

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)

Розробка структурної моделі  
інформаційної системи E-learning  
(тема)

Виконав:

студент \_\_\_\_\_ II курсу, групи \_\_\_\_\_ СПЗм-20-1  
Коткова О.М.  
(прізвище, ініціали)

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. Кучук Н.Г.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

\_\_\_\_\_ Коваленко А.А.  
(підпис) (прізвище, ініціали)

2023 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**

студенту \_\_\_\_\_ Котковій Оксані Миколаївні \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ Розробка структурної моделі інформаційної системи E-learning \_\_\_\_\_

затверджена наказом по університету від “ 25 ” березня 2022 р. № 33 СТз \_\_\_\_\_

2. Термін подання студентом роботи до екзаменаційної комісії \_\_\_\_\_ 18 травня 2022 р. \_\_\_\_\_

3. Вхідні дані до роботи \_\_\_\_\_ Windows, мова програмування – Java, середовище програмування – IntelliJ IDEA, гіперконвергентна мережа. \_\_\_\_\_

4. Перелік питань, що потрібно опрацювати у роботі \_\_\_\_\_

1) Аналіз існуючих підходів до синтезу елементів інформаційної системи e-learning. \_\_\_\_\_

2) Математична модель процесу синтезу інформаційної системи \_\_\_\_\_

3) Розробка програмного забезпечення процесу моделювання синтезу інформаційної системи \_\_\_\_\_

4) Аналіз результатів синтезу моделі \_\_\_\_\_

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) \_\_\_\_\_  
17 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	Аналіз предметної області	28.03.22-05.04.22	
2	Збір і аналіз вимог для тестів на проникнення	05.04.22-20.04.22	
3	Розробка методу автоматизованого тестування на проникнення з використанням технології deep learning	21.04.22-15.05.22	
4	Тестування розробки	16.05.22-20.05.22	
5	Оформлення матеріалів кваліфікаційної роботи	21.05.22-30.05.22	
6	Подання кваліфікаційної роботи керівникові та її попередній захист	13.03.2023	
7	Подання кваліфікаційної роботи на рецензування	12.05.23-16.05.23	

Дата видачі завдання 26 березня 2022 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

проф. Кучук Н.Г.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 86 с., 16 рис., 7 табл., 2 дод., 19 джерел.

**E-LEARNING, МОДЕЛЮВАННЯ, ГІПЕРКОНВЕРГЕНТНА МЕРЕЖА,  
ГЕНЕТИЧНИЙ АЛГОРИТМ, РОЗПОДІЛЕННЯ НАВАНТАЖЕННЯ.**

Метою кваліфікаційної роботи є – розробка інформаційної моделі системи e-learning, на гіперконвергентної основі, яка враховує її особливості та дозволяє встановити інформаційні взаємозв'язки між складовими системи та провести аналіз базової мережі.

Побудовано модель інформаційної структури системи підтримки електронних освітніх ресурсів системи e-learning; відповідно до цієї моделі, було створено модель синтезу інформаційної системи e-learning.

## ABSTRACT

Master's thesis: 86 pages, 16 figures, 7 tables, 2 appendices, 19 sources.

E-LEARNING, MODELING, HYPERCONVERGENT NETWORK, GENETIC ALGORITHM, LOAD DISTRIBUTION.

It was built a model of the information structure of the support system for electronic educational resources e-learning according to this model, was created by the synthesis model of information system e-learning. The result of the thesis was a program with a graphical user interface fusion models information systems e-learning, which takes as input the parameters of the Hyper-converged core network and the e-learning system, as a result of the synthesis is the optimal location of users, applications, and data blocks on the nodes of the core network considering the traffic load on the transaction system optimizing the loading of the nodes. Based on the results of the pilot study, it was proposed to improve the efficiency of the e-learning system.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	8
ВСТУП .....	9
1 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО СИНТЕЗУ ЕЛЕМЕНТІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ E-LEARNING.....	11
1.1 Аналіз сучасних технологій і розробок e-learning.....	11
1.2 Аналіз гіперконвергентних мереж та порівняння із звичайними рішеннями .....	21
1.3 Аналіз методів управління мережевими ресурсами.....	28
1.4 Аналіз методів моделювання інформаційної структури.....	35
2 МАТЕМАТИЧНА МОДЕЛЬ ПРОЦЕСУ СИНТЕЗУ ІНФОРМАЦІЙНОЇ СИСТЕМИ .....	38
2.1 Аналіз необхідності моделювання інформаційної інфраструктури .....	38
2.2 Модель основних інформаційних взаємозв'язків у середовищі e- learning.....	40
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОЦЕСУ МОДЕЛЮВАННЯ СИНТЕЗУ ІНФОРМАЦІЙНОЇ СИСТЕМИ .....	45
3.1 Особливості оптимізації з використанням генетичного алгоритму .....	45
3.2 Поєднання моделі та генетичного алгоритму .....	47
3.3 Розроблений алгоритм синтезу моделі інформаційної структури системи e-learning.....	48
3.4 Діаграма класів, які було використано при розробці .....	50
Діаграма варіантів використання відображена на рисунку 3.2, вона допомагає при розробці синтезу ПЗ.....	50
3.5 Схема бази даних .....	54
3.7 Пояснення вибору мови програмування.....	55
4 АНАЛІЗ РЕЗУЛЬТАТІВ СИНТЕЗУ МОДЕЛІ.....	57

4.1 Інтерфейс програмного комплексу .....	57
ВИСНОВКИ.....	66
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	67
ДОДАТОК А <b>Г</b> рафічний матеріал кваліфікаційної роботи.....	70
ДОДАТОК Б .....	80
Детальна структура класів .....	80

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ  
І ТЕРМІНІВ

ЕОР – електронні освітні ресурси;  
ММО – мережа масового обслуговування;  
ГБМ – гіперконвергентна базова мережа;  
E-learning – електронне навчання;  
M-learning – мобільне навчання;  
B-learning – змішане навчання;  
U-learning – загально проникаюче навчання;  
ПК – персональний комп'ютер;  
LMS – Learning Management System;  
HCI – гіперконвергентність;  
SDS – Software-Defined Storage;  
VSA – Virtual Storage Appliance;  
SAN – Storage Area Network;  
VSAN – Virtual SAN.

## ВСТУП

Прогрес та розповсюдження інформаційних технологій стимулює персоналізацію освіти, інтеграцію педагогічних та інформаційних технологій, перехід до відкритого змісту освіти. Все це забезпечується втіленням концепції e-learning, яка в купі з широким набором даних, та функціональними додатками, забезпечує навчання побудоване на використанні технології Internet.

Базою інформаційної системи E-learning слугують електронні освітні ресурси (EOR), котрі являють собою сукупність засобів програмного, інформаційного, технічного та організаційного забезпечення електронних видань. EOR складається з набору функціональних додатків та блоків даних з інформацією про студентів, викладачів та необхідних для навчання матеріалів [1, 2].

Актуальність роботи: для більшості ЗВО України постає проблема створення, розвитку та підтримки EOR, це вимагає коштів, що значно перебільшують їх бюджет, тому має сенс звернути увагу на платформу, що здатна зменшити ці витрати.

Також -технології поступово покидають хмарні платформи, на їх місце приходять конвергентні та гіперконвергентні платформи. Конвергентна інфраструктура передбачає об'єднання ресурсів системи в пул для роботи в дата-центрі; при гіперконвергентній інфраструктурі ресурси об'єднуються в одне ціле, а управління ними відбувається через загальну консоль адміністрування. Просте проектування системи, швидке розгортання у порівнянні зі звичайними рішеннями, інколи для управління системою достатньо одного системного адміністратора - всі ці переваги притаманні гіперконвергентній структурі. Це дозволяє суттєво зменшити витрати на обслуговування системи, що, на даний момент, є кращим рішенням для університетської системи e-learning [3, 4].

Використання однієї гіперконвергентної інфраструктури не достатньо для ефективного використання мережі. Структура гіперконвергентної базової мережі підтримки e-learning – це основний фактор, що впливає на якість функціонування системи. Тому важливо проаналізувати структуру при виборі варіантів побудови ГБМ та управління нею [5]. Основна мета аналізу структури є визначення параметрів потоків даних в каналах зв'язку мережі. Ці дані необхідні для оцінки навантаження каналів та вузлів мережі. Але потоки даних формуються завданнями e-learning, котрі використовують застосунки, які запускаються на вузлах мережі та генерують ще трафік між собою. Отже для аналізу мережі необхідні відомості про структуру, транзакції та застосунки e-learning, їх взаємодія й розміщення на вузлах ГБМ, що, потребує розробки відповідної математичної моделі [6]. У даній роботі представлено модель синтезу інформаційної системи e-learning.

Метою роботи є розробка інформаційної моделі інформаційної системи E-learning, на гіперконвергентної основі, яка враховує її особливості та дозволяє встановити інформаційні взаємозв'язки між складовими системи та провести аналіз базової мережі.

Об'єкт дослідження – це процеси, що протікають між набором додатків і блоків даних системи e-learning розгорнутої на гіперконвергентному сервері.

Методи дослідження: теорія множин, методи системного аналізу, евристичні методи (налаштування генетичного алгоритму).

Предмет дослідження – математична модель інформаційної структури системи підтримки інформаційної системи E-learning.

Завдання дослідження:

- 1) побудувати математичну модель (програмне рішення) інформаційної структури системи підтримки ЕОР інформаційної системи E-learning;
- 2) ввести вхідні дані (за перевіркою обмежень);
- 3) сформувані вихідні дані, для невеликих вхідних значень  $U, N, A, D$ ;
- 4) сформувані граф інформаційної системи E-learning.

# 1 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО СИНТЕЗУ ЕЛЕМЕНТІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ E-LEARNING

## 1.1 Аналіз сучасних технологій і розробок e-learning

Відмінними характеристиками сучасної освіти є інтенсивне використання в процесі навчання інформаційних технологій, що забезпечують рівноправне придбання й передачу знань, вільний доступ до навчальних матеріалів, перехід від принципу «освіта на все життя» до принципу «освіта через уже життя» – найважливішої проблеми в умовах інформаційного суспільства; розвиток соціальних і емоційних здібностей і навичок учнів; індивідуалізація процесу придбання і передачі знань. Таким чином, на рівні світового освітнього співтовариства, поряд з тенденціями глобалізації та стандартизації, визначилися основні напрямки вдосконалення освітніх процесів вищої школи – інформатизація і персоніфікація.

Питання розвитку електронного навчання і самоосвіти актуальні в роботах як вітчизняних, так і зарубіжних авторів. Так e-learning в умовах сучасної економіки і педагогіки розглянуто в роботах Е. Тихомирової [10], В.В. Бовт, [10, 19], Н. Соболевої [11], В. М. Платонова [17]. Моделям і методам e-learning приділено велику увагу в роботах І. Морозова [7], G. D'angelo [13, 14]. Огляд інформаційно-телекомунікаційних технологій e-learning виконаний А. Ю. Рудневим [17], П. Коноваловим [9], D. Casey [16]. Однак в публікаціях не відображена цілісна картина сучасного стану та тенденцій розвитку e-learning та самоосвіти. Необхідність використання електронного навчання в освітньому процесі розглядали А. Андреев, А. Arafeh, A.W. Bates, С. Гури-Розенбліт, Е. Д. Патаракіна, А. В. Хуторський. Проблемам і перспективам електронного навчання присвячені роботи В.Є. Леднева, А.Е. Сатуніной, А.А. Ступина, Е. Е. Ступіної. Результативність і значимість електронного навчання підкреслюють А. А. Андреев, В. В.

Гріншкун, С. Г. Григор'єв, Г. Драйдент, У. Хортон, В. Holberg і ін. Проблеми впровадження та можливості електронного навчання - А. Андреев, Г. Драйдент, А. Е. Сатуніна, А. А. Ступін і Е. Е. Ступина, А. В. Хуторський та ін. [12- 14].

Створення середовища навчання потребує постійного переосмислення й удосконалення існуючих форм та засобів навчання. Світовій освіті притаманні декілька ступенів навчання: традиційне навчання (з очною та заочною формами), дистанційне навчання, електронне навчання (e-learning), мобільне навчання (m-learning), всеохоплююче навчання (u-learning). Останні три ступені являються базовими для створення нової індустрії навчання, яка охоплює світовий ринок, безліччю online курсів з найрізноманітнішою тематикою та різними способами реалізації в освіті. Відповідно визначенню Європейської комісії e-learning – це використання нових технологій мультимедіа та Internet для підвищення якості навчання за рахунок поліпшення доступу до ресурсів і сервісів, а також віддаленого обміну знаннями та спільної роботи [12, 15].

Невизначеність термінології – це одна з проблем в дослідженні електронного навчання. З одного боку, існує безліч різних трактувань однакових за звучанням понять, з іншого боку – різним за звучанням поняттям надають однаковий сенс в описі цифрових технологій [22]. Зараз популярною є тема дистанційного та віртуального навчання, Internet-навчання, online-технологіях, а також тема мобільного навчання і тощо. Але всі зазначені поняття по суті представляють собою різні уявлення електронного навчання. Від цього термін e-learning широко використовується в міжнародній науковій спільноті. Таким чином електронне навчання визначається як цілеспрямований та організований процес інтерактивної взаємодії учнів, вчителів та засобами навчання, незалежних до їх розташування в просторі та часі, який реалізується в специфічній дидактичній системі [13].

E-learning – це передача знань між кінцевими користувачами,

управління процесом навчання завдяки новим інформаційним та телекомунікаційним технологіям. Технології електронного навчання дозволяють створювати електронні курси, системи управління навчанням та навчальними об'єктами, системи контролю знань, а також засоби створення знань. Якісне здійснення e-learning можливе завдяки використанню цифрових репозиторіїв – баз даних і знань, що зберігають знання в цифровому вигляді, структуровані відповідно до принципів об'єктного опису метаданими. Цифрові репозиторії дозволяють побудувати взаємозв'язки між об'єктами знань з можливістю контекстного пошуку. В протипагу дистанційному навчанню (наприклад, з обміном матеріалами навчання поштою), електронне навчання використовує усі переваги сучасних персональних комп'ютерів: графічний та звуковий супровід, тривимірну графіку, анімацію тощо. На відміну від комп'ютерного навчання (коли користувач працює один з персональним комп'ютером), e-learning використовує мережеві технології для передачі результатів навчання викладачу, можливості спільної роботи, консультацій та обговорення, обмін досвідом, допомога [8, 9, 10, 11].

Користувачами e-learning можуть бути як студенти по-одиноці, школи, освітні та тренінгові установи, так і комерційні підприємства, для конкурентоспроможності яких ключове значення набуває інтелектуальний капітал – знання та досвід співробітників. Електронне навчання дозволяє знаходити для навчання зручне вікно в робочому графіку, економити на транспортних витратах, охоплювати навчальними програмами безлічі користувачів, створювати корпоративне середовище накопичення і вдосконалення знань, забезпечувати комфортний, персоналізований стиль навчання [12]. Виникає ще один напрямок розвитку e-learning – електронна освіта для працівників комерційних компаній, невідривно від виробництва. Використання e-learning диктується необхідністю вирішення проблем: реалізації потреби в первісній освіті; здійснення дистанційної освіти (початкової й післядипломної); рішення працівниками підприємства

конкретного завдання без відриву від виробництва (отримання нових знань); реалізація потреби в довічній освіті; підвищення кваліфікації перепідготовка кадрів.

Технологічний розвиток електронного навчання призвів до виникнення багаточисельних методів навчання, через які реалізується ця технологія. Кожному з методів (таблиця 1.1) характерні три аспекти: роль викладача, вид взаємодії учня та викладача і учнів між собою, вид переданого знання [13].

