DOI: https://doi.org/10.36910/6775-2524-0560-2025-61-06
UDC 519.872

Dymova Hanna, Candidate of Technical Sciences, PhD, Associate Professor
https://orcid.org/0000-0002-5294-1756
Kherson State Agrarian and Economic University, Kherson/Kropyvnytskyi, Ukraine

DEVELOPMENT OF APPROXIMATE CALCULATION METHODS FOR MULTISERVER PRIORITY SYSTEMS

Dymova H. Development of Approximate Calculation Methods for Multiserver Priority Systems. The article addresses the pressing issue of enhancing the efficiency and reliability of multiserver queuing systems (QS) with absolute priorities. Such QS are critically important components in the management of complex real-time technical complexes, where requirements for performance and task resolution speed are extremely high. It is established that traditional exact analytical methods, which are effective for single-server systems, become impractical for the multiserver case due to the exponential growth of the state space, especially when it is necessary to account for queue lengths and the diversity of requests. To overcome these limitations, a combined approach is proposed, based on integrating approximate analytical calculations with the results of large-scale simulation modeling. Simulation experiments performed a threefold function: providing initial data, generating and verifying hypotheses, and conducting the final comprehensive system check. As a result of the research, engineering techniques were developed and substantiated for estimating three key system characteristics. Firstly, the average waiting time was approximated using the two-moment complementary Weibull distribution function. Secondly, a generalized formula for the continuous busy period $\pi(n)$ was derived, incorporating a novel correction factor Δ. This factor accounts simultaneously for the impact of channel utilization ρ, the service coefficient of variation v, and the number of channels n. Thirdly, an approximate formula for calculating the expected number of interruptions $\overline{k_j}$ for a request of a specific priority was developed. The validity of the developed formulas was systematically verified by comparing calculated and simulated data obtained from 200 000 observations. The high consistency of the results confirms that the proposed methodology is a reliable, simple, and practically oriented tool for the effective design and optimization of multiserver priority QS, particularly in scenarios where rigorous mathematical analysis is infeasible.

Keywords: multiserver systems, priority service, absolute priority, simulation modeling, busy period, waiting time, approximate calculation methods.

Димова Г. О. Розробка наближених методик розрахунку багатоканальних пріоритетних систем. У статті розглядається актуальна проблема підвищення ефективності та надійності функціонування багатоканальних систем масового обслуговування (СМО) з абсолютними пріоритетами. Такі СМО є критично важливими компонентами в управлінні складними технічними комплексами реального часу, де вимоги до продуктивності та оперативності вирішення завдань є вкрай високими. Встановлено, що традиційні точні аналітичні методи, які ефективні для одноканальних систем, стають непридатними для багатоканального випадку через експоненційне розростання простору станів, особливо при необхідності врахування черг та різноманітності заявок. Для подолання цих обмежень запропоновано комбінований підхід, що базується на інтеграції наближених аналітичних розрахунків з результатами масштабного імітаційного моделювання. Імітаційні експерименти виконували трояку функцію: надання первинних даних, формування та верифікація гіпотез, а також фінальна комплексна перевірка. В результаті дослідження було розроблено та обґрунтовано інженерні методики для оцінки трьох ключових характеристик системи. По-перше, середня тривалість очікування була апроксимована за двома моментами за допомогою додаткової функції розподілу Вейбулла. По-друге, виведено узагальнену формулу для періоду безперервної зайнятості $\pi(n)$, що включає поправочний коефіцієнт Δ, який вперше враховує одночасний вплив завантаження каналу ρ, коефіцієнта варіації обслуговування v та кількості каналів n. По-третє, розроблена наближена формула для розрахунку очікуваної кількості переривань $\overline{k_j}$ для заявки з певним пріоритетом. Достовірність розроблених формул систематично верифіковано шляхом порівняння розрахункових та імітаційних даних, отриманих за 200 000 спостережень. Висока узгодженість результатів підтверджує, що запропонована методика є надійним, простим і практично орієнтованим інструментом для ефективного проєктування та оптимізації багатоканальних пріоритетних СМО, зокрема там, де точний математичний аналіз є неможливим.

