УДК 004.94 DOi 10.30837/ bi.2024.1(100).05

Д. І. Гольдінер[1], О. І. Матвієнко[2]

[1]Аспірант, Харківський національний університет радіоелектроніки, Харків, Україна;
e-mail: denys.holdiner@nure.ua; ORCID ID: 0000-0002-1456-1867

[2]Канд. техн. наук, доцент, доцент кафедри прикладної математики,
Харківський національний університет радіоелектроніки, Харків, Україна;
e-mail: olha.matviienko@nure.ua; ORCID ID: 0000-0001-7492-7616

ЗМЕНШЕННЯ ЙМОВІРНОСТІ ВІДМОВИ В СИСТЕМАХ МАСОВОГО ОБСЛУГОВУВАННЯ З ОБМЕЖЕНОЮ ЧЕРГОЮ ІЗ ЗАСТОСУВАННЯМ ПРІОРИТЕЗАЦІЇ ЗА РОЗМІРОМ ТА ШТУЧНОГО ІНТЕЛЕКТУ

Стаття присвячена збільшенню ефективності обробки заявок у багатоканальних системах масового обслуговування з обмеженою чергою та відмовами у випадку її переповнення. Предметом даної статті є: методи та підходи до оптимізації обробки потоків заявок. Метою роботи є: запропонувати новий підхід до пріоритезації та балансування категорій вимог задля зменшення ймовірності відмови. Завдання статті полягає у: формулюванні досліджуваної системи масового обслуговування; визначенні джерела оптимізації; описі методу розбиття загального потоку вимог на категорії; переліку підходів до оцінки розмірів заявки; визначенні алгоритму обробки заявок із застосуванням пріоритезації за часом обробки; пропозиції рішення проблем оцінки навантаження та балансування пріоритетів із застосуванням штучного інтелекту. Використовуються такі методи: теорія масового обслуговування, UML діаграми, штучний інтелект. Було здобуто наступні результати: запропоновано підхід до зменшення ймовірності відмови в багатоканальних системах масового обслуговування з обмеженою чергою, за рахунок збільшення пріоритетів менших заявок; запропоновано методи оцінки складності вимог та розбиття загального потоку заявок на категорії згідно з їхнім розміром; Запропоновано підхід до балансування пріоритетів за допомогою штучного інтелекту.

МАСОВЕ ОБСЛУГОВУВАННЯ, ПРІОРИТЕТИ, ШТУЧНИЙ ІНТЕЛЕКТ, НАВЧАННЯ, ШТУЧНА НЕЙРОННА МЕРЕЖА, ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ, ПРИЙНЯТТЯ РІШЕНЬ, МОДЕЛЬ ПОВЕДІНКИ, АДАПТАЦІЯ, БАЛАНСУВАННЯ НАБОРІВ ДАНИХ, ЗАДАЧА КАТЕГОРІЗАЦІЇ

D. Goldiner, O. Matviienko. Rejection probability reduction in queueing systems with limited queue using size based prioritization and artificial intelligence. The article is dedicated to enhancing the efficiency of request processing in multi-channel mass service systems with limited queue and refusals in case of overflow. The subject of this article is the methods and approaches to optimizing the processing of request flows. The aim of the work is to propose a new approach to prioritizing and balancing request categories to reduce the probability of refusals. The tasks of the article include: formulating the studied mass service system; determining the source of optimization; describing the method of dividing the overall request flow into categories; listing approaches to determining the request sizes; defining the algorithm for processing requests using size-based prioritization; proposing solutions to load assessment and priority balancing problems using artificial intelligence. The following methods are used: mass service theory, UML diagrams, artificial intelligence. The following results were obtained: a method to reduce the probability of refusals in multi-channel mass service systems with a limited queue by increasing the priority of smaller requests was proposed; methods for assessing request complexity and dividing the overall request flow into categories according to their complexity were proposed; an approach to balancing priorities using artificial intelligence was proposed.

MASS SERVICE SYSTEM, PRIORITIES, ARTIFICIAL INTELLIGENCE, TRAINING, ARTIFICIAL NEURAL NETWORK, APPLICATION OF NEURAL NETWORKS, DECISION MAKING, BEHAVIOR MODEL, ADAPTATION, DATA BALANCING, CATEGORIZATION TASK

Вступ

У сучасному світі, навколо є безліч різноманітних процесів, що включають в себе обробку потоків вхідних даних. Значне поширення мають прикладні задачі, що можуть бути описані та оптимізовані за допомогою теорії масового обслуговування, наприклад:

– управління запасами;

– людськими ресурсами;

– оптимізація використання обладнання на виробництвах;

– керування взаємодією з клієнтами;

– обробка черг замовлень в торгівлі;

– побудова автоматизації виробництв;

– масштабування систем надання послуг;

– забезпечення збільшення пропускної здатності трафіка в цифрових мережах.