Таблиця 1.1 – Аналіз систем e-learning

Метод e-learning	Характеристика
Stand-alone e-learning	Оснований на навчанні та перевірці знань у автономному режимі. Застосовується у самонавчанні.
Assisted e-learning	Передбачає допомогу викладача, що знаходиться на відстані.
Сумісний e-learning	Даний метод включає в себе розширені функції Assisted e-learning з використанням online конференцій, віртуальних класів і т. п.
Неформальне навчання	За чим методом навчання не передбачає після закінчення отримання підтверджуючого документа.
m-learning	Відповідно до цього методу процес навчання відбувається завдяки мобільним технологіям.
Змішане навчання	Передбачає використання звичайного навчання наряду з різними електронними курсами.
ubiquities learning	Є переплетінням e-learning та m-learning технологій

Зважаючи на дані, що відображені у таблиці, найскладніше організувати змішане навчання [13].

Аналіз публікацій, присвячений проблемам e-learning, виявив, що для реалізації методів e-learning можуть бути використані різні комбінації інформаційних технологій (цифрові репозиторії, комп'ютерне оцінювання,

медіа технології) й побудовані на їх основі інструменти (рисунок 1.1) [13].

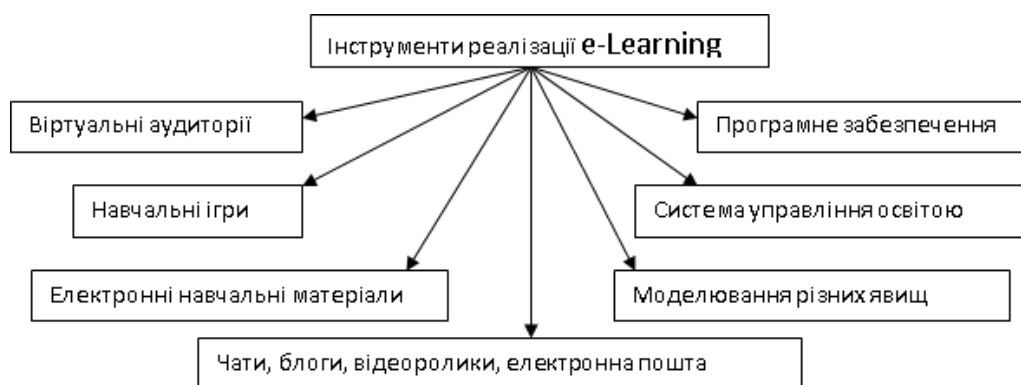


Рисунок 1.1 – Інструменти реалізації e-learning

Якість наданого контенту в середовищі e-learning є ключовим фактором в досягненні високих результатів. Контентом в e-learning середовищі є матеріали, що можуть бути розподілені й доступні через мережу. Основними категоріями контенту є: структурований гіпермедіа-контент та завдання для перевірки знань [13]. Незалежно від структури і форми контенту, процес його формування в основному має загальну структуру: створення контенту, його перетворення у формати, (необхідні для використання e-learning – платформою), публікація перетвореного контенту.

Найбільшими перевагами e-learning перед традиційним денним або заочним навчанням можна виділити гнучкість, відносно низьку вартість, відсутність додаткових витрат часу, навчання під час роботи, установка студентом власного темпу навчання при проходженні курсу, легке та швидке виправлення контенту, стійке запам'ятовування інформації і тривалий час утримання в пам'яті [14, 19].

Сучасному суспільству характерна висока активність його учасників, тому останнім часом все більшої популярності набувають такі способи отримання знань, як m-learning і u-learning. Розглянемо детальніше їх.

M-learning – це передача знань на мобільний персональний комп'ютер (смартфон, планшетний ПК) з використанням бездротових технологій зв'язку

(4G, 5G, Bluetooth, Wi-Fi, тощо). Завдяки мобільному пристрою можливо вийти в Internet, завантажити навчальні матеріали, пройти тест, бути учасником форуму. Мета m-learning – це створення більш гнучкого, доступного й індивідуального процесу навчання. Найкращою організацією контенту для мобільного навчання є надання його у вигляді інтерактивних карт знань, поділ за рівнем складності, надання малими порціями інформації в точному контексті [15].

Переваги m-learning – це економія коштів і часу; швидкий доступ до необхідної інформації будь-де, без необхідності в стаціонарному комп'ютері, використання освітніх ресурсів без необхідності в постійному виходу до мережі Internet; автоматизована доставка контенту і сервісів за індивідуальним запитом; модульність організації контенту; зручність і звичність мобільного пристрою. M-learning – це перспективний вид навчання, бо останнім часом мобільні технології відзначаються високим темпами розвитку, збільшення кількості й потужності мобільних пристроїв порад зі зменшенням вартості, що гарантує все більшу їх доступність для населення в майбутньому. Також успішність впровадження мобільного навчання являється збільшення потреби людей в швидкому отриманні знань [17, 19].

Недоліки m-learning – це використання мобільного телефону в навчанні, для якого характерна відносно висока вартість й послуг зв'язку, неергономічність деяких пристроїв через малий розмір дисплею й обмеженість клавіатури, необхідність в додатковому навчанні користувачів можливостям мобільних пристроїв, недостатня методична база для підготовки освітнього контенту [17].

Об'єднуючи середу e-learning з m-learning отримуємо середовище навчання здатне проникати всюди u-learning [16]. U-learning – все проникаюче e-learning. Таке навчання дозволяє учням, використовуючи бездротовий зв'язок будь-де завантажувати і вивчати мультимедійний освітній контент [20].

Архітектура u-learning (рис. 1.2) розвивалася з архітектури e-learning, отримавши за основу її базові складові: навчальні компоненти містять матеріали навчальних модулів; навчальні завдання; навчальні мультимедіа компоненти – допомагають пояснювати навчальний матеріал за допомогою читання або за допомогою особистої взаємодії «викладач – студент»; інструменти навчальної комунікації – методи, використовувані для комунікації між студентами і викладачами та студентами між собою; адміністративні функції – Internet орієнтовані додатки, що виконують різні адміністративні завдання [16].

Сховище даних, що містить всі компоненти e-learning – ключовий компонент архітектури u-learning. Навчальні компоненти й завдання представлені як текстова, графічна та інша мультимедійна інформація і можуть бути використанні в якості компонентів HTML-сторінок або мультимедійних додатків. Всі компоненти в залежності від їх призначення, розподіляються між технічними засобами з урахуванням особливостей браузера, платформи і технічних можливостей системи управління e-learning [16].

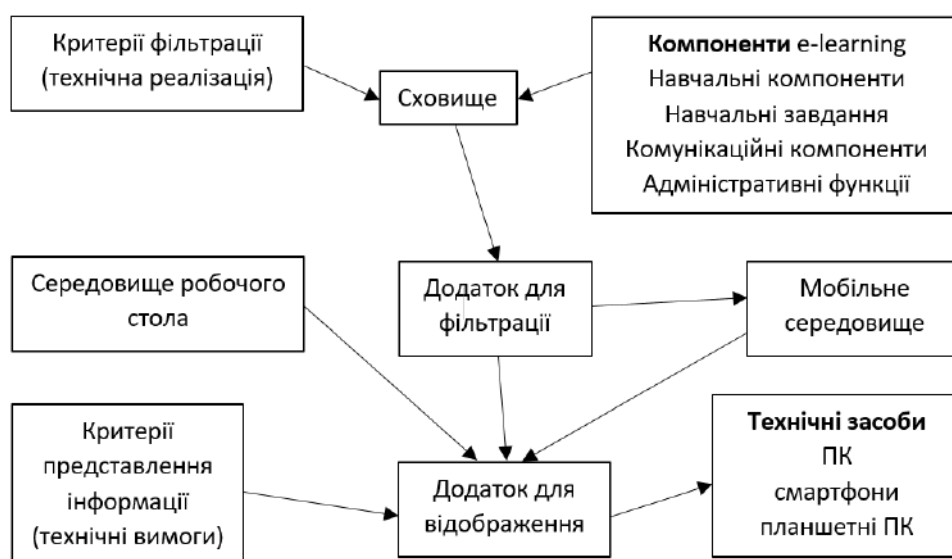


Рисунок 1.2 – Структура архітектури u-learning

Сильною стороною технології u-learning є надання найбільш широкого

спектру сучасних мобільних і комп'ютерних технологій для отримання знань. Однак, оскільки технологія u-learning організована на стику інформаційних технологій і знаходиться на вершині сучасних освітніх процесів, вона вимагає особливої організації середовища навчання. Тому для успішного впровадження технології u-learning в освітній процес потрібно вирішити питання нормативної стандартизації та методології такого виду освіти.

Холберг Б. зазначив, що електронне навчання ставить студентів у центр навчального процесу та проектування особистісно-орієнтованих програм. Він передбачає, що впровадження нових технологій спричинить за собою перетворення навчальних процесів: від провідної ролі викладача в освітньому процесі до висунення на перший план студента, і що ця трансформація дозволить студентам розвивати до ще більших висот свої навички міркування, здібності до рішення задач, пошуку інформації, комутативності, творчості [15, 26, 27].

Аналіз науково-методичної літератури виявив, що процес впровадження e-learning виявився складнішим, ніж очікувалося, і що використання переваг нових технологій неможливо без супроводу нового роду проблем [15]. Були виявлені наступні проблеми e-learning:

- проблеми критеріїв якості оцінювання електронних курсів;
- деякі студенти ще не готові до застосування інформаційних технологій з метою освіти;
- нездатність викладачів до ведення процесу підготовки в умовах e-learning;
- у навчальних закладах недостатньо фахівців в області e-learning, що здатні кваліфіковано підтримувати викладачів і студентів в процесі навчання;
- проблема ідентифікації особистості;
- відсутність контролю зовні;
- питання авторського права і як наслідок небажання викладачами викладати свої ресурси у відкритий доступ;

- великі витрати на впровадження і підтримку e-learning.

Перераховані проблеми e-learning становляться гострими тільки в випадку, коли e-learning буде реалізовано замість традиційного навчання. Однак, як би не розвивалися інформаційно-телекомунікаційні технології, електронне навчання не зможе повністю замінити традиційну форму освіти. Тому найбільш ефективними і перспективними вважається змішане навчання, засноване на об'єднанні принципів традиційного навчання із технологією e-learning. Це передбачає використання різних форм і технологій проведення навчального процесу, що робить його різноманітним і забезпечує активну участь самого студента в процесі добування знань. Перевагою застосованих в змішаному навчанні технологій є те, що вони сприяють взаємодії всіх суб'єктів освітнього процесу. E-learning включає в себе (рисунки. 1.3): технології, процеси і користувачів [23].

Система управління навчанням – LMS (Learning Management System), це платформа, в якій організовується і відбувається весь процес навчання. LMS складається з різного роду інструментів необхідних для різного роду роботи в мережі: форуми, чати, система тестування, система обміну файлами, електронна відомість, віртуальні класні кімнати, блоги, віртуальні лабораторії тощо. В LMS представлені можливості організації взаємодії між викладачем і студентами, роботи з інформаційними джерелами різного типу і можливості проектування освітнього процесу.

Якості оцінки педагогічної цінності LMS є [2, 5]:

- гнучкість – це здатність знаходити структуру, межі та змістове наповнення, яке задають суб'єкти (конкретний студент, навчальна група, конкретний викладач);
- адаптивність до існуючих освітніх технологій – LMS надає викладачеві зручні інструментальні засоби і освітні ресурси для інтеграції в навчальному процесі освітніх і інформаційних технологій;
- відкритість – це можливість інтегрування аудиторної і самостійної діяльності, забезпечувати комунікації за рамками освітньої установи.

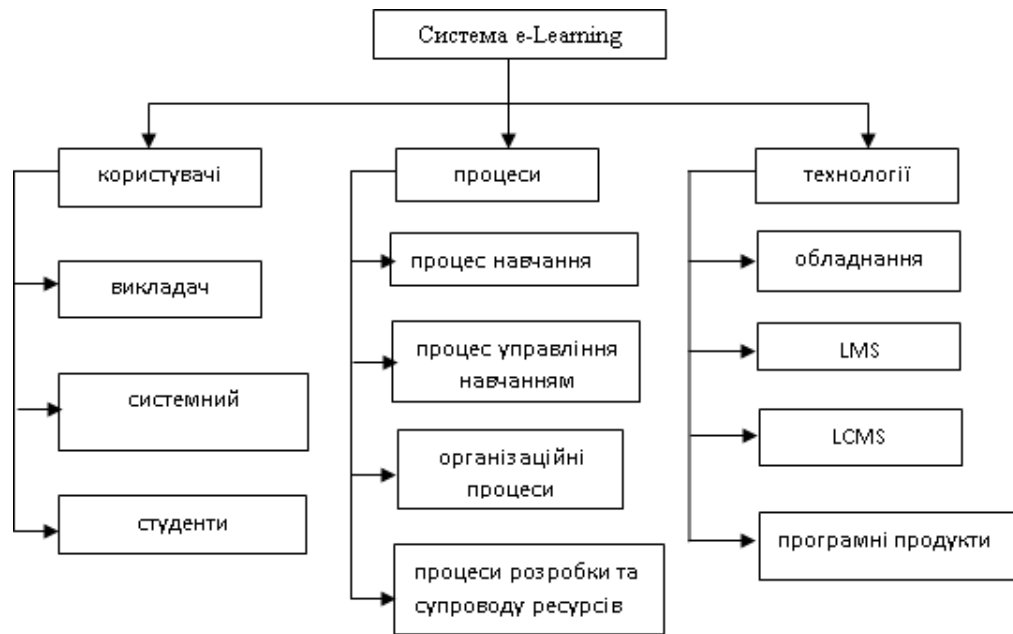


Рисунок 1.3 – Компоненти системи e-learning

Проблема вибору платформи, на якій буде побудована віртуальне навчальне середовище – ключова. Цей вибір залежить від цілого ряду чинників: які вимоги пред’являються до середовища, які функціональні характеристики повинні бути присутніми, на яких користувачів орієнтоване середовище і які засоби необхідні для придбання і підтримки необхідної платформи [30]. В даний час не існує LMS, яка однаково добре відповідала б вимогам різних дисциплін. Разом з тим, був вироблений список LMS, рекомендованих для використання в сфері електронного навчання. Виходячи з вимог до LMS і реальної ситуації, що складалася в освітній сфері, найбільш прийнятними і привабливими з точки зору простоти навчання та використаних рішень є системи Virtual Learning Environment, Moodle [20]. LMS має можливість інтегрувати різні електронні засоби навчання [9, 10, 19]. В. В. Гріншкун, С. Г. Григор’єв, С. П. Борисова виділяють наступні електронні засоби:

- сервісні програмні засоби загального призначення;
- програмні засоби для контролю і виміру рівня знань; умінь і навичок

учнів;

- електронні тренажери;
- програмні засоби лабораторій віддаленого доступу;
- інформаційно-пошукові довідкові системи;
- електронні підручники.

Для підвищення результативності самостійної роботи студентів необхідно комплексне використання перерахованих вище засобів [20, 21].

## 1.2 Аналіз гіперконвергентних мереж та порівняння із звичайними рішеннями

Гіперконвергентність (HCI) – початкова інтеграція в одне ціле двох і більше різнорідних компонентів. У вузькому сенсі гіперконвергентність – це програмно визначений підхід до управління зберіганням, об'єднуючий ресурси, обчислювальні ресурси і технології віртуалізації в одному фізичному пристрої, управління яким здійснюється як єдиною системою [12] (рисунок 1.4).



