

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр  
Освітній рівень

Мультикомп'ютерна система згідно топології «подвійне кільце»  
Назва теми

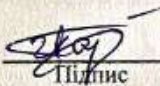
КвРКІ. 190106.19.01.03 ПЗ  
Шифр

Галузь знань 12 «Інформаційні технології»  
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»  
Шифр, назва


Освітня програма «Комп'ютерна інженерія та програмування»  
Назва

Виконала: студентка IV курсу, група K12-19-1

  
Підпис

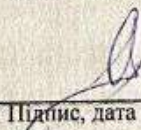
Гоба К.С.  
Ініціали, прізвище

Керівник

 2.06.2023  
Підпис, дата

Савенко О.С.  
Ініціали, прізвище

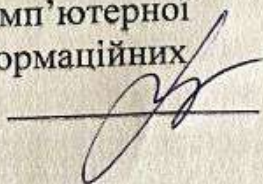
Нормоконтролер

  
Підпис, дата

С.М. Лисенко  
Ініціали, прізвище

До захисту допускаю:

Зав. кафедри комп'ютерної  
інженерії та інформаційних  
систем



Т.О. Говорушенко  
Підпис

Ініціали, прізвище

« 5 » червня 2023 р.

Хмельницький 2023

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЯ ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Г.О.Говорущенко

“ 11 ” 01 2023 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Гобі Каріні Сергіївній

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Мультикомп'ютерна система згідно топології «подвійне кільце»

Керівник проекту (роботи) Савенко О.С., д.т.н., професор

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

Затверджена наказом ректора університету від 1.03.2023 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2023 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 програмно-технічного засобу

Проектування програмно-технічного засобу


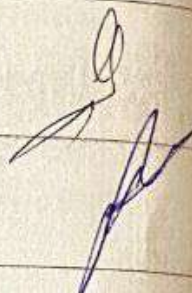
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Алгоритм та UML - діаграма підєднання складових мультикомп'ютерної системи

Схеми будов архітектур мультикомп'ютерних систем

Моделювання різних ситуацій використання мульткомп'ютерних систем

6. Консультанти розділів дипломного проекту (роботи)


Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прий
Нормоконтроль	Лисенко С.М., професор кафедри КПС		
Антиплагіат	Нічепорук А.О., доцент кафедри КПС		

7. Дата видачі завдання « 01 » 03 2023 р.

**КАЛЕНДАРНИЙ ПЛАН**

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Приміт
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	20.02.2022	викона
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.03.2023	викона
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	10.03.2023	викона
4	Робота над розділом 2 – проектування програмно-апаратного засобу	20.04.2023	викона
5	Робота над розділом 3 – проектування програмно-апаратного засобу	30.04.2023	викона
6	Оформлення пояснювальної записки згідно вимог	11.05.2023	викона
7	Попередній захист ВКР	26.05.2023	викона
8	Захист ВКР на засіданні ЕК	Червень 2023 року	

Студент

  
Підпис

К.С. Гоба  
Ініціали, прізвище

Керівник проекту (роботи)

  
Підпис

О.С. Савенко  
Ініціали, прізвище

№ р я д к а	ф о р м а т	Позначення	Найменування	К і л - л и с т і в	№ ек з	П р и м і т к а
1		КвРКІ 190106.19.01.03 ПЗ	Текстові документи Пояснювальна записка	69		
2		КвРКІ 190106.19.01.03 Е8	Графічні матеріали Алгоритм та UML- діаграма під'єднання складових мультимедійної системи	1		
3		КвРКІ 190106.19.01.03 Е8	Схеми будов архітектур мультимедійних систем	1		
4		КвРКІ 190106.19.01.03 Е8	Моделювання різних ситуацій використання мультимедійних систем	1		

КвРКІ 190106.19.01.03 ВП

Зм	Арк	№ докум	Підпис	Дата
Розробив		Гоба	<i>Гоба</i>	
Перевір.		Савенко	<i>Савенко</i>	2.06
Н. контр.		Лисенко	<i>Лисенко</i>	
Затв.		Говоруценко	<i>Говоруценко</i>	05.06

Літера	Аркуш	Аркушів
Відомість проекту		
ХНУ, КІ2-19-1		

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Мультикомп'ютерна система згідно топології «подвійне кільце»».

Автор роботи: Гоба Каріна Сергіївна.

Керівник роботи: Савенко Олег Станіславович.

Пояснювальна записка: 69 с., 43 рис., 0 табл., 4 дод., 60 джерел.

Графічна частина: 3 креслення.

### МУЛЬТИКОМП'ЮТЕРНА СИСТЕМА, ТОПОЛОГІЯ «ПОДВІЙНЕ КІЛЬЦЕ», ТОПОЛОГІЯ.

Метою є створення саме мультикомп'ютерної системи загального призначення на основі топології «подвійне кільце».

У цій роботі буде проаналізовано різні архітектури, що можуть використовуватись в системах з розподіленою пам'яттю, проведено аналіз наявних архітектурних рішень. У другому та третьому розділах буде описано проектування такої системи з детальним обґрунтуванням вибору та застосування апаратно-програмної бази. А також буде проаналізовано декілька сценаріїв використання мультикомп'ютерних систем.

Підпис студента







Дата 1.06.2023

## ЗМІСТ

<b>СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ</b> .....	<b>3</b>
<b>ВСТУП</b> .....	<b>4</b>
<b>1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ..</b>	<b>6</b>
1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей .....	6
1.2 Актуальність розвитку мультикомп'ютерних систем .....	10
1.3 Аналіз системного програмного забезпечення .....	10
1.4 Проміжне програмне забезпечення .....	14
1.5 Поняття топології та її види .....	16
1.6 Висновки .....	20
<b>2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ</b> .....	<b>22</b>
2.1 Вибір топології "подвійне кільце" .....	22
2.2 Види архітектур мультикомп'ютерної системи .....	25
2.2.1 Векторно-конвеєрні комп'ютери .....	25
2.2.2 Обчислювальні системи з розподіленою пам'яттю .....	27
2.2.3 Кластерні системи .....	29
2.2.4 Вибір архітектури мультикомп'ютерної системи .....	34
2.3 Вибір та опис апаратного забезпечення .....	37
2.3.1 Ноутбуки для мультикомп'ютерної системи .....	37
2.3.2 Комутатор з підтримкою кільцевої топології .....	38
2.3.3 Маршрутизатори для забезпечення зв'язку між різними сегментами кільцевої топології .....	39
2.3.4 Налаштування маршрутизатора Cisco .....	41
2.3.5 Мережевий адаптер .....	45
2.3.6 Кабель для підключення до обох кілець .....	46
2.4 Висновки .....	47

КВРКІ. 190106 19.01.03 ПЗ

Зм.	Арк.	№докум.	Підпис	Дата	Літера	Аркуш	Аркушів
Виконав		Гоба К. С.					
Перевір.		Савенко О.С.		2.06		2	69
Н.контр.		Лисенко С.М.			ХНУ, КІ-19-1		
Затверд.		Говорущенко Т. О.		05.06			

Мультикомп'ютерна система згідно топології «подвійне кільце»  
Пояснювальна записка

<b>3 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ .....</b>	<b>49</b>
3.1 Алгоритм під'єднання складових мультимп'ютерної системи.....	49
3.2 Вибір програмних засобів .....	50
3.3 Архітектура розподілених об'єктів .....	52
3.4 Моделювання різних ситуацій використання мультимп'ютерних систем по архітектурі типу клієнт - сервер .....	55
3.5 Архітектура програмного забезпечення .....	64
3.6 Висновки .....	65
<b>ВИСНОВКИ .....</b>	<b>67</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ .....</b>	<b>70</b>
Додаток А Лістинг програмного забезпечення.....	76
Додаток Б Алгоритм та UML - діаграма під'єднання складових мультимп'ютерної системи .....	78
Додаток В Схеми будов архітектур мультимп'ютерних систем .....	79
Додаток Г Моделювання різних ситуацій використання мультимп'ютерних систем .....	80

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

EOM - електронно-обчислювальна машина

SIMD - Single Instruction Multiple Data (одиначний потік команд)

MIMD - Multiple Instruction Multiple Data (множинний потік команд)

MSIMD - Multiple Single Instruction Multiple Data (архітектура з паралельно-централізованим керуванням потоком даних)

TCP - Transmission Control Protocol (протокол керування передаванням)

IP - Internet Protocol (протокол)

UDP - User Datagram Protocol (протокол датаграм користувача)

HTTP - Hypertext Transfer Protocol (протокол передачі даних)

FTP - File Transfer Protocol (протокол передавання файлів)

CORBA - Common Object Request Broker Architecture (загальна архітектура брокера об'єктних запитів)

DCOM - Distributed Component Object Model (власна технологія Microsoft для організації взаємодії між компонентами програмного забезпечення)

SOA - Service-oriented architecture (архітектурний шаблон програмного забезпечення, модульний підхід до розробки програмного забезпечення)

MPP - Massively parallel processing (масово-паралельна архітектура)

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		3

## ВСТУП

Електронні обчислювальні машини (ЕОМ) та обчислювальна техніка - це технології, які використовуються для збору, обробки, зберігання та передачі інформації. Ці технології мають велике значення для багатьох галузей, включаючи бізнес, науку, медицину та інженерію.

ЕОМ були винайдені в 20-му столітті і змінили світ. Вони забезпечили нові можливості для обробки інформації та виконання складних обчислень. Починаючи з перших ЕОМ, які використовувалися військовими під час Другої світової війни, обчислювальна техніка стала все більш доступною та потужною, що дозволило їй застосовуватися в різних сферах діяльності.

Сьогодні ЕОМ та обчислювальна техніка є невід'ємною частиною більшості сучасних організацій. Вони використовуються для збору та аналізу даних, розробки програмного забезпечення, візуалізації та симуляції процесів, а також для забезпечення комунікації та зв'язку.

Мультикомп'ютерна система - це комп'ютерна система, що складається з багатьох взаємопов'язаних комп'ютерів, які працюють як єдине ціле. У такій системі кожен комп'ютер зазвичай має свою власну пам'ять та процесор, але вони можуть обмінюватися даними та задачами через мережу.

Однією з основних переваг мультикомп'ютерних систем є їхній високий рівень масштабованості. Будь-яку систему можна легко розширити, додавши до неї додаткові комп'ютери, які будуть виконувати нові завдання. Це забезпечує велику гнучкість та можливість збільшення продуктивності системи в залежності від потреб користувачів.

Іншою перевагою мультикомп'ютерних систем є можливість забезпечення високої надійності. Якщо один з комп'ютерів в системі відмовить, інші комп'ютери можуть продовжувати роботу, забезпечуючи безперервну роботу системи. Крім того, мультикомп'ютерні системи можуть бути розміщені на великій відстані один від одного, що забезпечує високу рівень доступності.

Метою цієї роботи є створення саме мультикомп'ютерної системи загального призначення на основі топології «подвійне кільце». У роботі буде

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		4

проаналізовано різні архітектури, що можуть використовуватись в системах з загальною пам'яттю, проведено аналіз наявних архітектурних рішень. У другому та третьому розділах буде описано проектування такої системи з детальним обґрунтуванням вибору та застосування апаратно-програмної бази.

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		5

# 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Будь-яка обчислювальна система досягає своєї найвищої продуктивності, завдяки використанню високошвидкісних елементів і паралельному виконанню великого числа операцій.

Паралельні ЕОМ поділяються за класифікацією Флінна на машини типу SIMD (Single Instruction Multiple Data) - один потік команд при множині потоці даних; і MIMD (Multiple Instruction Multiple Data) - множинний потік команд.

Можна виділити чотири основні типи (рисунок 1.1) архітектури систем паралельної обробки [1-2].

1) конвеєрна і векторна обробка - основу конвеєрної обробки складає роздільне виконання деякої операції в кілька етапів з передачею даних одного етапу наступному;

2) машини типу SIMD - дані машини складаються з великого числа ідентичних процесорних елементів, що мають власну пам'ять. Всі процесорні елементи в такій машині виконують одну і ту ж програму;

3) машини типу MIMD - у мультипроцесорній системі кожен процесорний елемент виконує свою незалежну програму. У мультипроцесорах із загальною пам'яттю (сильно пов'язаних мультипроцесорах) є пам'ять даних і команд, доступна всім процесорним елементам. До машин типу MIMD відносять векторно-конвеєрні комп'ютери, масово-паралельні комп'ютери;

4) багатопроцесорні машини з SIMD процесорами - сюди відносять супер-ЕОМ, що представляють собою багатопроцесорні системи, в яких в якості процесорів використовуються векторні (MSIMD). Дані машини дають можливість використовувати одночасно векторні операції і гнучкі можливості MIMD архітектури.

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		6

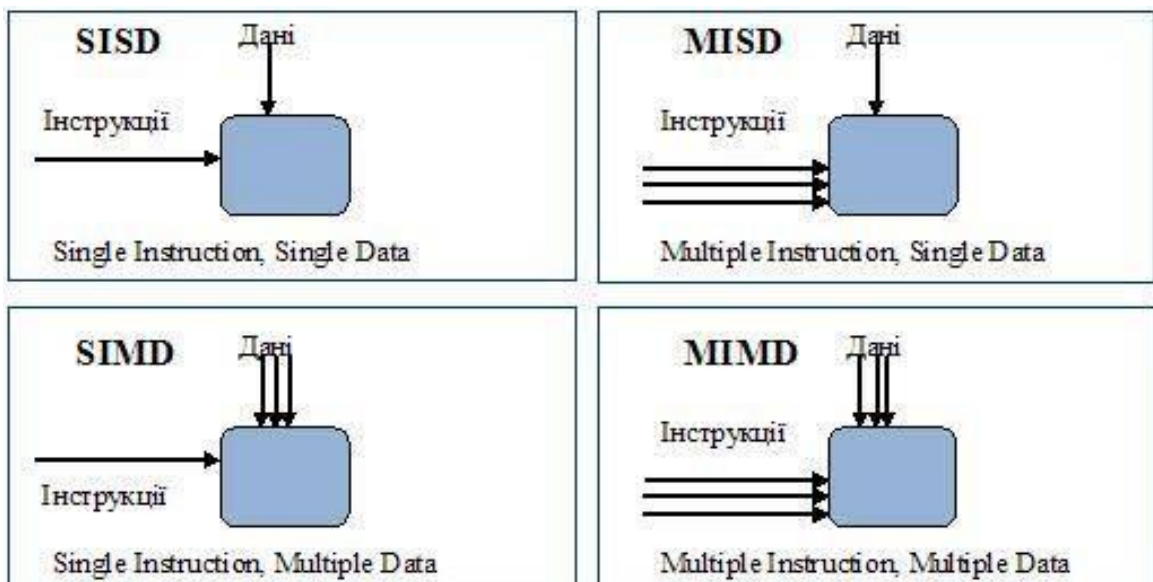


Рисунок 1.1 – Класифікація Флінна [3]

Основні складові мультикомп'ютерної системи вказані нижче [4].

1. Комп'ютери: це складова, без якої не можлива мультикомп'ютерність. Комп'ютери можуть бути звичайними персональними комп'ютерами, спеціалізованими пристроями або іншими пристроями з можливістю обробки даних.
2. Мережеве з'єднання забезпечує зв'язок між комп'ютерами в мультикомп'ютерній системі. Це може бути провідне чи бездротове з'єднання, яке залежить від конкретної реалізації.
3. Програмне забезпечення: це програмне забезпечення, яке управляє мультикомп'ютерною системою та забезпечує обмін даними між комп'ютерами. Це може бути програмне забезпечення розподіленої обробки даних, віртуальна машина, операційна система тощо.
4. Протоколи комунікації: це протоколи, які визначають правила передачі даних між комп'ютерами в мультикомп'ютерній системі. Це може бути TCP / IP, UDP, HTTP, FTP або інші.
5. Система керування ресурсами: це програмне забезпечення, яке забезпечує керування ресурсами мультикомп'ютерної системи, такими як пам'ять, процесорний час, мережеві ресурси та інше.

6. Застосунки: це програмне забезпечення, яке запускається на мультикомп'ютерній системі та забезпечує виконання завдань.

Існує кілька типів мультикомп'ютерних систем, залежно від того, як вони взаємодіють між собою [5 - 6].

1) кластер: група комп'ютерів, які працюють разом, щоб виконувати спільні завдання. Кожен комп'ютер у кластері виконує окрему частину розрахунків, а результати об'єднуються, щоб отримати кінцевий результат (рисунок 1.2);

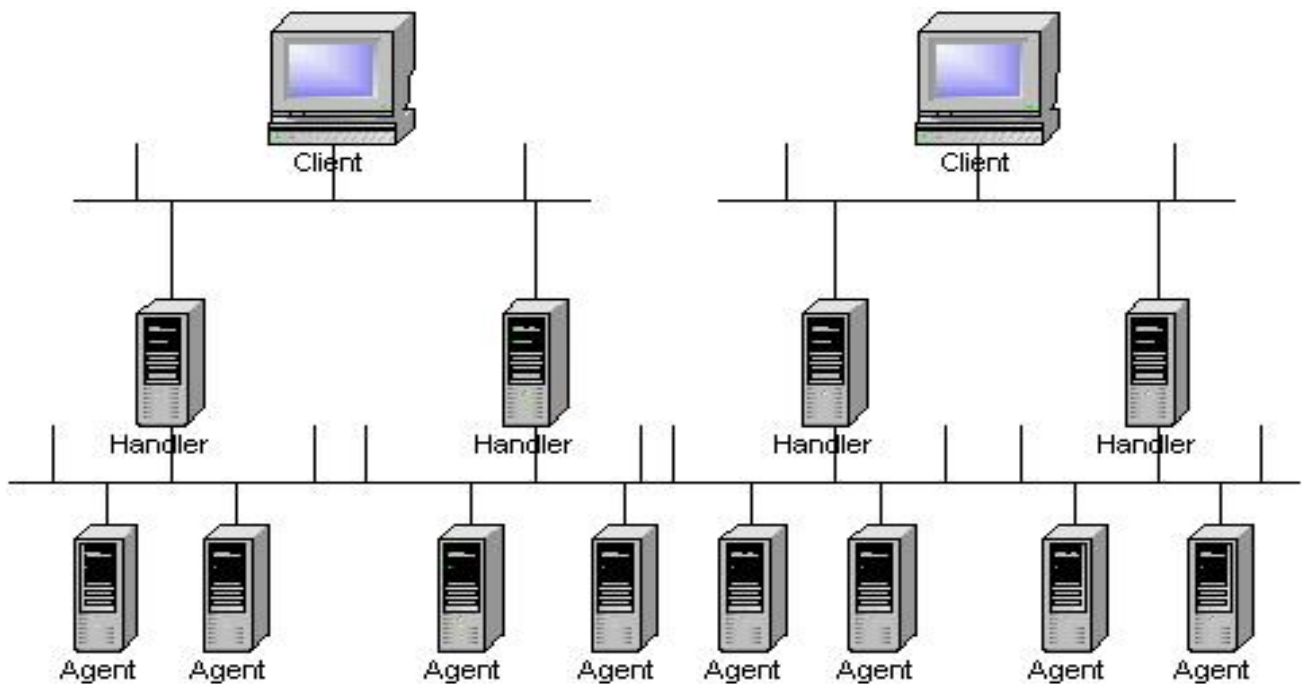


Рисунок 1.2 – Тип комп'ютерів Кластер [7]

2) сітка: група комп'ютерів, які спільно працюють над одним завданням, розподіляючи його на частини та оброблюючи його паралельно. Сітка може бути розподілена географічно та включати комп'ютери з різних мереж;

3) обчислювальна хмара: група віртуальних комп'ютерів, які надаються через Інтернет та використовуються для зберігання даних та виконання розрахунків (рисунок 1.3);

4) суперкомп'ютер: потужний комп'ютер, який складається з багатьох процесорів та використовується для обробки великих обсягів даних та виконання складних розрахунків (рисунок 1.4).

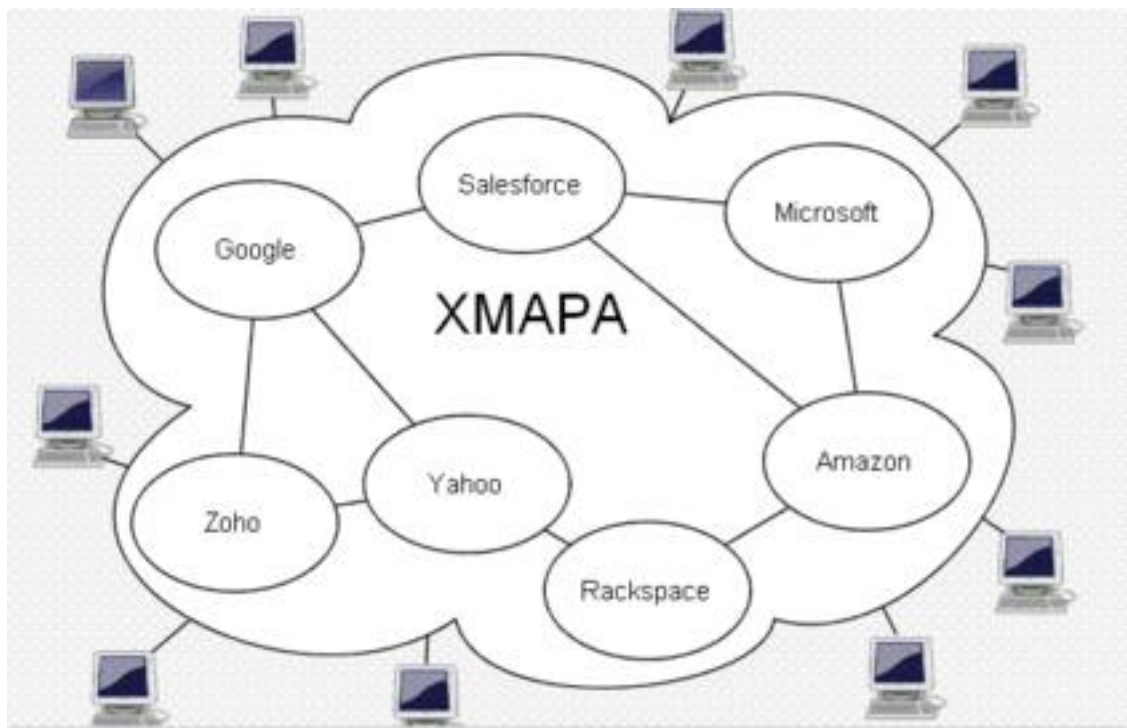


Рисунок 1.3 – Тип комп'ютерів Обчислювальна хмара [8]



Рисунок 1.4 – Тип комп'ютерів Суперкомп'ютер [9]

5) Клауд-платформа: група комп'ютерів та інфраструктури, що використовується для надання доступу до програмного забезпечення та послуг через Інтернет (рисунок 1.5).

