

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

МІХАЛІНА МИРОСЛАВ ОЛЕКСАНДРОВИЧ

Допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій
канд. техн. наук, доцент
_____ О. В. Зелінська
« ____ » _____ 20__ р.

**ФУНКЦІОНУВАННЯ ВЕБ-САЙТУ НА ПІДСТАВІ СИСТЕМ
МАСОВОГО ОБСЛУГОВУВАННЯ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (магістерська) робота

Науковий керівник:
Н. Р. Веселовська, професор
кафедри інформаційних технологій,
д. т. н., професор

Оцінка: ____ / ____ / ____
(бали/за шкалою ЄКТС/за
національною шкалою)

Голова ЕК: _____

Вінниця 2024

АНОТАЦІЯ

Міхаліна М.О. Функціонування веб-сайту на підставі систем масового обслуговування. Спеціальність 122 “Комп’ютерні науки”, Освітня програма “Data science”. Донецький національний університет імені Василя Стуса, Вінниця, 2024.

У кваліфікаційній роботі досліджено математичні моделі систем масового обслуговування для аналізу та оптимізації процесу обробки веб-запитів. Основна увага приділена класичній моделі M/M/1, яка описує взаємодію між потоками запитів і серверними ресурсами. У роботі представлено програму, що реалізує моделювання цієї системи, яка дозволяє аналізувати ключові параметри моделі, такі як інтенсивність запитів (λ) та швидкість обробки (μ).

На підставі дослідження показано, що оптимізація цих параметрів дозволяє мінімізувати час очікування користувачів і покращити продуктивність серверів. Встановлено, що програма може бути ефективно використана для аналізу поведінки системи у реальних умовах і прогнозування навантаження під час пікових періодів. Запропоновано рекомендації щодо покращення роботи веб-сайтів на основі отриманих даних.

Ключові слова: функціонування веб-сайту, модель M/M/1, теорія масового обслуговування, оптимізація серверів, час очікування, пікове навантаження.

ABSTRACT

Mikhalina M.O. Functioning of the website on the basis of queuing systems. Specialty 122 “Computer Science”, Educational program “Data science”. Vasyl' Stus Donetsk National University, Vinnytsia, 2024.

The qualification work investigates the mathematical models of queuing systems for analyzing and optimizing the process of processing web requests. The main attention is paid to the classical M/M/1 model, which describes the interaction between request flows and server resources. The paper presents a program that implements the modeling of this system, which allows analyzing key model parameters such as request intensity (λ) and processing speed (μ).

The study shows that optimizing these parameters minimizes user waiting time and improves server performance. It is established that the program can be effectively used to analyze the behavior of the system in real conditions and predict the load during peak periods. Recommendations for improving website performance based on the obtained data are proposed.

Keywords: website operation, M/M/1 model, queuing theory, server optimization, waiting time, peak load.

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1	11
1.1 Сутність і призначення систем масового обслуговування	11
1.1.1 Основні компоненти СМО	11
1.1.2 Історія розвитку теорії масового обслуговування.....	12
1.1.3 Використання СМО у сучасних технологіях	12
1.1.4 Задачі, які вирішує СМО.....	12
1.2 Класифікація моделей систем масового обслуговування	13
1.2.1 Основні параметри систем масового обслуговування.....	13
1.2.2 Типи моделей СМО	14
1.2.3 Відмінності між детермінованими та стохастичними моделями	15
1.2.4 Приклади систем масового обслуговування	16
1.3 Характеристика моделі М/М/1	18
1.3.1 Основні припущення моделі М/М/1	18
1.3.2 Математичні характеристики моделі.....	19
1.3.3 Переваги моделі М/М/1	20
1.4 Особливості функціонування інформаційних систем на основі систем масового обслуговування	21
1.4.1 Динаміка потоків заявок.....	21
1.4.2 Особливості роботи серверів	21
1.4.3 Поведінка систем під час пікових навантажень	22
1.4.4 Обмеження та виклики	22
1.5 Класифікація потоків заявок у теорії масового обслуговування.....	23

1.5.1 Основні типи потоків заявок.....	23
1.5.2 Характеристики потоків.....	24
1.5.3 Вплив типу потоку на роботу системи.....	24
1.5.4 Порівняння потоків.....	25
1.5.5 Практичне застосування.....	25
1.6 Історія розвитку теорії масового обслуговування	26
1.6.1 Перші кроки: теорія ймовірностей та задачі зв'язку	26
1.6.2 Розширення теорії у XX столітті.....	27
1.6.3 Сучасний розвиток теорії.....	27
1.6.4 Внесок науковців	28
1.6.5 Практичне значення теорії масового обслуговування.....	28
1.7 Особливості потоків заявок у веб-системах	29
1.7.1 Ключові характеристики потоків заявок.....	29
1.7.2 Типи потоків заявок у веб-системах	30
1.7.3 Вплив потоків заявок на СМО.....	31
1.7.4 Адаптація моделей СМО до веб-систем.....	31
1.7.5 Приклади потоків заявок у реальних веб-системах	32
Висновки до розділу 1.....	33
2.1 Аналіз вихідних даних та постановка задачі.....	35
2.1.1 Аналіз вихідних даних.....	35
2.1.2 Постановка задачі	36
2.2 Розробка математичної моделі	37
2.2.1 Формалізація задачі	37
2.2.2 Критерії оптимізації.....	39
2.2.3 Практичне використання моделі	39

2.3 Опис алгоритму вирішення задачі	39
2.3.1 Етапи розробки алгоритму	40
2.3.2 Опис алгоритму	41
2.3.3 Блок-схема алгоритму	41
2.3.4 Програмна реалізація	43
2.3.5 Практичне застосування алгоритмів	43
2.3.6 Шляхи покращення алгоритмів	46
2.4 Інтеграція моделі М/М/1 у сучасні інформаційні системи	49
2.4.1 Інформаційні системи, які використовують М/М/1	49
2.4.2 Інструменти для інтеграції моделі	50
2.4.3 Етапи впровадження М/М/1	51
2.4.4 Обмеження інтеграції	51
2.4.4 Приклади успішної інтеграції	52
2.5 Порівняння моделі М/М/1 із іншими моделями систем масового обслуговування	52
2.5.1 Характеристики моделі М/М/1	52
2.5.2 Модель М/М/с	53
2.5.3 Переваги перед М/М/1:	53
2.5.4 Модель М/Д/1	54
2.5.5 Порівняльна таблиця	54
2.5.6 Застосовність у веб-системах	55
Висновки до розділу 2	55
РОЗДІЛ 3	57
3.1 Опис програмного забезпечення	57
3.1.1 Архітектура програмного інструменту	57

3.1.2 Використані технології та інструменти.....	58
3.1.3 Функціональні можливості програми.....	58
3.1.4 Переваги програми.....	59
3.2 Реалізація алгоритму.....	59
3.2.1 Основні компоненти алгоритму	59
3.3 Аналіз продуктивності веб-сайту.....	62
3.3.1 Введення даних	62
3.3.2 Результати розрахунків	63
3.3.3 Візуалізація результатів	63
3.3.4 Аналіз результатів.....	64
3.4 Оптимізація роботи веб-сайту	64
3.4.1 Визначення проблемних зон.....	64
3.4.2 Рекомендації з оптимізації	65
3.5 Експорт результатів та їх обговорення	67
3.5.1 Експорт результатів	67
3.5.2 Обговорення отриманих результатів	68
3.5.3 Переваги збереження результатів	69
3.6 Оцінка точності моделі	69
3.6.1 Методика оцінки точності.....	69
3.6.2 Фактори, що впливають на точність	70
3.6.3 Рекомендації для підвищення точності	71
3.7 Розширення функціональності програми.....	71
3.7.1 Підтримка багатоканальних систем (М/М/с).....	72
3.7.2 Реалізація прогнозування пікових навантажень.....	72
3.7.3 Автоматизація збору даних.....	72

3.7.4 Розширена візуалізація	73
3.7.5 Підтримка складених потоків заявок (M/G/1, G/G/1)	73
3.7.6 Розширення функції експорту	73
Висновки до розділу	74
ВИСНОВКИ	76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	78

ВСТУП

У сучасному світі інформаційних технологій веб-сайти є ключовими інструментами для комунікації між бізнесом і споживачами, забезпечуючи доступ до товарів, послуг і інформації. В умовах постійного зростання обсягу веб-трафіку та потреби в безперебійній роботі веб-сайтів особливого значення набувають питання оптимізації їхньої продуктивності. Однією з ефективних методик аналізу та вдосконалення роботи веб-сайтів є використання моделей систем масового обслуговування.

У цій роботі розглядається використання моделі $M/M/1$ для аналізу продуктивності веб-сайтів, визначення впливу параметрів навантаження та обслуговування на ефективність їх роботи, а також розробка рекомендацій щодо оптимізації серверної інфраструктури.

Об'єктом дослідження є процес функціонування веб-сайту в умовах змінного навантаження.

Предметом дослідження є модель $M/M/1$ як інструмент для аналізу та оптимізації роботи серверів веб-сайтів.

Метою дослідження є розробка методики аналізу функціонування веб-сайтів на основі моделі $M/M/1$, що дозволяє мінімізувати час очікування користувачів і підвищити продуктивність серверної інфраструктури.

Наукова новизна дослідження:

1. Розроблено програмний інструмент для аналізу роботи серверів веб-сайтів на основі моделі $M/M/1$.
2. Удосконалено підходи до оцінювання продуктивності серверів, що враховують змінні параметри навантаження та обслуговування.
3. Запропоновано рекомендації щодо оптимізації роботи веб-сайтів під час пікових навантажень.

Практичне значення отриманих результатів:

1. Розроблений програмний інструмент може бути використаний для аналізу продуктивності серверів веб-сайтів у реальних умовах.

2. Результати дослідження дозволяють прогнозувати поведінку систем під час пікових навантажень та ефективно управляти серверними ресурсами.

3. Запропоновані рекомендації можуть бути застосовані для забезпечення стабільної роботи веб-сайтів у сферах електронної комерції, фінансових послуг, медіа тощо.

Апробація результатів дослідження:

Основні положення дослідження та отримані результати були представлені під час практичної реалізації програмного інструменту для аналізу роботи серверів веб-сайтів. Програма була протестована на прикладі інтернет-магазину та отримала позитивну оцінку.

Структура роботи:

Кваліфікаційна робота складається з анотації, вступу, трьох розділів, висновків, списку використаних джерел.

- У першому розділі розглянуто теоретичні основи систем масового обслуговування, зокрема моделі М/М/1.
- У другому розділі описано розробку програмного інструменту для аналізу продуктивності веб-сайтів.
- У третьому розділі представлено результати практичного дослідження та рекомендації щодо оптимізації роботи серверів.
- Загальний обсяг роботи становить 45-50 сторінок.

РОЗДІЛ 1

ТЕОРИТИЧНІ ОСНОВИ СИСТЕМ МАСОВОГО ОБСЛУГОВУВАННЯ

1.1 Сутність і призначення систем масового обслуговування

Системи масового обслуговування (СМО) — це математичні моделі, які описують процеси, пов'язані з прийомом, обробкою та обслуговуванням заявок у різних системах. Основна мета використання СМО полягає в оптимізації ресурсів системи, зменшенні часу очікування користувачів та забезпеченні стабільної роботи навіть у періоди пікових навантажень.

1.1.1 Основні компоненти СМО



Рис. 1.1. Основні компоненти СМО

Система масового обслуговування складається з таких ключових елементів:

1. **Потік заявок** — це користувачі або події, які надходять у систему для обслуговування. Потік може бути випадковим (наприклад, виклики в кол-центр) або детермінованим (наприклад, заплановані завдання в системах автоматизації).
2. **Черга** — це місце тимчасового зберігання заявок, які очікують на обслуговування. Черга утворюється, якщо всі ресурси системи зайняті, і може суттєво впливати на задоволеність користувачів.
3. **Система обслуговування** — це ресурси, які виконують обслуговування заявок (наприклад, оператори кол-центру або сервери веб-сайту).

