

УДК 004:62-50

doi: 10.53816/23061456_2025_7-8_36

**МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ПРОЦЕССОВ ОБРАБОТКИ ЗАДАЧ
С ДИНАМИЧЕСКОЙ ТРУДОЕМКОСТЬЮ
В РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ
С ПРОГРАММНО-ОПРЕДЕЛЯЕМОЙ АРХИТЕКТУРОЙ**

**A MODEL FOR REPRESENTING TASK PROCESSING PROCESSES
WITH DYNAMIC LABOR INTENSITY IN A DISTRIBUTED COMPUTING
ENVIRONMENT WITH A SOFTWARE-DEFINED ARCHITECTURE**

T.C. Швец

T.S. Shvets

Военно-космическая академия им. А.Ф. Можайского

В статье предложена модель обработки задач с переменной вычислительной сложностью в распределенной среде с программно-определяемой архитектурой. Основу модели составляют три ключевых компонента: механизм отрицательных заявок для управления перегрузками, контейнерная виртуализация для динамического масштабирования ресурсов и теория сетей очередей для анализа производительности. Модель представления процессов обработки задач с динамической трудоемкостью в распределенной вычислительной среде с программно-определяемой архитектурой ставит перед собой задачи оптимизации процессов вычислений, управления вычислительными ресурсами и обеспечения минимального времени выполнения задач. Такая модель должна обеспечивать высокую производительность, эффективное управление ресурсами и автоматизацию процессов адаптации к изменяющимся условиям. Основное внимание уделено динамическому управлению ресурсами при нестабильной нагрузке.

Ключевые слова: отрицательные заявки, контейнерная виртуализация, сети очередей, динамическая трудоемкость.

The article proposes a model for processing tasks with variable computational complexity in a distributed environment with a software-defined architecture. The model is based on three key components: a negative request mechanism for congestion management, container virtualization for dynamic resource scaling, and queue network theory for performance analysis. The model of representation of processes for processing tasks with dynamic labor intensity in a distributed computing environment with a software-defined architecture sets itself the task of optimizing computing processes, managing computing resources and ensuring minimal task execution time. Such a model should ensure high productivity, efficient resource management, and automation of adaptation processes to changing conditions. The main focus is on dynamic resource management under unstable load.

Keywords: negative applications, container virtualization, queue networks, dynamic labor intensity.

Введение

Современные информационные системы разрабатываются и функционируют с использованием широкого спектра технологий. Среди них, как одну из наиболее перспективных, необходимо выделить технологию, базирующуюся на микросервисной архитектуре с контейнерной виртуализацией.

Инфраструктура микросервисной информационной системы с контейнерной виртуализацией характеризуется динамическим изменением вычислительной структуры и параметров узлов (рис. 1).

В частности, при рассмотрении процессов масштабирования сервисов, реализуемых на основе контейнеризованных приложений, следует учитывать временные затраты на запуск необходимого числа контейнеров, проверку их работоспособности, а также остановку контейнеров, использование которых в данный момент не предполагается [4].

Длительность отдельных операций запуска, остановки контейнеров, проверки работоспособности (с учетом типа запущенного в контейнере приложения-сервиса) представляют собой случайные величины, учет которых необходимо реализовывать в соответствующих моделях.

В данной модели необходимо дополнительно учесть события, характеризующие управляемые воздействия на реконfigurирование в инфраструктуре. Причем учитывая случайный

характер моментов поступления в систему задач с динамической трудоемкостью в модель, необходимо учесть дополнительный поток указанных событий.

В данной работе представляется возможным представить данный поток в виде потока отрицательных заявок в СеМО.

