Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Метод асимптотичної оцінки продуктивності комп'ютерних систем

(тема)

Виконав:
здобувач 2 року навчання,
групи СПм-23-3
Віталій ФЕДОРОВ
(власне ім’я, прізвище)

Спеціальність
123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма
Системне програмування
(повна назва освітньої програми)

Керівник: проф. Валерій ГОРБАЧОВ
(посада, власне ім’я, прізвище)

Допускається до захисту

Зав. кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“ ____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачу Федорову Віталію Миколайовичу
(прізвище, ім’я, по батькові)

1. Тема роботи Метод асимптотичної оцінки продуктивності комп'ютерних систем

затверджена наказом по університету від “ 21 ” квітня 2025 р. № 296Ст

2. Термін подання студентом роботи до екзаменаційної комісії 16 червня 2025 р.

3. Вхідні дані до роботи

1) теорія систем масового обслуговування;
2) операційний аналіз;
3) система імітаційного моделювання GPSS.

4. Перелік питань, що потрібно опрацювати у роботі

1) аналіз проблеми вузьких місць;
2) аналіз методів обмежування;
3) методи аналізу продуктивності мережевих систем;
4) експериментальний аналіз продуктивності мережевих системи на основі вузьких місць.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Слайд-презентація — 16 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Цілі проекту	22.04.25-29.04.25	
2	Аналіз проблеми та огляд літератури	30.04.25-05.05.25	
3	Визначення проблеми	06.05.25-09.05.25	
4	Аналіз методів	22.05.25-05.06.25	
5	Реалізація завдання	03.06.25-05.06.25	
6	Програмна реалізація	06.06.25-09.06.25	
7	Підготовка кваліфікаційної роботи	15.06.25-16.06.25	
8	Підготовка презентації	10.06.25-12.06.25	

Дата видачі завдання 21 квітня 2025 р

Здобувач _____ (підпис)

Керівник роботи _____ (підпис) проф. Валерій ГОРБАЧОВ (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 77 с., 9 рис., 3 табл., 2 дод., 15 джерел.

ЗБАЛАНСОВАНА СИСТЕМА, ВУЗЬКІ МІСЦЯ, ОПЕРАЦІЙНИЙ АНАЛІЗ, ОЦІНКА ПРОДУКТИВНОСТІ, МОДЕЛЮВАННЯ ПРОДУКТИВНОСТІ, МЕРЕЖІ ЧЕРГИ.

Метою кваліфікаційної роботи є розгляд різних визначень вузьких місць і методів їх виявлення. Використання операційного аналізу як математична основи дослідження.

У ході виконання кваліфікаційної роботи були розглянуті асимптотичні методи виявлення вузьких місць. Була проведена експериментальна оцінка продуктивності системи з метою аналізу вузьких місць у мережі з використання аналітичного і імітаційного моделювання. Аналіз аналітичних моделей продуктивності замкнутих систем при екстремальних навантаженнях показав зону точних значень фактичної пропускної здатності та часу відгуку. Результати аналітичного і імітаційного продемонстрували задовільну розбіжність.

ABSTRACT

Master’s thesis: 77 pages, 9 figures, 3 tables, 2 appendices, 15 sources.

BALANCED SYSTEM, BOTTLEPOCKETS, OPERATIONAL ANALYSIS, PERFORMANCE ASSESSMENT, PERFORMANCE MODELING, QUEUE NETWORKS.

The purpose of the qualification work is to consider various definitions of bottlenecks and methods for their detection. Using operational analysis as a mathematical basis for the study.

During the qualification work, asymptotic methods for detecting bottlenecks were considered. An experimental evaluation of the system performance was carried out to analyze bottlenecks in the network using analytical and simulation modeling. Analysis of analytical models of the performance of closed-loop systems under extreme loads showed a zone of accurate values of actual throughput and response time. The analytical and simulation results demonstrated a satisfactory discrepancy.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

API – прикладний програмний інтерфейс (англ., Application programming interface)

DM – розробка даних (англ., Data Mining)

AHP – процес аналітичної ієрархії (англ. Analytic hierarchy process)

DES – дискретно-подійна система (англ. discrete-event system)

OA – операційний аналіз (англ. Operational Analysis)

ISP – інтернет-провайдер (англ., Internet Provider)

ВСТУП

Моделі мереж масового обслуговування зарекомендували себе як ефективні інструменти для аналізу сучасних комп'ютерних систем. У роботі представлені основні результати з використанням операційного підходу, який дозволяє аналітику перевірити, чи виконується кожне припущення в конкретній системі. Спочатку описується природа моделей мереж масового обслуговування та їх застосування для обчислення і прогнозування показників продуктивності Визначаються основні показники продуктивності, такі як коефіцієнт використання, середня довжина черги і середній час очікування, а також встановлюються операційні залежності між ними Після цього вводиться поняття балансу потоку завдань, яке використовується для вивчення асимптотичних значень пропускної спроможності і часу очікування. Поняття балансу переходів між станами, однокрокової поведінки та однорідності використовуються для того, щоб пов'язати частки часу, які займає кожен стан системи, з параметрами попиту на роботу та характеристиками пристрою, а також описано ефективні методи обчислення основних показників продуктивності. Нарешті, концепція декомпозиції використовується для спрощення аналізу шляхом заміни підсистем еквівалентними пристроями. Для всіх концепцій наведено приклади.

1 ОГЛЯД ПРОБЛЕМИ ТА МЕТА РОБОТИ

1.1 Аналіз оцінки продуктивності складної системи

Кваліфікаційна робота присвячена найпростішому підходу до аналізу складної системи з використанням мережі масового обслуговування: граничний аналіз. З дуже невеликими обчисленнями можливо визначити верхню та нижню межі пропускної здатності та відгуку системи, як функція інтенсивності робочого навантаження системи (кількість або швидкість прибуття).

Розглядаються методи обчислення двох класів меж продуктивності: асимптотичних меж і меж збалансованої системи. Асимптотичні межі справедливі для ширшого класу систем, ніж для збалансованої мережевої системи. Перевага збалансованої межі системи полягає в тому, що вони більш жорсткі, і, отже, забезпечують більше точну інформацію, ніж асимптотичні межі [5].

Розвиток цих методів дає цінну інформацію про основні фактори, що впливають на продуктивність складних систем. Зокрема, підкреслюється критичний вплив вузького місця системи. Межі можна обчислити швидко, тому аналіз підходить як техніка моделювання першого кроку, яка може використовуватися для усунення неадекватних альтернатив на ранній стадії аналізу.

Аналітичні моделі характеризують продуктивність реальних складних систем протягом заданих періодів часу. Для цього використовується математичний апарат, за допомогою якого ми можемо визначати формальні змінні, формулювати гіпотези та доводити теореми.

Теорія стохастичних процесів традиційно використовується як така основа. Більшість аналізів продуктивності починаються з стохастичної гіпотези і стохастичної моделі.

Стохастична гіпотеза: Поведінка реальної системи протягом певного періоду часу характеризується розподілом ймовірностей стохастичного процесу.

Зазвичай також висуваються додаткові гіпотези. Ці гіпотези, які стосуються природи стохастичного процесу, зазвичай вводять такі поняття, як стаціонарний стан, ергодичність, незалежність і розподіл конкретних випадкових величин. Всі ці гіпотези складають стохастичну модель.

Спостережувані аспекти реальної системи - наприклад, стани, параметри та розподіли ймовірностей - можуть бути ідентифіковані з величинами в стохастичній моделі, і рівняння, що пов'язують ці величини, можуть бути виведені. Хоча формально ці рівняння застосовні лише до стохастичного процесу, вони також можуть бути застосовані до спостережуваної поведінки самої системи за відповідних обмежувальних умов [5].

Стохастичні моделі дають багато переваг. Незалежні та залежні змінні можуть бути точно визначені, гіпотези можуть бути сформульовані лаконічно, а під час аналізу можна звертатися до значного масиву теорії. Однак стохастичне моделювання має певні недоліки, найголовнішим з яких є неможливість перевірки стохастичної гіпотези та додаткових гіпотез, які від неї залежать.

Стохастична гіпотеза - це твердження про причини, що лежать в основі поведінки реальної системи. Оскільки неможливо довести заявлені причини, вивчаючи спостережувані ефекти, істинність або хибність стохастичної гіпотези та її залежних додаткових гіпотез - для даної системи і періоду часу - ніколи не може бути безсумнівно встановлена за допомогою будь-яких вимірювань. Це справедливо, навіть якщо припустити, що похибка вимірювання дорівнює нулю і що всі можливі вимірювання були проведені.

Таким чином, аналітик ніколи не може бути впевненим, що рівняння, отримане на основі стохастичної моделі, може бути правильно застосоване до поведінки реальної системи, яку можна спостерігати.

1.2 Визначення вузьких місць складних систем

Вузькими місцями прийнято вважати певні ресурси або сервіси, які суттєво обмежують продуктивність складної системи. Для різних вимог застосування і різних способів роботи в літературі можна знайти численні визначення того, що сприяє виникненню вузьких місць. Однак, досі не існує єдиного визначення вузьких місць. Нижче наведено кілька основних визначень: ресурс, потужність якого менше вимоги, які до нього застосовуються; процес, який обмежує пропускну здатність і т.і.

З цих визначень ми бачимо різноманітність вузьких місць. Вони не лише спричинені фізичними обмеженнями, такими як ресурси, процеси, обладнання тощо, але й залежать від функції, оператора тощо. Деякі вузькі місця можуть з'являтися тимчасово, а деякі можуть залишатися статичними. У загальному розумінні вузьке місце - це "щось", що обмежує продуктивність системи. Але з різних точок зору вузькі місця не є ідентичними. У цьому розділі будуть представлені різні визначення вузьких місць. Ми класифікуємо ці визначення на дві категорії: на основі ПО (продуктивности обслуговування) та на основі взаємного впливу вузьких місць, з метою надання практичних рекомендацій з точки зору перспективи застосування.

1.2.1 Визначення на основі ПО (продуктивности обслуговування)

Визначення на основі ПО (продуктивности обслуговування) визначають вузькі місця системи відповідно до вимірювання продуктивності системи. У визначеннях ПО вимірювання важливими результатами є показники середнього часу очікування та коефіцієнта використання.

Вимірювання середнього часу очікування. При вимірюванні середнього часу очікування вузьким місцем вважається вузол системи з найбільшим середнім часом очікування [4].

$$B = \{i \vee W_i = max(W_1, W_2, \ldots, W_n)\} \qquad (1.1)$$

У рівнянні (1.1) W_i - це середній час очікування продукції в і-му вузлі. Для закону Літтла вимірювання середньої довжини черги також належить до цієї категорії. Цей метод підходить для аналізу мереж з необмеженими проміжними буферами. Для систем з обмеженими буферами та систем без буферів він не підходить. Якщо кілька пристроїв мають однаковий найбільший час очікування, цей метод не може визначити унікальне вузьке місце.

Вимірювання коефіцієнта використання. Пристрій з найбільшим коефіцієнтом використання та простою вважається вузьким місцем [13], при цьому використовується метод вимірювання коефіцієнта використання.

$$B = \{i \vee \rho_i = max(\rho_1, \rho_2, \ldots, \rho_n)\} \qquad (1.2)$$

У рівнянні (2), коефіцієнт використання і-го пристрою. $\rho_i = \lambda_i/\mu_i$, де $\lambda_i\mu_i$ - це інтенсивність надходження запитів та інтенсивність обслуговування і-го запиту відповідно. Оскільки кілька пристрої можуть мати однакове робоче навантаження, різниця між коефіцієнтами використання машин може бути дуже малою. Хоча цей метод легко автоматизувати, він може призвести до появи численних вузьких місць. Метод виявлення вузьких місць досліджує всі можливі комбінації вузьких місць, який швидко ускладнювався для великих систем.

1.2.2 Визначення на основі чутливості системи

Інший спосіб визначити вузьке місце - знайти пристрій, продуктивність якого найбільше впливає на загальну продуктивність системи. Для оцінки

чутливості продуктивності системи до змін параметрів пристроїв використаємо метод вимірювання.

Виробниче вузьке місце. В роботі [5] використано підхід теорії систем для визначення чутливості пропускної здатності машини до пропускної здатності системи. Вони вивчали цю проблему на прикладі марковської виробничої лінії. Продуктивність - це середня кількість деталей, вироблених останнім верстатом, і вона є функцією всіх параметрів верстатів і буферів:

$$\vec{PR} = \vec{PR}(p_1, r_1, \ldots, p_m, r_m, N_1, \ldots, N_{m-1}, c_1, \ldots, c_m)\,, \qquad (1.3)$$

де N_1 - розмір буферу перед і-м пристроєм;

c_i - час обробки.

Час обробки та час простою кожної машини є випадковими величинами, розподіленими за експоненціальним законом з параметрами p_1, r_1 відповідно. Представлено три типи вузьких місць. Визначення *вузького місця за часом безвідмовної роботи (UT-BN)* було дано в [10]. Якщо

$$\frac{\partial PR}{\partial T_{up_i}} > \frac{\partial PR}{\partial T_{up_j}}, j \neq i \qquad (1.4)$$

де m_i є вузьким місцем за часом роботи (UT-BN).

