A Computational Approach for Evaluating Steady-State Probabilities of a Multiprocessor Queueing System with a Waiting Time Restriction

Vladimir Sahakyan
Institute for Informatics and Automation Problems of NAS RA
Yerevan, Armenia
email: vladimir.sahakyan@sci.am

Artur Vardanyan
Institute for Informatics and Automation Problems of NAS RA
Yerevan, Armenia
email: artur.vardanyan@iiap.sci.am

***Abstract*—Distributed and parallel high-speed computing systems have become increasingly important in recent years for handling large amounts of data and performing complex scientific research using multi-agent intelligent methods. Accurate modeling tasks in science and technology require fast and large-scale computations. However, organizing computations in a cluster environment to optimally utilize system resources is a challenge that requires effective scheduling of task execution, which takes into account the required resources and execution time.**

In this context, this paper proposes a computational approach for evaluating steady-state probabilities in the environment of a multiprocessor queueing system. The proposed approach contributes to a practical partial solution to the challenges of organizing computations in a cluster environment and can be applied of the organizing process in modeling tasks that require fast and large-scale computations for accurate results to the optimal utilization of resources.

Overall, the paper contributes to the growing field of distributed and parallel computing by providing a practical approach to evaluating steady-state probabilities in a multiprocessor queueing system.

***Keywords*— Queueing Theory, Multiprocessor System, Multiprocessor Queueing System, Waiting Time Restriction.**

I. Introduction

For a more detailed formulation, consider a computing system consisting of m $(m \geq 1)$ computing processors (or cores). It is assumed that the number of tasks that can be queued is limited by a certain number of waiting slots: n $(n \geq 1)$[1]. The reason for denying the service can be the impossibility of placing a task in the queue and the impossibility of serving with user-defined constraints(time, number of processors, etc.). Each task in the system is characterized by four random parameters $(\nu, \beta, \omega, \gamma)$, where ν is the number of computing resources (processors, cores, cluster nodes, etc.,) needed by the task for servicing, β is the maximum time required to service a task, ω is the possible time a task can wait before being serviced, after which it leaves the system without being serviced, and γ is the time interval after which the task is allowed to start servicing the system from the moment the task enters the system. When $\gamma = 0$, the value of this parameter can be omitted, and in systems without a waiting time restriction, the value of the parameter ω is also omitted[2]. For tasks with a waiting time restriction, when they enter the system, the possibility of execution is checked. Then the task is either accepted after being placed in the service queue or it is denied service. The time required for the service of the task is partly conditional, that is, it is the maximum allowable value. In reality, it is random and may be less than the given one. Therefore, the order of services can be changed at the moment of receipt of tasks and completion of services.

Tasks receive a service denial if, at the time of entering the system, it turns out that it cannot be serviced according to the specified parameters (for example, start service at the specified time).

The system is observed when a task is placed in a queue or a service is completed. By considering all possible scenarios of the system operation, when the system goes into a state where i $(1 \leq i \leq m)$ tasks are being serviced, and j $(1 \leq j \leq n)$ tasks are waiting in the queue, a system of equations was obtained(due to finite numbers n and m, the number of possible states of the system is finite). This paper aims to analyze the solution to the aforementioned system of equations and evaluate steady-state probabilities.

II. System parameters

The system parameters are described as follows:
α - a random value of the time interval between neighboring entrances, which has the probability distribution:

$$P(\alpha < t) = A(t),$$

β - a random value of the task execution time, which has the probability distribution:

$$P(\beta < t) = B(t),$$

ω - a random value of the permissible waiting time for a task in the queue, which has the probability distribution:

$$P(\omega < t) = W(t),$$

https://doi.org/10.51408/csit2023_22

ν - a random value of the number of required computational resources for performing a task, which has the probability distribution:

$$P(\nu \le k) = V(t), k = 1, 2, ..., m.$$

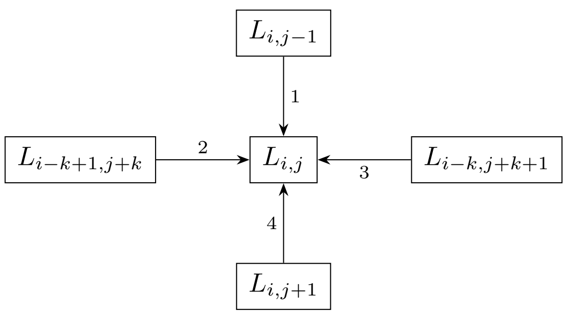
Tasks will be serviced in the order they enter the system, i.e., FIFO discipline is used[3]. Those tasks that the system receives when the queue is fully occupied (that is, there are already n tasks in the queue waiting to be serviced), receive system access and service denial.