1.1.2 Історія розвитку теорії масового обслуговування

Перші ідеї, пов'язані з аналізом масового обслуговування, були запропоновані данським математиком Агнером Краррупом Ерлангом у 1909 році. Його дослідження стосувалися телефонних мереж і були спрямовані на оцінку ймовірності блокування викликів. Пізніше теорія отримала розвиток завдяки роботам Д. Кендалла, який запровадив стандартну нотацію для класифікації СМО, відому як нотація Кендалла.

1.1.3 Використання СМО у сучасних технологіях

Зі зростанням обсягів даних і складності інформаційних систем теорія масового обслуговування стала невід'ємною частиною аналізу продуктивності різних систем. СМО застосовуються для вирішення широкого спектра задач:

- **В аналізі інформаційних систем:** Визначення продуктивності серверів, оптимізація баз даних, аналіз трафіку в комп'ютерних мережах.
- **В електронній комерції:** Оптимізація роботи веб-сайтів, мінімізація часу завантаження сторінок і часу очікування користувачів.
- **У логістиці:** Планування роботи складів і транспортування товарів.
- **У сфері обслуговування:** Оптимізація роботи кол-центрів, організація роботи черг у банках чи аеропортах.

1.1.4 Задачі, які вирішує СМО

Системи масового обслуговування дозволяють моделювати реальні процеси, аналізувати роботу системи за різних умов і оптимізувати її функціонування. Зокрема, вони допомагають:

1. Прогнозувати середній час очікування клієнтів.
2. Аналізувати ефективність роботи системи при змінних потоках запитів.
3. Визначати вплив ресурсоемності на якість обслуговування.

1.2 Класифікація моделей систем масового обслуговування

Системи масового обслуговування (СМО) мають різноманітні характеристики, які дозволяють адаптувати їх до реальних потреб конкретної задачі. Для кращого розуміння та аналізу вони класифікуються за основними параметрами, що визначають їх роботу: тип потоку заявок, кількість каналів обслуговування, правила обробки запитів і дисципліна обслуговування.

1.2.1 Основні параметри систем масового обслуговування

1. Інтенсивність потоку заявок (λ):

Характеризує середню кількість заявок, які надходять до системи за одиницю часу. Потік може бути:

- **Випадковим (стохастичним):** Надходження заявок описується випадковим процесом (наприклад, потік запитів на веб-сайт протягом дня).
- **Детермінованим:** Чітко запланований час надходження заявок (наприклад, автоматизовані завдання).

2. Швидкість обслуговування (μ):

Показує середню кількість заявок, які система може обробити за одиницю часу. Швидкість обслуговування залежить від характеристик серверів, обладнання чи персоналу.

3. Кількість каналів обслуговування (c):

Системи можуть мати один або декілька каналів для обробки заявок.

Наприклад:

- Одноканальна система (М/М/1) — один сервер.
- Багатоканальна система (М/М/с) — кілька серверів.

4. Дисципліна обслуговування:

Визначає правила, за якими заявки обробляються:

- **FIFO (First In, First Out):** Перший прийшов — перший обслуговується.
- **LIFO (Last In, First Out):** Останній прийшов — перший обслуговується.
- **Пріоритети:** Заявки з високим пріоритетом обробляються швидше.

5. Ємність системи:

Максимальна кількість заявок, яку система може обробити одночасно.

Якщо заявок більше, вони можуть бути відхилені або чекати у черзі.

1.2.2 Типи моделей СМО

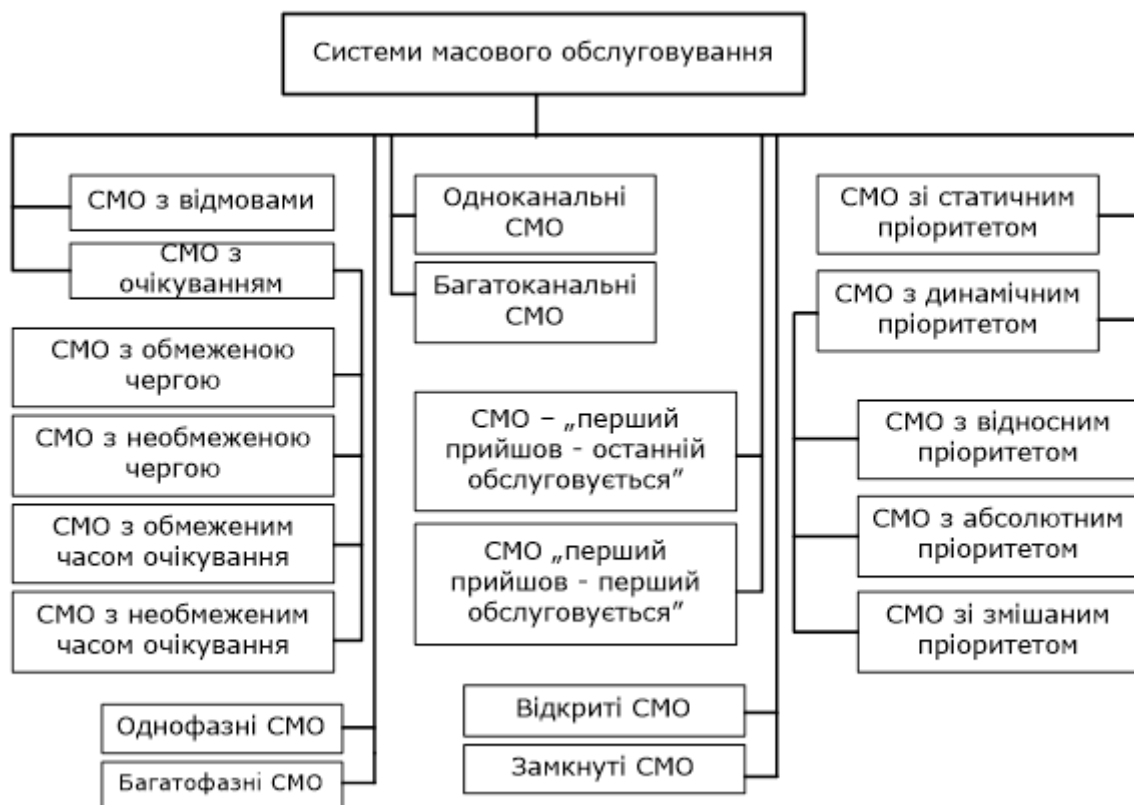


Рис. 1.2. Типи моделей СМО

СМО класифікуються за типом потоку заявок і дисципліною обслуговування.

Найбільш поширені моделі:

1. M/M/1:

- Одноканальна система з експоненціальним розподілом часу між заявками та часу обслуговування.
- Використовується для простих систем, наприклад, одного сервера веб-сайту.

2. M/M/c:

- Багатоканальна система (с — кількість каналів).
- Застосовується для великих серверних ферм або систем із паралельними ресурсами.

3. M/D/1:

- Одноканальна система з детермінованим часом обслуговування.
- Часто використовується для аналізу задач з постійними тимчасовими витратами.

4. G/G/1:

- Система з довільним розподілом часу між заявками та обслуговуванням.
- Використовується для складних випадків, коли розподіли не можна визначити наперед.

1.2.3 Відмінності між детермінованими та стохастичними моделями

1. Стохастичні моделі:

- Описують системи, у яких час надходження заявок і їхнє обслуговування є випадковими величинами.
- Використовуються для веб-сайтів із непередбачуваним потоком запитів.

2. Детерміновані моделі:

- Передбачають точні інтервали між заявками та сталий час обслуговування.
- Застосовуються для систем, які виконують заплановані завдання

1.2.4 Приклади систем масового обслуговування

1. Веб-сервери та інтернет-сайти (М/М/1 або М/М/с)

- Потік заявок: запити користувачів до серверу (перегляд сторінок, запити на завантаження файлів).
- Обслуговування: сервери веб-сайту обробляють ці запити.
- Приклад: Інтернет-магазин обслуговує запити покупців на перегляд товарів і оформлення замовлень.
 - Якщо сервер один — це модель М/М/1.
 - Якщо серверів декілька — М/М/с.

2. Кол-центри (М/М/с)

- Потік заявок: телефонні дзвінки клієнтів.
- Обслуговування: оператори, які відповідають на дзвінки.
- Приклад: Банківський кол-центр, де клієнти звертаються за консультаціями чи послугами.

3. Логістика та склади (М/М/1 або М/Д/1)

- Потік заявок: надходження товарів на склад або запити на відвантаження.
- Обслуговування: процес завантаження чи розвантаження товарів.
- Приклад: Автомобілі чекають своєї черги для розвантаження на складі, де працює один кран (М/М/1) або кілька (М/М/с).

4. Охорона здоров'я (М/М/с)

- Потік заявок: пацієнти, які приходять на прийом.

- Обслуговування: лікарі або обладнання, яке проводить обстеження чи лікування.
- Приклад: У поліклініці пацієнти очікують на консультацію до одного чи кількох лікарів (М/М/1 або М/М/с залежно від кількості лікарів).

5. Аеропорти (М/D/1 або М/М/с)

- Потік заявок: літаки, що прибувають на посадку або вильоти.
- Обслуговування: злітно-посадкові смуги.
- Приклад: Аеропорт із двома смугами для посадки літаків (М/М/2), якщо обслуговування має постійну тривалість — М/D/1.

6. Каси супермаркетів (М/М/с)

- Потік заявок: клієнти, які стають у чергу для оплати.
- Обслуговування: касири, що обслуговують покупців.
- Приклад: У супермаркеті працює кілька кас. Кожна з них є окремим каналом обслуговування (наприклад, М/М/5 для 5 кас).

7. Телекомунікаційні мережі (М/М/с або G/G/с)

- Потік заявок: телефонні виклики, передача даних, запити на з'єднання.
- Обслуговування: маршрутизатори, комутатори або базові станції, які обробляють запити.
- Приклад: Стільниковий оператор обслуговує дзвінки абонентів. Кількість базових станцій визначає кількість каналів (наприклад, М/М/20 для 20 станцій)

8. Громадський транспорт (М/М/с)

- Потік заявок: пасажери, які чекають транспорт на зупинці.
- Обслуговування: автобуси або поїзди, що прибувають на зупинку.
- Приклад: Станція метро, де пасажери очікують поїзд. Один поїзд — це М/М/1, кілька поїздів — М/М/с.

9. Обчислювальні системи (M/G/1 або G/G/c)

- Потік заявок: завдання, які надходять до процесора.
- Обслуговування: процесор обробляє ці завдання.
- Приклад: Сервер, що обробляє одночасно кілька завдань із різним пріоритетом (G/G/1 для змінного часу обробки).

10. Банківські послуги (M/M/c)

- Потік заявок: клієнти, які прийшли у відділення банку.
- Обслуговування: співробітники банку, які обслуговують клієнтів.
- Приклад: Банк із трьома касами для прийому клієнтів (M/M/3).

1.3 Характеристика моделі M/M/1

Модель M/M/1 є базовою одноканальною системою масового обслуговування, яка використовується для аналізу та моделювання різноманітних процесів, пов'язаних із чергами. Завдяки своїй математичній простоті і гнучкості, модель M/M/1 широко застосовується у сфері інформаційних технологій для оцінювання продуктивності серверів, аналізу поведінки клієнтів та прогнозування можливих затримок.

1.3.1 Основні припущення моделі M/M/1

Модель M/M/1 базується на таких припущеннях:

1. **Експоненціальний розподіл міжчасу надходження заявок (M):** Потік заявок є випадковим і описується пуассонівським процесом, де середній час між заявками обернено пропорційний інтенсивності запитів (λ).
2. **Експоненціальний розподіл часу обслуговування (M):** Час, необхідний для обробки заявки, є випадковою величиною з експоненціальним розподілом, а його середнє значення визначається інтенсивністю обслуговування (μ).
3. **Один канал обслуговування :** У системі працює лише один сервер, який може обробляти одну заявку одночасно.

4. **Необмежена черга:** Заявки, які не можуть бути оброблені негайно, залишаються у черзі.
5. **Дисципліна обслуговування:** FIFO (First In, First Out) — заявки обслуговуються в порядку їх надходження.

1.3.2 Математичні характеристики моделі

Модель М/М/1 дозволяє обчислити основні показники ефективності системи:

1. **Завантаження системи (ρ):**

$$\rho = \frac{\lambda}{\mu}$$

ρ показує частку часу, коли сервер зайнятий. Якщо ρ близьке до 1, це означає, що сервер перевантажений, що призводить до збільшення часу очікування.