Вони також дали визначення простою вузького місця в часі (DT-BN): Якщо

$$\left|\frac{\partial PR}{\partial T_{down_i}}\right| > \left|\frac{\partial PR}{\partial T_{down_j}}\right|, j \neq i \qquad (1.5)$$

У (1.5) m_i - це вузьке місце за часом простою (DT-BN). Тут використовуються абсолютні значення, оскільки $\partial PR/\partial T_{down_i}$ є від'ємною

величиною: збільшення T_{down} призводить до зменшення PR. Машина m_i є вузьким місцем (BN), якщо вона є одночасно UT-BN та DT-BN. Визначення вузьких місць на основі чутливості до часу циклу машин було наведено в [11]. Машина є c-вузьким місцем, якщо

$$\frac{\partial \vec{PR}}{\partial c_i} > \frac{\partial \vec{PR}}{\partial c_j}, j \neq i \qquad (1.6)$$

Тоді m_i визначається як c-вузьке місце (c-BN). Крім UT-BN, DT-BN та c-BN, в [24] наведено ще одне визначення вузького місця, яке базується на чутливості до обсягу виробництва. Верстат є вузьким місцем, якщо чутливість показника продуктивності системи до його продуктивності в ізольованому стані є найбільшою, порівняно з усіма іншими верстатами. m_i є вузьким місцем, якщо:

$$\frac{\partial \vec{PR}(p_1,\ldots,p_m,N_1,\ldots,N_m,)}{\partial p_i} > \frac{\partial \vec{PR}(p_1,\ldots,p_m,N_1,\ldots,N_m,)}{\partial p_j}, \forall i \neq j \qquad (1.7)$$

У (1.7) p_i це продуктивність *i-го* пристрою ізольовано. Зауважте, що ці визначення не є взаємовиключними, і що конкретний робочий центр може задовольняти одному або декільком з них у будь-який момент часу.

Досі ми обговорювали різні визначення вузьких місць. У наступному розділі ми представимо методи виявлення вузьких місць, засновані на цих визначеннях.

1.3 Методи аналізу ефективності

Мережі масового обслуговування широко використовуються для аналізу продуктивності багатопрограмних комп'ютерних систем.

Результати оцінки продуктивності складних систем з використанням аналітичних моделей мереж масового обслуговування. Традиційний підхід до їх отримання залежить від низки припущень, що використовуються в теорії стохастичних процесів:

1) система моделюється стаціонарним стохастичним процесом;

2) робочі місця стохастично незалежні;

3) кроки завдання від пристрою до пристрою слідують за ланцюжком Маркова;

4) система знаходиться у стохастичній рівновазі.

Теорія мереж масового обслуговування, що базується на цих припущеннях, зазвичай називається "марковською теорією мереж масового обслуговування" [8]. Слова, виділені курсивом у цьому списку припущень, ілюструють концепції, які аналітик повинен розуміти, щоб мати змогу розгортати моделі. Деякі з цих понять є складними. Деякі, такі як "рівновага" або "стаціонарність", не можуть бути доведені шляхом спостереження за системою протягом обмеженого періоду часу. Насправді, більшість з них можна спростувати емпірично - наприклад, параметри змінюються з часом, робочі місця є залежними, переходи від пристрою до пристрою не відповідають ланцюгам Маркова, системи можна спостерігати лише протягом коротких періодів, розподіл послуг рідко буває експоненціальним. Не дивно, що багато людей дивуються, що ці моделі так добре застосовуються до систем, які порушують так багато припущень аналізу!

Застосовуючи або перевіряючи результати теорії марковських мереж масового обслуговування, аналітики замінюють операційні (тобто безпосередньо виміряні} значення на стохастичні параметри в рівняннях. Неодноразові успіхи валідації спонукали нас дослідити, чи можуть традиційні рівняння теорії марковських мереж масового обслуговування також бути зв'язками між операційними змінними, і якщо так, то чи можуть вони бути виведені з використанням різних припущень, які можна безпосередньо перевірити і які, ймовірно, будуть мати місце в реальних

системах.

У роботі описано операційний підхід до моделювання мереж масового обслуговування. Всі основні рівняння та результати отримані на основі одного або декількох з трьох операційних принципів.

Всі величини повинні бути визначені таким чином, щоб їх можна було точно виміряти, а всі припущення сформульовані таким чином, щоб їх можна було безпосередньо перевірити. Достовірність результатів повинна залежати лише від припущень, які можна перевірити, спостерігаючи за реальною системою протягом обмеженого періоду часу.

Система повинна бути збалансована по потоку - тобто кількість прибуття на певний пристрій має бути (майже) такою ж, як і кількість відправлень з цього пристрою протягом періоду спостереження.

Пристрої повинні бути однорідними - тобто маршрутизація завдань не повинна залежати від довжини локальної черги, а середній час між завершенням обслуговування на даному пристрої не повинен залежати від довжини черги на інших пристроях.

Ці операційні принципи, які будуть детально розглянуті в наступних розділах, призводять до тих самих математичних рівнянь, що й традиційні марковські припущення. Однак операційні припущення можна перевірити, і є вагомі підстави вважати, що вони часто справджуються. Саме тому операційний аналіз мереж масового обслуговування пояснює успіх валідаційних експериментів. Тепер стало можливим використовувати технологію мереж масового обслуговування з набагато більшою впевненістю і розумінням.

На відміну від розглянутих тут методів обмеження, більш складні методи аналізу, представлені в наступних розділах, вимагають значно більшого обсягу обчислень - до такої міри, що їх неможливо виконати вручну.

1.4 Цілі роботи

Робота має чотири основні цілі, які пов'язані між собою і випливають одна з одної:

1) аналіз асимптотичних підходів для оцінювання продуктивності мережі;

2) операційний аналіз для оцінки продуктивності мережі;

3) аналіз методів виявлення для розпізнавання вузьких місць у комп'ютерних мережах;

4) експериментальна оцінка підходів для розпізнавання вузьких місць і розрахунку, прогнозування показників продуктивності в комп'ютерних мережах.

2 МЕТОДОЛОГІЯ ПРОЕКТУВАННЯ ПРОДУКТИВНОСТІ

2.1 Вступ

Основне питання: як можна планувати, проектувати, розробляти, розгортати і експлуатувати ІТ-послуги, які відповідають постійно зростаючим вимогам до продуктивності, доступності, надійності, безпеки і вартості? Або, якщо бути більш конкретним, чи є дана ІТ-система належним чином спроектованою і розрахованою на певні умови навантаження? Чи може система управління страховими відшкодуваннями задовольнити вимогу щодо часу відгуку в межах секунди? Чи є інфраструктура державної установи масштабованою і чи може вона впоратися з новими політиками безпеки в Інтернеті, необхідними для фінансових транзакцій? Чи можна впровадити механізми безпеки без шкоди для сприйняття користувачами? Чи здатна система бронювання круїзних лайнерів реагувати на очікуваний пік споживчих запитів, що виникає після рекламної кампанії на телебаченні?

Розбиваючи складність ІТ-системи на складові, можна проаналізувати функціональність кожного компонента, оцінити вимоги до сервісів, спроектувати та експлуатувати системи, які відповідатимуть очікуванням користувачів. Іншими словами, відповідь на вищезазначені питання вимагає глибокого розуміння архітектури системи та її інфраструктури. У цьому розділі представлено основні кроки методології інженерії продуктивності, необхідні для задоволення основних потреб у продуктивності ІТ-сервісів.

2.2 Проектування продуктивності

У багатьох системних проектах вимоги до продуктивності розглядаються лише на завершальних етапах процесу розробки програмного забезпечення. Як наслідок, багато систем мають проблеми з продуктивністю,

які призводять до затримок і фінансових втрат для компаній і користувачів [8]. Інженерія продуктивності аналізує очікувані характеристики продуктивності системи на різних етапах її життєвого циклу. Інженерія продуктивності 1) розробляє практичні стратегії, які допомагають спрогнозувати рівень продуктивності, якого може досягти система, і 2) надає рекомендації для досягнення оптимального рівня продуктивності. Обидва завдання покладаються на наступні активні дії, які формують основу методології:

1) зрозуміти ключовий факт, який впливає на продуктивність системи;

2) виміряйте систему та зрозумійте її завантаженість;

3) розробіть і перевірте модель робочого навантаження, яка відображає ключові характеристики фактичного робочого навантаження;

4) розробити та перевірити аналітичну модель, яка точно прогнозує продуктивність системи;

5) використовуйте моделі для прогнозування та оптимізації продуктивності системи.

Інженерію продуктивності можна розглядати як сукупність методів для підтримки розробки систем, орієнтованих на продуктивність, протягом усього життєвого циклу [8]. Фази життєвого циклу системи визначають робочий процес, процедури, дії та методи, які використовуються аналітиками для створення та підтримки IT-систем. На рисунку 2.1 наведено огляд методології проектування та аналізу систем, що базується на продуктивності. Ця методологія базується на моделях, які використовуються для забезпечення гарантій якості обслуговування IT-систем. До таких моделей належать: модель робочого навантаження, модель продуктивності, модель доступності, модель надійності та модель витрат. На ранніх стадіях проекту інформація та дані, доступні для розробки моделей, в кращому випадку є приблизними.

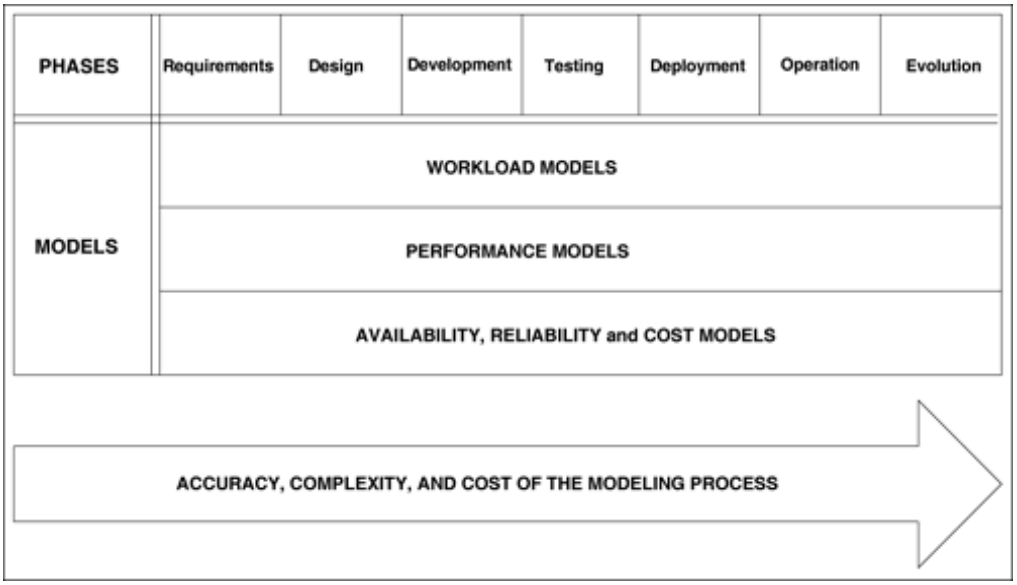

Рисунок 2.1 - Процес моделювання

Наприклад, на етапі аналізу вимог складно побудувати детальну модель робочого навантаження, яка б визначала вимоги, що висуваються транзакціями до ресурсів системи. Однак у міру розвитку проекту стає доступним більше інформації про систему, що зменшує похибку припущень моделі та підвищує довіру до оціночних прогнозів майбутньої продуктивності системи [9]. Таким чином, моделі робочого навантаження і продуктивності розвиваються пліч-о-пліч з системним проектом на всіх етапах його життєвого циклу. Протягом життєвого циклу системи можуть використовуватися різні типи моделей робочого навантаження та продуктивності. В основному, вони відрізняються за трьома аспектами: складність, точність і вартість. Рисунок 2.1 ілюструє спектр моделей для різних фаз, ранжованих від найменш складних до найбільш складних і дорогих. Очікується, що зі збільшенням складності та вартості уточнені моделі стають більш точними. У нижній частині діаграми можна використовувати розрахунки "за конвертом" як просту, легку у використанні методику з відносно низьким рівнем точності. У верхній частині діаграми використовуються детальні моделі працюючих систем, які є точними, але також дорогими і складними.

2.3 Методологія проектування продуктивності на основі моделей

У цьому розділі описано основні кроки, які використовуються для проектування та аналізу комп'ютерних систем з огляду на продуктивність. Методологія ґрунтується на моделях робочого навантаження та продуктивності і може бути використана на всіх етапах життєвого циклу системи. Інженерія продуктивності передбачає низку кроків, які слід виконувати систематично. На рисунку 2.2 наведено огляд основних кроків кількісного підходу до аналізу продуктивності системи.