Ключові слова: багатоканальні системи, пріоритетне обслуговування, абсолютний пріоритет, імітаційне моделювання, період безперервної зайнятості, час очікування, наближені методи розрахунку.

Formulation of the problem. Tasks of managing complex technical complexes in real time usually require the use of multi-machine or multiprocessor systems, both for performance reasons and taking into account reliability requirements and maintenance organization. Differences in the importance of tasks, their complexity, and requirements for the speed of solution lead to the necessity of introducing priority service disciplines.

Research analysis. Methods for analyzing single-server systems with priorities are developed quite well [1, 2, 3], although their numerical implementation poses a number of difficult problems. However, the complexity of these tasks increases sharply when moving to multiserver systems. Such tasks are usually solved only in the simplest (exponential) version with means identical for all types of requests – a case that is atypical and of little interest [4, 5]. All attempts to create really applicable techniques that take into account the number of requests of each type residing in channels and in queues [6] are knowingly doomed to failure due to the excessive growth of the state space. Simulation systems do not always help either:

GPSS World does not allow modeling multiserver devices with priority interruptions, moreover, even single-server ones with multiple interruptions.

In such cases, combinations of analytical approaches with simulation models can be effective, where the latter are used in three ways:

– to obtain primary input data;

– to generate and verify hypotheses (assumptions) embedded in analytical fragments of the algorithm;

– for final comprehensive verification.

Highlighting previously unresolved parts of the problem. The task of analyzing a multiserver priority system boils down to calculating the average unavailability time of the system for servicing a "tagged" request of type j. In the case of priority with interruptions, the average duration of an interruption multiplied by the number of interruptions is added to it. Solutions to these problems for a single-server system are elementary, but for multiserver systems, they are absent. To qualitatively understand the arising effects, it was necessary to resort to simulation modeling.

Purpose of the study. The purpose of this work is the development and substantiation of approximate engineering techniques for calculating key characteristics (waiting time, busy period duration, number of interruptions) of multiserver systems with absolute priorities by combining analytical approaches and simulation modeling.

Presentation of the main material. When modeling a system with absolute priority, the duration of request service was determined at the moment it was placed in the queue: upon entering the system using a properly configured random number generator, and after an interruption – with the remainder of the previously generated duration. Interrupted requests were placed at the head of the corresponding queue.

One of the new problems turned out to be determining the necessity of an interruption and choosing the channel to be interrupted. It proved expedient to have a list of occupied channels ordered by descending priorities of the requests being served (with equal priorities – by increasing arrival times in the system); in this case, it was sufficient to compare the priority of the newly arrived request with the one being served in the last channel. Naturally, this required reordering the list upon interruption and upon selecting a request from the queue after service completion.

The study of the final results of simulation modeling of multiserver systems could lead, at best, to empirical approximations with a scope limited to the investigated range of parameters. Therefore, the simulation model was supplemented with the collection of statistics on the sought-after "internal" indicators.

For calculating the average unavailability time, an analytical approach was found. The distribution of the waiting time for the start of service of a tagged request in a single-server system was approximated by two moments. For approximation, the complementary Weibull distribution function was used:

$$\overline{F}(t) = exp\left(\frac{-t^{\kappa}}{T}\right), \tag{1}$$

with which the theoretical moments are connected:

$$f_m = T^{\frac{m}{k}}\Gamma\left(1+\frac{m}{k}\right), m = 1,2 \dots , \tag{2}$$

where T is the simulation time, $\Gamma(x)$ is the gamma function, m is the average number of requests in the system, and k is the arrival of the k-th request.

In the n-server case, the differential distribution function of the waiting time is the n-th power of the above and reduces to the same distribution with the parameter T divided by n. Accordingly, the average waiting time ω is calculated according to:

$$\omega(n) = \left(\frac{\tau}{n}\right)^{\frac{1}{k}}\Gamma\left(1+\frac{1}{k}\right) \tag{3}$$

so that

$$\frac{\omega(n)}{\omega(1)} = \left(\frac{1}{n}\right)^{1/k}$$

Modeling confirmed the good accuracy of this method. Now let us consider methods for solving problems with request types ordered by descending priorities.