До математичних моделей можуть бути зведені задачі, що виникають в сферах: інформаційно-керуючих систем, системах контролю та управління процесами, телекомунікаційних технологіях та мережах, а також, у менеджменті. До кожної задачі можна підібрати відповідний тип системи масового обслуговування (СМО), який найкраще описуватиме специфіку фактичного процесу [1].

Зазвичай, прикладні задачі, що описуються за допомогою систем масового обслуговування, накладають обмеження, пов'язані з фактичними можливостями масштабування. Більшість задач, що мають

черги, не можуть дозволити собі зняти обмеження на кількість заявок, що перебувають в очікуванні. Також, зазвичай, не є можливим збільшувати чергу кожен раз, як вона переповнюється. Отже, набувають актуальності моделі з обмеженими чергами та відмовами у випадку переповнення. Такі СМО надають математичне обґрунтування та рекомендації до оптимізації процесів задля отримання кращої пропускної здатності. Тобто головна задача поставлена наступним чином: необхідно зменшити ймовірність відмови черговій заявці, що надходить до СМО, не змінюючи при цьому її структуру та основні характеристики.

Серед технологій, які яскраво про себе заявили в останнє десятиріччя, мають місце машинне навчання та штучний інтелект (ШІ).

Після спаду інтересу до штучного інтелекту в 70-х роках, у 80-х ця галузь отримала новий поштовх завдяки розробці експертних систем. Однак, технологія залишалась дуже вузькоспеціалізованою. Галузь машинного навчання значно розвинулася в 90-х роках, коли почали широко застосовуватись нейронні мережі, завдяки алгоритму зворотного поширення помилки. Що, у свою чергу, покращило ефективність моделей та їхню здатність до навчання. Головним стримуючим фактором залишалось обмеження обчислювальних потужностей і доступу до великих об'ємів даних [2]. У 2000-х роках ситуація почала значно покращуватись. На додачу до цього, глибокі нейронні мережі, що складаються з багатьох шарів, продемонстрували видатні результати у задачах розпізнавання образів, обробки мови та гри у складні ігри, як шахи або го. Починаючи з 2010 року ШІ стає все більш доступним як для професійного, так і побутового використання. У повсякденному житті дедалі частіше зустрічаються додатки та сервіси, які застосовують машинне навчання для збільшення ефективності або якості надання послуг. Останніми роками такі технології почали підтримуватись навіть у мобільних пристроях. Кожен наступний етап розвитку штучного інтелекту відкривав нові сфери його застосування, розширюючи можливості для покращення життя.

Це дослідження присвячено пошуку нових підходів до збільшення продуктивності багатоканальних СМО з обмеженою чергою та відмовами.

1. Аналіз поточного стану справ

Як правило, складні системи з високим навантаженням оптимізують за рахунок додавання пріоритетів. Розрізняють наступні види систем з пріоритетом [3, 4]:

– Абсолютний пріоритет – передбачає, що заявки з вищим пріоритетом завжди обслуговуються першими, з можливістю переривання обробки вимог з нижчим пріоритетом;

– Відносний пріоритет – заявки з вищим пріоритетом мають перевагу, але не можуть переривати виконання інших вимог;

– Пріоритет за класами обслуговування – передбачає класифікацію заявок на категорії, кожна з яких має свою чергу та пріоритет. Заявки одного типу обслуговуються за принципом FIFO (перший прийшов перший пішов), але між собою категорії взаємодіють згідно пріоритетів;

– Динамічний пріоритет – у цьому випадку пріоритет заявок може змінюватись з часом в залежності від певних умов та стану системи;

– Випадковий пріоритет – передбачає встановлення пріоритету заявці випадковим чином;

– Комбіновані методи – дозволяють довільну комбінацію раніше згаданих типів пріоритетів задля найкращого забезпечення потреб реальної задачі.

Такі підходи добре діють, але залишається проблема їхнього застосування для випадку, коли задача не регулює пріоритети на вимоги за замовченням. Для таких випадків необхідно напрацювати механізм переходу від звичайних СМО до таких, що використовують систему пріоритетів.