Модель неэкспоненциальной G-сети Исходные данные и описание модели

Дано:

Разомкнутая СеМО состоит из M узлов. Сеть представлена в виде ориентированного графа, нулевая вершина которого (исток) идентифицирует источник заявок, вершины $1, 2, \dots, M$ — узлы сети, а $(M+1)$ -я вершина — выход (сток) сети. Граф без циклов. В сеть поступает два потока заявок: положительных (целевых) и отрицательных (событий, приводящих к потере заявок в узлах сети). Предполагается, что указанные потоки — простейшие. Переходы целевых заявок между узлами определяются стохастической матрицей передач $R^+ = \{r_{ij}^+\}$, $i, j = \overline{0, M+1}$, где r_{ij}^+ — вероятность перехода заявки из i -го в j -й узел. Отрицательные заявки распределяются по рабочим узлам согласно вектору вероятностей $R^- = \{r_i^-\}$, причем они в узлах не обслуживаются. Прибывающая в узел отрицательная заявка инициирует процесс удаления всех положительных заявок (как находящихся на обслуживании, так и находящихся в очереди). После этого

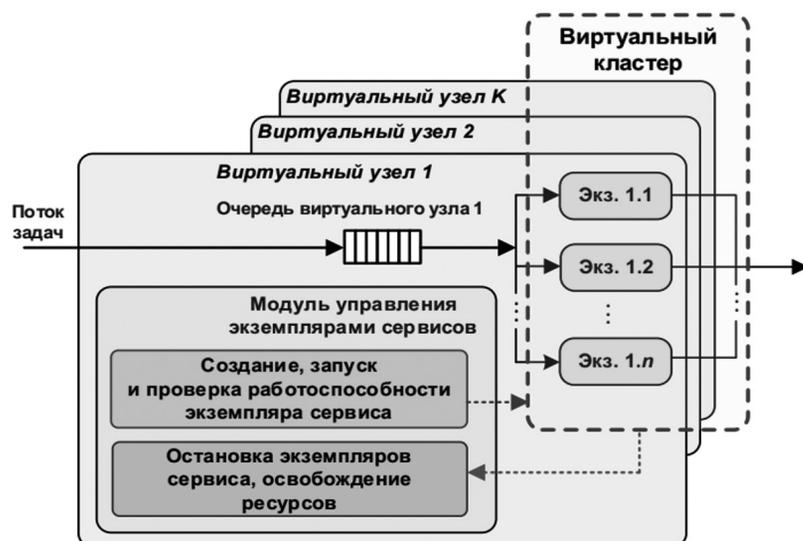


Рис. 1. Модель представления процесса обработки задач масштабируемым сервисом

отрицательная заявка мгновенно покидает узел. Положительная заявка, обслуженная в i -м узле, с вероятностью r_{ij}^+ переходит в j -й узел $r_{i(M+1)}^+$ покидает сеть. Каждая пораженная целевая заявка заново генерируется в источнике и отправляется в сеть. Дисциплины обслуживания в узлах FIFO. Очереди в узлах сети не ограничены. Для каждого i -го узла сети заданы моменты распределения чистой длительности обслуживания положительных заявок $\{b_{ij}\}$, $i = \overline{1, L}$, а также число каналов обслуживания n_i . Режим работы сети — стационарный.

Требуется:

Найти начальные моменты распределения времени пребывания положительной заявки в сети от первой ее генерации до попадания в сток.

Построить дополнительную функцию распределения времени пребывания положительной заявки в сети.

Традиционно, для расчета вероятностно-временных характеристик сложных СеМО (не имеющих аналитического решения) применяются аппроксимационные методы, в частности методы на основе потокоэквивалентной декомпозиции [2].

Особенности потокоэквивалентной декомпозиции G-сети

Для исследования характеристик СеМО в условиях, выходящих за пределы теоремы ВСМР, применяют аппроксимационные методы, основанные на потокоэквивалентной декомпозиции сети [1, 4].

Суть данных методов заключается в следующем. Задаются начальные приближения к потокам через узлы сети, после чего составляются и решаются уравнения баланса относительно средних значений входящих в узлы потоков. По результатам этих расчетов производится итерационный пересчет потоков, после стабилизации которых, каждый узел рассчитывается независимо как изолированная СМО. На заключительном этапе вычисляются агрегированные характеристики сети (распределение времени пребывания заявки в сети).