Modeling of continuous busy periods (CBP) of a multiserver system with homogeneous requests while maintaining a constant specific load per channel showed (Table 1) that with Markovian (M), i.e., exponential service distribution, the average duration of CBP is:

$$\pi(n) = \frac{b_1}{n(1 - \lambda b_1/n)} \tag{4}$$

where b_1 is the average processing duration of the main request of the busy period, n is the lowest priority of the current service, λ is the flow density.

In other cases, this dependence could be considered only as a rough approximation. In this regard, it was decided to seek a general formula in the form:

$$\pi(n) = b_1/\big(1 + \Delta(\rho, v, n)\big)n(1 - \rho) \tag{5}$$

where ρ is the channel utilization factor and v is the coefficient of variation of service.

The necessary correction Δ was calculated through the average value of the continuous busy period $\pi(n)$ observed in the experiment using the formula:

$$\Delta = (n - 1)\pi(n)b_1 - 1 \tag{6}$$

The aforementioned zero value of the correction for the exponential service distribution determined its multiplicative structure and the necessity of turning into zero at $v = 1$. In addition, in a wide range of coefficients of variation (from zero to two), the dependence on v under other equal conditions turned out to be close to linear. Furthermore, the modulus of the correction was approximately proportional to the utilization factor ρ. Finally, the correction increased non-linearly with the number of channels n, showing a tendency towards saturation, and by definition equaled zero at $n = 1$ (the latter requirement is another argument for the multiplicative form of the correction). As a result, the correction Δ should be accepted in the form:

$$\Delta = \rho(v - 1)(n - 1)/(4n) \tag{7}$$

The calculation using formulas (5) and (7) is implemented in the following Python function:

```
def calculate_pnz(b1: float, rho: float, v: float, n: int) -> float:
    """
    Розраховує середню тривалість періоду безперервної зайнятості (ПНЗ)
    для n-канальної системи за формулами (5) та (7) зі статті.

    Аргументи:
    b1 (float): Середня тривалість обробки головної заявки періоду зайнятості.
    rho (float): Коефіцієнт завантаження одного каналу (ρ).
    v (float): Коефіцієнт варіації часу обслуговування (v).
    n (int): Кількість каналів.

    Повертає:
    float: Середня тривалість ПНЗ (π(n)).
    """
    if n < 1:
        raise ValueError("Кількість каналів (n) має бути >= 1")
    if rho >= 1.0:
        print(f"Попередження: Коефіцієнт завантаження rho ({rho}) >= 1. Система
нестабільна.")
        return float('inf')
```

```
if n == 1:
    # Для одноканальної системи поправка Delta = 0 (згідно з (7))
    delta = 0.0
else:
    # Формула (7): Розрахунок поправки Delta
    delta = rho * (v - 1) * (n - 1) / (4 * n)

# Формула (5): Розрахунок ПНЗ
numerator = b1 * (1 + delta)
denominator = n * (1 - rho)

pi_n = numerator / denominator
return pi_n
```