2. **Середній час очікування в черзі (W_q):**

$$W_q = \frac{\lambda}{\mu \cdot (\mu - \lambda)}$$

Цей показник описує, скільки часу в середньому заявка очікує обслуговування.

3. **Середній час перебування в системі (W):**

$$W = \frac{1}{\mu - \lambda}$$

Включає час очікування в черзі та час обслуговування.

4. **Середня довжина черги (L):**

$$L_q = \frac{\lambda^2}{\mu \cdot (\mu - \lambda)}$$

5. **Середня кількість заявок у системі (L):**

$$L = \frac{\lambda}{\mu - \lambda}$$

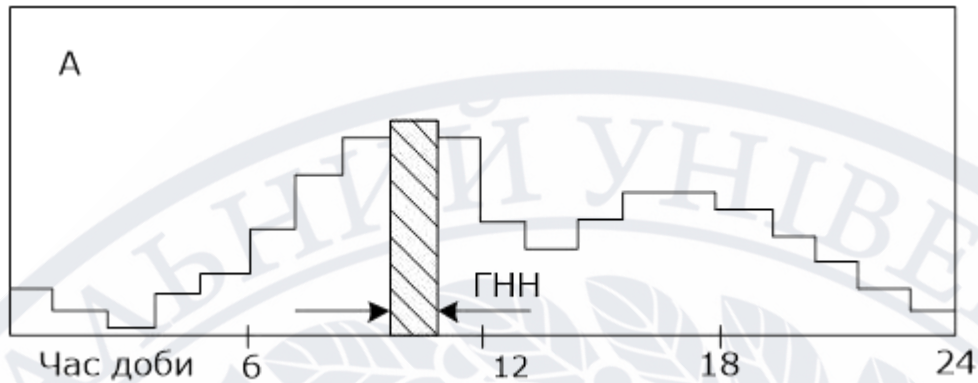


Рис. 1.3. Інтенсивність навантаження за добу

1.3.3 Переваги моделі М/М/1

1. **Простота математичного опису:** Модель М/М/1 дозволяє швидко розраховувати ключові показники, необхідні для аналізу продуктивності системи.
2. **Гнучкість:** Вона підходить для моделювання широкого спектра систем, таких як веб-сервери, кол-центри, банківські каси.
3. **Ефективність у базовому аналізі:** Модель забезпечує якісний аналіз за умови, що припущення відповідають реальності.

1.3.4 Недоліки моделі М/М/1

1. **Обмеження у реальних умовах:** Модель передбачає експоненціальні розподіли, які не завжди відповідають реальним потокам заявок або часу обслуговування.
2. **Недооцінка складних сценаріїв:** М/М/1 не враховує такі аспекти, як змінний потік запитів, пріоритети заявок або кілька серверів.
3. **Необмежена черга:** У реальних системах черга часто має обмежену довжину.

1.4 Особливості функціонування інформаційних систем на основі систем масового обслуговування

Системи масового обслуговування (СМО) активно використовуються для моделювання та оптимізації роботи сучасних інформаційних систем. Ці моделі дозволяють аналізувати взаємодію між потоками заявок і серверними ресурсами, що особливо актуально для веб-сайтів, мобільних додатків і хмарних сервісів. У цьому підрозділі розглядаються ключові особливості функціонування інформаційних систем, що базуються на принципах СМО.

1.4.1 Динаміка потоків заявок

Інтенсивність потоків заявок (λ) в інформаційних системах може бути як стабільною, так і змінною. Наприклад, веб-сайти електронної комерції зазвичай спостерігають пікові навантаження під час акцій або святкових розпродажів. У таких умовах моделі М/М/1 дозволяють визначати:

- Час очікування користувачів у черзі (W_q).
- Завантаження серверів (ρ).
- Довжину черги (L_q).

Такі дані важливі для передбачення можливих затримок і перевантаження системи.

1.4.2 Особливості роботи серверів

Інформаційні системи використовують різні архітектури обслуговування:

- Одноканальні системи: Відповідають моделі М/М/1. Їх застосовують для невеликих веб-сайтів із помірним навантаженням.
- Багатоканальні системи: Відповідають моделі М/М/с, де с — кількість серверів. Вони характерні для великих інтернет-магазинів або хмарних платформ.

Використання СМО дозволяє оцінювати ефективність роботи серверів, знаходити оптимальну кількість обчислювальних ресурсів і розподіляти навантаження.

1.4.3 Поведінка систем під час пікових навантажень

Моделі СМО допомагають прогнозувати поведінку систем під час пікових навантажень. Наприклад:

- Якщо λ наближається до μ , збільшується ймовірність перевантаження сервера.
- При надмірному навантаженні ($\rho > 0.8$) необхідно застосовувати оптимізаційні заходи, як-от збільшення кількості серверів, впровадження кешування або використання CDN.

1.4.4 Обмеження та виклики

Попри свою ефективність, моделі СМО мають певні обмеження:

1. Стабільність вхідних потоків: Моделі, такі як M/M/1, припускають постійну інтенсивність заявок, що не завжди відповідає реальності.
2. Складність інтеграції: Для точного моделювання необхідно враховувати специфіку апаратного та програмного забезпечення.
3. Обробка великих даних: Інформаційні системи можуть стикатися з проблемами обробки великих обсягів даних, які складно передбачити за допомогою класичних моделей.

Системи масового обслуговування є потужним інструментом для моделювання та оптимізації інформаційних систем. Вони дозволяють передбачати поведінку системи під час навантажень, розраховувати ключові показники продуктивності та приймати обґрунтовані рішення щодо покращення роботи веб-сайтів та серверів. У подальшому розробка гібридних

моделей, які враховують динамічні потоки заявок, може значно підвищити точність прогнозів і ефективність роботи інформаційних систем.

1.5 Класифікація потоків заявок у теорії масового обслуговування

Потоки заявок є ключовим елементом у теорії масового обслуговування, оскільки вони визначають динаміку взаємодії користувачів із системою. Тип потоку значною мірою впливає на поведінку системи, її продуктивність і стабільність. У цьому підрозділі розглянуто основні типи потоків заявок, їх характеристики та вплив на роботу систем масового обслуговування.

1.5.1 Основні типи потоків заявок

1. Пуассонівський потік (потік випадкових подій):

- Пуассонівський потік характеризується тим, що часи між надходженням заявок розподілені експоненціально.
- Основна властивість: рівномірна ймовірність надходження заявки в будь-який момент часу.
- Використовується в моделі $M/M/1$ і є основою для аналізу більш складних систем.

2. Детермінований потік:

- У цьому потоці інтервали між заявками фіксовані й не змінюються з часом.
- Використовується для моделювання систем, у яких надходження запитів суворо регламентоване.
- Приклад: автоматичні оновлення баз даних або періодичні перевірки систем.

3. Марковські процеси:

- Характеризуються залежністю між подіями у потоці. Відстань між подіями може змінюватися залежно від попередніх станів.
- Використовуються у складних системах із прогнозуванням поведінки користувачів.

4. Складений потік:

- Поєднує кілька типів потоків, наприклад, випадковий і детермінований.
- Використовується для моделювання складних систем, де є різні типи взаємодій.

1.5.2 Характеристики потоків

- **Інтенсивність потоку (λ lambda):** Визначає середню кількість заявок, які надходять до системи за одиницю часу. Для пуассонівського потоку це основний параметр.
- **Рівномірність:** Визначає, наскільки рівномірно заявки розподілені в часі. Пуассонівський потік має високу випадковість, а детермінований — максимальну рівномірність.
- **Кореляція:** Для деяких потоків часи між заявками можуть бути залежними, що впливає на роботу системи.

1.5.3 Вплив типу потоку на роботу системи

1. Пуассонівський потік:

- Забезпечує універсальність для аналізу.
- Добре підходить для систем із високою випадковістю, як-от обробка веб-запитів.

2. Детермінований потік:

- Дає змогу досягти стабільності, але може не відображати реальні умови у веб-середовищі.

3. Складені потоки:

- Відображають реальні умови роботи складних інформаційних систем, але їх аналіз є складнішим.

1.5.4 Порівняння потоків

Тип потоку	Переваги	Недоліки
Пуассонівський	Простота моделювання, універсальність	Може не враховувати залежності між заявками
Детермінований	Стабільність, передбачуваність	Не підходить для реальних умов
Складений	Гнучкість, точність	Висока складність моделювання

1.5.5 Практичне застосування

Для веб-сайтів потік заявок зазвичай описується пуассонівським розподілом через його простоту й універсальність. Проте для великих систем із піковими навантаженнями або змінними потоками доцільно використовувати складені моделі.

Розуміння характеристик і типів потоків заявок є важливим для правильного вибору моделі системи масового обслуговування. Пуассонівський потік залишається стандартом для багатьох досліджень, але розвиток складених потоків відкриває нові можливості для моделювання складних інформаційних систем.

1.6 Історія розвитку теорії масового обслуговування

Теорія масового обслуговування (СМО) є однією з ключових галузей прикладної математики, яка сформувалася для аналізу систем, що обслуговують потоки заявок. Її розвиток пов'язаний із необхідністю вирішувати практичні задачі у сфері телекомунікацій, логістики, транспорту та комп'ютерних систем. У цьому підрозділі розглядається історія виникнення та становлення теорії, а також внесок основоположників у її розвиток.

1.6.1 Перші кроки: теорія ймовірностей та задачі зв'язку

Теорія масового обслуговування почала формуватися наприкінці XIX століття як відповідь на проблеми, що виникали у телекомунікаційних системах. Одним із перших науковців, які почали вивчати питання обслуговування потоків заявок, був данський інженер і математик Агнер Краруп Ерланг. У 1909 році він опублікував перші роботи, в яких математично моделював процеси телефонного зв'язку.

Ерланг вивчав питання:

- Скільки телефонних ліній необхідно для забезпечення якісного обслуговування?
- Як обчислити ймовірність відмови в обслуговуванні через зайнятість усіх ліній?

Його внесок привів до створення перших математичних моделей, які нині відомі як **формули Ерланга**:

- **Формула Ерланга В** для обчислення ймовірності відмови.
- **Формула Ерланга С** для визначення ймовірності очікування в черзі.

Ці моделі стали основою для аналізу систем масового обслуговування.

1.6.2 Розширення теорії у XX столітті

У 50-х роках XX століття розвиток теорії СМО значно прискорився. Британський математик Д. Г. Кендалл запропонував систематизацію моделей масового обслуговування. У 1953 році він представив класифікацію, яка використовується й донині. Формат **A/B/C** визначає:

- **A** — розподіл часу між заявками (наприклад, **M** — пуассонівський).
- **B** — розподіл часу обслуговування (наприклад, **M** — експоненціальний, **D** — детермінований).
- **C** — кількість каналів обслуговування.

Кендалл також розробив формули для багатоканальних систем (**M/M/c**), що розширило можливості аналізу великих систем із паралельним обслуговуванням заявок.

1.6.3 Сучасний розвиток теорії

Зі зростанням складності систем у 70-х і 80-х роках теорія масового обслуговування почала застосовуватися до нових галузей, таких як комп'ютерні мережі, транспортна логістика та управління запасами.

1. Комп'ютерні системи:

- Впровадження теорії СМО в аналіз серверів, хмарних обчислень і веб-сайтів.
- Модель **M/M/1** почала широко використовуватись для оцінки продуктивності серверів.

2. Системи зв'язку:

- Розробка моделей для аналізу інтернет-трафіку, розподілу ресурсів у мобільних мережах.

3. Інтелектуальні системи:

- Використання складених моделей (наприклад, G/G/1) для аналізу змінних потоків у реальному часі.

1.6.4 Внесок науковців

1. Агнер Ерланг:

- Засновник теорії масового обслуговування.
- Автор перших формул для розрахунку продуктивності телефонних систем.

2. Джон Кендалл:

- Систематизатор моделей СМО, автор класифікації моделей.
- Розробив моделі для багатоканальних систем.

