Steady-State Probabilities and Task Rejection/Failure Probability Estimation in the Multiprocessor Queueing System

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

***Abstract*—This research analyzes a multiprocessor queueing system $M|M|m|n$, focusing on estimating steady-state probabilities and utilizing them to calculate the probabilities of task access rejection and task failure due to waiting time restrictions. The study presents the development of numerical algorithms using the NumPy library in Python to gain insights into system performance and efficiency.**

Numerical algorithms were developed to evaluate the steady-state probabilities by efficiently solving the system of equations using matrix operations and tools from the NumPy library. With these probabilities in mind, calculations were performed for specific system parameters to determine the probability of task access rejection and task failure due to waiting time restrictions. The research highlights the practical applicability of these findings to real-world scenarios and discusses the potential use of high-performance computing systems for optimization.

Overall, this research contributes to the understanding and enhancement of multiprocessor system efficiency. The developed numerical algorithms and implementation provide a foundation for further advancements in the field. By addressing challenges related to system performance and resource utilization, this study offers practical solutions for improving overall productivity in multiprocessor systems across various domains and applications.
***Keywords*— Multiprocessor System, Task Scheduling, Multiprocessor Queueing System, Waiting Time restriction.**

I. INTRODUCTION

In the previous study, research was conducted on the $M|M|m|n$ multiprocessor queueing system, which employed a first-in, first-out (FIFO) discipline for task servicing. The study focused on deriving a system of equations that describes the system's transition to a steady state. For a more comprehensive derivation, interested readers are encouraged to refer to the previous article [1].

The investigation involved a computing system consisting of m $(m \geq 1)$ processors and a queue capable of accommodating a maximum of n $(n \geq 1)$ tasks [2], [3]. By considering various operational scenarios, 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 computational approach was proposed to evaluate the $P_{i,j}$ steady-state probabilities in a multiprocessor queueing system. Furthermore, leveraging the previously derived steady-state probabilities, the calculations were made to assess two crucial factors in the queueing system. The first factor is the probability of task access rejection, while the second factor is the probability of task failure due to its waiting time restriction.

The primary objective of this article is to delve into the detailed design of computational algorithms, their implementation, and the evaluation of their performance based on previous research.

II. THE STEADY-STATE PROBABILITIES

The previous work [1] outlined the process of solving the system of equations comprising $(n+1) \times (m+1)$ equations and an equal number of unknown variables, representing the steady-state probabilities of the system. However, the homogeneity of the system posed a challenge in obtaining a unique solution for the system of equations.

It is worth noting that the variables represent probabilities of all relevant occurrences of the same event, and their summation should equal one. To address this condition, we modified one equation of the system to incorporate this requirement, resulting in a non-homogeneous system of linear equations. Leveraging the linearity of the system, various mathematical techniques are employed for solving linear systems, utilizing matrix representation and matrix operations as a common approach. To facilitate this process, the A matrix is constructed to represent the coefficients and the b vector to represent the values of linear combinations of $P_{i,j}$ probabilities in the equations.

The A matrix is a square matrix of size $(m+1)(n+1)\times(m+1)(n+1)$, with each element corresponding to the coefficient of the unknown $P_{i,j}$ terms in the equations. On the other hand, the $\mathbf{b}$ vector is a column vector of size $(m+1)(n+1) \times 1$, where each element represents the value on the right-hand side of the equations. Set up the matrix equation

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

where $\mathbf{x}$ is a column vector representing the unknown probabilities $P_{i,j}$.

To solve the system of equations, we developed a numerical algorithm implemented in Python, utilizing the powerful tools available in the NumPy library[4].

https://doi.org/10.51408/csit2023_24

The NumPy library provides efficient low-level implementations of standard linear algebra algorithms, relying on BLAS and LAPACK. Specifically, we utilized the "solve" function from the "numpy.linalg" library. This function is designed to solve linear matrix equations or systems of linear scalar equations. It computes the "exact" solution, $\mathbf{x}$, for a well-determined linear matrix equation of the form $A\mathbf{x} = \mathbf{b}$, where A is the coefficient matrix and $\mathbf{b}$ is the right-hand side vector. The approach leverages the capabilities of the "solve" function to obtain the solution for the system of equations representing the steady-state probabilities in the $M|M|m|n$ multiprocessor queueing system.