Table 1. Average Durations of Continuous Busy Periods

n	***p***	***S/C***	***D***	E_3	***M***	H_2
1	0,5	*S*	1,997	2,000	2,003	2,008
		C	2,000	2,000	2,000	2,000
	0,7	*S*	3,332	3,333	3,349	3,325
		C	3,333	3,333	3,333	3,333
	0,9	*S*	9,971	10,000	9,975	9,510
		C	10,000	10,000	10,000	10,000
2	0,5	*S*	0,917	0,974	0,998	1,053
		C	0,938	0,974	1,000	1,062
	0,7	*S*	1,496	1,599	1,673	1,781
		C	1,521	1,605	1,667	1,812
	0,9	*S*	4,421	4,732	4,972	5,110
		C	4,438	4,762	5,000	5,562
3	0,5	*S*	0,593	0,643	0,667	0,723
		C	0,611	0,633	0,667	0,722
	0,7	*S*	0,959	1,056	1,114	1,243
		C	0,981	1,044	1,111	1,241
	0,9	*S*	2,809	3,113	3,312	3,609
		C	2,833	3,122	3,333	3,813
4	0,5	*S*	0,441	0,480	0,502	0,553
		C	0,453	0,473	0,500	0,547
	0,7	*S*	0,708	0,774	0,836	0,950
		C	0,724	0,787	0,833	0,943
	0,9	*S*	2,048	2,321	2,517	2,747
		C	2,078	2,275	2,500	2,922
5	0,5	*S*	0,351	0,383	0,401	0,443
		C	0,360	0,377	0,400	0,440
	0,7	*S*	0,560	0,614	0,667	0,773
		C	0,573	0,627	0,667	0,760
	0,9	*S*	1,619	1,810	1,990	2,262
		C	1,640	1,848	2,000	2,360

The column headers D, E_3, M, H_2 in Table 1 denote types of service time distribution. Also, this table presents the results of calculating the average duration of the continuous busy period of the system with homogeneous requests using the simulation model (*S*) and by calculation (*C*). The H_2 distribution replaced the gamma distribution with a coefficient of variation of 2. In all cases, it was assumed that $b_1 = 1$.

Again, let us divide the input flow equally among the service channels. For a single-server system, the expected number of interruptions is $\overline{k_j} = \Lambda_{j-1} b_{j,1}$, where $\Lambda_{j-1} = \sum_{i=1}^{j-1} \lambda_j$ is the intensity of the flow of requests with the right to interrupt the j-th one. In the n-server case, a request arriving during the service of the j-th one may not interrupt it at all (if at least one channel is free or occupied by serving a lower-priority request). On the other hand, a tagged request can be interrupted by those that "initially" fell on other channels. It is clear that the first effect will prevail for requests of relatively high priority, and the second for low priority. Approximately, it can be assumed that the interruption of the j-th request occurs under the following conditions:

- it is present in at least one of the channels (probability equals $1-\left(1-\frac{\rho_j}{R_k}\right)^n$);
- there are no requests of lower priority in any of the other channels (probability $\left(R_j/R_k\right)^{n-1}$).

In these formulas, $\{R_j\}$ denotes the cumulative load factor of the system by requests up to the j-th type inclusive. The value k is the index of the last (lowest) priority of requests. Arriving interrupting requests fall on average on $\rho_i = \lambda_i b_{i,1}$ of those being interrupted. Thus, the average number of interruptions of the j-th request can be estimated by the formula:

$$\overline{k_j} = \frac{\Lambda_{j-1} b_{j,1}\left[1-\left(\frac{\rho_j}{R_k}\right)^n\right]\left(\frac{R_j}{R_k}\right)^{n-1}}{\left(\lambda_j b_{j,1}\right)} = \frac{\Lambda_{j-1}}{\lambda_j}\left[1-\left(\frac{\rho_j}{R_k}\right)^n\right]\left(\frac{R_j}{R_k}\right)^{n-1} \qquad (8)$$

