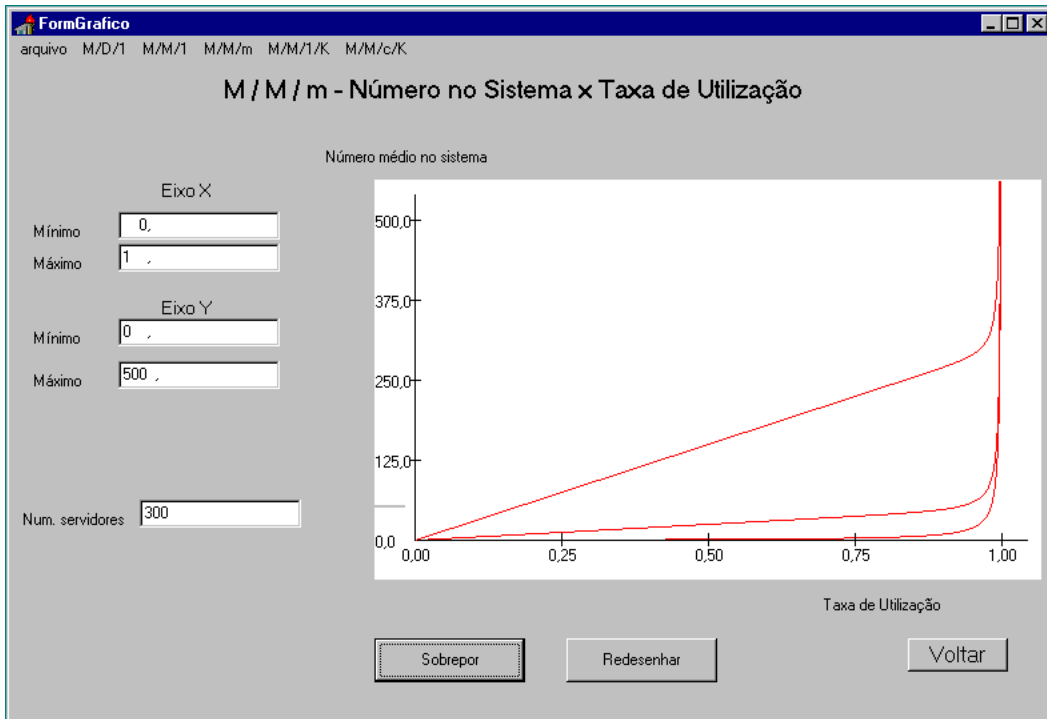


Figure 3. Example of a graphic generated by QSOLVE

The screenshot shows a software window titled "Erlang B" with a menu bar containing "arquivo". The main title is "Erlang B - Fila M / M / m / m". Below the title, there is a section for dimensioning options:

Selecione o que será dimensionado:

- Número de servidores (dado tx cheg. e tx serv.)
- Número de servidores (dado erlangs)
- Taxa de chegada
- Taxa de serviço
- Erlangs

Below this, there are input fields for:

- Taxa de chegada: 0.992
- Taxa de serviço: 1.
- Prob. bloqueio menor que: 0.016

A "Calcular" button is located to the right of these fields. Below the input fields, there is a "Resultados" section with a text area containing the text: "número de servidores maior ou igual a 4". A "Voltar" button is located at the bottom right of the window.

Figure 4. Example of the dimensioning the number of servers

For dimensioning the service rate, the user informs the server number, the arrival rate, initial service rate (lower limit), the step and the upper limit of the blocking probability.

5.1.4) Dimensioning the utilization factor

This process is analogous to the above described in 5.1.1.

For dimensioning the utilization factor, the user informs the server number, the initial utilization factor (upper limit), the step and the upper limit of the blocking probability.

5.2) Sub-module for dimensioning based on Erlang C (M/M/c queue) formulas

This process is analogous to presented in 5.1.

5.2.1) Dimensioning the number of servers

For the number of servers dimensioning, the user will inform: arrival rate, service rate and upper limit of the blocking probability.

5.2.2) Dimensioning the arrival rate

For the arrival rate dimensioning, the user will inform: server number, initial arrival rate (upper limit), step, service rate and upper limit of the blocking probability.

5.2.3) Dimensioning the service rate

For the service rate dimensioning, the user will inform the server number, the arrival rate, initial service rate (lower limit), the step and the upper limit of the blocking probability.

5.3) Sub-module for dimensioning based on a pre stored table with values for Erlang B (M/M/c/c queue)

This module shows a pre stored table [5], with the blocking probabilities for some cases (fig 5).