2. Постановка задачі

Будемо розглядати задачу про багатоканальну систему масового обслуговування із обмеженою чергою вимог та відмовою у випадку переповнення черги. У якій до n – однакових каналів обслуговування надходить сімейство з r пуассонівських потоків заявок інтенсивності λ_j, де $j = 1,2,\ldots,r$. Отже, сукупний потік є пуассонівським з інтенсивністю (1):

$$\lambda_{\bar{r}} = \sum_{j=1}^{r} \lambda_j \qquad (1)$$

Якщо на момент приходу нової заявки є хоча б один вільний канал, він негайно починає обробку. У випадку, коли усі канали зайняті, вимога стає останньою до загальної черги ємності k. Заявки покидають чергу для подальшого обслуговування у тій самій послідовності у якій вони надходили на очікування. Канал, що звільнюється від виконання, одразу починає обробку першої в черзі вимоги. При цьому кожна вимога обслуговується тільки одним каналом, і кожен канал може обслуговувати не більше однієї вимоги одночасно. У випадку, якщо вільними є декілька каналів обслуговування, для обробки буде обрано канал випадковим чином [4]. Час, необхідний на обробку однієї вимоги, є випадковою величиною з експоненціальним законом розподілу ймовірностей, та, загалом кажучи, відрізняється для різних потоків вимог (2):

$$F_j(x) = 1 - e^{-\nu_j x}, \nu > 0, j = 1,2,\ldots,r. \qquad (2)$$

В даному випадку ν_j є коефіцієнтом складності задачі. Вимоги, що мають менше значення, будуть обслуговуватись швидше, а з більшим – довше. Існує

безліч поширених практичних застосувань, що вписуються у дану модель, та яким притаманно мати задачі різної тривалості. Отже, задана система підпадає під умовне визначення: $M/M/n/m$, де:

– перша M вказує на Марківський вхідний потік вимог;

– друга M вказує на те що процес обробки вимог також є Марківським;

– n – визначає, що система є багатоканальною, і задає кількість каналів;

– m – описує обмежену ємність системи та визначає кількість місць для очікування $m > 0$.

Кожен вхідний потік вимог має задовольняти наступні вимоги [4]:

– стаціонарність потоку;

– відсутність післядії;

– ординарність.

Функція розподілу загального часу обслуговування вимог потоків всіх розмірів має вигляд (3) [3]:

$$F(x) = \frac{1}{\lambda}\sum_{j=1}^{r}\lambda_j F_j(x), \tag{3}$$

де $F_j(x)$ – функція розподілу часу обслуговування вимоги j-го розміру, що має середній час обслуговування $\frac{1}{\mu_j}$. Переривання обслуговування в досліджуваній системі не допускається. На даному етапі потоки заявок усіх розмірів мають рівний пріоритет, і поєднуються в загальний потік вимог.

Ймовірність надходження нової вимоги довільного розміру до системи за проміжок часу t може бути визначений наступним чином (4):

$$P_i(t) = \frac{(t\lambda_{\bar{r}})^i}{i!}e^{-t\lambda_{\bar{r}}}, \tag{4}$$

де $\lambda_{\bar{r}}$ – сукупна інтенсивність прибуття вимог усіх розмірів, що надходять до системи на одиницю часу, а i – кількість вимог, що присутні у системі, включно з наступною, в момент часу t.

Враховуючи обмеження на кількість вимог, що можуть очікувати на обслуговування у черзі, розрахуємо ймовірність відмови у виконанні вимоги через переповнення черги. Оскільки майбутнє протікання обслуговування, у контексті теорії ймовірностей, не залежить від того, що відбувалось до моменту часу t_0 в силу особливостей ймовірнісного розподілу [3, 4].

Інтенсивність навантаження j-ї категорії знайдемо через рівняння: $\rho_j = \frac{\lambda_j}{n\nu_j}$, де $j = 1,2,\ldots,r$. Тоді загальне навантаження системи можемо розрахувати так (5):

$$R_{zah} = \sum_{j=1}^{r}\rho_j, R_0 = 0, j = 1,2,\ldots,r, \tag{5}$$

Для описаної системи ймовірність відмови дорівнює ймовірності знаходження у системі рівно i вимог на момент надходження чергової заявки довільного розміру. Маємо наступний вираз (6), (7):

$$\begin{cases} 1 \le i \le n, & P_i = \dfrac{n^i R_{zah}^i}{i!}P_0, \\ n < i \le n+m, & P_i = \dfrac{n^n R_{zah}^i}{n!}P_0, \end{cases} \tag{6}$$

$$P_0 = \left[\sum_{i=0}^{n-1}\frac{n^i R_{zah}^i}{i!} + \frac{n^n}{n!}\cdot\frac{R_{zah}^n\left(1-R_{zah}^{m+1}\right)}{1-R_{zah}}\right]^{-1}. \tag{7}$$