The implementation of this formula, which also accounts for the special case for $n = 1$, is shown below:

```
import math

def calculate_interruptions(j: int, n: int, lambdas: list[float], b_times: list[float])
-> float:
    """
    Розраховує середню кількість переривань заявки j-го типу
    в n-канальній системі.
    Використовує окрему формулу для n=1 та формулу (8) для n > 1.

    Аргументи:
    j (int): Індекс пріоритету (1-індексований, тобто 1, 2, 3...).
    n (int): Кількість каналів.
    lambdas (list[float]): Список *загальних* інтенсивностей потоків [λ1, λ2, ..., λk].
    b_times (list[float]): Список середніх часів обслуговування [b1, b2, ..., bk].

    Повертає:
    float: Середня кількість переривань (k¯_j).
    """
    if j < 1 or j > len(lambdas):
        raise ValueError(f"Індекс 'j' ({j}) виходить за межі діапазону пріоритетів
(1..{len(lambdas)})")
    if n < 1:
        raise ValueError("Кількість каналів (n) має бути >= 1")
    if len(lambdas) != len(b_times):
        raise ValueError("Списки інтенсивностей (lambdas) та часів (b_times) повинні
мати однакову довжину")

    num_priorities = len(lambdas)

    # Λ_(j-1) = Σ(λ_i) for i = 1 to j-1
    # Важливо: j 1-індексований, зрізи Python 0-індексовані
    Lambda_j_minus_1 = sum(lambdas[:j-1])

    if n == 1:
        # Спеціальна формула для одноканальної системи
        return Lambda_j_minus_1 * b_times[j-1]

    # --- Розрахунок за формулою (8) для n > 1 ---

    # Розрахунок коефіцієнтів завантаження для кожного типу
    rhos = [l * b for l, b in zip(lambdas, b_times)]

    # λ_j
    lambda_j = lambdas[j-1]
    # ρ_j (у формулі) = ρ_j (у тексті) = λ_j * b_j
    rho_j = rhos[j-1]

    # R_j = Σ(ρ_i) for i = 1 to j
    R_j = sum(rhos[:j])

    # R_k = Σ(ρ_i) for i = 1 to k (total load)
    R_k = sum(rhos)

    if lambda_j == 0:
        print(f"Попередження: Інтенсивність lambda_{j} = 0. Повертаю 0 переривань.")
        return 0.0
    if R_k == 0:
        print("Попередження: Загальне навантаження R_k = 0. Повертаю 0 переривань.")
        return 0.0

    # Формула (8)
    term1 = Lambda_j_minus_1 / lambda_j

    # [1 - (ρ_j / R_k)^n]
    term2_base = max(0, rho_j / R_k)
    term2 = 1.0 - math.pow(term2_base, n)

    # (R_j / R_k)^(n-1)
    term3_base = max(0, R_j / R_k)
    term3 = math.pow(term3_base, n - 1)

    k_j_bar = term1 * term2 * term3

    return k_j_bar
```

In Table 2, the results of such a calculation (*C*) are compared with those obtained from the simulation model (*S*). The calculation values (*C*) were obtained using the aforementioned `calculate_interruptions` function. A system with three types of requests was considered, with average service durations $b_{1,1} = 0{,}45$, $b_{2,1} = 0{,}90$, $b_{3,1} = 1{,}35$ and flow intensities per channel $\lambda_{j1} = 0{,}2$, $\lambda_2 = 0{,}3$, $\lambda_3 = 0{,}4$ (the load factor of 0,9 was maintained by multiplying these intensities by the number of channels n). Since it turned out that the types of service duration distributions have practically no effect on the interruption multiplicity, the volume of the table has been reduced accordingly.

Table 2. Average Multiplicities of Interruptions

n	Type 2 (*S*)	Type 2 (*C*)	Type 3 (*S*)	Type 3 (*C*)
1	0,180	0,180	0,674	0,675
2	0,116	0,243	0,827	0,800
3	0,071	0,104	0,882	0,980
4	0,043	0,042	0,892	1,088
5	0,026	0,017	0,897	1,153

The table illustrates the qualitative correspondence of the results. For type 2 requests (higher priority), calculation formula (8) gives noticeable deviations from the simulation data at $n = 2$ and $n = 3$, however, the interruption multiplicities themselves are very small in these cases. For low-priority type 3 requests, where interruptions result in a more noticeable delay, the formula gives values close to simulation for $n \leq 2$, but the discrepancy increases with the number of channels. Nevertheless, the resulting error may be insignificant for the overall estimation of the residence time.

The proposed approaches were programmed using the Python language. The results of simulation modeling of multiserver systems with 200,000 observations of the highest priority requests were used as reference.

To verify the models, a single-server problem was solved with the above input data given in the section on interruption multiplicities. A sufficiently large total load coefficient ($R = 0{,}9$) ensured a significant role of waiting durations and interruptions of low-priority requests. The calculation results are summarized in Table 3. The column headers D, E_3, M, H_2, as in Table 1, denote types of service time distribution.

