

Хмельницький національний університет
Факультет програмування
та комп'ютерних і телекомунікаційних систем
Кафедра комп'ютерної інженерії та системного програмування

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Засоби динамічного розподілу ресурсів для розподіленої обробки потоків

Назва теми

КвРКІ.170285.17.02.17 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»
Шифр, назва

Освітня програма «Комп'ютерна інженерія»
Назва

Виконав: студент IV курсу, група КІ-17-2


Підпис

Б.Л. Охота
Ініціали, прізвище

Керівник


Підпис, дата

В.М. Чешун
Ініціали, прізвище

Нормоконтролер


Підпис, дата

І.В. Муляр
Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
Інженерії та системного
Програмування


Підпис

Ю.П. Кльоц
Ініціали, прізвище

«23» червня 2021 р.

Хмельницький 2021

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНИХ І ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Кафедра КІБЕРБЕЗПЕКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЯ ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ю.П. Кльоц

“ 11 ” 01 2021 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Охоті Богдану Леонідовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Засоби динамічного розподілу ресурсів для розподіленої обробки потоків

Керівник проекту (роботи) Чешун Віктор Миколайович, д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 05.02.2021 р. № 11

2. Строк подання студентом проекту (роботи) на кафедру 18.06.2021 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Дослідження предметної області та постановка задачі

Проектування Засобів динамічного розподілу ресурсів для розподіленої обробки потоків

Програмно-апаратна реалізація та тестування) Засоби динамічного розподілу ресурсів для розподіленої обробки потоків



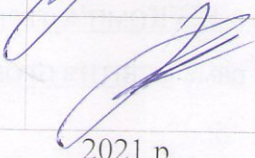

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Структура обробки потоку даних (E1)

Коригування ресурсів кожної черги пріоритетів з урахуванням коефіцієнтів установки як конфігурація a, b та c (E3)

Концептуальне проектування схеми, що враховує дорожній рух (E2)

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Муляр І.В., доцент кафедри КБКСМ		
Антиплагіат	Муляр І.В., доцент кафедри КБКСМ		

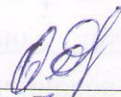
7. Дата видачі завдання « 08 » 02 2021 р.

КАЛЕНДАРНИЙ ПЛАН

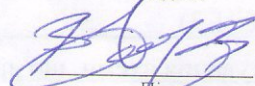
№ з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітки
1.	Підготовка вступного розділу	Березень - 1 декада	
2.	Огляд існуючих методів, засобів	Березень - 2 декада	
3.	Обґрунтування обраних рішень	Березень - 3 декада	
4.	Підготовка опису електричних схем	Квітень - 1 декада	
5.	Виконання розрахункової частини	Квітень - 1 декада	
6.	Підготовка ескізів креслень	Квітень - 2 декада	
7.	Формулювання висновків	Квітень - 3 декада	
8.	Розробка додатків	Травень - 1 декада	
9.	Погодження розділів з консультантом з нормоконтролю	Травень - 1 декада	
10.	Оформлення графічного матеріалу	Травень - 2 декада	
11.	Оформлення пояснювальної записки	Травень - 2 декада	
12.	Попередній захист кваліфікаційної роботи	Травень - 3 декада	
13.	Доопрацювання кваліфікаційної роботи	Травень - 3 декада	
14.	Подання роботи для перевірки на плагіат	Травень - 3 декада	
15.	Захист кваліфікаційної роботи	Червень - 1 декада	

Студент

Керівник проекту (роботи)


Підпис

Б.Л. Охота
Ініціали, прізвище


Підпис

В.М. Чешун
Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Засоби динамічного розподілу ресурсів для розподіленої обробки потоків».

Автор роботи: Охота Богдан Леонідович.

Керівник роботи: Чещун Віктор Миколайович.

Пояснювальна записка: 57 с., 18 рис., 18 джерел.

Графічна частина: 3 плакати.

Практична значимість дипломного проекту полягає в тому, що розробляються ефективні стратегії управління ресурсами для розподіленої обробки потоків щодо різноманітних цілей. Зокрема, було визначено чотири проблеми при розробці стратегій розміщення операторів та управління ресурсами для ЦСП. Для вирішення проблем пропонується п'ять дослідницьких питань, тоді як кожне питання фокусується на одній із проблем із запропонованими рішеннями. Питання дослідження були визначені для вирішення чотирьох проблем у потоковій обробці: 1) різноманітний попит на ресурси та швидкість передачі даних окремими операторами ЦСП, 2) різні чутливості до затримок у програмах ЦСП, 3) динамічний стан мережі та схема передачі даних, і 4) значна вартість мігруючих операторів з державним статусом.



Підпис студента

10.06.2021

Дата

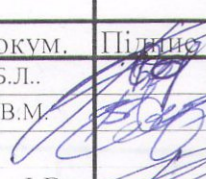
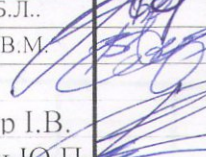
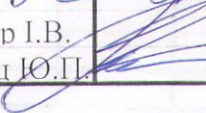
№	Форма	Позначення	Найменування	Кількість	№ екз	Примітка
			<u>Текстові документи</u>			
			<u>Текстові документи</u>			
1	A4	КВРКІ.170285.17.02.17 ПЗ	Пояснювальна записка	57		
			<u>Графічні матеріали</u>			
2	A2	КВРКІ.170285.17.02.17 Е1	Структура обробки потоку даних	1		
3	A2	КВРКІ.170285.17.02.17 Е2	Концептуальне проектування схеми	1		
4	A2	КВРКІ.170285.17.02.17 Е3	Коригування ресурсів кожної черги пріоритетів	1		

КВРКІ.170285.17.02.17 ВП

Зм	Арк	№ докум	Підпис	Дата				
Розробив		Охота Б.Л.			Засоби динамічного розподілу ресурсів для розподіленої обробки потоків	Літера	Аркуш	Аркушів
Перевір.		Чешун В.М.				У		1
Н. контр.		Муляр І.В.				ХНУ, КІ-17-2		
Затв.		Кльон Ю.П.						

ЗМІСТ

ВСТУП.....	4
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	5
1. ПОНЯТТЯ ПРО РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ.....	6
1.1 Дослідження предметної області.....	6
1.2 Вступ до обробки потоків.....	8
1.3 Кластер Apache.....	9
1.4 Паралельність топології.....	11
1.5 Висновки до розділу 1.....	12
2. МОДЕЛЬНЕ АВТОМАСШТАБУВАННЯ ДЛЯ РОЗПОДІЛЕНОГО ПОТОКУ ОБРОБКИ.....	13
2.1 Системна аналітична модель.....	16
2.2 Модель затримки оператора.....	16
2.3 Модель пропускнуої здатності топології.....	17
2.4 Огляд системи.....	19
2.5 Оцінювач моделі.....	25
2.6 Оцінка ефективності.....	26
2.7 Висновки до розділу 2.....	35
3. МОДЕЛЮВАННЯ СПОЛУЧЕННЯ ТА РЕСУРСІВ ОПЕРАТОРА ДЛЯ ЕФЕКТИВНОЇ ОБРОБКИ ПОТОКІВ.....	36
3.1 Огляд системи.....	43
3.2 Розміщення оператора з урахуванням трафіку та на основі розділів.....	44
3.3 Програми для тестування.....	46
3.4 Моделювання розміщення оператора та місцевості.....	50

						КВРКІ.170285.17.02.17 ПЗ		
Зм	Арк.	№докум.	Підпис	Дата	Засоби динамічного розподілу ресурсів для розподіленої обробки потоків Пояснювальна записка	Літера	Аркуш	Аркушів
Виконав		Охота Б.Л.						
Перевір.		Чешун В.М.						
Н.контр. Затвер.		Муляр І.В. Кльощ Ю.П.						
						ХНУ КІ-17-2		

3.5 Налаштування експерименту.....	52
3.6 Висновки до розділу 3.....	54
ВИСНОВКИ.....	55
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	56
ДОДАТОК А Копія графічної частини.....	58

ВСТУП

Сенсоризація реального світу дає змогу здійснювати моніторинг у реальному часі та інтерактивний контроль у широкому діапазоні доменів (наприклад, військова сфера, охорона здоров'я та техніка). Дані, спочатку генеровані з географічно розподілених датчиків, зазвичай обробляються великим обсягом і очікується, що вони будуть обробляти безперервні потоки даних майже в режимі реального часу. Однак широко застосовувані традиційні рішення для управління даними (наприклад, реляційні системи управління базами даних) не здатні задовольнити зростаючу вимогу до миттєвої обробки, що стосується зростаючого попиту базових обчислень, а також ємності зберігання та непередбачуваних запитів користувачів. Вони в першу чергу спрямовані на отримання результатів обробки з низькою затримкою та високою пропускнуою здатністю. Хоча менша затримка забезпечує миттєву обробку наборів даних і спрямування їх на подальшу обробку, більш високий рівень пропускнуої здатності вказує на те, що система здатна обробляти величезний обсяг даних. Для прискорення були застосовані методи, включаючи переупорядкування / поділ оператора, паралелізацію трубопроводу обробка потоку або підвищення рівня пропускнуої здатності.

Через суворі вимоги до обробки цифрових процесорів, більшість програм обробки потоків чутливі до системних затримок. Наприклад, одна секунда затримки становиться нестерпною в електронній торгівлі, оскільки точність результатів може бути потенційно знижена. Це також нестерпно в системах охорони здоров'я або мережі, оскільки будь-яка затримка може створити ризик для цілі моніторингу (пацієнта програм охорони здоров'я або мережі засобів виявлення аномалій).

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

DSP - Процесор цифрової обробки сигналів

RM – Requirements Management

QoS - набір методів для управління ресурсами пакетних мереж

DAG – Орієнтований ациклічний граф

RTT - час, потрібний для пересилання сигналу від передавача до отримувача

СУБД – Система управління базами даних

PKG – Формат файла

JVM - Java Virtual Machine

ЦП – Центральний процесор

ресурси може сильно варіюватись від операторів до операторів як різновид операцій, задіяних у DSP, а постійно мінливий рівень прийому даних ускладнює оцінку ефективності діяльності окремих операторів. Як результат, автомасштабування пропонується як ефективний прийом для забезпечення еластичності ресурсів, тоді як стратегія планування розміщення оператора є критично важливою при проектуванні автомасштабників. Отже, метод профілювання для операторів з урахуванням стану системи та робочого навантаження обов'язково необхідний, щоб забезпечити ефективне управління ресурсами для DSP. Теорія черг була використана і прийнята при вивченні часу очікування системи, яка здатна інтерпретувати навантаження, а також потребу в ресурсах. На рис. 1.1 зображено структуру обробки потоку даних.

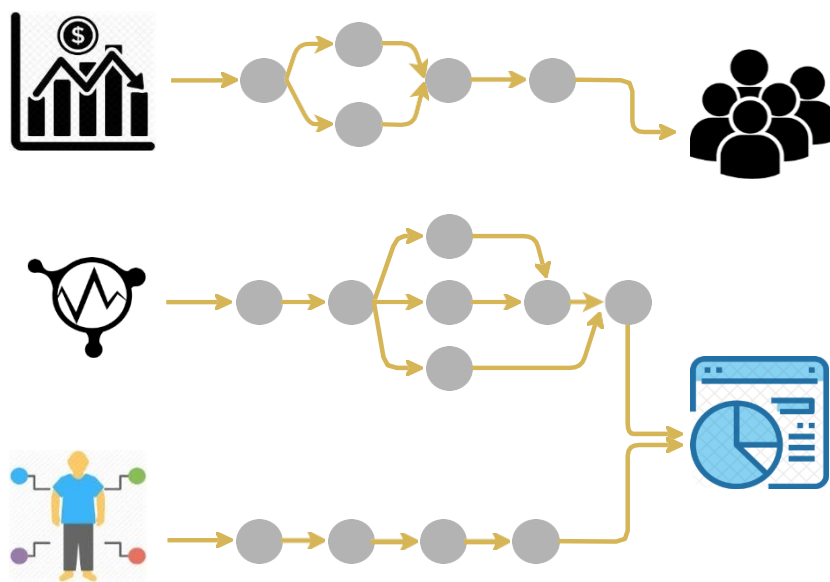


Рисунок 1.1 - Структура обробки потоку даних

Дослідницький інтерес до нової моделі оптимізованої обробки даних зростає протягом останніх років із появою концепції прийняття рішень на основі даних. Зараз обробка потоків стає основною та важливою функціональністю в широкому діапазоні доменів, включаючи фінансовий ринок, кібербезпеку та управління персоналом. Хоча величезний обсяг даних надходить у системи швидким,

безперервним та потоковим способом, для обробки таких потоків даних, що базуються на потоках, було запропоновано систему обробки потоків даних (DSP). Загалом, ефективність таких систем визначається часом реакції на зовнішню подію. У цьому розділі наведено основи обробки розподіленого потоку даних та загальний огляд існуючих систем управління ресурсами та планування в хмарі. Детальний огляд літератури буде наведено в розділі відповідних робіт у наступних розділах. На відміну від традиційних підходів до управління даними, які в основному орієнтовані на зберігання та отримання даних, обробка потоків даних вимагає обробки безперервних потоків даних з мінімальними затримками. Зокрема, потоки даних обробляються на льоту, і зазвичай зберігання та отримання включають лише кінцевий результат обробки замість проміжних даних.