Experimental calculations are performed using predefined system parameters, such as the intensity of the incoming stream, service rate, and service failure rate for tasks in the queue. For the calculations, we assumed a computing system with 256 ($m = 256$) computing processors (or cores) and a maximum queue capacity of 50 ($n = 50$) tasks. For the intensity of the incoming task stream, the intensity of service, and the intensity of service failure for a task from the queue have used these values: $a = 20$, $b = 100$, and $w = 5$.

The table below presents the results of our calculations, showcasing the calculated values for the steady-state probabilities corresponding to each state of the queueing system. These probabilities provide valuable insights into the performance and behavior of the multiprocessor queueing system under the given system parameters.

TABLE I
THE STEADY-STATE PROBABILITIES

Probability	Value
$P_{0,0}$	0.00014
$P_{0,1}$	0
$P_{0,2}$	0
...	...
$P_{0,50}$	0
$P_{1,0}$	0.00069
$P_{1,1}$	0.0115
$P_{1,2}$	0.0125
...	...
$P_{1,50}$	4.93×10^{-9}
$P_{2,0}$	0.00011
$P_{2,1}$	0.00657
$P_{2,2}$	0.00717
...	...
$P_{2,50}$	9.12×10^{-9}
$P_{3,0}$	0.00042
$P_{3,1}$	0.00251
$P_{3,2}$	0.00268
...	...
$P_{255,50}$	3.47×10^{-7}
$P_{256,0}$	8.61×10^{-126}
$P_{256,1}$	5.58×10^{-123}
$P_{256,2}$	3.98×10^{-120}
...	...
$P_{256,48}$	3.19×10^{-11}
$P_{256,49}$	3.37×10^{-9}
$P_{256,50}$	3.49×10^{-7}

Note that the calculation presented here was performed on a PC. However, it is crucial to emphasize that the designed numerical algorithm enables calculations with system parameters derived from real-world problems using high-performance computing systems.

This implementation facilitates accurate and efficient computation of the steady-state probabilities in the $M|M|m|n$ multiprocessor queueing system and allows for comprehensive investigations into the behavior and performance of the system under various scenarios.

III. Task Access Rejection and Failure Probabilities

This section focuses on calculating the probabilities of task access rejection and task failure (when a task leaves the queue due to the expiration of its waiting time) at a specific moment in time.

Upon task arrival, the system will reject access if the number of tasks waiting in the queue at that moment reaches its maximum capacity, which is n. In other words, we can calculate the probability of task access rejection at any point in time by utilizing the probabilities of being in all system states when the queue is fully occupied. This probability of task access rejection, denoted by P_r, can be calculated as follows:

$$P_r = \sum_{i=0}^{m} \frac{aP_{i,n}}{a + ib + nw} \tag{1}$$

where a represents the intensity of the incoming task stream, b represents the intensity of service, w represents the intensity of service failure for a task from the queue, and $P_{i,n}$ with $i = 0, 1, ..., m$ are the system state probabilities when i tasks are being serviced and the queue is fully occupied.

By using the system parameters assumed in the previous section, the task rejection probability is calculated using the formula (1), which yielded the following result:

$$P_r = 5.6 \times 10^{-8}$$

In the study referenced as [1], it is assumed that each task has a maximum allowable waiting time before being serviced, after which it leaves the system without being served. Calculate the probability of task failure at any point in time, denoted by P_f, using the system state probabilities. The formula to compute P_f is as follows:

$$P_f = \sum_{i=0}^{m} \sum_{j=0}^{n} \frac{jwP_{i,j}}{a + ib + jw} \tag{2}$$

Here, as above, a represents the intensity of the incoming task stream, b represents the intensity of service, w represents the intensity of service failure for a task from the queue, and $P_{i,j}$ with $i = 0, 1, ..., m$ are the system state probabilities when i tasks are being serviced and j tasks are waiting in the queue.