Рисунок 1.4 – Уявлення гіперконвергентної концепції

У широкому сенсі гіперконвергентність – це об'єднання

обчислювальних потужностей, мережевих ресурсів і засобів зберігання з використанням стандартних компонентів x86 в якості загальних будівельних блоків, при цьому інтелект більше не прив'язаний до окремих фізичних пристроїв – всі функції центру обробки даних виконуються як програмне забезпечення на гіпервізорі.

Ринок гіперконвергентних систем є перспективним й активно розвивається (рисунок 1.5.) [11].



Рисунок 1.5 – Ринок гіперконвергентних систем

Гіперконвергенція пропонує повноцінну альтернативу традиційної моделі виконання виділених систем зберігання. Вона передбачає відмову від окремого рівня зберігання, що розглядається, як одне з ключових переваг HCI. Гіперконвергенція дозволяє спростити архітектуру обчислювальних систем за рахунок відмови від окремого рівня зберігання.

На відміну від колишніх архітектур, гіперконвергентна не призводить до утворення ізольованих острівців зберігання. Щоб кожен сервер міг скористатися ресурсами зберігання на інших системах, вона опирається на програмно визначені рішення в області зберігання (Software-Defined Storage, SDS).

Гіперконвергентна система являє собою об'єднані в одному корпусі сервер, систему зберігання даних і мережевий комутатор. Проте найголовнішою частиною є адаптоване ПЗ, включаючи контролери. Це такий масовий пристрій, що володіє найширшою сумісністю і універсальне в

застосуванні. Гіперконвергентні системи користуються все більшою популярністю. Багато компаній малого і середнього рівня, які не мають великих штатів ІТ-фахівців, за допомогою подібних систем дуже швидко розгортають масштабовані системи зберігання і обробки даних [3,5].

Гіперконвергентні рішення виділяються покращеннями на рівні програмного контролера, що дозволяє легко їх масштабувати. Для збільшення ємності та продуктивності потрібно додати новий блок[3,5]. Замість посилення потужності за рахунок збільшення числа дисків, кількості пам'яті або процесорів, продуктивність збільшується за рахунок додавання більшої кількості модулів.

Таким чином, гіперконвергентна інфраструктура – це інфраструктура, в якій обчислювальні потужності, сховища, сервери, мережі об'єднуються в одне ціле за допомогою програмних засобів, а управління ними відбувається через загальну консоль адміністрування [4]. З цієї причини замість команди ІТ-фахівців для управління сховищами даних і серверним обладнанням часом досить одного системного адміністратора.

Гіперконвергентні системи зазвичай складаються з декількох фізичних модулів, що об'єднуються в горизонтально масштабований кластер. Кожен з них містить обчислювальне ядро, ресурси зберігання, мережеві компоненти і гіпервізор. Наявність гіпервізора не є обов'язковим, але він є у всіх основних продуктах відомих виробників, причому часто можна вибрати з двох і більше гіпервізорів.

Окремий пристрій має від одного до чотирьох вузлів, кожен з яких представляє собою самостійний сервер з процесором і пам'яттю в загальному шасі. Гіперконвергентні кластери зазвичай містять від 4 до 64 вузлів, хоча деякі виробники не вказують конкретних меж масштабування. Для того, щоб вузли могли спільно виготовляти ресурси зберігання, застосовується програмне забезпечення для створення віртуальної мережі зберігання або кластерна файлова система. Програмне забезпечення для реалізації

гіперконвергентної інфраструктури може пропонуватися як окремо, так і встановленим на фізичні пристрої.

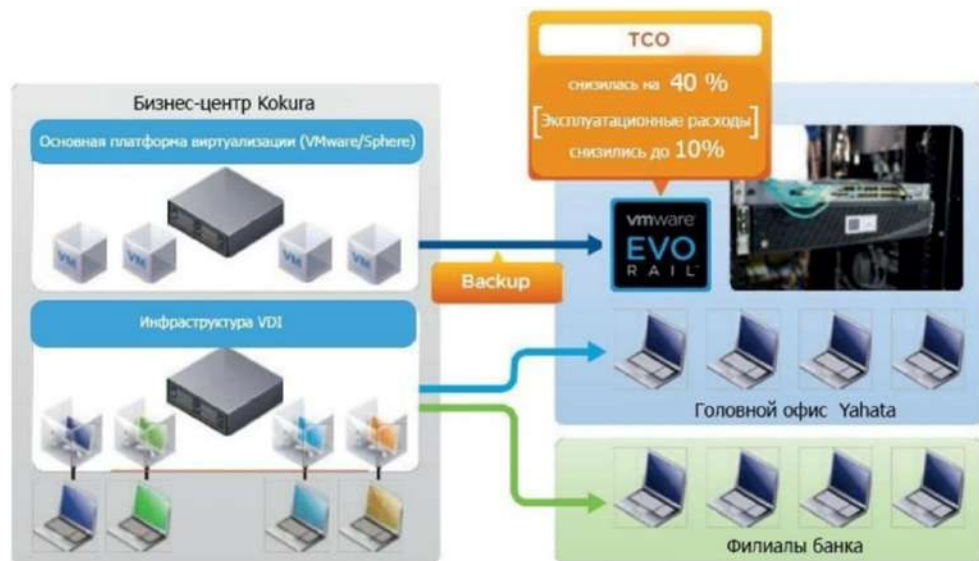


Рисунок 1.6 – Скорочення витрат банку Kokura, після упровадження мережі на гіперконвергентній інфраструктурі

В основному на ринку пропонуються повністю готові самодостатні обчислювальні системи. Найчастіше постачальники розробляють своє власне програмне забезпечення для реалізації функцій зберігання і управління, або ці функції виконуються як віртуальна машина на використуваному гіпервізорі. У багатьох продуктах використовується віртуальне пристрій зберігання (Virtual Storage Appliance, VSA) для об'єднання ресурсів зберігання до загального пулу. Інший підхід полягає у використанні платформи VMware EVO: RAIL, в якій функції зберігання та управління інтегровані в відповідний гіпервізор.

Дозволяючи позбавитися від виділеної мережі зберігання, гіперконвергентна інфраструктура реалізує віртуальну мережу зберігання (Virtual SAN, VSAN). Зазвичай така SAN передбачає використання VSA у вигляді віртуальної машини. Віртуальний пристрій зберігання надає функції контролера зберігання для гіпервізора в кластері. Ресурси зберігання

фізичного вузла надаються VSA. Група VSA створює віртуальну мережу зберігання. Винятком є реалізація VSAN компанії VMware. Сервіси зберігання розташовані в ядрі vSphere, так що рішення повністю інтегрується в vSphere.

Порівняємо гіперконвергентні системи з конвергентними й звичайними рішеннями. Важливою відмінністю між гіперконвергентними й конвергентними системами є те, що в конвергентній інфраструктурі кожен компонент в будівельному блоці є дискретним і може бути використаним окремо [33]. Що стосується гіперконвергентної інфраструктури, то це програмно-обумовлена технологія, тому всі компоненти інтегровані [33]. Такі системи побудовані найчастіше на базі систем зберігання даних і в якості обчислювального модуля використовується окремий сервер x86, який підключено за допомогою каналу 10 Гбіт/с.

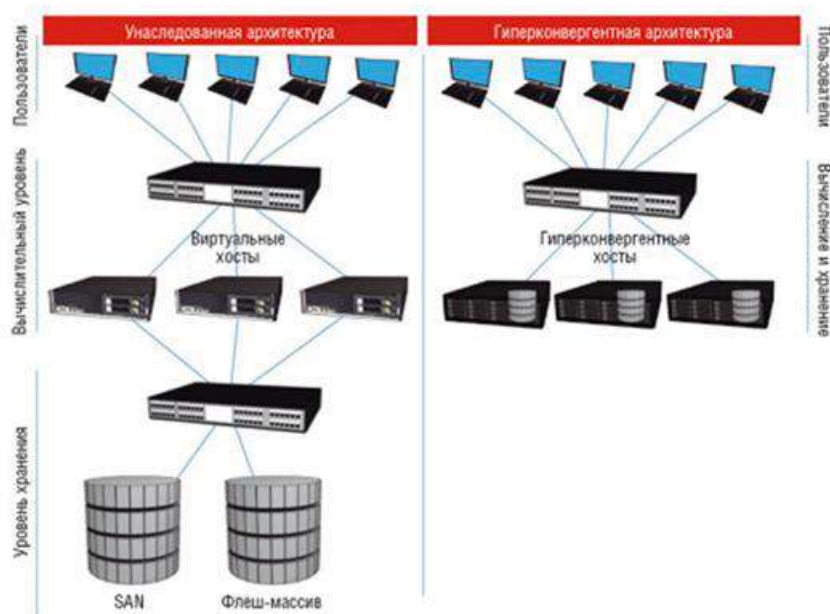


Рисунок 1.7 – Порівняння гіперконвергентної архітектури із звичайним успадкованим рішенням

Різниця між гіперконвергентними системами і просто серверами, оснащеними дисковими масивами, полягає в інжинірингу та ПЗ [36].

Гіперконвергентні рішення відрізняються поліпшеннями на рівні програмного контролера, і це дозволяє без зайвих зусиль масштабувати дані системи (рисунок 1.7). Чим більше пристроїв додається, тим більшу продуктивність і ємність виходить на виході. Замість збільшення потужності за рахунок додавання більшої кількості дисків, пам'яті або процесорів з'являється можливість збільшувати продуктивність за рахунок додавання більшої кількості модулів.

Крім спрощеної архітектури в гіперконвергентній системі простіша модель адміністрування. Гіперконвергентні системи управляються через єдину систему. Замість великої кількості додатків і групи ІТ-адміністраторів для управління масивом даних, віртуалізацією і серверним обладнанням, повним гіперконвергентним стеком може керувати всього одна команда (іноді один системний адміністратор).

Переваги гіперконвергентних систем [34] є наслідком суми двох компонентів: апаратної складової та програмного забезпечення:

- простота архітектури та управління;
- скорочення непродуктивних витрат;
- спрощена взаємодія з користувачами в умовах середовища з високим рівнем візуалізації;
- низька вартість;
- висока автоматизація і управління на основі заданих правил дозволяють гнучко і легко управляти розподілом ресурсів і робочого навантаження;
- скорочення числа керованих систем.

ОР та ресурси зберігання даних, об'єднуються гіперконвергентним вузлом. Така архітектура дає спроможність зменшити число окремих пристроїв. Підвищення ефективності роботи ГКС, забезпечується завдяки зменшенню витрат на обслуговування, та збільшення обчислювальних вузлів за цей рахунок (рисунок 1.7). Гіперконвергентні пристрої облегшують розгортання та обслуговування систем такого типу.

Гіперконвергентні системи розраховані не на вертикальне, а на горизонтальне масштабування.

Майже всі гіперконвергентні пристрої містять флеш-накопичувачі. Вони відрізняються від флеш-масивів тим, що знаходяться всередині фізичного сервера (рисунок 1.8), забезпечуючи меншу – в порівнянні з зовнішніми системами зберігання – затримку.

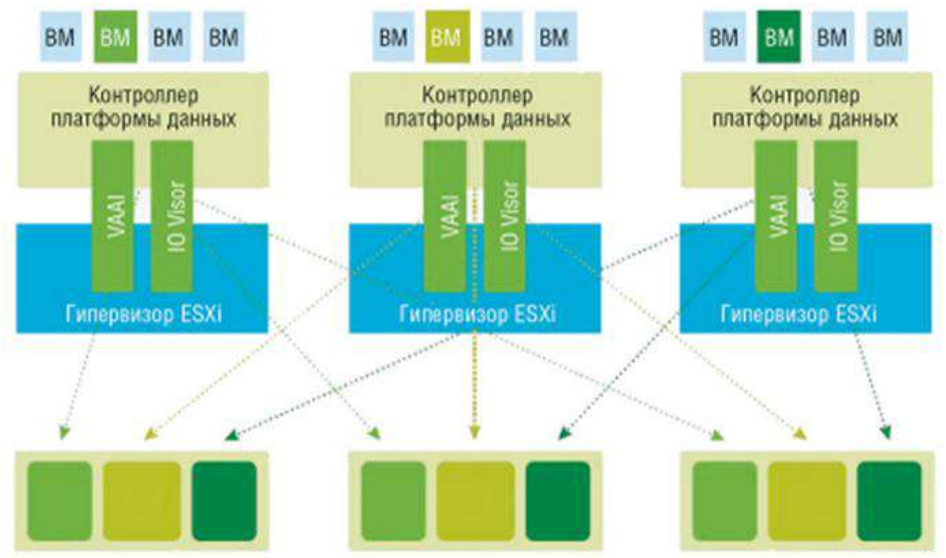


Рисунок 1.8 – Розподіл даних між усіма вузлами в кластері Cisco

Недоліки гіперконвергентних систем полягають в неможливості гранулярного поновлення або налаштування системи. Зростання сховища даних і підвищення продуктивності є критично важливими пунктами для будь-якої компанії. Якщо ємність системи зберігання даних кластера практично закінчилася, однак в обчислювальних ресурсах ще є запас, то вам все одно доведеться збільшити загальну обчислювальну потужність шляхом додавання нових пристроїв. Точно також, якщо потрібно налаштувати конфігурацію дисків сховища для певної програми, це може стати проблемою. Виняток полягає в побудові власної платформи. Можливо додати тільки систему зберігання даних або тільки обчислювальну потужність при виборі підходу VSAN.

### 1.3 Аналіз методів управління мережевими ресурсами

Управління комп'ютерною мережею – це виконання безлічі функцій необхідних для контролю, планування, виділення, впровадження, координації та моніторингу ресурсів комп'ютерної мережі, мереж зв'язку, позначаючи супровід і адміністрування цих мереж на верхньому рівні [37].

Воно включає в себе виконання таких функцій як початкове мережеве планування, розподіл частот, приречення маршрутів трафіку для підтримки балансування навантаження, розподіл криптографічних ключів, управління конфігурацією, стійкістю до відмов, безпекою, продуктивністю і обліковою інформацією.

Міжнародна організація по стандартизації описала FCAPS модель, в якій відображені ключові функції адміністрування та управління мережами: управління відмовами, управління конфігурацією, облік роботи мережі, управління продуктивністю, управління безпекою [37].

Управління продуктивністю служить для подання статистики роботи мережі в реальному часі, мінімізації заторів і вузьких місць, виявлення створених тенденцій і планування ресурсів для майбутніх потреб.

Усі засоби управління мережесим трафіком – мають своєю метою підвищити ефективність роботи та якість обслуговування інформаційних додатків, використовують відповідні засоби і можливості каналного, мережевого і транспортного рівнів транспортної мережі (ТМ). Найбільш поширені способи попередження перевантажень в мережі ґрунтуються на відкиданні пакетів, не суттєво впливаючих на рівень якості переданої інформації [3, 7]. Перший і широко використовуваний в цьому випадку алгоритм управління заснований на відкиданні всіх осередків, наступних за втраченим осередком, який належить даному IP пакету. Модифікація цього механізму управління є схема раннього відкидання комірок. У цьому випадку використовується алгоритм короткострокового передбачення рівня заповнення буфера комутатора і, якщо довжина черги перевищує певний рівень або деякий критичний поріг, то система управління приймає рішення

про відкидання всіх осередків, що належать певній кількості TSP сегментів [37]. Така схема управління трафіком дозволяє підвищити продуктивність транспортних сполучень. Однак вибір обґрунтованого значення величини критичного порога заповнення буфера вимагає аналізу характеру, в загальному випадку, нестаціонарних потоків даних і створення ефективних алгоритмів передбачення стану каналів.