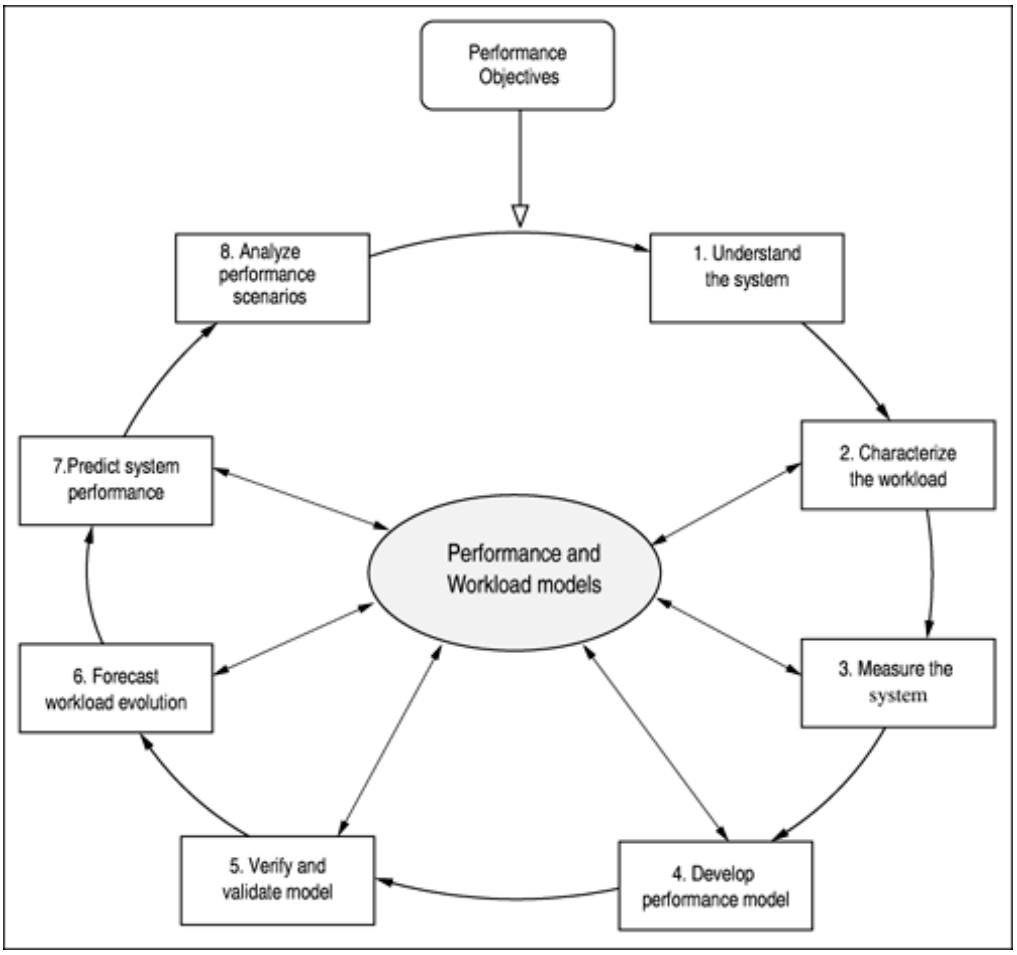

Рисунок 2.2 - Методологія інжинірингу продуктивності на основі моделей

Відправною точкою методології є визначення цілей продуктивності системи. Ці цілі повинні бути кількісно визначені як частина системних вимог. Цілі продуктивності використовуються для встановлення цілей рівня

обслуговування. Визначаються цілі рівня обслуговування, встановлюються бізнес-метрики та документуються цілі продуктивності системи. Після того, як система та її кількісні цілі визначені, можна перейти до циклу кількісного аналізу. Різні етапи циклу кількісного аналізу є наступними.

Розуміння системи. Перший крок - отримати глибоке розуміння архітектури системи та провести огляд на рівні архітектури з акцентом на продуктивність. Це означає відповісти на такі питання, як Які системні вимоги бізнес-моделі? Який тип програмного забезпечення (тобто операційна система, монітор транзакцій, СУБД, прикладне програмне забезпечення) буде використовуватися в системі? Цей крок дає систематичний опис архітектури системи, її компонентів та цілей. Це можливість проаналізувати питання продуктивності запропонованої архітектури.

Охарактеризуй робоче навантаження. На цьому кроці визначаються основні компоненти, які складають робоче навантаження. Вибір компонентів залежить від характеру системи та мети характеристики. Результатом цього кроку є твердження на кшталт "Досліджуване робоче навантаження складається з транзакцій електронного бізнесу, повідомлень електронної пошти та запитів на інтелектуальний аналіз даних". Продуктивність системи з великою кількістю клієнтів, серверів і мереж значною мірою залежить від характеристик її навантаження. Таким чином, життєво важливо у будь-якій роботі з проектування продуктивності чітко розуміти і точно характеризувати робоче навантаження [8,13]. Робоче навантаження системи можна визначити як сукупність усіх вхідних даних, які система отримує з навколишнього середовища протягом певного періоду часу. Наприклад, якщо досліджувана система є сервером бази даних, то її робоче навантаження складається з усіх транзакцій (наприклад, запитів, оновлень), оброблених сервером протягом інтервалу спостереження.

Виміряйте систему та отримайте параметри робочого навантаження. Третій крок передбачає вимірювання потужності системи та отримання значень параметрів моделі робочого навантаження. Вимірювання є ключовим

кроком для всіх завдань в інженерії продуктивності. Воно дозволяє зрозуміти параметри системи і встановити зв'язок між системою та її моделлю. Вимірювання продуктивності збираються з різних контрольних точок, ретельно обраних для спостереження і моніторингу досліджуваного середовища. Наприклад, уявімо, що за сервером бази даних спостерігають протягом 10 хвилин, і за цей час виконується 100 000 транзакцій. Робоче навантаження бази даних протягом цього 10-хвилинного періоду - це сукупність 100 000 транзакцій. Характеристики навантаження представлені набором інформації (наприклад, час надходження та завершення, процесорний час та кількість операцій I) для кожної з 100 000 транзакцій бази даних.

Розробка моделей продуктивності. На четвертому етапі кількісні методи та аналітичні (або імітаційні, або прототипові) моделі використовуються для розробки моделей ефективності систем. Моделі продуктивності використовуються для розуміння поведінки складних систем. Моделі використовуються для прогнозування продуктивності при зміні будь-якого аспекту робочого навантаження або архітектури системи. Прості моделі, засновані на операційному аналізі, є доступними для практиків програмної інженерії. Вони дають уявлення про те, як архітектурні рішення програмного забезпечення впливають на продуктивність [11].

Верифікація та валідація моделей. П'ятий крок спрямований на перевірку специфікацій моделі та валідацію результатів моделі. Цей крок застосовується як до моделей продуктивності, так і до моделей робочого навантаження. Модель продуктивності вважається валідованою, якщо показники продуктивності (наприклад, час відгуку, використання ресурсів, пропускна здатність), розраховані за допомогою моделі, збігаються з вимірами реальної системи в межах певної допустимої похибки. Як правило, використання ресурсів у межах 10%, пропускна здатність системи в межах 10% і час відгуку в межах 20% вважаються прийнятними [14]. Модель вважається верифікованою, якщо її результати є точним відображенням

продуктивності системи. Детальна інформація про методи калібрування ефективності доступна. У підсумку, на цьому кроці необхідно відповісти на такі питання, як Чи правильною є модель системи, що розглядається? Чи відображає модель поведінку критично важливих компонентів системи?

Прогнозування еволюції робочого навантаження. Більшість систем зазнають модифікацій та еволюції протягом свого життя. Зі зміною вимог до системи змінюються і робочі навантаження. Потреби зростають або зменшуються залежно від багатьох факторів, таких як функціональні можливості, що пропонуються користувачам, кількість користувачів, оновлення апаратного забезпечення або зміни в програмних компонентах. На шостому кроці прогнозується очікуване робоче навантаження на систему. Методи та стратегії прогнозування [12] змін робочого навантаження повинні давати відповіді на такі питання, як Яким буде середній розмір електронного листа до кінця наступного року. Якою буде кількість одночасних користувачів системи інтернет-банкінгу через шість місяців?

Прогнозування продуктивності системи. Рекомендації щодо продуктивності потрібні на кожному етапі життєвого циклу системи, оскільки кожне архітектурне рішення може потенційно створити бар'єри для досягнення цілей продуктивності системи. Таким чином, прогнозування продуктивності є ключовим у роботі з інженерії продуктивності, оскільки необхідно мати можливість визначити, як система буде реагувати на зміни в рівнях навантаження і поведінці користувачів або на нові програмні компоненти, інтегровані в систему. Для цього потрібні прогнозні моделі. Експерименти, як правило, не є життєздатними, оскільки виправлення дефектів продуктивності може вимагати структурних змін, які є дорогими. На сьомому кроці моделі продуктивності використовуються для прогнозування продуктивності системи за різних сценаріїв.

Аналізуйте сценарії продуктивності. Перевірені моделі продуктивності та робочого навантаження використовуються для прогнозування продуктивності системи за кількох різних сценаріїв, таких як модернізація

серверів, пришвидшення роботи мереж, зміни в робочому навантаженні системи, зміни в поведінці користувачів та зміни в програмному забезпеченні. Щоб допомогти знайти найбільш економічно ефективну архітектуру системи, на цьому кроці аналізуються різні сценарії. По суті, кожен сценарій складається з майбутньої функції системи та/або прогнозу робочого навантаження. Оскільки кожен елемент прогнозу несе в собі певний ступінь невизначеності, розглядається декілька можливих майбутніх сценаріїв. Аналізуються різні можливі системні архітектури. Створюється набір альтернатив, щоб системні інженери могли вибрати найбільш підходящий варіант з точки зору співвідношення витрат і вигод.

Дві моделі є центральними компонентами методології: модель робочого навантаження та модель продуктивності системи. Моделі робочого навантаження детально розглядаються в цьому розділі.

2.4 Модель робочого навантаження

Модель робочого навантаження - це представлення, яке імітує реальне робоче навантаження, що вивчається. Це може бути набір програм, написаних і реалізованих з метою штучного тестування системи в контрольованому середовищі. Модель робочого навантаження може також слугувати вхідними даними для аналітичної моделі системи непрактично мати модель, що складається з тисяч базових компонентів, щоб імітувати реальне робоче навантаження. Моделі робочого навантаження повинні бути компактними і репрезентативними щодо реального робочого навантаження [14].

2.4.1 Типи моделей робочого навантаження

Моделі робочого навантаження можна розділити на дві основні категорії.

Натуральні моделі будуються або з використанням основних компонентів реального робочого навантаження як будівельних блоків, або з використанням трас виконання реального робочого навантаження. Природний тест складається з програм, витягнутих з реального робочого навантаження системи. Ці програми підібрані таким чином, що вони відображають загальне навантаження на систему в певні періоди часу. Іншою природною моделлю, яка часто використовується в дослідженнях продуктивності, є трасування робочого навантаження. Вона складається з хронологічної послідовності даних, що представляють конкретні події t, які відбулися в системі під час сеансу вимірювання. Наприклад, у випадку веб-сервера доступ до журналу містить один запис на кожен HTTP-запит, оброблений сервером. Серед іншої інформації, кожен запис журналу містить ім'я хоста, який зробив запит, мітку часу та ім'я збереженого файлу. Цей тип журналу характеризує реальне робоче навантаження протягом певного періоду часу. Хоча траси демонструють відтворюваність і репрезентативність, вони мають недоліки. Зазвичай, траси складаються з величезних обсягів даних, що збільшує складність їх використання в моделюванні. Зазвичай важко модифікувати трасу для відображення різних сценаріїв навантаження. Крім того, траси підходять лише для імітаційних або прототипних моделей.

Штучні моделі не використовують жодного базового компонента реального робочого навантаження. Натомість ці моделі будуються за допомогою спеціальних програм та описових параметрів. Штучні моделі поділяються на два класи: виконувані та невиконувані моделі. Виконувані штучні моделі складаються з набору програм, спеціально написаних для експериментів з певним аспектом комп'ютерної системи. До класу виконуваних моделей належать такі робочі навантаження, як набори інструкцій, ядра, синтетичні програми, штучні бенчмарки та драйвери. Інструкційні мікси - це апаратні демонстраційні програми, призначені для тестування швидкості комп'ютера на простих обчислювальних операціях та

операціях вводу/виводу. Ядра програм - це фрагменти коду, вибрані з інтенсивних в обчислювальному плані частин реальної програми. Загалом, ядра зосереджені на вимірюванні продуктивності процесорів без урахування системи вводу/виводу. Синтетичні програми - це спеціально розроблені коди, які ставлять вимоги до різних ресурсів обчислювальної системи. На відміну від бенчмарків, синтетичні програми не схожі на реальне робоче навантаження. Бенчмарки, синтетичні програми та інші форми виконуваних моделей не є адекватними вхідними даними для моделей продуктивності. Коли продуктивність системи аналізується за допомогою аналітичних або імітаційних моделей, потрібні не виконувані представлення робочого навантаження. Оскільки підхід до інженерії продуктивності ґрунтується на використанні аналітичних моделей для прогнозування продуктивності, ця книга зосереджується на представленнях робочого навантаження, придатних для таких моделей.

Моделі невиконуваного навантаження описуються набором середніх значень параметрів, які відтворюють таке ж використання ресурсів, як і реальне навантаження. Кожен патер позначає аспект поведінки виконання базового компонента на досліджуваному темпі. Основними вхідними даними для аналітичних моделей є параметри, що описують центри обслуговування (тобто апаратні та програмні ресурси) та клієнтів (наприклад, транзакції та запити). Типовими параметрами є час між прибуттям компонентів (наприклад, транзакції та запиту), вимоги служби, розміри компонентів, виконавчі мікси (класи компонентів і відповідні їм рівні мультипрограмування).

Кожен тип системи може характеризуватися різним набором параметрів. Як приклад, розглянемо параметричну характеристику робочого навантаження файлового сервера розподіленої системи [5]. На продуктивність файлового сервера безпосередньо впливають декілька факторів: завантаження системи, можливості пристроїв та локальність посилань на файли. На основі цих факторів визначаються наступні

параметри:

Розподіл частоти запитів: описує участь кожного запиту (наприклад, читання, запис, створення, перейменування) у загальному робочому навантаженні.