Зм.	Арк.	№докум.	Підпис	Дата



Рисунок 1.5 – Тип комп'ютерів Клауд-платформа [10]

## 1.2 Актуальність розвитку мультикомп'ютерних систем

На сучасному етапі мультикомп'ютерні системи розвиваються у напрямку динамічних мережних утворень (аж до глобальних) з змінною конфігурацією та асинхронною роботою компонент, націленої на виконання задач зі слабозв'язаними частинами, наприклад переборного чи пошукового типу. Мультикомп'ютерна система забезпечує прозорий контрольований доступ користувачів до системи через InTerneT та прозоре підключення не використовуваних (idle) обчислювальних ресурсів до мультикомп'ютера [11].

Системи Globus і Legion [12] є відомими прикладами програмних засобів для побудови глобальних обчислювальних та інформаційних інфраструктур. Однією з перших задач, які вимагали значних обчислювальних ресурсів у глобальному масштабі, стала RSA Challenges - виклик, що полягав у розшифруванні кодів секретності. Поточна версія цієї задачі займається розшифруванням тексту, зашифрованого 64-бітним ключем, який має 264 комбінації. Більше 200 тисяч учасників брали участь у проекті, а досягнута продуктивність становила 75 млрд ключів в секунду.

Ще один масштабний проект – проект Оксфордського університету та фірми InTel зі створенню мультикомп'ютерної системи IPPPP (InTel PhilanTropic Peer-To-Peer Program) для пошуку нових ліків від лейкемії та ракових

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		10

захворювань. Суть задачі: перебрати мільйони хімічних сполук і оцінити їхню здатність протистояти захворюванню. Складність задачі - близько 24 млн. годин на один комп'ютер. Передбачається об'єднання мільйонів ПК в мережу, продуктивність якої оцінюється близько 50 Тфлопс.

Актуальність розвитку мультикомп'ютерних систем [13] продовжує зростати з прогресом технологій та збільшенням потреб у обчислювальних ресурсах. Ось кілька причин, чому мультикомп'ютерні системи залишаються актуальними.

Потужність обчислень: мультикомп'ютерні системи дозволяють поєднувати ресурси кількох комп'ютерів для виконання складних обчислень. Це особливо важливо у високопродуктивних обчислювальних задачах, таких як наукові дослідження, моделювання складних систем, аналіз даних тощо.

Розподілені системи: мультикомп'ютерні системи дозволяють створювати розподілені системи, в яких різні комп'ютери можуть виконувати окремі завдання та обмінюватися даними. Це дає змогу побудувати потужні та масштабовані системи, які можуть оптимально використовувати ресурси та забезпечувати високу доступність.

Висока швидкодія мережі: розвиток швидких мережних технологій, таких як 10-гігабітний ETherneT і високопропускність бездротових мереж, робить можливим ефективний обмін даними між комп'ютерами у мультикомп'ютерних системах. Це дозволяє досягти високої швидкодії передачі даних і зменшити затримки у комунікації.

Резервування та надійність: мультикомп'ютерні системи можуть бути сконфігуровані з резервними компонентами, такими як дубльовані сервери або мережеві шляхи, для забезпечення високої доступності та надійності системи. Це дозволяє уникнути відмов та зберегти продуктивність при можливих проблемах.

Розширюваність: мультикомп'ютерні системи можуть бути легко розширені шляхом додавання нових комп'ютерів або ресурсів до мережі. Це дозволяє адаптуватись до зростаючих потреб користувачів та забезпечувати масштабованість системи

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		11

### 1.3 Аналіз системного програмного забезпечення

Операційна система (ОС) [14] - це комплекс програмного забезпечення, яке управляє роботою комп'ютера або іншого пристрою. ОС є проміжним рівнем між користувачем та апаратним забезпеченням, що забезпечує управління та координацію різних компонентів системи.

ОС відповідає за керування ресурсами комп'ютера, такими як процесор, пам'ять, зовнішні пристрої введення-виведення, мережеві з'єднання тощо. Вона забезпечує запуск та завершення програм, керування файлами та директоріями, а також забезпечує безпеку та захист від зловмисного програмного забезпечення.

Основні класифікації операційних систем [15]:

За типом пристрою: операційні системи для настільних комп'ютерів, ноутбуків, серверів, мобільних пристроїв (смартфонів, планшетів), вбудованих систем (керування промисловими процесами, роботами тощо).

За методом роботи: однопрограмні та багатопрограмні операційні системи. Однопрограмні ОС дозволяють виконувати лише одну програму за раз. Багатопрограмні ОС дозволяють виконувати одночасно більше ніж одну програму.

За типом обробки даних: операційні системи з пакетною, партитивною та інтерактивною обробкою даних. Пакетна обробка даних означає, що програма виконується після того, як усі дані готові, партитивна обробка даних - поділ на кілька частин, а інтерактивна обробка даних дозволяє взаємодіяти з даними, використовуючи клавіатуру або мишу.

За типом інтерфейсу користувача: графічні та командні операційні системи. Графічні операційні системи дозволяють користувачам взаємодіяти з операційною системою за допомогою графічного інтерфейсу. Командні операційні системи використовують текстовий інтерфейс.

За типом рівня доступу: операційні системи з одно-, дво-, три-, або багаторівневим рівнем доступу. Ця класифікація визначає кількість рівнів доступу, які використовуються для захисту даних та ресурсів в операційній системі.

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		12

ОС виконує ряд функцій, які забезпечують правильну роботу комп'ютера, а також забезпечують взаємодію між користувачем та комп'ютером. Основні функції ОС включають:

- управління ресурсами;
- взаємодія з користувачем;
- управління файловою системою;
- управління процесами;
- забезпечення безпеки;
- управління мережею;

Складові операційної системи можна розділити на наступні категорії [16]:

1) ядро операційної системи - це основна частина операційної системи, яка забезпечує базові функції, такі як керування пам'яттю, взаємодія з апаратним забезпеченням, керування введенням/виведенням даних та інші;

2) драйвери пристроїв - це програми, які дозволяють операційній системі взаємодіяти з апаратним забезпеченням комп'ютера, таким як клавіатура, миша, дисковод, дисплей, звукова карта та інші.

3) командний інтерфейс - це інтерфейс, який дозволяє користувачам взаємодіяти з операційною системою через команди, які вводяться з клавіатури або миші. Командний інтерфейс може бути текстовим або графічним.

4) графічний інтерфейс - це інтерфейс, який дозволяє користувачам взаємодіяти з операційною системою через графічний інтерфейс, такий як вікна, кнопки, меню, іконки та інші.

5) системні утиліти - це додаткові програми, які допомагають управляти комп'ютером та операційною системою, такі як файловий менеджер, дискова утиліта, антивірусна програма та інші;

6) додатки - це програми, які працюють в операційній системі, такі як текстовий процесор, веб-браузер, медіаплеєр та інші.

ОС може бути розроблена для різних пристроїв, включаючи персональні комп'ютери, смартфони, планшети, сервери та інші. Деякі з найпопулярніших ОС включають Windows, macOS, Linux, Android та iOS [17].

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		13

Microsoft Windows займає лідируючу позицію на ринку операційних систем для настільних комп'ютерів з часткою близько 83,3%. MacOS від Apple Inc. є на другому місці з часткою 11,2%, а різновиди Linux займають третє місце з часткою 1,55%. У мобільному секторі (смартфонів і планшетів), Android від Google є домінуючою операційною системою з часткою ринку 87,5% та темпами зростання 10,3% щорічно, за якою сліdkує iOS Apple з часткою ринку 12,1% та зниженням частки ринку в 5,2% щорічно. Інші операційні системи складають всього 0,3% ринку. Різновиди Linux є домінуючими у секторах серверів та суперкомп'ютерів. Для багатьох додатків існують спеціалізовані класи операційних систем, такі як вбудовані системи та системи реального часу.

#### 1.4 Проміжне програмне забезпечення

Проміжне програмне забезпечення [18] (англ. middleware) - це програмне забезпечення, що забезпечує взаємодію між різними програмними компонентами або системами. Його можна описати як "проміжний шар" між програмними компонентами, що забезпечує їх взаємодію та координацію.

Проміжне програмне забезпечення може включати в себе різноманітні сервіси, такі як сервіси безпеки, сервіси транзакцій, сервіси кешування, сервіси маршрутизації, сервіси комунікації та інші. Воно забезпечує різні рівні абстракції, що дозволяють різним програмним компонентам взаємодіяти один з одним без прямої взаємодії.

Концепція проміжного програмного забезпечення (middleware) з'явилася в середині 1980-х років, коли розробники програмного забезпечення почали шукати способи полегшення інтеграції різних програмних компонентів та систем [19].

Перші приклади проміжного програмного забезпечення були створені для підтримки різноманітних мережевих протоколів та інтерфейсів взаємодії, таких як TCP/IP, CORBA, DCOM та інших. Ці технології дозволяли розробникам створювати складні програмні системи, що взаємодіють з різними протоколами та інтерфейсами взаємодії.

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		14

У наступні десятиліття розробники продовжували розробляти різноманітні інструменти та фреймворки для створення проміжного програмного забезпечення, яке забезпечувало більш складну функціональність та більш широкий спектр можливостей. До таких технологій відносяться, наприклад, мови програмування Java та C#, фреймворки Spring та .NET, а також сервіс-орієнтована архітектура (SOA) та інші.

Цей термін найчастіше застосовується до програмного забезпечення, яке забезпечує обмін та керування даними в розподілених програмах. У 2000 році на семінарі IETF проміжне програмне забезпечення було визначено як "набір шарів послуг, які знаходяться вище за транспортним рівнем (тобто по протоколу TCP/IP), але нижче за рівнем прикладного середовища" (тобто нижче за рівнем API-інтерфейсів додатків). В більш конкретному сенсі, проміжне програмне забезпечення можна описати як "з'єднання" (-) між клієнтом і сервером або між рівнозначними вузлами. Проміжне програмне забезпечення включає веб-сервери, сервери додатків, системи керування вмістом та схожі інструменти, які підтримують розробку та доставку додатків.

Проміжне програмне забезпечення використовується в багатьох різних галузях, включаючи веб-розробку, мобільну розробку, інтернет речей (IoT), обробку даних та інші. Це дозволяє програмістам зосередитися на розробці функціональності своїх програмних компонентів, не звертаючи увагу на деталі взаємодії між ними.

Звичайна програма проміжного програмного забезпечення складається з різних компонентів, які об'єднуються для створення потоку даних. Ці компоненти є основою архітектури проміжного ПЗ [20-21].

Консоль керування - це місце, де знаходяться правила програмного забезпечення, дії системи проміжного програмного забезпечення та конфігураційні налаштування.

Клієнтський інтерфейс - зовнішній компонент проміжного програмного забезпечення, який взаємодіє з програмою. Розробники використовують функції, надані клієнтським інтерфейсом, для виконання різних дій з програмою.

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		15

Внутрішній інтерфейс проміжного ПЗ - внутрішнє з'єднання, що об'єднує різні компоненти програмного забезпечення. Різні компоненти використовують цей внутрішній інтерфейс для спільної роботи та досягнення загальної мети.

Менеджер контрактів - менеджер контрактів визначає правила обміну даними в проміжному програмному забезпеченні. Він також гарантує, що різні програми дотримуються встановлених правил під час обміну даними.

Інтерфейс платформи - цей інтерфейс включає програмні компоненти, які взаємодіють з різними операційними системами. Він забезпечує сумісність проміжного програмного забезпечення з різними платформами.

Монітор виконання - монітор виконання постійно відстежує рух даних у системі. Його завдання полягає у виявленні та повідомленні про незвичайну поведінку, на яку інженерам потрібно реагувати.

Менеджер баз даних - не вся проміжна програмна система має такий компонент. Менеджер баз даних інтегрується з різними типами даних.

Менеджер сесій - цей компонент зберігає записи про дії з даними для звітності та забезпечує безперервний потік інформації.

Різні типи проміжного ПЗ [22]:

- 1) проміжне програмне забезпечення, орієнтоване на повідомлення;
- 2) проміжне програмне забезпечення API;
- 3) проміжне програмне забезпечення віддаленого виклику процедур (RPC);
- 4) проміжне програмне забезпечення транзакцій;
- 5) побототехнічне проміжне програмне забезпечення;
- 6) проміжне програмне забезпечення пристрою.

### 1.5 Поняття топології та її види

Топологія в мультикомп'ютерних системах визначає фізичну або логічну структуру підключення комп'ютерів та спосіб організації комунікаційного середовища між ними [23]. Основні типи топологій, які використовуються в мультикомп'ютерних системах, включають:

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		16

1) Зіркова топологія (Star Topology) (рисунок 1.6). Комп'ютери підключені до центрального комутатора або концентратора. Усі комунікації між комп'ютерами прямує через центральний вузол.

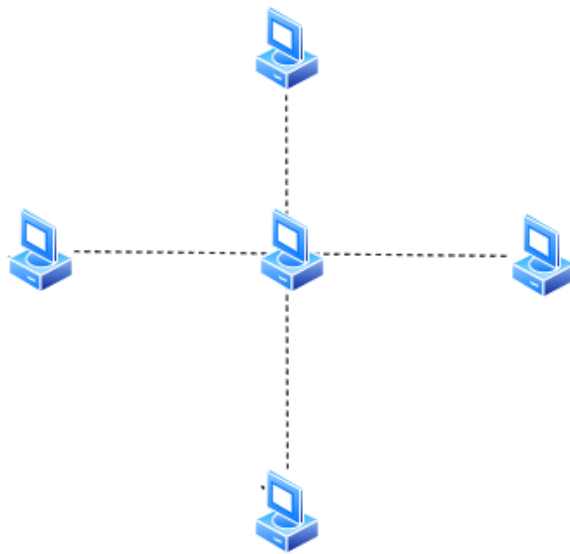


Рисунок 1.6 – Зіркова топологія

2) Лінійна топологія (Linear Topology) (рисунок 1.7). Комп'ютери підключені поспіль один до одного у вигляді лінії. Комунікація відбувається від одного комп'ютера до наступного вперед та назад.

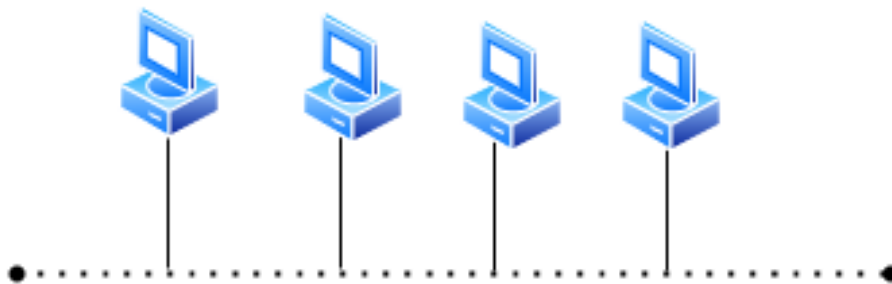


Рисунок 1.7 – Лінійна топологія

3) Кільцева топологія (Ring Topology) (рисунок 1.8). Комп'ютери підключені у формі кільця, де кожен комп'ютер має пряме з'єднання з двома сусідніми комп'ютерами. Дані передаються по колу в одному напрямку.

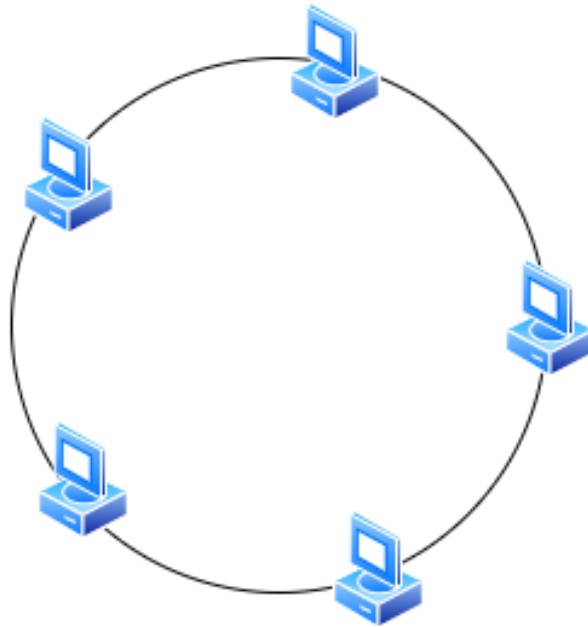


Рисунок 1.8 – Кільцева топологія

4) Шина (Bus Topology) (рисунок 1.9). Усі комп'ютери підключені до одного центрального кабелю (шини). Комунікація відбувається шляхом передачі сигналів по спільному кабелю.

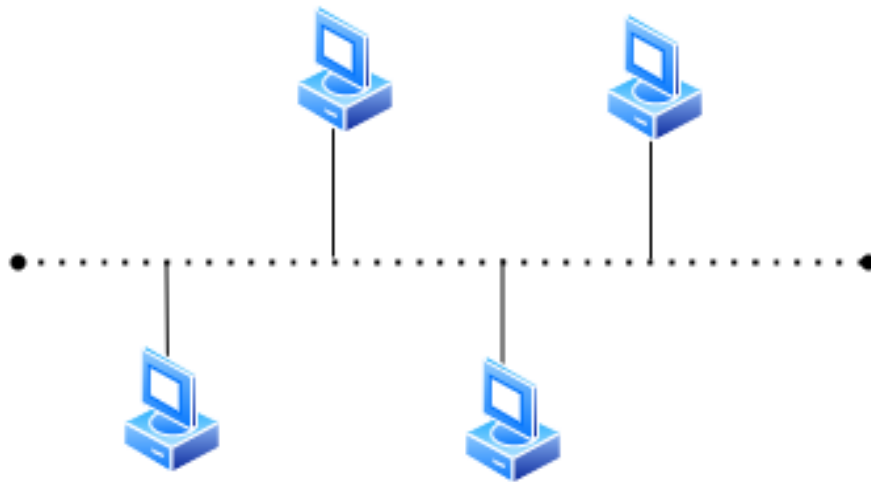


Рисунок 1.9 – Шина топологія

5) Меш (Mesh Topology) (рисунок 1.10). Кожен комп'ютер підключений безпосередньо до кожного іншого комп'ютера у системі. Це забезпечує багато можливих шляхів для комунікації між комп'ютерами.

Зм.	Арк.	№докум.	Підпис	Дата

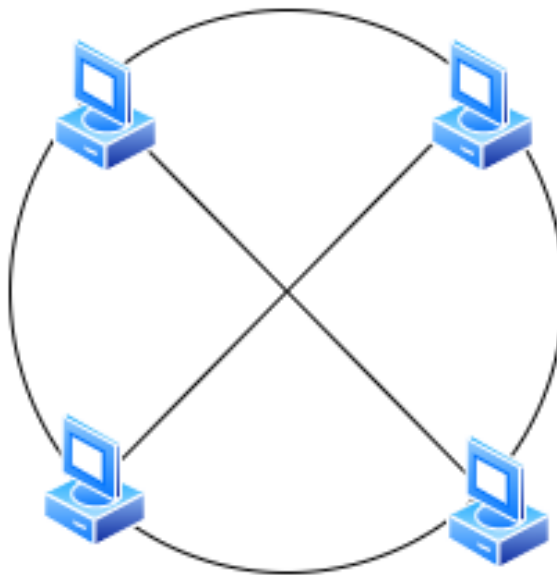


Рисунок 1.10 – Меш топологія

Топологія в мережах комп'ютерів відіграє важливу роль у визначенні способу підключення комп'ютерів та організації комунікаційного середовища. Основні ролі топології включають [24-25]:

- топологія визначає фізичний спосіб, яким комп'ютери підключені один до одного. Вона визначає, які кабелі, конектори та пристрої використовуються для забезпечення з'єднання між комп'ютерами;

- топологія визначає спосіб передачі даних між комп'ютерами. Вона визначає, як сигнали передаються від одного комп'ютера до іншого, як вони маршрутизуються та які правила комунікації застосовуються;

- топологія може мати вплив на надійність мережі. Деякі топології, такі як кільцева топологія, можуть забезпечити резервне з'єднання в разі відмови одного комп'ютера. Інші топології можуть бути менш стійкими до відмов;

- топологія також може впливати на масштабованість мережі. Деякі топології можуть бути легше розширювати та додавати нові комп'ютери, в той час як інші можуть бути складнішими для масштабування;

- різні топології можуть мати вплив на управління та підтримку мережі. Деякі топології можуть спрощувати конфігурацію та управління мережею, тоді як інші можуть потребувати більше уваги та налаштування.

Зм..	Арк.	№докум.	Підпис	Дата

Те, як мережа влаштована, може впливати на її функціональність, підключення та захист від простоїв. Питання «Що таке топологія мережі?» можна відповісти поясненням двох категорій топології мережі [26]:

1. Фізична – фізична топологія мережі стосується фактичних з'єднань (проводів, кабелів тощо), як влаштована мережа. Завдання з налаштування, обслуговування та ініціалізації вимагають уявлення про фізичну мережу.