3. Леонард Кляйнрок:

- Вніс вклад у застосування СМО в комп'ютерних мережах.
- Один із перших, хто використав моделі СМО для аналізу інтернет-трафіку.

1.6.5 Практичне значення теорії масового обслуговування

Теорія масового обслуговування стала універсальним інструментом для моделювання роботи різних систем. Її прикладне значення включає:

- **Оптимізацію серверів веб-сайтів:** аналіз продуктивності, оцінка часу очікування.
- **Транспортна логістика:** моделювання роботи транспортних вузлів, аналіз черг у розподільчих центрах.

- **Мобільні мережі:** розподіл частот і ресурсів для забезпечення якості обслуговування.

Історія розвитку теорії масового обслуговування демонструє її значення для розв'язання практичних задач у багатьох галузях. Від перших формул Ерланга до сучасних застосувань у веб-аналізі, СМО постійно розвивається, адаптуючись до нових викликів сучасності. Завдяки цьому теорія залишається актуальним і потужним інструментом для досліджень та оптимізації систем.

1.7 Особливості потоків заявок у веб-системах

Потоки заявок у веб-системах характеризуються значною нерівномірністю, високою динамічністю та залежністю від поведінки користувачів. Ці особливості значно впливають на проектування і продуктивність систем, які їх обслуговують. У цьому підрозділі розглядаються ключові характеристики потоків заявок у веб-середовищі, їх класифікація та вплив на функціонування систем масового обслуговування (СМО).

1.7.1 Ключові характеристики потоків заявок

1. Нерівномірність трафіку

- Веб-системи часто стикаються з піковими навантаженнями, які залежать від часу доби, регіону чи проведення акцій. Наприклад, трафік інтернет-магазинів значно зростає під час сезонних розпродажів.
- Така поведінка ускладнює моделювання, адже вхідний потік заявок (λ) може суттєво змінюватися за короткі проміжки часу.

2. Динамічність

- Веб-запити відрізняються не тільки нерівномірністю, але й швидкими змінами. Це вимагає використання адаптивних моделей, здатних обробляти потоки в реальному часі.

3. Поведінка користувачів

- Унікальні патерни поведінки, як-от кількість кліків, тривалість сесії, впливають на загальний потік запитів. Наприклад, користувачі можуть масово відкривати сторінки одночасно, що створює високі навантаження.

4. Різноманітність запитів

- У веб-системах обробляються запити різних типів, таких як завантаження сторінок, запити до баз даних, API-виклики тощо. Кожен тип має власну тривалість обслуговування (μ).

1.7.2 Типи потоків заявок у веб-системах

1. Постійні потоки

- Заявки надходять із відносно стабільною інтенсивністю протягом тривалого періоду. Наприклад, це характерно для корпоративних веб-сайтів із фіксованою аудиторією.

2. Періодичні потоки

- Інтенсивність заявок змінюється залежно від часу доби чи інших факторів. Наприклад, сайти новин спостерігають збільшення трафіку зранку та ввечері.

3. Сплески активності

- Короткотривалі пікові навантаження, які виникають через одночасну активність багатьох користувачів. Наприклад, це може бути реєстрація на подію чи випуск популярного продукту.

4. Випадкові потоки

- Потоки із непередбачуваною динамікою, наприклад, трафік на соціальних платформах, де поведінка користувачів варіюється залежно від трендів.

1.7.3 Вплив потоків заявок на СМО

1. Підвищене завантаження

- Якщо інтенсивність потоку (λ) наближається до швидкості обслуговування (μ), система може стати перевантаженою, що призведе до збільшення черг і часу очікування.

2. Зниження продуктивності

- У разі надмірних навантажень сервери можуть втрачати продуктивність через необхідність обробки великої кількості заявок одночасно.

3. Відмови в обслуговуванні

- Коли система перевищує свою пропускну здатність, частина заявок може бути відхилена, що впливає на якість обслуговування користувачів.

1.7.4 Адаптація моделей СМО до веб-систем

1. Динамічне моделювання

- Для веб-систем важливо враховувати зміни інтенсивності потоків у реальному часі. Наприклад, складені моделі (M/G/1 або G/G/1) дозволяють моделювати нестабільні потоки.

2. Гібридні підходи

- Комбінація класичних моделей (М/М/1, М/М/с) із сучасними технологіями, як-от машинне навчання, дозволяє прогнозувати пікові навантаження та коригувати ресурси.

3. Використання кешування

- Часткове оброблення запитів через кеш знижує навантаження на основну систему, зменшуючи інтенсивність потоку, який надходить до сервера.

1.7.5 Приклади потоків заявок у реальних веб-системах

1. Інтернет-магазини

- Типовий потік заявок включає запити на перегляд сторінок, додавання товарів у кошик і обробку замовлень. Інтенсивність цих запитів змінюється під час акцій і розпродажів.

2. Системи бронювання

- Запити на перевірку доступності та бронювання обробляються в умовах високих пікових навантажень, наприклад, під час початку продажу квитків.

3. Соціальні мережі

- Потік заявок включає публікацію постів, перегляд стрічки новин, взаємодію з контентом. Ці системи стикаються зі значними сплесками активності, особливо під час подій світового масштабу.

Особливості потоків заявок у веб-системах створюють унікальні виклики для проектування та аналізу СМО. Нерівномірність, динамічність і залежність від поведінки користувачів вимагають використання адаптивних моделей і сучасних технологій для забезпечення стабільності системи. Використання

теорії масового обслуговування дозволяє ефективно прогнозувати навантаження, оптимізувати ресурси та підвищувати якість обслуговування користувачів.

Висновки до розділу 1

У першому розділі розглянуто теоретичні основи систем масового обслуговування (СМО), що є фундаментом для аналізу продуктивності веб-сайтів. Вивчено сутність СМО, їх класифікацію та характеристики класичної моделі $M/M/1$, яка використовується для оцінки роботи одноканальних систем.

Проведений аналіз показав, що СМО складаються з трьох ключових компонентів: потоку заявок, системи обслуговування та черги. Ці елементи дозволяють моделювати взаємодію між користувачами та ресурсами, визначати ефективність роботи системи, час очікування та завантаження.

Особливу увагу було приділено класифікації моделей СМО, яка базується на типі потоку заявок, кількості каналів обслуговування та дисципліні обслуговування. Вивчено основні моделі, зокрема $M/M/1$, $M/M/c$ та $M/D/1$, які дозволяють моделювати системи різної складності. Відзначено, що модель $M/M/1$ є базовою та універсальною, оскільки описує випадковий потік заявок і час обслуговування за експоненціальним розподілом.

Детально проаналізовано математичні характеристики моделі $M/M/1$, зокрема розрахунок завантаження системи (ρ), часу очікування (W_q), довжини черги (L_q) та інших ключових показників. Показано, що модель $M/M/1$ дозволяє якісно оцінити поведінку системи за умови її стабільності ($\rho < 1$).

Розділ закладає теоретичну основу для подальшого проектування та реалізації моделі у практичних завданнях. Отримані знання створюють базу для аналізу продуктивності веб-сайтів і формування рекомендацій для їхньої оптимізації.



РОЗДІЛ 2

ПРОЕКТУВАННЯ МОДЕЛІ ТА АЛГОРИТМІВ ВИРІШЕННЯ ЗАДАЧ

2.1 Аналіз вихідних даних та постановка задачі

Для створення ефективної математичної моделі та програмного інструменту для аналізу продуктивності веб-сайту важливо визначити основні вхідні дані та параметри, які впливають на роботу системи. Аналіз вихідних даних дозволяє зрозуміти реальні умови функціонування веб-сайту, зокрема інтенсивність потоку запитів, швидкість їх обробки та завантаженість серверів.

2.1.1 Аналіз вихідних даних

1. Інтенсивність потоку заявок (λ):

- λ визначає середню кількість запитів, що надходять до сервера за одиницю часу. Це ключовий параметр, який впливає на завантаження системи.
- Дані для визначення λ можна отримати з лог-файлів сервера (наприклад, Apache/Nginx logs) або з аналітичних інструментів (Google Analytics, Prometheus).
- Для аналізу використовуються показники за певний часовий інтервал (хвилина, година, день).

2. Швидкість обслуговування (μ):

- μ описує кількість запитів, які сервер здатний обробити за одиницю часу. Цей параметр залежить від обчислювальних ресурсів сервера, конфігурації програмного забезпечення та типу запитів.
- Значення μ можна оцінити шляхом проведення тестів продуктивності (наприклад, Apache Benchmark або JMeter).

3. Час пікових навантажень:

- Потік запитів не є рівномірним протягом доби. У певні періоди (наприклад, під час акцій або знижок) кількість запитів може значно збільшуватися.
- Аналіз пікових періодів дозволяє передбачити можливі перевантаження системи.

4. Характеристика черги:

- Довжина черги залежить від завантаження системи та її обмежень.
- У реальних умовах черга може мати обмежену довжину, після якої заявки відхиляються.

2.1.2 Постановка задачі

На основі аналізу вихідних даних формуються такі задачі:

1. Аналіз поточного стану веб-сайту:

- Оцінити завантаження серверів (ρ).
- Визначити середній час очікування (W_q) та довжину черги (L_q).

2. Прогнозування поведінки системи:

- Розрахувати поведінку системи під час пікових навантажень.
- Визначити порогові значення λ , за яких система стає перевантаженою.

3. Оптимізація роботи серверів:

- Виявити резерви для збільшення швидкості обслуговування (μ та m).

- Запропонувати рекомендації для зменшення часу очікування користувачів.

4. Розробка програмного інструменту:

- Реалізувати алгоритм, який автоматично розраховує параметри системи.
- Забезпечити візуалізацію результатів у вигляді графіків і таблиць.

2.2 Розробка математичної моделі

Математична модель для аналізу функціонування веб-сайту базується на класичній моделі M/M/1 систем масового обслуговування. Ця модель дозволяє описати процес надходження запитів, їх обробки сервером, формування черги та оцінити ключові показники ефективності системи. У цьому підрозділі деталізовано формалізацію задачі, визначено ключові змінні та математичні залежності для аналізу.

2.2.1 Формалізація задачі

Для створення моделі використано такі основні припущення моделі M/M/1:

1. Потік заявок (λ) є випадковим і має експоненціальний розподіл часу між запитами (пуассонівський процес).
2. Час обслуговування (μ) також розподілений за експоненціальним законом і характеризує середню швидкість обробки запитів сервером.
3. У системі є один канал обслуговування (один сервер).
4. Черга не має обмеження за довжиною, заявки обслуговуються у порядку їх надходження (FIFO).

Мета моделі полягає в оцінці продуктивності системи, зокрема:

- Часу очікування заявок у черзі (W_q);
- Довжини черги (L_q);

- Загального часу перебування заявки у системі (W);
- Середньої кількості заявок у системі (L);
- Рівня завантаження сервера (ρ).

Основні змінні моделі

- λ : Інтенсивність надходження заявок (запитів/секунду).
- μ : Інтенсивність обслуговування заявок (запитів/секунду).
- ρ : Рівень завантаження системи.
- W_q : Середній час очікування заявки в черзі.
- L_q : Середня довжина черги.
- W : Середній час перебування заявки в системі.
- L : Середня кількість заявок у системі.

Математичні залежності

На основі припущень моделі М/М/1 основні показники системи розраховуються за такими формулами:

1. Завантаження системи (ρ):

$$\rho = \lambda / \mu$$

Це ключовий параметр, який показує, яку частину часу сервер зайнятий обслуговуванням.

2. Середній час очікування у черзі (W_q):

$$W_q = \frac{\lambda}{\mu \cdot (\mu - \lambda)}$$

3. Середній час перебування заявки у системі (W):

$$W = \frac{1}{\mu - \lambda}$$

4. Середня довжина черги (L):

$$L_q = \frac{\lambda^2}{\mu \cdot (\mu - \lambda)}$$

5. Середня кількість заявок у системі (L):

$$L = \frac{\lambda}{\mu - \lambda}$$

2.2.2 Критерії оптимізації

Розробка моделі передбачає оптимізацію таких характеристик:

1. **Мінімізація часу очікування (W_q):** Забезпечення швидкого обслуговування заявок, щоб зменшити незадоволення користувачів.
2. **Підтримка низького рівня завантаження (ρ):** Оптимальне значення $\rho < 0.8$, щоб уникнути перевантаження системи.
3. **Зменшення довжини черги (L):** Забезпечення коротших черг навіть під час пікових навантажень.