III. Basic Notations

To describe the transition of a system from one possible state to another, the following notation is introduced:
$L_{i,j}$ - the state of the system when i tasks are serviced and j tasks are waiting in the queue,
$P_{i,j}$ - the probability that the system is in the $L_{i,j}$ state:

$$P_{i,j} = P(L_{i,j}).$$

Below is a schematic representation of the cases when the system can pass from another state to state $L_{i,j}$:



The transitions shown in the diagram are described as follows:

1) The initial state of the system was $L_{i,j-1}$, and a new task arrived and joined the queue,
2) The previous state of the system was $L_{i-k+1,j+k}$, where $k = 1, 2, ..., \min(i, n-j)$. A task completed its service and left the system, and the first k tasks from the queue were accepted for service.
3) The previous state of the system was $L_{i-k,j+k+1}$, where $k = 0, 1, ..., \min(i-1, n-j-1)$. The first task in the queue left the queue (its waiting time ran out), and the first k tasks from the queue were accepted for service.
4) The previous state of the system was $L_{i,j+1}$, and a task from the queue (not the first task) left the queue (its waiting time ran out).

Due to the finite number of possible states of the system, the system goes into a stable mode of operation, i.e., steady state[4].

IV. Limitations and equations

In order to derive the probabilistic equations for the aforementioned system, the following distribution functions: $A(t)$, $B(t)$, $W(t)$, and $V(k)$ are incorporated:

$$A(t) = 1 - e^{-at},$$

$$B(t) = 1 - e^{-bt},$$

$$W(t) = 1 - e^{-wt},$$

$$V(k) = \frac{k}{m},\ k = 1, 2, ..., m :$$

where a is the intensity of the incoming stream, b is the intensity of service and w is the intensity of the failure of service for a task from the queue.

As the number of cases described in the previously presented scheme may vary depending on the values of the variables i, j, and k, it is important to initially focus on the cases determined by the borderline values of these variables. To capture these scenarios, we introduce the notations $\delta_i^{(1)}$, $\delta_{i,j,k}^{(2)}$, and $\delta_{i,j,k}^{(3)}$. Subsequently, these probabilities will be described after constructing the probabilistic equations.

Obviously, when $i = 0$ and $j = 0$, then

$$P_{0,0} = \frac{b}{a} P_{1,0}, \tag{1}$$

and when $i = 0$ and $1 \le j \le n$, then

$$P_{0,j} = 0. \tag{2}$$

In the case when $1 \le i \le m$ and $j = 0$, then

$$\begin{aligned} P_{i,0} = \frac{1}{a+ib} \Bigg[& a\delta_{i-1}^{(1)} P_{i-1,0} + \\ & + b \sum_{k=k_0}^{k_1} \left((i-k+1)\delta_{i,0,k}^{(2)} P_{i-k+1,k} \right) + \\ & + w \sum_{k=0}^{k2} \left(\delta_{i,0,k}^{(3)} P_{i-k,k+1} \right) \Bigg], \end{aligned} \tag{3}$$

where

$$k_0 = \begin{cases} 0, & \text{if } 1 \le i < m \\ 1, & \text{if } i = m \end{cases},$$

$$k_1 = min(i, n),$$

$$k_2 = min(i-1, n-1).$$

Note that in this case, the equations determining $P_{i,0}$ include only the case when the queue leaves the first task in the queue.

The scenario where $1 \le i \le m$ and $1 \le j < n$ is now examined.