Розподіл часу між надходженнями запитів: показує час між послідовними запитами. Він також вказує на інтенсивність завантаження системи.

Поведінка посилань на файли: описує відсоток звернень до кожного файлу у дисковій підсистемі.

Розмір читання і запису: вказує на навантаження на вхід/вихід. Цей параметр сильно впливає на час, необхідний для обслуговування запиту.

Вищевказані параметри визначають модель робочого навантаження і здатні керувати синтетичними програмами, які точно відображають реальне робоче навантаження.

В іншому дослідженні [10] робоче навантаження вводу/виводу розглядається під іншим кутом зору. Атрибути часу доступу фіксують структуру доступу в часі, яка включає процес надходження (детермінований, пуассонівський, стрибкоподібний), інтенсивність стрибків, кількість стрибків та частку стрибків. Атрибути просторової локалізації визначають зв'язок між послідовними запитами, наприклад, послідовним і випадковим доступом. Через затримки "пошуку" робочі навантаження з більшою часткою послідовних звернень, як правило, демонструють кращу продуктивність, ніж навантаження з випадковими доступом. Таким чином, для детальної моделі пристрою вводу/виводу важливо враховувати атрибути просторової локалізації робочого навантаження.

2.4.2 Кластерний аналіз

Розглянемо робоче навантаження, яке складається з транзакцій, що демонструють велику варіативність з точки зору їхніх вимог до процесора та

вводу/виводу. Усереднення вимог всіх транзакцій, швидше за все, призведе до створення моделі навантаження, яка не буде репрезентативною для більшості транзакцій. Тому транзакції в робочому навантаженні повинні бути згруповані, або кластеризовані, таким чином, щоб варіабельність всередині кожної групи була відносно невеликою в порівнянні з варіабельністю всього набору даних. Ці кластери відповідають різним класам багатокласової моделі продуктивності.

Процес створення цих відносно однорідних груп називається *кластерним аналізом*. Хоча кластерний аналіз може бути виконаний автоматично за допомогою спеціальних функцій програмних пакетів (наприклад, SPSS, SAS, WEKA), аналітик ефективності повинен знати основи цієї техніки.

Тут представлено добре відому техніку кластеризації, яка називається кластеризацією за k-середнім. Припустимо, що робоче навантаження описується набором p точок $w_i = (D_{i1}, D_{i2}, ..., D_{iM})$ в M-вимірному просторі, де кожна точка w_i представляє одиницю роботи (наприклад, транзакцію, запит), що виконується комп'ютерною системою. Компоненти точки w_i описують специфічні властивості транзакції, такі як запит на обслуговування на певному пристрої або будь-яке інше значення (наприклад, кількість операцій вводу/виводу, переданих байт), з якого можна отримати запит на обслуговування.

Багато алгоритмів кластеризації, включаючи *k*-середні, вимагають визначення метрики відстані між точками. Поширеною метрикою відстані є *евклідова відстань*, яка визначає відстань $d_{i,j}$ між двома точками $w_i = (D_{i1}, D_{i2}, \cdots, D_{iM})$ та $w_j = (D_{j1}, D_{j2}, \cdots, D_{jM})$ як рівняння

$$d_{i,j} = \sqrt{\sum_{n=1}^{M} (D_{i_n} - D_{j_n})^2}$$

Використання необроблених даних для обчислення евклідових відстаней може призвести до викривлень, якщо компоненти точки мають дуже різні відносні значення та діапазони. Наприклад, припустимо, що робоче навантаження веб-сервера описується точками, кожна з яких представляє HTTP-запит, типу (*f, c*), де *f* - розмір файлу, отриманого за допомогою HTTP-запиту, а *c* - процесорний час, необхідний для обробки запиту. Розглянемо точку (25, 30), де *f* вимірюється в Кбайтах, а *c* - в мілісекундах. Якщо *f* тепер вимірюється в мегабайтах, точка (25, 30) перетвориться на (0.025,30). Такі зміни одиниць виміру можуть призвести до зміни відносних відстаней між групами точок. Для мінімізації проблем, що виникають через вибір одиниць виміру та різні діапазони значень параметрів, слід використовувати методи масштабування [12].

Алгоритм *k* -середніх створює k кластерів. Кожен кластер представлений своїм центроїдом, тобто точкою, координати якої отримані шляхом обчислення середнього значення координат всіх точок кластера. Алгоритм починає з визначення *k* точок у робочому навантаженні, які виступають в якості початкових оцінок центроїдів *k* кластерів. Точки, що залишилися, розподіляються між кластерами з найближчим центроїдом. Процедура розподілу повторюється кілька разів над вхідними точками до тих пір, поки жодна точка не перейде до іншого кластера або поки не буде виконано максимальну кількість ітерацій. Маючи в якості вхідних даних *p* точок $w_i = (D_{i1}, {}_{i2}, \cdots, D_{iM})$ кроки, необхідні для алгоритму *k*-середніх, є такими.

Крок 1. Встановити кількість кластерів рівною *k*.

Крок 2. Виберіть *k* початкових точок, які будуть використовуватися як початкові оцінки центроїдів кластерів. Наприклад, можна вибрати або перші *k* точок вибірки, або *k* точок, що знаходяться на найбільшій відстані одна від одної. У цьому випадку потрібно обчислити всі відстані між точками.

Крок 3. Проаналізуйте кожну точку навантаження і віднесіть її до кластера, центроїд якого знаходиться найближче. Положення центроїда

перераховується щоразу, коли до кластера додається нова точка.

Крок 4. Повторюйте крок 3 до тих пір, поки жодна точка не змінить свою кластерну приналежність під час повного проходу або поки не буде виконано максимальну кількість проходів.

Іншим алгоритмом кластеризації на основі відстані є алгоритм мінімального остовного дерева, описаний в [13]. Кластеризація робочих навантажень електронної комерції на основі фракталів розглядається в [14].

2.5 Моделі ефективності

Моделювання продуктивності є ключовим методом для розуміння проблем в IT-системах. Оскільки важко оцінити продуктивність, IT-системи повинні розроблятися з урахуванням рівня обслуговування. Іншими словами, розробник IT-сервісу повинен *anpiopi* знати межі системи. Наприклад, розробник повинен знати максимальну кількість транзакцій за секунду, яку система здатна обробити (тобто верхню межу пропускної здатності), або мінімальний час відгуку, який може бути досягнутий системою обробки транзакцій (тобто нижню межу часу відгуку).

Аналітичні моделі продуктивності охоплюють фундаментальні аспекти комп'ютерної системи і пов'язують їх один з одним за допомогою математичних формул та/або обчислювальних алгоритмів. В основному, аналітичні моделі продуктивності вимагають вхідної інформації, такої як інтенсивність робочого навантаження (наприклад, швидкість прибуття, кількість клієнтів і час на роздуми) і попит на послуги, який основний компонент робочого навантаження пред'являє до кожного ресурсу системи. Кілька алгоритмів на основі мереж масового обслуговування для розв'язання відкритих і закритих моделей з декількома класами наведені в частині II цієї книги і реалізовані в робочих книгах MS Excel. Методи включають точні та наближені розв'язки. У багатьох випадках достатньо лише відносної або наближеної оцінки продуктивності. Простого знання того, що пропускна

здатність становить приблизно 120 т/с для одного варіанту системи і приблизно 300 т/с для іншого варіанту, є достатньою інформацією для вибору одного варіанту над іншим.

Детальні моделі продуктивності вимагають параметрів, які можна згрупувати в наступні категорії.

Параметри системи. Характеристики системи, що впливають на продуктивність, включають кількість виконуваних процесів, характер та інтенсивність міжкомпонентної взаємодії, розміри буферів, максимальну кількість підтримуваних потоків та мережеві протоколи.

Параметри ресурсів. Властивості ресурсу, які впливають на продуктивність, включають час пошуку диска, затримку, швидкість передачі, пропускну здатність мережі, затримку маршрутизатора та швидкість процесора.

Параметри робочого навантаження. Ці параметри виводяться з характеристики робочого навантаження і поділяються на дві підкатегорії.

Параметри інтенсивності робочого навантаження - вимірюють навантаження на систему, що виражається в кількості одиниць роботи, які претендують на системні ресурси. Прикладами можуть бути кількість сеансів, запущених за день, кількість транзакцій на день до сервера додатків, кількість операцій, що потребують захищеного з'єднання/сек, та кількість транзакцій з базою даних, виконаних за одиницю часу.

Параметри попиту на обслуговування робочого навантаження. Вказують загальну кількість часу обслуговування, необхідного для кожного базового компонента на кожному ресурсі. Приклади включають процесорний час транзакцій на сервері додатків, загальний час передачі відповідей з веб-сервера назад клієнту і загальний час вводу/виводу на сервері бази даних для функції запиту.

Основна мета аналізу та проектування IT-систем - гарантувати, що цілі продуктивності будуть досягнуті. IT-послуги є складними і можуть бути дуже дорогими. Важливо звести до мінімуму кількість здогадок, коли справа

доходить до проектування, впровадження та експлуатації систем, що надають ці послуги.

2.6 Заключні зауваження

У цій главі представлено основні концепції інженерії продуктивності. Описано мотивуючий приклад, пов'язаний з плануванням, розробкою та впровадженням програми для колл-центру. Наведено п'ять прикладів того, як міркування продуктивності повинні бути включені в аналіз, проектування, експлуатацію та еволюцію системи. Основне питання в інженерії продуктивності полягає в тому, як гарантувати, що стовбур буде відповідати своїм цілям продуктивності. Описано кроки, необхідні для здійснення інжинірингу продуктивності. Центральним питанням методології є моделі, необхідні для представлення робочого навантаження і продуктивності системи. Базовий набір практичної методології інжинірингу продуктивності включає наступне: вкажіть цілі продуктивності системи, розуміти поточне середовище, виконайте характеристику робочого навантаження та створіть модель робочого навантаження, розробити модель ефективності, перевірка та валідація моделі системи, яка складається з моделей робочого навантаження та продуктивності, прогнозування зростання навантаження, використовуйте модель системи для аналізу продуктивності різних системних архітектур і різних сценаріїв робочого навантаження, виберіть найкращу альтернативу (на основі прогнозів моделі ефективності), яка забезпечує найкраще співвідношення ціни та якості, задовольняючи при цьому визначені рівні обслуговування.

3 МЕЖІ ПРОДУКТИВНОСТІ МЕРЕЖЕВИХ СИСТЕМ

3.1 Аналіз методів оцінки меж продуктивності

Ця робота присвячена найпростішому корисному підходу до аналізу комп'ютерних систем з використанням моделей мереж масового обслуговування: граничному аналізу. За допомогою дуже невеликих обчислень можна визначити верхню та нижню межі пропускної здатності системи та часу відгуку як функції інтенсивності навантаження на систему (кількості або частоти прибуття клієнтів). Ми описуємо методи обчислення двох класів меж продуктивності: асимптотичні межі та межі збалансованої системи. Асимптотичні границі підходять для ширшого класу систем, ніж границі збалансованої системи. Їх також простіше обчислювати. Перевагою границь збалансованої системи є те, що вони є жорсткішими, а отже, надають більш точну інформацію, ніж методи асимптотичні границі.

Існує кілька особливостей методів асимптотичних границь, які роблять їх цікавими та корисними.

1) Розвиток цих методів дає цінну інформацію про основні фактори, що впливають на продуктивність комп'ютерних систем. Зокрема, виділено та кількісно оцінено критичний вплив "вузького місця" системи.

2) Граничні значення можна розрахувати швидко. Таким чином, аналіз граничних значень підходить як метод моделювання першого кроку, який можна використовувати для усунення неадекватних альтернатив на ранній стадії дослідження.

3) У багатьох випадках декілька альтернатив можуть розглядатися разом, і один граничний аналіз може надати корисну інформацію про них усіх.

На відміну від розглянутих тут методів обмеження, більш складні методи аналізу, представлені в наступних розділах, вимагають значно

більшого обсягу обчислень - до такої міри, що їх неможливо виконати вручну.

Методи асимптотичних границь є найбільш корисними в дослідженнях з визначення розміру системи. Такі дослідження передбачають досить довгострокове планування, а отже, часто ґрунтуються на попередніх оцінках характеристик системи. За такої неточності в знаннях про систему, швидке визначення меж може бути більш доцільним, ніж більш детальний аналіз, що призводить до конкретних оцінок показників ефективності. Дослідження розмірів системи зазвичай передбачають розгляд великої кількості конфігурацій-кандидатів. Часто один ресурс (наприклад, центральний процесор) є домінуючим, тому що решта системи може бути сконфігурована відповідно до потужності цього ресурсу. Аналіз обмежень дозволяє розглядати *як одну альтернативу* групу конфігурацій-кандидатів, які мають однаковий критичний ресурс, але відрізняються за структурою попиту в інших сервісних центрах.

Методика граничних значень також може бути використана для оцінки потенційного приросту продуктивності від альтернативних модернізацій існуючих систем. У Розділі 3.3 ми покажемо, як графіки обмежень можуть дати уявлення про те, наскільки потрібно зменшити попит на послуги в "вузькому місці", щоб досягти поставлених цілей продуктивності. (Попит на послуги в центрі може бути зменшений або шляхом перенесення частини роботи з центру, або шляхом заміни центру більш швидким пристроєм).