Розглядатимемо випадки, коли $R_{zah} > n$, оскільки за такої умови забезпечується наповнення черги і відмова через брак місця для очікування. Отже, ймовірність відмови через переповнення черги може бути розрахована виразом (8), (9):

$$P_{vidm} = P_{n+m} = \frac{n^n R_{zah}^{n+m}}{n!}P_0 \tag{8}$$

$$P_{vidm} = \frac{n^n R_{zah}^{n+m}}{n!}\left[\sum_{i=0}^{n-1}\frac{n^i R_{zah}^i}{i!} + \frac{n^n}{n!}\cdot\frac{R_{zah}^n\left(1-R_{zah}^{m+1}\right)}{1-R_{zah}}\right]^{-1} \tag{9}$$

Даний вираз означає, що запит отримає відмову щодо обслуговування, якщо всі канали й місця очікування будуть зайняті. При цьому розбиття загального потоку заявок на категорії за розміром, саме по собі, не змінює загальну пропускну здатність системи.

Алгоритм дії даної системи зображено на рис.1.

Перед нами постає задача про збільшення пропускної здатності описаної системи задля зменшення загальної ймовірності відмови. При цьому, з точки зору результатів, для нас всі заявки мають однакове значення, незалежно від їхнього розміру та часу перебування в системі.

3. Вирішення проблеми

В даній статті будемо розглядати рішення поставленої задачі за рахунок розбиття загального потоку заявок на сімейство потоків за розміром. З подальшим наданням меншим задачам більшого пріоритету. Суть підходу полягає в тому, що ми від простої СМО переходимо до системи з комбінованими пріоритетами [5]. Цільова схема має гібрид пріоритетів за класами вимог з динамічними, відносними пріоритетами. Це означає, що ми визначимо фіксовану кількість категорій заявок. І сортуватимемо вимоги на вході в систему. При цьому пріоритет буде гнучко підлаштовуватись під стан системи [6]. Надання пріоритету меншим задачам дозволить зменшити загальну ймовірність відмови через переповнення черги в системі. Такий вибір грунтується на тому, що причина відмови залежить від кількості задач, а не їх складності. На відміну від продуктивності системи, яка у свою чергу, залежить від часу, необхідного на обробку заявки, і не залежить від кількості заявок в черзі. Відповідно, надаючи перевагу меншим задачам, ми можемо збільшити кількість опрацьованих задач за відрізок часу. І таким чином мінімізувати кількість відмов [5, 6].