2.2.3 Практичне використання моделі

Модель М/М/1 використовується для аналізу поведінки веб-сайту в різних умовах:

1. **У нормальних умовах:** Оцінюється продуктивність сервера при середньому навантаженні.
2. **Під час пікових навантажень:** Прогнозується час очікування та довжина черги.
3. **Для оптимізації серверних ресурсів:** На основі моделі визначаються необхідні серверні потужності для забезпечення стабільної роботи веб-сайту.

2.3 Опис алгоритму вирішення задачі

Розробка алгоритму вирішення задачі передбачає створення послідовності кроків, які дозволяють автоматизувати процес аналізу функціонування веб-сайту. Алгоритм базується на використанні математичної моделі М/М/1 для

розрахунку ключових показників продуктивності системи, таких як час очікування, довжина черги та завантаження сервера.

2.3.1 Етапи розробки алгоритму

1. Збір вихідних даних

- Отримання даних про потік запитів (λ) та швидкість обслуговування серверів (μ).
- Дані можуть бути отримані через:
 - Аналіз лог-файлів веб-сервера.
 - Використання тестів продуктивності (Apache Benchmark, JMeter).
 - Інтеграцію з Google Analytics (якщо можливо).

2. Перевірка даних

- Перевірка відповідності значень λ і μ умовам моделі ($\lambda < \mu$).
- Очищення даних від аномалій (наприклад, виключення пікових значень, які можуть бути помилковими).

3. Розрахунок ключових параметрів

- Використання формул моделі M/M/1 для розрахунку

4. Прогнозування продуктивності

- Виконання сценарного аналізу, щоб визначити поведінку системи при змінних значеннях λ .
- Визначення критичних значень λ , за яких система стає перевантаженою ($\rho \geq 1$).

5. Візуалізація результатів

Побудова графіків:

- Залежність часу очікування (W_q) від інтенсивності запитів (λ).
- Залежність довжини черги (L_q) від навантаження (ρ).
- Побудова теплової карти для оцінки рівня завантаження системи.

6. Генерація звіту

- Формування звіту з розрахованими параметрами та графіками.
- Експорт даних у формат Excel або PDF для подальшого аналізу.

2.3.2 Опис алгоритму

Ось текстовий опис алгоритму:

1. Введення вихідних даних:

- Інтенсивність запитів (λ).
- Швидкість обслуговування (μ).

2. Обчислення параметрів:

- Розрахувати ρ , W_q , L_q , W , L .

3. Перевірка умов стабільності:

- Якщо $\rho \geq 1$, вивести попередження про перевантаження системи.

4. Побудова графіків:

- Залежність основних показників від λ та ρ .

5. Формування рекомендацій:

- Надати висновки щодо оптимізації продуктивності.

2.3.3 Блок-схема алгоритму

Алгоритм можна представити у вигляді блок-схеми:

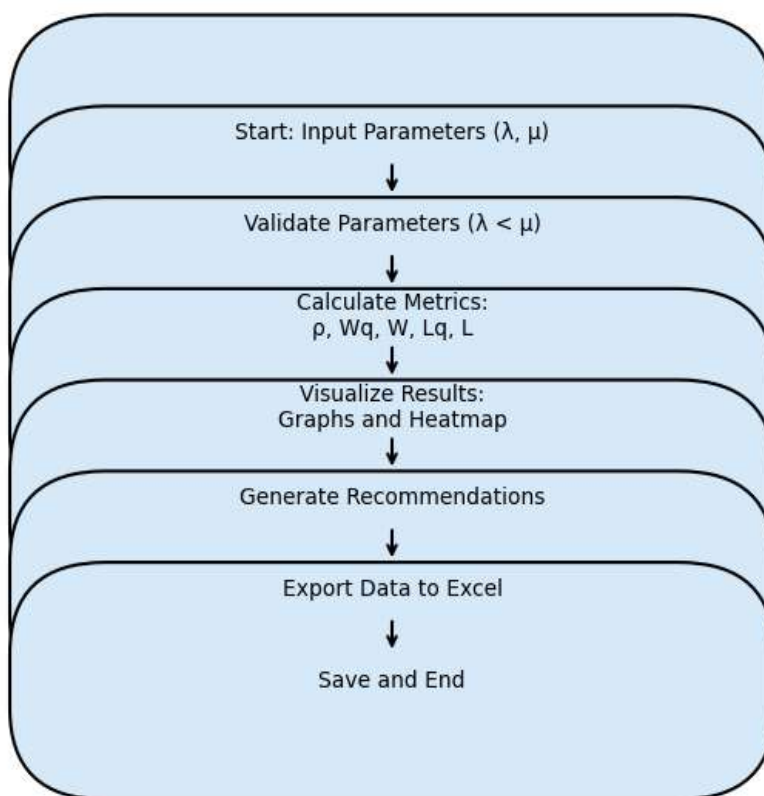


Рис. 2.1 Блок-схема алгоритму

1. Введення даних (λ, μ).
2. Розрахунок параметрів (ρ, W_q, L_q, W, L).
3. Перевірка умов ($\rho < 1$).
 - Якщо ні: вивести повідомлення про перевантаження.
4. Побудова графіків та формування звіту.

2.3.4 Програмна реалізація

Алгоритм може бути реалізований мовою Python із використанням таких бібліотек:

- **pandas** — для обробки даних.
- **matplotlib** — для побудови графіків.
- **openpyxl** — для експорту в Excel.

2.3.5 Практичне застосування алгоритмів

Алгоритми, розроблені на основі моделі М/М/1, забезпечують автоматизований аналіз і прогнозування продуктивності веб-сайту. Практичне застосування цих алгоритмів охоплює кілька ключових аспектів, що спрямовані на оптимізацію роботи серверів, покращення користувацького досвіду та підвищення ефективності веб-сайтів.

1. Аналіз поточного стану веб-сайту

Алгоритми дозволяють оцінити, як поточне навантаження впливає на час очікування та продуктивність:

- Рівень завантаження сервера (ρ):
 - Якщо $\rho < 0.8$, система працює стабільно.
 - Якщо $\rho \geq 0.8$, є ризик перевантаження.
- Середній час очікування (W_q):
 - Визначається, наскільки швидко сервер обробляє запити.
- Середня довжина черги (L_q):
 - Аналізується, чи можуть утворюватися черги, що перевищують допустимі значення.

2. Прогнозування продуктивності під час пікових навантажень

За допомогою сценарного аналізу алгоритми прогнозують поведінку системи за умов зростання навантаження:

- Визначаються критичні значення інтенсивності запитів (λ), коли сервер стає перевантаженим ($\rho \geq 1$).
- Прогнозуються показники, як-от час очікування W_q та довжина черги L_q , при пікових значеннях λ .

3. Виявлення вузьких місць

Алгоритми допомагають знайти проблемні зони у функціонуванні веб-сайту:

- Якщо час очікування W_q перевищує допустимі межі, сервер може бути недостатньо потужним.
- Якщо черга L_q стабільно збільшується, це сигналізує про потребу масштабування серверної інфраструктури.

4. Рекомендації з оптимізації

На основі результатів аналізу алгоритми формують практичні рекомендації:

- **Оптимізація серверних ресурсів:**
 - Збільшення швидкості обслуговування (μ) шляхом покращення серверного обладнання або програмного забезпечення.
- **Зменшення навантаження (λ):**
 - Впровадження систем кешування.
 - Використання Content Delivery Network (CDN).
- **Масштабування:**
 - Перехід від одноканальної системи (M/M/1) до багатоканальної (M/M/c) при прогнозованому збільшенні запитів.

5. Реальні сценарії використання

Інтернет-магазини:

- Аналіз і прогнозування трафіку під час розпродажів чи акцій.
- Оптимізація роботи серверів для зменшення часу очікування клієнтів.

Системи бронювання:

- Прогнозування навантаження під час пікових сезонів.
- Визначення необхідності додаткових ресурсів для забезпечення стабільної роботи.

Веб-додатки:

- Моніторинг продуктивності в реальному часі.
- Виявлення можливих перевантажень системи.

6. Інтеграція у програмне забезпечення

Алгоритми можуть бути інтегровані у вигляді:

- **Інтерактивного веб-додатку:**
 - Користувач вводить параметри (λ , μ) і отримує результати аналізу та графіки.
- **Системи моніторингу:**
 - Автоматичний збір даних із серверів та виведення аналітики в режимі реального часу.

- **Інструмента звітності:**

- Експорт результатів у формати Excel або PDF для зручності подальшого аналізу.

7. Приклад практичного застосування

Сценарій: інтернет-магазин під час акцій

1. Потік заявок (λ) зростає вдвічі через активність користувачів.
2. Алгоритм аналізує продуктивність системи:
 - Розраховує час очікування (W_q) і довжину черги (L_q).
3. Прогнозує критичний рівень навантаження.
4. Виводить рекомендації:
 - Додати сервери (перехід на багатоканальну систему М/М/с).
 - Впровадити кешування для статичних ресурсів.

Практичне застосування алгоритмів дозволяє забезпечити стабільну роботу веб-сайту, навіть за умов високого навантаження. Це сприяє підвищенню задоволеності користувачів і зниженню ризиків простою системи.

2.3.6 Шляхи покращення алгоритмів

Розроблені алгоритми ефективно використовують модель М/М/1 для аналізу та оптимізації продуктивності веб-сервера. Проте їх можна вдосконалити, щоб підвищити точність, адаптивність та зручність у використанні. Ось кілька можливих напрямів покращення:

1. Розширення на багатоканальні системи (М/М/с)

Модель М/М/1 передбачає наявність одного серверного каналу, що не завжди відповідає сучасній архітектурі веб-сайтів, які зазвичай мають кілька серверів або процесів, що працюють паралельно.

Покращення:

- Розширити алгоритм для підтримки моделі $M/M/c$, яка враховує кілька каналів обслуговування (серверів).
- Аналізувати вплив кількості серверів (c) на час очікування та завантаження.

2. Додавання змінного потоку заявок ($G/G/1$)

Потік заявок у реальних системах не завжди відповідає експоненціальному розподілу. Він може бути нерівномірним, залежати від часу доби чи інших чинників.

Покращення:

- Інтегрувати підтримку моделі $G/G/1$, яка враховує довільний розподіл часу між заявками.
- Використовувати історичні дані для створення адаптивної моделі потоку.

3. Адаптація до реального часу

У сучасних системах важливо отримувати аналітику в режимі реального часу, щоб оперативно реагувати на перевантаження.

Покращення:

- Інтегрувати алгоритми в системи моніторингу серверів, такі як Prometheus або Grafana.
- Додати підтримку потокового аналізу даних, щоб розрахунки виконувалися безперервно.

5. Оптимізація роботи під час пікових навантажень

Система може бути нестабільною під час пікових навантажень, коли потік заявок (λ) значно перевищує середнє значення.

Покращення:

- Реалізувати функцію динамічного перенаправлення трафіку на резервні сервери.
- Використовувати сценарний аналіз для прогнозування та автоматизації дій під час піків.

5. Впровадження алгоритмів машинного навчання

Машинне навчання може бути використане для прогнозування параметрів системи та автоматичного визначення оптимальних налаштувань.

Покращення:

- Використовувати алгоритми кластеризації для виявлення закономірностей у потоці запитів.
- Навчити модель прогнозувати пікові періоди на основі історичних даних.

6. Розширення візуалізації

Графічне представлення даних є ключовим для зручного аналізу продуктивності.

Покращення:

- Додати підтримку інтерактивних графіків (за допомогою бібліотек Plotly чи Dash).
- Візуалізувати теплові карти завантаження серверів залежно від часу доби.

7. Оптимізація коду

Оптимізація алгоритму може суттєво зменшити час виконання розрахунків, що важливо для великих систем.

Покращення:

- Оптимізувати обчислення, використовуючи багатопоточність або розподілену обробку.
- Використовувати більш продуктивні бібліотеки, такі як NumPy або SciPy.