1.2 Вступ до обробки потоків

Додатки, які обробляють потоки інформації, загалом можна класифікувати на два типи - обробку потоку даних (DSP) та обробку складних подій. Хоча DSP схожі на СУБД, оскільки вони обробляють дані через послідовність реляційних операцій з алгеброю, прагне мати справу з множинними, пов'язаними між собою події за допомогою виразів високого рівня. Зокрема, дані зберігаються та індексуються в реляційних базах даних, щоб служити для цілей обробки, і людиною є ініціювати процедуру обробки. Однак дані у форматі безперервних потоків не обов'язково зберігаються, і зовнішня подія в DSP може спричинити взаємодію між користувачем та даними, тобто аномалія, виявлена в системі виявлення вторгнень або попередження файлу в системах моніторингу навколишнього середовища. Отже, очікується отримання приблизного результату, який постійно оновлюється програмами DSP.

Програми DSP дозволяють обробляти безперервні потоки даних у масштабованій та відмовостійкій формі за підтримки розподілених та паралельних методів програмування. Ці кортежі подібні до рядків у реляційних

базах даних, що складається з імені та пов'язаних значень властивостей. На практиці для збору даних може бути залучена велика кількість географічно розподілених датчиків, які розглядаються як джерело даних DSP. Зазвичай є позначка часу, пов'язана з набором даних, щоб відобразити час їх створення або передачі. Як споживач потокових даних, DSP обробляють потоки даних відповідно до заздалегідь визначених конвеєрних операторів [2].

Функції, оголошені в операторах, подібні до операцій, визначених у базі даних запити та типові функції, включаючи, але не обмежуючись ними.

Агрегація: операція згрупування підмножини наборів даних, що стосуються певних умови галузі (наприклад, Aurora, Borealis, StreamBase, IBM InfoSphere Streams, S4, Apache Storm, STREAM та System S). Як перше покоління потокової обробки, Aurora походить від імперативної мови (тобто SQL). Оскільки дані можуть загубитися через більший попит на час реагування, Aurora застосувала зменшення навантаження на основі специфікацій QoS і спрямована на досягнення кращого компромісу між затримкою та пропускну здатністю. Крім того, Aurora надала процедуру тимчасового зберігання для збереження історичних даних для відновлення несправності. Крім того, планувальник розподіляє обчислювальні ресурси відповідно до навантаження оператора та заздалегідь визначеними обмеженнями QoS.

Як одна із найнадійніших розподілених обчислювальних систем у режимі реального часу з відкритими джерелами, Apache Storm¹ призначений для роботи з необмеженим потоком даних у швидкій, масштабованій та відмовостійкій формі. Він широко впроваджений у промисловий секторі відіграє значну роль у додатках для обробки великих даних (наприклад, Twitter, Yahoo!, Groupon і Badu). У цьому розділі запропоновано короткий вступ до Apache Storm, включаючи термінологію, робочий механізм та компонування компонентів в кластері Storm.

1.3 Кластер Apache

На рисунку 1.2 показано архітектуру кластера Apache Storm, який складається майстра і робочі вузли. Головних вузлів може бути будь-яка кількість і кожен із них виконується. Більш конкретно, провідний Nimbus визнає подання заявки та підписує завдання робочим вузлам. Тим часом робочі вузли запускають демон Supervisor, який отримує завдання від Nimbus і виконує виконання програм DSP. Storm UI daemon², який також працює у робочих вузлах, надає інтерфейс прикладних програм як API REST (Representational State Transfer), що дозволяє здійснювати віддалений моніторинг кластера Storm. Більше того, доступність кластеру Storm забезпечується службою Zookeeper³. Пропонується Zookeeper для забезпечення високонадійної розподіленої координації. Зокрема, демон Zookeeper забезпечує високу пропускну здатність та координацію з низькою затримкою між розподіленими процесами разом із загальним простором імен у пам'яті.

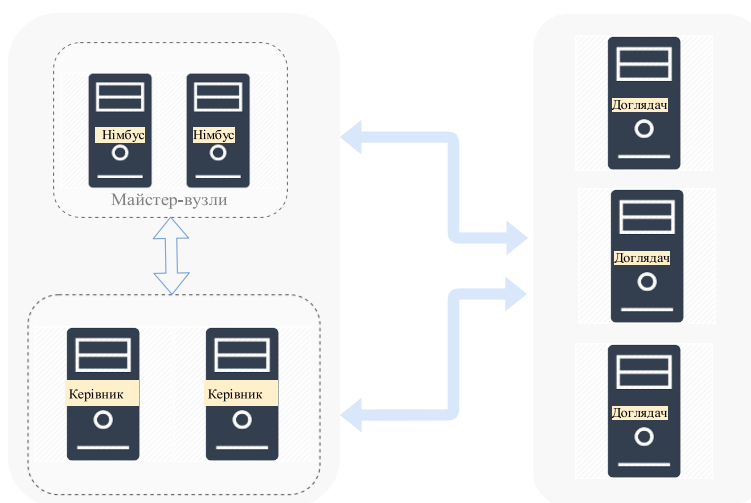


Рисунок 1.2 - Грозовий кластер Apache

Топологія, як окрема програма DSP, є розрахунком модель поточкових даних в режимі реального часу в Apache Storm. Точка джерела топології позначається як <https://storm.apache.org/releases/2.0.0/STORM-UI-REST-API.html> <https://zookeeper.apache.org/>, і він отримує дані з інтегрованих генераторів даних або зовнішніх репозиторіїв даних. Незважаючи на те, що може бути один або декілька носиків, які збирають дані із зовнішніх, вони випромінюють потоки даних нижчим

механізмів групування потоків, і ми коротко оглядаємо три з них, які зазвичай застосовуються в реальних сценаріях.

Часткове групування ключів (PKG) працює аналогічно групуванню полів. Він призначає кортежі на основі вартості конкретного поля, тоді як рівень використання ресурсів буде враховуватися при прийнятті рішень щодо розподілу. PKG застосовує техніку "сили двох варіантів" і вводить розділення ключів. Зокрема, він призначає два вузли-кандидати, отримані двома різними хеш-функціями для кожного ключа. Потім PKG динамічно призначає дані одному з двох варіантів, який має найменше навантаження.

Стани оператора зазвичай реалізуються у форматі пар ключ-значення, і державні кортежі даних повинні передаватися з ключем як ідентифікатором. Докладніше по суті, управління станами здійснюється на ключовій основі, і обов'язково потрібно направляти дані за допомогою одного і того ж ключа до одного і того ж пункту призначення (або екземпляру нижчих операторів). Apache Storm управляє станами, реалізуючи стани в пам'яті, і гарантує стійкість Redis4. Redis - це сховище структур даних із відкритим кодом, яке дозволяє постійно керувати даними. Кожного разу, коли болти із закріпленням стану беруть участь у топологіях Stormy, система автоматично створює носик контрольної точки та встановлює додатковий потік для повідомлень контрольної точки.

1.5 Висновки до розділу 1

Було досліджено основні виклики в управлінні ресурсами та складання графіків для DSP та визначено кілька дослідницьких питань для вирішення кожної із проблем. Дослідницький інтерес до нової моделі оптимізованої обробки даних зростав протягом останніх років із появою концепції прийняття рішень на основі даних. Зараз обробка потоків стає основною та важливою функціональністю в широкому діапазоні доменів, включаючи фінансовий ринок, кібербезпеку та управління персоналом.

надмірного забезпечення бажаного для подолання випадкових кількостей робочих завантажень. Система іноді відчуває кількість, яка поєднується з недостатнім та надмірним забезпеченням, коли частота регулювання є малою, або існує затримка між зміненими робочими завантаженнями та застосованими реконфігураціями [3].

Очікується, що ефективний автоматичний масштабувальник буде незалежним від характеристики робочого завантаження, адаптивним до динамічного середовища потокової передачі даних та до повного розподілу ресурсів у хмарних екосистемах. Тому пропонується залучити цикл MARE до проектування методів автоматичного масштабування, які вимагають чотирьох процесів, а саме моніторинг, аналіз, планування та виконання. Початково, що застосовується цикл MARE у виробничих потоках, починається з моніторингових систем, які співпрацюють і вибирають основні характеристики системи через перші проміжки часу. Функції слід переглядати безпосередньо, щоб вони могли точно описати поведінку систем та допомогти у прийнятті рішень щодо масштабування. Потім автоскалер виконує аналіз та прогнозування на основі дані, зібрані з монітора, проведеного іншою фазою. У зв'язку з інтенсивністю робочого завантаження основним показником опису ресурсів, прогноз кількості робочих завантажень необхідний для досягнення високої продуктивності в хмарних додатках. Нарешті, стадія планування та виконання вирішує розподіл ресурсів та проведення дій відповідно до інструкцій, отриманих в результаті. Про яку еластичність при потоковій обробці як правило, відносини до гнучкості розміщення оператора, автоматичне масштабування в DSP головним використанням стосується рівня паралелізму операторів та стратегії розміщення операційних автоекземплярів. Також передбачається, що етап планування є найкращим для проектування автомасштабників, який визначає протокол розміщення операторів в автоматичному масштабуванні DSP. На цьому етапі слід вирішити основну мету масштабування, щоб гарантувати, що отримані рішення можуть досягти встановлених цін.

Основна мета автоматичного масштабування - забезпечити еластичність ресурсів, щоб забезпечити високопродуктивне виконання хмарних додатків,

перспективі. Як результат, ефективність початкових етапів може бути нижчою, а тривалість фази навчання може бути нездійсненою довгим часом щодо характеру онлайн-навчання. Отже, аналіз часових рядів та вивчення підкріплення є практичними та ефективними рішеннями в довгостроковій перспективі, оскільки на початкових етапах їх не можна забезпечити оптимального рішення для нових розроблених програм.

2.1 Системна аналітична модель

Щоб забезпечити еластичність систем DSP, запропоновано активний підхід до автоматичного масштабування, заснований на теорії масового обслуговування. Оскільки автоматичне масштабування в DSP в основному відноситься до розміщення оператора, автоскалер призначений для розробки ефективних рішень щодо розміщення робочого навантаження додатків та динамічного реагування на зміну попиту користувачів. Оператори DSP, як час прибуття, так і час обробки мають загальний розподіл та доступ до кількох каналів обслуговування. Якщо потрібно, це вказує на те, що для окремих операторів можна зберегти кілька серверів, щоб забезпечити різний рівень паралельності. Крім того, модель затримки приймає завантаження та обчислювальну потужність як вхідну інформацію та оцінку часу очікування відповідно до формули Аллена-Канніна (АС).