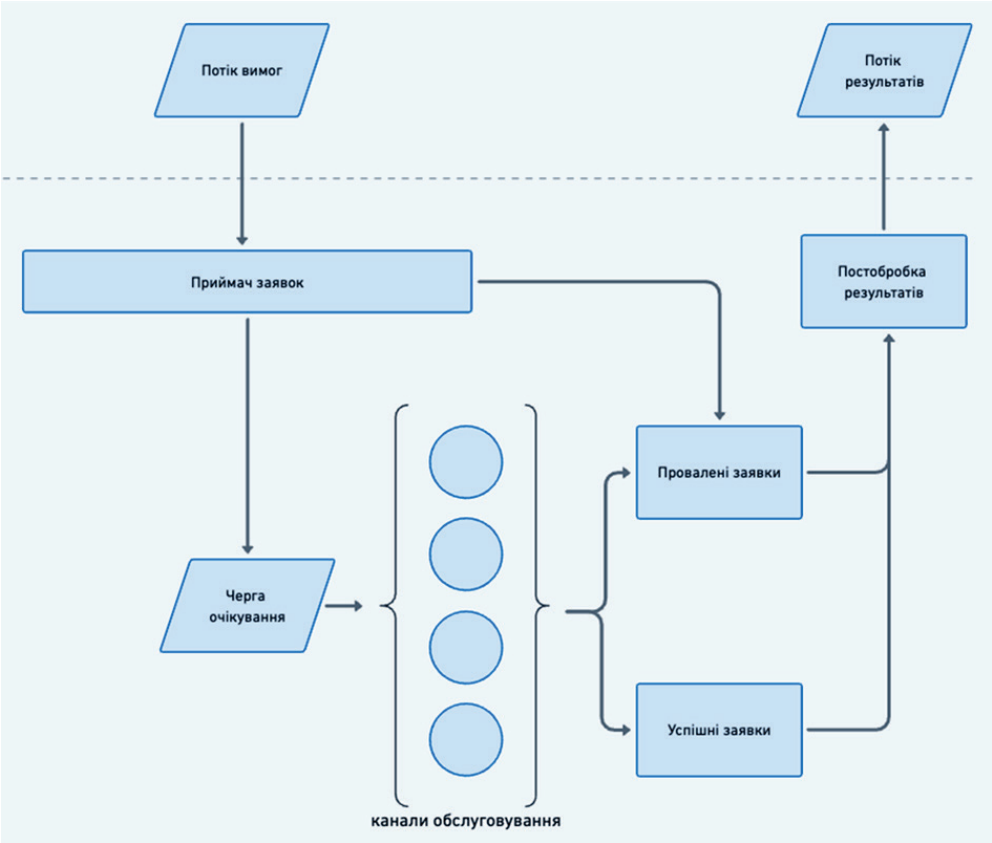

Рис. 1. Алгоритм обробки заявок

Встановлення пріоритетів для вимог, що очікують – один із ефективних способів керування розмірами черги та часом перебування в ній [5]. Для системи з пріоритетами всі вимоги поділяються на категорії, а заявки більш високої категорії при обслуговуванні мають певні переваги перед вимогами з нижчим пріоритетом. Оскільки ми не маємо на меті припинити обслуговування довгих вимог, введемо кілька додаткових правил. По перше – ми не перериватимемо обробку вимоги, навіть якщо вона має нижчий пріоритет (застосування відносного пріоритету). По друге – наявність заявок вищої категорії в черзі не забороняє взяття в роботу менш важливої задачі, а тільки надають додаткових преференцій при виборі. Для кількісної оцінки ефективності системи масового обслуговування з пріоритетами використовують ті самі характеристики, як і для простої системи. Але, на додачу до загальних показників, це робиться для кожної категорії окремо.

Оскільки початковий потік вимог є не ранжованим, постає потреба у визначенні способу його розбиття на окремі категорії заявок. Цю задачу буде вирішувати модуль пріоритезації. В його обов'язки входитиме:

– оцінка складності заявки та призначення певної категорії;

– оцінка поточного завантаження системи та ведення статистики щодо допущення задач різної складності;

– прийняття рішення про надання вимоги на виконання.

3.1. Оцінка складності

За критерій розбиття ми обираємо час, необхідний для обробки вимоги. Оскільки саме ця ознака повинна надати нам підстави для ефективнішого планування роботи. Для оцінки складності заявки можна застосувати один з наступних підходів або їхню комбінацію:

– самоідентифікація заявки;

– особливості функції ймовірнісного розподілу часу обслуговування загального потоку заявок;

– застосування навченої моделі штучного інтелекту.

Найпростішим рішенням може бути делегація зобов'язання за визначення розміру на саму заявку. Тобто, в момент класифікації вимога має надати чисельне значення еквівалентне рівню складності її обробки. Таким чином, алгоритм категоризації може бути суттєво спрощений. Однак, постають кілька недоліків, а саме: заявка може надати недостовірну інформацію; вимоги різного походження можуть використовувати відмінні відносні шкали оцінки складності. Для усунення першої проблеми ми можемо скомбінувати даний підхід з більш достовірними опціями. У другому випадку нам знадобиться напрацювати алгоритм узгодження оцінок.

При застосуванні функції розподілу початкового вхідного потоку вимог нам буде необхідно зібрати певну статистику залежності часу виконання від параметрів. Наступним етапом буде визначення того, за яких вхідних даних час обслуговування потрапляє в зону математичного очікування, а за яких – відхиляється в менший або більший бік. Завдяки цьому, для подальших надходжень матимемо алгоритм категоризації за розміром та призначення пріоритету.

На відміну від попереднього підходу, застосування штучного інтелекту (ШІ) вимагає заздалегідь виконаного навчання моделі під потреби конкретної задачі [7]. Однак, якість визначення категорії буде вищою за рахунок більш адаптивного принципу дії ШІ та машинного навчання. Додатково можна продовжувати вдосконалення моделі прямо під час обробки потоків заявок. Тим самим гнучко підлаштовуватись під зміни у вхідних даних.

Застосування двох принципів одночасно може підвищити якість передбачення складності задачі та покращити ефективність оптимізації [8]. В такому разі аналіз функції розподілу може використовуватись для побудови початкових припущень, які в подальшому будуть уточнюватись за допомогою штучного інтелекту.

3.2. Оцінка поточного завантаження

Важливим етапом оптимізації є контроль поточного завантаження системи. При цьому нас не цікавить кількість одночасно працюючих каналів обслуговування. Оскільки, з точки зору модуля пріоритезації, всі сценарії, за яких система працює з недобором роботи, означають відсутність необхідності розділяти вхідний потік [3]. Для правильного функціонування необхідно збирати інформацію щодо поточного складу черги. Таким чином, ми будемо розуміти відносно чого ми пріоритезуємо задачі. Відповідно, ширина часових діапазонів (складності вимог) для категорій задач та загальна кількість класів заявок залежить від розміру черги.