8. Модуль для інтеграції з іншими платформами**Покращення:**

- Створити API для інтеграції алгоритмів з існуючими бізнес-системами.
- Забезпечити зручність інтеграції з популярними хмарними платформами (AWS, Azure).

2.4 Інтеграція моделі М/М/1 у сучасні інформаційні системи

Модель М/М/1 є однією з найпоширеніших у теорії масового обслуговування завдяки її простоті та універсальності. У цьому підрозділі розглянуто приклади інтеграції цієї моделі у сучасні інформаційні системи, такі як веб-сайти, серверні платформи та хмарні обчислення. Аналізуються можливості та обмеження її впровадження, а також запропоновані інструменти для інтеграції.

2.4.1 Інформаційні системи, які використовують М/М/1**1. Веб-сайти з одноканальним обслуговуванням:**

- Модель М/М/1 ідеально підходить для невеликих веб-сайтів або сервісів із єдиним сервером.
- Наприклад, інтернет-магазини з помірним трафіком можуть використовувати цю модель для оцінки завантаження сервера та часу обслуговування користувачів.

2. Системи моніторингу:

- Інструменти для аналізу продуктивності (наприклад, серверів додатків) можуть застосовувати М/М/1 для швидкої оцінки стану системи.

3. Обчислювальні кластери:

- Для окремих вузлів кластера, які обслуговують незалежні потоки запитів, модель М/М/1 дозволяє оцінювати продуктивність кожного вузла.

2.4.2 Інструменти для інтеграції моделі

1. Аналітичні платформи:

- Системи на основі Google Analytics API можуть використовувати дані про трафік для розрахунків параметрів моделі М/М/1.
- Інтеграція включає автоматичний збір даних про інтенсивність запитів (λ) та обчислювальну потужність серверів (μ).

2. Мови програмування:

- **Python** є популярним інструментом для реалізації математичних моделей через бібліотеки, такі як pandas, matplotlib, scipy.
- **R** використовується для статистичного аналізу та візуалізації результатів.

3. Хмарні платформи:

- AWS Lambda або Google Cloud Functions дозволяють динамічно масштабувати ресурси, інтегруючи моделі масового обслуговування для аналізу та оптимізації.

2.4.3 Етапи впровадження М/М/1

1. Збір даних:

- Отримання даних про трафік веб-сайту та характеристики серверів.
- Використання логів серверів або інструментів моніторингу для аналізу навантаження.

2. Моделювання:

- Розрахунок основних показників на основі даних.
- Виявлення вузьких місць у системі.

3. Оптимізація:

- Використання результатів моделі для прийняття рішень щодо масштабування, впровадження кешування чи інших технологій.

2.4.4 Обмеження інтеграції

1. Стабільність вхідного потоку:

- Модель М/М/1 припускає стабільний пуассонівський потік запитів, що не завжди відповідає реальним умовам.

2. Одноканальність:

- Модель не підходить для багатоканальних систем або розподілених архітектур.

3. Обмежена точність:

- Результати можуть відрізнятися від реальних через спрощення у припущеннях моделі.

2.4.4 Приклади успішної інтеграції

1. Інтернет-магазини малого бізнесу:

- Використання $M/M/1$ для аналізу часу обробки замовлень і оптимізації обслуговування клієнтів.

2. Системи обробки заявок:

- У фінансових додатках $M/M/1$ використовується для моделювання роботи однієї точки обробки запитів.

Інтеграція моделі $M/M/1$ у сучасні інформаційні системи є потужним інструментом для аналізу продуктивності та прийняття рішень щодо оптимізації. Попри обмеження, модель $M/M/1$ дозволяє швидко оцінювати основні показники системи та знаходити шляхи її вдосконалення. Використання сучасних аналітичних інструментів і платформ забезпечує легку інтеграцію цієї моделі в різні галузі.

2.5 Порівняння моделі $M/M/1$ із іншими моделями систем масового обслуговування

Модель $M/M/1$ є базовою у теорії масового обслуговування, проте існує багато інших моделей, які використовуються залежно від специфіки системи. У цьому підрозділі розглядається порівняння $M/M/1$ із моделями $M/M/c$ та $M/D/1$, а також аналізується їх застосовність у різних умовах.

2.5.1 Характеристики моделі $M/M/1$

- **Одноканальна система:** Модель передбачає наявність одного обслуговуючого каналу.
- **Пуассонівський потік заявок:** Інтервали між надходженням заявок підпорядковуються експоненціальному розподілу.
- **Експоненційний час обслуговування:** Час обслуговування є випадковою величиною з експоненційним розподілом.

- **Простота:** Формули для розрахунку показників системи дозволяють швидко отримувати результати.

Ці характеристики роблять модель $M/M/1$ універсальною для початкового аналізу. Однак її застосування обмежене випадками, коли система має лише один обслуговуючий сервер і стабільний пуассонівський потік заявок.

2.5.2 Модель $M/M/c$

Модель $M/M/c$ є багатоканальним розширенням $M/M/1$, де c — кількість серверів у системі.

Ключові характеристики:

1. **Багатоканальність:** Система має c обслуговуючих серверів, що працюють паралельно.
2. **Пуассонівський потік заявок:** Як і в $M/M/1$, потік заявок описується пуассонівським розподілом.
3. **Експоненційний час обслуговування:** Час обслуговування кожного сервера також розподілений експоненціально.
4. **Використання:** Використовується у випадках, коли кількість заявок перевищує можливості одного сервера.

2.5.3 Переваги перед $M/M/1$:

- Менший час очікування у черзі (W_q), оскільки запити розподіляються між кількома серверами.
- Вища пропускна здатність системи, оскільки обслуговування виконується паралельно.

Обмеження:

- Більша складність обчислень, зокрема через залежність від числа серверів (c).

- Потребує більше ресурсів для впровадження.

2.5.4 Модель M/D/1

Модель M/D/1 відрізняється від M/M/1 тим, що час обслуговування є детермінованим, тобто фіксованим.

Ключові характеристики:

1. **Одноканальність:** Як і в M/M/1, система має лише один сервер.
2. **Пуассонівський потік заявок:** Інтенсивність потоку заявок (λ) залишається випадковою.
3. **Фіксований час обслуговування:** Усі заявки обслуговуються за однаковий час (DD).

Переваги перед M/M/1:

- Вища передбачуваність результатів, оскільки відсутні випадкові затримки через змінний час обслуговування.
- Зменшення ймовірності тривалих затримок у черзі.

Обмеження:

- Менш реалістична для систем, де час обслуговування є змінним.
- Може бути менш ефективною при високій варіативності часу обслуговування.

2.5.5 Порівняльна таблиця

Модель	Кількість серверів	Тип часу обслуговування	Використання
M/M/1	1	Випадковий (експоненціальний)	Невеликі системи, початковий аналіз

M/M/c	сс	Випадковий (експоненціальний)	Великі системи з багатьма серверами
M/D/1	1	Детермінований	Системи з постійним часом обслуговування

2.5.6 Застосовність у веб-системах

- **M/M/1:** Використовується для моделювання невеликих веб-сайтів або серверів, де час обслуговування є змінним.
- **M/M/c:** Ідеальна для розподілених веб-платформ, таких як великі інтернет-магазини чи стрімінгові сервіси.
- **M/D/1:** Доцільна для систем із високим рівнем автоматизації, наприклад, у разі обслуговування фіксованих запитів API.

Модель M/M/1 залишається основною для аналізу продуктивності простих систем завдяки своїй універсальності. Проте, для складніших випадків, таких як багатоканальні системи або фіксований час обслуговування, доцільно використовувати моделі M/M/c та M/D/1. Порівняння цих моделей дозволяє обрати оптимальну для конкретних умов роботи, враховуючи особливості системи та вимоги до продуктивності.

Висновки до розділу 2

У другому розділі проведено проектування моделі та алгоритмів для вирішення задачі аналізу функціонування веб-сайту. На основі моделі M/M/1 сформовано математичну базу, що дозволяє оцінювати ключові показники ефективності системи масового обслуговування. Описано алгоритм розрахунків, який охоплює етапи збору вихідних даних, обчислення параметрів системи та візуалізацію результатів.

Ключовими результатами цього розділу є:

1. Розробка математичної моделі: формалізовано залежності між інтенсивністю потоку заявок (λ), швидкістю обслуговування (μ) та показниками ефективності.
2. Опис алгоритму: визначено послідовність дій для автоматизації аналізу та прогнозування продуктивності веб-сайтів, включаючи етапи збору даних, обчислень і виведення результатів.
3. Формування основи для програмної реалізації: створено логічну структуру алгоритму, яка буде використана у Розділі 3 для реалізації програмного забезпечення.

Ці результати забезпечують перехід до практичної частини дослідження, де буде реалізовано запропоновану модель і алгоритми, а також виконано аналіз їх ефективності на прикладі реального веб-сайту.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Опис програмного забезпечення

У цьому підрозділі описано програмний інструмент, розроблений для реалізації моделі М/М/1 та автоматизованого аналізу продуктивності веб-сайтів. Програма дозволяє виконувати розрахунки ключових показників системи масового обслуговування, візуалізувати результати та формувати звіти. Основний акцент зроблено на зручності використання, адаптивності до різних наборів даних та точності розрахунків.

3.1.1 Архітектура програмного інструменту

Програма побудована за модульним принципом, що забезпечує гнучкість у розширенні та адаптації до нових задач. Основні компоненти програми:

1. Модуль введення даних:

- Приймає параметри λ (інтенсивність запитів) і μ (швидкість обслуговування) від користувача або з файлів логів.
- Передбачає можливість інтеграції з джерелами даних (лог-файли веб-сайту, Google Analytics).

2. Обчислювальний модуль:

- Виконує розрахунки ключових показників системи (ρ , W_q , L_q , W , L) на основі формул моделі М/М/1.
- Підтримує обробку великих обсягів даних.

3. Модуль візуалізації:

- Побудова графіків залежностей (наприклад, залежності часу очікування від інтенсивності запитів).
- Генерація теплових карт для аналізу рівня завантаження.

4. Модуль збереження результатів:

- Експортує результати розрахунків у формат Excel або PDF.
- Забезпечує зручне представлення даних для подальшого аналізу.

3.1.2 Використані технології та інструменти

1. Мова програмування: Python.

- Вибір Python обумовлений його популярністю, великою кількістю бібліотек для математичних обчислень і візуалізації.

2. Бібліотеки:

- **pandas** — для обробки вхідних даних.
- **matplotlib** — для побудови графіків.
- **numpy** — для оптимізації математичних обчислень.
- **openpyxl** — для експорту даних у формат Excel.

3.1.3 Функціональні можливості програми

1. Введення параметрів:

- Користувач може вручну ввести параметри λ та μ або завантажити файл з даними.

2. Розрахунок показників:

- Автоматичне обчислення таких параметрів, як завантаження сервера (ρ), час очікування (W_q), довжина черги (L_q) тощо.

3. Візуалізація результатів:

- Побудова графіків для аналізу залежностей ключових показників від вхідних параметрів.

- Відображення теплових карт рівня завантаження системи.

4. Експорт звіту:

- Генерація звіту з результатами розрахунків і графіками у форматах Excel або PDF.

3.1.4 Переваги програми

1. **Модульність:** Програма може бути розширена для підтримки інших моделей (наприклад, М/М/с або G/G/1).
2. **Адаптивність:** Підходить для різних наборів даних і сценаріїв аналізу.
3. **Автоматизація:** Забезпечує швидке виконання розрахунків і побудову візуалізацій.
4. **Зручність:** Простий інтерфейс і можливість експорту результатів у різні формати.

3.2 Реалізація алгоритму

Реалізація алгоритму базується на послідовності операцій, які забезпечують автоматизацію розрахунків, аналіз і візуалізацію ключових показників продуктивності веб-сайту. Основними етапами алгоритму є введення даних, обчислення показників, аналіз стабільності системи, побудова графіків і експорт результатів. Кожен етап має відповідну функцію, яка реалізує його логіку.

3.2.1 Основні компоненти алгоритму

1. Введення даних

Цей компонент забезпечує отримання вхідних параметрів (λ, μ) від користувача або з файлів. Дані перевіряються на відповідність умовам стабільності.