Друга схема відкидання комірок заснована на програмному управлінні ресурсами мережі і вимагає апріорного їх резервування для кожного з утворених віртуальних з'єднань. Основна перевага такого підходу полягає в тому, що можна гарантувати нульовий рівень втрат осередків, викликаних переповнення пристроїв буферизації. Однак обсяг ресурсів буферної пам'яті, який дозволяє ефективно використовувати таку схему управління потоком, залежить від величини затримки при поширенні пакетів і флуктуації пропускної здатності лінії зв'язку.

Як показали дослідження, така схема має недостатні екстраполяційні можливості, але дозволяє ефективно управляти перевантаженнями на вельми коротких проміжках часу. Її основний недолік полягає в необхідності використання значних обсягів пам'яті для буферизації даних. Досвід розробки додатків для мережі Internet показує, що істотного підвищення ефективності використання буферної пам'яті можна досягти за рахунок застосування алгоритмів прогнозування при статистичному об'єднанні трафіка.

Якість функціонування системи передачі залежить від величини затримки при передачі пакетів і наявності каналу підтвердження. В останньому випадку в моделі системи необхідно додати механізм управління у вигляді контуру зворотного зв'язку. При підтвердженні кожного окремого пакета процес формування сигналу зворотного зв'язку можна пов'язати зі зміною стану мережевого з'єднання, яке характеризується номером очікуваного до прийому пакета. Зауважимо, що пакети можуть передаватися по віртуальному каналу без отримання підтвердження, поки їх число в каналі

менше деякого числа. Кращим можна вважати алгоритм управління, який забезпечує мінімальну затримку при заданій продуктивності або найбільшу продуктивність при заданому часу затримки.

Далі нас цікавлять системи, для яких характерна незалежність територіального положення користувачів з періодичним зростанням інтенсивності запитів, а саме система обслуговування університетського e-learning [3, 9]. На якість функціонування гіперконвергентної системи негативно впливають непередбачені аварійні ситуації, як з апаратними, так і з програмними засобами підтримки системи на її вузлах. При виникненні таких ситуацій гіперконвергентний сервер повинен оперативного перерозподілити наявні ресурси з метою зменшення негативного впливу проблемної ситуації [4, 11]. Далі сформоване завдання оперативного перерозподілу завдань по вузлах гіперконвергентної мережі в процесі їх розподіленої обробки. Для її вирішення необхідно знайти таке розбиття множини завдань, при якому середня затримка пакета даних в мережі приймала мінімальне значення.

Формалізація задачі: цільова функція задачі пошуку оптимального розбиття множини завдань  $Q$ , обновлюваних в обчислювальній мережі, на підмножини і їх розподілу по вузлах  $u_a \in Y$ , визначається виразом:

$$F(y) = \frac{1}{u_{Q_{max}} \sum_{b=1}^{h_Q} \sum_{a=1}^{h_y} m_{Q_{b,a}}}, \quad (1.1)$$

де  $u_{Q_{max}}$  – незалежна від розподілу  $y$  величина, яка визначає максимальну сумарну інтенсивність обміну задач з вузлами обчислювальної мережі відповідно до виразу  $u \sum_{b=1}^{h_Q} \sum_{i=1}^{h_y} u_{Q_{b,i}}$ ;

$h_Q$  – число оброблюваних в обчислювальній мережі завдань;

$h_y$  – число вузлів обчислювальної мережі;

$u_{Q_b,i}$  – інтенсивність обміну завдання  $z_b \in Z$  з вузлом  $y_i \in Y$ ;

$m_{z_b,a}$  – необхідний обчислювальний ресурс вузла  $y_a \in Y$ , необхідний

для виконання підзадачі завдання  $z_b \in Z$ ;

$s_{z_b,a}$  – штраф при розподілі підзадачі завдання  $z_b \in Z$  на вузол  $y_a \in Y$ ,

який визначається виразом  $s_{z_b,a} = \sum_{i=1}^{h_y} (u_{z_b,i} \cdot h_{w_{a,i}}) / \phi_{z_b}$ ;

$h_{w_{a,i}}$  – довжина найкоротшого маршруту між вузлами  $y_a$  і  $y_i$ , що

визначається числом каналів передачі даних, що входять в цей маршрут;

$\Phi_{Q_b}$  – необхідний обчислювальний ресурс для обробки завдання  $z_b \in Z$ .

Шуканий розподіл  $\gamma$  має відповідати таким вимогам [34]:

$$1) \forall y_a \in Y \left| \sum_{b=1}^{h_z} m_{z_b,a} \leq \phi_{y_a} ; \right.$$

$$2) \forall z_b \in Z \left| \sum_{a=1}^{h_y} m_{z_b,a} \leq \phi_{z_b} ; \right.$$

$$3) \sum_{a=1}^{h_y} \phi_{y_a} \geq \sum_{b=1}^{h_z} \phi_{z_b} ;$$

$$4) s_{z_b,a} \geq 0, m_{z_b,a} \geq 0 \text{ для } 1 \leq a \leq h_y, 1 \leq b \leq h_z,$$

де  $\phi_{y_a}$  – допустимий обчислювальний ресурс вузла .

З урахуванням наведених умов, завдання пошуку оптимального розбиття множини завдань  $Q$ , оброблюваних в обчислювальній мережі, на підмножини і їх розподілу по вузлах  $y_a \in Y$  може бути сформульована таким чином.

Нехай задана множина завдань  $Q$  і вузлів  $Y$  обчислювальної мережі, які визначаються кортежами:

$$\langle Z, \phi_z, U_z \rangle \text{ і } \langle Y, \phi_y, H_w \rangle, \quad (1.2)$$

де  $\phi_z = (\phi_{z_1}, \dots, \phi_{z_{h_z}})$  – вектор необхідних обчислювальних ресурсів

для обробки множини завдань  $Z$ ;

$U_z = \left\| \| u_{z_b, i} \| \right\|$  – матриця інтенсивності обміну завдань множини  $Z$  з

вузлами множини  $Y$ ;

$\phi_y = (\phi_{y_1}, \dots, \phi_{y_{h_y}})$  – вектор доступних обчислювальних ресурсів

множини вузлів  $Y$  обчислювальної мережі;

$H_w = \left\| \| h_{w_{a,i}} \| \right\|$  – матриця довжин найкоротших маршрутів між кожною

парою вузлів обчислювальної мережі  $y_a$  і  $y_i$ ,  $1 \leq a \leq h_y$ ,  $1 \leq i \leq h_y$ .

Необхідно знайти такий розподіл  $\gamma$ , що задовольняє умовам 1 – 4, щоб вираз (1.1) приймав мінімальне значення.

При побудові алгоритму рішення сформульованої задачі зручно прийняти умову збалансованості: загальний сумарний доступний обчислювальний ресурс вузлів множини  $Y$  дорівнює загальному сумарному необхідному обчислювальному ресурсу завдань множини  $Z$ , тобто:

$$\sum_{a=1}^{h_y} \phi_{y_a} = \sum_{b=1}^{h_z} \phi_{z_b}. \quad (1.3)$$

З цією метою необхідно ввести фіктивний  $(h_y + 1)$ -й вузол з доступним обчислювальним ресурсом  $\phi_{y_{h_y+1}}$  або фіктивне  $(h_z + 1)$ -е завдання з необхідним обчислювальним ресурсом  $\phi_{z_{h_z+1}}$ , при яких

$$\sum_{a=1}^{h_y+1} \phi_{y_a} = \sum_{b=1}^{h_z+1} \phi_{z_b}, \quad (1.4)$$

і прийняти штраф:

$$s_{z_{h_z+1,a}} = 0, 1 \leq a \leq h_y; s_{z_b, h_y+1} = \max_{\substack{1 \leq b \leq h_z \\ 1 \leq a \leq h_y}} s_{z_b, a}, 1 \leq b \leq h_z.$$

Роздивимося поетапно метод, який здійснює пошук оптимального розподілу завдань по вузлах обчислювальної мережі, що забезпечує розбиття множини завдань, що вирішуються в мережі на підмножини і їх розподіл по вузлах мережі, що дозволяє мінімізувати середню затримку пакета даних в процесі розподіленої обробки завдань.

Правило мінімальних штрафів. Базове розбиття множини завдань, що вирішуються в ОМ, на підмножини і їх розподіл по вузла мережі, визначається матрицею  $M_Z^{(Y)}$ , заповненою за правилом мінімальних штрафів [4], елемент  $m_{z_b, a}$  якої визначає обчислювальний ресурс вузла  $y_a \in Y$ , виділений для обробки підзадачі завдання  $qz_b \in Z$ . Для побудови базового розподілу формується матриця штрафів  $S_Q$ , елемент  $s_{z_b, a}$  якої визначає штраф при виділенні підзадачі завдання  $z_b$  вузлом  $y_a$  одиниці обчислювального ресурсу.

Суть правила мінімальних штрафів полягає в тому, що заповнення матриці  $M_Z^{(Y)}$  починається послідовно з елемента  $m_{z_b, a}$ , якому в матриці штрафів  $S_Z$  відповідає найменше значення штрафу  $s_{z_b, a}$ . Елементу  $m_{z_b, a}$  присвоюється значення  $\min(\phi_{z_b}, \phi_{y_a})$ . Змінюються значення необхідного обчислювального ресурсу для обробки завдання  $z_b$  і доступного обчислювального ресурсу вузла  $y_a$ :

$$\phi_{z_b} = \phi_{z_b} - \min(\phi_{z_b}, \phi_{y_a});$$

$$\phi_{y_a} = \phi_{y_a} - \min(\phi_{z_b}, \phi_{y_a}).$$

З подальшого розгляду в матриці  $S_Z$  виключається або рядок, що відповідає заявці  $z_b$ , потреба в обчислювальних ресурсах якої повністю задоволена, або стовбець, відповідний вузлу  $u_a$ , доступний обчислювальному ресурсу, що повністю витрачений, або стовбець і рядок, якщо повністю витрачений доступний обчислювальний ресурс вузла  $u_a$  і задоволена потреба в обчислювальних ресурсах заявки  $z_b$ .

З решти елементів матриці  $S_Z$  знову послідовно вибираються елементи з найменшим значенням штрафу, і процес розподілу обчислювальних ресурсів триває до тих пір, поки потреби всіх заявок множини  $Z$  не будуть задоволені в обчислювальних ресурсах. Незаповненим елементам матриці  $M_Z^{(\gamma)}$  присвоюються нульові значення.

Формування системи потенціалів. Для перевірки базового розподілу на оптимальність будується система потенціалів. Систему потенціалів можна побудувати тільки для невиродженого плану розподілу. План розподілу  $\gamma$  є невиродженим, якщо число елементів матриці  $p_{y_a}$ , відмінне від нуля, так само дорівнює  $h_z + h_y - 1$  [5]. При побудові базового розподілу може виявитися, що ненульових елементів матриці  $M_Z^{(\gamma)}$  менше, ніж  $h_z + h_y - 1$ , тобто план розподілу  $\gamma$  є виродженим. В цьому випадку нульовим елементам матриці  $M_Z^{(\gamma)}$  з найменшим значенням  $p_{z_b}$  послідовного присвоюються значення умовного базисного нуля до тих пір, поки число елементів матриці  $M_Z^{(\gamma)}$ , відмінних від нуля, не стане дорівнювати  $h_z + h_y - 1$ .

Кожному вузлу  $u_a$  ставиться у відповідність потенціал  $p_{y_a}$ , а кожній заявці  $Q_b$  – потенціал  $p_{z_b}$ . Потенціали  $p_{y_a}$  і  $p_{z_b}$  вибираються таким чином, щоб для кожного ненульового елемента матриці  $M_Z^{(\gamma)}$  виконувалась умова  $p_{z_b} + p_{y_a} = s_{z_b,a}$ , а для нульового елемента матриці  $M_Z^{(\gamma)}$  –  $p_{z_b} + p_{y_a} \leq s_{z_b,a}$

умова. Так як число всіх потенціалів дорівнює  $h_z + h_y$ , а відмінних від нуля елементів матриці  $M_z^{(y)} - h_z + h_y - 1$ , то для визначення значень  $p_{z_b}$  і  $p_{y_a}$  вирішується система рівнянь  $p_{z_b} + p_{y_a} = s_{z_b,a}$  з  $h_z + h_y$  невідомими, для чого одному з невідомих присвоюється довільне значення, і система має єдине рішення.

Даний метод пошуку оптимального розподілу завдань по вузлах обчислювальної мережі, який забезпечує раціональне розбиття множини завдань, що вирішуються в обслуговуваній мережею системі, на підмножини і їх розподіл по вузлах мережі в процесі розподіленої обробки, що дозволяє мінімізувати середню затримку пакета даних в ній. Якість розподілу визначається за допомогою спеціальної цільової функції, що враховує штрафи при наявності затримок пакетів. Розглядався перехід до збалансованої системи. Початковий перерозподіл визначився за допомогою правила мінімальних штрафів. При виконанні циклу наближення розподілу до оптимального формувалася система потенціалів і розраховувалася базова оцінка оптимізації функціоналу. На кожному циклі поліпшення розподілу використовувалося правило пошуку оптимізуючого контуру. Отриманий розподіл завдань з урахуванням збоїв, що виникли, є раціональним і зменшує середню затримку пакета даних в гіперконвергентної системі.

#### 1.4 Аналіз методів моделювання інформаційної структури

Для аналізу інформаційної структури мережі досить часто використовують математичну модель у вигляді мереж масового обслуговування (ММО) [4]. Отже, роздивимося ММО. ММО – це система, яка виконує обслуговування заявок, що надходять до неї [42]. Основними елементами системи є вхідний канал обслуговування та вихідний потік заявок, загальний потік заявок та черга заявок. Надходять заявки на обслуговування у випадковій або постійній інтервали часу. Розглянемо їх

детальніше. При випадковому характері потоку, обслуговування може призвести до черги на вході ММО. Або канали зв'язку можуть простоювати та бути недовантаженими взагалі. В ситуації надходження заявки коли всі прилади зайняті, остання потрапляє у буфер і чекає початку обслуговування. Заявки, які знаходяться у буфері і складають чергу для обслуговування. Коли всі задані буфера зайняті, заявці відмовляється в обслуговуванні і втрачається.

Процес роботи мережі масового обслуговування - це випадковий процес із дискретним станом і неперервним часом [12]. ММО може змінювати стан, завдяки стрибку в момент реалізації подій. Вихідний потік - це заявки, що вже обслуговані.

Всі мережі маєть пропускну здатність. Вона дозволяє справлятися з потоком заявок та встановленими вимогами.

Головна задача масового обслуговування - це побудова моделей, які об'єднують умови, щоби були задані для роботи.

Класифікація ММО прозводиться за кількістю каналів (одноканальні та багатоканальні) та за наявністю черг (з відмовами та з очікуванням).

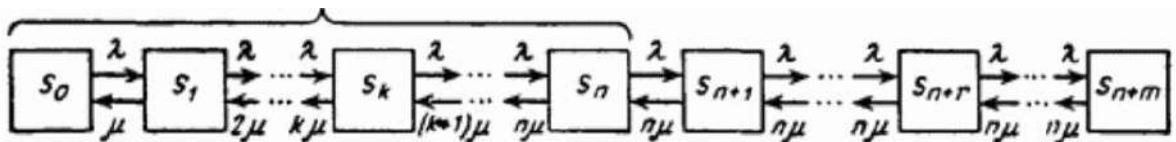


Рисунок 1.9 – Граф станів багатоканальної ММО з очікуванням

До основних характеристик ММО відносять ймовірність втрати заявки (ймовірність відмови. Залежить від інтенсивності потоку заявок і інтенсивності обслуговування. Ця величина дорівнює ймовірності того, що всі обслуговувані канали зайняті; ймовірності того, що зайнята деяка кількість каналів обслуговування; середнє число зайнятих каналів. Також цей показник характеризує ступінь завантаження системи; середнє число каналів,

вільних від обслуговування; коефіцієнт простою каналів; коефіцієнт завантаження каналів [11].

