

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

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр

Освітній рівень

Мультикомп'ютерна система згідно топології «потовищене дерево»

Назва теми

К6РК1. 190247.19.02.26 ПЗ

Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Виконав: студент IV курсу, група K12-19-2


Підпис

Б. Р. Трет'яков

Ініціали, прізвище


Керівник


Підпис, дата

К. М. Березька

Ініціали, прізвище

Нормоконтролер

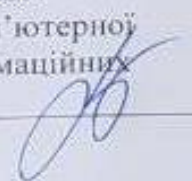

Підпис, дата

С.М. Лисенко

Ініціали, прізвище

До захисту допускаю:

Зав. кафедри комп'ютерної
інженерії та інформаційних
систем


Т.О. Говоруценко

Підпис

Ініціали, прізвище

« 9 » 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 «Комп'ютерна інженерія та програмування»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

 11 " 01 2023 р

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

Трет'якову Богдану Романовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Мультикомп'ютерна система згідно топології «потовщене дерево»

Керівник проекту (роботи) Березька К.М., к.т.н., доцент

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

Затверджена наказом ректора університету від 01.03.2023 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 мультикомп'ютерної системи згідно топології «потовщене дерево»

Моделювання та тестування мультикомп'ютерної системи згідно топології «потовщене дерево»



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

Схема графа топології «потовщене дерево»

Схема топології «потовщене дерево»

Схема результату моделювання

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
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)	Примітки
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	20.02.2023	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.03.2023	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	10.03.2023	виконано
4	Робота над розділом 2 – проектування мультикомп'ютерної системи згідно топології «потоващене дерево»	20.04.2023	виконано
5	Робота над розділом 3 – реалізація та тестування мультикомп'ютерної системи згідно топології «потоващене дерево»	30.04.2023	виконано
6	Оформлення пояснювальної записки згідно вимог	24.05.2023	виконано
7	Попередній захист ВКР	26.05.2023	виконано
8	Захист ВКР на засіданні ЕК	Червень 2023 року	

Студент


 Підпис

 Б.Р. Трет'яков
 Інженер, проєктує

Керівник проекту (роботи)


 Підпис

 К.М. Березька
 Інженер, проєктує

№ р я д к а	Ф о р м а т	Позначення	Найменування	К і л ь н і с т і в	№ с к і	П р и м і т к и
			<u>Текстові документи</u>			
1		КвРКІ 190247.19.02.26 ПЗ	Пояснювальна записка	57		
			<u>Графічні матеріали</u>			
2		КвРКІ 190247.19.02.26 Е8	Схема графа топології «потовищене дерево»	1		
3		КвРКІ 190247.19.02.26 Е8	Схема топології «потовищене дерево»	1		
4		КвРКІ 190247.19.02.26 Е8	Схема результату моделювання	1		

КвРКІ 190247.19.02.26 ВП

Зм	Ари	№ докум	Підпис	Дата
Розробит	Тришак		<i>[Signature]</i>	
Перевір	Тришак		<i>[Signature]</i>	09.06
Навчир	Димов		<i>[Signature]</i>	
Затв	Савицький		<i>[Signature]</i>	09.06

Відомість проєкту

Літера	Аркуш	Аркушів
У	1	57

ХНУ, КІЗ-19-2

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Мультикомп'ютерна система згідно топології «потовщене дерево»»

Автор роботи: Трет'яков Богдан Романович.

Керівник роботи: Березька Катерина Миколаївна.

Пояснювальна записка: 57 с., 19 рис., 3 табл., 4 дод., 40 джерела.

Графічна частина: 6 презентаційних слайдів.

МУЛЬТИКОМП'ЮТЕРНІ СИСТЕМИ, ТОПОЛОГІЯ, ПОТОВЩЕНЕ ДЕРЕВО

Метою роботи є розробка мультикомп'ютерної системи згідно топології «потовщене дерево».

Об'єктом дослідження є Мультикомп'ютерна система згідно топології «потовщене дерево»

Предметом дослідження є формалізований опис та схеми мультикомп'ютерної системи згідно топології «потовщене дерево»


Підпис студента

08.06.2023

Дата

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	4
ВСТУП	5
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ..	7
1.1 Мультикомп'ютерна система	7
1.2 Аналіз предметної області та визначення вимог	18
1.3 Висновки	21
2 ПРОЄКТУВАННЯ МУЛЬТИКОМП'ЮТЕРНОЇ СИСТЕМИ ЗГІДНО	
ТОПОЛОГІЇ «ПОТОВЩЕНЕ ДЕРЕВО»	22
2.1 Топологія «потовщене дерево»	22
2.2 Вибір програмного забезпечення для моделювання мультикомп'ютерної системи	26
2.3 Cisco Packet Tracer	29
2.4 Вибір апаратного забезпечення для моделювання мультикомп'ютерної системи	31
2.5 Висновки	40
3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ МУЛЬТИКОМП'ЮТЕРНОЇ СИСТЕМИ ЗГІДНО	
ТОПОЛОГІЇ «ПОТОВЩЕНЕ ДЕРЕВО»	42
3.1 Створення моделі мультикомп'ютерної системи згідно топології «потовщене дерево»	42
3.2 Реалізація програмного забезпечення	48
3.3 Протокол Cisco Spanning Tree	50
3.4 Тестування розробленої мультикомп'ютерної системи	55
3.4.1 Перевірка з'єднання: пінгування між різними комп'ютерами та пристроями в мережі	56
3.4.2 Тестування пропускної здатності	60
3.4 Висновки	61

КвРК1 190247.19.02.26 ПЗ								
Зм.	Арк.	№докум.	Підпис	Дата	Мультикомп'ютерна система згідно топології «потовщене дерево»	Літера	Аркуш	Аркушів
Виконав		Третяк І.Р.	<i>[Підпис]</i>	07.08		У1		57
Перевір.		Березняк К.М.	<i>[Підпис]</i>		Пояснювальна записка	ХНУ КІ2-19-2		
Н.контр.		Лисенко С.М.	<i>[Підпис]</i>	09.08				
Затвер.		Скобуринська Т.О.	<i>[Підпис]</i>					

ВИСНОВКИ	62
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	63
ДОДАТОК А	69
Лістинг коду проміжного програмного забезпечення.....	69
ДОДАТОК Б	71
Копія креслення «Схема граф топології «потовщене дерево»».....	71
ДОДАТОК В	72
Копія креслення «Схема топології «потовщене дерево»».....	72
ДОДАТОК Г	73
Копія креслення «Схема результати моделювання».....	73

						КвРКІ190247.19.02.26 ПЗ	Арк
							3
Зм.	Арк.	№ докум.	Підпис	Дата			

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

DSM - Distributed Shared Memory

MCS - Мультикомп'ютерна система

STP - Spanning Tree Protocol

BPDU - Bridge Protocol Data Units

DSM - Розподілення поділюваної пам'яті

MPI - Message Passing Interface

					КвРКІ190247.19.02.26 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

ВСТУП

Мультикомп'ютерні системи є важливою тенденцією розвитку обчислювальної техніки в сучасному світі. Їх основна перевага полягає в можливості паралельного виконання завдань, що дає змогу значно пришвидшити роботу комп'ютера. Такі системи складаються з великої кількості процесорів, які об'єднуються в єдину систему. Це дозволяє досягти потужності, що в рази перевищує максимальну швидкодію класичних ЕОМ. Мультикомп'ютерні системи знаходять застосування в багатьох сферах науки, бізнесу, економіки та виробництва. Зокрема, вони дозволяють вирішувати складні завдання, які на однопроцесорних системах вимагають значних затрат часу. Для роботи мультикомп'ютерних систем потрібні спеціальні операційні системи, які часто базуються на відомих ОС, але модифіковані.

Мультикомп'ютерні системи можуть бути різних типів залежно від того, як саме відбувається об'єднання процесорів у єдину систему і які ресурси використовуються. Такі системи називаються паралельними комп'ютерами, мультикомп'ютерами або мультипроцесорними системами. Топологія «потовщене дерево» є однією з найбільш ефективних топологій для створення мультикомп'ютерних систем. У цій топології використовуються вузли, які мають багато виходів, що дозволяє забезпечити швидкий обмін інформацією між комп'ютерами.

У системі з топологією «потовщене дерево» вузли розміщуються на різних рівнях дерева, де більш високі рівні містять менше вузлів, а нижні рівні містять більше вузлів. Зв'язки між вузлами також розбиваються на рівні, причому зв'язки між вузлами на одному рівні мають однакову пропускну здатність. В системі з топологією «потовщене дерево» кожен вузол має два зв'язки до вищих рівнів дерева, тобто зв'язки до двох вузлів на наступному рівні. Це дозволяє створювати кілька незалежних шляхів між будь-якими двома вузлами у системі і забезпечує високу надійність та доступність системи. Одним з найбільш поширених

					КвРКІ190247.19.02.26 ПЗ	Арк. 5
Зм.	Арк.	№ докум.	Підпис	Дата		

застосувань топології «потовщене дерево» є використання її у великих центрах обробки даних та хмарних сервісах.

Метою даної роботи є дослідження мультикомп'ютерної системи згідно топології «потовщене дерево», визначення переваг та недоліків цієї топології у порівнянні з іншими топологіями та аналіз її ефективності. У даній роботі будуть проаналізовані різноманітні архітектури, що можуть використовуватись у системах зі загальною пам'яттю, проведено детальний аналіз існуючих архітектурних рішень. В другому та третьому розділах будуть описані етапи проектування та реалізації такої системи з докладним поясненням вибору та використання апаратно-програмної бази.

					КвРКІ190247.19.02.26 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Мультикомп'ютерна система

Мультикомп'ютерна система (MCS) - це комп'ютерна система, яка складається з кількох комп'ютерів, які працюють разом для вирішення спільної задачі. Кожен комп'ютер в MCS може бути різного типу та містити різні компоненти, такі як процесор, пам'ять, диск і т.д.

MCS дозволяє розподіляти завдання між кількома комп'ютерами, забезпечуючи більшу потужність обчислювальних ресурсів і забезпечуючи можливість виконувати задачі, які були б неможливими для окремого комп'ютера. Кожен комп'ютер у системі може працювати незалежно, але взаємодіяти з іншими комп'ютерами, щоб досягти спільної мети.

Мультикомп'ютерні системи застосовуються в багатьох галузях, включаючи наукові дослідження, фінанси, медицину та інтернет-технології. Наприклад, вони можуть використовуватися для моделювання складних процесів, обробки великих об'ємів даних, управління мережами та багато іншого.

Мультикомп'ютерні системи можуть бути організовані по-різному в залежності від потреб користувача. Наприклад, вони можуть бути зібрані в класичній мережі з локальним підключенням або розташовуватися на різних вузлах великої мережі.

Одним з основних видів мультикомп'ютерних систем є кластери. Кластер - це група комп'ютерів, які об'єднуються в єдину систему і працюють разом. Кластери можуть бути використані для вирішення завдань з великою обчислювальною потужністю, таких як обробка відео, обробка медичних зображень, молекулярне моделювання, технічні розрахунки і т.д.