2.2 Модель затримки оператора

Затримка системи описує час, витрачений на обробку наборів даних. Затримку ми визначаємо явно, коли весь час перебування набору даних у системі, включаючи час виконання та час очікування. Якщо потрібно, під час виконання головного випадку через пропускну здатність хосту, він залишається незмінним, якщо основна обчислювальна ємність залишається незмінною. Час очікування може сильно змінитися, коли потік відчуває кількість робочих завантажень. Для забезпечення

динамічного регулювання ресурсів для різного завантаження, ми посередньо оцінюємо час очікування та розгляду затримки.

На рисунку 2.1 показано програму DSP з чотирма операторами, які послідовно підключені. Незважно помітити, що обробка може бути потенційно прискорена для збільшення рівня паралельності певних операторів, особливо якщо оператором є вузьке місце.

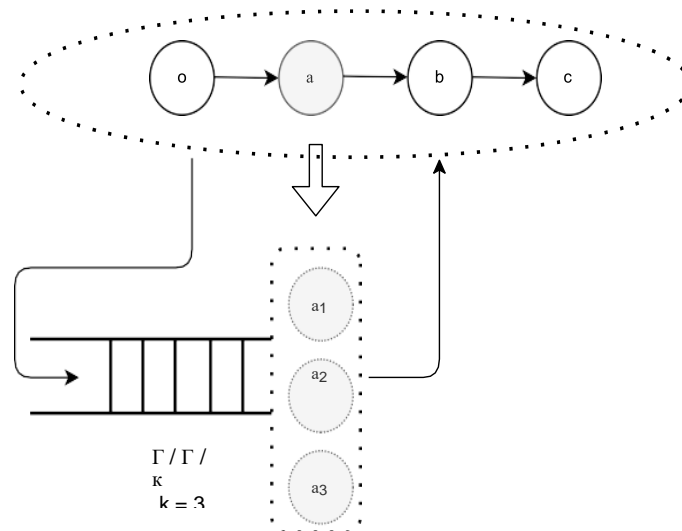


Рисунок 2.1 - Модель черги $G / G / k$ для операторів

Формулу Аллена-Канніна (АС) можна застосувати для приблизно оцінки системного часу даних у черзі із загальним розподілом як часу прибуття, так і часу обслуговування .

2.3 Модель пропускної здатності топології

Визначається пропускна здатність як загальна кількість кортежів може бути виготовлена для тривалого інтервалу, а отже, пропускна здатність топології обчислюється на основі оцінки пропускної здатності окремих операторів. Більше того, побудована для оператора модель пропускної спроможності вимагає швидкості передачі даних як вхідних даних, щоб відображати динамічні зміни робочих завантажень, зазначених даним оператором. Зверніть увагу, що

словами, перший алгоритм визначає розмір вузла для кожного оператора, посилення на цільову пропускну здатність та результати оцінок.

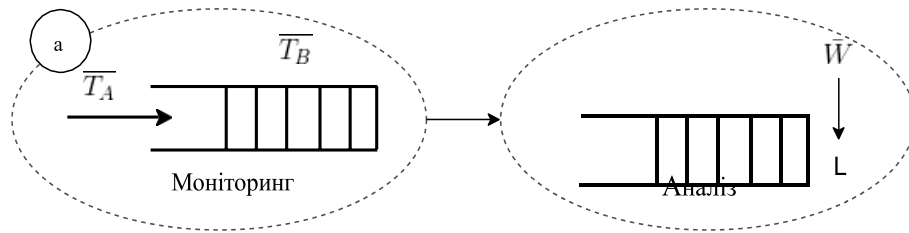


Рисунок 2.3 - Етап моніторингу та аналізу для оцінки затримок.

Для подальшого підвищення точності результатів запропонованих моделей введено процедуру самокорекції для дослідження зовнішніх збурень. Без ознайомлення великі затримки обчислень, система виявляє помилкові оцінки системи виховання в сучасному і розглядає цю помилку як порушення часу. Фаза планування визначає оптимальні рішення для перерозподілу операторів на інших сайтах після оцінки результатів за цільовою продуктивністю.

Обраний тип хостів для кожного оператора може розглядати як орієнтир для вказівки розміру для окремих операторів. Вирішено проблему вузького місця, з якою може бути відкритий будь-який оператор під час виконання, замовлення операторів відповідно до їх очікуваної затримки. Він завжди буде спрямований на визначення пріоритетних операторів, які вказують більшу затримку. На другому кроці вводиться інший алгоритм вибору конкретного хосту для кожного екземпляра оператора щодо доступності ресурсів та запропонованого типу (з першого кроку). Результатом цього алгоритму є відображення між екземплярами оператора та запропонованими хостами. Зрештою, останній етап циклу МАРЕ (якщо фаза виконується) стосується виконання фактичних дій перепланування відповідно до взаємозв'язку відображення [5].

Як результат, метод планування, який націлений на справедливий розподіл операторів DSP, не здатний гарантувати угоди про якість обслуговування.

2.4.1 Схема управління і розміщення ресурсів qos-aware в Stormi

У цьому розділі пропонується метод динамічного планування, керувати ресурсами відповідно до їх очікувань щодо якості обслуговування. Планувальник розроблений як система управління, і він сформульований як задача оптимізації, де набір функцій витрат визначений для досягнення наступних цілей:

а) зменшення суми випадків порушення якості обслуговування в усіх додатках;

б) збільшити рівень використання центрального процесора;

(с) уникнути додаткових витрат, спричинених частими реконструкціями.

Експериментальні результати показали ефективність запропонованого планувальника, що збільшує загальне використання ресурсів в середньому на 23% та зменшує порушення якості QoS на 29% проти стратегії планування чесності Storm.

Що стосується величезного обсягу даних, високої швидкості трафіку даних та різноманітності формату даних, що бере участь у потоках даних, традиційної системи управління базами даних (СУБД) або моделі пакетної упаковки (наприклад, Nadoor), неможливою коробкою безперервних поточкових даних у режимах реального часу. Отже, були запропоновані нові поточкові системи обробки з основною підтримкою забезпечення результатів обробки для коротких затримок, поки система поповнюється найсвіжішою інформацією. Такі системи, також відомі як додаткові великі дані, можна знайти в таких сферах, як фінансовий ринок, розумне місто, кібербезпека та управління охороною здоров'я, а також типовим показником ефективності такої системи є середній час реалізації на зовнішніх поділах.

розробляє деталі проектування та реалізації основних модулів, включаючи монітор, оцінювач моделі та оптимізатор.

2.4.3 Загальні властивості

Беручи до уваги загальну практику, яка застосовує додатки в хмарних екосистемах, DSP зазвичай конкурують за постачання ресурсів на такій платформі для декількох орендарів. Хоча деякі програми DSP дуже чутливі до затримок і вимагають миттєвої обробки їх даних, наприклад, порядком мілісекунд, існують додатки, можна подовжити для більш тривалого часу відгуку, наприклад, у порядку годин або навіть днів. Для вирішення різноманітності програм DSP в очікуванні QoS, ми визначаємо колекцію пріоритетних черг для класифікації додатків відповідно до їх чутливості до затримки. У запропонованому методі планування базові ресурси управляються індивідуально для кожної черги пріоритетів, щоб потенційно уникнути конкуренції між різними типами додатків. Більше того, загальна продуктивність може бути гарантована.

Розроблено систему управління для збору системи та додатків, моделювання поведінки системи та оптимізації продуктивності. На рисунку 2.4 показано концептуальне проектування системи.

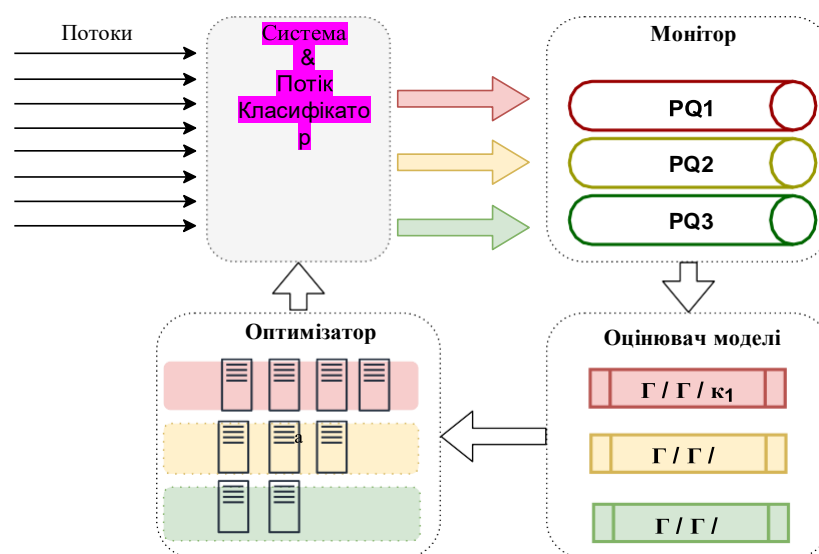


Рисунок 2.4 - Концептуальне проектування запропонованої системи управління

Поки поточкові дані безперервно потрапляють в систему для обробки, потоки даних попередньо обробляються в системному класифікаторі. Зокрема, класифікатор відстежує програму, до якої належить даний набір даних, оскільки програми були позначені пріоритетами на момент їх надсилання до системи. Ярлики надаються кінцевими користувачами, щоб визначити очікуваний час відповіді. Потім набір даних буде спрямований у відповідні черги, які збереглися для його програми, тоді як кілька серверів зберігаються для розміщення додатків у кожній черзі. На другому кроці монітор збирає стан системи (наприклад, наявність ресурсів) та показники програми (наприклад, швидкість передачі даних, пропускну здатність системи) під час виконання [6].

Оцінювач моделі застосовує теорію масового обслуговування для моделювання кожної пріоритетної черги як багатоканальна станція технічного обслуговування та оцінює достатню затримку для прогнозування порушень якості обслуговування. Нарешті, оптимізатор посилається на набір цільових функцій і має на меті знайти оптимальне рішення щодо кількості ресурсів, що надаються для кожної черги пріоритетів. Потім програми, призначені для однакових черг, розподілять виділені їм ресурси. Оскільки динамічний характер програм DSP, робоче навантаження може час від часу змінюватися, і, отже, ресурси, що надаються для кожної черги, будуть динамічно коригуватися відповідно до вимог QoS та уникати випадків порушення.

2.4.4 Класифікатор потоку та Інтернет-монітор

У цьому підрозділі детально розглядаються дані класифікатора потоків та моніторингу процесів. На початку класифікатора пов'язує дані про певну групу хостів для упаковки, а монітор продовжує збирати всю відповідну інформацію про ресурси та програми.

Класифіковані програми розповсюджуються до черги, які мають подібні вимоги до упаковки. Більше конкретно, кожна черга пріоритетів, пов'язаних зі

запропонованого планувальника у відповідь на особливі інтереси користувачів (наприклад, рідше коригування ресурсів, висока толерантність до порушень якості обслуговування).

Створено кластер Apache Storm з дев'ятьма вузлами нагляду, один доглядач зоопарку та один вузол німба. Усі експерименти були проведені в цьому кластері, який побудований на хмарі Nectar1 (Національні інструменти та ресурси співпраці eResearch). Щодо неоднорідності ресурсів, що беруть участь у комерційному хмарному середовищі, вузли супервізора класифікуються на три групи на основі їх асоційованої кількості ядер центрального процесора та обсягу пам'яті. Зокрема, великі віртуальні машини оснащені 12 ядрами і 48 ГБ оперативної пам'яті, середні віртуальні машини мають 8 ядер і 32 ГБ оперативної пам'яті, а малі віртуальні машини пов'язані з 4 ядрами і 16 ГБ оперативної пам'яті. У таблиці 2.3 узагальнено розмір віртуальних машин та кількість вузлів, передбачених для кожної категорії.