Erlang B - Tabela

arquivo

Erlang B - Consulta à Tabela

Carga de Tráfego (Erlangs)

Probabilidade de Bloqueio

N	1.0%	1.2%	1.5%	2.0%	3.0%	5.0%	7.0%	10%	15%	20%	30%	40%	50%
1	0,0101	0,0121	0,0152	0,0204	0,0309	0,0526	0,0753	0,111	0,176	0,250	0,429	0,667	1,00
2	0,153	0,168	0,190	0,223	0,282	0,381	0,470	0,595	0,796	1,00	1,45	2,00	2,73
3	0,455	0,489	0,535	0,602	0,715	0,899	1,06	1,27	1,60	1,93	2,63	3,48	4,59
4	0,869	0,922	0,992	1,09	1,26	1,52	1,75	2,05	2,50	2,95	3,89	5,02	6,50
5	1,36	1,43	1,52	1,66	1,88	2,22	2,50	2,88	3,45	4,01	5,19	6,60	8,44
6	1,91	2,00	2,11	2,28	2,54	2,96	3,30	3,76	4,44	5,11	6,51	8,19	10,4
7	2,50	2,60	2,74	2,94	3,25	3,74	4,14	4,67	5,46	6,23	7,86	9,80	12,4
8	3,13	3,25	3,40	3,63	3,99	4,54	5,00	5,60	6,50	7,37	9,21	11,4	14,3
9	3,78	3,92	4,09	4,34	4,75	5,37	5,88	6,55	7,55	8,52	10,6	13,0	16,3
10	4,46	4,61	4,81	5,08	5,53	6,22	6,78	7,51	8,62	9,68	12,0	14,7	18,3
11	5,16	5,32	5,54	5,84	6,33	7,08	7,69	8,49	9,69	10,9	13,3	16,3	20,3
12	5,88	6,05	6,29	6,61	7,14	7,95	8,61	9,47	10,8	12,0	14,7	18,0	22,2
13	6,61	6,80	7,05	7,40	7,97	8,83	9,54	10,5	11,9	13,2	16,1	19,6	24,2
14	7,35	7,56	7,82	8,20	8,80	9,73	10,5	11,5	13,0	14,4	17,5	21,2	26,2
15	8,11	8,33	8,61	9,01	9,65	10,6	11,4	12,5	14,1	15,6	18,9	22,9	28,2
16	8,88	9,11	9,41	9,83	10,5	11,5	12,4	13,5	15,2	16,8	20,3	24,5	30,2
17	9,65	9,89	10,2	10,7	11,4	12,5	13,4	14,5	16,3	18,0	21,7	26,2	32,2
18	10,4	10,7	11,0	11,5	12,2	13,4	14,3	15,5	17,4	19,2	23,1	27,8	34,2
19	11,2	11,5	11,8	12,3	13,1	14,3	15,3	16,6	18,5	20,4	24,5	29,5	36,2
20	12,0	12,3	12,7	13,2	14,0	15,2	16,3	17,6	19,6	21,6	25,9	31,2	38,2
21	12,8	13,1	13,5	14,0	14,9	16,2	17,3	18,7	20,8	22,8	27,3	32,8	40,2
22	13,7	14,0	14,3	14,9	15,8	17,1	18,2	19,7	21,9	24,1	28,7	34,5	42,1
23	14,5	14,8	15,2	15,8	16,7	18,1	19,2	20,7	23,0	25,3	30,1	36,1	44,1

Volta

Figure 5. Erlang B pre stored table

6. Conclusion

In this work, we presented an auxiliary system to queueing theory teaching, giving a better notion of the influence of each parameter in the final result for each kind of queue. The system gives the results quickly, allowing the user to test values and to analyse the parameter influence.

With QSOLVE, the user can see some graphics and compare the behavior of different kinds of queues. The user can also dimension the various parameters of M/M/c/K and M/M/c queues based on Erlang B and Erlang C formulas.

Future work includes new graphics, additional characteristics parameters for each kind of queue, inclusion of new queues and the development of simulation modules that will help in the study of more complex and analytically intractable systems.

References

- 1) Kleinrock, L. 1975. Queueing Systems, Vol. I. New York: Willey.
- 2) Kleinrock, L. 1975. Queueing Systems, Vol. II. New York: Willey.
- 3) Gross, D., and Harris, C. M. 1985. Fundamentals of Queueing Theory (2nd ed.). New York: Willey

- 4) Bertsekas, D. P. , and Gallager, R. 1992. Data Networks (2nd ed), Prentice Hall.
- 5) Yacoub, M. D. , 1993. Foundations of Mobile Radio Engineering, CRC Press.