Також для систем з очікуванням використовують додаткові характеристики: середній час очікування заявки в мережі; ймовірність того, що час перебування в черзі не перевищить якої-небудь певної величини; середня довжина черги; середнє число заявок у сфері обслуговування; ймовірність того, що число вимог в черзі, які очікують початку обслуговування, більше деякої заданої величини [8].

У першому розділі було розглянуто сучасні технології розробки e-learning (електронного навчання), розглянуто актуальність і необхідність технології, методи e-learning, інструменти реалізації e-learning, компоненти e-learning, розглянуто структуру архітектури загально проникаючого навчання (u-learning), як похідної від системи e-learning. Також було виявлено проблеми притаманні електронному навчанню.

Приділено увагу гіперконвергентним мережам. Дано визначення гіперконвергентності, відповідно до нього розгорнуто тему гіперконвергентних мереж, надано їх концепцію, проведено порівняння із конвергентними й звичайними мережами, виявлено переваги, економічну ефективність та перспективність використання. Розглянуто переваги і недоліки гіперконвергентних мереж, виявлено, що переваги гіперконвергентних мереж значно випереджають їх недоліки, усе це підкреслено реальними випадками використання.

Розглянуто методи управління мережевими ресурсами. Дано відповідь на питання: що із себе представляє управління комп'ютерною мережею. Далі надано мету управління мережевими ресурсами, та наведено деякі методи управління. Виділено метод управління завантаженням гіперконвергентної мережі, який розподіляє завдання по вузлах обчислювальної мережі.

Розглянуто метод моделювання інформаційної структури, а саме мережі масового обслуговування із поясненням суті метода, які види мереж роздивляється і які параметри для оцінки мережі здатен видавати.

## 2 МАТЕМАТИЧНА МОДЕЛЬ ПРОЦЕСУ СИНТЕЗУ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Гіперконвергентна мережа інформаційної системи E-learning – це головний аспект, який грає значущу роль у запитах системи. Для управління та варіантів побудови ГКС необхідно проаналізувати її майбутню структуру [5]. Аналіз структури в каналах зв'язку мережі необхідний для визначення потрібних параметрів потоків даних, це потрібно для оцінки навантаження структурних елементів мережі. Потоки даних e-learning формуються завданнями застосунків. Вони генерують ще трафік між собою та запускаються на вузлах мережі. Для проведення аналізу мережі потрібні відомості про структуру, транзакції та застосунки e-learning, їх взаємодія та розміщення на вузлах ГБМ [6], що потребує розробки відповідної математичної моделі.

### 2.1 Аналіз необхідності моделювання інформаційної інфраструктури

Проблеми аналізу структури ГБМ: відсутній єдиний підхід до формування структури; залежність структури мережі від параметрів ЕОР; відсутні відпрацьовані методи формального опису структури мережі, які можна використати при проведенні розрахунків [2]. Все це вимагає нових принципів аналізу структури мережі, незалежних від технології створення ГБМ. Досить часто для рішення такої задачі використовують математичну модель у вигляді мереж масового обслуговування (ММО) [4]. Однак, такий підхід має наступні складності [4]: ММО вимагає розрізнення потоків запитів і даних; завантаження кожного каналу визначається інтенсивністю потоків даних усіх типів, а не тільки потоків запитів; застосунок може багаторазово взаємодіяти з іншими додатками на інших вузлах мережі, що ускладнює процес аналізу; передача по каналах зв'язку транзитних потоків; множина непотрібних результатів аналізу ММО. Зазначені особливості ММО

призводять до нехтування деякими параметрами мережі, що призводить до втрати точності при моделюванні [4]. Отже, маємо основні принципи аналізу ГБМ:

- головна мета аналізу є дослідження потоків даних в мережі – це основний фактор, що впливає на всі її характеристики;
- основою аналізу і формування структури мережі є додатки, що виконуються та взаємодіють;
- при аналізі необхідно узгодити параметри додатків з можливостями мережі.

У зв'язку з цим доцільно застосувати підхід, заснований на дослідженні взаємодії додатків як незалежних джерел і приймачів даних. В цьому випадку можливо побудувати інформаційну модель системи, а потім, в залежності від розміщення додатків по вузлах мережі, визначити параметри потоків даних між вузлами мережі, тобто побудувати технічну модель мережі. При цьому повністю враховуються всі взаємодії між застосунками. Також, є можливість аналізу складних мережевих структур шляхом декомпозиції на підмережі. Для аналізу структури і розрахунку характеристик ГБМ необхідно визначити правила її опису, для побудови моделі для розрахунку завантаження вузлів і каналів мережі. Оскільки мережа створюється для підтримки інформаційного забезпечення e-learning, для побудови мережі необхідно знати інформаційну структуру мережі, яка визначає інформаційні потоки між вузлами на яких встановлено програмне забезпечення e-learning. Інформаційна структура мережі, в даному випадку, це сукупність інформаційних ресурсів системи (джерела і приймачі), розміщених на вузлах мережі й інформаційні потоки між вузлами, що виникають при транзакціях EOP. Враховуючи інформаційну структуру мережі можна приймати рішення про організацію каналів зв'язку між вузлами мережі, визначати необхідні параметри каналів зв'язку і мережевого обладнання, тобто формувати її технічну структуру.

Отже, для повного аналізу структури ГБМ необхідно провести аналіз складових її інформаційної та технічної структур і зв'язати результати

аналізу.

## 2.2 Модель основних інформаційних взаємозв'язків у середовищі e-learning

На рисунку 2.1. представлено узагальнену структурну схему основних інформаційних взаємозв'язків у середовищі e-learning. Головний управляючий вузол ГКС це гіпервізор, його функція заключається в тому, що він приймає транзакції та запускає відповідні до них додатки, які умовно поділяються на 2 групи. Одні (група QA) працюють безпосередньо з інформаційними вузлами та інші (група QB) здійснюють обмін з розподіленим сховищем даних.



Рисунок 2.1 – Інформаційні взаємозв'язки e-learning

Окреслимо множини, що залучені для моделювання інформаційних взаємозв'язків у середовищі e-learning:

$M_U$  –користувачі e-learning,  $\dim M_U = U$ ;

$M_N$  –вузли ГКС,  $\dim M_N = N$ ;

$M_A$  –додатк e-learning,  $\dim M_A = A$ ;

$M_E$  –транзакції e-learning,  $\dim M_E = E$ ;

$M_D$  –фрагменти сховища даних,  $\dim M_D = D$ .

Окреслене значення транзакцій  $e \in M_E$  ( $e = \overline{1, E}$ ) задає такий кортеж:

$$E_e = \langle \overline{a_e}, \overline{d_e}, \overline{u_e}, \overline{w_e} \rangle,$$

де  $\overline{a_e} = (a_{e1}, \dots, a_{eA})$  – булевий вектор інформаційних додатків e-learning,

$A_e = (a_{ea}); \overline{d_e} = (d_{e1}, \dots, d_{eD})$  – булевий вектор інформаційних фрагментів сховища

$D_e = (d_{ed}); \overline{u_e} = (u_{e1}, \dots, u_{eU})$  – булевий вектор користувачів, що запускають транзакцію;

$\overline{w_e} = (w_{ij})$  – булева матриця послідовності запуску застосунків,  $i, j = 1, A$ .

Параметри застосунку «а» транзакції «е» задаються кортежом:

$$A_{ae} = \langle \overline{\lambda_{ae}}, \overline{\beta_{ae}} \rangle,$$

де  $\overline{\lambda_{ae}} = (\lambda_{ae1}, \dots, \lambda_{aeD})$  – обсяги даних для застосунку «а» транзакції «е», котрі знаходяться у відповідному фрагменті сховища;

$\beta_{ae} = (\beta_{ae1}, \dots, \beta_{aeA})$  – обсяги даних для обміну з іншими застосунками при виконанні транзакції «е».

Булева матриця задає розміщення застосунків по вузлах ГКС  $G = (g_{an})$ , юзерів - булева матриця  $H = (h_{un})$ . Матрицею:  $S = (s_{an})$  визначається розміщення фрагментів сховища по вузлах. Визначемо модель наступним кортежем множин та вищезазначених матриць,

$$\mathfrak{S} = \langle M_U, M_N, M_E, M_D, \{E_e\}, \{A_{ae}\}, G, H, S \rangle. \quad (2.1)$$

Кортеж 2.1 має низку обмежень, розглянемо їх детально.

1) Транзакція може використовувати хоча б один застосунок (2.2),

$$\sum_{i=1}^A a_{ei} \geq 1 \forall e \in \overline{1, E}. \quad (2.2)$$

2) Транзакція може не звертатися до сховища даних (2.3),

$$\sum_{i=1}^D d_{ei} \geq 1 \forall e \in \overline{1, E}. \quad (2.3)$$

3) Транзакція потрібна бодай одному користувачеві (2.4),

$$\sum_{i=1}^U u_{ei} \geq 1 \forall e \in \overline{1, E}. \quad (2.4)$$

4) Користувач має запустити бодай одну транзакцію (2.5),

$$\sum_{e=1}^E u_{iu} \geq 1 \forall u \in \overline{1, U}. \quad (2.5)$$

5) Кожен застосунок має бути встановленим (2.6),

$$\sum_{i=1}^N g_{ai} \geq 1 \forall a \in \overline{1, A}. \quad (2.6)$$

6) Вузол має принаймні один активний застосунок (2.7),

$$\sum_{i=1}^N g_{ni} \geq 1 \forall n \in \overline{1, N}. \quad (2.7)$$

7) Всі застосунки мають бути встановлені (2.8),

$$\sum_{i=1}^A \sum_{j=1}^N g_{ij} = A. \quad (2.8)$$

8) Всі користувачі мають бути розподілені за визначеними інформаційними вузлами (2.9),

$$\sum_{i=1}^U \sum_{j=1}^N h_{ij} = N. \quad (2.9)$$

- 9) Всі користувачі мають бути закріплені до одного означеного інформаційного вузла (2.10),

$$\sum_{i=1}^N h_{ui} = 1 \quad \forall u \in \overline{1, U}. \quad (2.10)$$

- 10) Закріплення юзерів по інформаційних вузлах має бути довільним (2.11),

$$\sum_{i=1}^U h_{in} \in [0, U]. \quad (2.11)$$

- 11) По вузлах мережі розподілені всі фрагменти сховища даних (2.12)

$$\sum_{i=1}^U \sum_{j=1}^N s_{ij} = D. \quad (2.12)$$

- 12) Данні можуть розташовуватися на декількох вузлах:

$$\sum_{i=1}^N s_{di} \geq 1. \quad (2.13)$$

- 13) Фрагменти сховища даних можуть розміщуватися по вузлах  $\epsilon$  незалежно:

$$\sum_{i=1}^D s_{in} \in [0, D] \quad \forall i \in \overline{1, N}. \quad (2.14)$$

У підсумку можна сказати, що кортеж (2.1) у сукупності із обмеженнями визначають математичну модель інформаційної структури ГКС e-learning.

У даному розділі представлена розроблена модель. Проаналізовано необхідність моделювання інформаційної структури, а саме проблеми моделювання, пов'язані з неповнотою існуючих методів, або їх надлишковості (на прикладі мереж масового обслуговування). Виділено

основні принципи аналізу гіперконвергентної базової мережі. Дано розроблений підхід до моделювання інформаційної структури, заснований на дослідженні взаємодії застосунків як незалежних джерел і приймачів даних.

Було представлено розроблену модель основних інформаційних взаємозв'язків у середовищі e-learning. Розглянуто схему основних інформаційних взаємозв'язків у середовищі e-learning. Зазначено основні множини, задіяні в процесі моделювання, розглянуто параметри кожної транзакції та параметри обміну даними кожного додатку й блоку даних. На основі цієї інформації сформовано кортеж множин і матриці, що описують модель системи, включаючи одразу вхідні й вихідні дані синтезу. Надано обмеження на елементи кортежу.

### 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОЦЕСУ МОДЕЛЮВАННЯ СИНТЕЗУ ІНФОРМАЦІЙНОЇ СИСТЕМИ

#### 3.1 Особливості оптимізації з використанням генетичного алгоритму

На сьогоднішній день, для вирішення проблем моделювання та оптимізаційних задач, все більш широкого попиту обирають еволюційні методи дослідження, а саме генетичні алгоритми. Генетичний алгоритм має низку особливостей, наприклад схрещення, що виконує перехресний вибір рішень-кандидатів, аналогічна процедура схрещення проходить у навколишньому середовищі. Розглянемо процедуру виконання генетичного алгоритму. Поперше необхідно закодувати задачу. Це треба зробити так, щоб шукане рішення було відображено у вигляді масиву аналогічному, як у хромосоми. Такий масив даних називається хромосома. Стартовий набір даних будується випадковим чином (рисунок 3.1) .

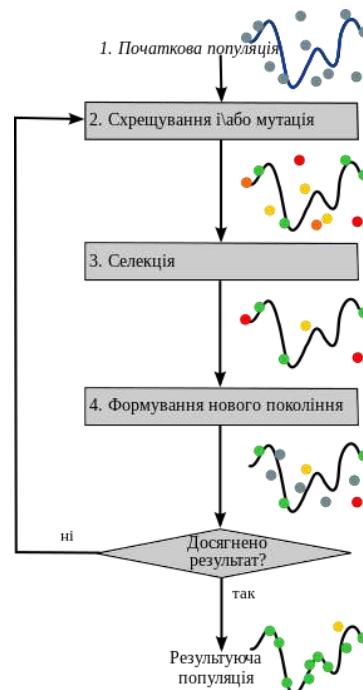


Рисунок 3.1 – Схематичне створення ГА

Цей процес представлено осіб'ю або початковою популяцією. Далі оцінюються особи. Це проходить за допомогою функції пристосування (яка має велике значення на результат кожної особи), вона присвоює важливе значення пристосованості, саме воно визначає ступінь виживання особи [13]. Далі використовуємо отримані значення пристосованості. Мається на увазі, що обираються особи, які мають можливість до селекції, бо не всі особи допущені до схрещення. Для цього використовуються генетичні оператори.