Таблиця 2.3 - Розмір та кількість вузлів, наданих для кожного типу наглядачів

Клас	Ядро процесора	ОЗП	№ вузлів
<i>a</i>	4	16GB	3
<i>ed</i>	8	32GB	3
<i>age</i>	12	48GB	3

Оскільки планувальник розроблений для автоматичного регулювання розподілу ресурсів для програм DSP відповідно до їх вимог QoS, дев'ять наглядачів (також відомих як робочі вузли) будуть динамічно зіставлені з певними чергами пріоритетів для розміщення групи додатків. Тут призначено три черги пріоритетів $Q = 3$ з метою оцінки і припускаємо, що залучені

на основі їх чутливості затримки. Зокрема, топологіям у формі зірки надається найвищий пріоритет оцінка, оскільки вона менш терпима до будь-якої затримки. Більше того, програми у формі лінійних призначені для каналу з найнижчим пріоритетом, а алмазні програми спрямовуються до середнього каналу. Крім того, оскільки додаток Twitter було реалізовано як лінійну топологію, воно, таким чином, спрямоване до черги з найнижчим пріоритетом, як інші лінійні топології. Зверніть увагу, що призначення черг може бути переконфігуровано та налаштовано відповідно до вимог користувача. Крім того, може існувати більше черг, що підтримують різноманітність вимог до якості обслуговування програм DSP, хоча це складніше і трудомісткіше для прийняття рішень щодо планування, коли в системі задіяно більше черг.

У таблиці 2.4 наведені топологічні форми, чутливість затримки та індекс черги для всіх чотирьох заявок, що беруть участь в оцінці.

Таблиця 2.4- Тестування програм DSP і призначення черги

Застосування	Чутливість затримки	Індекс черги
<i>ea</i>	низький	3
<i>dad</i>	середній	2
<i>a</i>	високий	1
<i>Te</i>	низький	3

Спочатку провели низку експериментів, щоб обґрунтувати ефективність запропонованого планувальника з точки зору уникнення порушень якості обслуговування. Зокрема, заявки були надіслані в кластер Storm під час конфігурації з трьома планувальниками, а саме QoS-обізнаним, рівним і найкращим планувальником. Порушення QoS були формалізовані як відхилення

середньої затримки від цільового рівня для кожної черги. Він спрямований на підтримання рівня порушення якості обслуговування на рівні нуля, що вважається найкращим сценарієм, коли ресурси, передбачені для черг, точно відповідають запитам на обробку даних програм. Більше того, негативне значення порушення передбачає надмірне забезпечення ресурсами, а позитивне значення означає, що ресурсів недостатньо для вирішення поточного навантаження. Іншими словами, бажано тримати модуль порушення якості обслуговування близько до нуля, оскільки або надмірне, або недостатнє забезпечення ресурси можуть призвести до неоптимальних рішень [9].

Рисунок 2.2 та рисунок 2.3 показують порушення якості обслуговування. Загальні тенденції, налаштовані за допомогою планувальників Even, QoS та найкращих зусиль, також зображені поліноміальними лініями тренду на обох малюнках.

На переконфігурацію пішло приблизно п'ять хвилин, коли операторам потрібно перейти до нещодавно наданих обчислювальних ресурсів і запропонований планувальник може наблизитися до очікувань QoS (із QoS порушення як нуль) для топологій в обох чергах після такого періоду відновлення.

Рівномірний планувальник поступово наближається до очікувань якості обслуговування та досягає прийняттого рівня (близько 0,5) з 16h хвилини. Планувальник найкращих зусиль має подібну тенденцію, як планувальник парних програм із меншою кількістю порушень якості обслуговування. Більше того, рівень порушення зменшується до 0,5 приблизно на п'ять хвилин раніше, ніж планувальник Even. Це в першу чергу викликано динамічною реакцією, проведеною планувальником найкращих зусиль, коли 1 відчуває велике навантаження. Тим часом планувальник, який обізнаний із QoS, досягає нульових порушень протягом 7h хвилини і, як правило, надає трохи більше ресурсів через свою обізнаність про пріоритетність програми. Оскільки 1 це черга з найсуворішими вимогами до обробки та розміщення додатків, надзвичайно чутливих до затримок, вона завжди намагається надати додаткові ресурси для вирішення різкого збільшення попиту на ресурси. Як показано на

малюнку 2.5, планувальник, який відповідає QoS тримає рівень порушення приблизно на нулі з 15h хвилини і, як правило, працює трохи краще, ніж очікували згодом.

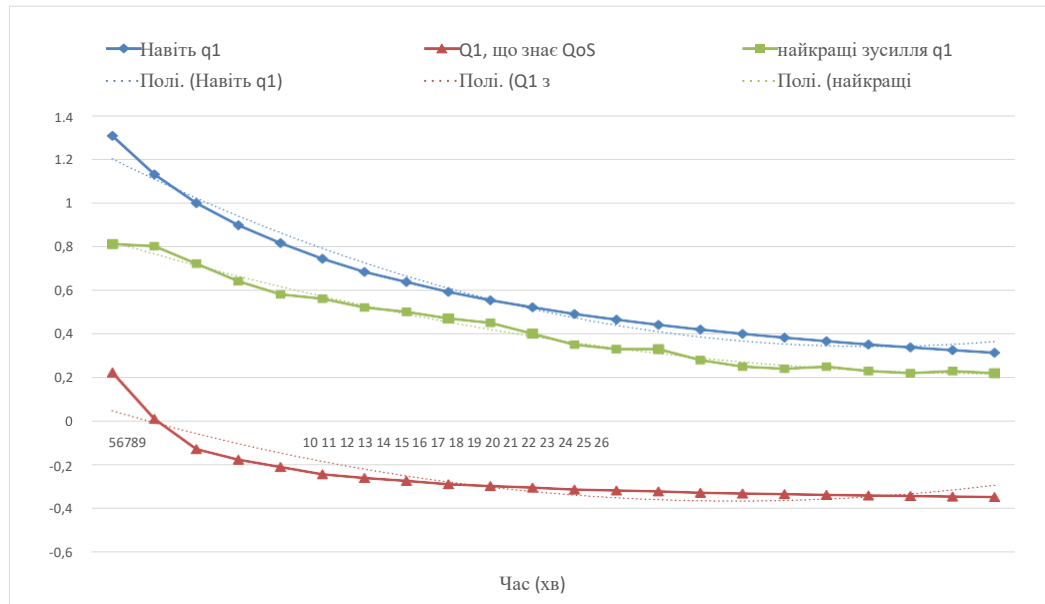


Рисунок 2.5 - Порушення якості QoS

Тим часом, навіть планувальник має негативні порушення переважно під час виконання, оскільки забезпечує більше ресурсів, ніж потрібно. На відміну від цього, планувальник найкращих зусиль зазнає подібних порушень, як це відбувається за 1, і досягає допустимого рівня порушення з 10h хвилини. Помічено, що запропонований планувальник, який обізнаний із QoS, приймає чіткі рішення для програм DSP з різною толерантністю до затримки, і він надає пріоритет тим, хто чутливий до затримок (наприклад, додатки в 1). Однак інші два планувальники, як правило, приймають однакові рішення для різних програм. Планувальник найкращих зусиль торгує використанням ресурсів із порушеннями QoS, і бажано застосовувати його, коли додатки мають подібні вимоги QoS. Таким чином, робимо висновок, що запропонований планувальник, який обізнаний із QoS, може ефективно управляти ресурсами, що стосуються різноманітності вимог QoS серед додатків DSP.

Планувальника QoS, раптове підвищення рівня таких порушень якості погіршить загальну продуктивність. Хоча запропонованому планувальнику вдається стабілізуватись незабаром після періоду реконфігурації, ми намагаємося дослідити можливий вплив рішення про перерозподіл у майбутній роботі, щоб уникнути таких коливань та ще більше підвищити ефективність цього планувальника [10].

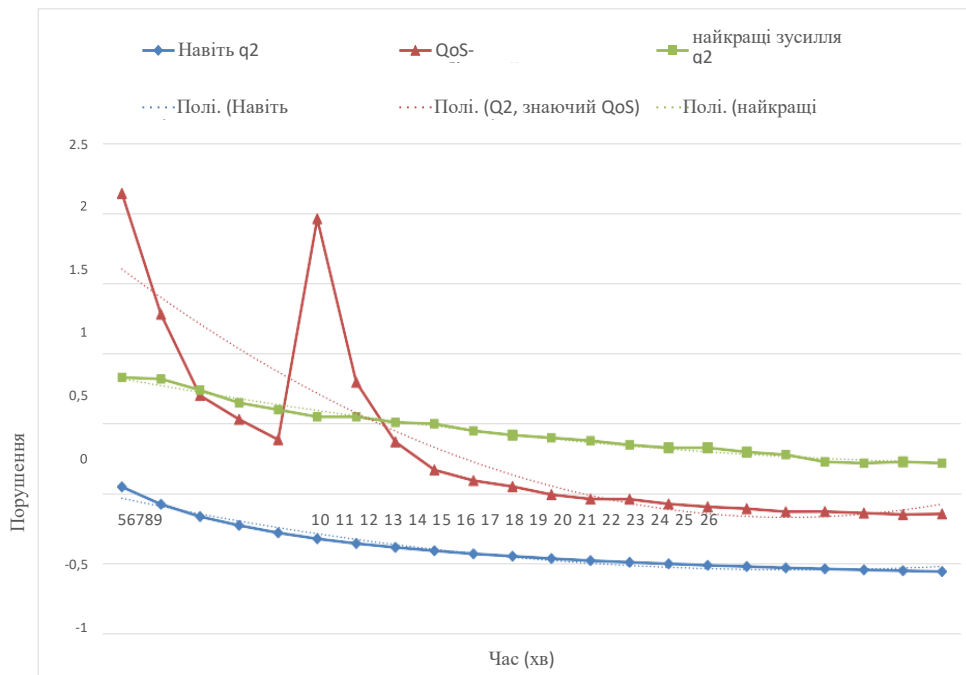


Рисунок 2.6 - Порушення якості QoS, з якими стикаються додатки

Потім порівнюється рівень використання ресурсів планувальника, що знає QoS, та навіть планувальник, коли вони розміщують однаковий набір програм DSP. На рисунку 2.5 і рисунку 2.6 узагальнено середній рівень використання центрального процесора віртуальних машин, пов'язаних з кожною чергою. Ресурси, що належать до кожної черги, спочатку класифікуються як забезпечені та не надані, де надані ресурси поділяються на спожиті та невитрачені. Щоб мінімізувати витрати на підтримку базової інфраструктури, одночасно гарантуючи вимоги до якості обслуговування, вона спрямована на максимізацію співвідношення між фактично спожитими та забезпеченими ресурсами. Іншими словами, основною метою є консолідація робочих місць у меншій кількості

вузлів та переведення незабезпечених ресурсів у режим очікування для економії витрат на підтримку. На рисунку 2.7 показано використання ЦП рівного планувальника. Оскільки він завжди намагається розподілити завдання серед доступних хостів рівномірно, забезпечується 100% доступних ресурсів. Як результат, на обробку поданого споживається 20% наданого ресурсу DSP-програми.

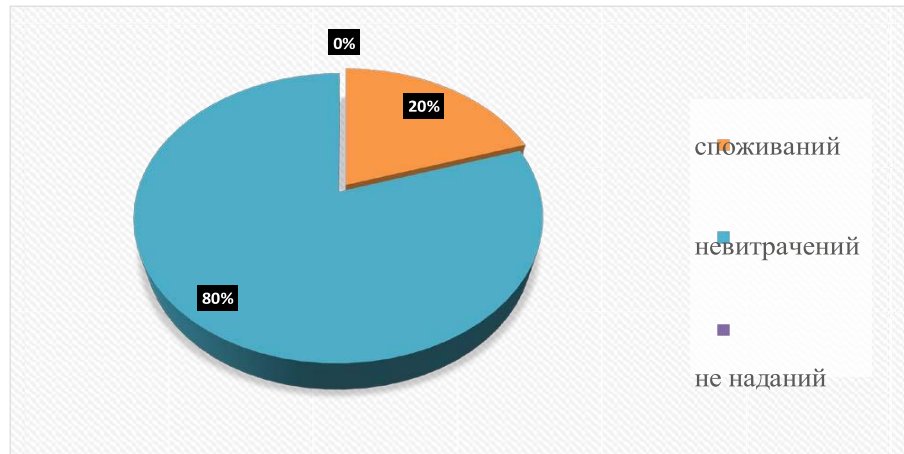


Рисунок 2.7- Використання центрального процесора планувальника Even.