Наше обговорення граничного аналізу обмежується випадком одного класу. Існують узагальнення для багатьох класів, але вони не використовуються широко. Однією з причин цього є те, що методи граничного аналізу є найбільш корисними для дослідження пропускної спроможності центру вузького місця, для чого достатньо однокласових моделей. Крім того, основною привабливістю методів обмеження на практиці є їх простота, яка була б втрачена, якби в моделі було включено декілька класів.

Моделі, які ми розглядаємо в цій главі, можна описати за допомогою наступних параметрів:

K - кількість серверів;

D_{max} - найбільший час обслуговування в будь-якому окремому сервері;

D - сума усіх часів обслуговування у сервері;

тип класу клієнта (термінал (замкнута система) або транзакція (розікнута система));

Z - середній час на роздуми (якщо клас термінального типу).

Для моделей з транзакційним типом навантаження межі пропускної здатності вказують на максимальну швидкість прибуття запитів, яку може обробити система, тоді як межі часу відгуку відображають найбільший і найменший можливий час відгуку, з яким ці клієнти можуть зіткнутися в залежності від швидкості прибуття в систему. Для моделей з пакетним або термінальним типом навантаження, межі вказують на максимальну і мінімальну можливу пропускну здатність системи і час відгуку як функції від кількості клієнтів в системі. Ми називаємо верхню межу пропускної здатності та нижню межу часу відгуку оптимістичними межами (оскільки вони вказують на найкращу можливу продуктивність), а нижню межу пропускної здатності та верхню межу часу відгуку - песимістичними межами (оскільки вони вказують на найгіршу можливу продуктивність). Хоча в наступних розділах ми розглядаємо лише граничні значення пропускної здатності системи та часу відгуку, фундаментальні закони наступної глави можуть бути використані для перетворення їх в граничні значення інших показників ефективності, таких як пропускна здатність та використання центрів обслуговування.

3.2 Асимптотичні межі робочого навантаження

Асимптотичний аналіз границь дає оптимістичні та песимістичні оцінки пропускної здатності системи та часу відгуку в однокласових мережах

масового обслуговування. Як випливає з назви, вони отримані шляхом розгляду (асимптотично) екстремальних умов легких і важких навантажень. Достовірність граничних значень залежить лише від одного припущення: що середній час обслуговування клієнта в сервері не залежить від того, скільки інших клієнтів в даний момент перебуває в системі або в яких серверах обслуговування вони знаходяться.

Тип інформації, що надається асимптотичними границями, залежить від того, чи є робоче навантаження системи відкритим (транзакційний тип) або закритим (пакетний або термінальний тип). Ми почнемо з найпростішого випадку - з транзактного типу навантаження.

3.2.1 Робоче навантаження транзакного типу. Відкрита система

Для транзакційних навантажень межа пропускної здатності вказує на максимально можливу швидкість прибуття клієнтів, яку система може успішно обробити. Якщо кількість заявок перевищує цю межу, то в міру надходження заявок постійно зростає кількість необроблених клієнтів. Таким чином, в довгостроковій перспективі нова заявка буде чекати невизначено довго (оскільки на момент її надходження в черзі вже може бути будь-яка кількість інших заявок). У цьому випадку ми говоримо, що система є насиченою (saturated). Таким чином, межа пропускної здатності - це швидкість надходження, яка відокремлює її від насичення.

Ключем до визначення граничної пропускної здатності є закон використання: $U_k = X_k \cdot S_k$ для кожного серверу **k**. Якщо позначити інтенсивність прибуття до системи як λ , то $X_k = \lambda \cdot V_k$ і закон використання можна переписати як $U_k = \lambda \cdot D_k$, де D_k - попит на послуги в центрі k. Щоб визначити граничну пропускну здатність, ми просто відзначимо, що поки всі центри мають невикористану потужність (тобто, мають коефіцієнт використання менше одиниці), можна обслуговувати збільшений потік клієнтів. Однак, коли будь-який з центрів стає насиченим (тобто має

коефіцієнт використання одиницю), вся система стає насиченою, оскільки ніяке збільшення швидкості прибуття клієнтів не може бути успішно обслужено. Таким чином, межа пропускної здатності - це найменша інтенсивність прибуття λ_{sat} , при якій будь-який центр насичується. Очевидно, що центр, який насичується при найменшій інтенсивності прибуття, є вузьким сервером (вузьким місцем) - сервером з найбільшим часом обслуговування на сервері. Нехай max - індекс вузького серверу. Тоді

$$U_{\max}(\lambda) = \lambda D_{\max} \le 1 .$$

Таки чином

$$\lambda_{sat} = \frac{1}{D_{\max}} .$$

Таким чином, для частоти прибуття більшої або рівної $\frac{1}{D_{max}}$ система є насиченою, в той час як система здатна обробляти частоту прибуття меншу за

$$\frac{1}{D_{max}} .$$

Асимптотичні межі часу відгуку вказують на найбільший та найменший можливий час відгуку, з яким стикаються клієнти, коли інтенсивність прибуття до системи становить λ . Оскільки система є нестабільною, якщо $\lambda_{sat} > \lambda$, ми обмежимо наше дослідження випадком, коли інтенсивність надходження заявок менша за межу пропускної здатності. Існує дві екстремальні ситуації:

1) У найкращому випадку жоден запит ніколи не заважає іншим, так що затримки в черзі не виникають. У цьому випадку час відгуку системи для кожного запиту є просто сума його відвідувань (запитів на обслуговування), яку ми позначимо через *D*.

2) У найгіршому випадку n клієнтів прибувають разом кожні n / λ

одиниць часу (швидкість прибуття системи становить $\frac{n}{n\lambda} = \lambda$) . Клієнти в кінці партії змушені стояти в черзі за клієнтами на початку партії, і, таким чином, стикаються з великим часом відповіді. Зі збільшенням розміру партії n все більше і більше клієнтів чекають все довше і довше. Таким чином, для будь-якої постульованої песимістичної межі часу відгуку для швидкості прибуття системи λ можна підібрати розмір партії n, достатньо великий, щоб перевищити цю межу. Ми дійшли висновку, що не існує песимістичної межі для часу відгуку, незалежно від того, наскільки малим може бути коефіцієнт прибуття λ .

Ці результати є дещо незадовільними. На щастя, межі пропускної здатності та часу відгуку надають більше інформації у випадку закритих (пакетних та термінальних) типів навантаження.

3.2.2 Робоче навантаження термінального типу. Замкнута система

На рисунку 3.1 показано загальний вигляд асимптотичних границь пропускної здатності та часу відгуку для термінального навантаження. Межі вказують на те, що точні значення фактичної пропускної здатності та часу відгуку повинні лежати в заштрихованих частинах рисунків. Загальна форма і положення цих значень показані кривими на рисунку.

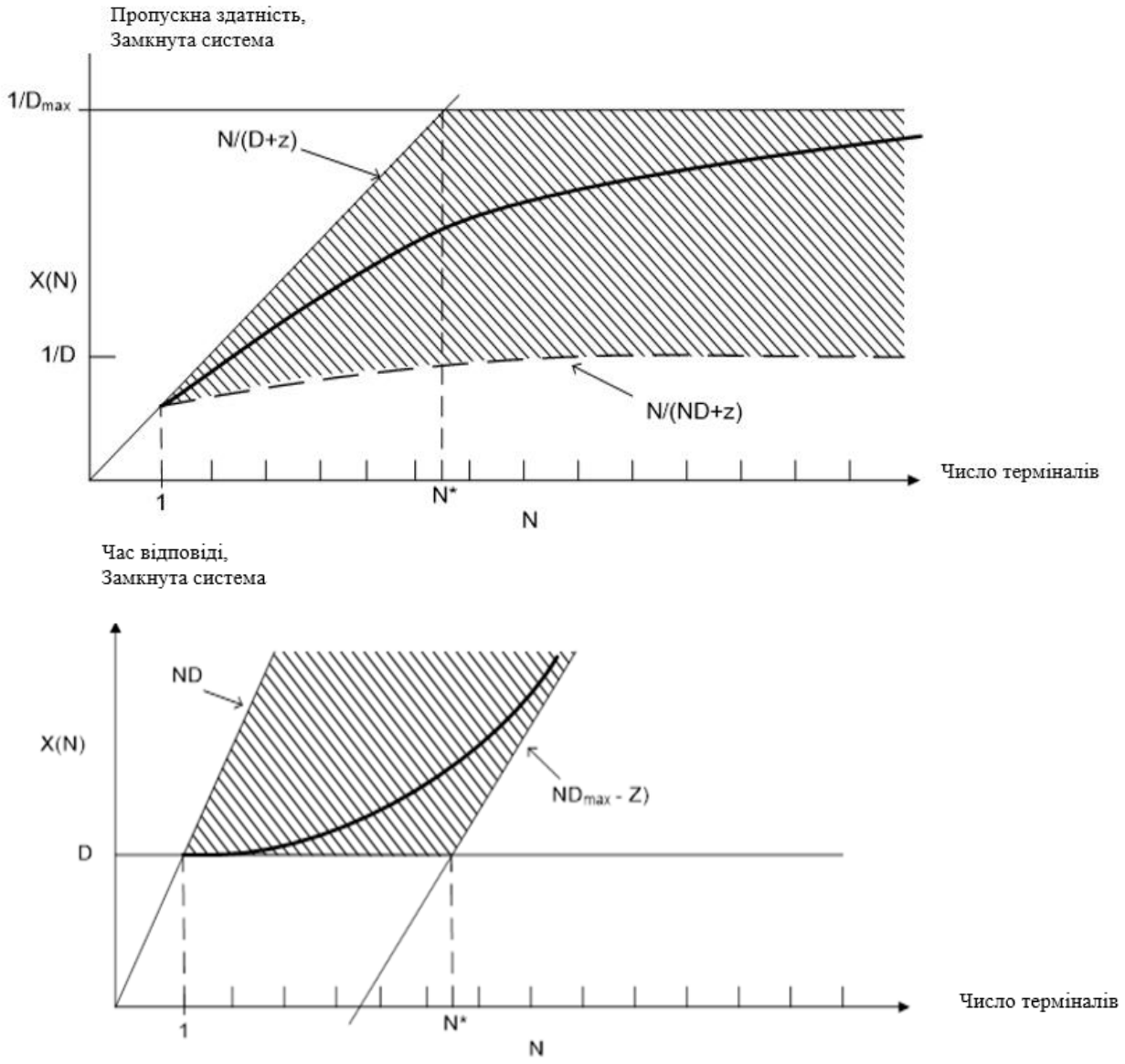

Рисунок 3.1 - Асимптотичні межі ефективності

Щоб отримати межі, показані на рисунках, ми спочатку розглянемо межі пропускної здатності, а потім використаємо закон Літтла, щоб перетворити їх у відповідні межі часу відгуку. Наш аналіз представлений в термінах робочих навантажень терміналів. Приймаючи час на роздуми, Z, за нуль, ми отримуємо результати для пакетних навантажень.

Ми почнемо з ситуації з великим навантаженням (багато клієнтів). Коли кількість клієнтів в системі (*N*) стає великою, завантаженість всіх серверів зростає, але, очевидно, що жоден з них не може перевищувати одиницю. Із закону коефіцієнта використання ми маємо для кожного центру

k наступне:

$$U_к(N) = X(N) \cdot D_k \leq 1 .$$

Кожен центр обмежує максимально можливу пропускну здатність, якої може досягти система. Оскільки вузьке місце (max) насичується першим, воно обмежує пропускну здатність системи найсильніше. Ми робимо такий висновок:

$$X(N) \leq \frac{1}{D_{\max}} .$$

Інтуїтивно це зрозуміло, адже якщо кожен клієнт потребує в середньому D_{max} , одиниць часу на обслуговування в "вузькому місці", то в довгостроковій перспективі клієнти, безумовно, не можуть обслуговуватися швидше, ніж один раз на D_{max} одиниць часу.

Далі розглянемо ситуацію з невеликим навантаженням (декілька клієнтів). В крайньому випадку, один клієнт в системі досягає пропускної здатності *1/(D+Z),* оскільки кожна взаємодія складається з періоду обслуговування (середньої тривалості) і часу на роздуми (середньої тривалості *Z*).

$$D = \sum_{k-1}^{K} D_k .$$

Коли до системи додається більше клієнтів, виникають дві граничні ситуації.

1) Найменша можлива пропускна спроможність виникає тоді, коли

кожен додатковий клієнт змушений стояти в черзі за всіма іншими клієнтами, що вже знаходяться в системі. У цьому випадку, з N клієнтів в системі, *D (N - 1)* одиниць часу витрачається на очікування в черзі, D одиниць часу витрачається на обслуговування, а *Z* одиниць часу витрачається на роздуми, так що пропускна здатність кожного клієнта становить *1/(ND+Z)*. Таким чином, пропускна здатність системи дорівнює *N/(ND + Z)*.

2) Максимально можлива пропускна здатність досягається тоді, коли кожен додатковий клієнт не затримується жодним іншим клієнтом в системі. У цьому випадку час очікування в черзі не витрачається, D одиниць часу витрачається на обслуговування, а Z одиниць часу витрачається на роздуми. Таким чином, *пропускна здатність кожного клієнта становить 1/(D + Z),* а пропускна здатність системи - *N/(D + Z)*.