2. Логічна – логічна топологія мережі – це уявлення вищого рівня про те, як налаштована мережа, включаючи те, які вузли з'єднуються один з одним і якими способами, а також як дані передаються через мережу. Логічна топологія мережі включає будь-які віртуальні та хмарні ресурси.

Відображення топології важливе не лише для керування окремою мережею. Це також ключовий аспект основних обов'язків постачальників керованих послуг (MSP) для сотень або навіть тисяч різних клієнтів у кількох мережах.

## 1.6 Висновки

У першому розділі ми почали заглиблюватись в роботу обчислювальних систем, познайомились з класифікацією Флінна, який поділяє паралельні EOM на чотири класифікації: конвеєрна і векторна обробка, машини типу SIMD, машини типу MIMD, багатопроцесорні машини з SIMD процесорами. Паралельна обробка дозволяє досягти високої продуктивності обчислювальних систем шляхом використання високошвидкісних елементів і паралельного виконання багатьох операцій.

Розібрались з складовими мультикомп'ютерної системи. Комп'ютери виконують обробку даних, мережеве з'єднання забезпечує зв'язок між комп'ютерами, програмне забезпечення управляє системою і обмінюється даними, протоколи комунікації визначають правила передачі даних, система керування ресурсами керує ресурсами системи, а застосунки виконують завдання.

Ознайомились з операційними системами, тому що вони є важливим комплексом програмного забезпечення, які управляють роботою комп'ютера або

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		20

пристрою. Вони виступають посередником між користувачем та апаратним забезпеченням, забезпечуючи керування та координацію різних компонентів системи.

Описали не менш важливі топології в мультикомп'ютерних системах, які визначають структуру підключення комп'ютерів та спосіб організації комунікаційного середовища.

Також провели аналіз вже готових відомих мультикомп'ютерних систем, та зрозуміли їхню роботу та призначення.

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		21

## 2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ

### 2.1 Вибір топології “подвійне кільце”

Системи з розподіленою пам'яттю (мультикомп'ютери), у них кожен комп'ютер має свою локальну оперативну пам'ять, а в інших процесорів доступ до цієї пам'яті відсутній. Щоб працювати на комп'ютері з розподіленою пам'яттю необхідно створювати копії початкових (вихідних) даних на кожному процесорі.

Кожен вузол має один чи декілька процесорів і свою власну локальну пам'ять. Розподіленість пам'яті означає, що кожен процесор має безпосередній доступ тільки до локальної пам'яті свого вузлу [27].

Топологія мережі - це спосіб, яким взаємодіють пристрої та мережні канали в комп'ютерній мережі. Це може бути фізична топологія, яка описує фактичне розташування пристроїв та кабелів в мережі, або логічна топологія, відображає шляхи передачі даних в мережі. розташування пристроїв та кабелів в мережі, або логічна топологія, яка відображає шляхи передачі даних в мережі.

Топологія подвійних кілець (рисунок 2.1) – впливає з топології кілець (рисунок 2.2).

Кільцева топологія - це тип комп'ютерної мережі, в якій пристрої підключені до одного логічного кільця. Кожен пристрій підключений до двох сусідніх пристроїв, створюючи закритий кільцевий шлях для передачі даних. Кільцева топологія може бути реалізована як фізична топологія з допомогою коаксіального кабелю або оптичного волокна, або як логічна топологія, в якій даний кільцевий шлях створюється програмним забезпеченням.

У кільцевій топології передача даних відбувається за допомогою технології "токену", яка передається вздовж кільця від одного пристрою до наступного. Коли пристрій бажає передати дані, він очікує, доки токен досягне його, після чого він додає свої дані до токена і передає його далі по кільцю. Коли дані досягають призначеного пристрою, вони вилучаються з токена, і токен продовжує свій шлях по кільцю. Перший вузол з'єднаний з останнім вузлом, щоб з'єднати цикл.

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		22

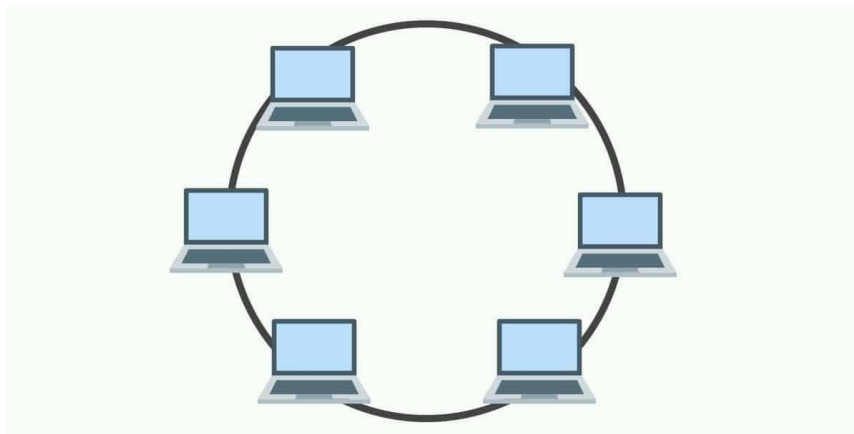


Рисунок 2.1 — Мережа з топологією «кільце»

Кільцева топологія має декілька переваг, зокрема, вона є дуже надійною, оскільки якщо один з пристроїв відмовляє, інші пристрої можуть продовжувати передачу даних по кільцю. Крім того, кільцева топологія має досить просту структуру, що дозволяє легко додавати нові пристрої до мережі. Однак кільцева топологія також має свої недоліки, зокрема, вона може бути дуже повільною в порівнянні з іншими типами топологій, оскільки передача даних відбувається послідовно через кожен пристрій на кільці.

Для керування мережею та інших пристроїв в межах цієї топології вибирається один вузол [28]. Кільцеві топології можуть бути напівдуплексними (двонаправлену передачу даних), але також можуть бути зроблені дуплексними. Щоб зробити кільцеву топологію повнодуплексною, потрібно мати два з'єднання між мережевими вузлами, створивши тим самим подвійну кільцеву топологію.

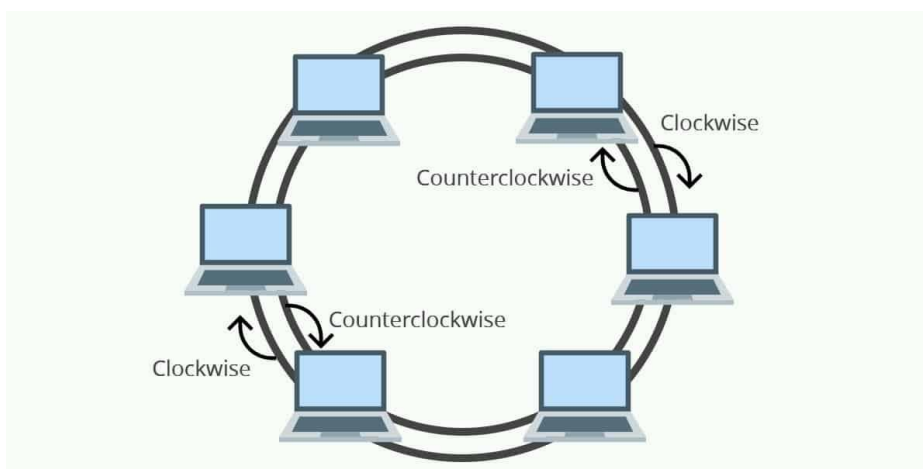


Рисунок 2.2 — Мережа з топологією «подвійне кільце»

Зм.	Арк.	№докум.	Підпис	Дата

Як вже згадувалося раніше, якщо кільцеві топології налаштовані на двонаправлене з'єднання, то вони називаються топологіями подвійних кілець. У таких топологіях кожен вузол має два з'єднання - по одному в кожному напрямку, що дозволяє передавати дані як за годинниковою стрілкою, так і проти неї.

Проведемо характеристики саме цієї топології:

- діаметр - залежить від кількості вузлів у мережі та довжини лінії зв'язку між вузлами. Оскільки в подвійному кільці кожен вузол має два з'єднання, то діаметр становить два рази довжину між вузлами, які мають максимальну відстань між собою в кільці:

$$D = 2 * L * (N/2), \quad (2.1)$$

де  $D$  – діаметр топології;

$L$  – довжина лінії зв'язку між вузлами;

$N$  – кількість вузлів у мережі;

- зв'язність:

$$Z = 2 * n - 2, \quad (2.2)$$

де  $n$  – кількість вузлів в кожному кільці.

Формула надсилання даних по топології подвійне кільце залежить від того, чи використовується однопотоковий (half-duplex) чи двопотоковий (full-duplex) режим передачі. У випадку однопотокового режиму надсилання даних відбувається послідовно, по колу, від одного вузла до наступного, а випадку двопотокового режиму - одночасно в обох напрямках:

- Формула для однопотокового режиму надсилання даних по топології подвійне кільце може бути наступною:

$$T = (N * L) / R, \quad (2.3)$$

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		24

де  $T$  - час передачі повідомлення;

$N$  - кількість вузлів в мережі;

$L$  - довжина повідомлення,

$R$  - швидкість передачі даних.

- Формула для двопотокового режиму може бути наступною:

$$T = (N * L) / (2 * R), \quad (2.4)$$

де  $T$ ,  $N$  і  $L$  мають ті ж самі значення, але  $R$  враховується як швидкість передачі даних в одному напрямку, оскільки в обох напрямках дані передаються одночасно.

## 2.2 Види архітектур мультикомп'ютерної системи

### 2.2.1 Векторно-конвеєрні комп'ютери

У середині 1960-х років компанія CDC (заснована Сеймуром Креєм) створила високопродуктивний комп'ютер з новою векторною архітектурою, що призвело до з'явлення терміну "суперкомп'ютер". Головною ідеєю цієї архітектури було розпаралелювання обробки даних, що дозволяло одночасно застосовувати одну і ту ж операцію до масиву значень (вектора). Ця ідея була успішно використана на різних рівнях функціонування комп'ютера. Одним з класичних прикладів суперкомп'ютерів є Cray-1, який був представлений в 1976 році і відрізнявся векторно-конвеєрною архітектурою, що дозволяла ефективно обробляти великі масиви даних [29]. Основні особливості архітектури цього класу комп'ютерів наступні:

- конвеєризація виконання команд;
- незалежність функціональних пристроїв, тобто декілька операцій можуть виконуватись одночасно;
- векторна обробка (набір даних обробляється однією командою);

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		25



Наприклад, якщо половина програми складається з послідовних обчислень, які неможливо векторизувати, то навіть якщо друга половина програми виконується ідеально швидко завдяки векторизації, загальне прискорення програми не буде більшим за два рази.

Сучасні комп'ютерні процесори використовують технологію конвеєрної обробки, що дозволяє обробляти кілька команд одночасно. Обробка команд розділяється на п'ять основних етапів, відомих як мікрокоманди: вибірка команди, розшифровка команди, вибірка операндів, виконання команди та збереження результатів. Кожен етап виконується тільки один раз і в одному порядку, послідовно один за одним. Ця технологія дозволяє ефективно виконувати команди, забезпечуючи безперервну роботу конвеєра. Кожна частина конвеєра називається шаблоном, а загальна кількість етапів визначає довжину конвеєра [31].

### 2.2.2 Обчислювальні системи з розподіленою пам'яттю

Масивно-паралельні комп'ютери складаються [32] з обчислювальних вузлів, які спілкуються між собою за допомогою комунікаційного середовища. Кожен вузол має свою локальну пам'ять та один чи кілька процесорів. Така розподіленість пам'яті означає, що процесор має доступ лише до локальної пам'яті свого вузла. Для доступу до пам'яті інших вузлів використовують спеціально спроектоване для даної системи комунікаційне середовище або стандартне.

Архітектура масивно-паралельних комп'ютерів має декілька переваг, зокрема, низьку вартість відносно продуктивності та можливість легкого розширення за рахунок додавання нових процесорів. Різноманітність цих комп'ютерів полягає в особливостях організації комунікаційного середовища. Існують різні топології, такі як прямокутна решітка, ієрархічна система комутаторів, або трьохвимірний торус.

У мультикомп'ютері не існує загальної пам'яті, тому процесори не можуть просто взаємодіяти між собою. Замість цього використовується мережа міжз'єднань для передачі повідомлень між процесорами. Також в системі є певна

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		27

програмна структура, яка забезпечує відсутність загального віртуального адресного простору для всіх процесів [33].

Кожен вузол має свої процесори та локальну пам'ять, та може взаємодіяти з іншими вузлами за допомогою спеціального комунікаційного середовища або стандартних протоколів.

На рисунку 2.4 зображено приблизну архітектуру багатопроцесорних систем з розподіленою пам'яттю.

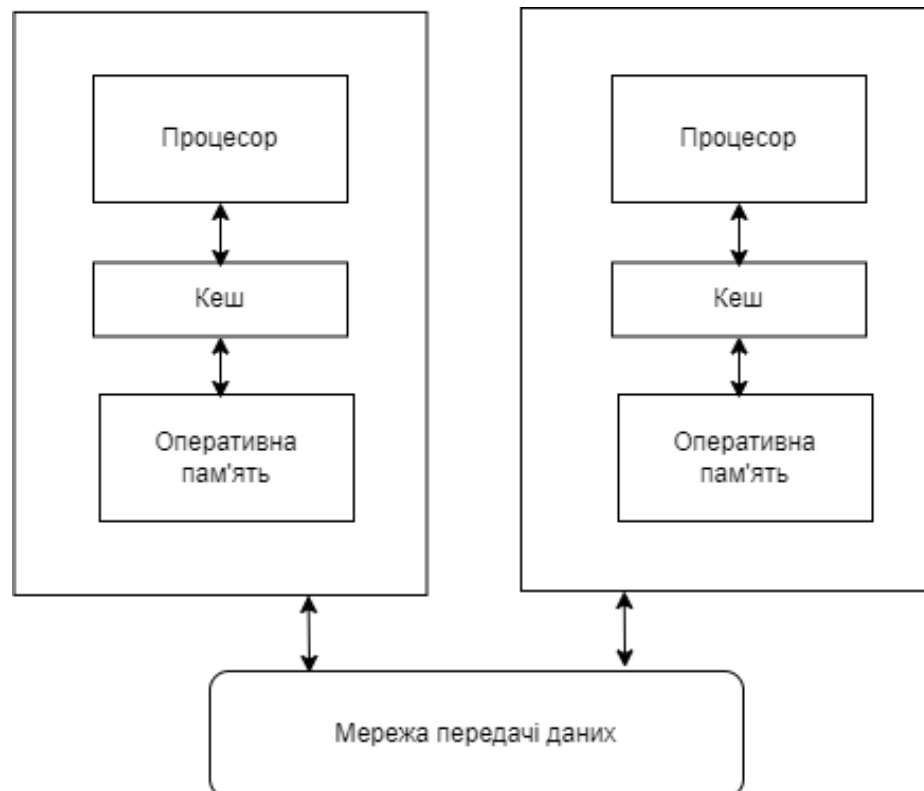


Рисунок 2.4 — Архітектура багатопроцесорних систем з розподіленою пам'яттю

Така схема називається мультикомп'ютером, або системою з розподіленою пам'яттю (рисунок 2.5 а).

Якщо, наприклад, процесор звертається до інформації, яка зберігається в пам'яті іншого процесора, то він не може просто отримати цю інформацію, як у мультипроцесорі. Наприклад, якщо процесор знаходиться у лівому кутку рису. 2.5 б (привласним цьому процесору номер 0) виявить, що частина його об'єкта буде виявлена в дальній секції, що потрапляє до наступного процесора (нехай це буде процесор 1), він може просто відстежувати інформацію з пам'яті, щоб отримати

зображення відтвореного хвоста. Однак якщо те саме виявить процесор 0 на рисунку 2.5 б.



Рисунок 2.5 – Умовне представлення мультикомп'ютера [34]

- а) – 16 процесорів, кожний зі своєю власною пам'яттю
- б) – зображення розподілене між 16 ділянками пам'яті

### 2.2.3 Кластерні системи

Кластер - це група незалежних комп'ютерів, яка може бути доступна як єдина система. Кластери використовуються для вирішення одного або кількох завдань. Існують три основних типи кластерів: кластери високої готовності (high-availability clusters), кластери балансування навантаження (load-balancing clusters) і обчислювальні кластери (compute clusters). Однією з переваг кластерів є те, що вони не потребують додаткового периферійного обладнання, що значно знижує вартість системи [35].

На рисунку 2.6 можна побачити архітектуру кластерних обчислень.

У простій формі кластерні обчислення можна описати як систему, що складається з двох або більше комп'ютерів або систем, які називаються вузлами. Ці вузли працюють спільно для виконання програм та інших завдань. Користувачі, що використовують ці вузли, мають враження, що вони працюють з

однією системою, що створює ілюзію єдиного ресурсу, відомого як віртуальні машини. Це поняття відоме як прозорість системи. Інші важливі характеристики для таких платформ включають надійність, балансування навантаження і продуктивність.

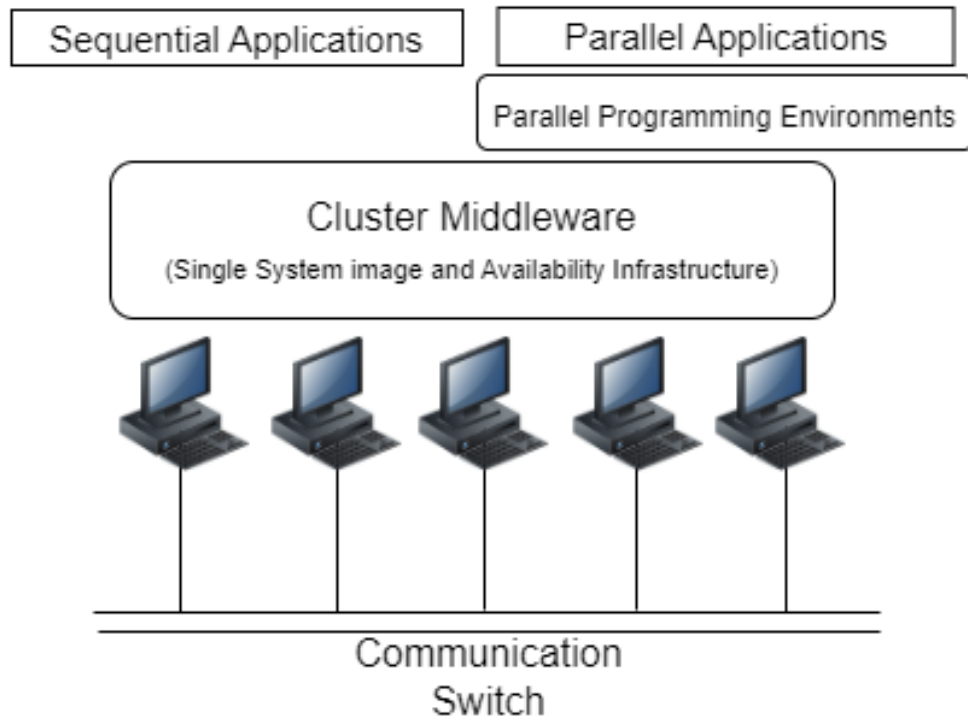


Рисунок 2.6 – Архітектура кластерних обчислень

Типи кластерів:

1. Кластери високої доступності та відмовостійкі кластери (рисунок 2.7) - це моделі кластерів, які забезпечують неперервну доступність служб та ресурсів за допомогою системи неявного резервування. Основна ідея цих кластерів полягає в тому, що у разі відмови одного вузла програми та служби можуть бути доступні на інших вузлах. Ці типи кластерів є основою для критичних місій, поштових систем, файлових систем і серверів додатків.

2. Кластери балансування навантаження (рисунок 2.8) – цей кластер розподіляє весь вхідний трафік/запити на ресурси від вузлів, які запускають ті самі програми та машини. У цій моделі кластера всі вузли відповідають за відстеження замовлень, і якщо вузол виходить з ладу, тоді запити розподіляються

між усіма доступними вузлами. Зазвичай таке рішення використовується на фермах веб-серверів.