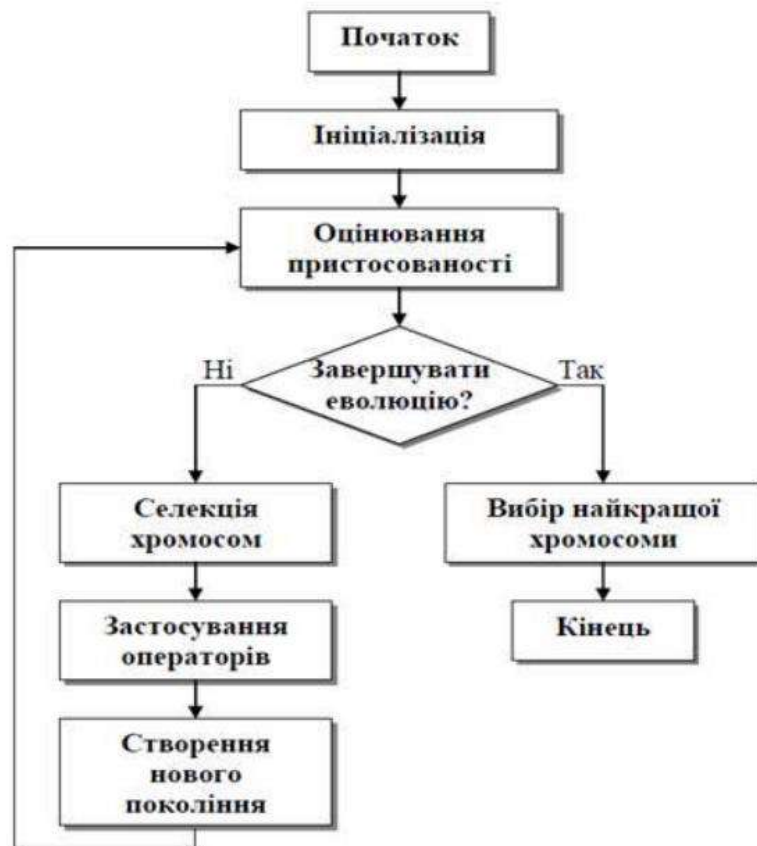


Рисунок 3.2 – Еволюційний цикл

Таким чином створюється наступне покоління осіб. Особи, що були отримані, тобто наступне покоління, проходить аналогічні процедури

(селекція і мутація). Ми розглянули повний цикл моделювання еволюційного процесу. Він може продовжуватися не один життєвий цикл. Кожний цикл називається поколінням. Для зупинки процесу еволюцій, необхідно задати критерій зупинки.

Необхідно розглянути такі критерії. Поперше це може бути оптимальне рішення. А по друге алгоритм можна запрограмувати на певний час.

Оптимальні та надоптимальні рішення, генетичні алгоритми спроможні шукати для великих даних, або складних завдань [13].

Розглянемо етапи ГА, схема даного процесу представлено на рисунку 3.2. Спочатку створюємо популяцію (початкова популяція), обчислюємо фітнес функції, далі необхідно задати критерій зупинки.

### 3.2 Поєднання моделі та генетичного алгоритму

Відповідно до розробленої моделі інформаційної структури системи e-learning, розгорнутої на гіперконвергентному сервері критерієм для оцінки даної структури являється збалансованість навантаження на вузли, а показником – сума відхилень завантаження вузлів від середнього навантаження. Не складно дійти до висновку, що чим менше показник відхилень завантаження тим вище показники якості функціонування системи e-learning. Маючи на меті отримати найефективнішу структуру базової мережі підтримки e-learning та додаючи до цього знання про генетичний алгоритм стає очевидним, показник ефективності структури базової мережі є не що інше, як значення фітнес функції в генетичному алгоритму.

Звідси маємо вхідні данні – це інформація про кількість додатків (A), користувачів (U), транзакцій (E), вузлів (N) та блоків даних (D) функціонуючих в системі e-learning. Також нам потрібна інформація по транзакціям, а саме які додатки ( $\overline{a_e} = (a_{e1}, \dots, a_{eA})$ ) і блоки даних ( $\overline{d_e} = (d_{e1}, \dots, d_{eD})$ ) використовуються, а також порядок запуску додатків

$(\overline{W}_e = (w_{ij}))$ , але цього недостатньо, бо необхідно мати уяву про функціонування кожного додатка в умовах певної транзакції для цього необхідні знання про об'єми обміну даними ( $A_{ae} = \langle \overline{\lambda}_{ae}, \overline{\beta}_{ae} \rangle$ , де  $\overline{\lambda}_{ae} = (\lambda_{ae1}, \dots, \lambda_{aeD})$  і  $\overline{\beta}_{ae} = (\beta_{ae1}, \dots, \beta_{aeA})$ ). Для того щоб імітувати реальне функціонування системи і взаємодію її з користувачами нам потрібна інтенсивність запуску користувачами транзакцій.

Після роботи генетичного алгоритму вихідними даними, що будуть характеризувати структуру базової мережі підтримки e-learning буде розміщення користувачів (H), додатків (G), блоків даних (S) по вузлам мережі.

Все це разом сформує кортеж моделі інформаційної системи e-learning:

$$(\mathfrak{S} = \langle M_U, M_N, M_E, M_D, \{E_e\}, \{A_{ae}\}, G, H, S \rangle).$$

### 3.3 Розроблений алгоритм синтезу моделі інформаційної структури системи e-learning

В основі алгоритму синтезу моделі інформаційної структури системи e-learning лежить генетичний алгоритм з тим доповненням, що зупинкою для алгоритму буде слугувати умова кількості поколінь, або наближення до цільового значення фітнес функції. Отже, залишається лиш роздивитися алгоритм розрахунку значення фітнес функції і параметри розробленої мережі, маючи усі необхідні данні:

Розрахуємо за матрицею інтенсивності запуску транзакцій користувачами ( $\Omega = (\omega_{i,j}); \omega_{i,j} \geq 0; i = 1, \dots, U; j = 1, \dots, E$ ) сумарну інтенсивність запитів на запуск j-й транзакції:

$$\bar{\omega} = (\omega_j), \omega_j = \sum_{i=1}^U \omega_{i,j}. \quad (3.1)$$

Знайдемо сумарну інтенсивність запуску  $k$ -ого ( $k = 1, \dots, A$ ) додатку всіма транзакціями системи, тут  $a_{j,k}$  – булев вектор необхідних додатків для  $j$ -ї транзакції:

$$\Theta = (\theta_k), \theta_k = \sum_{j=1}^E \omega_j a_{j,k}. \quad (3.2)$$

Роздивимося функціонування  $j$ -ї транзакції на інформаційних вузлах  $n$  та  $m$  ( $n, m = 1, \dots, N$ ):

$$Z_{jnm1} = \sum_{\xi=1}^A a_{j,\xi} g_{\xi,n} (\sum_{\eta=1}^A a_{j,\eta} g_{\eta,m} \beta_{\xi,\eta}). \quad (3.3)$$

Визначимо для  $(j, n, m)$  об'єми даних, що передаються із сховища даних:

$$Z_{jnm2} = \sum_{\xi=1}^A a_{j,\xi} g_{\xi,n} (\sum_{\eta=1}^D d_{j,\eta} s_{\eta,m} \lambda_{\xi,\eta}). \quad (3.4)$$

Об'єми даних в мережі можливо визначити за формулою:

$$Z = (Z_{n,m}), Z_{n,m} = \sum_{j=1}^E (Z_{jnm1} + Z_{jnm2}). \quad (3.5)$$

Зараз можливо побудувати матрицю інтенсивності обміну між інформаційними вузлами мережі при виконанні  $j$ -ї транзакції:

$$C_j = (C_{j,n,m}), C_{j,n,m} = \omega_j Z_{j,n,m}. \quad (3.6)$$

Відповідно, матриця навантаження на інформаційні вузли (також це навантаження каналів зв'язку й мережевого обладнання) визначається за формулою:

$$C = (C_{n,m}), C_{n,m} = \sum_{j=1}^A C_{j,n,m}. \quad (3.7)$$

Визначимо середнє завантаження одного вузла мережі:

$$C_{\text{ср.}} = (\sum_{n=1}^N \sum_{m=1}^N C_{n,m}) / N. \quad (3.8)$$

Для характеристики системи критерієм є збалансованість навантаження на вузли, а показником – сума відхилень навантаження від середньої:

$$\Psi = \sum_{n=1}^N \sum_{m=1}^N |C_{n,m} - C_{\text{ср.}}|. \quad (3.9)$$

Отже, маємо фітнес функцію для генетичного алгоритму:  $\Psi \rightarrow \min$ .

Також розрахуємо навантаження вузлів мережі. Інтенсивність потоку запитів на запуск додатка  $i$ , встановленого на вузлу  $n$ :

$$B = (b_{i,n}) = \Theta_{\text{діаг.}} G, b_{i,n} = \theta_i g_{i,n}. \quad (3.10)$$

Завантаження вузлів з елементами сховища даних:  $j$ -я транзакція,  $n$ -й вузол,  $d$ -й блок даних:

$$\varphi_{j,n,d} = \theta_j S_{n,d} d_{j,n}. \quad (3.11)$$

Відповідно, до блоку даних  $d$  на вузлу  $n$  інтенсивність запитів до сховища даних дорівнює:

$$\Phi = (\varphi_{n,d}), \varphi_{n,d} = \sum_{j=1}^E \varphi_{j,n,d}. \quad (3.12)$$

### 3.4 Діаграма класів, які було використано при розробці

Діаграма варіантів використання відображена на рисунку 3.2, вона допомагає при розробці синтезу ПЗ.

Діаграма класів відображає класи системи, а саме взаємозв'язок між ними, атрибути та методи. Такі діаграми можуть бути побудовані різними шляхами і методами, це залежить від цільового використання [15]:

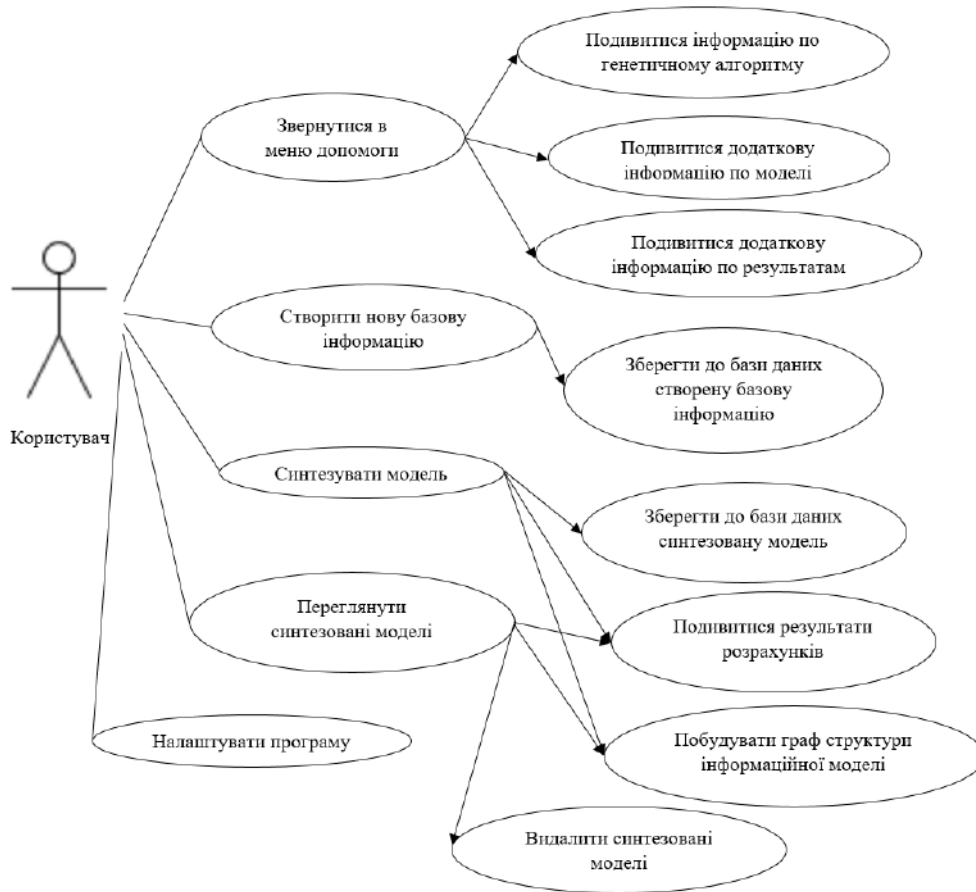


Рисунок 3.3 – Діаграма варіантів використання програми синтезу інформаційної системи e-learning

У даному випадку побудови діаграми класів використовуються з точки зору реалізації. У ній описано класи і взаємозв'язки між ними.

Для завдання видимості членів класу (тобто будь-який атрибут або метод), ці позначення повинні бути розміщені перед ім'ям учасника [45]:

- «+» – публічний (public);
- «-» – приватний (private);
- «#» – захищений (protected);
- «/» – похідний (derived);

- «~» – пакет (package).

Взаємозв'язок – це особливий тип логічних відносин між сутностями, показаних на діаграмах класів і об'єктів. Відносини в UML зображені на рисунку 3.4.

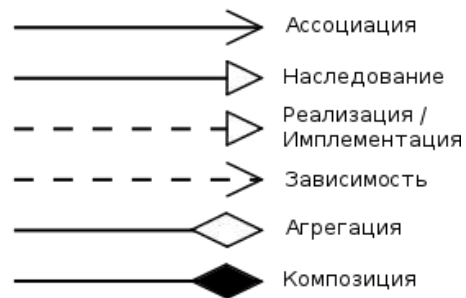


Рисунок 3.4 – Види відносин в UML

Маючи уявлення про діаграму класів та структуру моделі, а також маючи генетичний алгоритм маємо діаграму класів (рисунок 3.4), що зображує класи необхідні для функціонування генетичного алгоритму.

Також нам необхідно зберігати до бази даних базову інформацію та результати синтезу інформаційної структури системи e-learning, для цього маємо діаграму класів (рисунок 3.5).

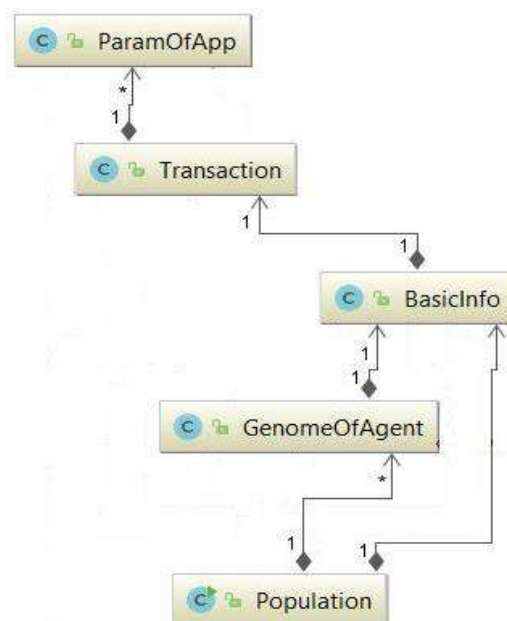


Рисунок 3.5 – Діаграма класів, для функціонування ГА

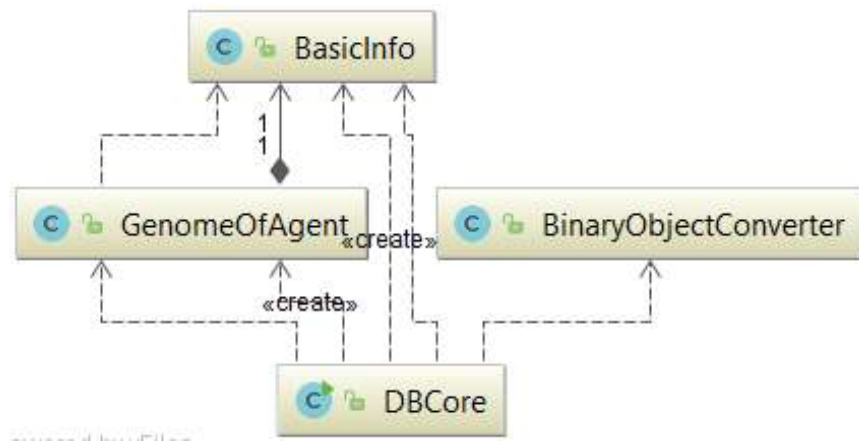


Рис. 3.6. – Діаграма класів, необхідних для роботи з базою даних

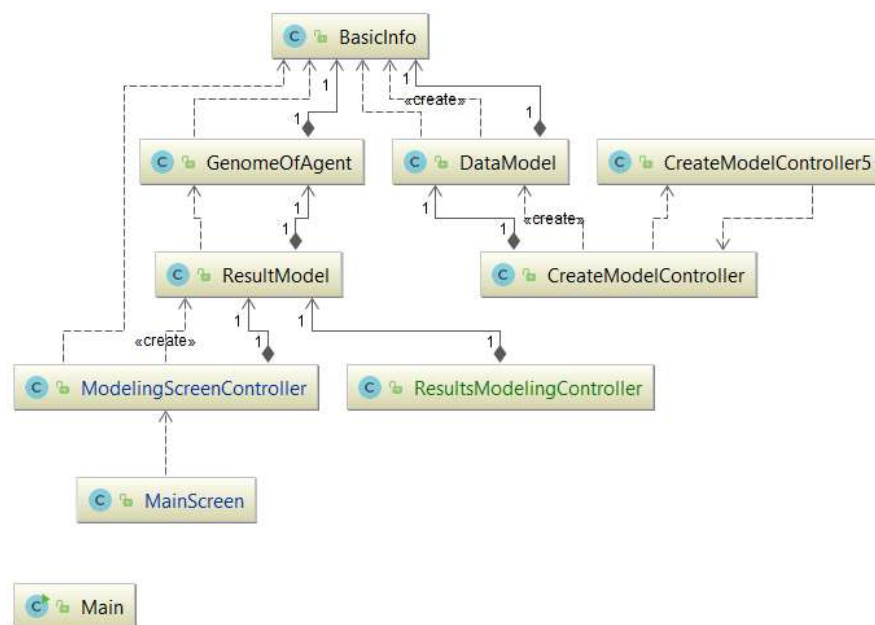


Рисунок 3.7 – Діаграма класів, необхідних для роботи графічного інтерфейсу користувача

Також необхідно організувати графічний інтерфейс програми для зручної роботи користувача, при цьому використовувалася патерн програмування MVC, що дозволить зробити зручне перенесення та розширення можливостей програми. Відповідна діаграма класів представлена на рисунку 3.7.