Нарешті, було досліджено ефективність запропонованого планувальника з різними цілями. Більш конкретно, провели серію експериментів, варіюючи коефіцієнти γ , η та δ у цільовій функції, виражені рівнянням. Застосовано три набори конфігурацій щодо коефіцієнтів, а деталі наведені в таблиці 2.5. Зокрема, обґрунтовується ефективність планувальника, що знає про QoS, трьома сценаріями:

- 1) для вирішення важливості зобов'язань щодо QoS, а тим часом мінімізувати споживання ресурсів (*a*);
- 2) уникнути високих витрат, пов'язаних із переконфігурацією (*b*);
- 3) зосередитись насамперед на зобов'язаннях щодо якості обслуговування (*c*).

Експерименти в кожному наборі конфігурації працювали близько 50 хвилин, і досі існують три черги пріоритетів для програм зіркових, діамантових та Twitter-топологічних топологій. Заявки були спрямовані на 1, 2 та 3

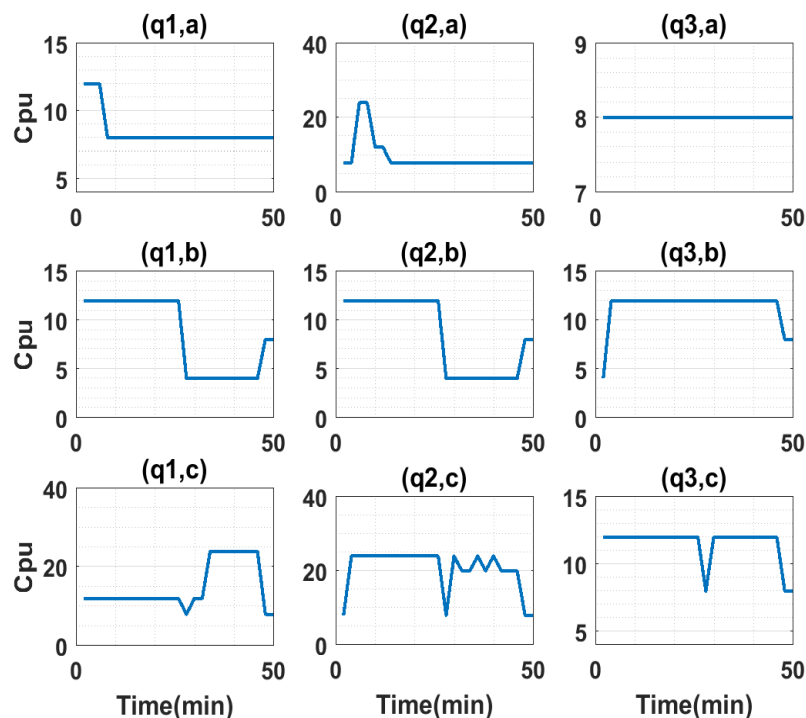


Рисунок 2.8- Коригування ресурсів кожної черги пріоритетів з урахуванням коефіцієнтів установки як конфігурація *a*, *b* та *c*.

2.7 Висновки до розділу 2

Хмарне середовище пропонує еластичність ресурсів, дозволяючи ефективне автоматичне масштабування ресурсу для певних програм. Враховуючи невизначеності та складності, спричинені обробкою потоків, ми запропонували підхід на основі моделі для оцінки пропускнуої здатності системи та затримки, а також розробити рішення щодо масштабування відповідно до результатів оцінки. У цьому розділі представлені дві математичні моделі, що стосуються пропускнуої здатності та затримки, та описується на основі моделі автоматичний масштаботор, призначений для додатків обробки розподіленого потоку. Більше того, протягом усього часу проводилась оцінка щодо ресурсного потенціалу та різноманітних моделей робочого навантаження. Розділ оцінки містить обґрунтування моделей за допомогою трьох програм тестування з різними формами, оскільки компоновання топології може вплинути на результати оцінки запропонованих моделей.

відбуватися лише між цими двома вузлами, першою причиною спостерігається погіршення пропускної здатності.

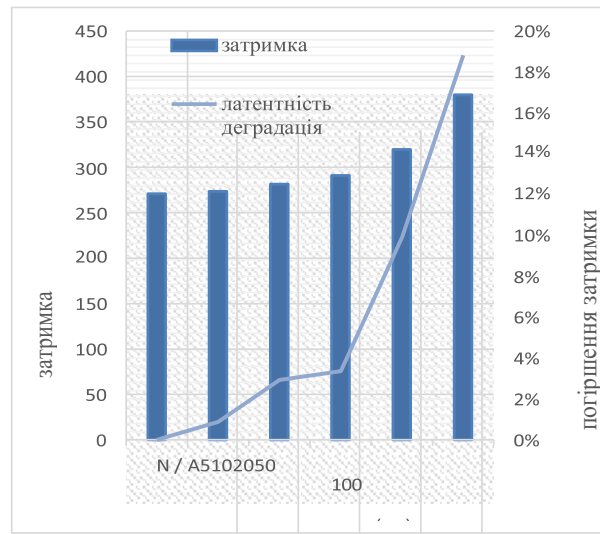


Рисунок 3.1 - Погіршення стану при введенні різного рівня затримок

Рисунок 3.2 ілюструє середню пропускну здатність в кортежах в секунду для топології *a* і *b*, які були першими двома поданими топологіями. Як виявили результати, пропускну здатність топології *b* зменшується приблизно на 42%, а топологія відчуває на 24% менше рівень пропускної здатності, оскільки в систему подано ще дві топології. Однак на практиці очікується, що кілька програм DSP будуть виконуватися одночасно і неприпустимо погіршувати продуктивність існуючих програм під час запуску нових. Тому, по суті, потрібно розробити динамічний метод з урахуванням взаємодії між операторами, а також стану мережі, щоб розмістити операторів для ефективної обробки потоків. На практиці вузли кластера DSP обслуговуються для декількох програм DSP, і тому затримка та пропускну здатність сильно залежать від трафіку між операторами. Як свідчать результати, для ефективної обробки потоків необхідно врахувати затримку мережі та передачу даних [13]. Злиття операторів у більші обчислювані компоненти було запропоновано як ефективний підхід до оптимізації продуктивності DSP, оскільки дозволяє економити значні витрати на комунікацію між операторами. Хоча розподіл усієї топології на один вузол не міг би дослідити наявний

паралелізм трубопроводу, розміщення операторів у окремих вузлах може призвести до значних витрат на зв'язок.

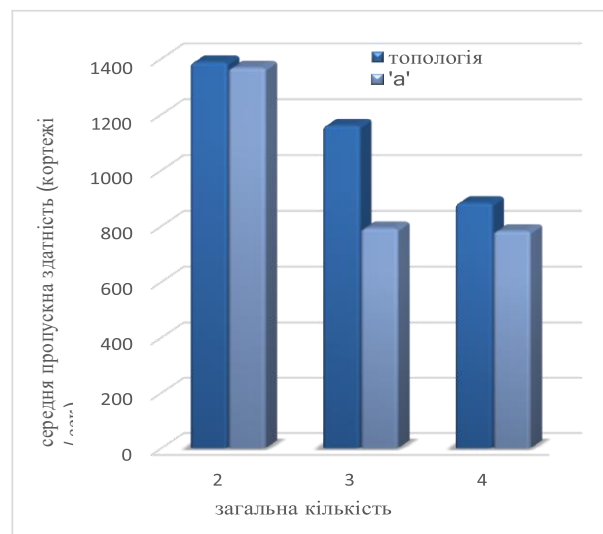


Рисунок 3.2- Середня пропускна здатність, коли працює різна кількість топологій в той самий час

Тому ми застосовуємо багатопотокове злиття в топології, що дозволяє підтримувати кілька копій (або екземплярів) для злитих операторів у похідній топології. Багатопотокове злиття дозволяє нам мінімізувати непотрібне пропускну здатність, одночасно отримуючи переваги від паралелізму конвеєра між географічно розподіленими хостами. Іншими словами, злиті оператори зберігають певні рівні паралельності, тоді як копії (або екземпляри) можуть бути розміщені у вибраній групі хостів. У цьому дослідженні для розділення топологій використовується пакет розділів Metis.

Топологія зазвичай представлена у вигляді прямого ациклічного графіка (DAG), де вершини позначають обчислювані оператори, а ребра ілюструють потік даних між сусідніми операторами. Нехай $T = (O, D)$ позначає топологію, де O - колекція операторів, а D являє собою набори потоків. Metis дозволяє кожній вершині на графіку асоціюватися з розміром та вагою, а також дозволяє ребрам кріпитися з вагою. Щоб задовольнити різноманітні потреби в ресурсах окремих операторів, призначено вимогу ЦП оператора як розмір відповідної вершини, який відображається як LO . Крім того, бажано розподілити обчислювальні інтенсивні

оснащений значними обчислювальними можливостями. Тому потенційно можна уникнути додаткових витрат на зв'язок та споживання смуги пропускання.

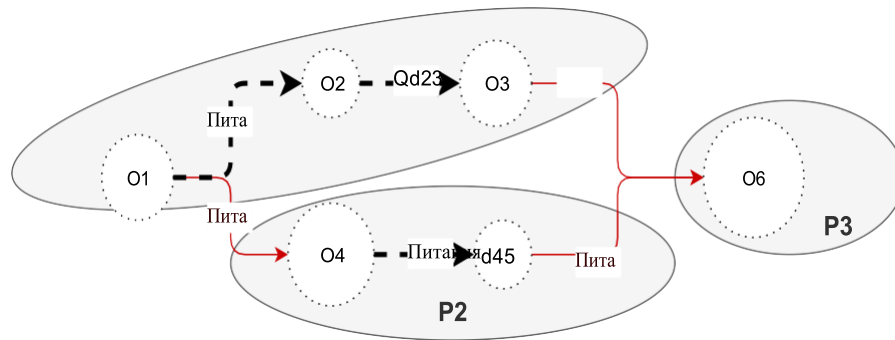


Рисунок 3.3- Результат розподілу топології вибірки

Беручи до уваги неоднорідність розподілених ресурсів, встановлюється тривимірний простір і розміщуємо хости відповідно до їхньої ємності ресурсів, часу зворотного зв'язку на рівні мережі (RTT) та затримок на рівні додатків. Підтримуючи таку систему координат, динамічні зміни розподілених ресурсів, а також умови мережі, можна вчасно захопити та відобразити для прийняття ефективних рішень щодо розміщення.

Відображення віртуальних машин (VM) у n -мірну систему координат відповідно до спостережуваної затримки забезпечує чітке уявлення про стан мережі та між операторським трафіком базового розподіленого середовища. Динамічна зміна координат може додатково інформувати систему з оптимальними конфігураціями розподілу ресурсів. У цій роботі мережеві хости будуть першими, розміщеними у двовимірному просторі, в якому евклідова відстань (E) між вузлом і може точно описати вартість передачі даних між ними. Якщо припустити, що хости обслуговуються лише для виконання DSP, тоді затримка на рівні програми буде домінувати між операторським трафіком від операторів, розташованих на цих двох вузлах. Посилаючись на алгоритм Вівальді для побудови цього простору затримок, який був прийнятий розподіленою хеш-таблицею Chord, і оцінка на великій кількості хостів показала його ефективність та точність.

Іншими словами, він спрямований на мінімізацію споживання пропускної здатності мережі для передачі даних. На рисунку 3.4 показані результати кластеризації 20 вузлів у встановленій моделі ресурсів. Після того, як вузли стабілізували свої координати в просторі, вони були згруповані в 3 групи.