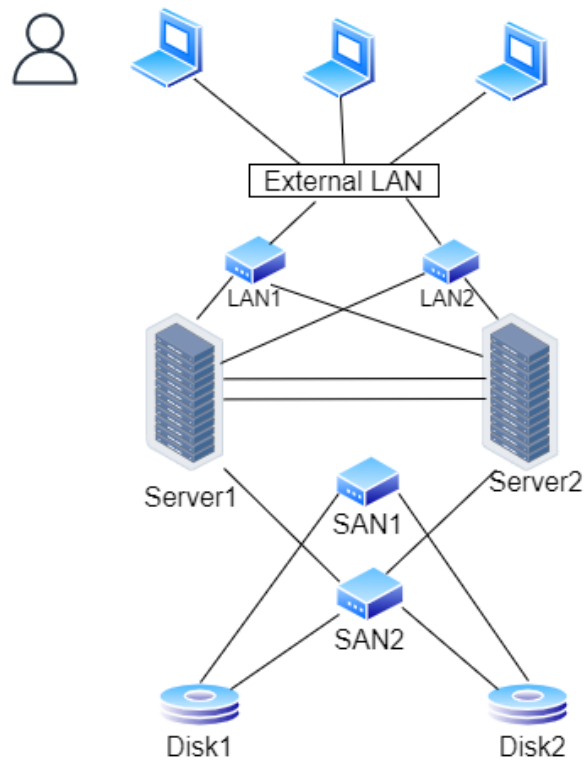


Рисунок 2.7 – Архітектура кластерів високої доступності та відмовостійкі

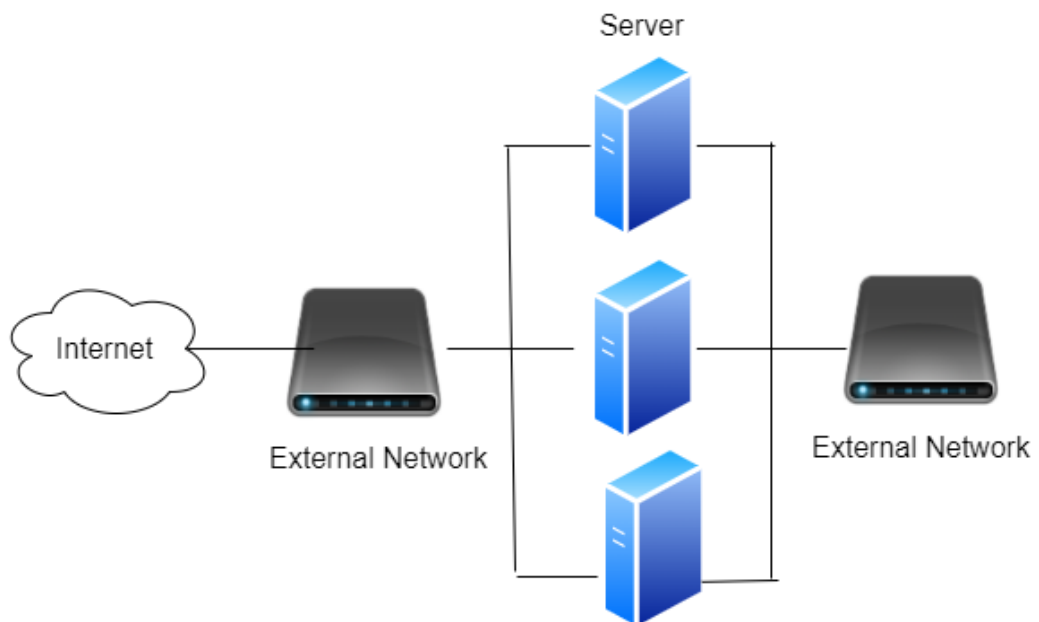


Рисунок 2.8 – Архітектура кластерів балансування навантаження

3. Кластери високої доступності та балансування навантаження (рисунок 2.9) – ця модель кластера поєднує в собі функції обох кластерів, що забезпечує підвищену доступність і масштабованість послуг і ресурсів. Цей тип кластерів зазвичай використовується для серверів електронної пошти, Інтернету, новин і FTP.

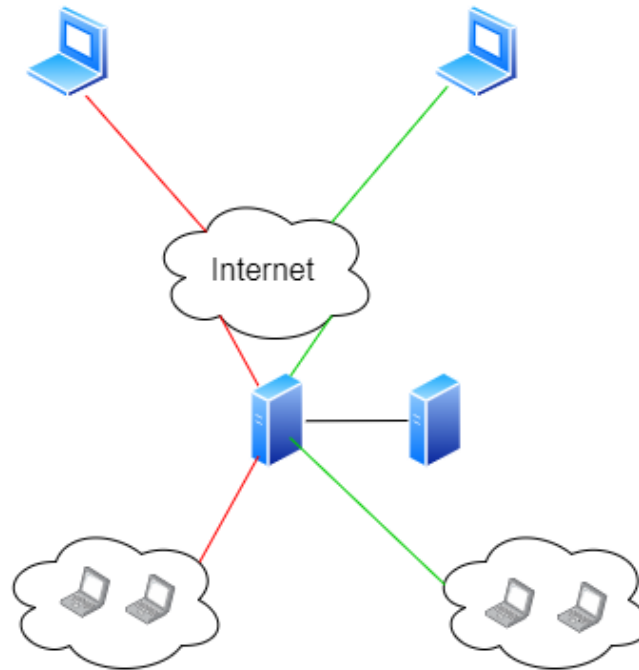


Рисунок 2.9 – Архітектура кластерів високої доступності та балансування навантаження

4. Кластери з розподіленою та паралельною обробкою (рисунок 2.10) – ця модель кластера підвищує доступність і продуктивність програм, які мають великі обчислювальні завдання. Велике обчислювальне завдання ділиться на менші завдання та розподіляється між станціями. Такі кластери зазвичай використовуються для наукових обчислень або фінансового аналізу, які потребують високої потужності обробки.

Кластери або комбінації кластерів використовуються, коли вміст має вирішальне значення, а служби повинні бути доступними. Постачальники послуг Інтернету або Інтернет-провайдери та сайти електронної комерції вимагають високої доступності та балансування навантаження в масштабований спосіб.

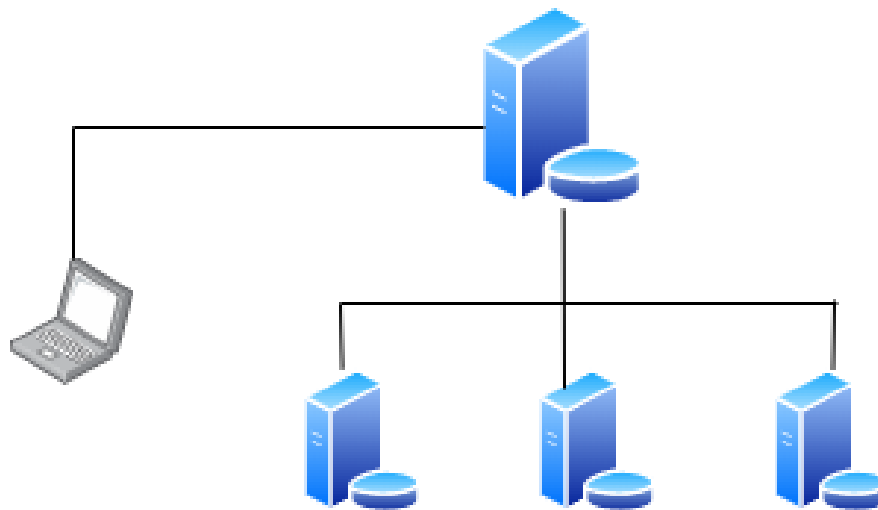


Рисунок 2.10 – Архітектура кластерів з розподіленою та паралельною обробкою

Паралельні кластери широко використовуються в кіноіндустрії, оскільки їм потрібна високоякісна графіка та анімація. Якщо говорити про кластери Беовульфа, то вони переважно використовуються в науці, техніці та фінансах для виконання різноманітних критичних проектів. Дослідники, організації та підприємства використовують кластери, щоб вимагати покращеної масштабованості, управління ресурсами, доступності та обробки за доступними цінами [36-37].

Кластерні обчислення пропонують широкий спектр переваг. Деякі з них включають наступне.

**Економічна ефективність.** У порівнянні з мейнфреймами кластерні обчислення вважаються набагато більш економічно ефективними. Ці обчислювальні системи пропонують підвищену продуктивність у порівнянні з комп'ютерними пристроями мейнфрейму.

**Швидкість обробки.** Швидкість обробки кластерних обчислень виправдана швидкістю обробки мейнфреймів та інших суперкомп'ютерів у світі.

**Розширюваність – масштабованість і розширюваність** є ще одним набором переваг, які пропонують кластерні обчислення. Кластерні обчислення представляють можливість додавання будь-якої кількості додаткових ресурсів і систем до існуючої обчислювальної мережі.

Підвищена доступність ресурсів – Доступність відіграє життєво важливу роль у кластерних обчислювальних системах. Збій будь-якого підключеного активного вузла можна легко передати іншим активним вузлам на сервері, забезпечуючи високу доступність.

Покращена гнучкість – у кластерних обчисленнях чудові специфікації можна оновити та розширити шляхом додавання нових вузлів до існуючого сервера.

#### 2.2.4 Вибір архітектури мультикомп'ютерної системи

Більша частина теперішніх «серйозних» комп'ютерів реалізовані в класі MIMD-архітектур – основним і найактуальнішим підкласом якої є MPP система [38].

MPP системи використовуються в різних галузях науки та техніки для виконання складних обчислень, обробки великої кількості транзакцій на секунду, управління великими базами даних тощо. Спочатку це були суперкомп'ютери, які використовувалися переважно для наукових розрахунків, але зараз багато таких систем знаходять своє застосування у комерції. В цілому, можна сказати, що MPP мультикомп'ютери витіснили SIMD-машини, векторні суперкомп'ютери та матричні процесори.

У більшості комп'ютерів з архітектурою MPP (масивно-паралельна обробка даних) використовуються стандартні процесори, такі як Intel Pentium, Sun UltraSPARC, IBM RS/6000 і DEC Alpha. Однак, мультикомп'ютери відрізняються від звичайних комп'ютерів завдяки наявності високопродуктивної комунікаційної мережі, яка забезпечує передачу повідомлень з низьким часом затримки і великою пропускну здатністю. Обидві характеристики, а саме час затримки і пропускну здатність, мають велике значення, оскільки повідомлення зазвичай є малими за розміром (менше 256 байтів), але великий обсяг трафіку становлять великі повідомлення (більше 8 КБ) [39-40].

Дуже значима та захоплива характеристикою MPP систем є обробка величезних обсягів введення/виведення даних. У таких системах, зазвичай, ми

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		34

маємо справу з обробкою великих масивів даних, іноді навіть в розмірі кількох терабайтів. Ці дані потрібно розподілити по численних дисків і передавати між пристроями системи з високою швидкістю.

Основним критерієм, який використовується для визначення обчислювальної системи як архітектури з масовою паралельною обробкою (MPP - Massively Parallel Processing), є кількість процесорів  $n$ . Не існує жорсткої межі, але зазвичай, якщо  $n$  становить 128, то цю систему вважають MPP, в той час як при  $n$  32 її ще не розглядають як таку.

Основні характеристики, що відносять обчислювальну систему до класу MPP, можна узагальнити таким чином:

- використання стандартних мікропроцесорів;
- фізично розподілена пам'ять;
- наявність мережі з'єднань з високою пропускнуою здатністю і малими затримками;
- інтеграція добре масштабованої системи, здатної працювати зі змінним числом процесорів (від декількох до тисяч);
- асинхронна архітектура MIMD з можливістю передачі повідомлень між процесорами;
- програма представляє собою набір процесів, кожен з яких має власний адресний простір.

В приблизному розумінні, можна стверджувати, що на центральному процесорі виконується ядро операційної системи (як планувальник завдань), тоді як на підлеглих процесорах виконуються додатки. Підлеглість між процесорами може бути реалізована на апаратному рівні або на програмному рівні.

Узагальнена структура MPP-системи показана на рисуюнок 2.11.

Основні мотивації для розвитку систем масової паралельної обробки можна сформулювати таким чином [41] :

- перш за все, необхідність створення операційних систем з високою продуктивністю, яка забезпечує швидку обробку завдань.
- Другим чинником є зниження вартості операційної системи.

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		35

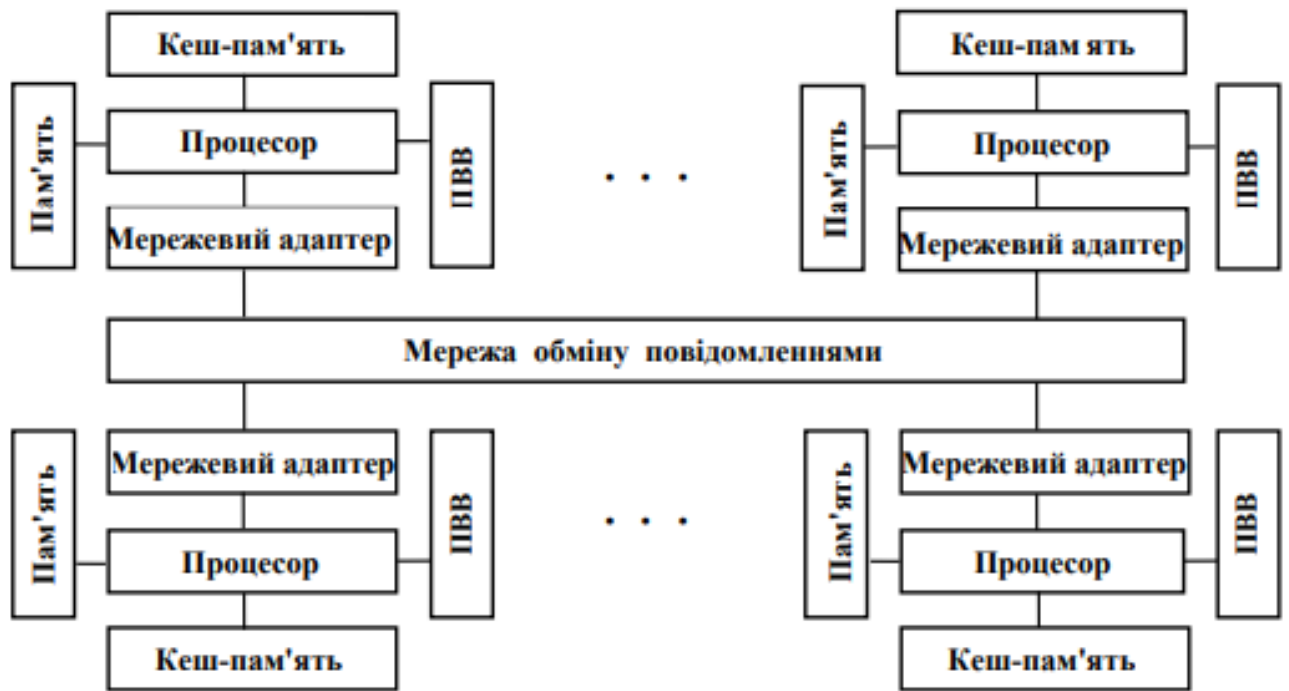


Рисунок 2.11 – Структура MPP-системи

Одним із визначальних аспектів MPP-систем є наявність одного центрального керуючого пристрою (процесора), який розподіляє завдання серед підлеглих пристроїв. Схема взаємодії між центральним пристроєм та підлеглими пристроями може бути узагальнено таким чином [42]:

- центральний управляючий пристрій формує чергу завдань, які мають різні рівні пріоритету;
- завдання передаються підлеглим пристроям по мірі їх вільності. Підлегли пристрої повідомляють центральний процесор про прогрес виконання завдань, включаючи повідомлення про завершення або потребу в додаткових ресурсах.
- центральний пристрій має засоби контролю за роботою підлеглих процесорів, зокрема для виявлення аварійних ситуацій, переривання виконання завдань у випадку появи більш пріоритетного завдання тощо.

MPP-системи, завдяки своїй масштабованості, вже сьогодні займають лідируючі позиції щодо досягнутої продуктивності. Однак, паралелізація в MPP-системах представляє собою складну задачу. Ефективність паралелізації значною мірою залежить від конкретних архітектурних особливостей MPP-системи. Для

синхронізації паралельно виконуваних процесів необхідний обмін повідомленнями, які повинні бути передані від будь-якого вузла системи до будь-якого іншого вузла. Час передачі інформації між вузлами залежить від початкової затримки та швидкості передачі. Продуктивність процесорів набагато перевищує пропускну здатність каналів зв'язку, що робить інфраструктуру каналів зв'язку в МРР-системах найбільш проблематичною.

Слабким пунктом МРР-системи завжди був і залишається центральний управляючий пристрій (ЦУП) - коли він виходить з ладу, вся система стає непрацездатною. Для підвищення надійності ЦУП використовують спрощення або дублювання його апаратних компонентів. Сфера застосування операційних систем з масовим паралелізмом постійно розширюється. Різні системи цього класу успішно експлуатуються у провідних суперкомп'ютерних центрах по всьому світу.

### 2.3 Вибір та опис апаратного забезпечення

Для реалізації мультикомп'ютерної системи згідно топології подвійне кільце знадобиться наступне апаратне забезпечення:

1. Два комп'ютери (ноутбуки).
2. Два комутатори з підтримкою кільцевої топології.
3. Два маршрутизатори для забезпечення зв'язку між різними сегментами кільцевої топології.
4. Два мережевих адаптери на кожному комп'ютері для підключення до обох кілець і забезпечення надійності.
5. Два кабелі на кожен комп'ютер для підключення до обох кілець.
6. Додаткове програмне забезпечення для забезпечення безперервності роботи і маршрутизації даних через кільце.

#### 2.3.1 Ноутбуки для мультикомп'ютерної системи

Для коректної роботи системи за відповідними параметрами було обрано 2

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		37

однакових ноутбуків (рисунок 2.12), які мають наступні характеристики:

- процесор Восьмиядерний Intel Core i9-11900H (2.5 — 4.9 ГГц);
- 16 Гб оперативної пам'яті;
- твердотілий накопичувач на 1 Тб;
- підтримка Wi-Fi 6E та Bluetooth 5.0;
- порт Thunderbolt 4 (USB-C), порт USB 3.2 Gen 2 Type-C, порт HDMI 2.1, порт LAN та пристрій для читання карток SD.



Рисунок 2.12 – Приклад ноутбука для мультикомп'ютерної системи [43]

### 2.3.2 Комутатор з підтримкою кільцевої топології

Комутатор DES-1210-52/ME (рисунок 2.13) може бути використаний як частина мультикомп'ютерної системи. Завдяки своїм характеристикам, він може забезпечити необхідні з'єднання і комутацію між комп'ютерами у мережі. За допомогою 48 портів 10/100Base-TX і 4 комбо-портів 1000Base-T/SFP, комутатор дозволяє підключити до 52 пристроїв до мережі. Це забезпечує потужність передачі даних до 4 Гбіт/с, що є важливим для високопродуктивних мультикомп'ютерних систем. Крім того, комутатор має вбудований захист від статичної електрики, що забезпечує стабільність роботи системи при перепадах напруги. Також він має набір функцій безпеки і аутентифікації, що допомагає

захистити мережу від потенційних загроз. Тому комутатор DES-1210-52/ME може бути використаний для побудови ефективної мультикомп'ютерної системи з високою пропускнуою здатністю і надійністю.



Рисунок 2.13 – Приклад комутатора для мультикомп'ютерної системи [44]

### 2.3.3 Маршрутизатори для забезпечення зв'язку між різними сегментами кільцевої топології

Маршрутизатори - це мережеві пристрої, що виконують функцію маршрутизації даних у комп'ютерних мережах. Вони використовуються для передачі пакетів даних між різними мережевими сегментами, що можуть мати різні адресні простори (рисунок 2.14).

Маршрутизатори приймають пакети даних з одного інтерфейсу, визначають найкращий шлях для доставки цих пакетів до їх призначення і пересилають їх через відповідні вихідні інтерфейси. Вони дозволяють мережам спілкуватися між собою та зовнішнім мережевим середовищем, роблять можливим підключення до Інтернету та надають функції мережевої безпеки, які контролюють доступ та фільтрують трафік.

Маршрутизатори можуть бути фізичними пристроями або програмними рішеннями, які працюють на спеціалізованому апаратному обладнанні або загального призначення комп'ютерах.



Рисунок 2.14 – Приклад маршрутизатора [45]



Рисунок 2.15 – Маршрутизатор Cisco RV134W WIRELESS-N VPN (RV134W-E-K9-G5) [46]

Для реалізації потрібної конфігурації мультикомп'ютерної системи було обрано 2 однакових маршрутизатори Cisco RV134W WIRELESS-N VPN (RV134W-E-K9-G5) (рисунок 2.15), що мають наступні характеристики:

- WAN-порт: ETherneT; USB 4G; USB 3G;
- Wi-Fi: 2.4 ГГц; 5 ГГц;

Зм.	Арк.	№докум.	Підпис	Дата

- бездротові можливості: IEEE 802.11b/g/n/ac;
- пропускна здатність IPsec VPN (3DES, AES) 20 Мбіт/с;
- налаштування віртуальної приватної мережі (VPN), що дозволяє забезпечити безпечний тунельний зв'язок між різними комп'ютерами або мережами через незахищену мережу;
  - широкий спектр мережевих протоколів, таких як IPv4, IPv6, DHCP, DNS, NAT, IPSec, PPTP і т.д. Це дозволяє їм ефективно працювати з різними типами мереж та забезпечувати сумісність з іншими пристроями та мережевими системами.

### 2.3.4 Налаштування маршрутизатора Cisco

Для отримання доступу до налаштувань маршрутизатора нам потрібно відкрити будь-який веб-браузер на комп'ютері, до якого він підключений, і ввести одну з наступних адрес в адресному рядку: 192.168.1.1 або 192.168.0.1. (рисунок 2.16) [47].

Після відкриття адреси веб-інтерфейсу маршрутизатора, ви побачите запит на введення імені користувача та паролю. Вам потрібно ввести "admin" як ім'я користувача та "admin" як пароль (це стандартні дані для входу).

Зверніть увагу, що ця інформація може бути надрукована на наклейці на нижній частині роутера, тому важливо зберегти конфіденційність цих даних і забезпечити відповідну фізичну безпеку маршрутизатора.

Однією з найважливіших задач є правильна настройка маршрутизатора для роботи з вашим інтернет-провайдером. Якщо це не буде зроблено або буде зроблено неправильно, це може призвести до відсутності інтернет-з'єднання. Хоча мережа Wi-Fi з'явиться, вона не зможе забезпечити доступ до Інтернету.

Налаштування інтернету:

- Перейдіть на вкладку "SeTup".
- "InTernrT ConnecTion Type" оберіть "STaTic IP".
- Прописуємо налаштування видані провайдером "InTerneT IPv4 Address, SubneT Mask, DefaulT GaTeway, DNS 1, DNS 2".

- Time Zone вказуємо нашу.
- Тиснемо на кнопку “Save SeTTings”.