Унаслідок використання пріоритету за класами обслуговування під кожен клас виділяється окрема черга [8, 9]. Це забезпечить збереження послідовності обслуговування заявок, що були розподілені до однієї категорії. Розміри цих черг залежать від функції ймовірнісного розподілу, що регулює час обробки заявок загального потоку вимог та визначаються перед початком роботи системи. Разом з тим, наповненість кожної з черг є важливою ознакою. За такої будови наша система буде давати відмову тільки тим заявкам, які неможливо додати в чергу згідно з визначеною категорією [7]. Відповідно, ми маємо неперервно слідкувати за наповненістю кожної з черг категорій задля розуміння завантаженості СМО. Для цього використовуватимемо метрику – кількість вільних місць на певний клас заявок.

3.3. Балансування пріоритетів

Головним завданням, яке вирішує модуль пріоритезації є прийняття рішення про обрання класу заявок, що використовуватиметься як джерело для постанови в чергу, з якої беруть собі роботу канали обслуговування. З технічної точки зору це означає, що даний елемент системи, окрім черги, яка використовується для подачі вимог на виконання, взаємодіє з окремими чергами для задач кожного з класів.

Таким чином, на рис. 2 можемо побачити оновлений алгоритм для заданої СМО. Зміни торкнулися передусім механізму прийому заявок в систему, оскільки черга виконання залишилась тою самою. Але тепер їй передує аналіз та сортування заявок по чергах відповідних розмірів. Після чого, враховуючи поточний склад черги виконання, а також наповненість черг по категорії заявок, відбувається прийняття рішення про постанову вимоги певного розміру до черги виконання.

Застосування такого рішення дозволяє уникнути додаткової категорії відмов та збільшити пропускну здатність системи. Оскільки кількість місць очікування в системі збільшилась, маємо зміни у формулі, яка описує загальну ймовірність відмови в системі. Для найгіршого сценарію, коли всі заявки, що надходять до системи мають один розмір отримаємо наступний вираз (10):

$$P_{vidm} = P_{n+m+m_j} = \frac{n^n R_{zah}^{n+m+m_j}}{n!} P_0, \qquad (10)$$

де m – кількість місць очікування в черзі з якої отримують заявки канали обслуговування, та m_j – розмір черги для j-ї категорії, $j = 1, 2, \ldots, r$.

Сортувальник заявок має повідомляти відправнику заявок про переповнення черги певного розміру шляхом обміну сигналами. Це необхідно робити з метою зміни пріоритетів та уникання подальших відмов через переповнення черги категорії. Окрім того, ми маємо пропускати великі задачі до системи, щоб не припинити їх обслуговування [10]. Про це має дбати модуль пріоритезації. За таких обставин наш алгоритм не гарантує співпадіння послідовності надходження заявок до системи з послідовністю надходження на виконання. Але ми зберігаємо цю послідовність в рамках кожної окремої категорії задач. Послідовність надходження результатів обслуговування при кількості каналів більше за один – не має обмежень щодо відповідності послідовності через асинхронну та випадкову природу процесу обслуговування.