Оскільки кластеризація вузлів насамперед визначається мережею затримки, розміщення операторів може бути спрощене як вибір одного з кластера c (замість усього набору вузлів), і розміщення операторів з тієї ж топології у

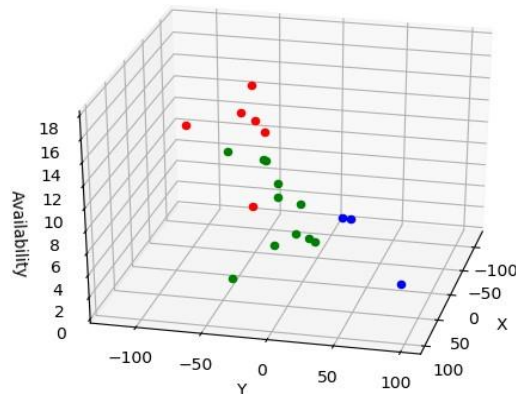


Рисунок 3.4- Результати кластеризації після 20 вузлів, що стабілізують свої координати, включені за допомогою алгоритма Вівальді

З меншою групою вузлів, які беруть участь у розміщенні оператора, вартість, пов'язана з обчисленням розміщення, може бути потенційно знижена. Що ще важливіше, важка комунікація буде спрямована на пару вузлів з невеликою затримкою (вузли всередині кластера), а отже, вона ефективно уникає картографування великих зв'язків між вузлами в ідентифікованих лініях із високою затримкою.

Запропонований метод спрямований на мінімізацію обсягу між операторським зв'язком, що виникає у з'єднаннях із високою затримкою. Це запобігає операторам, які потребують широкої передачі даних, і намагається розмістити похідні великі одиниці в мінімальну кількість господарів. Іншими словами, похідні розділи будуть призначені одному хосту, якщо є такий, який оснащений достатньою місткістю. В іншому випадку очікується, що ці оператори будуть розміщені в

хостах із меншою затримкою, щоб уникнути тривалої затримки. Потім ми детально представляємо постановку проблеми та запропоновані алгоритми [14].

3.1 Огляд системи

Загальна архітектура системи, представлена на рисунку 3.5, складається з трьох абстрактних шарів. Нижній рівень забезпечує інфраструктурну інформацію щодо мережеских хостів. Як обговорювалося в моделі ресурсів, хости розміщуються у тривимірній системі координат, в якій їх затримка в мережі та наявність ресурсів своєчасно відображаються від їх розташування та відстаней між ними. Визначається мережескі посилання з меншою затримкою, а отже, хости кластеризовані на основі спостережуваної затримки. Результат кластеризації динамічно оновлюється відповідно до зміни трафіку. Тим часом верхній рівень збирає інформацію на рівні програми, яка містить, але не обмежується затримкою операторів, попитом на ресурси та між операторським трафіком. Між шаром нанесення та шаром інфраструктури.

Запропонований метод визначає взаємозв'язок відображення між похідними розділами та кластерними хостами для прискорення обробки потоків. Кожний раз, коли не вдається виявити такого хоста зі значною ємністю, розділи будуть додатково розділені, доки всі похідні компоненти не зможуть бути вмонтовані принаймні в один з активних хостів. Більше того, він спрямований на виділення операторів, що походять з однієї і тієї ж топології в межах конкретного кластера, щоб уникнути зв'язку з великою затримкою в програмах DSP.

На основі результатів розділення топології (злиття оператора) та картографування мережі (встановлення моделі ресурсів) формулюється задача, яка дозволяє мінімізувати час, витрачений на передачу наборів даних між похідними розділами.

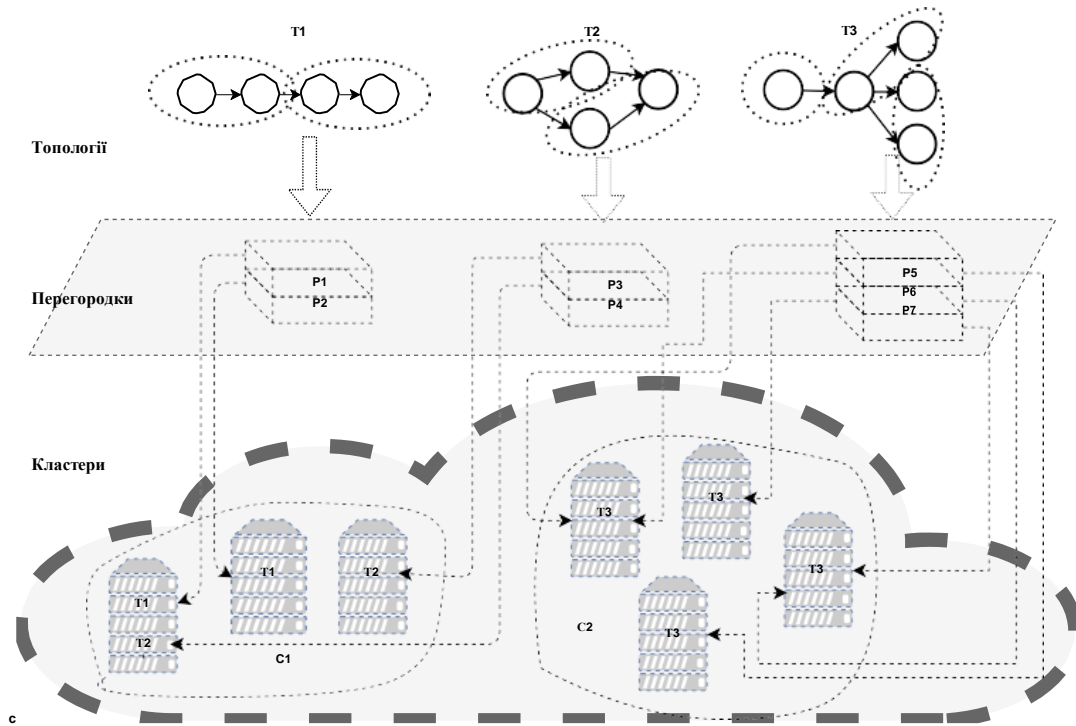


Рисунок 3.5- Концептуальне проектування схеми, що враховує дорожній рух та розділи

3.2 Розміщення оператора з урахуванням трафіку та на основі розділів

Впроваджено запропонований метод в Apache Storm як спеціальний планувальник з урахуванням стану мережі та трафіку між операторами. Під час виконання стратегія планування за замовчуванням Apache Storm, похідна від кругової схеми, застосовуватиметься спочатку до нової поданої топології. Інформація про топологію включає залучених операторів, затримку, і обсяг між операторською передачею даних. Крім того, список хостів необхідний для ініціалізації [15].

Інформація про топологію включає залучених операторів, затримку, і обсяг між операторською передачею даних. Крім того, список хостів required необхідний для ініціалізації. Метод selectCluster кластеризує хости відповідно до останніх координат, посылаючись на своєчасно оновлену модель ресурсів. Потім він

призначає кластер хостів як кандидат-хостів для обслуговування виконання топології. Зокрема, він вибирає кластер з більшістю примірників, які вже працюють на ньому, щоб мінімізувати витрати на перенесення екземплярів операторів. Якщо кількість екземплярів операторів, які працюють у різних кластерах, дорівнює, буде обрано той, який має найвищу доступність ресурсів. Наприклад, існують екземпляри операторів, і ці екземпляри розподіляються між хостами, які належать до двох кластерів *A* та *B*.

На рисунку 3.6 показаний приклад DAG та ваги, пов'язані з операторами. Помічено, що оператор-джерело має вагу 6, що дорівнює рівню операторів у цій DAG. Значення ваги зменшується вздовж трубопроводу, і, отже, оператор раковини зважується як 1. В результаті оператори з меншою затримкою або більше ближче до джерела потоку повинні бути розміщені перед іншими.

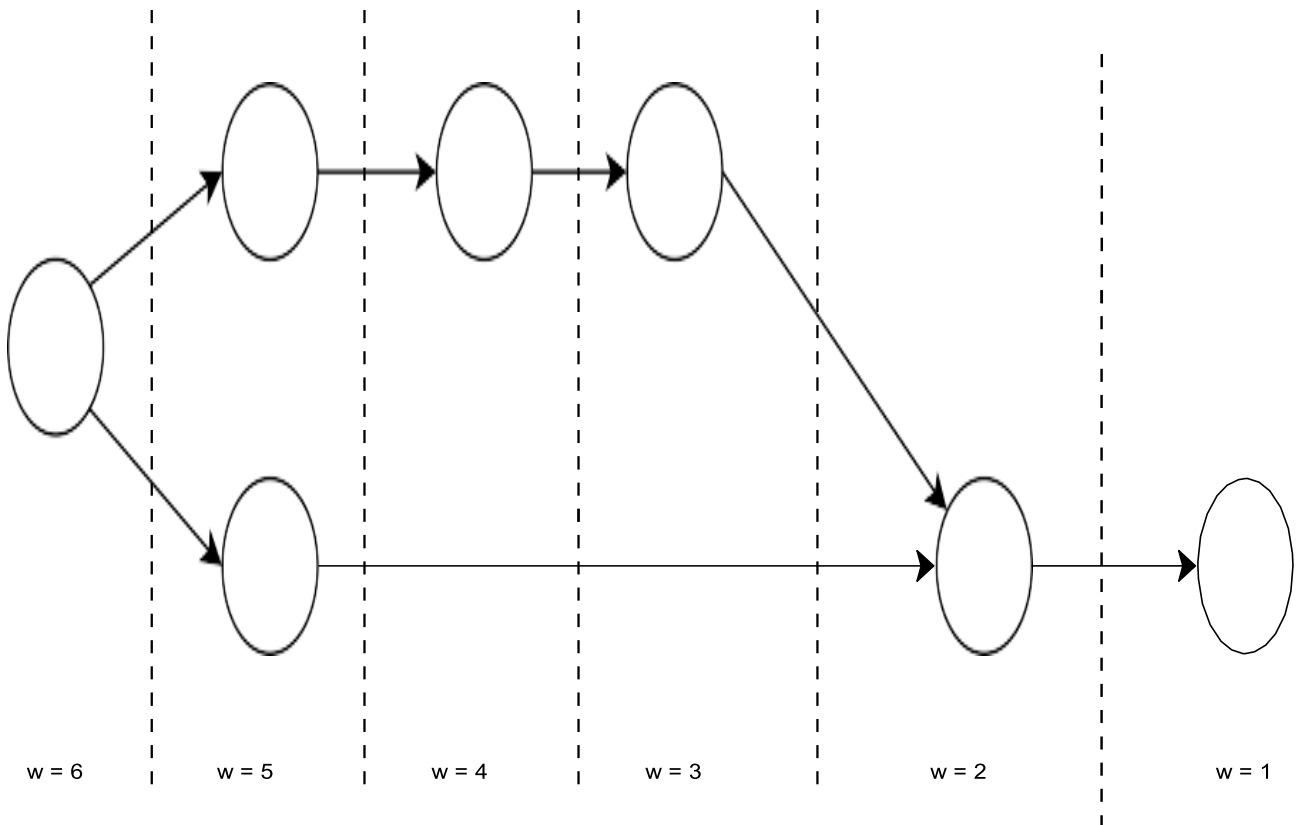


Рисунок 3.6 - Приклад ваги операторів у DAG

Більш конкретно, пропускна здатність планувальника мережі трафіку на 11% перевищує метод *min_Traffic*, тоді як він споживає на 7,78% менше ресурсів для того самого набору програм [17].

Крім того, підхід *max_Resource* призведе до значного споживання смуги пропускання, коли швидкість випромінювання збільшена. Запропонований метод мережі трафіку, навпаки, може запобігти непотрібному мережевому зв'язку та підвищити загальну пропускну здатність. Це особливо добре працює при роботі з вищою нормою викидів.

Далі досліджується ефективність роботи трьох планувальників, тоді як інтервал ковзання, середній інтервал та швидкість випромінювання в *SlidingWindowTopology* були налаштовані як *a*, *b* та *c*. Кожна постановка експериментів проводилась тричі, методами *min_Traffic*, *max_Resource* та *network* тоді як решта налаштувань були ідентичними. Було позначено три зв'язки між чотирма операторами в *SlipWindowTopology* як o1-o2, o2-o3, o3-o4. Отже, кількість даних, що передається за кожним посиленням, може бути чітко представлена. Замість того, щоб просто відстежувати загальний рівень пропускну здатності, коли розгорнуті з різними планувальниками, ми зафіксували трафік між операторами протягом 30 хвилин, щоб інтерпретувати кореляцію між продуктивністю системи та моделі спілкування в межах топології.