The screenshot shows the Cisco Linksys E4200 web interface. The top navigation bar includes 'Setup', 'Wireless', 'Security', 'Storage', 'Access Restrictions', 'Applications & Gaming', 'Administration', and 'Status'. The 'Setup' page is active, with sub-tabs for 'Basic Setup', 'IPv6 Setup', 'DDNS', 'MAC Address Clone', and 'Advanced Routing'. The left sidebar contains sections for 'Language', 'Internet Setup', 'Network Setup', 'DHCP Server Setting', 'Time Settings', 'Port Lights', and 'Reboot'. The main content area shows the following settings:

- Language:** English
- Internet Connection Type:** Automatic Configuration - DHCP
- Domain Name:** [Empty field]
- MTU:** Auto (Size: 1500)
- Host Name:** Cisco00095
- IP Address:** 192 . 168 . 1 . 1
- Subnet Mask:** 255.255.255.0
- DHCP Server:** Enabled (DHCP Reservation button)
- Start IP Address:** 192 . 168 . 1 . 100
- Maximum Number of Users:** 50
- IP Address Range:** 192 . 168 . 1 . 100 to 149
- Client Lease Time:** 0 minutes (0 means one day)
- Static DNS 1:** 0 . 0 . 0 . 0
- Static DNS 2:** 0 . 0 . 0 . 0
- Static DNS 3:** 0 . 0 . 0 . 0
- WINS:** 0 . 0 . 0 . 0
- Time Zone:** (GMT-08:00) Pacific Time (USA & Canada)
- Automatically adjust clock for daylight saving changes:** Checked
- Port Lights:** On
- Reboot:** Reboot button

At the bottom right, there are 'Save Settings' and 'Cancel Changes' buttons. The Cisco logo is visible in the bottom right corner of the interface.

Рисунок 2.16 – Веб - інтерфейс налаштування

#### Налаштування інтернету:

- Перейдіть на вкладку “SeTur”.
- “InTernrT ConnecTion Type” оберіть “STaTic IP”.

Зм..	Арк.	№докум.	Підпис	Дата

- Прописуємо налаштування видані провайдером “InTernet IPv4 Address, Subnet Mask, Default GaTeway, DNS 1, DNS 2”.
  - Time Zone вказуємо нашу.
  - Тиснемо на кнопку “Save SeTTings”.
  - На цьому етапі повинен з’явитися інтернет через маршрутизатор.
- На рисунку 2.17 зображений інтерфейс правильно вибраних опцій для налаштування інтернету.

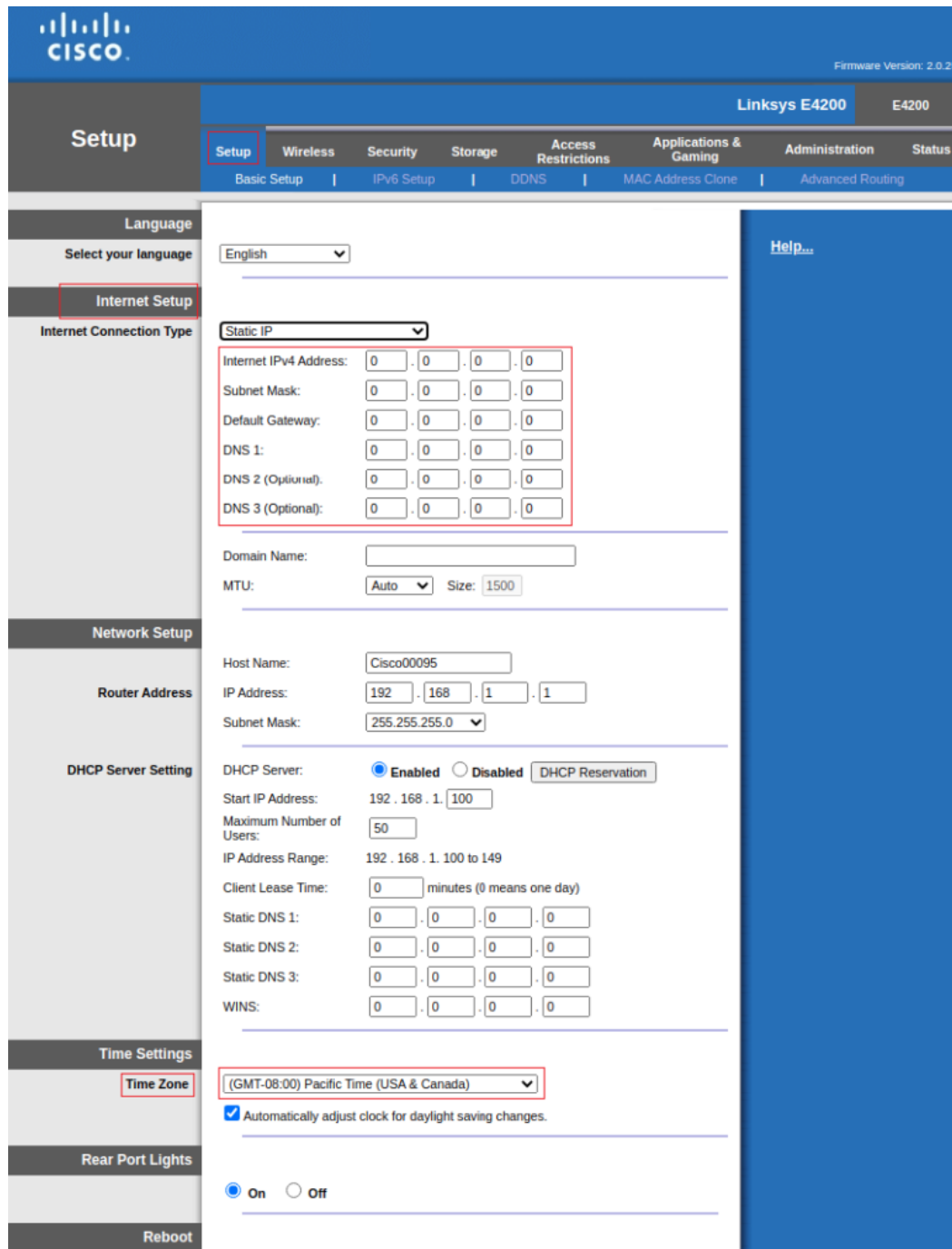


Рисунок 2.17 – Налаштування інтернету

Налаштування Wi-Fi (рисунок 2.18):

1. Перейдіть на вкладку “Wireless”.
2. “ConfiguraTion View” оберіть “Manual”.
3. Напишіть назву бездротової мережі (SSID).
4. Метод аутентифікації рекомендовано WPA2 Personal.
5. Придумайте і запишіть пароль, який буде захищати вашу Wi-Fi мережу (Passphrase).
6. Тиснемо на кнопку “Save SeTTings”.

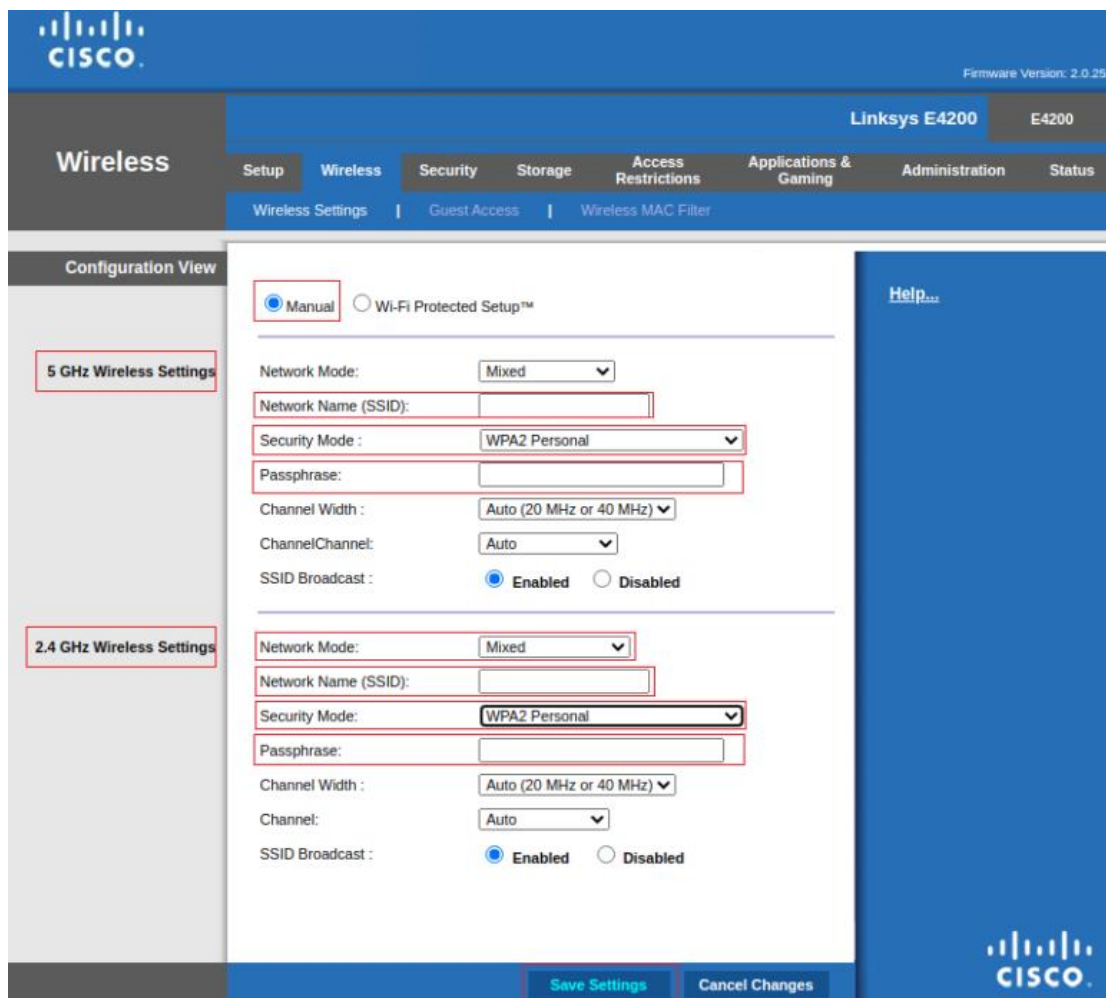


Рисунок 2.18 – Налаштування Wi-Fi

Налаштування маршрутизатора Cisco завершено. Залишилося тільки

Зм..	Арк.	№докум.	Підпис	Дата

перезавантажити його. Це можна зробити через кнопку “Reboot” (рисунок 2.19) в налаштуваннях (SeTur).



Рисунок 2.19 – Перезавантаження

Після налаштування та перезавантаження, з’явиться Wi-Fi мережа, з ім’ям, яке ми встановили в процесі налаштування. А для підключення до бездротової мережі, використовуйте пароль, який ми так само встановили при налаштуванні маршрутизатора.

### 2.3.5 Мережевий адаптер

Для підключення комп’ютерів до обох кілець і забезпечення надійності зв’язку можна використовувати два мережевих адаптери на кожному комп’ютері. Програмне забезпечення на комп’ютері може бути налаштоване таким чином, щоб обидва адаптери працювали одночасно і забезпечували резервування шляху між кільцями.

Загальною практикою для такої конфігурації є використання мережевих адаптерів з підтримкою технології Link Aggregation (також відомої як Ethernet bonding або port Trunking). Ця технологія дозволяє об’єднувати два фізичних адаптери в одну логічну групу, що забезпечує вищу пропускну здатність та надійність.

Прикладом такого мережевого адаптера є "Dell InTel X540-T2 10GbE 03DFV8" (рисунок 2.20). Цей адаптер має два фізичних порти, які можна підключити до різних кілець. Використовуючи технологію Link Aggregation, обидва порти можуть бути об’єднані в одну логічну інтерфейсну групу, що

забезпечує збільшену пропускну здатність та резервування шляху між кільцями. Швидкість передачі даних - 10 Гбіт/с.

Важливо також переконатися, що мережеві адаптери підтримують необхідні стандарти і протоколи для роботи в топології подвійного кільця, наприклад, Spanning Tree Protocol (STP) для запобігання петлям в мережі.



Рисунок 2.20 –Мережева карта Dell InTel X540-T2 10GbE 03DFV8 [48]

### 2.3.6 Кабель для підключення до обох кілець

Кабель вита пара FinMark UTP CAT5e 4P 24AWG 305м внутрішній (рисунок 2.21) є прикладом кабелю, який може бути використаний для підключення в мультикомп'ютерній системі згідно з топологією подвійне кільце.

Цей кабель має категорію CAT5e, що означає, що він відповідає певним стандартам швидкості і підтримує передачу даних зі швидкістю до 1000 Мбіт/с. Кабель має 4 пари виті пари і кількість жилок 24AWG, що забезпечує якісну передачу сигналу.

Він призначений для внутрішньої установки, що робить його ідеальним для підключення комп'ютерів та інших пристроїв в мережі з топологією подвійне кільце. Кабель поставляється у бухтах довжиною 305 метрів, що забезпечує достатню довжину для прокладання кабелю по мережі.

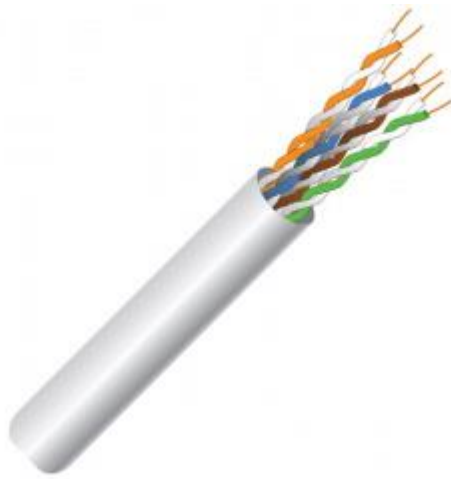


Рисунок 2.21 – Кабель для підключення до обох кілець [49]

## 2.4 Висновки

У цьому розділі було більше детально розглянуто топологію «кілець» та похідну від неї топологію «подвійне кілець». Зрозуміли відмінність між цими топологіями, та порівняли їхні будови. Визначили основні характеристики (діаметр та зв'язність) та покази ефективності роботи. Описали різні архітектури мультикомп'ютерних систем з паралельною обробкою даних, такі як векторно-конвеєрні та обчислювальні системи з розподіленою пам'яттю (до яких відносяться масивно-паралельні комп'ютери. І через те, що більша частина теперішніх «серйозних» комп'ютерів реалізовані в класі обчислювальних систем з розподіленою пам'яттю – основним і найактуальнішим підкласом якої є MPP система, ми обрали її як основну архітектуру для нашої мультикомп'ютерної системи.

Для реалізації мультикомп'ютерної системи згідно топології подвійне кілець ми обрали наступне апаратне забезпечення:

- два комп'ютери (ноутбуки);
- два комутатори;
- два маршрутизатори ;
- два мережевих адаптери;
- кабелі на кожен комп'ютер для підключення до обох кілець;

- кодаткове програмне забезпечення.

Також ми вибрали конкретні назви та види ноутбука, комутаторів, маршрутизаторів, мережевих адаптерів та кабелів і заказали детальний опис та характеристику кожному. Зобразили детальний опис налаштування маршрутизатора Cisco.

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		48

### 3 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ

#### 3.1 Алгоритм під'єднання складових мультикомп'ютерної системи

Схему підключення усіх компонентів до одного ноутбука по топології подвійне кільце можна побачити на рисунку 3.1.

##### 1. Підготовка обладнання:

- підключити два мережевих адаптери до кожного комп'ютера;
- підключити кабелі до кожного комп'ютера для підключення до обох кілець;

##### 2. Підключення комутаторів:

- підключіть перший комутатор до комп'ютера 1 та комп'ютера 2 за допомогою двох кабелів, використовуючи різні порти на комутаторі;
- підключіть другий комутатор до комп'ютера 1 та комп'ютера 2 за допомогою двох кабелів, також використовуючи різні порти на комутаторі;

##### 3. Підключення маршрутизаторів:

- підключіть перший маршрутизатор до першого комутатора за допомогою кабелю;
- підключіть другий маршрутизатор до другого комутатора за допомогою кабелю;

##### 4. Налаштування програмного забезпечення:

- налаштуйте IP-адреси на мережевих адаптерах комп'ютерів та маршрутизаторів відповідно до потребностей вашої мережі;
- встановіть програмне забезпечення для керування мережевими пристроями, що використовуються у системі;

##### 5. Перевірка з'єднання:

- після завершення підключення перевірте з'єднання між комп'ютерами за допомогою пакетів Ping або інших інструментів для мережевої діагностики.

Таким чином, ми підключили кожен комп'ютер до обох комутаторів за допомогою двох кабелів, а також маршрутизатори будуть підключені до

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		49

комутаторів за допомогою кабелів. Це створить топологію подвійне кільце і забезпечить надійність і зайвість в системі.

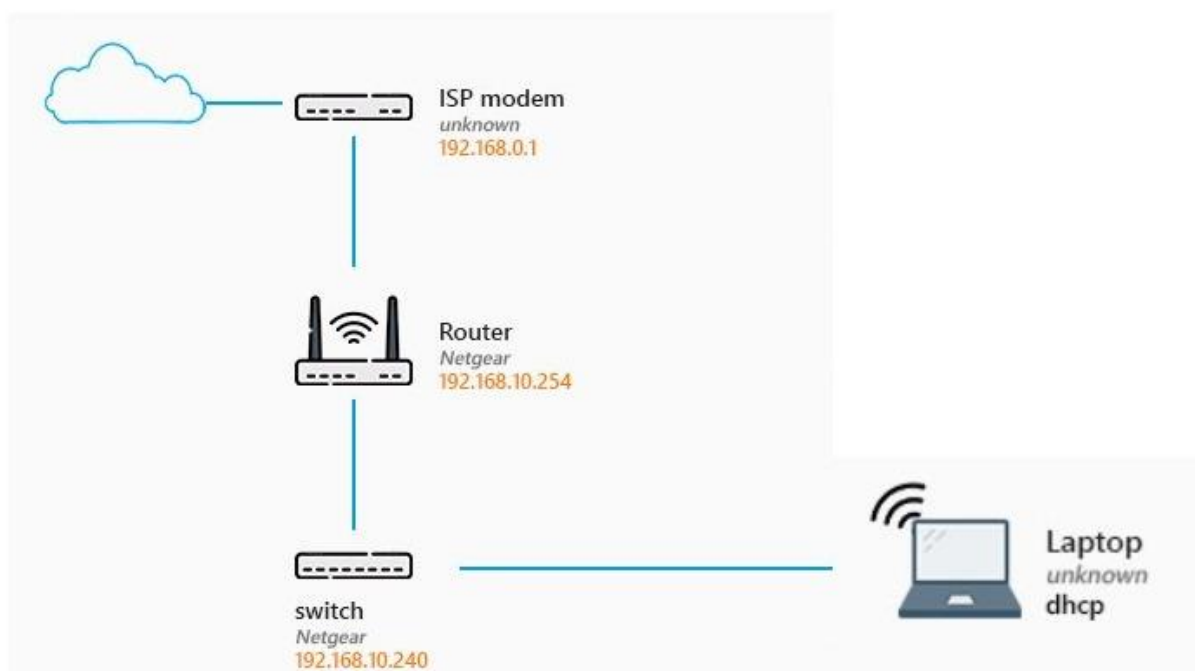


Рисунок 3.1 – Схема підключення компонентів до одного ноутбука

### 3.2 Вибір програмних засобів

C++ є одним з відмінних виборів для розробки мультикомп'ютерної системи з топологією подвійного кільця. C++ є мовою програмування низького рівня, яка дозволяє ефективно керувати ресурсами комп'ютера та прямо працювати з сокетом. Вона має потужні можливості для оптимізації та роботи з мережевими протоколами. C++ також підтримує багатопотоковість та розширюваність коду [50-51]:

C++ є мовою низькорівневого програмування, що дозволяє ефективно використовувати ресурси комп'ютера. В мультикомп'ютерних системах, де передача великих обсягів даних і низька латентність є важливими, продуктивність мови C++ дозволяє оптимізувати роботу з мережевими протоколами та обробку даних.

Можливість безпосередньої роботи з сокетом. C++ надає доступ до низькорівневих API для мережевого програмування, зокрема до сокетів. Це

дозволяє розробникам прямо керувати мережевими з'єднаннями, передачею даних та керуванням мережевими протоколами, що є важливим для реалізації мультикомп'ютерної системи з топологією подвійного кільця.

C++ є крос-платформеною мовою програмування, що дозволяє розробляти програми, які можуть працювати на різних операційних системах. Це важливо для мультикомп'ютерних систем, де комп'ютери можуть використовувати різні операційні системи.

C++ підтримує об'єктно-орієнтований підхід, що дозволяє розробникам створювати модульні, розширювані та повторно використовувані компоненти системи. Це сприяє структуризації та організації коду, що є важливим для складних мультикомп'ютерних систем.

C++ має підтримку багатопотокового програмування, що дозволяє ефективно використовувати ресурси комп'ютера та забезпечувати паралельну обробку даних в мультикомп'ютерних системах.

Сокети (socketS) у мові C++ є інтерфейсом, який, який полегшує взаємодію між різними додатками та дозволяє програмі здійснювати мережеві комунікації. Завдяки сокетам можна легко реалізувати більшість популярних протоколів, які використовуються щодня в Інтернеті. Наприклад, HTTP, FTP, POP3, SMTP та багато інших. Усі вони використовують протоколи передачі даних TCP або UDP, які можна легко програмувати за допомогою бібліотеки socketS/winsock [52].

Використання сокетів у розробці мультикомп'ютерних систем з топологією подвійного кільця може бути досить корисним [53].

Сокети у мові C++ підтримують різні мережеві протоколи, такі як TCP/IP і UDP. Це дозволяє здійснювати надійне та ненадійне (безз'єднане) передавання даних у мережі. Сокети є крос-платформеними, що означає, що вони можуть бути використані на різних операційних системах, таких як Windows, macOS та Linux.