### 3.5 Схема бази даних

Схема бази даних – її структура, описана на формальній мові, що підтримується СУБД. У реляційних базах даних схема визначає таблиці, поля в кожній таблиці (зазвичай із зазначенням їх назви, типу, обов'язковості), і обмеження цілісності (первинний, потенційні і зовнішні ключі та інші обмеження) [16].

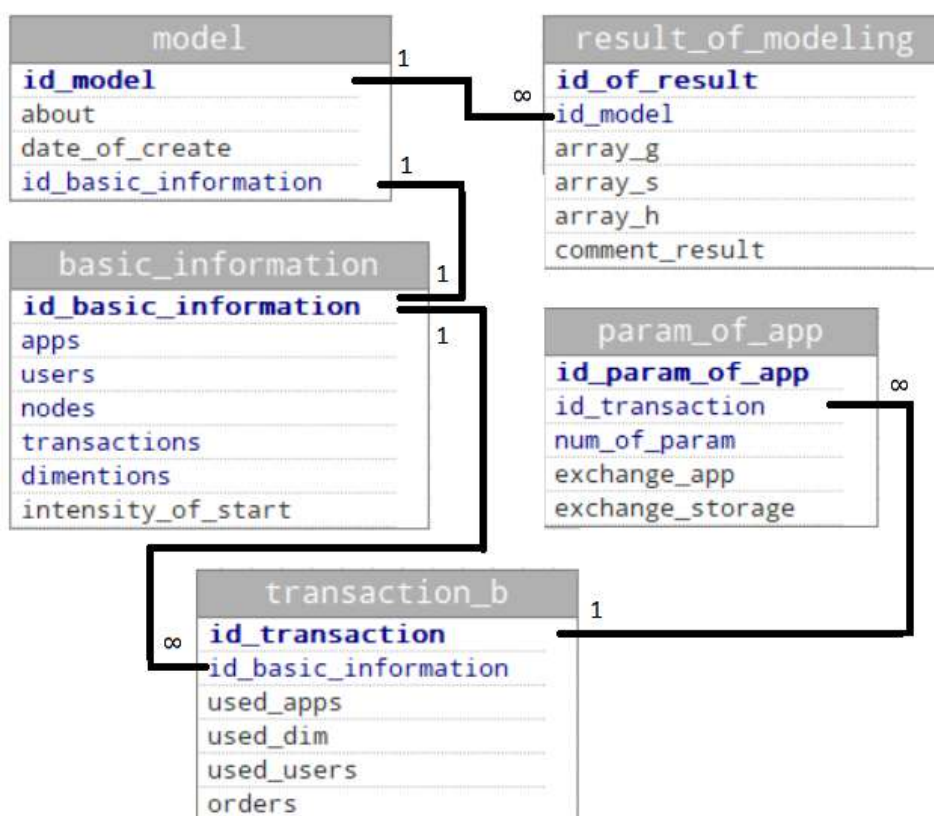


Рисунок 3.8 – Схема бази даних, що зберігає вхідні та вихідні дані

Схеми в загальному випадку зберігаються в словнику даних. Хоча схема визначена на мові бази даних у вигляді тексту, термін часто використовується для позначення графічного представлення структури бази даних. Основними об'єктами графічного представлення схеми є таблиці і зв'язку, які визначаються зовнішніми ключами [10].

Маючи інформацію про вхідну на вихідну інформацію моделі не

складно дійти висновку, що для збереження її до бази даних необхідна реляційна база даних.

### 3.7 Пояснення вибору мови програмування

З діаграми варіантів використання програмою, зрозуміло, що програма матиме великий інтерфейс роботи з користувачем. В той же час схема генетичного алгоритму матиме великі витрати ресурсів тільки в випадку великої розмірності задачі, тому доцільно використовувати мову програмування Java, бо вона також зможе забезпечити переносимість майбутньої програми із незначними програмами в ефективності роботи в порівнянні із C/C++. Для створення графічного інтерфейсу будемо використовувати платформу на основі Java – JavaFX. Схема бази даних говорить, що таблиці не мають між собою надто складних взаємозв'язків і самих таблиць невелика кількість, отже було би використовувати вбудовану базу даних, що не перевантажена надмірним функціоналом із збереження переваг реляційних баз даних, а саме SQLite базу даних, яка також дозволить використовувати SQL мову запитів.

Роздивимось детальніше мову програмування Java. Java – сильно типізована об'єктно-орієнтована мова програмування, розроблена компанією Sun Microsystems (в подальшому придбаної компанією Oracle) [17]. Програми Java зазвичай транслюються в спеціальний байт-код, тому вони можуть працювати на будь-якій комп'ютерній архітектурі, за допомогою віртуальної Java-машини [17]. Часто до недоліків концепції віртуальної машини відносять зниження продуктивності. Ряд удосконалень декілька збільшив швидкість виконання програм на Java.

JavaFX – платформа на основі Java для створення додатків з насиченим графічним інтерфейсом [48]. Може використовуватися як для створення настільних додатків, що запускаються безпосередньо з-під операційних систем, так і для інтернет-додатків (RIA), що працюють в браузерах, і для

додатків на мобільних пристроях [18].

Після аналізу для вибора бази даних, значні переваги були у SQLite.

Кілька процесів або потоків можуть одночасно без будь-яких проблем читати дані з однієї бази. Запис в базу можна здійснити тільки в тому випадку, якщо ніяких інших запитів в даний момент не обслуговується; в іншому випадку спроба запису закінчується невдачею, і в програму повертається код помилки. Іншим варіантом розвитку подій є автоматичне повторення спроб запису протягом заданого інтервалу часу.

Проведено аналіз генетичного алгоритму. Дано визначення генетичного алгоритму, представлено етапи алгоритму із докладним описом, описано умови зупинки алгоритму, представлено область використання алгоритму та нюанси використання.

Представлено суміщення генетичного алгоритму із розробленим кортежем моделі із подальшим виявленням вхідних і вихідних даних.

Надано розроблений алгоритм синтезу моделі інформаційної структури, в основі якого лежить генетичний алгоритм. Далі представлено алгоритм із формулами обчислення фітнес функції, що буде характеризувати пристосованість агенту. Також надано формули для обчислення навантаження на вузли та мережеве обладнання системи.

Наведено інформацію про діаграми варіантів використання комп'ютерних програми, далі, відповідно до цієї інформації та знань про кортеж моделі, наведено діаграму варіантів використання програми, що надалі допоможе в розробці програми.

Представлено інформацію про діаграми класів, їх види, та деталі реалізації. Відповідно до мети створення діаграми класів вибрано точку зору реалізації з урахуванням генетичного алгоритму, необхідністю в збереженні результатів та створенні візуального інтерфейсу - спроектовано діаграму класів, необхідних для створення програми.

Представлено схему розробленої бази даних із декількома таблицями, що здатна зберігати вхідні та вихідні дані та не перевантажена зв'язками.

## 4 АНАЛІЗ РЕЗУЛЬТАТІВ СИНТЕЗУ МОДЕЛІ

### 4.1 Інтерфейс програмного комплексу

Початок роботи ПЗ на віртуальній машині JDK. При наявності файлів форматів «.jar» та «.exe» необхідно перевірити версію ОС. Вона має бути Windows починаючи з 7-ої. Після запуску робочого файлу повинен відобразитися інтерфейс головного екрану (рисунок 4.1), на ньому представлено інформація про розробника, також розтошовано кнопку для вхідних даних, можна синтезувати модель та ознайомлюватися з результатами моделювання.

Обравши «створити модель», користувач в декілька етапів вводить дані в програму. На рисунку 4.2 першим кроком описується нову модель. У верхньому полі встановлюється дата створення. Другим кроком, який представлено на рисунку 4.3, задаються множини A, N, U, E, D, що приймають участь у моделюванні. Третім кроком роботи ПЗ, що представлено на рисунку 4.4, задається інтенсивність запуску транзакцій між користувачами. Четвертим кроком заповнюються параметри користувачем даними по кожній із транзакцій, що детально представлено на рисунку 4.5, далі заповнюються обсяги обміну даних (рисунок 4.6) між блоками даних та додатками. В ході введення вхідних даних користувач взаємодіє з ПЗ. На рисунку 4.7 відображене вікно закриття, також інформація про не заповнені поля представлена на рисунку 4.8, також є інформація про некоректно введені данні (рисунок 4.9).

Користувач обравши меню про початок моделювання баче вікно із налаштуваннями ГА (рисунок 4.10). Далі користувач натиснувши кнопку «Почати моделювання» бачить відкрите вікно, що містить результати інформаційного синтезу (рисунок 4.11). Результати моделювання є можливість зберегти, або просто переглянути (рисунок 4.12).

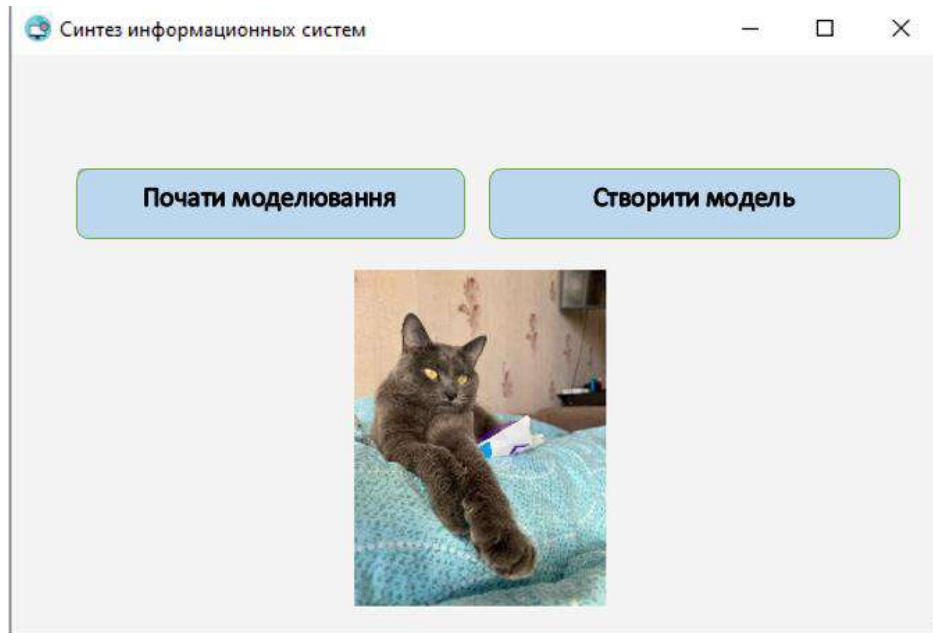


Рисунок 4.1 – Головний інтерфейс ПЗ

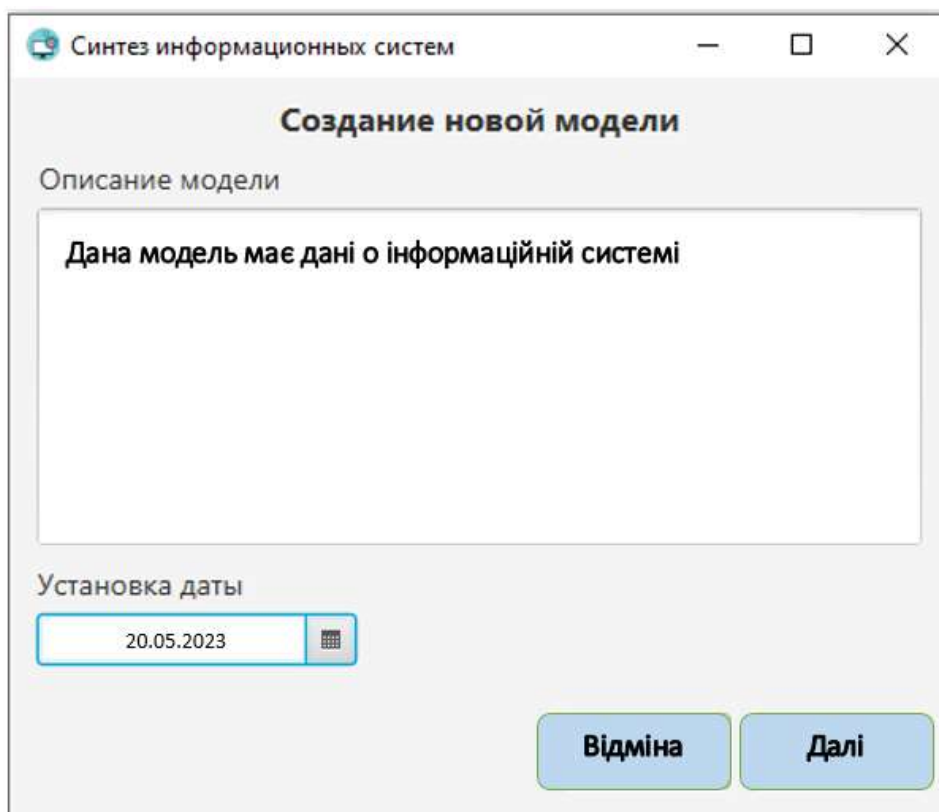


Рисунок 4.2 – Перший крок

**Установка параметров**

Количество приложений: 8

Количество узлов: 10

Количество пользователей: 6

Количество транзакций: 5

Количество блоков данных: 7

Отмена    Далее

Рисунок 4.3 – Другой шаг

**Интенсивности запуска транзакций**

		Транзакции				
		1	2	3	4	5
Пользователи	1	25	0	0	0	0
	2	0	10	10	0	0
	3	10	0	0	15,4	0
	4	7	0	0	0	25
	5	0	13	0	0	9
	6	0	0	2	0	0