Наведені вище спостереження можна підсумувати як асимптотичні межі пропускної здатності системи:

$$\frac{N}{ND+Z} \le X(N) \le \min\left(\frac{1}{D_{\max}}, \frac{N}{D+Z}\right) .$$

Зауважте, що оптимістична межа складається з двох компонентів, перший з яких застосовується при великому навантаженні, а другий - при малому навантаженні. Як показано на Рисунку 3.1, існує певний розмір популяції *N**, при якому для всіх N, менших за *N**, застосовується оптимістична межа легкого навантаження, тоді як для всіх N, більших за *N**, застосовується межа важкого навантаження. Точка перетину виникає там, де значення обох границь є рівними:

$$N^* = \frac{D+Z}{D_{\max}} .$$

Ми можемо отримати обмеження на час відгуку R(N), перетворивши наші обмеження на пропускну здатність за допомогою закону Літтла. Почнемо з переписування попереднього рівняння:

$$\frac{N}{ND+Z} \le \frac{N}{R(N)+Z} \le \min(\frac{1}{D_{\max}}, \frac{N}{D+Z}) .$$

Інвертування кожного компонента для того, щоб виразити обмеження на R(N), дає результат:

$$\max(D_{\max}, \frac{D+Z}{N}) \le \frac{R(N)+Z}{N} \le \frac{ND+Z}{N} ,$$

або

$$\max(D, ND_{\max} - Z) \le R(N) \le ND$$

3.2.3 Підсумковий огляд асимптотичних границь

У таблиці 3.1 зібрані усі аналітичні вирази асимптотичних оцінок. Алгоритм 3.1 показує кроки, за допомогою яких можна обчислити асимптотичні границі для термінального навантаження. (Обчислення для транзакційних навантажень є тривіальними.) Зауважте, що всі границі є прямими лініями, за винятком песимістичної границі пропускної здатності для термінальних навантажень. Отже, якщо відомі D та D_{max} , обчислення асимптотичних меж, виражених як функції від кількості клієнтів у мережі, займає лише кілька арифметичних операцій. Обсяг обчислень не залежить як від кількості серверів у моделі, так і від діапазону популяцій клієнтів, які нас цікавлять.

Таблиця 3.1 - Зведення асимптотичних меж

Параметр	Тип системи	Межі
X	Закнута	$\frac{N}{ND+Z} \leq X(N) \leq \min(\frac{N}{D+Z}, \frac{1}{D_{max}})$
	Разомкнута	$X(\lambda) \leq 1/D_{max}$
R	Закнута	$\max(D, ND_{max} - Z) \leq R(N) \leq ND$
	Разомкнута	$D \leq R(\lambda)$

4 АНАЛІЗ ПРОДУКТИВНОСТІ СИСТЕМИ НА ОСНОВІ ВУЗЬКИХ МІСЦЬ

4.1 Закон збереження потоків і системні операційні залежності

В даному розділі розглянемо можливості операційного аналізу для оцінки показників ефективності мережі [15] . З математичної точки зору ця проблема зводиться до зв'язування операційних змінних певними залежностями (рівняннями).

4.1.1 Рівняння балансу потоків

Основні результати операційного аналізу щодо мережі формуються у вигляді співвідношень, які пов'язують операційні змінні. Ці співвідношення ґрунтуються на гіпотезі про баланс (збереженні) потоків в мережі (Flow Balance): кількість заявок, які надійшли до деякого вузла на протязі тривалого періоду часу T, дорівнює кількості заявок, які залишили цей вузол за час T. Коли потоки в мережі збалансовані , X_i можна розглядати як продуктивність вузла i. Гіпотеза про баланс потоків в мережі може бути представлена у вигляді рівняння:

$$A_j = C_j \qquad j = \overline{0, K} \ . \tag{4.1}$$

Ця гіпотеза визначає умови роботи мережі СМО в сталому режимі, тобто вважається, що заявки завжди залишають вузли мережі на аналізованому періоді часу T. Вже згадана гіпотеза може бути використана не тільки для періоду спостереження за системою T. Це може виявитися непоганим наближенням і для випадку тривалого періоду спостереження T, коли відношення $(A_j - C_j)/ C_j$ є

незначним. Це буде дотримуватися, якщо початкова довжина черги $n_i(0)$ буде такою ж, як і кінцева $n_i(T)$.

Гіпотеза про баланс потоку дозволяє записати рівняння балансу (збереження) потоків заявок:

$$X_j = \sum_{i=0}^{K} X_i q_{ij}, \quad j = \overline{0,K} \qquad (4.2)$$

,

або

$$X_0 = X_0 q_{00} + X_1 q_{10} + ... + X_k q_{k0},$$
$$X_1 = X_0 q_{01} + X_1 q_{11} + ... + X_k q_{k1},$$
$$.$$
$$.$$
$$X_k = X_0 q_{0k} + X_1 q_{1k} + ... + X_k q_{kk},$$

Графічна інтерпретація рівняння балансу потоків представлена на рисунку 4.1:

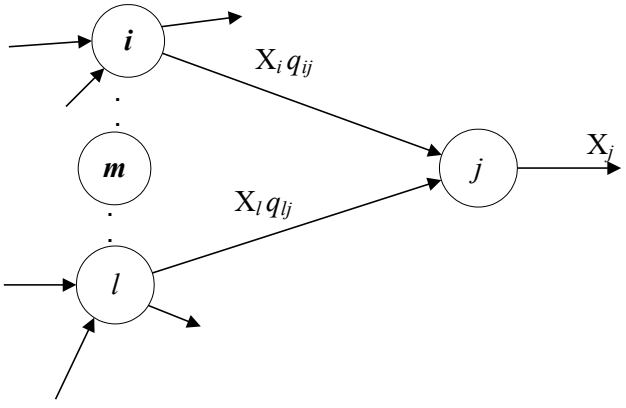

Рисунок 4.1 – Графічна інтерпретація рівняння балансу потоків

Доведемо справедливість виразу (4.2). Відомо що

$$\sum_{i=0}^{K} C_{ij} = A_j, \ i = 0,....K \ .$$

За умови, що $q_{ij}=C_{ij}/C_i$, а $A_j = C_j$ знаходимо $A_j = C_j = \sum_{i=0}^{K} C_i q_{ij}$. Поділивши ліву і праву частини останнього співвідношення на час спостереження Т, отримаємо систему лінійних рівнянь балансу потоків (4.2). Практично, маємо (К + 1) лінійно незалежних рівнянь і (К + 1) невідоме, матриця $|q_{ij}|$ вважається заданою.

Зауважимо, припущення про баланс потоків, дозволяє замінити C_{0j} на A_{0j}. Це очевидно, тому що всі заявки, що надійшли в вузол j із зовнішнього середовища, будуть обслужені

Система рівнянь балансу потоків являє собою значну цінність, тому що показує взаємозв'язок між структурою мережі, представленої матрицею $|q_{ij}|$ і продуктивністю вузлів X_i .

У загальному випадку, для розімкнутої мережі в виразі (4.2) маємо (К + 1) лінійно незалежних рівнянь і (К + 1) невідомих. Ці рівняння мають єдине рішення щодо невідомих X_i. Зазвичай, значення X_0 задається ($\lambda = X_0$) в початковому стані, тоді отримуємо К лінійно незалежних рівнянь і (К+1) невідомих. Рівняння (4.2) матимуть як і раніше єдине рішення.

Якщо мережа замкнута, в системі (4.2) одне рівняння є лінійною комбінацією всіх інших. Наприклад, X_0 - рівняння, може бути отримано як сума X_i - рівнянь. Отримуємо К лінійно незалежних рівнянь і (К + 1) невідоме. У цьому випадку єдине рішення відсутнє. В цьому випадку необхідно задати якийсь невідомий параметр.

Рівняння балансу потоку містять надзвичайно важливу інформацію і можуть бути вирішені за умови додаткових вимірів окремих операційних змінних.

Гіпотеза про баланс потоків дає можливість визначати коефіцієнт використання вузла в наступному вигляді для даного випадку:

$$U_i = \lambda_i \, S_i; \qquad (4.3)$$

Вираз (4.3) - це закон коефіцієнта використання вузла, який виконується за умови, що $A_i = C_i$, на протязі всього періоду спостереження T (в цьому випадку $\lambda_i = X_i$).

Нагадаємо, що без урахування гіпотези про баланс потоків (5.9), маємо:

$$U_i = X_i S_i;$$

4.1.2 Коефіцієнт відвідування вузла. Закон примусового потоку

У розділі було розглянуто поняття середній час виконання запиту (або запиту на обслуговування) і показано, що один запит може генерувати кілька відвідувань, наприклад, до жорсткого диску, принтера або іншого ресурсу системи. Якщо потік запитів в мережі збалансований, дуже корисним є поняття коефіцієнта відвідування вузла i.

Визначимо коефіцієнт відвідування вузла i, як середнє число звернень до вузла i на один обслужений запит. Як буде показано нижче, ці коефіцієнти можуть бути обчислені однозначно з системи рівнянь балансу потоків (4.2). Разом із середнім часом обслуговування в вузлі і їх можна використовувати для обчислення продуктивності X_i, середнього часу обслуговування D_i і часу відповіді системи при незначних і великих навантаженнях.

Коефіцієнт відвідування вузла i позначимо як Vi. За визначенням:

$$V_i = \frac{X_i}{X_0}. \qquad (4.4)$$

Нагадаємо, що X_i - це інтенсивність вихідного потоку запитів вузла i, а X_0 - це інтенсивність вихідного потоку заявок системи. З (4.4) легко отримати:

$$V_i = \frac{C_i}{C_0}.$$

З виразу (4.4) отримаємо співвідношення, яке називають операційним законом примусового потоку (Forced Flow Law):

$$X_i = V_i \, X_0 \quad . \qquad (4.5)$$

Цей закон говорить про те, що потік в будь-якій частині системи визначає інші потоки, де б вони не знаходилися в системі.

Приклад. Оцінка продуктивності системи. Заявки, які надходять в систему, генерують в середньому 5 звернень до диску. Продуктивність диска становить 10 звернень/сек. Яка продуктивність системи X_0?

Практично, потрібно визначити продуктивність системи не маючи інформації про її структурі та характеристиках зв'язків, існуючих, наприклад, між диском і будь-якиою іншиою частиною системи. Проте, припущення про баланс потоків і відомий коефіцієнт відвідування вузла i (або закон примусового потоку) дозволяє нам визначити продуктивність системи:

$X_0 = X_i / V_i$ = (10 звернень/сек)/(5 звернень /запит) = 2 запита/сек.

Середній час обслуговування D_i вузла i, може бути визначено з використанням коефіцієнта відвідування вузла V_i як:

$$D_i = V_i \times S_i. \qquad (4.6)$$

Покажемо це:

$$D_i = U_i / X_0 = (B_i / T) / (C_0 / T) = B_i / C_0 = (C_i \times S_i) / C_0 = (C_i / C_0) \times S_i = V_i \times S_i.$$

Примітка. На відміну від S_i, яке визначає середній час обслуговування одного звернення до вузла i, D_i - це середній час обслуговування одного запиту в вузлі i.

У багатьох випадках, коефіцієнти відвідувань і часи обслуговувань отримати нелегко. Проте, рівняння (4.6) показує, що середній час обслуговування одного запиту в вузлі може бути обчислений безпосередньо з коефіцієнта використання і продуктивності системи. Для випадку неоднорідних запитів закон запиту на обслуговування має вигляд:

$$D_{i,r} = U_{i,r} / X_{0,r} = V_{i,r} \times S_{i,r},$$

де r - це тип запросу, а i - це вузол.

Визначення коефіцієнтів відвідування. Систему рівняння балансу потоку можна записати як еквівалентну систему рівнянь, в якій замість інтенсивності потоків використовуються коефіцієнти відвідування кожного вузла мережі. Для цього у виразі (4.2) замінимо X_j на $V_j X_0$, в результаті, отримаємо рівняння коефіцієнтів відвідування вузлів:

$$V_j = \sum_{i=0}^{K} V_i q_{ij}, \ \ j = \overline{0, K} \ , \qquad (4.7)$$

або

$$V_0 = V_0 q_{00} + V_1 q_{10} + \ldots + V_k q_{k0},$$

$$V_1 = V_0 q_{01} + V_1 q_{11} + \ldots + V_k q_{k1},$$

.

.

$$V_k = V_0 q_{0k} + V_1 q_{1k} + \ldots + V_k q_{kk},$$

Розглянемо випадок j = 0. Покажемо, що твердження $C_{00}=0$ призводить до $q_{00} = 0$, маємо $V_0 q_{00} = 0$.

$$V_0 = 1.$$

Відповідно до закону збереження потоків $A_0 = C_0$, і враховуючи, що $V_i = \frac{C_i}{C_0}$ та $q_{i0} = \frac{C_{i0}}{C_i}$, маємо $V_i q_{i0} = C_{i0} / C_0$. Тоді для j = 0 рівняння має вигляд:

$$V_0 = V_0 q_{00} + \sum_{i=1}^{K} V_i q_{i0} = \sum_{i=1}^{K} C_{i0}/C_0 = A_0/C_0 = 1, \; j = \overline{1,K}.$$

Система рівнянь (4.7) для випадку $j = \overline{1,K}$ матиме вигляд:

$$V_j = q_{0j} + \sum_{i=1}^{K} V_i q_{ij}, \; j = \overline{1,K}.$$

Вираз (4.7) являє собою систему (K + 1) лінійно незалежних рівнянь з (K + 1) невідомими $V_i, \; j = \overline{0,K}$: єдине рішення завжди можливо, в припущенні, що мережа функціонально пов'язана (operationally connected) і

дотримується закон збереження потоків. З огляду на те, що $V_0 = 1$, практично маємо систему (K) лінійно незалежних рівнянь з K невідомими V_i, $\mathrm{j} = \overline{0, \mathrm{K}}$. Вирішення цієї системи не забезпечує визначення продуктивності вузлів.