Інший вид мультикомп'ютерних систем - це розподілені системи. У розподілених системах завдання розподіляються між комп'ютерами, які можуть знаходитися в різних місцях, і виконуватися паралельно. Розподілені системи

					КвРКІ190247.19.02.26 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

широко використовуються в Інтернеті та обробці даних, таких як обробка транзакцій у фінансових системах, аналіз даних в соціальних мережах, обробка даних з супутникових систем і т.д.

Окрім цього, існують і інші типи мультикомп'ютерних систем, такі як GRID системи, які забезпечують спільний доступ до ресурсів великої мережі комп'ютерів, та обчислювальні хмари, які надають користувачам доступ до великої кількості обчислювальних ресурсів через Інтернет.

Узагалі, мультикомп'ютерні системи є потужним інструментом для вирішення складних завдань, що потребують великої обчислювальної потужності, та забезпечують більшу ефективність використання ресурсів за рахунок паралельного виконання завдань на різних комп'ютерах. Використання мультикомп'ютерних систем дозволяє зменшити час виконання завдань, збільшити продуктивність та ефективність обробки даних.

Одним з головних викликів при розробці мультикомп'ютерних систем є синхронізація та координація роботи різних комп'ютерів, що може бути складною задачею. Також необхідно забезпечити безпеку та надійність мультикомп'ютерної системи, щоб уникнути втрати даних або порушення їх цілісності.

У цілому, мультикомп'ютерні системи є важливим інструментом для вирішення складних завдань в областях науки, промисловості та бізнесу. Вони дозволяють отримувати більш точні результати та зменшувати час виконання завдань, що є важливим для досягнення успіху в сучасному світі.

Для мультикомп'ютерних операційних систем структури даних, необхідні для керування системними ресурсами, не мають задовольняти умову їх спільного використання, оскільки їх не потрібно розміщувати у загальній пам'яті. Єдиним можливим видом зв'язку є передача повідомлень (message passing). Мультикомп'ютерні операційні системи здебільшого організовані так як показано на рис.1.1.

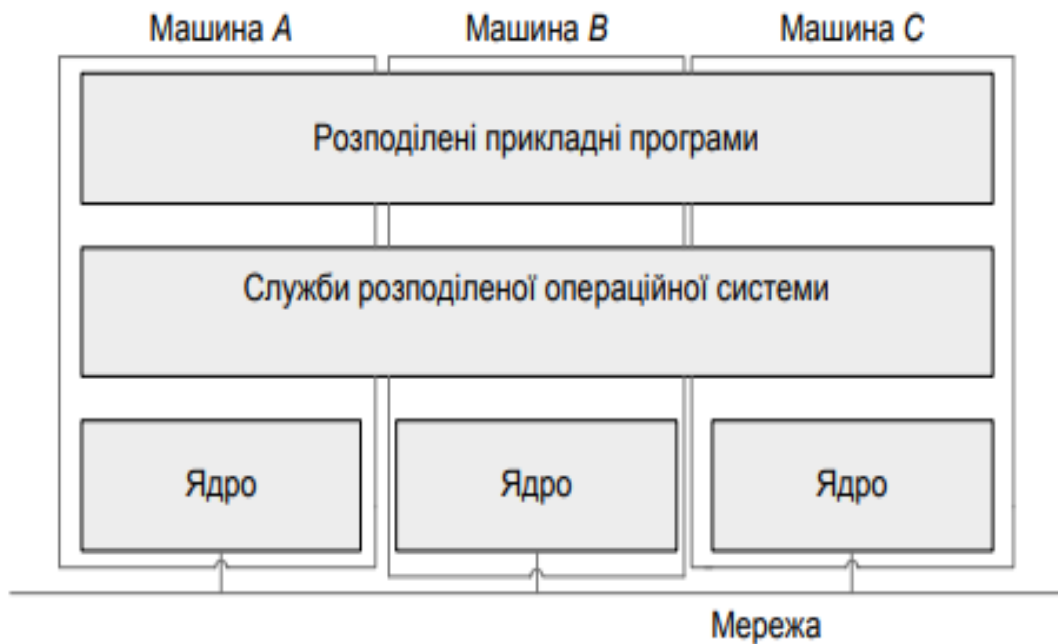


Рисунок 1.1 - Загальна структура мультикомп'ютерних операційних систем

Кожний вузол системи має своє ядро, яке містить модулі для керування локальними ресурсами (пам'яттю, локальним процесором, локальними дисками та ін.), а також окремий модуль для міжпроцесорної взаємодії, тобто відсилання повідомлень на інші вузли і прийому повідомлень від них. Над кожним локальним ядром міститься рівень програмного забезпечення загального призначення, що реалізує операційну систему у вигляді віртуальної машини, яка підтримує паралельну роботу із різними завданнями. Цей рівень може навіть надавати абстракцію мультипроцесорної машини, тобто повну програмну реалізацію пам'яті, яка спільно використовується. Додаткові засоби необхідні, наприклад, для обрання завдань процесорам, маскуванню збоїв апаратури, забезпечення прозорості збереження даних і загального обміну між процесами.

Системи з розподіленою поділюваною пам'яттю – це системи, які використовують віртуальну пам'ять кожного окремого вузла для підтримання загального віртуального адресного простору, що зумовлює використання розподіленої поділюваної пам'яті (Distributed Shared Memory) зі сторінковою організацією. Принцип роботи цієї пам'яті такий: у системі з DSM адресний

простір розділено на сторінки (зазвичай по 4 або 8 Кбайт), які розподілено по всіх процесорах системи. Коли процесор адресується до пам'яті, що не є локальною, відбувається внутрішнє переривання. Операційна система зчитує в локальну пам'ять сторінку, що містить зазначену адресу, і перезапускає виконання інструкції, яка спричинила переривання та яка після цього успішно виконується. Як тимчасове сховище інформації використовується не диск, а віддалена оперативна пам'ять.

Одним з покращень базової системи, що підвищує її продуктивність, є реплікація сторінок, які оголошуються закритими для запису, наприклад, таких, що містять текст програми, константи «тільки для читання» або інші закриті на запис структури. Ще одним покращенням є можливість реплікації також не закритих на запис сторінок, оскільки виконується лише читання, то ніякої різниці між реплікацією закритих і не закритих на запис сторінок немає. Однак, якщо реплікована сторінка змінюється, то необхідно вживати спеціальних заходів для запобігання появі низки несумісних копій. Зазвичай усі копії, крім однієї, перед записуванням вважаються хибними.

Додаткового підвищення продуктивності можна досягти відходом від строгої відповідності між сторінками, які реплікуються, щоб окрема копія тимчасово відрізнялася від інших. Практика показує, що цей підхід насправді може допомогти, але слід обов'язково відслідковувати можливу несумісність. Оскільки основною передумовою розробки DSM була простота програмування, то послаблення відповідності між сторінками не набуває реального застосування. Ще однією проблемою під час розробки ефективних систем DSM є розмір сторінок. Витрати на передачу сторінки мережею передусім визначаються витратами на підготовку до передачі, а не обсягом переданих даних. Відповідно, великий розмір сторінок може зменшити загальну кількість сеансів передачі у разі необхідності доступу до великої кількості послідовних елементів даних.

Натомість, якщо сторінка містить дані двох незалежних процесів, що виконуються на різних процесорах, операційна система буде змушена постійно

					КвРКІ190247.19.02.26 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

пересилати цю сторінку від одного процесора до другого, як показано на рис. 1.2.. Розміщення даних двох незалежних процесів на одній сторінці називають помилковим поділом (false sharing).

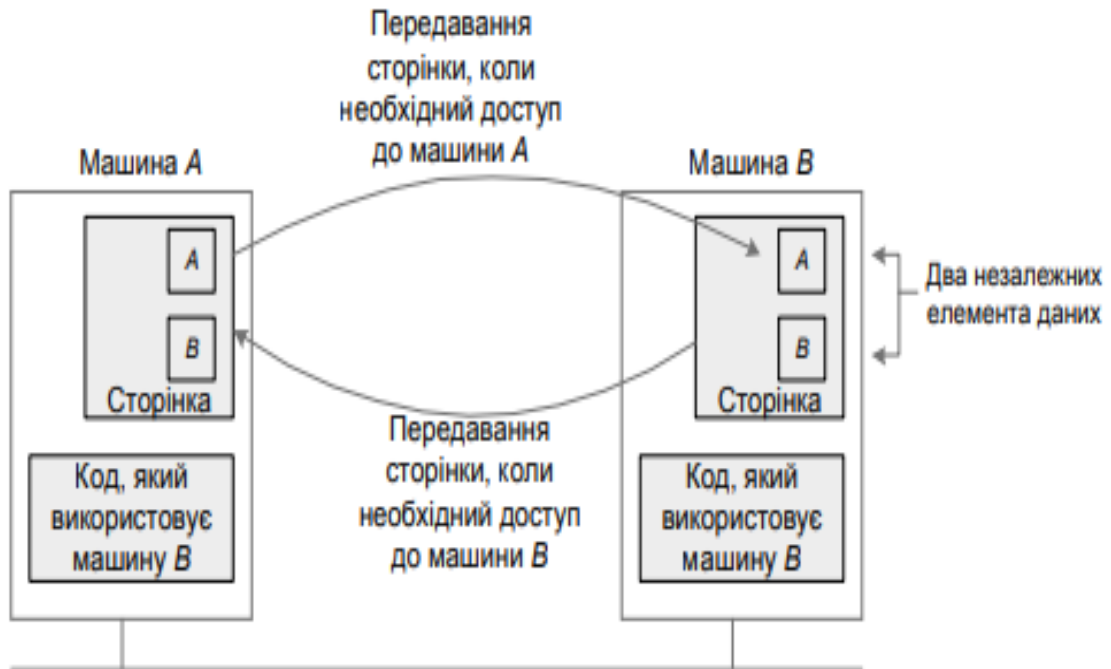


Рисунок 1.2 - Помилковий поділ сторінки двома незалежними процесами

Топологія - це фізична або логічна структура, яка визначає спосіб підключення комп'ютерів, пристроїв та мережевих ресурсів у мережевій інфраструктурі. Вона визначає, як пристрої з'єднані між собою, як вони обмінюються даними та які шляхи сполучення використовуються для передачі інформації. Топологія мережі може бути фізичною або логічною. Фізична топологія визначає фактичне розташування комп'ютерів і пристроїв у мережі та способи їх підключення, такі як з'єднання за допомогою кабелів чи бездротових з'єднань. Логічна топологія визначає шляхи передачі даних між вузлами мережі незалежно від фактичного розташування пристроїв.