3.3.2 Розміщення для операторів

Структура обробки потоків була розроблена у форматі динамічних запитів, які зазвичай отримують та оновлюють історичні дані із зовнішнього сховища. Однак тимчасові результати, в деяких випадках, потрібно зберігати для агрегування чи іншої вдосконаленої семантики. Як результат, ефективна схема управління ресурсами та планування повинна поважати значення державних операторів для прийняття рішень про розміщення.

Стан в потоковій обробці визначається як проміжні результати, пов'язані з послідовністю значень у часі, і, як правило, очікується отримання станів у

наступних операціях. Оператори, не пов'язані з державами, відомі як оператори без громадянства, видають результати обробки в залежності від заданого вводу та суто пересилають результати до подальшого потоку. Хоча вихід операторів без стану залежить від останніх даних, що потрапляють у систему, обчислювальні агрегати або інші вдосконалені статистичні дані вимагають постійного оновлення результату обробки у міру надходження нових даних.

Проміжні результати обробки розглядаються як стан обробки, а стан буфера відстежує дані на вході або виході черги кожного оператора. Більше того, дані в операторах, що містять дані, зазвичай форматовуються в парах ключ-значення, що гарантує, що значення з однаковими ключами можуть бути спрямовані в один і той же потік даних. Зв'язок відображення між ключами та відповідними потоками є записані як стан маршрутизації.

Ефективне управління станами DSP є складним завданням, оскільки необхідно враховувати різноманітні аспекти. Як результат, процедура оновлення стану очікується ефективною та простою у виконанні. Державна міграція - ще один важливий процес використання ресурсів, оскільки схема управління ресурсами може закінчитися перерозподілом кількох державних операторів. Іншими словами, операторам із зазначенням стану потрібно переходити з одного хоста на інший, а отже, операційна функція, а також пов'язані з оператором стани, повинні негайно мігрувати, щоб відновити обробку протягом короткого періоду. Крім того, поточкові дані постійно надходять у систему, і міграція операторів, що мають статус, може порушити узгодженість станів. Нарешті, він завжди спрямований на рівномірне розподілення операторів між хостами. Однак збалансувати навантаження між машинами важко, коли йдеться про потоки, в яких ключі станів не розподілені рівномірно.

Якщо розмір стану оператора відносно невеликий, достатньо зберегти стани в пам'яті, оскільки зберігання в пам'яті забезпечує швидкий доступ до станів без введення додаткових накладних витрат на міграцію стану.

3.4 Моделювання розміщення оператора та місцевості

Оскільки розміщення оператора зазвичай трактується як створення взаємозв'язку відображення між операторами та хостами, моделюється розміщення операторів як двійкова матриця для опису існуючих відносин (E) між вузлами та їх розміщеними операторами. Нехай P представляє можливі схеми розміщення, похідні від а задану стратегію управління ресурсами (наприклад, поінформованість про QoS). Крім того, ми припускаємо, що завжди є кілька потоків, які обробляються паралельно, і система забезпечена хостами для обслуговування виконання. $H(h_0, h_1, \dots, h_{n-1})$ перераховує доступні хости. Потім, наявне розміщення E , а також можливі рішення щодо розміщення $\in P$ моделюється як двійкова матриця з рядками.

3.4.1 Місце розташування для державного оператора з урахуванням місцевості

За допомогою вищезазначеного матричного представлення існуючих та можливих схем розміщення можна зробити швидку оцінку витрат щодо коригувань, особливо коли задіяні оператори з державним статусом. Модель витрат підтримує інтеграцію із схемами управління ресурсами щодо метрик QoS або мережевих умов, і вона розглядає місцевість штату як важливий показник для розробки ефективного розміщення стратегія операторів.

Алгоритм показує розрахунок вартості міграції для операторів із зазначенням стану E до заданого місця розташування. Нехай P перераховані всі доступні варіанти розміщення, отримані за схемою управління ресурсами RM , потім кожне розміщення $\in P$ досліджується і витрати обчислюються, щоб знайти той, який дає мінімальне значення. Іншими словами, алгоритм б повторюється до тих пір, поки не буде знайдена мінімальна вартість міграції після ітерації всіх можливих місць розміщення в P . Припускається, що розмір стану однаковий для операторів, тоді як алгоритм може бути змінений, щоб прийняти розмір стану як додатковий аргумент. Спочатку алгоритм вимагає отримання існуючого місця розташування E ,

запропонована схема, весь набір операторів O , вектор двійкового рядка S , який вказує, чи пов'язаний оператор із станами, місцезнаходження контрольної точки C , якщо це можливо (є принаймні один оператор із зазначенням стану), і остання система координат U що описує затримку мережі між доступними вузлами. Більше того, тимчасова вартість міграції c визначається як 0 на початку.

Щоб оцінити запропоновану стратегію розміщення з урахуванням місцевості, представлено два DSP-додатки, що містять статус і розширив планувальник, виведений модулем для обчислення вартості міграції. За час виконання протягом майже 20 хвилин були зафіксовані рівні затримки та пропускну здатності тестувальних програм, здатних відображати процес міграції. Для порівняння запропонованого методу із існуючою схемою управління станом Apache Storm, ми двічі провели один і той же набір експериментів і зберегли послідовну конфігурацію системи. Крім того, збирали середню затримку та пропускну здатність за кожен інтервал спостереження та досліджували різницю в продуктивності системи до та після застосування запропонованого методу для перерозподілу операторів. Помічено, що пропускну здатність зазнає регулярних стрибків через періодичні контрольні точки.

3.5 Налаштування експерименту

Експерименти проводились в рамках кластера DSP, що складається з шести вузлів супервізора для обслуговування виконання та обслуговування стану для розгорнутих програм DSP. Існує два типи хостів, що стосуються пов'язаної з ними кількості ядер процесора та кількості оперативної пам'яті. Ресурси, виділені кожному класу хостів, і кількість вузлів у кожній категорії наведені в таблиці 3.1. Крім того, виділений вузол Zookeeper та вузол nimbus також включені в кластер Storm для надання основних послуг Apache Storm та забезпечення надійної обробки потоків даних. На початковому етапі застосовується існуюча схема планування Apache Storm для збору метрики системи та реального ресурсу в режимі реального часу.

менші вигоди вона отримує від застосування методу державного визначення місцевості [18].



Рисунок 3.7 - Пропускна здатність топології statefulspoutwindow

Більше того, було збільшено швидкість потрапляння даних у носик до 1000 кортежів на секунду обґрунтував результати попередніх експериментів.

Було розглянуто операцію з увімкненням та відповідні витрати на перенесення операторів з укладанням у поточній обробці. DSP підтримують агрегації та вдосконалену аналітику потокових даних шляхом асоціації станів з операторами. Однак, на відміну від операторів без громадянства, міграція операторів, що мають статус, вимагає додаткових ресурсів та затримує обробку. Тому запропоновано державну стратегію, орієнтовану на місцевість, для розміщення операторів, що мають статус. Було розраховано вартість міграції, представляючи розміщення як матрицю та посилаючись на мережеву затримку між початковим та кінцевим хостами. Запропонований метод був реалізований як індивідуальний модуль, який може бути інтегрований з різними цілями схем управління ресурсами. Для цілей оцінки ми інтегрували його з методом розміщення оператора, який відповідає мережі та базується на розділах. Експериментальні результати показали ефективність запропонованого методу при роботі з цифровими процесорами, що пов'язано з вищим співвідношенням операторів, що містять дані про стан, і зокрема, що відчувають велике навантаження. У нашій подальшій роботі передбачається розробити систему контролю та синхронізації стану для покращення продуктивності подальшої обробки потоку даних.

3.6 Висновки до розділу 3

Ми розглянули взаємозв'язок між мережевими умовами, а також міжвузловим передаванням даних у програмах DSP і запропонували метод для прискорення обробки потоку шляхом мінімізації міжвузлового трафіку та переміщення масової комунікації в мережевих лініях з меншою затримкою. Ми створили тривимірну модель ресурсів, щоб зафіксувати доступність системи та мережі затримки, і вузлів розміщуються в моделі для позначення витрати на зв'язок та їх координати можуть змінюватися внаслідок різниці моделей комунікацій із навантаженням. Потім ми кластеризували вузли відповідно до їх місцезнаходження в просторі, і він спрямований на розміщення операторів, що належать до одного і того ж DSP-додатку, до групи хостів, що мають меншу затримку.

Крім того, ми застосували парадигму багаторівневої рекурсивної бісекції для розділення топологій DSP на кілька підграфів для розподілу цілей. Він приймає рішення щодо розділів, посиляючись на взаємодію між операторами та потребу в ресурсах окремих операторів. Експериментальні результати показали ефективність запропонованого методу в підвищенні продуктивності та поліпшенні використання ресурсів. Це особливо добре працює з програмами, які мають незбалансований взаємозв'язок між операторами або з великим обсягом вхідних даних.

ВИСНОВКИ

Ця кваліфікаційна робота мала на меті розробити ефективні стратегії управління ресурсами для розподіленої обробки потоків щодо різноманітних цілей. Зокрема, було визначено чотири проблеми при розробці стратегій розміщення операторів та управління ресурсами. Для вирішення проблем пропонується п'ять дослідницьких питань, тоді як кожне питання фокусується на одній із проблем із запропонованими рішеннями.

Питання дослідження були визначені для вирішення чотирьох проблем у потоковій обробці:

- 1) різноманітний попит на ресурси та швидкість передачі даних окремими операторами ЦСП.
- 2) різні чутливості до затримок у програмах.
- 3) динамічний стан мережі та схема передачі даних.
- 4) значна вартість мігруючих операторів з державним статусом.

Переглядаючи відповідні роботи та доступні методи, ми запропонували низку рішень з акцентом на визначені проблеми.

Запропонований метод головним чином зосереджений на оптимізації продуктивності окремих програм DSP шляхом пошуку множин операторів з відповідним розміром. Метод автоматичного масштабування оцінювали за допомогою програм DSP у різних топологічних формах. Хоча результати показали різний рівень підвищення пропускної здатності та затримки, виявлено, що запропонований метод автоматичного масштабування прискорює обробку та покращує рівень пропускної здатності додатків у формі алмазу та зірки.

14. Телекомунікаційні системи та мережі: Навч. посібник. У 3 томах. Том 3. Мультісервісні мережі. За ред. В.П. Шувалова. Москва: Гаряча лінія-Телеком, 2005. 592 с.

15. Технології передачі даних. За ред: Хелд Г . Київ: Алерта. 2003. 720 с.

16. Основи мереж передачі даних. Курс лекцій. За ред: В.Г. Оліфер, Н.А. Оліфер. Інтернет-університет інформаційних технологій, 2005. 176 с.

17. Офіційний сайт компанії P.J. Satellit, Lda. URL: <http://2253.ao.all.biz/>.(дата звернення: 22.05.2021).

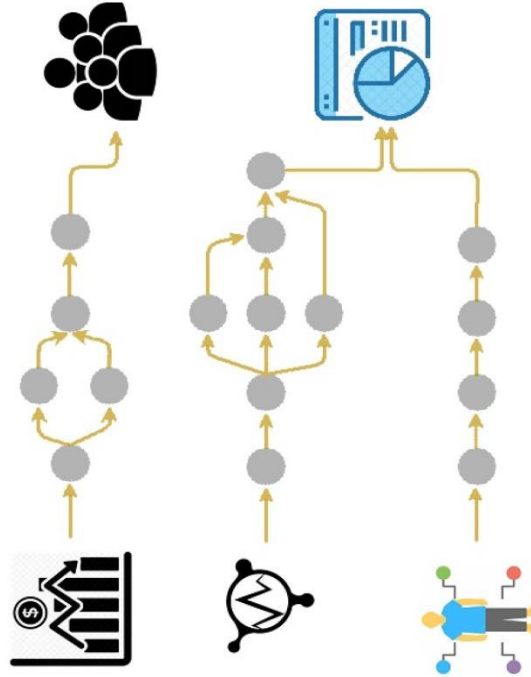
18. Передача дискретної інформації: Підручник для вузів. За ред: Ємельянов Г.А., Шварцман В.О. Москва: Радіо і зв'язок, 1982. 240 с.