$$\begin{aligned} P_{i,j} = \frac{1}{a+ib+jw} \Bigg[& a(1-\delta_{i-1}^{(1)}) P_{i,j-1} + \\ & + b \sum_{k=k_0}^{k_1} \left((i-k+1)\delta_{i,j,k}^{(2)} P_{i-k+1,k+j} \right) + \\ & + w \sum_{k=0}^{k_2} \left(\delta_{i,j,k}^{(3)} P_{i-k,k+j+1} \right) + wjP_{i,j+1} \Bigg], \end{aligned} \tag{4}$$

where k_0 is determined as in the previous case, and

$$k_1 = min(i, n-j),$$

$$k_2 = min(i-1, n-j-1).$$

Note that in this scenario, the penultimate term of the equations determining $P_{i,j}$ accounts for the situation where the first task lefts the queue, while the final term considers the scenario where a non-first task left the queue. It remains to consider the last two borderline cases: the first, when $1 \leq i < m$ and $j = n$, then

$$P_{i,n} = \frac{1}{a + ib + nw}\Big(a(1 - \delta_{i-1}^{(1)})P_{i,n-1} + \\ + b(i+1)\delta_{i,n,0}^{(2)}P_{i+1,n}\Big), \tag{5}$$

and the last, when $i = m$ and $j = n$, then

$$P_{m,n} = \frac{a(1 - \delta_{m-1}^{(1)})P_{m,n-1}}{a + mb + nw} : \tag{6}$$

Before to describe $\delta_i^{(1)}$, $\delta_{i,j,k}^{(2)}$ and $\delta_{i,j,k}^{(3)}$ probabilities, some events are defined:

Definition 1. Let A_i be the event that

$$\sum_{l=1}^{i} \nu_l \leq m.$$

Definition 2. Let B_i be the event that

$$\sum_{l=1}^{i} \nu_l < m.$$

Definition 3. Let $C_{i,j,k}$ be the event that if $j = 0$

$$\sum_{l=1}^{i-k} \nu_l + \sum_{l=i-k+2}^{i+1} \nu_l \leq m,$$

and if $1 \leq j \leq n$

$$\sum_{l=1}^{i-k} \nu_l + \sum_{l=i-k+2}^{i+1} \nu_l \leq m < \sum_{l=1}^{i-k} \nu_l + \sum_{l=i-k+2}^{i+2} \nu_l.$$

Definition 4. Let $D_{i,k}$ be the event that

$$\sum_{l=1}^{i-k+1} \nu_l \leq m < \sum_{l=1}^{i-k+2} \nu_l.$$

Definition 5. Let $E_{i,k}$ be the event that

$$\sum_{l=1}^{i-k} \nu_l \leq m < \sum_{l=1}^{i-k+1} \nu_l.$$

After establishing the definitions, let us proceed to the description of $\delta_i^{(1)}$, $\delta_{i,j,k}^{(2)}$, and $\delta_{i,j,k}^{(3)}$.
The probability $\delta_i^{(1)}$ will be determined as follows:

$$\delta_i^{(1)} = \begin{cases} 1, & \text{if } i = 0 \\ P\Big(A_{i+1} \Big/ B_i\Big), & \text{if } 1 \leq i < m \\ 0, & \text{if } i = m \end{cases}, \tag{7}$$

here $P\Big(A_{i+1} \Big/ B_i\Big)$ represents the conditional probability of event A given that the event B_i has occurred.
The probability $\delta_{i,j,k}^{(2)}$ will be determined as follows:

$$\delta_{i,j,k}^{(2)} = \begin{cases} 1, & \text{if } k = 0 \\ P\Big(C_{i,j,k} \Big/ D_{i,k}\Big), & \text{if } k \leq i-1 \\ 0, & \text{if } k = i \end{cases}, \tag{8}$$

where $1 \leq i < m$ and $1 \leq j \leq n$, the conditional probability $P\Big(C_{i,j,k} \Big/ D_{i,k}\Big)$ signifies the probability of the event $C_{i,j,k}$ given that the event $D_{i,k}$ has occurred.

And the probability $\delta_{i,j,k}^{(3)}$ will be determined as follows:

$$\delta_{i,j,k}^{32)} = \begin{cases} 1, & \text{if } k = 0 \\ P\Big(C_{i,j,k} \Big/ E_{i,k}\Big), & \text{if } k \leq i-1 \\ 0, & \text{if } k = i \end{cases}, \tag{9}$$

here $1 \leq i < m$, $1 \leq j \leq n$ and $P\Big(C_{i,j,k} \Big/ E_{i,k}\Big)$ represents the conditional probability of event $C_{i,j,k}$ given that the event $E_{i,k}$ has occurred.
Consequently, by solving the system of equations defined by (1) to (6), the state probabilities of each $L_{i,j}$ ($0 \leq i \leq m$ and $0 \leq j \leq n$) will be determined. That system of equations is linear and solving it allows us to obtain the desired $P_{i,j}$ ($0 \leq i \leq m$ and $0 \leq j \leq n$) probabilities.