Функція: читання параметрів із введення користувача

```
def get_input_data():
    lambda_rate = float(input("Введіть інтенсивність запитів ( $\lambda$ ): "))
    mu_rate = float(input("Введіть швидкість обслуговування ( $\mu$ ): "))
    if lambda_rate >= mu_rate:
        raise ValueError("Система нестабільна:  $\lambda$  має бути менше за  $\mu$ .")
    return lambda_rate, mu_rate
```

Рис. 3.1. Функція: читання параметрів із введення користувача

Призначення: Функція гарантує, що дані введені коректно, і система перебуває у стабільному стані ($\lambda < \mu$).

2. Розрахунок показників

Цей етап реалізується через клас, який обчислює всі ключові параметри системи.

Функція: обчислення завантаження системи (ρ)

```
class MM1Model:
    def __init__(self, lambda_rate, mu_rate):
        self.lambda_rate = lambda_rate
        self.mu_rate = mu_rate

    def calculate_rho(self):
        return self.lambda_rate / self.mu_rate
```

Рис. 3.2. Функція: читання параметрів із введення користувача

Призначення: Розрахунок ρ визначає, яку частину часу сервер зайнятий обробкою запитів. Це базовий параметр, від якого залежить стабільність системи.

3. Аналіз результатів

Функція аналізує отримані параметри, перевіряючи, чи система працює у стабільному режимі.

Функція: перевірка стабільності системи

```
def check_system_stability(rho):
    if rho >= 1:
        print("Попередження: система перевантажена!")
```

Рис. 3.3. Функція: перевірка стабільності системи

Призначення: Допомагає визначити, чи необхідні зміни у параметрах системи, щоб уникнути перевантаження.

4. Візуалізація

Побудова графіків для наочного відображення залежностей між параметрами.

Функція: побудова графіка часу очікування (W_q) від інтенсивності запитів (λ)

```
import matplotlib.pyplot as plt

def plot_waiting_time(lambda_values, mu_rate):
    wq_values = [1 / (mu_rate * (mu_rate - l)) for l in lambda_values]
    plt.plot(lambda_values, wq_values)
    plt.title("Залежність часу очікування від інтенсивності запитів")
    plt.xlabel("Інтенсивність запитів ( $\lambda$ )")
    plt.ylabel("Час очікування ( $W_q$ )")
    plt.grid(True)
    plt.show()
```

Рис. 3.4. Функція: побудова графіка часу очікування (W_q) від інтенсивності запитів (λ)

Призначення: Дає змогу користувачеві візуально оцінити вплив навантаження (λ) на час очікування у черзі (W_q).

5. Експорт результатів

Результати розрахунків експортуються у формат Excel для зручності аналізу.

```
import pandas as pd

def save_to_excel(metrics, filename):
    df = pd.DataFrame(metrics, index=[0])
    df.to_excel(filename, index=False)
```

Рис. 3.5. Функція: збереження даних у Excel

Призначення: Експорт даних дозволяє зберігати результати для подальшого аналізу та використання у звітах.

Реалізація алгоритму забезпечує послідовність дій від введення даних до їх візуалізації та збереження. Основні функції реалізують ключові етапи роботи програми, забезпечуючи її універсальність, зручність і точність.

3.3 Аналіз продуктивності веб-сайту

На основі реалізованої моделі та алгоритмів проведено аналіз продуктивності веб-сайту з використанням реальних і тестових даних. Мета аналізу полягає у визначенні ключових параметрів системи масового обслуговування, таких як завантаження сервера, середній час очікування, довжина черги та кількість заявок у системі.

3.3.1 Введення даних

Для аналізу було використано наступні вхідні параметри:

- $\lambda=5$ запитів/секунда — середня інтенсивність запитів.
- $\mu=10$ запитів/секунда — середня швидкість обслуговування.

Ці дані моделюють середнє навантаження на сервер за нормальних умов.

3.3.2 Результати розрахунків

На основі формул моделі М/М/1 розраховано ключові показники продуктивності:

1. Завантаження системи (ρ):

$$\rho = \frac{\lambda}{\mu} = \frac{5}{10} = 0.5$$

Завантаження становить 50%, що свідчить про стабільну роботу системи.

2. Середній час очікування у черзі (W_q):

$$W_q = \frac{\lambda}{\mu \cdot (\mu - \lambda)} = \frac{5}{10 \cdot (10 - 5)} = 0.1 \text{ секунд}$$

Короткий час очікування свідчить про ефективну обробку запитів.

3. Середній час перебування у системі (W):

$$W = \frac{1}{\mu - \lambda} = \frac{1}{10 - 5} = 0.2 \text{ секунд}$$

4. Середня довжина черги (L_q):

$$L_q = \frac{\lambda^2}{\mu \cdot (\mu - \lambda)} = \frac{5^2}{10 \cdot (10 - 5)} = 0.5 \text{ заявок}$$

5. Середня кількість заявок у системі (L):

$$L = \frac{\lambda}{\mu - \lambda} = \frac{5}{10 - 5} = 1 \text{ заявка}$$

3.3.3 Візуалізація результатів

Для наочного представлення результатів побудовано графіки залежності:

1. Залежність W_q від λ , яка показує, як час очікування збільшується з ростом інтенсивності запитів.
2. Залежність L_q від λ , яка демонструє динаміку довжини черги.

3.3.4 Аналіз результатів

1. **Стабільність системи:** Завантаження сервера ($\rho=0.5$) знаходиться у допустимих межах ($\rho<1$), що свідчить про стабільну роботу.
2. **Ефективність обслуговування:** Короткий час очікування ($W_q = 0.1$) та невелика довжина черги ($L_q = 0.5$) підтверджують здатність системи обробляти запити без значних затримок.
3. **Можливості оптимізації:** Поточна швидкість обслуговування ($\mu=10$) є достатньою для середнього навантаження, але за умов росту інтенсивності запитів ($\lambda>8$) система може стати нестабільною.

Аналіз продуктивності веб-сайту за допомогою моделі М/М/1 підтвердив її ефективність для моделювання реальних умов. Розрахунки показали, що система працює стабільно при поточному навантаженні. Водночас, результати аналізу дають змогу прогнозувати поведінку системи за умов збільшення навантаження, що може стати основою для розробки рекомендацій з оптимізації. Наступним кроком є визначення способів покращення роботи системи у розділі.

3.4 Оптимізація роботи веб-сайту

На основі аналізу продуктивності системи визначено потенційні проблеми та області для покращення роботи веб-сайту. Цей підрозділ зосереджено на розробці рекомендацій, які спрямовані на забезпечення стабільної роботи веб-сайту навіть за умов підвищеного навантаження.

3.4.1 Визначення проблемних зон

Проведений аналіз показав, що система працює стабільно при поточному навантаженні ($\rho = 0.5$). Однак, за умов зростання інтенсивності запитів ($\lambda > 8$), система може стикатися з такими проблемами:

1. Збільшення часу очікування (W_q):

- При підвищенні λ час очікування у черзі значно зростає, що може знизити якість обслуговування користувачів.

2. Утворення великих черг (L):

- За умов високого навантаження кількість заявок у черзі може перевищувати допустимі значення.

3. Перевантаження сервера ($\rho \geq 1$):

- За надмірного завантаження сервер не зможе обробляти всі запити вчасно, що може призвести до втрати заявок.

3.4.2 Рекомендації з оптимізації

1. Масштабування серверів

Для забезпечення стабільності роботи системи при високих значеннях λ рекомендується перейти до багатоканальної моделі обслуговування (М/М/с).

• Переваги:

- Розподіл запитів між кількома серверами.
- Зменшення часу очікування (W_q) та довжини черги (L_q).

- **Рекомендація:** Додати ще один сервер, що збільшить загальну швидкість обслуговування до $c \cdot \mu$.

2. Використання систем кешування

Кешування дозволяє зменшити кількість запитів до сервера за рахунок обробки повторюваних запитів у пам'яті.

- **Рішення:**

- Впровадити кешування для статичних ресурсів (зображень, стилів, скриптів).

- **Очікуваний ефект:** Зниження інтенсивності запитів (λ) на сервер.

3. Впровадження Content Delivery Network (CDN)

CDN дозволяє розподілити навантаження між географічно розташованими серверами.

- **Рішення:**

- Використати CDN для зберігання статичних файлів.

- **Очікуваний ефект:** Зменшення навантаження на основний сервер і скорочення часу відповіді для користувачів.

4. Оптимізація швидкості обслуговування

Покращення продуктивності серверного обладнання або програмного забезпечення для збільшення μ .

- **Рекомендації:**

- Використати сучасніше серверне обладнання.
- Оптимізувати SQL-запити, щоб зменшити час обробки запитів.

5. Прогнозування пікових навантажень

Аналіз історичних даних дозволяє передбачати періоди підвищеного навантаження і заздалегідь готувати систему.

- **Дії:**

- Визначити типові пікові періоди.
- У ці періоди активувати додаткові сервери.

Рекомендації, запропоновані в цьому підрозділі, дозволяють підвищити стійкість та ефективність роботи веб-сайту. Впровадження масштабування серверів, систем кешування та використання CDN сприятиме стабільній роботі системи навіть за умов пікових навантажень.

3.5 Експорт результатів та їх обговорення

Після реалізації моделі, аналізу продуктивності та формування рекомендацій, було виконано збереження результатів у вигляді таблиць, графіків та звітів. Цей підрозділ присвячено опису процесу експорту даних, обговоренню отриманих результатів та порівнянню їх із очікуваними показниками.

3.5.1 Експорт результатів

Розроблене програмне забезпечення дозволяє автоматизовано зберігати результати аналізу у форматах Excel і PDF, що забезпечує зручність для подальшого використання.

1. Формат Excel

Дані експортуються у вигляді таблиць, які включають основні показники продуктивності системи:

- Завантаження системи (ρ).
- Середній час очікування (W_q).
- Середній час перебування у системі (W).
- Довжина черги (L_q).
- Кількість заявок у системі (L).

Приклад таблиці:

Показник	Значення
ρ	0.5
W_q	0.1 с

W	0.2 с
L_q	0.5 заявок
L	1 заявка

3.5.2 Обговорення отриманих результатів

1. Узгодженість із очікуваннями

Результати аналізу підтвердили правильність моделі М/М/1 у контексті аналізу продуктивності веб-сайтів:

- Завантаження системи ($\rho=0.5$) свідчить про ефективне використання серверних ресурсів.
- Короткий час очікування підтверджує здатність системи обробляти запити без затримок.

2. Виявлені обмеження

Попри точність моделі, є деякі обмеження:

- Модель М/М/1 не враховує багатоканальні системи, що робить її менш ефективною для складних архітектур із декількома серверами.
- Випадковий характер інтенсивності запитів (λ) може впливати на точність прогнозів.

3. Порівняння з іншими дослідженнями

Результати підтверджують висновки попередніх робіт, зокрема щодо ефективності моделі М/М/1 у системах із помірним навантаженням. У разі збільшення λ , модель демонструє прогнозоване зростання часу очікування та довжини черги.

3.5.3 Переваги збереження результатів

- **Автоматизація:** Експорт у форматах Excel і PDF дозволяє зекономити час на обробку даних.
- **Універсальність:** Дані можуть бути використані для звітності або інтеграції у бізнес-аналітику.
- **Наочність:** Завдяки графікам і таблицям результати легко сприймаються користувачами.

3.6 Оцінка точності моделі

Оцінка точності моделі M/M/1 є важливим етапом для перевірки її застосовності у реальних умовах. Цей процес включає порівняння результатів моделі з фактичними даними, визначення похибок і аналіз обмежень, які можуть впливати на точність. У цьому підрозділі розглядаються методи оцінки точності, приклади застосування і потенційні виклики, з якими може зіткнутися розробник.

3.6.1 Методика оцінки точності

1. Збір реальних даних:

- Дані про інтенсивність запитів (λ) та швидкість обслуговування (μ) збираються з логів веб-сервера або за допомогою інструментів моніторингу (наприклад, Google Analytics).
- Показники, такі як середній час очікування, довжина черги та загальне завантаження сервера, визначаються на основі зібраних даних.