В зависимости от типа СеМО, узлов и характеристик потоков в сети этапы метода могут модифицироваться. Например, при расчете рассматриваемой модели G-сети, с простейшими

потоками и неэкспоненциальными узлами, в общую схему потокоэквивалентной декомпозиции необходимо внести следующие изменения.

Предположение о простейших потоках, циркулирующих в сети, после решения уравнений баланса потоков позволяет сразу перейти к расчету узлов сети как независимых СМО. Но так как «убитые» положительные заявки в узлах повторно генерируются в источнике и отправляются в сеть, то возникает необходимость в пересчете суммарной приведенной интенсивности потока положительных заявок с учетом повторно отправленных из источника.

При расчете узлов СеМО каждый узел представляется как СМО $M/H_2/n$ с учетом воздействий отрицательных заявок. Это обстоятельство требует внесения изменений в диаграммы и соответствующие им матрицы переходов, а также вывода условия баланса заявок с учетом ухода положительных заявок из узла из-за воздействия отрицательных.

Методика расчета временных характеристик. Этапы метода. Общая идея метода

Модель G-сети представляет собой описание функционирования сложной системы в условиях возмущений, которые описываются с помощью отрицательных заявок. Хотя указанные возмущения имеют однозначное представление в виде вектора передач R^{\wedge} - и суммарной средней интенсивности их потока Λ^{\wedge} -, их влияние на элементы сети и на общий целевой показатель системы является неопределенным. Кроме того, во-первых, интенсивность и направление деструктивного воздействия среды могут непредсказуемо изменяться; во-вторых, состояние системы является также трудно предсказуемым из-за сложности внутренней структуры и проходящих в ней процессов; в-третьих, модель системы в виде G-сети описывает ее функционирование с учетом воздействий, что не позволяет выделить из нее модель объекта и модель среды.

В настоящей работе предлагается итерационный метод организации обработки заданий на основе изменения структуры сети и параметров ее узлов. Целевым показателем является время выполнения операций (полной обработки данных) с заданной вероятностью. Модификация

сети проводится итерационно на двух уровнях. В каждой итерации может проводиться одна операция изменения структуры сети, в рамках которой проводится весь цикл изменения параметров для выбранной структуры.

Алгоритмическая реализация метода

При расчете сети обслуживания заявка должна пройти без «убийства» случайную, вообще говоря, последовательность ее узлов, определяемую маршрутной матрицей. Здесь ключевую роль играет вероятность q успешного прохождения узла при условии, что заявка до него добралась. Поток «эффективных» (находящих жертву) отрицательных заявок имеет интенсивность $\lambda^-(1-p_0)$. Здесь p_0 — вероятность свободного состояния системы. Соответственно доля убитых положительных заявок составит $(1-p_0)\lambda^-/\lambda^+$, а интересующая нас вероятность «счастливого» прохождения узла

$$q = 1 - \frac{\lambda^-}{\lambda^+}(1-p_0). \quad (1)$$

Для расчета временных показателей работы сети необходимо вычисление моментов распределения времени пребывания в узле отдельно для обслуженной и убитой заявок. Для обоих вариантов время ожидания распределено одинаково. Время пребывания на обслуживании в первом варианте окажется «штатным», а во втором — усеченным (остаточное время обслуживания, расчет моментов распределения которого общеизвестен). Соответственно моменты распределения времени занятости канала окажутся средневзвешенными, и именно они будут определять динамику продвижения очереди. Очевидна необходимость итерационного пересчета характеристик узлов.