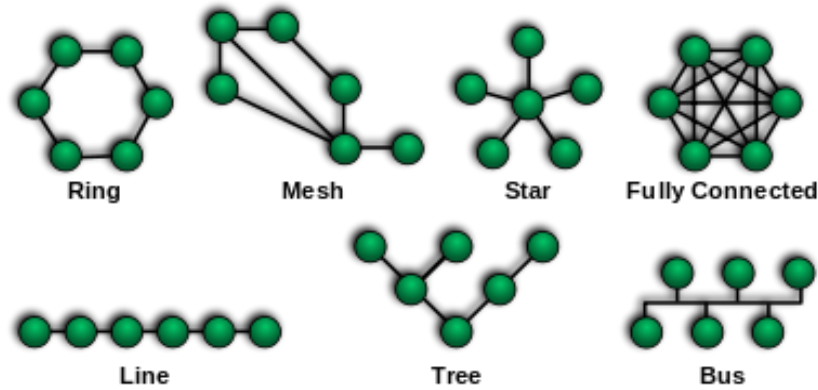


Рисунок 1.3 - Топологія комп'ютерних мереж

На відміну від геометрії, де розглядаються переважно метричні характеристики, такі як довжина, кут і площа, у топології ці характеристики вважаються несуттєвими і натомість вивчаються такі фундаментальніші властивості фігури, як зв'язність (кількість шматків, дірок тощо) або можливість неперервно zdeформувати її до сфери і зворотно (це можливо для поверхні куба, але неможливо для поверхні тора).

Топологія комп'ютерної мережі відображає структуру зв'язків між її основними функціональними елементами. В залежності від компонентів, що розглядаються, розрізняють фізичну і логічну структури локальних мереж. Фізична структура визначає топологію фізичних з'єднань між комп'ютерами. Логічна структура визначає логічну організацію взаємодії комп'ютерів між собою. Доповнюючи одна одну, фізична та логічна структури дають найповніше уявлення про комп'ютерну мережу.

Існує безліч способів з'єднання мережевих пристроїв. Виділяють 3 базових топології:

1. Шина - являє собою загальний кабель (званий шина або магістраль), до якого приєднані всі робочі станції. На кінцях кабелю для запобігання відбиття сигналу знаходяться термінатори. Термінатор, узгоджувач - поглинач енергії (зазвичай резистор) на кінці довгої лінії, опір якого дорівнює хвильовому опору даної лінії.

2.Зірка - це єдина топологія мережі з явно виділеним центром, до якого підключаються всі інші абоненти. Обмін інформацією йде винятково через центральний комп'ютер, на який лягає більше навантаження, тому нічим іншим, крім мережі, він, як правило, займатися не може.

3.Кільце - У комп'ютерній мережі з топологією «кільце» комп'ютери підключаються до кабелю, замкнутого в коло. Тому у кабелі просто не може бути вільного кінця, на який треба поставити термінатор. Сигнали передаються по кільцю в одному напрямі і проходять через кожен комп'ютер. На відміну від пасивної топології «шина», тут кожен комп'ютер виступає в ролі повторювача, підсилюючи сигнали і передаючи їх наступному комп'ютеру. Тому, якщо вийде з ладу один комп'ютер, припиняє функціонувати вся мережа.

Аксіоматика топології побудована на засадах теорії множин, але провідну роль у дослідженнях з сучасної топології відіграють насамперед алгебраїчні і геометричні методи. Об'єктами дослідження топології є топологічні простори, спільне узагальнення таких структур як граф, поверхня у тривимірному просторі і множина Кантора, та відображення між ними. При цьому досліджуються властивості топологічних просторів як в малому (локальні), так і в цілому (глобальні).

Серед різноманітних напрямків топології відзначимо наближену до теорії множин загальну топологію, яка вивчає такі загальні властивості абстрактних топологічних просторів як компактність або зв'язність, та алгебричну топологію, яка намагається описати топологічні простори за допомогою їхніх алгебраїчних інваріантів, наприклад чисел Бетті та фундаментальної групи.

У контексті комп'ютерних мереж топологія відноситься до структури та організації комунікаційних каналів та зв'язків між пристроями. Вона визначає, як вузли мережі підключені один до одного та як вони обмінюються даними

Крім того, топологія мережі визначає, які пристрої є активними в мережі, тобто які з них можуть передавати та приймати дані, а які є пасивними та лише

приймають дані. Вона також впливає на швидкість передачі даних та на стійкість мережі до збоїв та помилок.

У мультикомп'ютерних системах, коли кожен комп'ютер має свою власну пам'ять, може виникати проблема зі синхронізацією даних між комп'ютерами. Цю проблему можна вирішити за допомогою розподіленої пам'яті(Рис1.4), яка дозволяє кожному комп'ютеру доступатись до спільної пам'яті, що знаходиться на іншому комп'ютері. Кеш-пам'ять, з іншого боку, є видом швидкодіючої пам'яті, що знаходиться на процесорі та призначена для зберігання найбільш важливих та найчастіше використовуваних даних.

Кеш-пам'ять є важливою складовою комп'ютерної розподіленої системи (КРС), оскільки дозволяє знизити час доступу до даних, що зберігаються на віддаленому сервері. Кеш-пам'ять є швидкодіючим тимчасовим сховищем для даних, які часто використовуються.

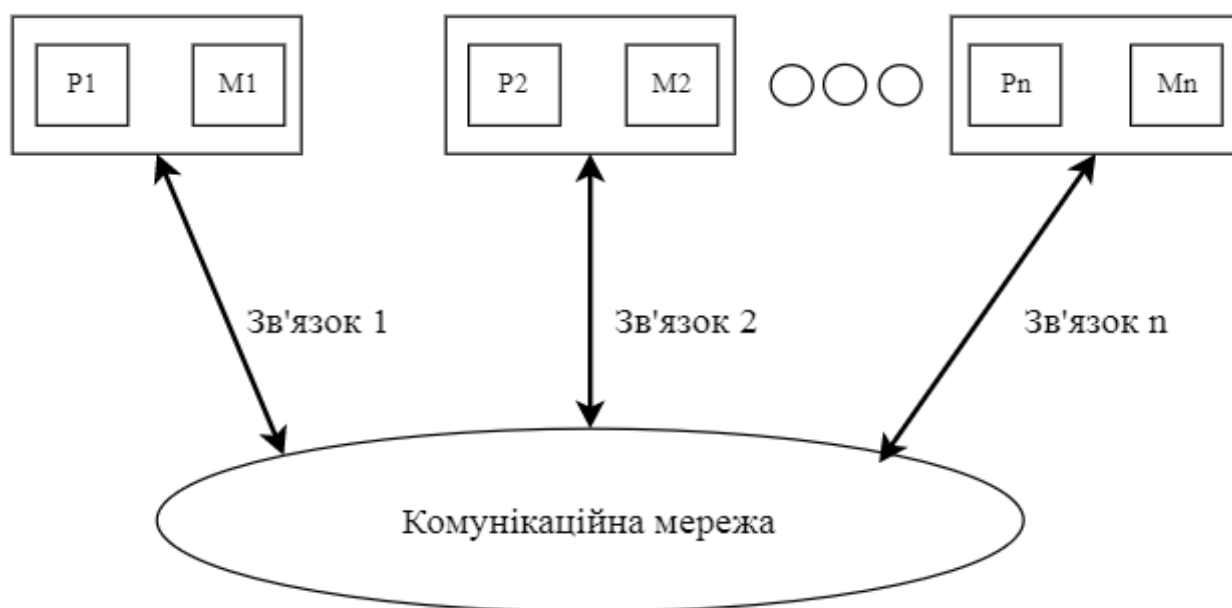


Рисунок 1.4 - Система з розподільною пам'яттю

В КРС, кеш-пам'ять може бути реалізована на різних рівнях: на рівні клієнта, на рівні проміжного програмного забезпечення (наприклад, кеш-пам'ять на сервері проксі), або на рівні сервера. Кеш-пам'ять на рівні клієнта зазвичай

використовується для зберігання копій даних, що розташовані на сервері, для швидкого доступу до них. Кеш-пам'ять на сервері може бути використана для зменшення навантаження на основний сервер, що зберігає дані, шляхом зберігання копій найбільш часто запитуваних даних.

У КРС, які використовують реплікацію даних, кеш-пам'ять може бути також використана для забезпечення швидкого доступу до даних, що зберігаються на резервних серверах. Це може допомогти забезпечити високу доступність системи, оскільки в разі відмови одного сервера, дані можуть бути доступні на іншому сервері в кеш-пам'яті. У КРС з розподіленим сховищем даних (наприклад, Hadoop Distributed File System), кеш-пам'ять може бути використана для зберігання блоків даних, які часто використовуються при обробці даних, що зменшує час доступу до них та забезпечує швидкість обробки.

Кеш-пам'ять в КРС може бути використана для покращення швидкодії та доступності системи, зменшення навантаження на основний сервер, забезпечення більш ефективного використання ресурсів та підвищення загальної продуктивності системи. Однак, використання кеш-пам'яті також може мати певні недоліки. Наприклад, якщо дані в кеш-пам'яті застарілі, то можуть виникнути проблеми з консистентністю даних, тобто дані, які зберігаються в кеш-пам'яті можуть не відповідати даним, які зберігаються на сервері.

Крім того, використання кеш-пам'яті може займати значну кількість пам'яті та збільшувати час розгортання системи. Таким чином, використання кеш-пам'яті в КРС має свої переваги та недоліки, і прийняття рішення про використання кеш-пам'яті повинно базуватися на конкретних вимогах та характеристиках системи

Для ефективного використання кеш-пам'яті в КРС необхідно приділити увагу наступним аспектам:

– Розмір кеш-пам'яті: розмір кеш-пам'яті повинен бути достатнім для зберігання найбільш часто використовуваних даних. При цьому необхідно уникати зайвого використання пам'яті, яке може призвести до збільшення часу розгортання та зниження продуктивності системи;

- політика кешування: необхідно вибрати оптимальну політику кешування, яка відповідає вимогам системи. Наприклад, можна використовувати політику "least recently used" (LRU), яка видаляє найменш використовувані дані з кеш-пам'яті, або політику "time to live" (TTL), яка використовує таймер для визначення того, коли дані стають застарілими та повинні бути видалені з кеш-пам'яті;
- консистентність даних: необхідно забезпечити консистентність даних між кеш-пам'яттю та основним сховищем даних. Для цього можна використовувати різні підходи, наприклад, кешування по часу життя даних, відслідковування змін в основному сховищі даних та оновлення кеш-пам'яті відповідно до цих змін;
- синхронізація кеш-пам'яті: у випадку, якщо кеш-пам'ять використовується на різних серверах, необхідно забезпечити синхронізацію даних між цими серверами. Для цього можна використовувати різні підходи, наприклад, реплікацію даних між серверами або синхронізацію даних за допомогою протоколів розподіленої блокування.

Вузли зазвичай запам'ятовують повідомлення в буферах (тимчасові комірки пам'яті, де повідомлення чекають до того часу, коли вони можуть бути послані або одержані), і виконують посилення-отримання повідомлення під час обробки даних. Паралельна обробка повідомлення і поточного обчислення здійснюється під керуванням операційної системи.