Таблиця 4.1 – Основні співвідношення операційного аналізу

Коефіцієнта використання вузла і	U_i	$= X_i S_i$
Середнє число заявок та середній час відповіді для вузла і	$\bar{n}_i$	$= X_i R_i$
Продуктивність системи або закон вихідного потоку	X_0	$= \sum_{i=1}^{K} X_i q_{i0}$
Середній час обслуговування запиту у вузлу	D_i	$= \frac{U_i}{X_0} = \mathrm{Vi} \times \mathrm{Si}$
Коефіцієнт відвідування вузла V_i	X_i	$= V_i X_0$
Рівняння балансу потоків	X_j	$= \sum_{i=0}^{K} X_i q_{ij}, \ j = \overline{0, K}$
Час відповіді розімкнутої мережі СМО	R	$= \sum_{i=1}^{K} V_i R_i = \overline{N} / X_0$
Час відповіді розімкнутої мережі СМО	R	$= \frac{\overline{N}}{X_0} = \sum_{i=1}^{K} V_i R_i$
Час відповіді замкненої мережі СМО	R'	$= M/X_0 - Z$

4.2 Експериментальна оцінка продуктивності системи

4.2.1 Аналіз вузьких місць у мережі з використання аналітичного моделювання

Пошук вузьких місць у мережі є важливим аспектом аналізу роботи

[3,6]. Вузьке місце утворюється тим вузлом мережі, коефіцієнт використання якого наближається до одиниці. У цьому вузлі створюється велика черга заявок, яка за умови U≈1 стає нескінченною, тому мережа перетворюється на нестійкий режим роботи. Такий вузол стає «насиченим» заявками. Вузькі місця у мережі визначають її пропускну здатність. Тому під час аналізу роботи мережі потрібно приділяти особливу увагу пошуку таких місць.

Цей розділ розглядає асимптотику продуктивності та часу відповіді замкнутих (чому?) систем, коли N, кількість завдань у системі, збільшується. Ми припускатимемо, що коефіцієнти відвідувань та середні часи обслуговування інваріантні змінам N.

Звернемо увагу, що відношення продуктивностей будь-яких двох вузлів дорівнює відношенню їх коефіцієнтів відвідувань:

$$X_i / X_j = V_i / V_j.$$

Так як $U_i = X_i S_i$, отримуємо:

$$U_i/U_j = V_i S_i/V_j S_j.$$

Наше припущення про інваріантність свідчить, що це відносини задовольняють будь-яким N.

Вузол і перевантажений (насичений), якщо коефіцієнт використання сягає 100%. Якщо $U_i = 1$, отримуємо:

$$X_i = 1/S_i.$$

Це означає, що перевантажений пристрій обслуговує запити на межі – в середньому один запит кожні Si сек. У загальному випадку $U_i \leq 1$ і $X_i \leq 1/S_i$ (це очевидно, тому що максимум X досягається, коли $U_{max} = 1$).

Нехай індекс b означає будь-який вузол, здатний до перевантаження зі

збільшенням N. Цей вузол називають критичним вузлом, оскільки обмежує загальну продуктивність системи. Будь-яка мережа має хоча б один критичний елемент (це відбувається тому, що при збільшенні N завжди знайдеться хоча б один вузол, що досягає насичення).

Оскільки відносини Ui/U_j незмінні, пристрій і з найбільшим значенням V_iS_i буде першим, яке досягне 100% використання при збільшенні N. Тому можна сказати, що пристрій є критичним елементом,

$$V_bS_b = \max \{V_1S_1 ..... V_KS_K\}.$$

Критические элементы определяются устройством и параметрами рабочей нагрузки.

Если N становится большим, мы наблюдаем $U_b = 1$ и $X_b = 1/S_b$. Поскольку $X_0/X_b = 1/V_b$, получаем:

Критичні елементи визначаються пристроєм та параметрами робочого навантаження.

Якщо N стає більшим, ми спостерігаємо $U_b = 1$ і $Xb = 1/S_b$. Оскільки $X_0/X_b = 1/V_b$, отримуємо:

$$X_0 = 1/V_bS_b.$$

Цей вираз визначає максимально можливе значення продуктивності системи.

Нехай N = 1. Враховуючи, що V_iS_i – це середній час обслуговування одного запиту (запит на обслуговування) пристроєм і тоді сума

$$R_0 = V_1S_1+\ldots+V_KS_K,$$

ігнорує затримки в чергах, визначає найменший можливий середній час відповіді системи (це так, тому що V_iS_i – це середній час обслуговування

одного запиту пристроєм i). N, кількість завдань у системі Насправді R_0 – це час відповіді, коли N = 1. Тому $X_0 = 1/R_0$ коли N = 1.

Властивості продуктивності системи X_0 показані малюнку 4.2. Як функція N, X_0 монотонно зростає від $1/R_0$ при N = 1 до асимптоти $1/V_bS_b$. Але залишається нижче лінії нахилу $1/R_0$, що виходить із центра координат. Для випадку N = k, затримки запитів у чергах зазвичай неможливо досягти продуктивністю значення k/R_0.

Якщо припустити, що при реалізації k запитів завжди вдається уникнути затримок у чергах, тоді $X_0 = k/R_0$, асимптота перевантаження, вимагає, щоб $k/R_0 \le 1/V_bS_b$, або

$$k \le N^* = \frac{R_0}{V_bS_b} = \frac{V_1S_1 + \ldots + V_KS_K}{V_bS_b} \le K .$$

Пояснимо це обмеження в такий спосіб. к > N* означає з упевненістю наявність черг десь у системі та вплив вузького місця на продуктивність системи. Тоді як N* означає навантаження, нижче за яке вузьке місце ще не впливає на продуктивність системи. Назвемо N* точкою насичення системи.

Ці результати поширюються і оцінку часу відгуку замкнутої системи чи системи керованої терміналами. Для M терміналів та часу обмірковування Z, середній час відгуку $R = M/X_0 - Z$. Коли M = 1 (в системі одна заявка), R має бути мінімальним і дорівнює R_0. Оскільки X_0 не може перевищувати $1/V_bS_b$ (тобто $\max X_0 = 1/V_bS_b$), різниця $(M/\max X_0 - Z) = (M\ V_bS_b - Z) = \text{const}$. і буде мінімальним. З цього випливає значення R при інших значеннях X_0 задовольняє нерівності

$$R \ge MV_bS_b - Z.$$

З іншого боку продуктивність вузла i $X_i = 1/V_iS_i$ завжди вища за продуктивність насиченого вузла b $X_b = 1/V_bS_b$ тому отримуємо

$$R \geq MV_bS_b - Z \geq MV_iS_i - Z_i, \quad i=l,\ldots,K.$$

У міру збільшення M, R наближається до асимптоту MV_bS_b - Z. Щодо асимптоти. Вона лінійна, т.к. V_bS_b інваріантні (константи). Це показано на рисунку 4.3. З малюнка видно, що асимптота часу відгуку припиняє горизонтальну вісь (MV_bS_b - Z=0) у точці

$$M_b = Z/V_bS_b = Z\ X_b\ .$$

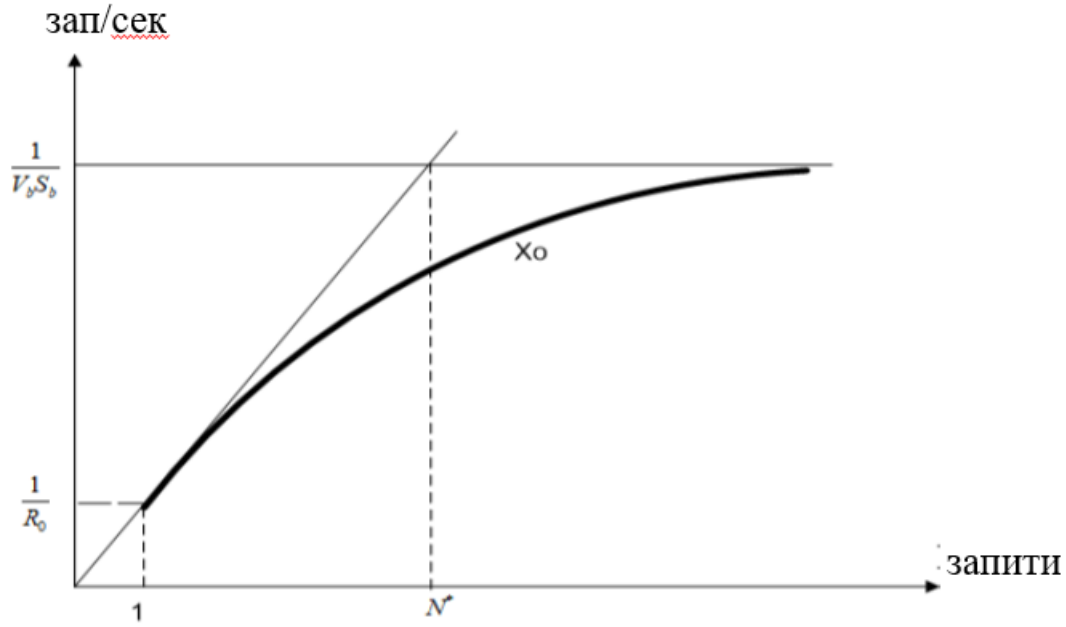

Рисунок 4.2 - Залежность продуктивності мережі від кількості запитів у системі

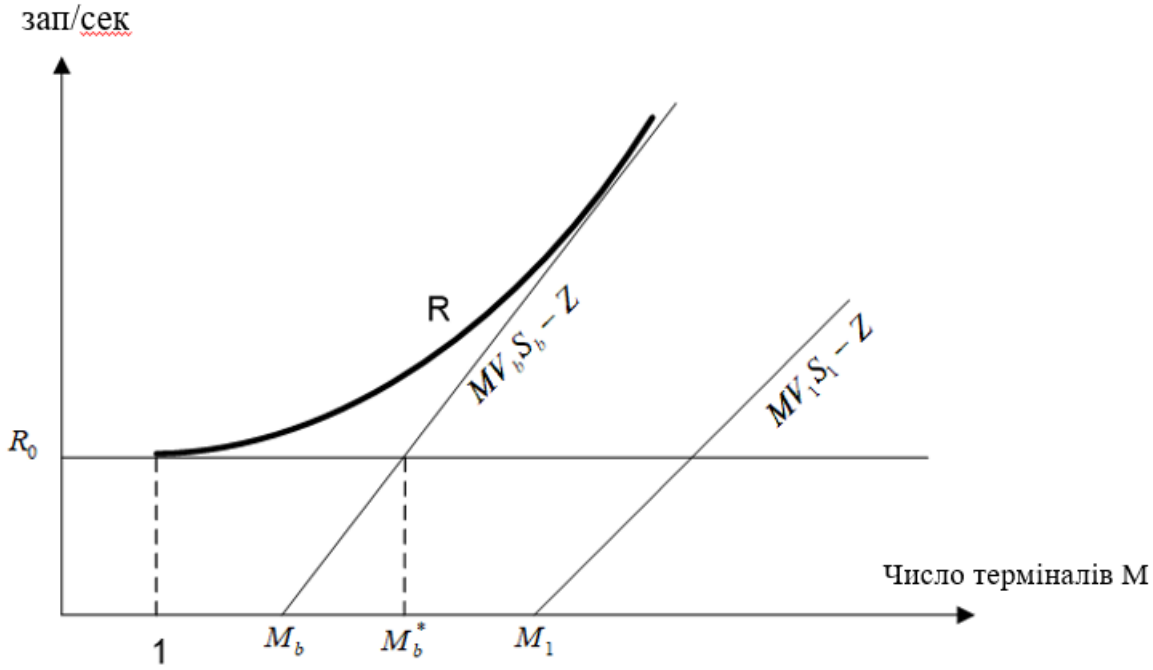

Рисунок 4.3 - Залежність часу відгуку від числа терміналів

Це множення середнього часу очікування терміналів (Z) і насиченого потоку заявок через термінали ($maxX_0 = 1/V_bS_b$). За законом Літтла, M_b позначає середню кількість терміналів, що обдумують, в момент насичення системи. Асимптота часу відгуку перетинає мінімальний час відгуку R_0 ($R_0 = MV_bS_b - Z$) у момент

$$M_b^* = (R_0 + Z)/V_bS_b = N^* + M_b.$$

Коли число терміналів, що обмірковують, більше ніж M_b^*, з упевненістю можна стверджувати про появу черг у центральній частині системи.

Зазначимо, що асимптоти часу відгуку і точки перетину M_b і M_b^* залежать лише від M, Z, V_b і S_b. Так само важливо відзначити, що коли $Z = 0$ ці результати призводять до асимптот часу відгуку замкнутої системи.