Нумерация вершин

Ключевым элементом обсуждаемой расчетной схемы является расчет длительности неполного маршрута в зависимости от места гибели положительной заявки, что невозможно при повторном прохождении узлов. В связи с этим приходится делать допущение об отсутствии в сети циклических маршрутов (исключая повторную

генерацию убитой заявки). Тогда можно перенумеровать узлы на основе отношений предшествования — например, с помощью известного алгоритма Форда. Перенумерация позволяет искать предшественников узла j только среди узлов с номерами $i = \overline{1, j-1}$. Более того, на уровне программной реализации имеет смысл просматривать этот список в обратном порядке и прекращать просмотр при первом же нарушении непосредственного предшествования.

Далее мы всюду предполагаем, что упомянутая перенумерация проведена.

Особенности расчета узлов

Прибытие отрицательной заявки «выбивает» из канала положительную, уменьшая число заявок в системе на единицу, и в этом смысле эквивалентно завершению обслуживания. Однако этот эффект не растет по числу занятых каналов и, следовательно, не может быть учтен пересчетом их быстродействия. С другой стороны, его нельзя отразить и через уменьшение интенсивности прибытия положительных заявок λ^+ , поскольку прибытие отрицательной заявки в свободную систему не оказывает на нее никакого действия. Следовательно, для обхода узлов с положительными и отрицательными заявками необходимо разрабатывать специальные методы. Метод расчета узлов с положительными и отрицательными заявками представлен в [5].

Баланс межузловых потоков

Обозначим λ^+ суммарную интенсивность внешнего потока и зададим матрицу вероятностей межузловых переходов $R = \{r_{i,j}\}, i, j = \overline{0, M+1}$. Здесь узел «0» — источник заявок, « $M+1$ » — сток (при попадании в него фиксируется окончание пребывания заявки в сети), а прочие узлы считаются рабочими и производят фактическое обслуживание. Заявка, получившая полное обслуживание, переходит в один из очередных рабочих узлов или покидает сеть; «убитая» заявка мгновенно вновь появляется в источнике и адресуется в один из рабочих узлов.

Указанные соображения приводят к системе линейных алгебраических уравнений относительно интенсивностей $\{\lambda_i^+\}$, входящих в узлы потоков:

$$\lambda_i^+ = \Lambda^+ r_{0,i} + \sum_{j=1}^M \lambda_j^+ [q_j r_{j,i} + (1 - q_j) r_{0,i}], \quad i = \overline{1, M}. \quad (2)$$

Второе слагаемое в квадратных скобках соответствует доле повторно генерируемых заявок, направляемых из источника в i -й узел. Отметим, что вероятности $\{q_j\}$ зависят от интенсивностей входящих в узлы положительных и отрицательных потоков, что определяет необходимость решения системы (2) внутри итерационного цикла. На первом шаге для всех j разумно принять

$$q_j = \int_0^{\infty} e^{-\lambda_j t} dB_j(t), \quad (3)$$

(вероятность неприбытия отрицательных заявок за время обслуживания положительной), а далее уточнять их согласно (1).

Конечные результаты решения системы (2) следует проверить на загрузку: для всех узлов должно выполняться условие $n/b + \lambda^- > \lambda^+$ (максимальная интенсивность ухода заявок должна превышать интенсивность их прибытия). Таким образом, необходимо выполнение условия

$$\frac{n}{b} > \lambda^+ - \lambda^-. \quad (4)$$

При его нарушении необходима корректировка исходных данных. Это могут быть изменение маршрутизации заявок, увеличение количества каналов в критических узлах или повышение быстродействия каналов — с учетом экономических и иных соображений.

По окончательному решению системы (2) можно определить приведенную интенсивность потока из источника, состоящую из внешнего потока и потоков заявок, обслуживание которых было прервано в различных узлах:

$$\Lambda_{\Sigma}^+ = \Lambda^+ + \sum_{j=1}^M \lambda_j (1 - q_j). \quad (5)$$

Формирование конечного результата

В этом пункте при организации вычисления сумм учитываются соображения, изложенные в разделе о нумерации вершин. Первый узел имеет только одного предшественника (источ-

ник), единичную вероятность предшественника и нулевую задержку в нем.