Процесори не розділяють спільну пам'ять і кожен процесор має доступ до його власного адресного простору. Узли в системі передачі повідомлень можуть бути з'єднані різноманітними способами - від спеціалізованих комутаторів до географічно розподілених мереж. Підхід передачі повідомлень є масштабованим до великих пропорцій, тобто число процесорів може бути збільшено без істотного зменшення ефективності їх взаємодії. Для організації зв'язку системи передачі повідомлень використовують статичні мережі, зокрема гіперкубічні мережі, які були досить популярні протягом багатьох років. Двовимірні і тривимірні решітчасті мережі так само широко використовувались в системах передачі

повідомлень. Два важливі чинники повинні розглядатися в проектуванні комунікаційної мережі для систем передачі повідомлень. Це - пропускна здатність і мережний час очікування. Пропускна здатність визначена як кількість бітів інформації, переданих за одиницю часу (біт/сек). Мережний час очікування визначений як час, потрібний для передачі повідомлення. В 1987 році була введена блокова маршрутизація як альтернатива традиційній маршрутизації з проміжним зберіганням для того, щоб скоротити розмір необхідних буферів і щоб зменшити час очікування повідомлення. У блоковій маршрутизації пакет ділиться на менші блоки, які називають блоками керування потоком даних, так, що ці блоки рухаються конвеєрним методом разом з блоком заголовка до вузла призначення. Коли блок заголовка блокується через мережний затор, наступні за ним блоки блокуються також. В основі архітектури будь-якої багатопроекторної комп'ютерної системи лежить здатність до обміну даними між її компонентами.

Це забезпечується комунікаційною мережею (КММ), яка з'єднує між собою вузли комп'ютерної системи за допомогою каналів передачі даних (каналів зв'язку). В ролі вузлів можуть виступати процесори, модулі пам'яті, пристрої введення-виведення, комутатори або декілька перерахованих елементів, об'єднаних у функціональній пристрій. Організація внутрішніх комунікацій комп'ютерної системи називається топологією. Топологію комунікаційної мережі визначає множина вузлів, які об'єднані множиною каналів. Зв'язок між вузлами зазвичай реалізується по двоточковій схемі (point-to-point). Будь-які два вузли, зв'язані каналом зв'язку, називають суміжними вузлами або сусідами. Кожен канал з'єднує один вузол-джерело з одним вузлом-приймачем. Канал характеризується кількістю сигнальних ліній, частотою або швидкістю передачі бітів по кожній сигнальній лінії, затримкою - часом пересилання біта з одного вузла до іншого.

Для більшості каналів затримка знаходиться в прямій залежності від фізичної довжини лінії зв'язку та швидкості, розповсюдження сигналу. Вузол у мережі може бути термінальним, тобто джерелом або приймачем даних, комутатором, що пересилає інформацію з вхідного порту на вихідний, або суміщати обидві ролі. У

мережах із прямими зв'язками кожен вузол одночасно є як термінальним вузлом, так і комутатором, і повідомлення пересилаються між термінальними вузлами безпосередньо. У мережах з непрямыми зв'язками вузол може бути або термінальним, або комутатором, але не одночасно, тому повідомлення передаються опосередковано, за допомогою виділених комутуючих вузлів. Існують також такі топології, які не можна однозначно зарахувати ні до прямих, ні до непрямих. Будь-яку пряму КММ можна зобразити у вигляді непрямої, розділивши кожен вузол на двотермінальний вузол і вузол комутації.

Сучасні прямі мережі реалізуються саме таким чином - комутатор відділяється від термінального вузла і поміщається у виділений маршрутизатор. Основна перевага прямих КММ полягає в тому, що комутатор може використовувати ресурси термінальної частини свого вузла. Це стає істотним, якщо врахувати, що, як правило, останній включає комп'ютер або процесор.

1.2 Аналіз предметної області та визначення вимог

Аналіз предметної області та визначення вимог для мультикомп'ютерної системи згідно топології "потовщене дерево" включає розуміння основних цілей, вимог та обмежень, що впливають на систему. Основні етапи такого аналізу та визначення вимог включають наступні кроки:

1. Функціональних цілі мультикомп'ютерної системи

Основна функціональна ціль мультикомп'ютерної системи з топологією "потовщене дерево" полягає в забезпеченні паралельної обробки завдань. Це досягається шляхом розділення завдань на підзадачі і обробки їх паралельно на різних комп'ютерах у системі. Такий підхід призводить до покращення продуктивності та зменшення часу виконання завдань, оскільки декілька комп'ютерів працюють одночасно над різними частинами завдання.

Потовщене дерево забезпечує ієрархічну організацію комп'ютерів, де кожен комп'ютер може мати декілька підключених до нього підкомп'ютерів. Це дає можливість створювати багат шарові структури для обробки завдань. Комунікація

					КвРКІ190247.19.02.26 ПЗ	Арк. 18
Зм.	Арк.	№ докум.	Підпис	Дата		

між комп'ютерами здійснюється за допомогою комунікаційних каналів, які забезпечують передачу даних між вузлами системи.

Така система може мати такі функціональні переваги:

- збільшена швидкодія та продуктивність завдяки паралельній обробці;
- знижений час виконання завдань шляхом розділення їх на підзадачі;
- масштабованість системи шляхом додавання нових комп'ютерів або підкомп'ютерів;
- забезпечення високої доступності та надійності завдяки резервуванню комп'ютерів та комунікаційних каналів.

2. Розуміння потреб та очікувань від системи.

Розуміння потреб та очікувань від мультикомп'ютерної системи згідно з топологією "потовщене дерево" є ключовим аспектом при проектуванні та розробці такої системи. Перш за все, важливо визначити обсяг обробки даних, які система буде опрацьовувати. Це означає розуміння кількості та складності завдань, які система повинна виконувати, а також обсягу даних, які потрібно обробляти. Це допомагає забезпечити, що система має достатній обчислювальний потенціал і ресурси для виконання цих завдань ефективно та вчасно.

Далі, розуміння потреб у паралельній обробці є важливим аспектом. Топологія "потовщене дерево" дозволяє розділити завдання на підзавдання і обробляти їх паралельно на різних комп'ютерах. Це покращує продуктивність системи і знижує час виконання завдань. Розуміння, які завдання можуть бути оброблені паралельно і які мають залежності між собою, допомагає визначити оптимальну організацію системи. Масштабованість є ще одним важливим аспектом. Розуміння, як система може збільшувати свою потужність та розмір, додаючи нові комп'ютери або підкомп'ютери, дозволяє планувати майбутні розширення. Забезпечення можливості додавання нових вузлів і збільшення обчислювальних ресурсів допомагає забезпечити, що система може масштабуватися залежно від зростаючих потреб.

Висока доступність є ще одним важливим вимогом. Забезпечення надійності та доступності системи шляхом резервування комп'ютерів та комунікаційних каналів допомагає зменшити вплив відмов окремих компонентів на роботу системи в цілому. Розуміння критично важливих компонентів системи та розробка механізмів резервування і відновлення допомагає забезпечити неперервну роботу системи.

Швидкодія та продуктивність є іншими ключовими функціональними цілями. Забезпечення ефективної обробки завдань та оптимального використання ресурсів системи допомагає досягти більшої продуктивності. Це може включати оптимізацію алгоритмів обробки даних, ефективну передачу даних між комп'ютерами та ефективне розподілення завдань між вузлами системи.

Нарешті, простота управління та моніторингу є важливим аспектом. Забезпечення зручного інтерфейсу для управління системою та моніторингу її стану допомагає адміністраторам ефективно керувати системою, виявляти та вирішувати проблеми. Детальне розуміння потреб управління, контролю та моніторингу допомагає розробити зручні інструменти та інтерфейси для адміністрування системи.

Процес аналізу потреб та очікувань повинен включати взаємодію з зацікавленими сторонами, проведення досліджень і консультацій, а також збір та аналіз даних для розуміння контексту і вимог предметної області.

3. Визначення функціональних вимог, встановлення основних функцій, які система повинна виконувати. важлива функція полягає в керуванні ресурсами системи. Система повинна мати механізми для ефективного розподілу ресурсів, таких як обчислювальна потужність, пам'ять, мережеві ресурси та інші, між різними комп'ютерами. Це допомагає забезпечити оптимальне використання ресурсів і підтримувати ефективну роботу системи.

Додатковою функцією може бути забезпечення високої доступності системи. Це означає наявність механізмів резервування та відновлення, щоб забезпечити безперебійну роботу системи навіть у разі відмов окремих компонентів.

					КвРКІ190247.19.02.26 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

Також можуть бути встановлені функції для керування та моніторингу системи, що дозволяють адміністраторам ефективно керувати ресурсами, виявляти та вирішувати проблеми та забезпечувати безперебійну роботу системи.

1.3 Висновки

Під час дослідження предметної області мультикомп'ютерних мереж було з'ясовано багато важливих аспектів та особливостей, пов'язаних з цією технологією, зокрема було встановлено, що така система є потужним інструментом для обробки великих обсягів даних та виконання складних обчислювальних завдань. Вона дозволяє розподіляти завдання між комп'ютерами, що призводить до покращення продуктивності та ефективності системи. Мультикомп'ютерні системи мають широкі застосування в різних галузях, включаючи наукові дослідження, фінансовий сектор та виробництво. Продовження досліджень у цій області є важливим для вдосконалення технологій та розширення можливостей мультикомп'ютерних систем.

					КвРКІ190247.19.02.26 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ МУЛЬТИКОМП'ЮТЕРНОЇ СИСТЕМИ ЗГІДНО ТОПОЛОГІЇ «ПОТОВЩЕНЕ ДЕРЕВО»

2.1 Топологія «потовщене дерево»

Потовщене дерево (Fat tree) - це топологічна структура, яка зазвичай використовується для побудови мережі з декількома рівнями підключення. Вона складається з кореневого вузла, кількох рівнів вузлів-маршрутизаторів та кінцевих вузлів. На першому рівні вузлів розміщуються кореневі маршрутизатори. На другому рівні вузлів розміщуються маршрутизатори, які підключені до корневих маршрутизаторів та виконують функцію дистрибуції трафіку в мережі. На третьому рівні вузлів розміщуються маршрутизатори, які забезпечують підключення до попереднього рівня та підключення кінцевих вузлів. На кожному рівні може бути будь-яка кількість вузлів. Кінцеві вузли можуть бути комп'ютерами, пристроями з підтримкою мережевого підключення, або іншими пристроями, які не є маршрутизаторами. Вони підключаються до останнього рівня маршрутизаторів.