Thus, the assumptions regarding the average duration of the CBP and interruption multiplicity obtained from the analysis of simulation experiments allowed obtaining a simple and sufficiently accurate technique for the approximate calculation of multiserver priority systems.

Table 3. Average Residence Times in a System with Absolute Priority

n	Type	*S/C*	*D*	E_3	*M*	H_2
1	1	*S*	0,472	0,480	0,493	0,559
		C	0,472	0,480	0,495	0,561
	2	*S*	1,233	1,311	1,486	2,192
		C	1,242	1,323	1,478	2,216
	3	*S*	10,038	12,875	18,014	40,557
		C	10,104	12,741	18,364	41,745
2	1	*S*	0,452	0,452	0,454	0,459
		C	0,453	0,454	0,455	0,462
	2	*S*	1,000	1,020	1,060	1,223
		C	1,006	1,023	1,064	1,229
	3	*S*	5,718	6,851	9,536	20,065
		C	5,640	6,840	9,264	20,108
3	1	*S*	0,450	0,450	0,450	0,449
		C	0,451	0,451	0,452	0,453
	2	*S*	0,942	0,948	0,969	1,019
		C	0,947	0,954	0,959	1,040
	3	*S*	4,113	4,928	6,456	13,168
		C	4,161	4,858	6,398	13,314
4	1	*S*	0,450	0,451	0,451	0,446
		C	0,450	0,451	0,452	0,451
	2	*S*	0,919	0,923	0,937	0,951
		C	0,925	0,929	0,926	0,976
	3	*S*	3,350	3,911	5,053	9,597
		C	3,424	3,979	5,083	10,036
5	1	*S*	0,450	0,450	0,450	0,447
		C	0,450	0,455	0,450	0,450
	2	*S*	0,910	0,911	0,916	0,920
		C	0,925	0,918	0,923	0,947
	3	*S*	2,956	3,399	4,274	7,920
		C	2,985	3,416	4,311	8,122

Conclusions and prospects for further research. The paper proposes a combined approach to the analysis of complex multiserver queuing systems with absolute priorities. Due to significant difficulties in obtaining exact analytical solutions for such systems, simulation modeling was used to verify and substantiate approximate calculation techniques.

The following have been developed and verified:

1. An analytical approximation for calculating waiting time based on the Weibull distribution, which confirmed its accuracy.
2. A formula for calculating the average duration of the continuous busy period (CBP) with a correction (7) that takes into account the number of channels, load, and coefficient of variation.
3. An approximate formula (8) for estimating the average number of request interruptions.

As shown in Table 3, the comparison of the final results (average residence time) by the simulation model (*S*) and by the developed technique (*C*) demonstrates good consistency. Thus, the assumptions derived from the analysis of simulation experiments allowed creating a simple engineering tool for the approximate calculation and design of multiserver priority systems in conditions where exact analysis is practically impossible.

References:

1. Jaiswal N. K. Priority Queues. New York : Academic Press, 1968. 240 p.
2. Kleinrock L. Queueing Systems. Vol. 1: Theory. New York : Wiley, 1975. 417 p.
3. Dymova H. Calculation of Characteristics of Queuing Systems Using the Erlang Method and Conservation Laws. Інфокомунікаційні та комп'ютерні технології. Київ. № 2(6), 2023 р. С. 60-65. DOI: https://doi.org/10.36994/2788-5518-2023-02-06-06, https://visn-icct.uu.edu.ua/index.php/icct/article/view/140, https://dspace.ksaeu.kherson.ua/handle/123456789/9534
4. Davis R. H. Waiting-Time Distribution of a Multiserver Priority Queueing System. Operations Research. 1966. Vol. 14, No. 1. P. 133–142.
5. Gail H. R., Hantler S. L., Taylor B. A. Analysis of a non-preemptive priority multiserver queue. Advances in Applied Probability. 1988. Vol. 20, No. 4. P. 852–879.
6. Koba O. V. Teoriya masovoho obsluhovuvannya : pidruchnyk [Theory of mass service: textbook]. Kyiv: NTUU "KPI", 2012. 244 p. [in Ukrainian]