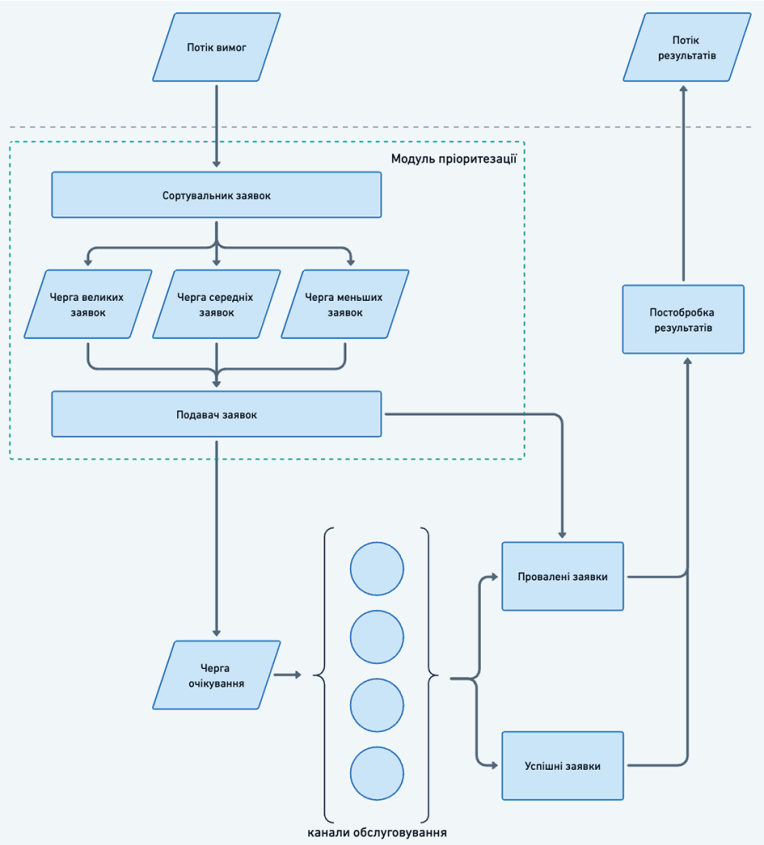

Рис. 2. Алгоритм обробки заявок з розбиттям

4. Результати дослідження

Розбиття загального вхідного потоку заявок на класи за їх розміром – є необхідною передумовою для подальших удосконалень моделі системи. Запропоновано декілька підходів до визначення складності вимоги, серед яких:

– самоідентифікація заявки;
– застосування функції ймовірнісного розподілу;
– застосування штучного інтелекту.

Було розглянуто переваги та недоліки для кожного з вище наведених рішень. Задля підвищення якості категоризації можливе застосування довільних комбінацій даних підходів.

Для оцінки результатів дослідження пропонується використовувати декілька чисельних метрик, які мають якнайкраще характеризувати наслідки застосування оптимізації. Головною метрикою буде загальна ймовірність відмови новоприбулій довільній заявці через переповнення черги очікування. Для глибшого розуміння природи оптимізації розраховуватимемо показники ймовірності відмови за умови повної пріоритезації по кожному з розмірів вимог. На рис. 3 зображено три графіки кожен з яких відповідає різним стратегіям:

– пріоритет великим задачам (пунктирна лінія);
– без застосування пріоритезації (цілісна лінія);
– пріоритет меншим задачам (штрих-пунктирна лінія).

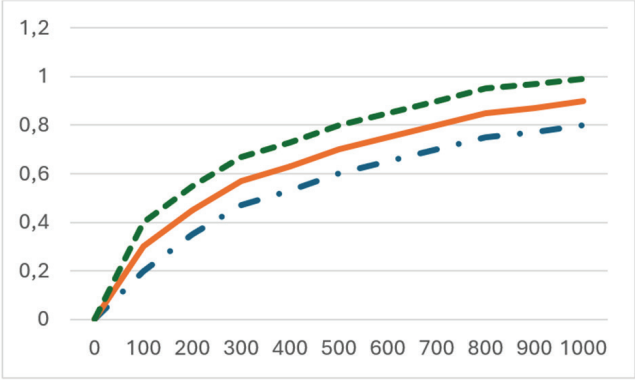

Рис. 3. Ймовірність відмови за пріоритетних категорій

Як бачимо, за умов коли інтенсивність надходження задач суттєво перевищує пропускну здатність системи, пріоритезація певної категорії заявок призводить до суттєвих змін пропускної здатності системи. Це пояснюється витісненням непріоритетних задач, а також збільшенням відсотку відмов для такої категорії. Відповідно, коли ми надаємо пріоритет меншим задачам, ми майже повністю перестаємо обслуговування великих задач, і навпаки.

Оскільки ми зацікавлені у збережені балансу розміру виконаних задач, щоб оптимізація не призвела до повного усунення всіх заявок, розподілених до класу більших, будемо розглядати балансні політики. Вони призводять до потрапляння в проміжок між лініями графіків пріоритету менших задач та без пріоритету [11]. Це дозволить збільшити загальну

пропускну здатність системи, надаючи при цьому можливість оброблятись для більших задач.

Наступним кроком буде розробка алгоритму, який даватиме стабільне потрапляння в зону балансу. Задля чисельного вимірювання успішності механізму пріоритезації, введемо метрику, яка буде відображати пропорційність добору задач. Вона представлятиме собою співвідношення відмов великим заявкам до загальної кількості відмов за проміжок часу t (11).

$$V(t) = \frac{V_L}{V_{zah}}, \quad (11)$$

де V_L – це відсоток відмов для великих задач, а V_{zah} – загальна кількість відмов у системі за проміжок часу t. Визначивши розмір проміжку часу, за який вестимуться спостереження, ми отримаємо періоди коригування балансу [8]. Після завершення збору аналітики по відмовах у системі, настає корекція коефіцієнту пріоритезації, який змінить певним чином пропорцію великих, середніх та маленьких заявок що перебувають у черзі.