Once again, using the system parameters assumed in the previous section, the task rejection probability was calculated by using the formula (2), which resulted in the following outcome:

$$P_f = 0.32$$

These calculated values offer valuable insights into the anticipated performance of the system and facilitate a comprehensive evaluation of its efficiency. This, in turn, leads to improved resource utilization and overall system productivity, providing tangible benefits to various domains and applications reliant on such systems.

IV. Future Work

The findings from this study lay the groundwork for several potential avenues of future research that can further enhance the efficiency and performance of the multiprocessor queueing system with waiting time restrictions. In particular, an estimation of the expected virtual waiting time for the aforementioned queueing system with the waiting time restriction.

The expected virtual waiting time represents a crucial performance metric in queueing systems, as it directly impacts user experience, resource utilization, and overall system efficiency. By leveraging the steady-state probabilities obtained through this research, we can devise methodologies to more accurately estimate the expected virtual waiting time for the analyzed queueing system, as presented in [5]. This effort extends to encompass a more intricate scenario: the queueing system with waiting time restrictions.

The estimation of the expected virtual waiting time opens up an opportunity to develop a pre-scheduler capable of estimating and providing the average waiting time that incoming tasks will experience before accepting service. This pre-scheduler can significantly enhance task management and resource allocation in the queueing system, leading to improved overall system efficiency and user satisfaction.

Further research can delve into the application of advanced analytical techniques, such as stochastic modeling and simulation, to gain deeper insights into the complex behavior of the queueing system with waiting time restrictions. These techniques can provide valuable predictions and enable the exploration of various what-if scenarios, facilitating the identification of optimal system configurations and resource allocation strategies. Performing sensitivity analysis and investigating the effects of different parameter variations on system performance can provide valuable insights into the robustness of the proposed queueing system with waiting time restrictions. Identifying critical parameters that significantly influence performance can guide system administrators in making informed decisions to optimize the system's efficiency under various conditions.

Translating the research findings into practical implementations and conducting real-world validation is essential to assess the effectiveness of the proposed methodologies.

In conclusion, the future work presented in this section offers a roadmap for further advancing the understanding and application of multiprocessor queueing systems with waiting time restrictions. Addressing these research directions has the potential to significantly improve system performance, optimize resource utilization, and enhance user experience across a broad range of domains and applications.

V. Conclusion

This research conducted a thorough investigation of the $M|M|m|n$ multiprocessor queueing system, focusing on the analysis of the $M|M|m|n$ multiprocessor queueing system with a primary focus on estimating steady-state probabilities and utilizing them to calculate the probabilities of task access rejection and task failure due to waiting time restrictions. Through the development of numerical algorithms and their implementation using the NumPy library in Python, valuable insights into system performance and efficiency were gained.

The findings provide a detailed understanding of steady-state probabilities for each system state and estimate some useful probabilities. Optimizing these factors using high-performance computing systems and real-world parameters, to enhance the performance and resource utilization of multiprocessor systems.

This research contributes to the field by offering practical solutions for improving multiprocessor system efficiency. The developed algorithms and insights gained from our calculations lay the groundwork for further advancements. Addressing challenges related to system performance enables increased productivity and improved resource utilization in various industries and applications.

References

[1] Vladimir Sahakyan, Artur Vardanyan, "A Computational Approach for Evaluating Steady-State Probabilities of a Multiprocessor Queueing System with a Waiting Time Restriction", *Computer Science and Information Technologies 2023, Conference Proceeding*, Yerevan, 2023.

[2] 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

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

[5] Vladimir Sahakyan, Artur Vardanyan, "About Virtual Wait Times in a Multiprocessor System", *Computer Science and Information Technologies 2023, Conference Proceeding*, Yerevan, 2023.