Функції для роботи з сокетом. Мова C++ надає різноманітні функції та класи для роботи з сокетом. Це включає функції для створення сокетів, з'єднання з мережевими пристроями, передачі та отримання даних через сокет тощо.

Мова C++ має вбудовану підтримку багатопотоковості, що дозволяє створювати мультипотоківі програми з використанням сокетів. Це особливо

важливо для мультикомп'ютерних систем, де одночасно може відбуватися багато мережових операцій.

У мові С++ існують різні сторонні бібліотеки, які спрощують роботу з сокетом, забезпечуючи високорівневі інтерфейси та додаткові функціональні можливості.

Сокети допомагають у взаємодії між різними додатками шляхом забезпечення комунікації і передачі даних через мережу. Вони надають програмістам інтерфейс для створення з'єднання між додатками, незалежно від того, чи працюють вони на одному комп'ютері чи розташовані на різних вузлах мережі [54-55].

Вони дозволяють додаткам відправляти і отримувати дані через мережу, встановлювати з'єднання, передавати повідомлення і керувати потоком даних. Загалом, сокети є потужним і універсальним інструментом для забезпечення взаємодії між різними додатками через мережу. Вони дозволяють програмістам створювати мережові додатки з різними функціями і можливостями комунікації.

### 3.3 Архітектура розподілених об'єктів

Це концепція інформаційної системи, в якій об'єкти системи розташовані на різних фізичних вузлах мережі і взаємодіють між собою через передачу повідомлень та викликів методів. У цій архітектурі кожен об'єкт може мати власний стан і поведінку, а взаємодія між об'єктами відбувається за допомогою механізмів віддаленого виклику процедур (RPC) або віддалених об'єктів (RMI). Архітектура розподілених об'єктів дозволяє створювати розподілені системи, в яких різні компоненти можуть працювати на різних вузлах мережі і спільно виконувати завдання [56].

Сокети дозволяють додаткам працювати в різних режимах, таких як клієнт-сервер (рисунок 3.2) або peer-To-peer (рисунок 3.3) , і забезпечують надійну і ефективну комунікацію між додатками. Вони використовуються в широкому спектрі застосувань, включаючи мережові програми, веб-сервери, електронну пошту, чат-додатки та багато інших.

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		52

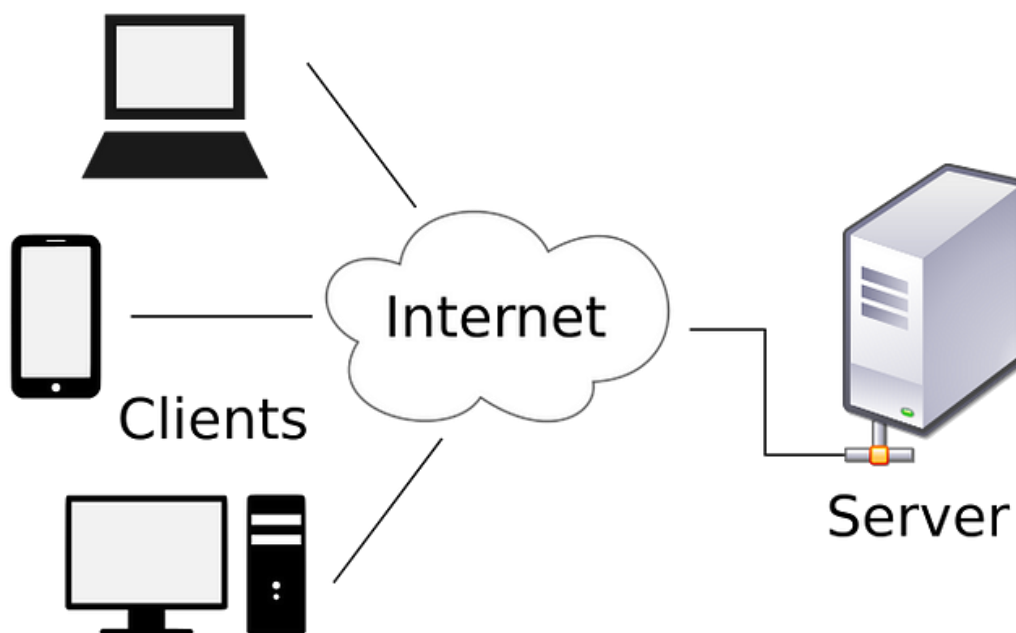


Рисунок 3.2 — Архітектура типу клієнт – сервер [57]

З широким впровадженням Всесвітньої павутини та HTTP в середині 1990-х рр. Інтернет перетворився з ранньої однорангової мережі на мережу споживання контенту. Завдяки цьому перетворенню архітектура клієнт-сервер стала найпоширенішим підходом для передачі даних із новими термінами, такими як «веб-сервер», що закріпили ідею виділених комп'ютерних систем і моделі сервера для цього вмісту [58].

Архітектура клієнт-сервер призначає один комп'ютер або хост як сервер, а інші ПК як клієнти. У цій моделі сервера сервер має бути весь час онлайн із хорошим з'єднанням. Сервер надає своїм клієнтам дані, а також може отримувати дані від клієнтів. Деякі приклади широко використовуваних клієнт-серверних програм – HTTP, FTP, rsync і хмарні служби. Усі ці додатки мають певну функціональність на стороні сервера, яка реалізує протокол, але ролі постачальника та споживача ресурсів чітко розподілені [59].

Однорангова модель (peer-To-peer) відрізняється тим, що всі хости мають однакові привілеї та діють як постачальники, так і споживачі ресурсів, таких як пропускна здатність мережі та комп'ютерна обробка.

Зм..	Арк.	№докум.	Підпис	Дата

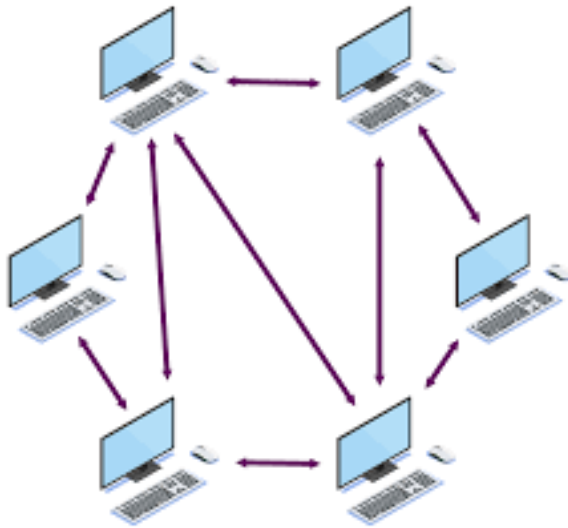


Рисунок 3.3 — Архітектура типу peer-To-peer

Кожен комп'ютер вважається вузлом системи, і разом ці вузли утворюють мережу P2P. Перший Інтернет був розроблений як однорангова мережа, де всі комп'ютерні системи були однаково привілейованими, а більшість взаємодій були двонаправленими.

Найбільш очевидною проблемою, з якою стикаються всі програми клієнт-сервер, є доступність. З моделлю виділеного сервера сервер повинен бути онлайн і постійно доступний для клієнтів, інакше програма просто не працюватиме. Багато речей можуть впливати на доступність сервера через проблеми з програмним забезпеченням, помилки операційної системи та апаратні збої. Помилки маршрутизації та збої в мережі також можуть впливати на доступність [60].

Насправді, з такою кількістю речей, які можуть піти не так (будь-яка з яких виводить з ладу ваш сервер, що виводить з ладу вашу програму), не дивно, що значний час і ресурси витрачаються на забезпечення високої доступності серверів і спроби заздалегідь передбачити проблеми. Конкретні відділи, такі як операційний, часто повністю присвячені проблемі доступності, а цілі галузі, такі як мережі доставки вмісту (CDN) і хмарні обчислення, створені для подолання обмежень доступності клієнт-серверної моделі, як правило, шляхом виділення ще

більше ресурсів для на стороні сервера моделі для забезпечення доступності. Усе це ускладнює роботу та збільшує вартість, оскільки висока доступність вимагає, щоб система перемикалася на резервне обладнання або постачальника послуг Інтернету, якщо з будь-якої причини було порушено, щоб програма продовжувала працювати безперебійно.

Ця проблема досить складна, оскільки вам потрібно підтримувати синхронізацію даних між вашим живим сервером і сервером резервного копіювання, підтримувати альтернативних постачальників послуг і правильно планувати оновлення програмного та апаратного забезпечення заздалегідь, щоб підтримувати безперебійну роботу служби.

У одноранговій мережі кожен клієнт є також сервером. Якщо центральна машина недоступна, послугу може надавати будь-який доступний клієнт або група клієнтів, кожен з яких працює як вузли в мережі. Однорангова система знайде найкращих клієнтів і запросить у них обслуговування. Це забезпечує доступність послуг, яка не залежить від однієї машини та не потребує розробки будь-якого складного рішення високої доступності.

#### 3.4 Моделювання різних ситуацій використання мультимедійних систем по архітектурі типу клієнт - сервер

Сценарій 1. Перевірка підключення компонентів ( 2 ноутбуки, 2 комутатори та маршрутизатор), надіславши повідомлення від одного комп'ютера до іншого (рисунок 3.4). Алгоритм виконання:

- налаштувати два комутатори та маршрутизатори відповідно до кільцевої топології. Встановити відповідні IP-адреси для кожного пристрою;
- підключити обидва ноутбуки до комутаторів за допомогою мережевих адаптерів та кабелів. Кожен ноутбук повинен мати два мережевих адаптери, щоб підключитись до обох кілець;
- зв'язку між різними сегментами кільцевої топології. Встановити правила
- маршрутизації для пересилання повідомлень між ноутбуками;

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		55

- відправка повідомлення: на першому ноутбуку написати програму, яка відправляє повідомлення до другого ноутбука. Використовуйте відповідні сокети та мережеві функції мови програмування C++ для створення з'єднання та передачі даних;
- маршрутизація повідомлення: повідомлення буде передаватись через перший маршрутизатор, який визначає найкоротший шлях до другого ноутбука на основі налаштованих правил маршрутизації. Маршрутизатори пересилають повідомлення через кільце до маршрутизатора, який відповідає за другий ноутбук.
- прийом повідомлення: другий ноутбук отримує повідомлення через другий маршрутизатор, який передавав повідомлення від першого ноутбука.

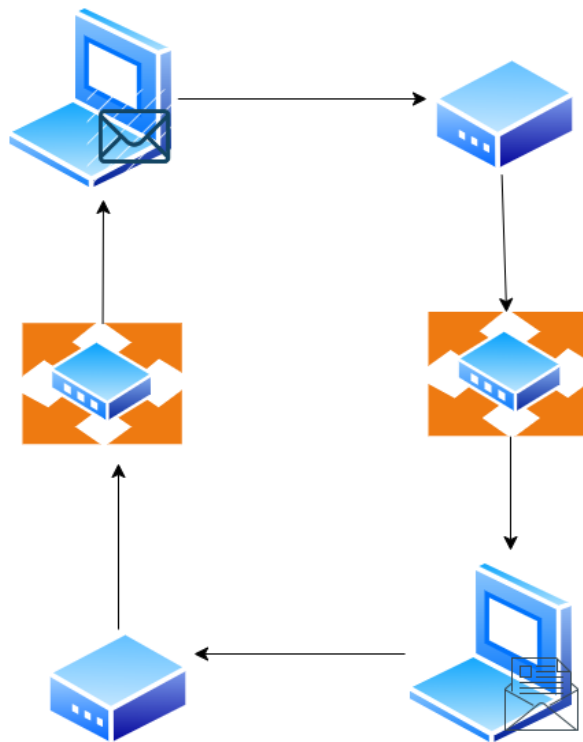


Рисунок 3.4 — Сценарій 1

Сценарій 2. Доданок двох чисел, яке потрібно розв'язати користувачу на одній комп'ютерній станції і це програмне забезпечення, яке пов'язує всі компи в кільце оцінює завантаженість кожного окремого компа і перекидає йому розв'язання цього завдання (рисунок 3.5). Алгоритм виконання:

- користувач вводить два числа для додавання на своїй комп'ютерній станції;
- програмне забезпечення на комп'ютерній станції оцінює завантаженість кожного комп'ютера у кільці;
- програмне забезпечення відправляє запит до кожного комп'ютера у кільці, щоб отримати інформацію про його завантаженість;
- програмне забезпечення вибирає найменш завантажений комп'ютер у кільці для обробки завдання;
- результат додавання надсилається з комп'ютерної станції до вибраного комп'ютера через мережевий протокол (наприклад, TCP/IP);
- вибраний комп'ютер отримує запит на обробку завдання та виконує додавання двох чисел;
- вибраний комп'ютер повертає результат обчислення до програмного забезпечення на комп'ютерній станції;
- програмне забезпечення на комп'ютерній станції отримує результат та відображає його користувачу.

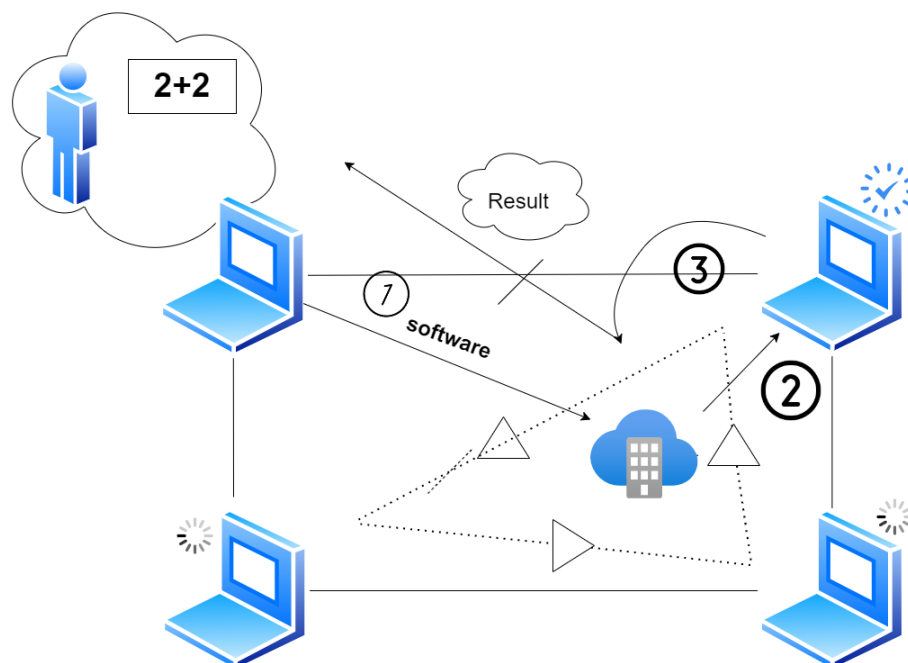


Рисунок 3.5 — Сценарій 2

Сценарій 3. Колективна робота над проектом (рисунок 3.6).

Користувачі у різних комп'ютерах можуть спільно працювати над проектом, ділитися файлами, виконувати спільні завдання та обмінюватися даними через мультимп'ютерну систему по топології подвійне кільце.

Алгоритм виконання:

- користувачі запускають програмне забезпечення для колективної роботи над проектом на своїх комп'ютерах;
- кожен користувач підключається до мультимп'ютерної системи по топології кільце через мережевий протокол (наприклад, TCP/IP);
- комп'ютери у мультимп'ютерній системі утворюють кільце топології, де кожен комп'ютер з'єднаний з попереднім і наступним комп'ютерами.;
- користувачі можуть ділитися файлами, виконувати спільні завдання та обмінюватися даними. Кожен користувач може внести зміни у спільні файли або створити нові файли.;
- коли користувач зберігає зміни або виконує спільне завдання, програмне забезпечення розподіляє ці зміни по мультимп'ютерній системі;
- кожен комп'ютер у кільці отримує зміни та оновлює локальні файли або виконує відповідні дії;

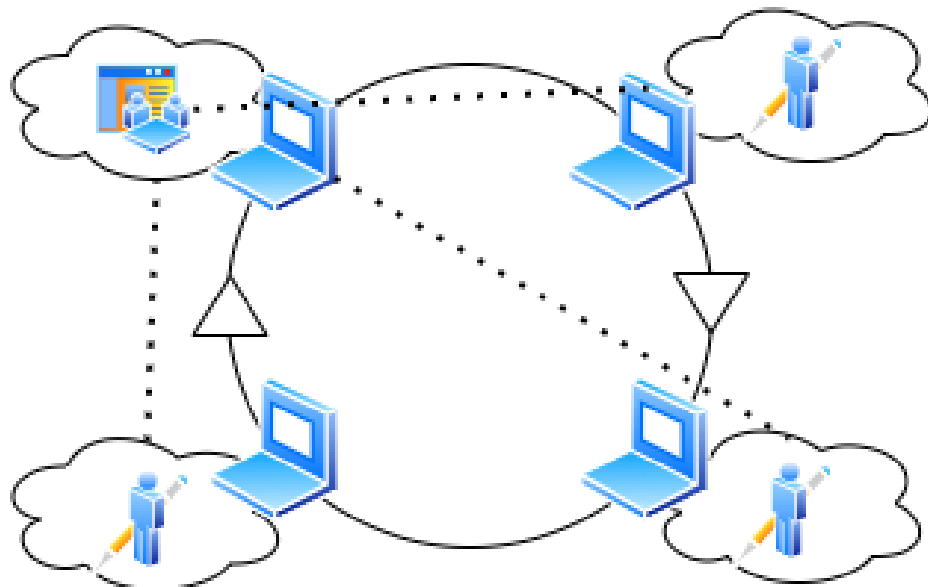


Рисунок 3.6 — Сценарій 3

Зм.	Арк.	№докум.	Підпис	Дата

- користувачі можуть бачити оновлені файли та зміни, внесені іншими користувачами у мультимедійній системі;
- колективна робота продовжується, а користувачі можуть продовжувати внесення змін, обмін даними та виконання спільних завдань;
- коли робота над проектом завершується, користувачі можуть зберегти остаточну версію файлів та завершити з'єднання з мультимедійною системою.

#### Сценарій 4. Відеоконференції (рисунок 3.7)

Клієнти підключаються до сервера для участі в відеоконференціях. Сервер керує передачею відео- та аудіоданих між клієнтами, забезпечує синхронізацію та забезпечує ефективну комунікацію.

#### Алгоритм виконання:

- клієнти запускають програмне забезпечення для відеоконференцій на своїх комп'ютерах;
- кожен клієнт підключається до мультимедійної системи по топології кільце через мережевий протокол (наприклад, TCP/IP);
- комп'ютери у мультимедійній системі утворюють кільце топології, де кожен комп'ютер з'єднаний з попереднім і наступним комп'ютерами;
- клієнти надсилають запити до сервера для підключення до відео - конференції та передачі відео- та аудіоданих;
- сервер отримує запити від клієнтів і керує процесом відеоконференції;
- сервер забезпечує синхронізацію між клієнтами, контролює якість передачі відео- та аудіоданих та забезпечує ефективну комунікацію;
- кожен комп'ютер у кільці отримує відео- та аудіодані від попереднього комп'ютера та передає їх наступному комп'ютеру в кільці;
- клієнти можуть бачити відео- та аудіодані інших учасників відеоконференції та взаємодіяти з ними;
- коли відеоконференція завершується, клієнти відключаються від сервера і закривають програмне забезпечення для відеоконференцій.

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
						59
Зм..	Арк.	№докум.	Підпис	Дата		

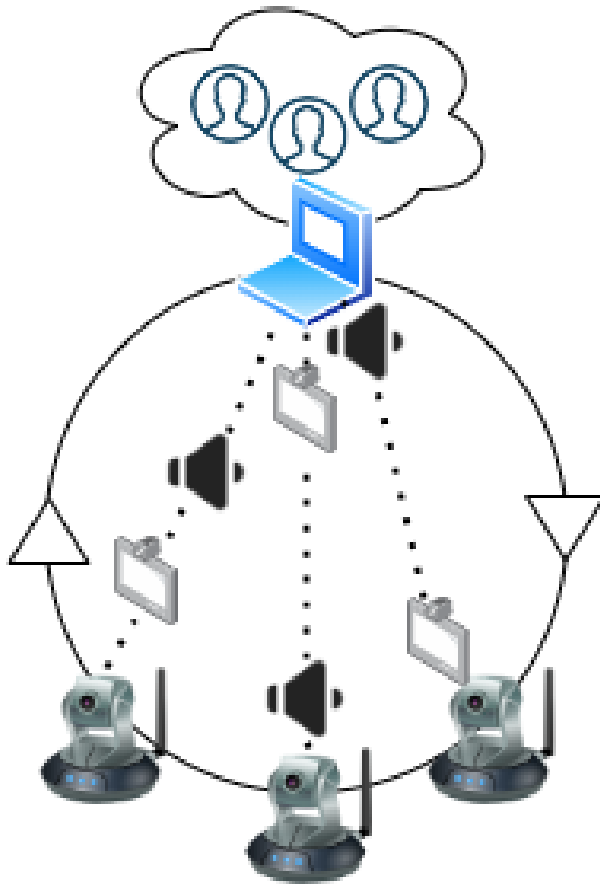


Рисунок 3.7 — Сценарій 4

#### Сценарій 5. Система контролю доступу (рисунок 3.8а та 3.8б)

Клієнти взаємодіють з сервером для аутентифікації, авторизації та керування доступом до ресурсів. Сервер зберігає та обробляє інформацію про права доступу та встановлює правила для контролю доступу до різних об'єктів.