ДОДАТОК А

(обов'язковий)

Копія графічної частини

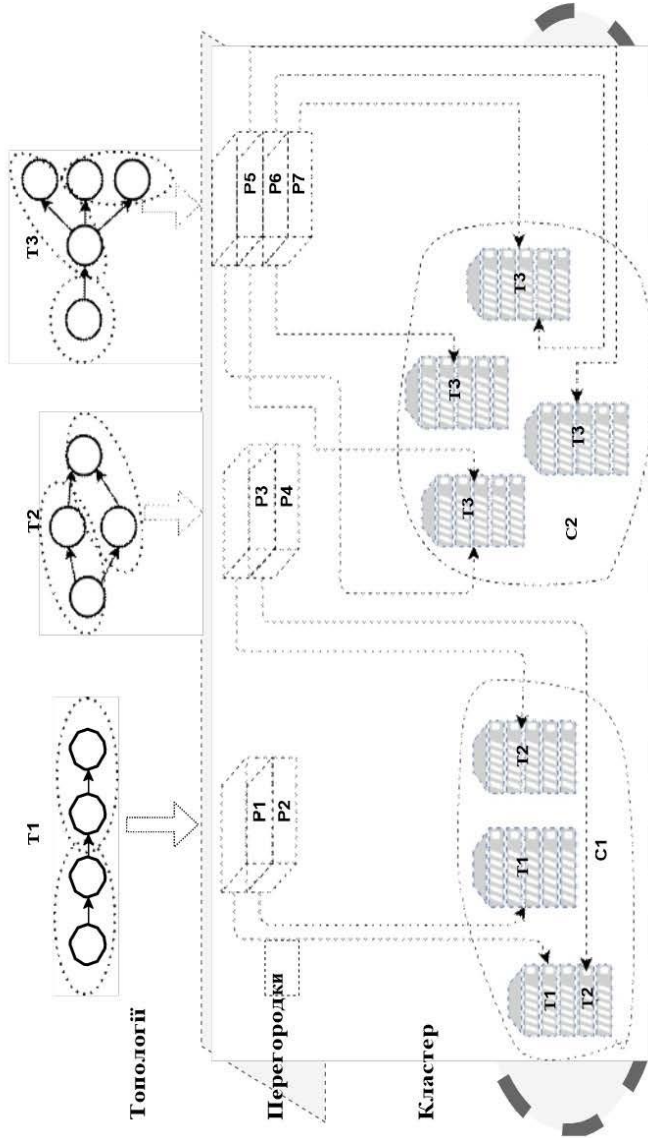
Структура обробки потоку даних



КвРКІ: 170285-17.02.17Е1

КвРКІ: 170285-17.02.17Е1		Гриво	Місяць	Місяць
Зам. Акк.	№ докум.	Підпис	Відом.	
Розроб.	Опубл.	Б.П.	Структура обробки потоку даних	
Перевір.	Стор.	В.М.	Архив.	3
Н. контр.	Мусат	Л.В.	Архив.	1
Т. контр.	Колот	Ю.П.	ХНУ, ГР КІ-17-2	
Зав. пр.				

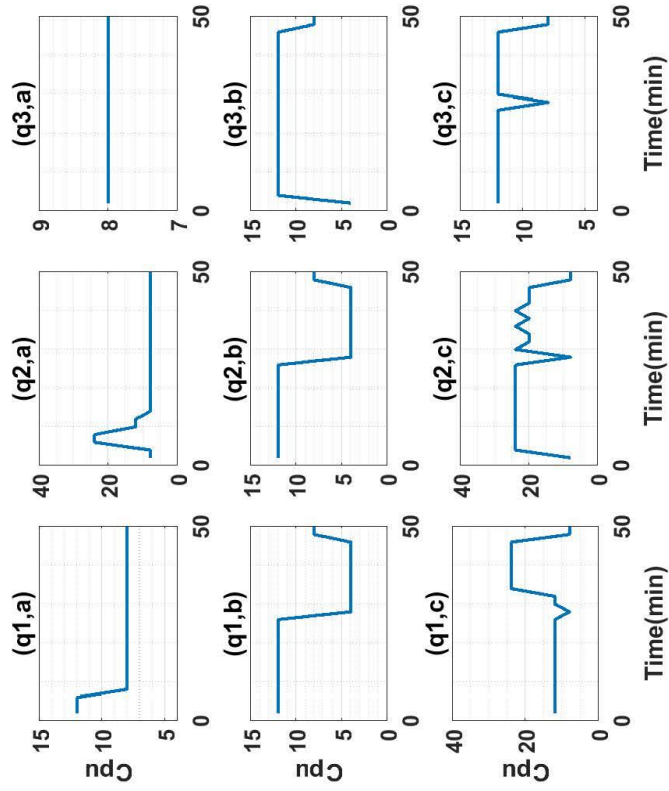
Концептуальне проектування



№	№ докум.	Підпис	Дата
	Розроб.	Сухом Б.О.	
	Перевір.	Мельни В.М.	
	Н. контр.	Мельни В.	
	Т. контр.		
	Затв.	Кочуб Ю.П.	

Літера	Місця	Місця
Концептуальне проектування схемат.		
Аркул 2	Аркул 3	
ХНУ, ГР. КІ-17-2		

Коригування ресурсів кожної черги пріоритетів з урахуванням коефіцієнтів установки як конфігурація а, b та с



КвРКЛ 170285-17.02.17.Е3		Література	Масштаб	Місце
ЗМІ	Автори	№ докум.	Підпис	Дата
Розроб.	Згодн.	Б.п.		
Перевір.	Науков.	Б.п.		
П.контр.	Контр.	І.п.		
Т.контр.				
Завт.	КавецьО.П.			
Характеристика ресурсів кожної черги пріоритетів з урахуванням коефіцієнтів установки як конфігурація а, b та с		Автори		
		Автори 3		
		ХНУ, ІР, Ю-17-2		

User name:
Кафедра кибербезпеки

Check ID:
1008327476

Check date:
18.06.2021 12:27:26 EEST

Check type:
Doc vs Internet

Report date:
18.06.2021 12:30:34 EEST

User ID:
100005590

File name: Пояснювальна записка01_Охота

Page count: 56 Word count: 14493 Character count: 108657 File size: 3.23 MB File ID: 1008398886

Text modifications detected (similarity score might be affected)

0.92% Matches

Highest match: 0.28% with Internet source (<https://www.utupub.fi/handle/10024/150020>)

0.92% Internet sources 134

No Library search was conducted.

0% Quotes

Exclusion of quotes is off

Exclusion of references is off

0% Exclusions

No exclusions

Modifind

Text modifications detected. Find more details in the online report.

Replaced characters 13

Suspicious formatting 21 Pages

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 0.0%

Словари проверки: en_US, ru_RU, ua_UA. Ошибок в документах: 6%

ID: 94726 Название: Засоби динамічного розподілу ресурсів для розподіленої обробки потоків Добавлено в БД: 2021-06-18 Авторы: Охота Б.Л.. Руководители: Чешун В.М. Консультанты: Опоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	82975	576	133 (0%)	3 (1%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
освітнього ступеня «бакалавр»

Студент Охота Богдан Леонідович

Тема Засоби динамічного розподілу ресурсів для розподіленої обробки потоків
Спеціальність 123 – Комп'ютерна інженерія

Обсяг кваліфікаційної роботи освітньо-кваліфікаційного рівня «бакалавр»:

кількість листів креслень 3; кількість сторінок записки 57.

1. Короткий зміст роботи та прийнятих рішень Метою кваліфікаційної роботи є проєктування та реалізація засобів динамічного розподілу ресурсів для розподіленої обробки потоків.

2. Висновок про відповідність кваліфікаційної роботи завданню Кваліфікаційна робота є в повній мірі відповідає поставленому завданню

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: Розділ 1 – досліджено основні виклики в управлінні ресурсами та складання графіків для DSP та визначено кілька дослідницьких питань для вирішення кожної із проблем. Розділ 2 – було представлено дві математичні моделі, що стосуються пропускну здатності та затримки, та описується на основі моделі автоматичний масштабатор, призначений для додатків обробки розподіленого потоку. Більше того, протягом усього часу проводилась оцінка щодо ресурсного потенціалу та різноманітних моделей робочого навантаження. Розділ 3 – розглянуто взаємозв'язок між мережевими умовами, а також міжвузловим передаванням даних у програмах DSP і запропонували метод для прискорення обробки потоку шляхом мінімізації міжвузлового трафіку та переміщення масової комунікації в мережових лініях з меншою затримкою.

4. Позитивні сторони роботи Кваліфікаційна робота має комплексну практичну цінність. Сенсоризація реального світу дає змогу здійснювати моніторинг у реальному часі та інтерактивний контроль у широкому діапазоні доменів.

5. Негативні сторони роботи Розроблений в роботі модуль має вузький функціонал

6. Оцінка графічного оформлення та пояснювальної записки роботи Графічне оформлення виконане відповідно до темі кваліфікаційної роботи з дотриманням стандартів. В загальному графічне оформлення виконане якісно, пояснювальна записка відповідає нормам щодо її оформлення.

7. Відгук про роботу в цілому В загальному кваліфікаційна робота заслуговує позитивної оцінки.

8. Інші зауваження

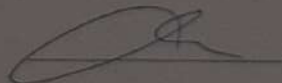
9. Оцінка кваліфікаційної роботи Враховуючи всі позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що вона заслуговує оцінку «задовільно» D.

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи)

Богданова О.В.

Здобувачем КВІТ

« 14 » червня 2021.

 (підпис)

Завідувачу кафедри КІСП
к-т.техн.наук, доцент. Кльоц Ю.П.

Охота Богдан Леонідович

ПІБ здобувача вищої освіти

ФПКТС, 4 курсу, групи КІ-17-2

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 29.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповішений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

18.06.2021

дата



підпис

5. Негативні сторони роботи Розроблений в роботі модуль має вузький функціонал

6. Оцінка графічного оформлення та пояснювальної записки роботи Графічне оформлення виконане відповідно до теми кваліфікаційної роботи з дотриманням стандартів. В загальному графічне оформлення виконане якісно, пояснювальна записка відповідає нормам щодо її оформлення.

7. Відгук про роботу в цілому В загальному кваліфікаційна робота заслуговує позитивної оцінки.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи Враховуючи всі позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що вона заслуговує оцінку «задовільно» Р.

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи)

Борисенко О.В.

Зіт. Коляда КМІТ

« 14 » серпня 2021.

 (підпис)

Завідувачу кафедри КІСП
к-т.техн.наук, доцент, Кльоц Ю.П.

Охота Богдан Леонідович

Піш. здобувач вищої освіти

ФПКТС, 4 курсу, групи КІ-17-2

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 29.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а) Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповищення (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

18.06.2021

дата


підпис

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА СИСТЕМНОГО ПРОГРАМУВАННЯ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованою системою виявлення текстових збігів/ідентичності/схожості:

Назва: Засоби динамічного розподілу ресурсів для розподіленої обробки потоків

Автор: Охта Богдан Леонідович

Спеціальність: 123 – Комп'ютерна інженерія та програмування

Освітня програма: освітньо-професійна

Науковий керівник: Чешун Віктор Миколайович, старший викладач

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданій поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданій поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та дороблена і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укривти запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформлені посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
- 4) в якості запозичень в окремих місцях системою зафіксовано послідовності чотирьохрозрядних двійкових кодів, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
- 5) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів та україномовними скороченнями індексів в формулах, що не є модифікацією тексту (Тут текст можна і треба модифікувати)

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 0,92% і адресується до 134 періоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи



V. M. Чешун

Гарант ОП



S. M. Лисенко

Завідувач кафедри КІСП



Ю. П. Ключ