После завершения итераций для каждого узла вычисляются:

а) моменты $\{\omega_j\}$ распределения времени ожидания в очереди; моменты $\{\upsilon_j\}$ пребывания в узле обслуженной заявки и $\{\hat{\upsilon}_j\}$ — прерванной;

б) кумулянтная вероятность Q_j^i добраться до j -го узла. Она равна отношению интенсивности входящего в узел потока к общей интенсивности потока из источника:

$$Q_j^i = \lambda_j / \Lambda_{\Sigma}^+.$$

в) вероятность успешного прохождения j -го узла

$$Q_j = q_j Q_j^i.$$

г) вероятность убиения заявки именно в j -м узле

$$\hat{Q}_j = (1 - q_j) Q_j^i.$$

д) кумулянтные моменты распределения времени пребывания в сети j -убитой заявки

$$\hat{V}_j = \left(\sum_{i=1}^{j-1} r_{i,j} V_i \right) * \hat{\upsilon}_j$$

и аналогичные моменты для j -прошедшей

$$V_j = \left(\sum_{i=1}^{j-1} r_{i,j} V_i \right) * \upsilon_j,$$

(здесь и далее * означает оператор свертки в моментах).

Теперь можно рассчитать наборы моментов средней длительности жизни в цикле для убиенной заявки

$$A = \sum_{j=1}^M \hat{Q}_j \hat{V}_j$$

моментов времени пребывания в сети при успешном (полном) ее прохождении

$$B = \sum_{j=1}^M r_{j,M+1} V_j,$$

и, наконец, вероятность «счастливого» прохождения сети

$$x = \sum_{j=1}^M r_{j,M+1} Q_j.$$

Искомое распределение полного времени ожидания ответа сети мы сначала получим в терминах преобразований Лапласа — Стилтеса (ПЛС), построенных по моментам соответствующих распределений. Пусть

$\alpha(s)$ — ПЛС распределения длительности неполного прохода,

$\beta(s)$ — то же для успешного прохода.

Тогда результирующая ПЛС

$$\gamma(s) = \sum_{k=1}^{\infty} [(1-x)\alpha(s)]^{k-1} x\beta(s) = \frac{x\beta(s)}{1-(1-x)\alpha(s)}.$$

Умножим обе части этого равенства на знаменатель правой и разложим все ПЛС по степеням параметра. В коэффициенты этого разложения будут входить моменты известных и искомого распределений. Приравнявая множители при одинаковых степенях s в левой и правой частях равенства, получаем рекуррентное выражение для моментов времени реакции сети

$$g_k = b_k + \frac{1-x}{x} \sum_{i=0}^{k-1} \frac{k!}{i!(k-i)!} g_i a_{k-i}, \quad k = 1, 2, \dots$$

при начальных

$$g_0 = 1, \quad g_1 = b_1 + \frac{1-x}{x} a_1.$$

По найденным моментам можно построить аппроксимацию дополнительной функции распределения времени пребывания заявки в сети до ее «полного удовлетворения».

Проверка адекватности модели G-сети и численного метода расчета.

Исходные данные

Вычислительная сеть представлена СеМО с тремя узлами S1, S2, S3. Узлы S0 и S4 — источник заявок и сток соответственно. Из источника в сеть поступают два пуассоновских потока положительных и отрицательных заявок со средними интенсивностями $\Lambda^+ = 3$ з/с и $\Lambda^- = 1,5$ з/с. Количество обслуживающих приборов в узлах примем $n = n_2 = n_3 = 2$. Обслуживание положительных заявок в узлах осуществляется со средними длительностями $b_1 = 0,25$ с, $b_2 = 0,3$ с, $b_3 = 0,15$ с и с коэффициентом вариации $v = 1,9$. Воспользуемся процедурами подбора параметров Г-аппроксимации и расчета моментов Г-распределения по подобранным параметрам. В результате получим для каждого узла три начальных момента распределения обслуживания (табл. 1).