Приклад топологічної структури «потовщене дерево» зображено на рис.2.1

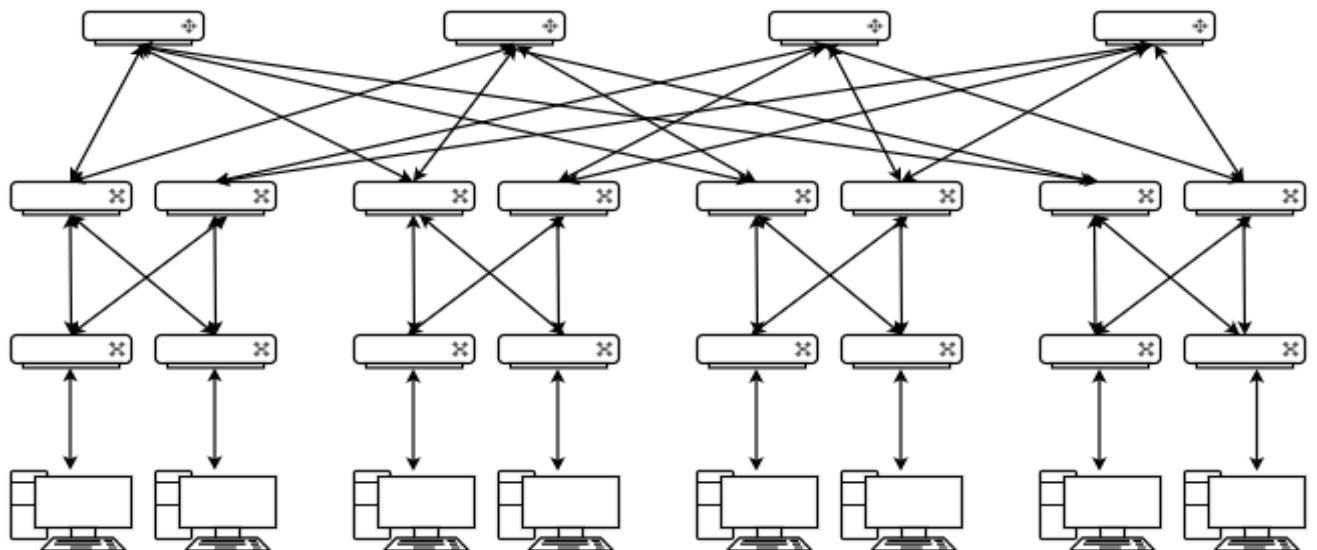


Рисунок 2.1 - Топологія «потовщене дерево»

Ключовою характеристикою топології Fat Tree є використання надлишкових каналів на кожному рівні, що підвищує надійність і відмовостійкість мережі. Вона також забезпечує безліч шляхів між будь-якою парою вузлів, що дозволяє балансувати навантаження і ефективно використовувати мережеві ресурси.

Однією з особливостей топології Fat Tree є використання багатошляхової маршрутизації з рівною вартістю (ECMP). ECMP дозволяє рівномірно розподіляти трафік між кількома шляхами, максимізуючи пропускну здатність мережі та мінімізуючи перевантаження.

Ще однією перевагою топології Fat Tree є її масштабованість. У міру зростання дата-центрів до мережі можна додавати додаткові комутатори, не порушуючи існуючих з'єднань. Така масштабованість робить її придатною для великомасштабних середовищ центрів обробки даних.

В цілому, топологія Fat Tree забезпечує високу пропускну здатність, відмовостійкість, масштабованість і ефективну маршрутизацію в мережах центрів обробки даних, що робить її популярним вибором для обробки зростаючих вимог сучасних додатків центрів обробки даних.

Мультикомп'ютерна система з топологією «потовщене дерево» (Fat Tree) є однією з найпоширеніших топологій для побудови великих дата-центрів, в яких потрібно забезпечити високу пропускну здатність та надійність мережі.

Топологія потовщеного дерева базується на ідеї розбиття мережі на підмережі різної пропускну здатності. Для цього використовуються комутатори різного рівня, кожен з яких пов'язаний з комутаторами вищого та нижчого рівня. Комутатори верхнього рівня мають більшу пропускну здатність, ніж комутатори нижнього рівня, тому вони використовуються для забезпечення більш великих потоків даних між підмережами.

Для забезпечення високої надійності мережі, топологія потовщеного дерева містить декілька шляхів між кожною парою комутаторів. Це забезпечує резервне керування у випадку відмови одного з каналів.

На додаток до своїх основних характеристик, топологія Fat Tree пропонує кілька переваг:

– висока пропускна здатність: топологія "потовщене дерево" забезпечує високу сукупну пропускну здатність завдяки використанню декількох паралельних шляхів між комутаторами. Це забезпечує ефективну передачу даних і знижує ймовірність перевантаження мережі;

– низька затримка: Завдяки наявності декількох шляхів між вузлами, топологія "потовщене дерево" мінімізує кількість переходів, необхідних для зв'язку. Це призводить до низької затримки, що має вирішальне значення для додатків, що працюють в режимі реального часу, і передачі великих обсягів даних;

– надмірність і відмовостійкість: Надлишкові зв'язки в топології Fat Tree гарантують, що навіть у разі виходу з ладу каналу або комутатора будуть доступні альтернативні шляхи. Це підвищує надійність і відмовостійкість мережі, оскільки трафік може бути перенаправлений без перебоїв;

– масштабованість: Топологія Fat Tree має високу масштабованість і може вмістити велику кількість пристроїв. У міру зростання потреб центру обробки даних в пропускній здатності до мережі можна додавати додаткові комутатори і канали зв'язку, не впливаючи на загальну топологію і продуктивність;

– простота обслуговування та модернізації: Модульний характер топології Fat Tree спрощує обслуговування і модернізацію. Окремі комутатори або канали можуть бути замінені або модернізовані без впливу на всю мережу, що зводить до мінімуму час простою і перебої в роботі;

– збалансоване навантаження на мережу: топологія Fat Tree рівномірно розподіляє мережевий трафік по декількох каналах, запобігаючи перевантаженню певних каналів. Ця функція балансування навантаження оптимізує використання ресурсів і забезпечує ефективну роботу мережі;

– міркування щодо вартості: Впровадження топології "товсте дерево" може вимагати значних інвестицій у комутатори, кабелі та мережеву інфраструктуру. Однак переваги високої пропускної здатності, відмовостійкості та

					КвРКІ190247.19.02.26 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

масштабованості виправдовують витрати для організацій з високими вимогами до центрів обробки даних.

Варто зазначити, що проектування і реалізація топології Fat Tree може бути складним процесом, що вимагає ретельного планування і врахування таких факторів, як пропускна здатність комутаторів, пропускна здатність каналів зв'язку і моделі трафіку. Мережеві протоколи та алгоритми маршрутизації також повинні бути налаштовані належним чином, щоб повною мірою скористатися перевагами топології.

Створення мультикомп'ютерної системи з топологією "потовщене дерево" може викликати деякі проблеми. Ось деякі з них:

- підтримка топології: Реалізація топології "потовщене дерево" може бути досить складною. Необхідно встановити правильне з'єднання між вузлами і забезпечити належне керування даними між ними;
- синхронізація: Керування даними в системі може бути складним завданням. Синхронізація даних між вузлами системи потребує відповідного механізму керування. Це може призвести до затримок, які можуть вплинути на продуктивність системи;
- розділена архітектура: Топологія "потовщене дерево" передбачає розділення вузлів на групи. Це може призвести до проблем з керуванням даними, оскільки вони мають бути передані між різними групами. Це може призвести до неефективного використання ресурсів і зниження продуктивності системи;
- потреба відповідного програмного забезпечення: Для забезпечення роботи мультикомп'ютерної системи з топологією "потовщене дерево" необхідне відповідне програмне забезпечення. Це може призвести до додаткових витрат на придбання і підтримку необхідного програмного забезпечення;
- слабка масштабованість: Топологія "потовщене дерево" може бути обмеженою в масштабуванні. Якщо потрібно додати новий вузол, це може призвести до перепланування всієї топології. Це може бути часом затратно та складно.

					КвРКІ190247.19.02.26 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

2.2 Вибір програмного забезпечення для моделювання мультикомп'ютерної системи

Загалом, при виборі програмного забезпечення для мультикомп'ютерної мережі за топологією потовщеного дерева важливо враховувати низку факторів, таких як:

- мережева підтримка - програмне забезпечення повинно мати необхідні функціональні можливості для моделювання топології потовщеного дерева. Воно повинно підтримувати створення вузлів, комутаторів, маршрутизаторів та з'єднань між ними, а також налаштування маршрутизації та комутації;

- програмне забезпечення повинно мати можливості для моделювання ресурсів комп'ютерної системи, таких як процесори, пам'ять, сховище тощо. Це дозволяє визначати параметри ресурсів та виконувати моделювання їх використання в мультикомп'ютерній системі;

- повинно надавати засоби для аналізу продуктивності мультикомп'ютерної системи. Це може включати збір та аналіз даних про завантаження ресурсів, час виконання завдань, пропускну здатність мережі тощо;

- масштабованість: здатність працювати з великою кількістю вузлів та пристроїв, а також забезпечувати необхідну пропускну здатність;

- реалістичність: важливо, щоб програмне забезпечення дозволяло моделювати реалістичні параметри мережі, такі як швидкість передачі даних, затримки, втрати пакетів та інші фактори, що впливають на роботу мережі.

Виходячи з цих факторів можна виділити кілька програмних забезпечень які можуть бути використані у моделюванні мультикомп'ютерної системи (таблиця 2.1)

Серед заданих програм найбільш підходящою за метою дослідження є програмне забезпечення Cisco Packet Tracer, він надає можливість моделювати, налаштовувати та візуалізувати різні аспекти мережевого середовища.

					КвРКІ190247.19.02.26 ПЗ	Арк. 26
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.1 - Програмні забезпечення та їх переваги та недоліки

Назва платформи	Переваги	Недоліки
GNS3	<ul style="list-style-type: none"> - Підтримка віртуальних мережевих пристроїв та віртуалізованого обладнання. - Зручний інтерфейс для створення та налаштування мережевих сценаріїв. - Широкий спектр можливостей віртуалізації та розгортання. 	<ul style="list-style-type: none"> - Вимагає наявності реального обладнання або образів оперативної системи. - Вимагає більшого рівня технічної експертизи для налаштування та використання.
EVE-NG	<ul style="list-style-type: none"> - Підтримка віртуальних мережевих пристроїв та віртуалізованого обладнання. - Зручний інтерфейс для створення та налаштування мережевих сценаріїв. - Широкий спектр можливостей віртуалізації та розгортання. 	<ul style="list-style-type: none"> - Вимагає потужних обчислювальних ресурсів для ефективної роботи. - Вартість платної версії може бути високою для окремих користувачів.

Кінець Таблиці 2.1 - Програмні забезпечення та їх переваги та недоліки

OMNeT++	<ul style="list-style-type: none"> - Підтримка моделювання розподілених систем та протоколів. - Можливість програмування на різних мовах, таких як C++ та NED. - Багатофункціональність та гнучкість в налаштуванні моделей мережі. 	<ul style="list-style-type: none"> - Вимагає більшої технічної експертизи та знань програмування. - Важкощі в налагодженні та аналізі великих моделей мереж.
Cisco Packet Tracer	<ul style="list-style-type: none"> - Легкий у використанні та інтуїтивний інтерфейс користувача. - Вбудована підтримка пристроїв та протоколів Cisco. - Можливість візуалізації та налаштування мережевих компонентів. 	<ul style="list-style-type: none"> - Обмежений функціонал порівняно з іншими програмними засобами. - Підтримує переважно обладнання та протоколи Cisco.
NS-3	<ul style="list-style-type: none"> - Велика кількість вбудованих моделей та протоколів. - Потужні можливості для налаштування та розширення. 	<ul style="list-style-type: none"> - Вимагає високого рівня технічної експертизи та знань програмування. - Може бути складним у використанні для новачків.

2.3 Cisco Packet Tracer

Cisco Packet Tracer - це інструмент для моделювання мереж, розроблений компанією Cisco Systems. Він дозволяє користувачам проектувати, налаштовувати та діагностувати мережеву інфраструктуру, використовуючи віртуальні пристрої та можливості мережевого моделювання. Packet Tracer надає візуальний інтерфейс, за допомогою якого користувачі можуть створювати мережеві топології, перетягуючи на робочу область такі пристрої, як маршрутизатори, комутатори, ПК і сервери. Ці віртуальні пристрої можуть бути з'єднані між собою за допомогою різних мережевих протоколів, що дозволяє користувачам моделювати і тестувати різні мережеві конфігурації.