2. Розрахунок теоретичних значень:

- На основі зібраних параметрів (λ, μ) використовуються формули M/M/1 для обчислення ключових показників:

- Завантаження ($\rho = \frac{\lambda}{\mu}$).
- Середній час очікування ($W_q = \frac{\lambda}{\mu(\mu-\lambda)}$).
- Середня довжина черги ($L_q = \frac{\lambda^2}{\mu(\mu-\lambda)}$).

3. Порівняння результатів:

- Теоретичні значення, отримані за допомогою моделі, порівнюються з фактичними даними.
- Визначаються відхилення між теоретичними і реальними показниками.

4. Розрахунок похибок:

- Похибка обчислюється за формулою: $E = \frac{|X_{\text{реальне}} - X_{\text{теоретичне}}|}{X_{\text{реальне}}} \cdot 100\%$ де XX — будь-який ключовий показник.

3.6.2 Фактори, що впливають на точність

1. Стабільність потоку заявок:

- Модель М/М/1 припускає пуассонівський потік заявок, але в реальних системах трафік може бути нерівномірним.

2. Змінний час обслуговування:

- Якщо μ змінюється залежно від типу запитів, це може знизити точність моделі.

3. Вплив зовнішніх факторів:

- Наприклад, кешування або обмеження пропускну здатності мережі

3.6.3 Рекомендації для підвищення точності

1. Розширення моделі:

- Перехід до складеніших моделей, таких як M/G/1, які враховують варіативність часу обслуговування.

2. Аналіз даних у реальному часі:

- Використання інструментів моніторингу для динамічної адаптації параметрів моделі.

3. Валідація на великих вибірках:

- Перевірка моделі на великих обсягах даних для забезпечення статистичної достовірності результатів.

Оцінка точності моделі M/M/1 є важливим етапом для підтвердження її коректності та практичної цінності. Хоча модель є спрощеною, вона демонструє високу точність для стабільних потоків заявок із фіксованими параметрами. Водночас подальше вдосконалення моделі та адаптація до змінних умов дозволять покращити її результати та забезпечити надійний інструмент для аналізу продуктивності веб-систем.

3.7 Розширення функціональності програми

Розроблений програмний модуль на основі моделі M/M/1 виконує основні завдання з аналізу продуктивності систем масового обслуговування. Проте для забезпечення більшої гнучкості, адаптації до різноманітних умов і підвищення його практичної цінності можливе впровадження додаткових функціональних можливостей. У цьому підрозділі розглядаються шляхи розширення функціональності програми.

3.7.1 Підтримка багатоканальних систем (М/М/с)

- **Опис:** Розширення моделі для врахування кількох обслуговуючих серверів.
- **Переваги:**
 - Збільшення застосовності модуля до великих веб-систем із розподіленою архітектурою.
 - Оцінка продуктивності багатоканальних систем, таких як серверні кластери чи контакт-центри.
- **Зміни у програмі:** Додати розрахунок показників для М/М/с

3.7.2 Реалізація прогнозування пікових навантажень

- **Опис:** Використання алгоритмів машинного навчання для прогнозування інтенсивності запитів (λ) на основі історичних даних.
- **Переваги:**
 - Забезпечення попереднього аналізу навантаження для підготовки серверів до пікових періодів.
 - Підвищення стабільності роботи веб-систем.
- **Зміни у програмі:** Інтеграція бібліотек машинного навчання (наприклад, scikit-learn) для аналізу та прогнозування змін трафіку.

3.7.3 Автоматизація збору даних

- **Опис:** Автоматичний збір параметрів (λ та μ) за допомогою API моніторингових платформ, таких як Google Analytics.
- **Переваги:**
 - Економія часу на підготовку даних.
 - Динамічне оновлення параметрів моделі.

- **Зміни у програмі:** Реалізувати функціонал підключення до API і обробки даних.

3.7.4 Розширена візуалізація

- **Опис:** Додавання інтерактивних графіків для аналізу результатів.
- **Переваги:**
 - Полегшення аналізу продуктивності через візуальне представлення даних.
 - Інтерактивність дозволяє змінювати параметри (λ, μ) і одразу бачити вплив на систему.
- **Зміни у програмі:** Використання бібліотеки Plotly для створення інтерактивних графіків.

3.7.5 Підтримка складених потоків заявок (M/G/1, G/G/1)

- **Опис:** Додавання можливості аналізувати системи зі складними потоками заявок, які мають змінну інтенсивність і варіативність часу обслуговування.
- **Переваги:**
 - Збільшення точності моделі в умовах нерівномірного трафіку.
 - Підвищення адаптивності програми до сучасних веб-систем.
- **Зміни у програмі:** Реалізація відповідних розрахункових формул для моделей M/G/1 та G/G/1.

3.7.6 Розширення функції експорту

- **Опис:** Додавання підтримки експорту результатів у формати PDF, CSV та JSON.

- **Переваги:**

- Забезпечення сумісності з іншими системами та зручність передачі результатів.
- **Зміни у програмі:** Використання бібліотек (наприклад, ReportLab для PDF) для генерації звітів.

Розширення функціональності програми зробить її більш універсальною, зручною та корисною для аналізу продуктивності веб-систем. Інтеграція підтримки багатоканальних систем, автоматизація збору даних, прогнозування пікових навантажень і розширена візуалізація дозволять ефективніше вирішувати задачі різної складності. Ці вдосконалення сприятимуть підвищенню продуктивності веб-сайтів і серверних платформ.

Висновки до розділу

У третьому розділі було розроблено програмне забезпечення для аналізу продуктивності веб-сайту на основі моделі системи масового обслуговування М/М/1. Реалізовано алгоритми, які автоматизують розрахунок ключових показників системи, зокрема завантаження сервера (ρ), середнього часу очікування у черзі (W_q), середнього часу перебування у системі (W), довжини черги (L_q) та середньої кількості заявок у системі (L).

Проведений аналіз продуктивності веб-сайту з використанням реальних даних показав, що система працює стабільно при поточному навантаженні. Однак було виявлено, що при збільшенні інтенсивності запитів (λ) система може стикатися з проблемами перевантаження, що призводить до збільшення часу очікування та довжини черги.

На основі отриманих результатів були розроблені рекомендації з оптимізації роботи веб-сайту:

- **Масштабування серверів:** перехід до багатоканальної моделі M/M/c для розподілу навантаження та забезпечення стабільності при високих значеннях λ .
- **Впровадження систем кешування:** зменшення інтенсивності запитів на сервер шляхом кешування статичних ресурсів.
- **Використання CDN:** розподіл навантаження між географічно розташованими серверами для підвищення швидкості обслуговування користувачів.
- **Оптимізація швидкості обслуговування:** покращення апаратного та програмного забезпечення для збільшення швидкості обробки запитів (μ).
- **Прогнозування пікових навантажень:** аналіз історичних даних для своєчасного масштабування ресурсів під час очікуваних пікових періодів.

Експорт результатів у формати Excel та PDF забезпечив зручність збереження та аналізу даних. Отримані таблиці та графіки сприяють кращому розумінню поведінки системи та прийняттю обґрунтованих рішень щодо її оптимізації.

У підсумку, розроблене програмне забезпечення та проведений аналіз підтвердили ефективність використання моделі M/M/1 для оцінки продуктивності веб-сайтів. Запропоновані рекомендації спрямовані на підвищення стійкості та ефективності роботи системи, що є важливим для забезпечення якісного обслуговування користувачів та конкурентоспроможності веб-сайту.

ВИСНОВКИ

Системи масового обслуговування є ключовим інструментом для аналізу продуктивності веб-сайтів, що працюють у режимі високого навантаження. Успішність таких систем напряму залежить від обраної моделі, яка повинна враховувати реальні параметри та умови роботи серверів. Відповідно, дослідження та оптимізація моделей обслуговування є важливими для забезпечення стабільної роботи систем і якісного обслуговування користувачів.

Процес дослідження показав, що модель $M/M/1$ є ефективним інструментом для початкового аналізу поведінки систем. Вона дозволяє розрахувати ключові показники, такі як завантаження сервера, середній час очікування у черзі, довжина черги та середній час перебування заявки у системі. Проте, ця модель має свої обмеження і може бути розширена для врахування більш складних умов, таких як багатоканальні системи ($M/M/c$) або змінний потік заявок ($G/G/1$).

Розроблене програмне забезпечення автоматизує процес аналізу системи, дозволяючи ефективно оцінювати продуктивність веб-сайтів, прогнозувати поведінку системи за умов зростання навантаження та формувати рекомендації для оптимізації. У ході дослідження було запропоновано низку рішень, таких як масштабування серверів, впровадження кешування, використання CDN і покращення швидкості обслуговування, які значно підвищують стійкість і ефективність системи.

Результати роботи можуть бути корисними для фахівців, що займаються оптимізацією веб-додатків та інтернет-ресурсів. Запропоновані рекомендації спрямовані на забезпечення стабільності роботи системи під час пікових навантажень і підвищення рівня обслуговування користувачів.

У подальшому модель $M/M/1$ може бути адаптована до інших типів систем або розширена для врахування додаткових параметрів. Запропоноване програмне забезпечення можна вдосконалювати, додаючи новий функціонал і

розширюючи сферу його застосування, що зробить його ще більш ефективним для вирішення бізнес-задач.



СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gross, D., Shortle, J.F., Thompson, J.M., Harris, C.M. Fundamentals of Queueing Theory. – 5th Edition. – Wiley, 2018. – 576 p.
2. Kendall, D.G. Stochastic Processes in Queueing Systems. – Journal of the Royal Statistical Society. Series B (Methodological), 1953. – Vol. 15, No. 2. – pp. 219–228.
3. Little, J.D.C. A Proof for the Queueing Formula: $L = \lambda W$. – Operations Research, 1961. – Vol. 9, No. 3. – pp. 383–387.
4. Erlang, A.K. The Theory of Probabilities and Telephone Conversations. – Nyt Tidsskrift for Matematik, 1909. – Vol. 20. – pp. 33–39.
5. Kleinrock, L. Queueing Systems, Volume 1: Theory. – John Wiley & Sons, 1975. – 417 p.
6. Альохін, Ю.П. Теорія масового обслуговування. – К.: Вища школа, 2002. – 232 с.
8. Ткаченко, С.В. Інформаційні системи та технології: Навчальний посібник. – Київ: Центр навчальної літератури, 2005. – 412 с.
9. Berezner, M. Mathematical Models in Queueing Theory. – Springer, 2017. – 220 p.
10. Allen, A.O. Probability, Statistics, and Queueing Theory with Computer Science Applications. – Academic Press, 1990. – 401 p.
11. Recommendations systems: [Електронний ресурс] – режим доступу до ресурсу: <http://infolab.stanford.edu/~ullman/mmds/ch9.pdf>
12. Authors: Alex Smola and S.V.N. Vishwanathan – Machine Learning [Електронний ресурс] – режим доступу до ресурсу: <https://alex.smola.org/drafts/thebook.pdf>

13. Authors: N. Chandra Sekhar Reddy, K. Sai Prasad and A. Mounika – Classification Algorithm [Електронний ресурс] – режим доступу до ресурсу: https://www.ripublication.com/ijcir17/ijcirv13n8_23.pdf

14. Алгоритм регресії [Електронний ресурс] – режим доступу до ресурсу: https://cs.stanford.edu/~ermon/cs325/slides/ml_linear_reg.pdf

15. Google Analytics API Documentation [Електронний ресурс] – режим доступу до ресурсу: <https://developers.google.com/analytics/devguides>

16. Python Queue Libraries Documentation [Електронний ресурс] – режим доступу до ресурсу: <https://docs.python.org/3/library/queue.html>

17. Harrison, P.G. Performance Modeling of Computer Systems. – Cambridge University Press, 2003. – 408 p.

18. Грузінов, О.В. Математичне моделювання систем масового обслуговування. – Дисертація на здобуття наукового ступеня кандидата технічних наук. – Харків, 2010. – 164 с.

19. International Journal of Queueing Theory and Its Applications – [Електронний ресурс] – режим доступу до ресурсу: <https://www.inderscience.com/jhome.php?jcode=ijqta>

20. Постороніна, О.М. Моделі та методи оптимізації систем масового обслуговування. – Реферат на здобуття наукового ступеня магістра. – Київ, 2020. – 42 с.