Структура сети, направления и соответствующие им вероятности переходов заявок представлены на рис. 2. и в табл. 2.

Таблица 1

Подбор параметров и расчет начальных моментов Г-распределения

| | Исходные данные | | Параметры Г-распределения | | Моменты Г-распределения | | |
|--------|-----------------|-----|---------------------------|--------|-------------------------|--------|--------|
| | b | v | a | μ | b_1 | b_2 | b_3 |
| Узел 1 | 0,25 | 1,9 | 0,277 | 1,108 | 0,25 | 0,2881 | 0,5921 |
| Узел 2 | 0,3 | | 0,277 | 0,9234 | 0,3 | 0,4149 | 1,0231 |
| Узел 3 | 0,15 | | 0,277 | 1,8467 | 0,15 | 0,1037 | 0,1279 |

Таблица 2

Матрица передач R^+

| Узлы | S0 | S1 | S2 | S3 | S4 |
|------|----|-----|-----|-----|-----|
| S0 | 0 | 0,5 | 0,5 | 0 | 0 |
| S1 | 0 | 0 | 0 | 0,5 | 0,5 |
| S2 | 0 | 0 | 0 | 0,5 | 0,5 |
| S3 | 0 | 0 | 0 | 0 | 1 |
| S4 | 0 | 0 | 0 | 0 | 0 |

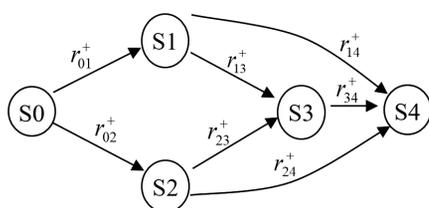


Рис. 2. Структура G-сети с переходами заявок согласно матрице R^+

Результаты расчета G-сети численным методом

| | q_1 | q_2 | q_3 | ω_1 | ω_2 | ω_3 | v_1 | v_2 | v_3 |
|------|-------|-------|-------|------------|------------|------------|--------|--------|-------|
| S1 | 0,078 | 0,036 | 0,017 | 0,038 | 0,009 | 0,002 | 0,145 | 0,044 | 0,019 |
| S2 | 0,139 | 0,072 | 0,039 | 0,068 | 0,017 | 0,005 | 0,183 | 0,064 | 0,030 |
| S3 | 0,012 | 0,003 | 0,001 | 0,006 | 0,001 | 0,0002 | 0,078 | 0,015 | 0,004 |
| Сеть | – | – | – | – | – | – | 0,6602 | 0,8397 | – |

Отрицательные заявки суммарной интенсивности $\Lambda^- = 1,5$ распределены по рабочим узлам с вероятностями $\{0,3; 0,3; 0,4\}$ соответственно.

Расчет G-сети численным методом

Произведем расчет узлов сети, найдем вероятности поражения целевых заявок в них и итерационно пересчитаем потоки в сети.

После пересчета потоков перейдем к расчету сети разработанных процедур. Результаты расчета приведены в табл. 3.

Заключение

Предложенный метод организации обработки заданий позволяет на основе модели G-сети сформировать требования к значениям параметров и структуре сети для достижения требуемого уровня целевого показателя функционирования системы в условиях влияния деструктивных факторов.

Разработанный метод расчета временных характеристик неэкспоненциальной G-сети позволяет получить значение времени обработки заявки от первого ее появления до полного обслуживания сетью с учетом вынужденных повторных генераций и отправки на обслуживание из-за воздействия отрицательных заявок.

Список источников

1. Вычислительные системы и их программное обеспечение. Модели, методы и средства исследования: учебник; под ред. Ю.И. Рыжикова и А.Д. Хомоненко. МО РФ, 1996. 312 с.

2. Рыжиков Ю.И. Алгоритмический подход к задачам массового обслуживания: монография. СПб.: ВКА им. А.Ф. Можайского, 2013. 496 с.