Однією з ключових особливостей Packet Tracer є його здатність імітувати поведінку мережевих пристроїв і протоколів. Користувачі можуть налаштовувати параметри пристроїв, такі як IP-адреси, протоколи маршрутизації, VLAN і списки контролю доступу, і спостерігати, як ці конфігурації впливають на роботу мережі. Механізм імітації в Packet Tracer точно імітує поведінку мережевих пристроїв Cisco, забезпечуючи реалістичне середовище для проектування мережі та пошуку і усунення несправностей. Packet Tracer також підтримує симуляцію мережевих служб, таких як DHCP, DNS, FTP і HTTP. Це дозволяє користувачам тестувати функціональність мережевих служб і розуміти, як вони взаємодіють з мережевою інфраструктурою. Крім того, Packet Tracer пропонує ряд функцій для спільної роботи, що дозволяє користувачам спільно працювати над мережевими проектами. Користувачі можуть ділитися своїми мережевими топологіями, конфігураціями і результатами моделювання з іншими, полегшуючи командну роботу і обмін знаннями.

Cisco Packet Tracer широко використовується в освітніх установах, особливо в мережевих курсах і сертифікаційних програмах. Він пропонує практичний підхід до навчання, дозволяючи студентам проектувати і моделювати мережі, тестувати конфігурації і усувати несправності в безпечному і контрольованому середовищі.

					КвРКІ190247.19.02.26 ПЗ	Арк. 29
Зм.	Арк.	№ докум.	Підпис	Дата		

За допомогою Packet Tracer студенти можуть отримати практичний досвід роботи з мережевими концепціями і протоколами, такими як IP-адресація, маршрутизація, комутація, VLAN і безпека. Вони можуть створювати складні мережеві топології, налаштовувати пристрої та спостерігати, як дані проходять через мережу. Цей інтерактивний досвід навчання допомагає студентам розвинути глибше розуміння принципів роботи мережі та покращує їхні навички вирішення проблем.

Packet Tracer також надає ряд вбудованих інструментів і ресурсів для підтримки навчання. До них відносяться велика бібліотека мережевих пристроїв і модулів, вичерпна довідкова документація, а також доступ до широкого спектру мережевих протоколів і технологій. Студенти можуть експериментувати з різними функціями та конфігураціями, досліджувати вплив змін у режимі реального часу та аналізувати поведінку мережі.

Крім того, Packet Tracer пропонує можливості моделювання, які дозволяють студентам тестувати мережеві проекти і перевіряти їх функціональність. Вони можуть оцінити продуктивність мережі, виявити потенційні вузькі місця і оптимізувати проект відповідно до конкретних вимог. Цей ітеративний процес проектування, моделювання та вдосконалення допомагає студентам розвивати навички критичного мислення і готує їх до реальних сценаріїв розгортання мережі.

Cisco Packet Tracer має зручний та інтуїтивно зрозумілий інтерфейс(рис 2.2), який дозволяє користувачам легко створювати та налаштовувати мережеві сценарії. У програмі доступні різні інструменти, такі як панель пристроїв, панель засобів, панель керування, які дозволяють створювати, редагувати та керувати пристроями та мережами.

Ще однією перевагою Packet Tracer є його сумісність з іншими мережевими інструментами та пристроями Cisco. Можна легко інтегрувати свої змодельовані мережі з фізичними пристроями, такими як маршрутизатори та комутатори, для більш комплексного навчання.

комутаторів для мультикомп'ютерних систем згідно топології потовщене дерево - це підтримка Spanning Tree Protocol. Цей протокол дозволяє уникнути петель в мережі та забезпечують резервне шляхування.

Також важливо враховувати сумісність комутаторів з іншими пристроями в мережі, такими як маршрутизатори і сервери. Комутатори повинні мати достатню швидкість передачі даних для підтримки потреб мережі. Крім того, необхідно враховувати тип комутатора. Існують різні типи комутаторів, включаючи некеровані, керовані та комутатори рівня . Некерований комутатор - це тип мережевого пристрою, який дозволяє декільком пристроям підключатися до локальної мережі (LAN) і взаємодіяти один з одним. На відміну від керованих комутаторів, некеровані комутатори, як правило, є пристроями типу "підключи і працюй", які потребують мінімального налаштування та конфігурації.

Некеровані комутатори призначені для забезпечення базового зв'язку між пристроями в мережі. Вони не мають розширених функцій і можливостей керованих комутаторів, таких як підтримка VLAN, налаштування якості обслуговування (QoS) або інструменти моніторингу та управління мережею. Це робить їх ідеальним вибором для невеликих, простих мереж, де вартість і простота використання є основними міркуваннями.

Керовані комутатори - це тип мережевих пристроїв, які надають розширені функції та можливості для управління та оптимізації локальних мереж (LAN). На відміну від некерованих, керовані комутатори дозволяють краще контролювати мережевий трафік і можуть бути налаштовані на підтримку широкого спектру мережевих додатків і протоколів. Комутатори 3-го рівня призначені для забезпечення функцій комутації та маршрутизації в одному пристрої. Це робить їх універсальним і економічно ефективним рішенням для багатьох типів мереж.

Як і керовані комутатори, комутатори рівня 3 використовують MAC-адреси для переадресації трафіку в межах однієї локальної мережі. Однак вони також мають можливість маршрутизувати трафік між різними підмережами або VLAN

всередині мережі, що робить їх корисним інструментом для сегментації та оптимізації мережевого трафіку.

Комутатори 3-го рівня працюють на мережевому рівні (Layer 3) моделі OSI, що означає, що вони можуть приймати рішення про маршрутизацію на основі мережевих адрес, а не тільки фізичних. Це дозволяє їм маршрутизувати трафік більш ефективно та інтелектуально, ніж традиційні комутатори.

На додаток до функцій маршрутизації, комутатори рівня 3 можуть також включати розширені функції, такі як налаштування якості обслуговування (QoS), списки контролю доступу і підтримка віртуальних локальних мереж (VLAN). Ці функції можуть допомогти підвищити продуктивність, безпеку і масштабованість мережі.

В цілому, комутатори 3-го рівня є цінним інструментом для багатьох типів мереж, пропонуючи поєднання функцій комутації і маршрутизації в одному пристрої.

При використанні комутатора в мультикомп'ютерної системи з топологією "потовщене дерево", потрібно звертати увагу на такі дані :

- пропускна здатність: Комутатор повинен мати достатню пропускну здатність для обробки трафіку від усіх комп'ютерів, що підключені до системи;
- комутатор повинен мати достатню кількість портів для підключення всіх комп'ютерів в системі. Топологія потовщеного дерева може використовувати головні гілки, підгілки та листки, тому кількість портів залежатиме від загальної кількості комп'ютерів;
- у системі використовуються віртуальні локальні мережі (VLAN), тому комутатор повинен підтримувати цю функцію. Вона дозволяє групувати комп'ютери в окремі віртуальні мережі для поліпшення безпеки та ефективності мережі;
- комутатор повинен мати можливість виконувати базові функції маршрутизації, які дозволяють пересилати пакети даних між різними гілками або підгілками потовщеного дерева.

З урахуванням вимог, найкращою моделю комутатора буде Cisco Catalyst 2960X(рис 2.3) . Цей комутатор має декілька портів Fast Ethernet або Gigabit Ethernet та може бути використаний для розгалуження та створення потовщеного дерева. Він має розширені функції безпеки, високу продуктивність та широку пропускну здатність для обробки інтенсивного трафіку мережі.



Рисунок 2.3 - Комутатор Cisco Catalyst 2960

Таблиця 2.2 - Характеристика комутатора Cisco Catalyst 2960

Число портів 10/100 Mbps	24
Число портів 10/100/1000 Mbps	2
Максимум Vlan	64
Пропускна здатність, Гбіт/с	8,8
Максимум MAC-адресів	8000
Об'єм ОЗП/flash-пам'яті, Мб	64/32
Споживана потужність (Ватт)	30W

2. Маршрутизатори - це мережеві пристрої, які використовуються для пересилання даних між комп'ютерами у комп'ютерних мережах. Вони виконують функцію пересилання пакетів даних з одного вузла мережі до іншого, використовуючи оптимальний шлях. Основна функція маршрутизатора - вибрати оптимальний шлях для пересилання пакетів даних з вихідного вузла до призначеного вузла. Він приймає пакети даних, аналізує їх заголовки (які містять

адреси призначення та джерела), і використовує таблиці маршрутизації для визначення найкращого шляху для передачі пакета. Маршрутизатор також виконує інші функції, такі як фільтрація пакетів, перетворення протоколів, забезпечення безпеки мережі, розділення мереж на підмережі (сегменти), керування пропускнуою здатністю мережі і багато іншого. У більших мережах можуть використовуватись декілька маршрутизаторів, які працюють разом для забезпечення ефективної передачі даних по всій мережі. Вони співпрацюють між собою, обмінюючися інформацією про стан мережі і використовуючи різні алгоритми маршрутизації для оптимального керування трафіком. Маршрутизатори використовуються в Інтернеті і багатьох локальних мережах для забезпечення передачі даних між різними вузлами мережі та маршрутизації трафіку по найкоротшому шляху.

Маршрутизатори також можуть виконувати функцію Network Address Translation (NAT), що дозволяє підключеним до них пристроям використовувати одну зовнішню IP-адресу для комунікації з Інтернетом. Це особливо корисно в ситуаціях, коли у вас є декілька пристроїв у локальній мережі, але доступна лише одна зовнішня IP-адреса. У більш складних мережах маршрутизатори можуть бути розташовані на різних рівнях, утворюючи ієрархію маршрутизаторів. Це дозволяє розділити мережу на підмережі і ефективно керувати трафіком. Наприклад, великі організації можуть мати центральний маршрутизатор, який забезпечує з'єднання з Інтернетом, і ряд регіональних маршрутизаторів, що обслуговують окремі відділення або підрозділи. Для мультимедійної системи згідно з топологією "потовщене дерево" маршрутизатору потрібні наступні умови:

- маршрутизатор повинен підтримувати протоколи маршрутизації, які використовуються для визначення найкоротшого шляху між комп'ютерами в мережі;
- в мультимедійних системах потовщеного дерева часто використовуються віртуальні локальні мережі (VLAN), щоб групувати комп'ютери за функціональністю або відділами. Маршрутизатор повинен підтримувати VLAN

і мати можливість налаштувати VLAN-и та виконувати маршрутизацію між ними;

– мультиплексування портів: Маршрутизатор повинен мати достатню кількість портів для підключення комп'ютерів в мережі. В нашому випадку кількість портів у маршрутизаторі повинна бути не менше 16;

– у мультикомп'ютерних системах потовщеного дерева важливо мати можливість резервування пропускної здатності в разі відмови мережевих з'єднань. Маршрутизатор повинен підтримувати протоколи резервування пропускної здатності, такі як Rapid Spanning Tree Protocol (RSTP) або Multiple Spanning Tree Protocol (MSTP), для забезпечення високої доступності мережі.