Theorem. The system of equations (1) to (6) with variables $P_{i,j}$ ($0 \leq i \leq m$, $0 \leq j \leq n$) has a solution for the given parameters. Furthermore, that solution satisfies the following condition:

$$\sum_{i=0}^{m} \sum_{j=0}^{n} P_{i,j} = 1 : \tag{10}$$

Obviously, the (10) condition follows from the fact that the sum of the probabilities of all possible outcomes of the same event is one.
The theorem provides a formal guarantee that the system of equations has a solution that accurately determines the steady-state probabilities for each state of the multiprocessor queueing system. This result is significant in understanding and analyzing the behavior of the system under various conditions and parameters.

V. Supporting probability formulas

This section delves into the derivation of supporting probabilities and formulas that play a crucial role in calculating the values of the probabilities determined from the formulas presented in equations (7), (8) and (9). This section will present the derivations step-by-step, highlighting the key mathematical principles and reasoning behind each result, shedding light on their significance in the context of the computation process of the steady-state probabilities in our multiprocessor queueing system.

In the previous publications[1], [3], three lemmas were derived and established that are instrumental to this current analysis. These lemmas provide key insights and results that will be utilized in this section to advance the derivation of supporting probabilities. The three lemmas are as follows:

Lemma 1. The probability that i tasks occupy k processors can be calculated as follows:

$$P\left(\sum_{j=1}^{i}\nu_j = k\right) = \frac{1}{m^i}\binom{k-1}{i-1},$$

where $1 \le i \le k \le m$.

Lemma 2. The probability that i tasks occupy no more than k processors can be calculated as follows:

$$P\left(\sum_{j=1}^{i}\nu_j \le k\right) = \frac{1}{m^i}\binom{k}{i},$$

where $1 \le i \le k \le m$.

Lemma 3.

$$P\left(\sum_{i=1}^{k}\nu_i \le s < \sum_{i=1}^{k+1}\nu_i\right) = \frac{1}{m^{k+1}}\left(m - \frac{s-k}{k+1}\right)\binom{s}{k},$$

where $1 \le k \le s \le m$.

Let us now delve into the derivation of formulas for calculating the values of the probabilities determined from the formulas presented in equations (7), (8), and (9).

To begin, let us focus on the probabilities determined by the formula (7). This formula captures the probabilities related to a specific event or condition in the system.

Lemma 4. The probability value $\delta_i^{(1)}$ is determined as follows:

$$\delta_i^{(1)} = \begin{cases} 1, & \text{if } i = 0 \\ \dfrac{1}{i+1}, & \text{if } 1 \le i < m \\ 0, & \text{if } i = m \end{cases}. \tag{11}$$

Lemma 5. The probability value $\delta_{i,j,k}^{(2)}$ is determined as follows:

$$\delta_{i,j,k}^{(2)} = \begin{cases} 1, & \text{if } k = 0 \\ p_1, & \text{if } k \le i-1 \text{ and } j = 0 \\ p_2, & \text{if } k \le i-1 \text{ and } 1 \le j \le n \\ 0, & \text{if } k = i \end{cases}, \tag{12}$$

where $1 \le i < m$, and the quantities p_1 and p_2 are determined by the formulas.

Lemma 6. The probability value $\delta_{i,j,k}^{(3)}$ is determined as follows:

$$\delta_{i,j,k}^{(3)} = \begin{cases} 1, & \text{if } k = 0 \\ q_1, & \text{if } k \le i-1 \text{ and } j = 0 \\ q_2, & \text{if } k \le i-1 \text{ and } 1 \le j \le n \\ 0, & \text{if } k = i \end{cases}, \tag{13}$$

where $1 \le i < m$, and the quantities q_1 and q_2 are determined by the formulas .

VI. The steady-state probabilities

This section will outline the process of solving the system of equations derived earlier. First, we recall the system of equations represented by equations (1) to (6). These equations describe the probabilities of each state in the system, denoted as $P_{i,j}$, where $0 \le i \le m$ and $0 \le j \le n$. Our goal is to find a solution that satisfies these equations for the given system parameters.

The first step in solving the system of equations is to analyze its structure and properties. It is observed that the system of equations is linear, as the probabilities are expressed as linear combinations of other probabilities. But it is important to note that the system of equations represented by equations (1) to (6) is homogeneous. By recognizing the homogeneity of the system of equations, the solution can be approached by changing one of the equations of the system with the condition (10), as a result of which a non-homogeneous system of linear equations will be obtained. This linearity allows us to employ various mathematical techniques for solving linear systems.