3. Рыжиков Ю.И., Хомоненко А.Д. Итеративный метод расчета многоканальных систем с

произвольным распределением времени обслуживания // Проблемы управления и теория информации. 1980. № 3. С. 32–38.

4. Рыжиков Ю.И. Компьютерное моделирование систем с очередями: курс лекций. СПб.: ВКА им. А.Ф. Можайского, 2007. 163 с.

5. Швец Т.С. Имитационное моделирование. Теория и практика (ИММОД-2023) // Расчет характеристик многоканального узла неэкспоненциальной сети с отрицательными заявками с групповым удалением положительных. 2023. С. 770–779.

6. Радченко В.Г., Богданов И.Н. Распределенные вычислительные системы. СПб.: БХВ-Петербург, 2021. 480 с.

7. Курячий А.В., Башлаков К.А. Программно-конфигурируемые сети и виртуализация. М.: ДМК Пресс, 2020. 320 с.

8. Лось Д.А., Королюк В.В. Теория массового обслуживания в современных вычислительных системах. М.: Физматлит, 2019. 256 с.

9. Петров А.М., Соколов И.К. Алгоритмы динамического распределения ресурсов в гетерогенных облачных средах // Информационные технологии. 2022. Т. 28. № 5. С. 45–52.

10. Кузнецов Е.Р., Михайлов Д.Ю. Механизмы отрицательных заявок в распределенных системах реального времени // Программная инженерия. 2021. № 8. С. 12–20.

11. Громов Ю.Ю., Белов А.А. Контейнерная виртуализация в задачах обработки данных с динамической нагрузкой // Научно-технические ведомости СПбГПУ. 2023. Т. 14. № 3. С. 78–89.

12. Отчет НИР «Разработка моделей обработки задач с динамической трудоемкостью». М.: ИППИ РАН, 2021. 95 с.

Resources

1. Computing systems and their software. Models, methods and research tools: textbook ed.

by Yu.I. Ryzhikov and A.D. Khomonenko. Ministry of Defense of the Russian Federation, 1996. 312 p.

2. Ryzhikov Yu.I. Algorithmic approach to queuing tasks: monography, St. Petersburg: Mozhaisky Central Research Institute, 2013. 496 p.

3. Ryzhikov Yu.I., Khomonenko A.D. Iterative calculation method for multichannel systems with arbitrary service time distribution // Management problems and information theory. 1980. No 3. Pp. 32–38.

4. Ryzhikov Yu.I. Computer modeling of systems with queues: a course of lectures. St. Petersburg: Mozhaisky Academy of Fine Arts, 2007. 163 p.

5. Shvets T.S. Simulation modeling. Theory and practice (IMMOD-2023) // Analysis of the characteristics of a multichannel node of a non-exponential network with negative signals with group deletion of positive ones. 2023. Pp. 770–779.

6. Radchenko V.G., Bogdanov I.N. Distributed computing systems. St. Petersburg: BHV-Petersburg, 2021. 480 p.

7. Kuryachiy A.V., Bashlakov K.A. Software-configurable networks and virtualization. Moscow: DMK Press, 2020. 320 p.

8. Los D.A., Korolyuk V.V. Theory of queuing in modern computing systems. Moscow: Fizmatlit, 2019. 256 p.

9. Petrov A.M. Sokolov I.K. Algorithms of dynamic resource allocation in distributed cloud environments // Information Technologies. 2022. Vol. 28, No 5. Pp. 45–52.

10. Kuznetsov E.R., Mikhailov D.Y. Mechanisms of negative applications in distributed real-time systems // Software Engineering. 2021. No 8. Pp. 12–20.

11. Gromov Yu.Yu., Belov A.A. Container virtualization in dynamic load data processing tasks // Scientific and Technical Bulletin of St. Petersburg State University. 2023. Vol. 14, No 3. Pp. 78–89.

12. Research report «Development of models for processing tasks with dynamic labor intensity». M.: IPPI RAS, 2021. 95 p.