З урахуванням вимог, найкращою моделлю для виконання даної задачі є маршрутизатор IR829-2LTE-EA-EK9(рис 2.4)



Рисунок 2.4 - Маршрутизатор IR829-2LTE-EA-EK9

Маршрутизатор Cisco IR829-2LTE-EA-EK9 є потужним промисловим пристроєм, спеціально розробленим для застосування в умовах, де надійне з'єднання та мережева функціональність є важливими. Він може бути використаний у різних сценаріях, включаючи мобільні мережі, транспортні системи, розумні міста та мультикомп'ютерні системи. Він є ідеальним вибором для мультикомп'ютерних систем, де потрібне надійне і швидке мережеве підключення.

Він пропонує високу продуктивність і забезпечує ефективну маршрутизацію даних між комп'ютерами в системі.

Маршрутизатор також підтримує бездротовий доступ Wi-Fi, що дозволяє комп'ютерам і пристроям у системі безпроводово підключатися до мережі. Це особливо зручно для мультикомп'ютерних систем, де кабельне підключення може бути обмеженим або незручним. Окрім надійного і швидкого підключення, маршрутизатор IR829-2LTE-EA-EK9 має розширені можливості безпеки. Він підтримує функції шифрування, автентифікації і контролю доступу, що дозволяє забезпечити захист мережевого трафіку і запобігти несанкціонованому доступу до системи.

Маршрутизатор також має гнучкість налаштування, що дозволяє його адаптувати до конкретних вимог і потреб мультикомп'ютерної системи. Завдяки різноманітним опціям налаштування і розширення, ви можете налаштувати маршрутизатор таким чином, щоб він відповідав конкретним вимогам вашої системи.

3.Мережеві кабелі.

Ethernet-кабелі є одними з найбільш поширених типів мережевих кабелів і використовуються для підключення комп'ютерів, серверів, коммутаторів і маршрутизаторів до локальної мережі. Існує кілька категорій Ethernet-кабелів, які відрізняються за швидкістю передачі даних і пропускну здатністю. Ось огляд декількох типів Ethernet-кабелів:

– категорія 5e (Cat 5e): Кабель категорії 5e є покращеною версією оригінального кабелю категорії 5. Він забезпечує швидкість передачі даних до 1000 Мбіт/с (1 Гбіт/с) і широко використовується для домашніх і невеликих бізнесових мереж. Кабель Cat 5e складається з чотирьох пар проводів, з яких лише дві пари використовуються для передачі даних;

– категорія 6 (Cat 6): Кабель категорії 6 покращує пропускну здатність по порівнянню з Cat 5e і забезпечує швидкість передачі даних до 10 Гбіт/с на відстані до 55 метрів. Кабель Cat 6 також має чотири пари проводів і покращену

екранировку, що допомагає знизити електромагнітну перешкоду і спотворення сигналу;

– категорія 6а (Cat 6a): Кабель категорії 6а є подальшим покращенням кабелю Cat 6 і підтримує швидкість передачі даних до 10 або навіть 40 Гбіт/с на відстані до 100 метрів. Cat 6a має кращу екранировку, що дозволяє знизити перешкоди і забезпечити більш стабільну передачу даних.

Крім того, Ethernet-кабелі можуть бути виконані у вигляді прямого кабелю (Straight-through) або перехресного кабелю (Crossover), в залежності від того, які пристрої ви підключаєте.

Прямі кабелі (Straight-through cables) - це тип Ethernet-кабелів, які використовуються для підключення різних типів пристроїв, наприклад, комп'ютерів, серверів або коммутаторів, до маршрутизаторів або модемів. У прямих кабелях проводи в кожній парі проводів підключені однаковим чином з одного кінця кабелю до іншого кінця(Рис2.5).

Перехресні кабелі (Crossover cables) - це тип Ethernet-кабелів, які використовуються для підключення двох пристроїв однакового типу без використання комутатора або маршрутизатора. У перехресних кабелях проводи в певних парах проводів перехрещуються (crossed over) для забезпечення правильного з'єднання пристроїв(Рис2.6).

Для моделювання мультикомп'ютерної мережі з топологією "потовщене дерево", ви могли б використовувати прямі кабелі для підключення кінцевих пристроїв до комутаторів або маршрутизаторів, а перехресні кабелі для підключення комутаторів між собою або маршрутизаторів між собою у вузлах мережі, де відбувається розгалуження та потовщення топології дерева.

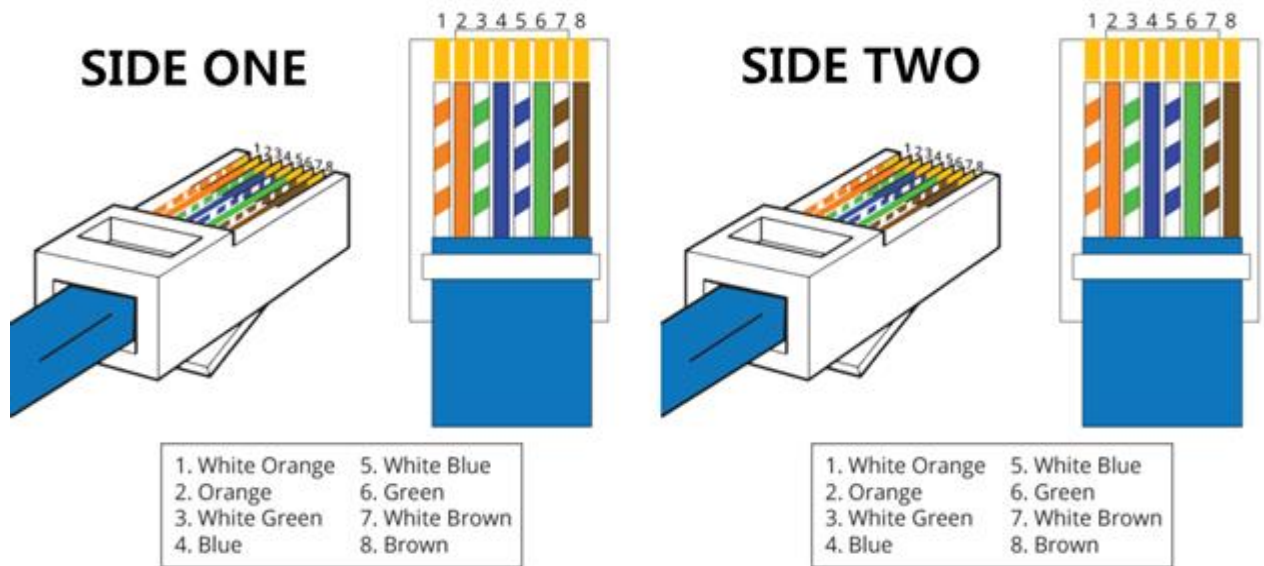


Рисунок 2.5 - Прямі кабелі

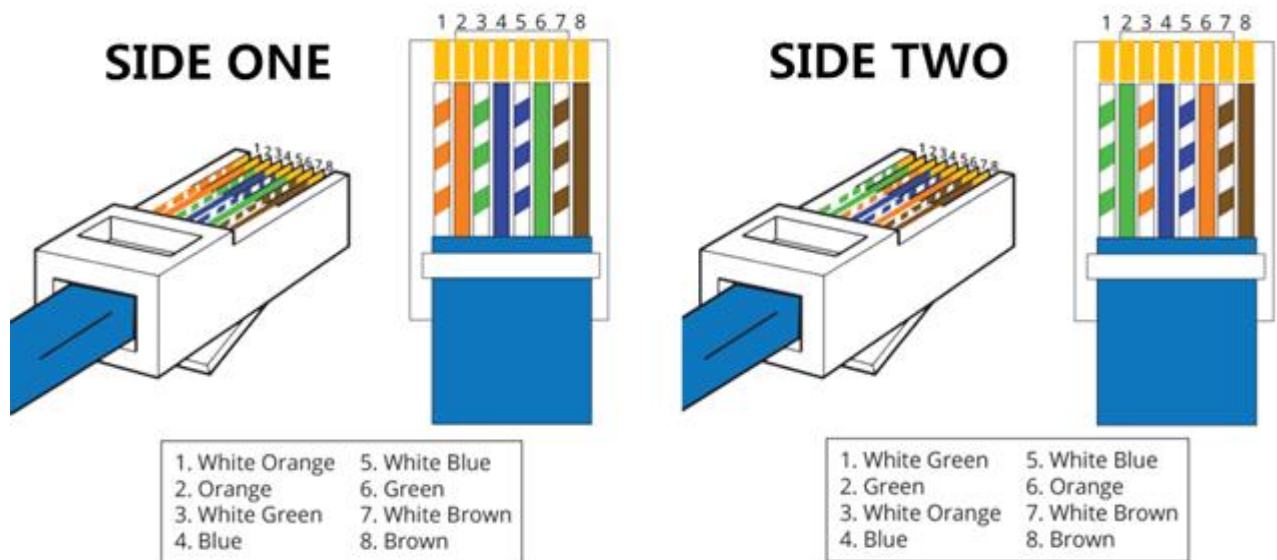


Рисунок 2.6 - Перехресні кабелі

4. Персональний комп'ютер.

Для апаратного забезпечення мультикомп'ютерної системи нам знадобляться комп'ютери з такими характеристиками:

Процесор: Для більшості завдань у мультикомп'ютерних системах, зазвичай, достатньо потужних процесорів, таких як Intel Core i5 або Intel Core i7. Однак, вибір конкретного процесора залежить від потреб і вимог вашої системи;

Оперативна пам'ять: Рекомендована кількість оперативної пам'яті залежить від типу завдань, які будуть виконуватися в системі. Зазвичай, мінімальна кількість оперативної пам'яті становить 8 ГБ, але для більш вимогливих застосунків може бути необхідно більше;

Жорсткий диск: Жорсткий диск використовується для зберігання операційної системи, даних та програмного забезпечення. Розмір жорсткого диска повинен бути достатнім для зберігання всіх необхідних файлів та даних;

Мережева карта: Кожен комп'ютер повинен мати мережеву карту (Network Interface Card - NIC), яка дозволяє підключатися до мережі. Більшість сучасних комп'ютерів вже мають вбудовану мережеву карту.

Для правильної роботи системи було обрано комп'ютер ARTLINE Business B29 v40 з такими характеристиками(табл. 2.3):

Таблиця 2.3 - Характеристики комп'ютера ARTLINE Business B29 v40

Процесор	Шестиядерный Intel Core i5-10400 (2.9 - 4.3 ГГц)
Видеокарта	Intel UHD Graphics 630
Объем оперативной памяти	8 ГБ
Объем SSD	240 ГБ

2.5 Висновки

Під час проектування мультикомп'ютерної системи з використанням топології "потовщене дерево" були розглянуті декілька важливих аспектів, пов'язаних з цією технологією. Топологія "потовщене дерево" є ефективним рішенням для мультикомп'ютерних систем. Вона дозволяє створювати ієрархічну структуру з центральним комутатором та підключеними до нього комутаторами. Це дозволяє покращити продуктивність та забезпечити надійність мережі.