Для апроксимації коефіцієнта пріоритезації пропонується застосовувати штучний інтелект. Модель якого попередньо має бути натренована на широкій вибірці даних. Важливо, щоб функція ймовірнісного розподілу вхідного потоку співпадала з функцією, що буде використовуватись при навчанні моделі штучного інтелекту. Із поглибленням інтеграції та довчанням моделі під час її застосування, будуть розширюватись можливості щодо виявлення певних патернів (послідовностей) поведінки в балансуванні навантаження [9]. Це дозволить передбачати наперед надмірне завантаження системи та завчасно адаптуватись за рахунок зміни пріоритетів. Дані переваги, у свою чергу, в повній мірі виправдовують додаткову складність, пов'язану з використанням штучного інтелекту для вирішення проблеми збільшення пропускної здатності СМО.

Висновки

Метод розбиття загального потоку заявок багатоканальної системи масового обслуговування з обмеженою чергою та відмовами на категорії за часом необхідним для обробки може бути дієвим способом зменшити загальну ймовірність відмови через переповнення черги. Такий підхід дозволяє більш ефективно задіяти вже наявні ресурси, за умови відсутності можливості збільшити кількість каналів обслуговування. Задля підвищення ефективності балансування пріоритетів, бажано на перших етапах використовувати особливості функції ймовірнісного розподілу інтенсивності надходження заявок до системи. В подальшому рекомендується застосування навченої моделі штучного інтелекту до оцінки складності задач, а також балансування пріоритетів між більшими та меншими вимогами. Такий метод збільшення продуктивності системи може бути інтегрований в архітектуру застосунку, призначеного для програмного моделювання СМО.

Було вперше запропоновано застосування оптимізації обробки заявок шляхом розбиття вхідного потоку вимог на класи згідно з розмірами та поєднання із використанням штучного інтелекту для оцінки складності заявок і подальшого балансування пріоритетів.

В подальшому дані підходи мають бути застосовані та реалізовані у програмному забезпечені, призначеного для моделювання СМО. Такий продукт, у свою чергу, надасть можливість напрацювати певні балансні стратегії та порівняти їх ефективність на практиці.

Список літератури

[1] Newell G. Applications of queuing theory. Second edition. – 1982. – Chapman and Hall – P. 1-302.

[2] Haenlein M., Kaplan A. A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence // California Management Review. – 2019. – №. 61(4). – P. 5-14.

[3] Ложковський А. Теорія масового обслуговування в телекомунікаціях. – 2010. – Одеса: ОНАЗ ім. О. С. Попова – 112 с.

[4] Литвинов А. Теорія систем масового обслуговування. – 2018. – Харків: ХНУМГ ім. О. М. Бекетова – 141 с.

[5] Alotaibi F., Ullah. I, Ahmad S. Modeling and Performance Evaluation of Multi-Class Queuing System with QoS and Priority Constraints // Electronics. – 2021. – №. 10(4). – P. 500.

[6] Beshley M., Kryvinska N., Beshley H., Yaremko O.; Pyrih J. Virtual Router Design and Modeling for Future Networks with QoS Guarantees // Electronics. – 2021. – № 10(10). – P. 1139.

[7] Vercellino C., Scionti A., Varavallo G., Viviani P., Vitali G., Terzo O. A Machine Learning Approach for an HPC Use Case: the Jobs Queuing Time Prediction // Future Generation Computer Systems. – 2023. – №. 143. – P. 215-230.

[8] Malik S., Gupta K., Gupta D., Singh A., Ibrahim M., Ortega-Mansilla A., Goyal N., Hamam H. Intelligent Load-Balancing Framework for Fog-Enabled Communication in Healthcare // Electronics. – 2022. – №. 11(4). – P. 566.

[9] Malik N., Sardaraz M., Tahir M., Shah B., Ali G., Moreira F. Energy-Efficient Load Balancing Algorithm for Workflow Scheduling in Cloud Data Centers Using Queuing and Thresholds // Applied Sciences. – 2021. – №. 11(13). – P. 5849.

[10] Apachidi X., Katsman Yu. Development of a Queuing System with Dynamic Priorities // Key Engineering Materials. – 2016. – №. 685. – P. 934-938.

[11] Chikriy A., Gubarev V., Kondratenko Y., Turovyerova N. Multi-channel Queuing Systems with the Dynamic Priority // Journal of Automation and Information Sciences. – 2009. – №. 41(8). – P. 49-54.

Надійшла до редколегії 14.05.2024