Отмена    Далее

Рисунок 4.4 – Третий шаг

Синтез информационных систем

### Данные транзакции

Используемые приложения

1	2	3	4	5	6	7	8
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Используемые блоки данных

1	2	3	4	5	6	7
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Пользователи, запускающие транзакцию

1	2	3	4	5	6
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Порядок запуска приложений

	1	2	3	4	5	6	7	8
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Параметры приложений

< > 1 / 5

Отмена Сохранить

Рисунок 4.5 – Четвертый шаг

Параметры приложений

Обмен между приложениями

1	2	3	4	5	6	7	8
1	0	0	1	0	0	1	0

Обмен между хранилищами данных

1	2	3	4	5	6	7
0	0	1	0	1	0	0

< > 1 / 8

Отмена Далее

Рисунок 4.6 – Обмен данными

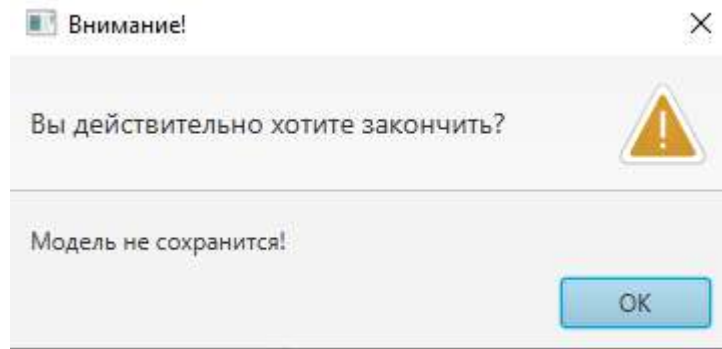


Рисунок 4.7 – Завершення моделювання

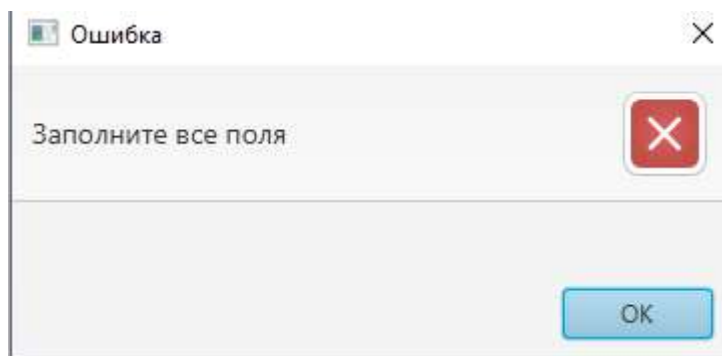


Рисунок 4.8 – Незаповнені поля

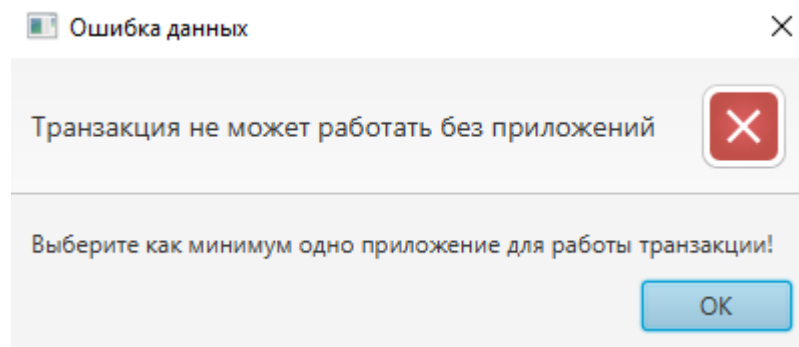


Рисунок 4.9 – Вікно помилки

**Исходные данные для моделирования**

Тип генетического алгоритма:  По количеству итераций  К граничному значению

Базовая модель: 20.05.2023

Общее количество агентов: 150

Количество мутаций агентов: 120

Количество скрещиваний агентов: 50

Количество итераций/Целевое значения: 1500

Відміна Почати моделювання

Рисунок 4.10 – Вікно налаштування ГА

**Результаты моделирования**

G:  
 0010000000  
 0000010000  
 0010000000  
 0000000001  
 1000000000  
 0000001000  
 0000010000  
 0000000010

S:  
 0000100000  
 0000000100  
 0000100000  
 0000000010  
 0000010000  
 0010000000  
 0000000100

суммарная интенсивность запуска транзакции j:  
 42.0 23.0 12.0 15.4 34.0

суммарная интенсивность запуска приложения k:  
 126.4 0.0 0.0 126.4 0.0 0.0 126.4 0.0

Отмена Визуализировать Сохранить

Рисунок 4.11 – Вікно результатів

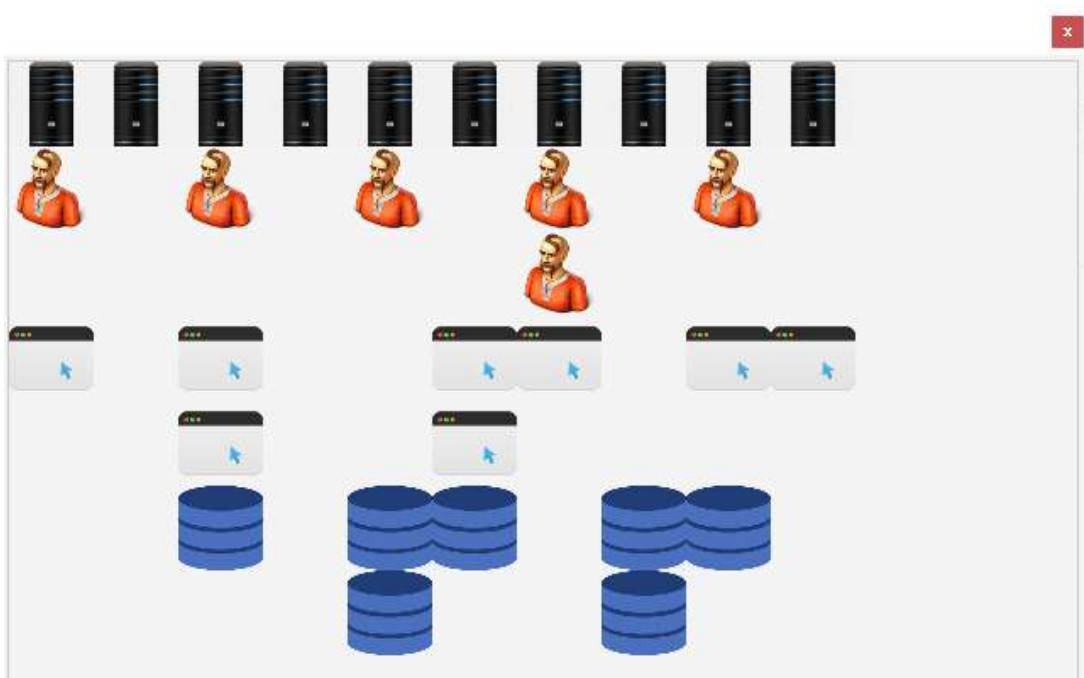


Рисунок 4.12 – Прототип графу станів

#### 4.2 Особливості використання програмного комплексу

Синтезована модель інформаційної структури системи e-learning дозволяє розподілити навантаження по гіперконвергентній мережі, цим самим збільшивши ефективність функціонування системи в цілому. Недоліки, що притаманні рішенню – є недоліками генетичного алгоритму: рішення не завжди оптимальне та необхідне багаторазове повторення синтезу для одної і тієї ж моделі, бо результати зазвичай випадкові й неоптимальні, також є проблеми з евристичним налагодженням алгоритму.

Наприклад, для даних  $U=6$ ,  $N=10$ ,  $A=7$ ,  $E=3$ ,  $D=3$ , такі результати доволі оптимальні (рисунок 4.13), хоча є деякі неоптимальні розстановки блоків даних.

$\mathcal{G}$   
 {0, 0, 0, 0, 1, 0, 0, 0, 0, 0}  
 {1, 0, 0, 0, 0, 0, 0, 0, 0, 0},  
 {0, 1, 0, 0, 0, 0, 0, 0, 0, 0},  
 {0, 0, 0, 0, 0, 0, 0, 0, 1, 0},  
 {0, 0, 1, 0, 0, 0, 0, 0, 0, 0},  
 {0, 0, 0, 0, 0, 1, 0, 0, 0, 0},  
 {0, 0, 0, 1, 0, 0, 0, 0, 0, 0};

$\mathcal{H}$   
 {1, 0, 0, 0, 0, 0, 0, 0, 0, 0},  
 {0, 0, 0, 0, 0, 1, 0, 0, 0, 0},  
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},  
 {0, 0, 1, 0, 0, 0, 0, 0, 0, 0},  
 {0, 0, 0, 1, 0, 0, 0, 0, 0, 0},  
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0}};

$\mathcal{S}$   
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 1},  
 {0, 0, 0, 0, 0, 1, 0, 0, 0, 0},  
 {0, 0, 0, 0, 0, 0, 1, 0, 0, 0}};

Рисунок 4.13 – Результати синтезу моделі інформаційної структури

```

рассчитаем работу функционирования
транзакции j на информационных узлах n и m
0000000000
0000000000
0010000200
0000000000
0000000000
0000000000
0000000000
0000000000
0020000400
0000000000
0000000000

определим объемы данных передаваемых из хранилища данных:
0000000000
0000000000
0001000100
0000000000
0000000000
0000000000
0000000000
0000000000
0002000200
0000000000
0000000000
  
```

Рисунок 4.14 – Деякі параметри, що розраховує програма

Також програма дозволяє розрахувати і вивести для користувача об'єми даних, що передаються із сховища, та між додатками, загальні об'єми даних в мережі, отримати навантаження на інформаційні вузли, середнє навантаження (рисунок 4.14).

Представлено розроблену програму з поясненням запуску програми при наявності виконавчого файлу, розглянуто інтерфейс використання програми та надано відповідні зображення. Показано типовий алгоритм використання програми з вводом початкових даних, збереженням їх до бази даних та синтезом моделі.

Надано інформацію про результати роботи програми, їх аналіз. В результаті виведено переваги та недоліки притаманні створеному алгоритму, що також притаманно генетичним алгоритмам взагалі.

## ВИСНОВКИ

В кваліфікаційній роботі було розглянуто модель синтезу інформаційної системи e-learning, Базою для створення якої була створена модель інформаційних взаємозв'язків системи електронного навчання розгорнутої на гіперконвергентному сервері.

Проведено аналіз існуючих технологій і розробок в електронному навчанні, проаналізовано перспективну технологію гіперконвергентних мереж, що давало би більш економічно ефективне рішення для університетської системи підтримки e-learning. Проаналізовано методи управління мережевими ресурсами. Далі було створено моделі інформаційної структури системи e-learning. Після розглядання генетичного алгоритму стало можливим зі створеної моделі отримати модель синтезу інформаційної системи e-learning.

Результатом виконання кваліфікаційної роботи є програма, що здатна через графічний інтерфейс приймати від користувача вхідні дані, що характеризують склад системи e-learning із власними додатками, блоками даних і транзакціями що описують їх взаємодію та інформацію про склад гіперконвергентної мережі і на виході отримувати розташування користувачів, блоків даних і додатків по вузлам мережі, що є оптимальним в умовах даної інтенсивності запуску транзакцій користувачами. Дана програма є досягненням поставленої мети, та реалізує усі поставленні задачі.

Розроблене програмне забезпечення дозволяє підвищити ефективність використання базової гіперконвергентної мережі, а, отже, і збільшити якість функціонування системи e-learning в цілому, що являється необхідною складовою створення такої системи в умовах обмеженого бюджету університету.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Kuchuk N., Kotkova O., Rybalchenko A. Analysis of information structure modeling methods. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління : Матеріали дванадцятої міжнародної науково-технічної конференції. – Баку: ВА ЗС АР; Харків : НТУ «ХПІ»; Київ : НАУ; Харків : ДП «ПДПРОНДІАВІАПРОМ»; Жиліна : університет, 2022. – С. 22.
2. Kuchuk N. H. Comprehensive performance criterion for hyper-converged infrastructure / N. Kuchuk // Телекомунікаційні та інформаційні технології. – Київ : Державний університет телекомунікацій, 2019. – Вип. № 3 (64). – С. 55 – 63.
3. Web Application Performance: 7 Common Problems and How to Solve Them – Режим доступу : <https://stackify.com/web-application-problems/> Ian Molyneaux The Art of Application Performance Testing, 2nd Edition / Ian Molyneaux – O'Reilly Media, Inc.
4. Performance Testing : What is, Types, Metrics & Example – Режим Тестування. фундаментальна теорія: <https://dou.ua/forums/topic/13389/Dr. Dobbs Journal, chneier B. Attack Trees. December 1999 – С.3>
5. Camtepe S., Yener B. A Формальний спосіб для Attack Modeling and Detection. <http://cs.rpi.edu/research/pdf/06-01.pdf>
6. Phillips C., Swiler L.. A Graph-Based System for Network-Vulnerability Analysis // In Proceedings of New Security Paradigms Workshop, Charlottesville, VA, 1998.
7. Tvoroshenko I., and Maksimenko H. To the question of analysis of existing mechanisms of web application testing, Abstracts of I International scientific- practical conference «Problems of modern science and practice» (September 21-24, 2021). Boston, USA, pp. 403-409. 59.
8. Tvoroshenko I., and Maksimenko H. Research of regression and

modular testing of web applications, Abstracts of IV International scientificpractical conference «Science, theory and practice» (October 12-15, 2021). Tokyo, Japan, pp. 406-411.

9. Brown A.W. Large-scale Component-Based Development [Text] / A.W. Brown. - Prentice-Hall, 2010. - 300 p.

10. Cohn M. Agile Estimating and Planning [Text] / Mike Cohn. - Prentice Hall, 2015. - 368 p.

11. Didkovska M. Criteria for integration testing of component-based software [Text] / M.Didkovska // *Електроника и связь*. - К.: 2004. - №23. - С. 90-94.

12. Myers G.J. The Art Of Software Testing [Text] / G.J. Myers - New York: John Wiley & Sons, Inc., 2004. - 254 p. - ISBN 0-471-46912-2.

13. Offutt J. Generating tests from UML specifications [Electronic resource] / J. Offutt, A. Abdurazik // *Second International Conference on the Unified Modeling Language*. - Fort Collins, CO, IEEE Computer Society Press, 1999. - С. 416-429

14. Page-Jones M. Fundamentals of Object-Oriented Design in UML / Page-Jones M. - New York: Addison-Wesley, 2000. - 435 p.

15. Patton R. Software Testing [Text] / R. Patton. - 2nd Edn. - Indianapolis: Sams, 2005. - 408p.

16. Кріспін Л. Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд : пер. з англ. / Л. Кріспін., Д. Грегори – М.: «Вільямс», 2011. – 463 с. – ISBN 0-471-46912-2.

17. Кучук Н. Синхронізація інформаційної структури мережі на гіперконвергентній платформі з обчислювальними ресурсами її вузлів / Н. Г. Кучук // *Системи управління, навігації та зв'язку*. Полтава : ПНТУ, 2019, випуск 6(58). С. 48-54.

18. Кучук Н. Синхронізація інформаційної структури мережі на гіперконвергентній платформі з обчислювальними ресурсами її вузлів / Н. Г. Кучук // *Системи управління, навігації та зв'язку*. Полтава : ПНТУ, 2019,

випуск 6(58). С. 48-54.

19. Липаев В. В. Тестирование компонентов и комплексов программ: підруч. / Липаев В. В. – М.-Берлін: Дірект-Медіа, 2015. – 528с. – ISBN 978-5-89638-115- 0.