##### Частина 1 – Аутентифікація:

- клієнт запускає програмне забезпечення для системи контролю доступу на своєму комп'ютері;
- клієнт встановлює з'єднання з сервером, використовуючи мережевий протокол (наприклад, TCP/IP);
- клієнт надсилає запит до сервера для аутентифікації, передаючи ідентифікаційні дані, наприклад, логін та пароль;
- сервер отримує запит аутентифікації від клієнта і перевіряє ідентифікаційні дані. Якщо дані вірні, сервер продовжує виконання, в іншому випадку відхиляє запит.

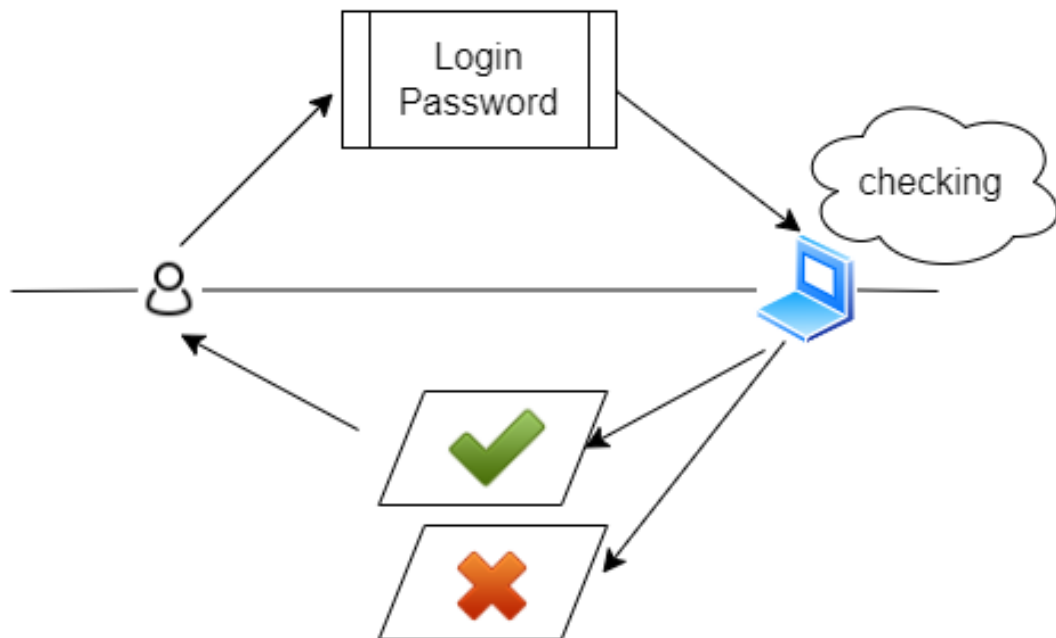


Рисунок 3.8а — Сценарій 5 (Частина 1 –Автентифікація)

Частина 2 – Запит на доступ:

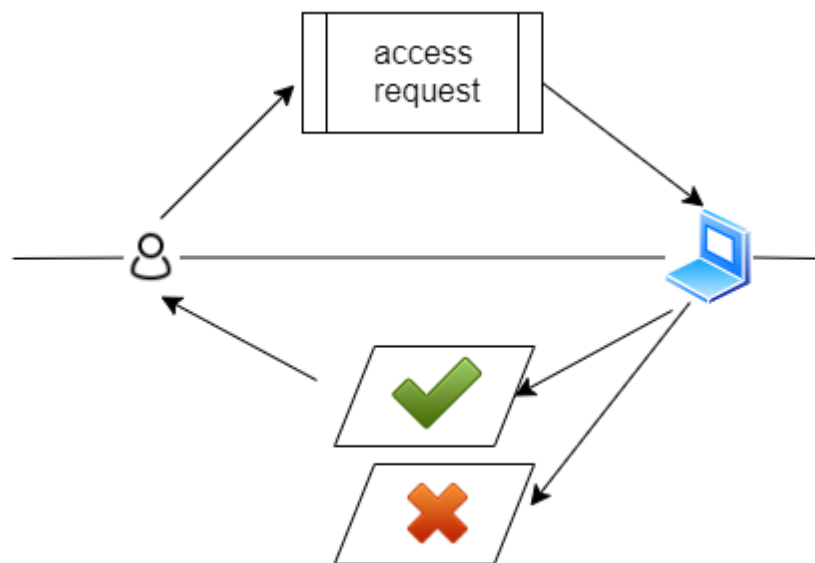


Рисунок 3.8б — Сценарій 5 (Частина 2 – Запит на доступ)

- після успішної аутентифікації, клієнт надсилає запит до сервера для авторизації, передаючи інформацію про потрібні ресурси або дії, до яких потрібен доступ;

Зм..	Арк.	№докум.	Підпис	Дата

- сервер отримує запит авторизації від клієнта і перевіряє права доступу користувача до вказаних ресурсів або дій. Якщо права доступу надані, сервер дозволяє виконання запиту, в іншому випадку відхиляє запит;
- якщо запит був успішно авторизований, сервер надає клієнту доступ до вказаних ресурсів або дозволяє виконання вказаних дій;
- сервер зберігає та обробляє інформацію про права доступу до ресурсів та встановлює правила контролю доступу до різних об'єктів;
- після завершення взаємодії, клієнт закриває з'єднання з сервером.

#### Сценарій 6. Система віддаленого керування

Клієнти використовують сервер для віддаленого керування різними пристроями або системами. Це може бути віддалене керування серверами, мережевими пристроями, системами безпеки тощо. Клієнти надсилають команди серверу, який передає їх відповідним комп'ютерам або пристроям для виконання.

Алгоритм виконання:

Частина 1 (рисунок 3.9а):

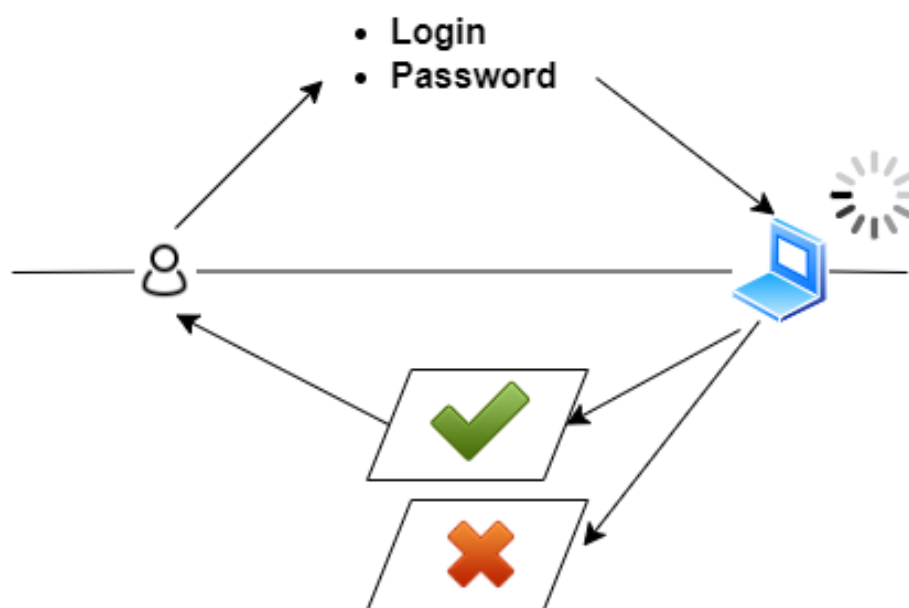


Рисунок 3.9а — Сценарій 6 (Частина 1 – Запит на доступ)

- клієнт запускає програмне забезпечення для системи віддаленого керування на своєму комп'ютері;

Зм..	Арк.	№докум.	Підпис	Дата

- клієнт встановлює з'єднання з сервером, використовуючи мережевий протокол (наприклад, TCP/IP);
- сервер отримує команду від клієнта і перевіряє права доступу користувача до відповідного пристрою або системи;
- якщо права доступу надані, сервер передає команду відповідному комп'ютеру або пристрою, який підпорядкований серверу.

Частина 2 (рисунок 3.9б):

- клієнт надсилає команду до сервера для віддаленого керування, передаючи необхідну інформацію про пристрій або систему, яку потрібно керувати, та команди для виконання;
- комп'ютер або пристрій отримує команду від сервера і виконує відповідні дії відповідно до команди;
- якщо виконання команди потребує відповіді або стану пристрою, комп'ютер або пристрій повертає відповідь до сервера;
- сервер отримує відповідь від комп'ютера або пристрою і передає її клієнту;
- клієнт отримує відповідь від сервера і відображає її користувачу.

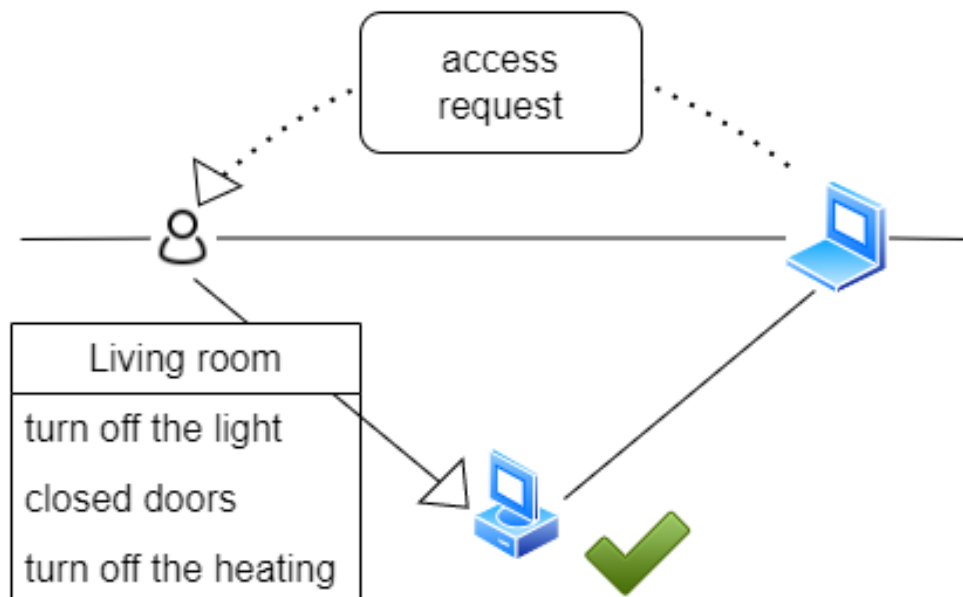


Рисунок 3.9б — Сценарій 6 (Частина 2)

### 3.5 Архітектура програмного забезпечення

Завдання до якого ми писали програмне забезпечення це сценарій 2 з підрозділу 3.4 і звучить воно так: доданок двох чисел, яке потрібно розв'язати користувачу на одній комп'ютерній станції і це програмне забезпечення, яке пов'язує 3 комп'ютери в кільце оцінює завантаженість кожного окремого компа і перекидає розв'язання цього завдання найменш завантаженому.

У цій UML діаграмі (рисунок 3.10) представлені три класи: Server, Client і LoadBalancer. Server та Client взаємодіють між собою, а також з LoadBalancer. Server має атрибути socket, address і port, які використовуються для мережевого з'єднання. Кожен клас має набір публічних методів для взаємодії та виконання необхідних операцій.

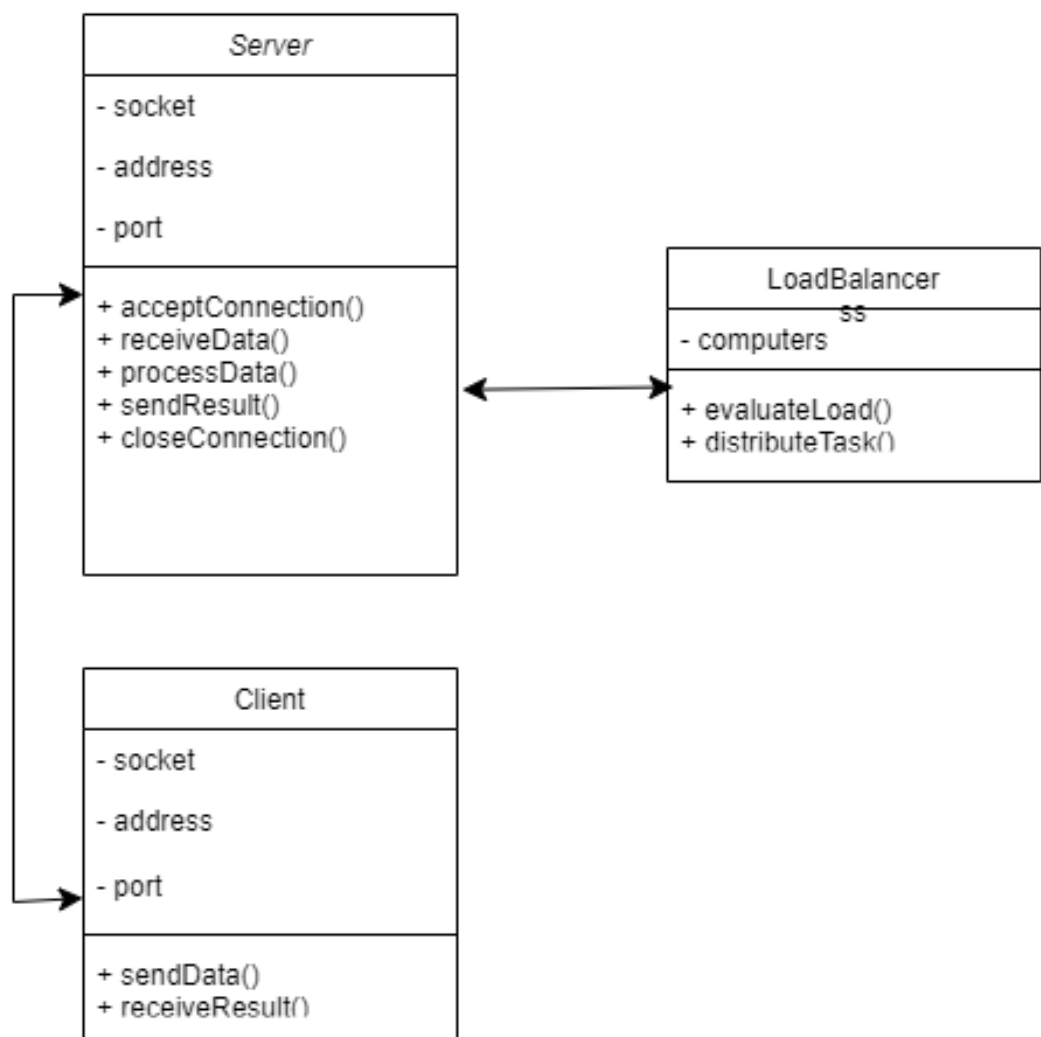


Рисунок 3.10 – UML-діаграма архітектури програмного забезпечення

1. Створення сокета - створюється сокет за допомогою функції `socket()` з параметрами `AF_INET` (для використання IPv4) і `SOCK_STREAM` (для забезпечення з'єднання на основі TCP).

2. Прив'язка сокета до адреси і порту - задається адреса сервера та порт, до яких будуть підключатися клієнти. Функція `bind()` використовується для прив'язки сокета до цих значень.

3. Прослуховування вхідних з'єднань - сервер починає прослуховувати вхідні з'єднання використовуючи функцію `listen()` з параметром, що вказує на максимальну кількість з'єднань, які можуть бути прийняті.

4. Прийняття з'єднання - коли клієнт намагається підключитися до сервера, використовується функція `accept()`, щоб прийняти з'єднання. Ця функція створює новий сокет `new_sock`, який буде використовуватися для обміну даними з клієнтом.

5. Отримання чисел від клієнта - за допомогою функції `recv()`, сервер отримує два числа, які надіслані клієнтом через `new_sock`. Ці числа зберігаються в змінних `num1` та `num2`.

6. Виконання додавання чисел - сума двох чисел обчислюється і зберігається в змінній `sum`.

7. Виведення результату - сума чисел виводиться на консоль.

8. Закриття з'єднання - після завершення обробки завдання, з'єднання з клієнтом закривається за допомогою функції `close()`.

9. Передача завдання наступному комп'ютеру - завантаженість кожного комп'ютера в кільці оцінюється, і завдання перекидається до найменш завантаженого комп'ютера. Для цього можна використовувати зовнішній алгоритм оцінки завантаженості.

### 3.6 Висновки

У третьому заключному розділі написали алгоритм та намалювали схему підключення усіх компонентів до ноутбука.

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		65

Обрали мову C++ для написання проміжного програмного забезпечення та описали її переваги для підключення компонентів та розробки завдання в мультимедійній системі.

Провели порівняння між структурою будови мультимедійної системи ( архітектура типу peer-To-peer чи тип клієнт – сервер. І обрали архітектуру типу клієнт – сервер.

І в останньому підпункті третього підрозділу описали сценарії, намалювали схеми зображення різних ситуацій використання мультимедійних систем по архітектурі типу клієнт – сервер:

- перевірка підключення компонентів;
- доданок двох чисел, яке потрібно розв'язати користувачу на одній комп'ютерній станції;
- колективна робота над проектом;
- відеоконференції;
- система контролю доступу;
- система віддаленого керування.

## ВИСНОВКИ

Електронні обчислювальні машини (ЕОМ) та обчислювальна техніка мають велике значення у сучасному світі і застосовуються в різних сферах діяльності, таких як бізнес, наука, медицина та інженерія. Вони забезпечують збір, обробку, зберігання та передачу інформації.

ЕОМ з'явилися у 20-му столітті і принесли нові можливості для обробки інформації та виконання складних обчислень. З того часу обчислювальна техніка стала доступнішою та потужнішою, що дозволило їй застосовуватися у різних галузях.

Сьогодні ЕОМ та обчислювальна техніка є невід'ємною частиною більшості сучасних організацій. Вони використовуються для збору та аналізу даних, розробки програмного забезпечення, візуалізації та симуляції процесів, а також для забезпечення комунікації та зв'язку.

Мультикомп'ютерні системи однією з найважливіших складових обчислювальної техніки. Вони складаються з багатьох взаємопов'язаних комп'ютерів, які працюють як єдине ціле. Мультикомп'ютерні системи мають переваги високого рівня масштабованості і можливості забезпечення високої надійності. Вони легко розширюються за допомогою додаткових комп'ютерів і можуть продовжувати працювати, навіть якщо один з комп'ютерів в системі відмовить.

Створення мультикомп'ютерної системи загального призначення на основі топології "подвійне кільце" є метою даної роботи. Вона передбачає аналіз різних архітектур, які можуть використовуватися в системах з розподіленою пам'яттю, а також опис проектування такої системи з детальним обґрунтуванням вибору та застосування апаратно-програмної бази.

В першому розділі було проведено детальний аналіз обчислювальних систем, зосередившись на класифікації Флінна та її різних типах паралельної обробки. Було розглянуто різні складові мультикомп'ютерних систем, такі як комп'ютери, мережеві з'єднання, програмне забезпечення, протоколи комунікації та система керування ресурсами. Також було оглянуто важливість операційних

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		67

систем у роботі комп'ютерів і їх роль як посередника між користувачем і апаратним забезпеченням. Описано різні топології мультикомп'ютерних систем, які визначають структуру підключення комп'ютерів та організацію комунікаційного середовища. Крім того, проведено аналіз наявних мультикомп'ютерних систем для кращого розуміння їхньої роботи та призначення.

В другому розділі була детально розглянута топологія "кілець" та її похідна, топологія "подвійне кілець". Виявлено відмінності між цими топологіями та проведено порівняння їх структур. Були визначені основні характеристики (діаметр та зв'язність) та показано ефективність роботи цих топологій. Описано різні архітектури мультикомп'ютерних систем з паралельною обробкою даних, зокрема векторно-конвеєрні системи та обчислювальні системи з розподіленою пам'яттю, зокрема масивно-паралельні комп'ютери. Виходячи з того, що більшість сучасних "серйозних" комп'ютерів реалізовані у класі обчислювальних систем з розподіленою пам'яттю, основною та найактуальнішою підкласом яких є MPP системи, було обрано MPP систему як основну архітектуру для мультикомп'ютерної системи. У розділі було наведено апаратне забезпечення, яке було вибрано для реалізації мультикомп'ютерної системи за топологією "подвійне кілець". До цього апаратного забезпечення входять два ноутбуки, два комутатори, два маршрутизатори, два мережеві адаптери, кабелі для підключення до обох кілець та додаткове програмне забезпечення. Також було надано конкретні назви та види ноутбуків, комутаторів, маршрутизаторів, мережевих адаптерів та кабелів, а також надано детальний опис та характеристику кожного з них. Було згадано налаштування маршрутизатора Cisco.

У третьому заключному розділі було написано алгоритм і намальовано схему підключення всіх компонентів до ноутбука. Для написання проміжного програмного забезпечення була обрана мова програмування C++, і були описані її переваги для підключення компонентів та розробки завдання в мультикомп'ютерній системі. Було проведено порівняння між структурою будови мультикомп'ютерної системи, зокрема архітектурою типу peer-To-peer та типу клієнт-сервер, і була обрана архітектура типу клієнт-сервер. В останньому

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		68

підпункті третього підрозділу були описані сценарії та намальовані схеми зображення різних ситуацій використання мультимедійних систем за архітектурою типу клієнт-сервер. Серед них були перевірка підключення компонентів, додавання двох чисел, колективна робота над проектом, відеоконференції, система контролю доступу та система віддаленого керування.