Підіб'ємо підсумок: параметри робочого навантаження або рівняння коефіцієнтів відвідувань дозволяють аналітику визначити коефіцієнти

відвідувань, V_i. Характеристики пристроїв дозволяють визначити середній час обслуговування одного відвідування S_i. Більше із творів V_iS_i визначає критичний пристрій b. Сума цих добутків визначає найменший можливий час відгуку R_0. Продуктивність системи при насиченні дорівнює $1/V_bS_b$. Точка насичення N* центральної підсистеми – R_0/V_bS_b і N^*+Z/V_bS_b терміналів які обмірковують, почнуть насичувати (перевантажувати) систему, керовану терміналами.

Аналіз, що призводить до рисунків 4.2 та 4.3, може дати загальне уявлення про ефекти запропонованих змін. Наприклад, зменшення V_iS_i для пристрою, що не є критичною точкою (тобто зменшення часу обслуговування або коефіцієнта відвідування) не впливатиме на критичну точкую. Не буде змін в асимптоті $1/V_bS_b$ та будуть незначні зміни у мінімальному часі відгуку R_0. Зменшення добутку V_iS_i всім критичних пристроїв видалить критичні точки; підніметься асимптота $1/V_bS_0$ та зменшиться R_0. Однак, цей ефект буде відмічатися до тих пір, поки V_bS_b буде найбільшим з V_iS_i: занадто сильне поліпшення пристрою b пересуне критичну точку в інше місце.

Приклад. Розрахуємо характеристики замкнутої мережі, зображеної на рисунку 4.4 де наведені значення операційних змінних S_k, q_{kj} і Z.

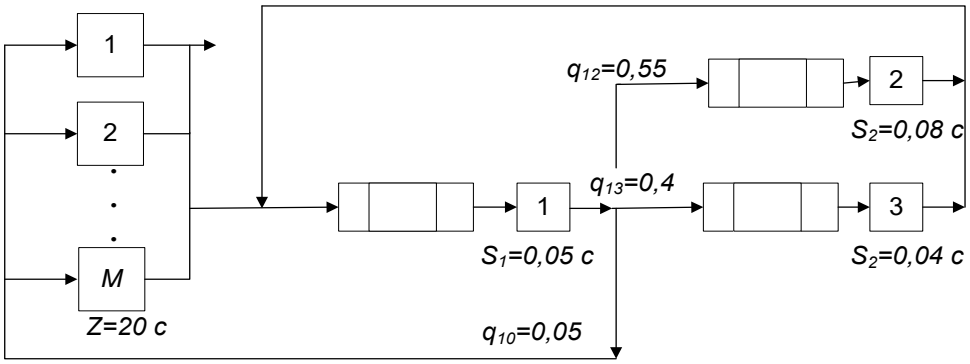

Рисунок 4.4 – Мережа СМО

Час роздуму становить Z = 20 секунд;

Середній час обслуговування S_k на пристрої і наведено на рисунку;

Частота маршрутизації q_{kj}, частка задач, що переходять на наступний пристрій j після завершення обслуговування на пристрої k, наведена на рисунку.

Припустимо, що за допомогою вимірювань було визначено, що:

Пропускна здатність системи $X_0 = 0,715$ запитів/сек,

Середній час перебування запиту в мережі $R = 5,2$ сек.

Використовуючи метод аналітичного, можна знайти відповідь на таке питання:

Яка середня кількість пристроїв M взаємодіє з мережею протягом періоду спостереження? Іншими словами знайти середнє число запитів у системі протягом періоду спостереження.

Рівняння балансу потоків для коефіцієнтів відвідуваності цієї мережі:

$$V_0=1=q_{10}V_1=V_1,$$

$$V_1=V_0+V_2+V_3,$$

$$V_0=q_{12}V_1=0,55V_1,$$

$$V_3=q_{13}V_1=0,4V_1.$$

Розв'язавши цю систему рівнянь, отримаємо

$$V_1 = 20,\ V_2=11;\ V_3=8.$$

Обчислити значення V_kS_k для кожного з вузлів мережі:

$$V_1S_1=20*0,5=1\text{sec.}$$

$$V_2S_2=11*0,8=0,88\text{sec.}$$

$$V_3S_3=8*0,04=0,32\text{sec.}$$

Таким чином, використовуючи метод аналітичного моделювання, було обчислено:

1) середнє число запитів у системі, $R = 5,2$сек та мінімальний середній час перебування одного запиту в мережі $R_0 = 2,2$сек.

$$R_0= 1 + 0,88 + 0,32 = 2,2\text{сек.}$$

2) знайдено вузьке місце, вузол 1 (CPU). Оскільки

$$V_1S_1 > V_2S_2 > V_3S_3,$$

тоді потенційним вузьким місцем у мережі є перший вузол. CPU має максимальний середній час обслуговування одного запиту.

3) збудовані асимптоти часу відгуку і продуктивності (рисунки 4.5 і 4.6). Вони повністю схожі на загальні асимптоти, наведені на рисунку 3.1.

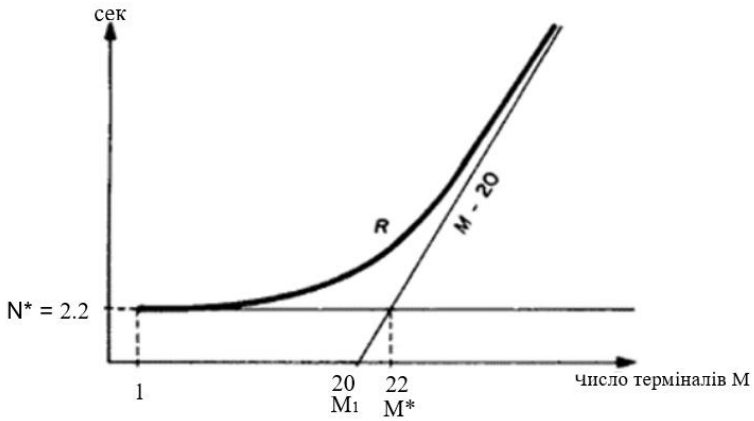

Рисунок 4.5 - Асимптоти часу відгуку

Рисунок 4.5 показує асимптоти кривої часу відгуку. M є число терміналів з користувачами, які розмірковують над відповіддю:

$$M_1 = Z/V_1S_1 = 20 \text{ терміналів.}$$

N* означає навантаження, нижче за яке вузьке місце ще не впливає на продуктивність системи. Назвемо N* точкою насичення системи.Точка перевантаження центральної підсистеми (комп'ютерної підсистеми)

$$N^* = R_0 /V_1S_1 = 2.2 \text{ задач.}$$

Число терміналів, необхідне досягнення перевантаження $M_1^* = 22$.

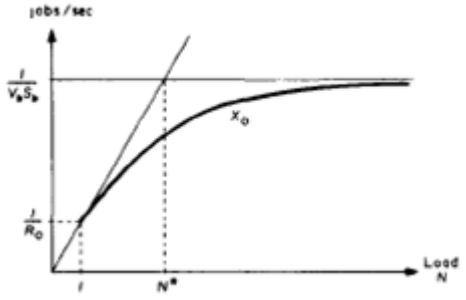

Рисунок 4.6 - Асимптоти продуктивності

4.2.2 Аналіз вузьких місць у мережі з використання імітаційного моделювання

В роботі було використано технологію імітаційного моделювання для асимтотичної оцінки параметрі і аналізу вузьких місць мережевої системи. Була використана імітаційна система GPSS. GPSS модель наведено у додатку Б

Порівняльний аналіз результатів аналітичного і імітаційного моделювання наведено в таблиці 4.2.

Таблиця 4.2 – Порівняльний аналіз результатів аналітичного і імітаційного моделювання

Компоненти аналіт\імітац	R (сек)	R_0 (сек)	D_i(sec)	X_0 (зап\сек)	U_i
CPU			1.00\1.25		0.85
Диск 1			0.88\0.67		0.51
Диск 1			0.32\0.45		0.54
Система	5.5\4.8	2.3\3.1		0,715\0.91	

Підсумок. Результати аналітичного і імітаційного моделювання мають задовільну розбіжність

ВИСНОВКИ

Теорія мереж масового обслуговування ґрунтується на передумові тестованості. Всі основні показники ефективності - використання, коефіцієнти завершення, середній розмір черги, середній час відгуку, розподіл навантаження - визначаються так, як вони були б на практиці, на основі даних, отриманих за певний період часу. Аналітик може перевірити, чи виконуються основні припущення - баланс потоку, однокрокова поведінка та однорідність - протягом будь-якого періоду спостереження.

Операційні закони - це тотожності між операційними величинами. Вони є перевіркою узгодженості - невиконання операційного закону вказує на помилку в даних. Вони спрощують збір даних, показуючи альтернативи для обчислення величин продуктивності.

Баланс потоку завдань передбачає, що пропускна спроможність в будь-якій точці системи визначається пропускною спроможністю в будь-якій точці системи. Оскільки збільшення навантаження призводить до насичення певного пристрою, це припущення дозволяє визначити асимптоти пропускної здатності та часу відгуку; єдиними даними, необхідними для такого "аналізу вузьких місць", є коефіцієнт відвідуваності та вихідна потужність пристроїв при насиченні.

Більшість помилок у цих результатах виникають через припущення про однорідність. Однорідність стверджує, що між пристроєм і рештою системи немає ніякої взаємодії, окрім залежності від довжини локальної черги. У реальній системі функція обслуговування буде залежати від шаблону, за яким решта системи надсилає запити до пристрою, і цей шаблон може залежати від форми розподілу розмірів запитів на цьому пристрої.

На практиці помилки, пов'язані з цими припущеннями, не є серйозними. Навіть коли додаткове припущення про однорідний час обслуговування використовується для подальшого спрощення аналізу, ці моделі оцінюють завантаження, пропускну здатність і час відгуку системи

зазвичай з точністю до 10%, а середню довжину черги і час відгуку пристрою - з точністю до 30%. Удосконалення моделі пристроїв, щоб зробити явним вплив розподілу розміру запиту, підвищує точність, особливо при прогнозуванні розподілу довжини черги. Про розподіл часу відгуку для цих систем відомо дуже мало.

Щоб використати ці результати для прогнозування ефективності, аналітик повинен оцінити значення параметрів для прогнозного періоду, а потім використати ці оцінки в рівняннях для обчислення очікуваних показників ефективності в прогнозному періоді. Ми не запропонували остаточного вирішення проблеми оцінки параметрів. Та й не можемо: вона належить до сфери індуктивної математики, тоді як операційний аналіз є галуззю дедуктивної математики. Ми проілюстрували на прикладах види припущень про інваріантність, які використовують аналітики для оцінювання параметрів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Balbo G., G. Serazzi, “Asymptotic Analysis of Multiclass Closed Queueing Networks: Common Bottleneck”, Performance Evaluation, Vol.26, No.1, pp 51-72, 1996

2. Balbo G. , G. Serazzi, “Asymptotic Analysis of Multiclass Closed Queueing Networks: Multiple Bottlenecks”. Performance Evaluation, Vol.30, No.3, pp 115-152, 1997

3. Berger A., L. Bregman, et. al., “Bottleneck Analysis in Multiclass Closed Queueing Networks and Its Application”, Queueing Systems, 31(3-4), pp 217-237, 1999

4. Bolch G., S. Greiner, et. al., “Queueing networks and Markov chains: modeling and performance evaluation with computer science applications”, John Wiley and Sons, 1998.

5. Bukchin J., “A comparative study of performance measures for throughput of a mixed model assembly line in a JIT environment”, International Journal of Production Research, Vol.36, No.10, pp 2669-2685, 1998

6. Casale G., G. Serazzi, “Estimating Bottlenecks of Very Large Models”, Performance Evaluation Stories and Perspectives-G.Kotsis Editor, Austrian Computing Society, pp 89-104, 2003

7. Casale G., G. Serazzi, “Bottlenecks Identification in Multiclass Queueing Networks using Convex Polytopes”, In Proc. IEEE/ACM MASCOTS 2004, IEEE Comp. Soc., pp 223–230, 2004

8. D. A. Menascé and V. A. F. Almeida, Capacity Planning for Web Services: Metrics, Models, d Methods, Prentice Hall, Upper Saddle River, New Jersey, 2002

9. V. A. F. Almeida and D. A. Menascé, "Capacity planning: An essential tool for managing services," IEEE IT Professal, vol. 4, no. 4, July / August 2002.

10. D. Ardagna and C. Francalanci, "A cost-orted methodology for the design of Web-based IT architectures, Proc. 17th ACM Symposium on Applied

Compng, Madrid, March 2002.

11. Y. Jiang and Y. Liu, Stochastic Network Calculus. Springer, London, 2008.

12. Y.J. May, “A note on applying stochastic network calculus,” 2010.

13. J. Ros-Giralt, N. Amsel, S. Yellamraju, J.R. Ezick, R.A. Lethin, Y. Jiang, A. Feng, and L. Tassiulas, “A quantitative theory of bottleneck structures for data networks,” 2022, arXiv:2210.03534.

14. J. Ros-Giralt, M. Veeraraghavan, A. Bohara, S. Yellamraju, M.H. Langston, R. Lethin, Y. Jiang, L. Tassiulas, J. Li, and Y. Tan, “On the bottleneck structure of congestion-controlled networks,” Proceedings of the ACM on Measurement and Analysis of Computing Systems, vol. 3, 2019.

15. Горбачов В.О. Моделювання систем, Навчальний посібник, Видавництво ХНУРЕ, 2005