Next, a common approach is employed by utilizing matrix representation and matrix operations to solve the system of equations. To facilitate this process, the A matrix is constructed, which represents the coefficients, and the b vector, which represents the values of linear combinations of $P_{i,j}$ probabilities in the equations.

The A matrix is a square matrix of size $(m+1)(n+1)\times(m+1)(n+1)$, where each element corresponds to the coefficient of a particular $P_{i,j}$ term in the equations. The $\mathbf{b}$ vector is a column vector of size $(m+1)(n+1)\times 1$, with each element representing the value on the right-hand side of the equations. By setting up the matrix equation

$$A\mathbf{x} = \mathbf{b},$$

where $\mathbf{x}$ is a column vector representing the unknown probabilities $P_{i,j}$, various matrix operations can be applied to solve $\mathbf{x}$.

A numerical algorithm has been developed to solve the system of equations, which has been implemented in the Python programming language using various tools from the NumPy library[6]. The algorithm provides an efficient and accurate method for computing the steady-state probabilities of the multiprocessor queueing system.

The detailed description and implementation of the algorithm go beyond the scope of this paper and will be presented in a separate article, which is intended for publication. The separate article will provide a comprehensive explanation of the algorithm's steps, underlying mathematical techniques, and code implementation.

By presenting the algorithm separately, we aim to provide a more thorough and focused discussion on the technical aspects of the solution methodology. It will also allow for a detailed analysis of the algorithm's performance, computational complexity, and potential optimizations.

Further details regarding the algorithm's development, implementation, and performance evaluation will be provided in the upcoming publication.

VII. Conclusion

In this paper, a computational approach is proposed for evaluating steady-state probabilities in a multiprocessor queueing system with a waiting time restriction. The proposed approach addresses the challenges of organizing computations in a cluster environment and contributes to the optimal utilization of system resources. By formulating the system as a set of equations based on the number of tasks being serviced and waiting in the queue, we could derive a finite set of equations representing the system's states. We, then, solved these equations to obtain the steady-state probabilities for each state of the system.

The practical implications of our approach are significant, as it enables efficient scheduling of task execution in distributed and parallel computing systems. The approach can be applied to various modeling tasks, enabling accurate and efficient utilization of resources.

Future research can focus on further optimization of the computational approach, considering additional factors such as dynamic task arrivals, varying task requirements, and resource allocation strategies. Additionally, applying the approach to real-world case studies and comparing it with the existing methods would provide valuable insights into its effectiveness and performance. Overall, our work lays the foundation for advancing the understanding and application of queueing theory in multiprocessor systems with waiting time restrictions.

References

[1] Vladimir Sahakyan, Artur Vardanyan, "The Queue Distribution in Multiprocessor Systems with the Waiting Time Restriction", *Mathematical Problems of Computer Science*, Yerevan, vol. 51, pp. 82–89, 2019. DOI: https://doi.org/10.51408/1963-0035

[2] Vladimir Sahakyan, Artur Vardanyan, "About the possibility of executing tasks with a waiting time restriction in a multiprocessor system", *AIP Conference Proceedings*, vol. 2757, pp. 030003, 2023. DOI: https://doi.org/10.1063/5.0135784

[3] Vladimir Sahakyan, Artur Vardanyan, "The Queue State for Multiprocessor System with Waiting Time Restriction", *Computer Science and Information Technologies 2019, Conference Proceeding*, Yerevan, pp. 116–119, 2019. DOI: https://doi.org/10.1109/CSITechnol.2019.8895093

[4] Bocharov P.P., D'Apice C., Pechinkin A.V., Salerno S., "Queueing Theory", *VSP*, Utrecht, Boston, pp. 94-98, 2004. ISBN: 90-6764-398-X

[5] Vladimir Sahakyan, Artur Vardanyan, "The Steady State Distribution for M/M/m/n Model with the Waiting Time Restriction", *Mathematical Problems of Computer Science*, Yerevan, vol. 54, pp. 34–40, 2021. DOI: https://doi.org/10.51408/1963-0057

[6] NumPy Contributors, *NumPy Documentation*, Available online at: https://numpy.org/doc/stable/, Accessed on: July 26, 2023.