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		69

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

- 1) P. Mehrotra and J. Van Rosendale. Programming distributed memory architectures. In *Advances in Languages and Compilers for Parallel Computing*. MIT Press, 1991.
- 2) Литвин О. С. Співак, С. М.. Паралельні та розподілені обчислення. 2014.
- 3) Класифікація комп'ютерних систем за класифікацією Флінна. URL: [https://upload.wikimedia.org/wikipedia/commons/b/b1/1\\_%D1%84%D0%BE%D1%82%D0%BE.png](https://upload.wikimedia.org/wikipedia/commons/b/b1/1_%D1%84%D0%BE%D1%82%D0%BE.png) (дата звернення 05.03.2023)
- 4) Kammerer, Klaus, et al. Ambalytics: A scalable and distributed system architecture concept for bibliometric network analyses. *Future Internet*, 2021, 13.8: 203 р.
- 5) Литвин, О. С.. *Вивчення основ паралельних та розподілених обчислень із використанням одноплатних мікрокомп'ютерів Raspberry Pi*. 2019. 137 с.
- 6) Ігнатченко М. С. Лінгвістичне забезпечення скінченно-елементного моделювання у паралельних обчислювальних системах . Запорізький національний університет, Запоріжжя, 2021
- 7) Кластерна технологія. URL: [https://wiki.tntu.edu.ua/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%B8\\_%D0%B1%D0%BE%D1%80%D0%BED1%82%D1%8C%D0%B1%D0%B8\\_%D0%B7\\_Dos\\_%D0%B0%D0%B1%D0%BE\\_DDos\\_%D0%B0%D1%82%D0%B0%D0%BA%D0%B0%D0%BC%D0%B8](https://wiki.tntu.edu.ua/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%B8_%D0%B1%D0%BE%D1%80%D0%BED1%82%D1%8C%D0%B1%D0%B8_%D0%B7_Dos_%D0%B0%D0%B1%D0%BE_DDos_%D0%B0%D1%82%D0%B0%D0%BA%D0%B0%D0%BC%D0%B8) (дата звернення 06.03.2023)
- 8) Хмарні обчислення. URL: <https://studfile.net/preview/5370365/page:2/> (дата звернення 06.03.2023)
- 9) Суперкомп'ютер. URL: <https://itechua.com/articles/183978> (дата звернення 07.03.2023)
- 10) Хмарні технології. URL: <https://naurok.com.ua/metodichni-rekomendacii-schodo-vikoristannya-hmarnih-tehnologiy-v-roboti-vchitelya-yaki-dozvoliyayut-organizovuvati-osvitniy-proces-u-vidkritomu-informaciyному-osvitnomu-seredovischi-219224.html> (дата звернення 09.03.2023)
- 11) Глоба В. С. Розробка інформаційнихресурсів та систем. 2020.

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		70

12) K. Yu, D. Han, C. Youn, S. Hwang and J. Lee, "Power-aware task scheduling for big.LITTLE mobile processor," International SoC Design Conference (ISOCC), 2013, pp. 208-212, DOI: 10.1109/ISOCC.2013.6864009

13) Князева А. О. Підвищення ефективності паралельних обчислень на розрядному рівні у системах на кристалі. 2021. Master's Thesis. КПІ ім. Ігоря Сікорського.

14) Класифікація програмного забезпечення. URL: <https://sites.google.com/site/inform323/klasifikacia-pz> (дата звернення 15.03.2023) .

15) Характеристика програмного забезпечення. URL: <http://www.kievoit.ippo.kubg.edu.ua/kievoit/2013/56/56.html> (дата звернення 16.03.2023)

16) Галочкін О. В. Операційні системи. Частина 1. 2022. 242 с.

17) Найпопулярніші операційні системи в світі. URL: <https://marketer.ua/ua/statsoperating-system-2017/> (ата звернення 17.03.2023)

18) Проміжне програмне забезпечення. Дата звернення 18.03.2023 URL: <https://victana.lviv.ua/knyhy/konspektyleksii/133krosplatformenneproqramuvannia-ta-khmarni-servisy/569-lektsiia-21-promizhne-proqramne-zabezpechennia-2016-r>

19) Козіна О.А. *Архітектура проміжного програмного забезпечення для узгодження даних в мультимарних системах* / О.А. Козіна, В.І. Панченко, О.М. Рисований // Вісник НТУ "ХПІ". Серія: Інформатика та моделювання. – Харків: НТУ "ХПІ". – 2021. – №2 (6). – С.101-112.

20) Дегтярьова Л.М., Гроза П.М., Сомов С.В. Навчальний посібник з дисципліни «Технології розробки програмного забезпечення» для студентів спеціальності 123 «Комп'ютерна інженерія». Полтава: ПолтНТУ, 2017. 218 с

21) Ковтун, Л. О., Франчук, Р., Ткачук, В. М. *Вибір архітектури програмного забезпечення інформаційної навчальної системи*. 2019. 552 с.

22) Класифікація програмного забезпечення. URL: <https://hinews.pp.ua/kompyuteri/9929-programne-zabezpechennya-klasifikasya-programnogo-zabezpechennya-ta-yogo-osnovn-tipi.html> (дата звернення 04.03.2023)

23) Вступ до топології мережі. URL: <https://www.cablify.ca/an-introduction-to-network-topology/> (дата звернення 18.03.2023)

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		71

24) Роль топології мережі <https://uk.warbletoncouncil.org/topologias-de-red-16302> (дата звернення 18.03.2023)

25) Круцкевич Н. Перспективи розвитку коп'ютерних мереж з реконфігурацією архітектури на базі пам'яті колективного доступу. 2004.

26) Гарнавський Ю. А., Кузьменко І. М. *Організація комп'ютерних мереж*. 2018. 259 с.

27) Мережеві топології. URL: <https://sites.google.com/site/kmposibnyk/lekci/lekcia3?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1> (дата звернення 20.03.2023).

28) Топологія мережі та її види. URL: <https://instagalleryapp.com/chistij-administrator-2/topologija-merezhi-6-pojasnenih-ta-porivnjanih/> (дата звернення 21.03.2023)

29) Jonathan M. D. Hill. An introduction to the data-parallel paradigm. Department of Computer Science, 1994. 699 – 714 p.

30) Приклади алгоритмів. URL: [https://uk.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0\\_%D0%BF%D0%B0%D1%80%D0%B0%D0%BB%D0%B5%D0%BB%D1%8C%D0%BD%D0%B8%D1%85\\_%D0%BE%D0%B1%D1%87%D0%B8%D1%81%D0%BB%D1%8E%D0%B2%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D1%85\\_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC](https://uk.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%BF%D0%B0%D1%80%D0%B0%D0%BB%D0%B5%D0%BB%D1%8C%D0%BD%D0%B8%D1%85_%D0%BE%D0%B1%D1%87%D0%B8%D1%81%D0%BB%D1%8E%D0%B2%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D1%85_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC) (дата звернення 15.05.2023)

31) Конвеєрна обробка команд: URL: <https://ua.waykun.com/articles/konveerna-obrobka-komand.php> (дата звернення 15.05.2023)

32) Bertsekas Dimitri P. Parallel and Distributed Computation. Numerical Methods / Dimitri P. Bertsekas, John N. Tsitsiklis. – New Jersey: Prentice Hall, Englewood Cliffs, 1989. – 718 p.

33) Масивно-паралельні комп'ютери. URL: [https://studwood.net/1045662/-informatika/proektna\\_chastina](https://studwood.net/1045662/-informatika/proektna_chastina) (дата звернення 25.03.2023)

34) Умовне представлення мультикомп'ютера. URL: <https://studfile.net/preview/2954941/> (дата звернення 10.04.2023)

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		72

- 35) Кластерні системи. URL: <https://www.esds.co.in/blog/cluster-computing-definition-architecture-and-algorithms/> (дата звернення 15.04.2023)
- 36) Кластерні системи. URL: <https://www.watelectronics.com/cluster-computing-architecture-its-types/> (дата звернення 16.04.2023)
- 37) Шеховцов В. А. *Операційні системи* / В. А. Шеховцов. – К. : Видавнича група ВНУ, 2005. – 576 с.
- 38) Рольщиков В.Б *Технології розподілених систем та паралельних обчислень*: конспект лекцій. Одеса, Одеський державний екологічний університет, 2016. 155 с.
- 39) Лопандя, М. В. *Обчислювальні системи з розподіленою пам'яттю..* PhD Thesis. Сумський державний університет. 2013.
- 40) Ian Foster. *Designing and Building Parallel Programs*. Addison-Wesley, 1995. 370 p.
- 41) Тютюнник М. Паралельні алгоритми та засоби для розв'язання деяких задач масових обчислень. *Конференція молодих вчених «Підстригачівські читання–20010»*. м. Львів, 25-26.
- 42) Русанова О. В., Писарчук О. О. *Планування обчислень в паралельних та розподілених комп'ютерних системах*. Лабораторний практикум. 2022.
- 43) Ноутбук. URL: <https://www.asus.com/ua-ua/laptops/forhome/-vivobook/asusvivobook-17x-m3704/> (дата звернення 22.05.2023).
- 44) Комутатор. URL: <https://rozetka.com.ua/ua/154818826/p154818826/> (дата звернення 22.05.2023).
- 45) Маршрутизатор. URL: <https://rozetka.com.ua/ua/42385/p42385/> (дата звернення 22.05.2023).
- 46) Маршрутизатор Cisco. URL: <https://secur.ua/marshrutizator-cisco-rv134w-wireless-n-vpn-router-rv134w-e-k9-g5.html> (дата звернення 22.05.2023).
- 47) Налаштування маршрутизатора Cisco. URL: <https://lucky.net/cisco/> (дата звернення 22.05.2023).
- 48) Мережева карта. URL: <https://hard.kiev.ua/merezheva-karta-dell-intel-x540t210gbe03dfv8/?gclid=Cj0KCQjw98ujBhCgARIsAD7QeAhycSSggpWJH5y260V>

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		73

[5IdpULEPJEE9rnfD2KthjSn5J-iISY7Q-HZAaAhTREALw\\_wcB](#) (дата звернення 22.05.2023)

49) Кабель мережевий. URL: <https://rozetka.com.ua/ua/326958808-/p326958808/> (дата звернення 22.05.2023).

50) Тарарака В.Д. *Обчислювальна техніка. Ч. II. Апаратні засоби персональних комп'ютерів*: навчальний посібник. Житомир ЖВІРЕ, 2004. – 308 с.

51) Уільямс, Ентоні. *Паралельне програмування на C++ діє. Практика розробки багатопотокових програм*. Litres, 2022.

52) A. Thomasian and P. F. Bay. Analytic queuing network models for parallel processing of task systems. IEEE Trans. Computs., C-35(12):1045--1054, 1986.

53) Сокет (програмний інтерфейс). URL: [https://uk.wikipedia.org/wiki/%D0%A1%D0%BE%D0%BA%D0%B5%D1%82\\_\(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B8%D0%B9\\_%D1%96%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81\)](https://uk.wikipedia.org/wiki/%D0%A1%D0%BE%D0%BA%D0%B5%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B8%D0%B9_%D1%96%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81)) (дата звернення 10.05.2023).

54) Використання сокетів у розробці мультикомп'ютерних систем. URL: <https://uk.go-travels.com/21025-socket-programming-for-computer-networking-4056385-8457705> (дата звернення 11.05.2023).

55) Програма Cisco Networking Academy, CCNA 1 і 2 Companion Guide переглянуто третє видання, Р.480

56) Коновалов, О. Ю. Планування архітектури розподілених обчислювальних систем. Наукові записки Українського науково-дослідного інституту зв'язку, 2013, 4: 53-60 с.

57) Клієнт-серверна архітектура. URL: <https://medium.com/@IvanZmerzlyi-/%D0%BA%D0%BB%D1%96%D1%94%D0%BD%D1%82%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%BD%D0%B0%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0%D1%82%D0%B0%D1%80%D0%BE%D0%BB%D1%96%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D1%96%D0%B29893d8048229> (дата звернення 02.04.2023)

58) Sallow, Amira B., et al. Client/Server remote control administration system: Design and implementation. Int. J. Multidiscip. Res. Publ, 2020, 3.2: 7.

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		74

59) Гладій, Б.; Якубів, П. Архітектура клієнт-сервер та веб-сервери. Збірник тез V Всеукраїнської студентської науково-технічної конференції „Природничі та гуманітарні науки. Актуальні питання“, 2012, 1: 51-51.

60) Однорангові та клієнт-серверні мережі. URL: <https://www.resilio.com/blog-/whats-the-difference-between-peer-to-peer-and-client-server> (дата звернення 10.04.2023).

					КвРКІП. 190106.19.01.03 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		75

**Додаток А**  
(обов'язковий)

Лістинг програмного забезпечення

```
#include <iostream>
#include <csdlib>
#include <csstring>
#include <unistd.h>
#include <sys/socket.h>
#include <arpa/inet.h>

#define PORT 8888

int main() {
    int sock, new_sock;
    struct sockaddr_in server, client;
    int len = sizeof(struct sockaddr_in);

    // Створення сокета
    sock = socket(AF_INET, SOCK_STREAM, 0);
    if (sock == -1) {
        std::cerr << "Не вдалося створити сокет" << std::endl;
        exit(1);
    }

    server.sin_family = AF_INET;
    server.sin_port = htons(PORT);
    server.sin_addr.s_addr = INADDR_ANY;

    // Прив'язка сокета до адреси та порту
    if (bind(sock, (struct sockaddr *)&server, sizeof(server)) < 0) {
        std::cerr << "Помилка прив'язки" << std::endl;
        exit(1);
    }

    // Прослуховування вхідних з'єднань
    listen(sock, 3);

    std::cout << "Сервер очікує з'єднання..." << std::endl;

    // Прийняття з'єднання від клієнта
    new_sock = accept(sock, (struct sockaddr *)&client, (socklen_t
*)&len);
```

```
if (new_sock < 0) {  
    sTd::cerr << "Помилка прийняття з'єднання" << sTd::endl;  
    exit(1);  
}
```

```
sTd::cout << "З'єднання прийнято" << sTd::endl;
```

```
// Отримання чисел від клієнта  
int num1, num2;  
recv(new_sock, &num1, sizeof(int), 0);  
recv(new_sock, &num2, sizeof(int), 0);
```

```
// Виконання додавання чисел  
int sum = num1 + num2;  
sTd::cout << "Сума чисел: " << sum << sTd::endl;
```

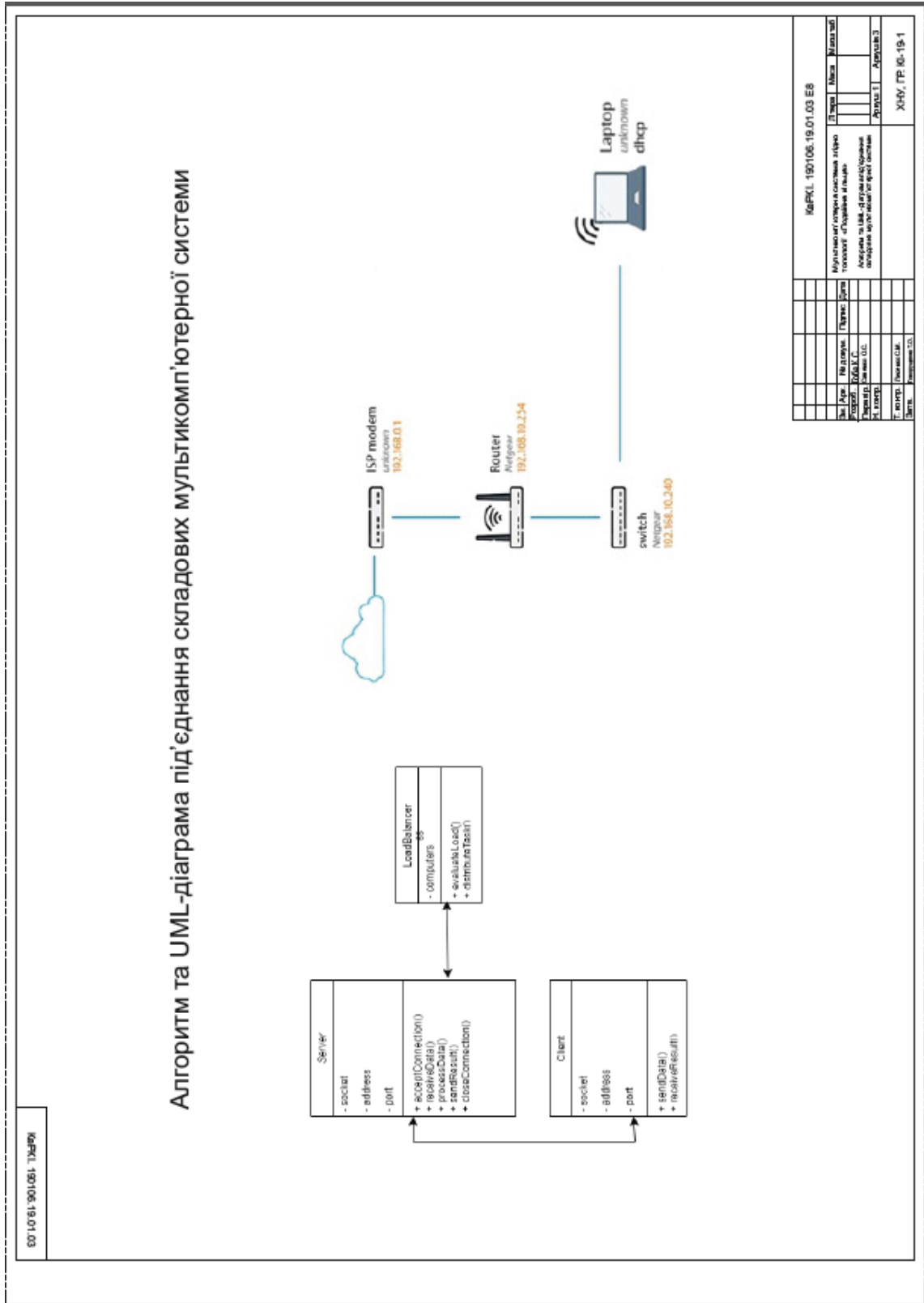
```
// Закриття з'єднання  
close(new_sock);  
close(sock);
```

```
return 0;
```

```
}
```

**Додаток Б**  
(обов'язковий)

Копія креслення «Алгоритм та UML- діаграма під'єднання складових мультимедійної системи»







Ім'я користувача:  
Кафедра КІ

Дата перевірки:  
09.09.2023 09:46:46 EEST

Дата звіту:  
09.09.2023 09:47:40 EEST

ID перевірки:  
1015033116

Тип перевірки:  
Doc vs Internet + Library

ID користувача:  
100005901

Назва документа: Гоба\_Мультикомп'ютерна система згідно топології «подвійне кільце»

Кількість сторінок: 78 Кількість слів: 12156 Кількість символів: 96384 Розмір файлу: 3.10 MB ID файлу: 1015048395

Виявлено модифікації тексту (можуть впливати на відсоток схожості):

7.96%

## Схожість

Найбільша схожість: 2.65% з джерелом з Бібліотеки (ID файлу: 1015044087)

0.79% Джерела з Інтернету

97

Сторінка 80

1.54% Джерела з Бібліотеки

125

Сторінка 81

## 0% Цитат

Не знайдено жодних цитат

Посилання

1

Сторінка 51

0%

## Вилучень

Немає вилучених джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

5355

Підозріле форматування

66  
сторінок

## Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 49%

ID: 114541 Назва: БКР Мультимедійна інтерна система згідно топології «подвійне кільце» Додано в БД: 2023-06-02 Автора: Гоба К.С. Керівник: Савенко О.С. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	81921	708	1547 (2%)	17 (2%)

### Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

## РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Гоца Каріна Сергіївна

Тема: Мультикомп'ютерна система згідно топології «подвійне кільце»

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 69

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є проектування та моделювання мультикомп'ютерної системи згідно топології «подвійне кільце»

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі було проведено детальний аналіз обчислювальних систем, зосередившись на класифікації Флінна та її різних типах паралельної обробки; розглянуто різні складові мультикомп'ютерних систем, такі як комп'ютери, мережеві з'єднання, програмне забезпечення, протоколи комунікації та система керування ресурсами; оглянуто важливість операційних систем у роботі комп'ютерів і їх роль як посередника між користувачем і апаратним забезпеченням; описано різні топології мультикомп'ютерних систем, які визначають структуру підключення комп'ютерів та організацію комунікаційного середовища; проведено аналіз наявних мультикомп'ютерних систем для кращого розуміння їхньої роботи та призначення.

В другому розділі була детально розглянута топологія "кільце" та її похідна, топологія "подвійне кільце". Виявлено відмінності між цими топологіями та проведено порівняння їх структур. Описано різні архітектури мультикомп'ютерних систем з паралельною обробкою даних, зокрема векторно-конвексні системи та обчислювальні системи з розподіленою пам'яттю, зокрема масивно-паралельні комп'ютери. У розділі було наведено апаратне забезпечення, яке було вибрано для

5. Негативні сторони роботи: -

6. Оцінка графічного оформлення та пояснювальної записки роботи:  
Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: \_\_\_\_\_

9. Оцінка дипломної роботи: відмінно

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Мартинюк  
Валерій Валерійович, д.т.н., професор, завідувач кафедри  
автоматизації та комп'ютерно-інтегрованих технологій ХНУ

“ 5 ” 06 2023 р.

 (підпис)

Завідувачу кафедри КПС  
д-р.техн.наук, проф. Говорущенко Т. О.

Гоби Каріни Сергіївни

ІНБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-19-1

### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, 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 плагіат оповіщений(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.

01.06.2023



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ  
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

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

Назва: Мультикомп'ютерна система згідно топології «подвійне кільце»

Автор: Гоба Каріна Сергіївна

Спеціальність: 123 – Компютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Савенко Олег Станіславович, д.т.н, професор

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, виправданий 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, виправданий 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 допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття 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:

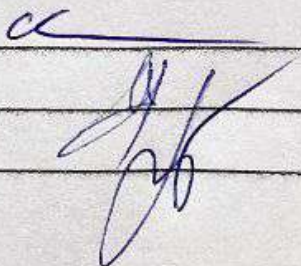
- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-30 джерелами на один фрагмент речення;
- 4) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 7.96% і адресується до 222 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІІС



О. С. Савенко

С. М. Лисенко

Т. О. Говорущенко