Для моделювання мультикомп'ютерної системи було вибрано відповідне програмне забезпечення. Вибір залежав від потреб проекту та доступності функцій, необхідних для належного моделювання системи. Вибір впав на Cisco Packet Tracer який є одним з популярних програмних засобів для моделювання мереж. Він надає

можливість створювати та налаштовувати мережеві пристрої, симулювати трафік та перевіряти працездатність мережі. Це робить його відмінним вибором для моделювання мультикомп'ютерної системи згідно топології "потовщене дерево".

При виборі апаратного забезпечення для моделювання мультикомп'ютерної системи було враховано такі фактори, як потужність процесора, обсяг пам'яті та кількість портів.

					КвРКІ190247.19.02.26 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ МУЛЬТИКОМП'ЮТЕРНОЇ СИСТЕМИ ЗГІДНО ТОПОЛОГІЇ «ПОТОВЩЕНЕ ДЕРЕВО»

3.1 Створення моделі мультимп'ютерної системи згідно топології «потовщене дерево»

Моделювання багатомп'ютерної системи за топологією потовщеного дерева передбачає створення ієрархічної мережевої структури, яка забезпечує ефективну комунікацію та координацію між взаємопов'язаними комп'ютерами. Цей підхід широко використовується в розподілених системах для мінімізації надмірності, зменшення перевантаженості мережі та забезпечення відмовостійкості.

Топологія потовщеного дерева спрямована на створення деревоподібної структури, де кожен комп'ютер в системі з'єднаний каналами зв'язку, утворюючи мережу. Побудований зв'язний граф, який охоплює всі комп'ютери, не створюючи жодних петель чи циклів. Така ієрархічна структура забезпечує ефективну передачу даних і полегшує управління системою.

При моделюванні багатомп'ютерної системи на основі топології потовщеного дерева враховується декілька факторів.

По-перше, корінь потовщеного дерева вибирається, як правило, шляхом виборів або на основі заздалегідь визначених критеріїв. Корінь слугує центральним вузлом, з якого доступні всі інші комп'ютери.

По-друге, потовщеного дерева має бути побудоване таким чином, щоб між будь-якими двома комп'ютерами існував унікальний шлях. Це гарантує, що повідомлення або дані можуть бути надійно передані без виникнення петель або двозначності(рис 3.1).

роль або рівень в ієрархії(рис. 3.2). Ці ролі можуть включати кореневі, проміжні або листові вузли, залежно від їх положення в структурі дерева. Така ієрархічна організація сприяє ефективному потоку даних і дозволяє легко адмініструвати і контролювати систему.

Мультикомп'ютерна система, організована за допомогою топології потовщеного дерева, складається з кількох вузлів, які з'єднані між собою для спільної обробки завдань. Кожен вузол у системі має свій власний процесор і пам'ять, і вони спілкуються між собою за допомогою протоколів комунікації.

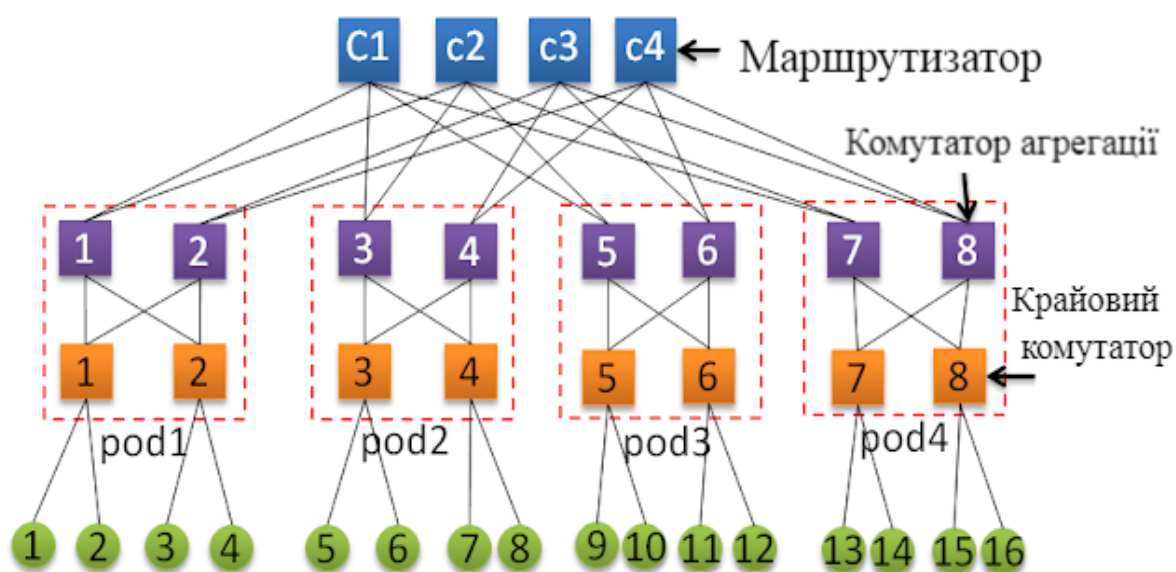


Рисунок 3.2 - Ролі кожного елемента в мультикомп'ютерній системі

Алгоритми та протоколи є основою роботи мультикомп'ютерних систем потовщеного дерева. Вони дозволяють ефективно використовувати ресурси системи, забезпечуючи високу продуктивність та масштабованість. Давайте розглянемо деякі алгоритми та протоколи, що використовуються в такій системі:

– розподілення завдань: Одним з найважливіших алгоритмів у мультикомп'ютерних системах є розподілення завдань між вузлами. Цей алгоритм визначає, які завдання будуть виконуватись на кожному вузлі, залежно від їх характеристик, наявної потужності та доступної пам'яті. Це може бути реалізовано шляхом розбиття задачі на підзадачі та призначення їх відповідним вузлам;

– координація та комунікація: У мультикомп'ютерних системах, заснованих на топології потовщеного дерева, потрібен ефективний механізм координації та комунікації між вузлами. Це може бути досягнуто за допомогою різних протоколів, таких як:

1. Протоколи передачі повідомлень: Для передачі повідомлень між вузлами можуть використовуватись протоколи, такі як Message Passing Interface (MPI) або протокол передачі повідомлень (Message Transfer Protocol). Ці протоколи дозволяють вузлам обмінюватись даними та повідомленнями для координації роботи системи.;

2. Протоколи синхронізації: Для забезпечення синхронізації вузлів у системі можуть використовуватись протоколи, такі як Barrier або Token Ring. Ці протоколи дозволяють вузлам чекати на завершення певної операції всіма вузлами або передавати контрольний сигнал вздовж потовщеного дерева.

– обробка даних: У мультикомп'ютерних системах потовщеного дерева для обробки даних можуть використовуватись різні алгоритми, такі як MapReduce або поділ-та-збір. Ці алгоритми дозволяють розбити задачу на менші підзадачі, обробити їх окремо на різних вузлах та потім зібрати результати обробки;

– планування ресурсів: В мультикомп'ютерних системах потовщеного дерева необхідне ефективне планування використання ресурсів, таких як процесорний час, пам'ять та мережева пропускна здатність. Алгоритми планування можуть бути розроблені для максимізації продуктивності системи та забезпечення рівномірного розподілу завдань та ресурсів між вузлами.

Моделювання багатокomp'ютерної системи на основі топології потовщеного дерева вимагає ретельного врахування вимог до системи, таких як масштабованість, відмовостійкість та продуктивність. Для створення структури остовного дерева можуть використовуватися різні алгоритми і методи, які гарантують, що вона відповідає бажаним цілям і обмеженням системи.

В цілому, топологія потовщеного дерева забезпечує ефективну основу для моделювання багатокomp'ютерних систем, забезпечуючи організовану та

ефективну комунікацію між взаємопов'язаними комп'ютерами. Завдяки створенню ієрархічної структури та оптимізації з'єднань цей підхід підвищує загальну продуктивність, відмовостійкість і керованість системи.

Мультикомп'ютерна система реалізована з варіантів організації комп'ютерних вузлів із використанням мережі. Вона базується на ієрархічній структурі, в якій вузли розташовані на різних рівнях (рис 3.3).

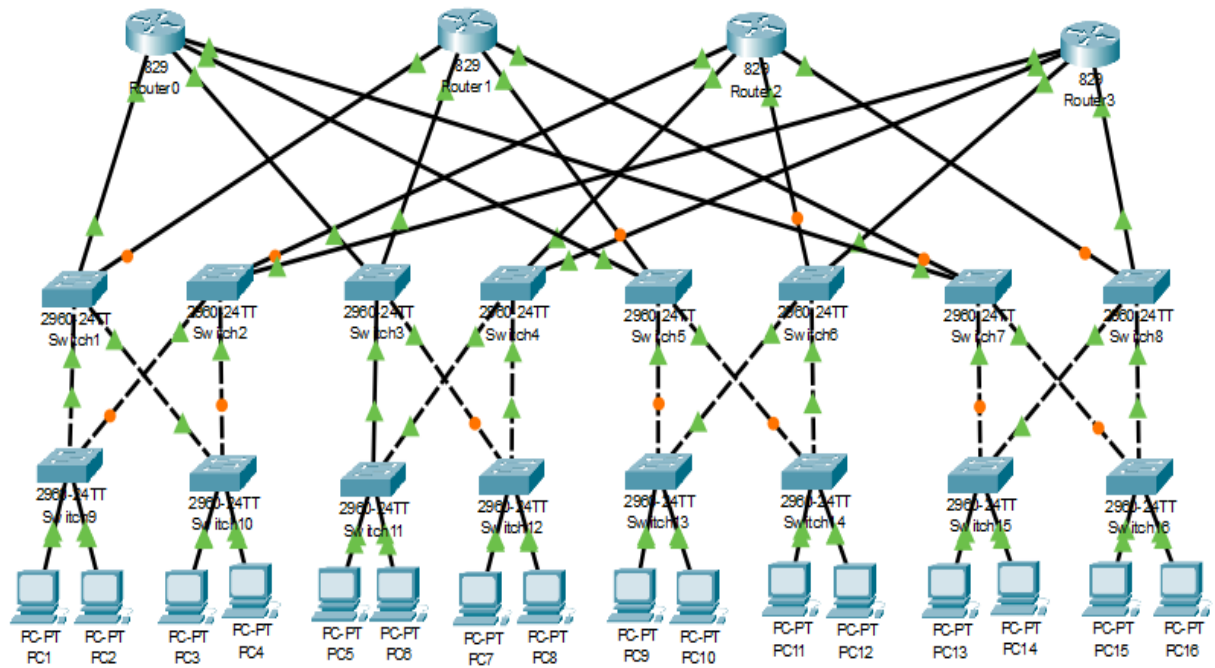


Рисунок 3.3 Модель мультикомп'ютерної системи згідно топології «потовщене дерево»

На першому рівні знаходяться 4 маршрутизатори які є центральними вузлами системи, вони забезпечують з'єднання з іншими мережами. Кожен маршрутизатор володіє власною IP-адресою та забезпечує маршрутизацію пакетів даних між різними мережами. Вони забезпечують глобальний доступ до ресурсів та зовнішніх мереж для всієї системи.

На другому знаходяться 8 комутаторів кожен з яких забезпечує з'єднання між пристроями в межах підмережі. Комутатори працюють на рівні каналу даних моделі OSI і забезпечують передачу даних на основі MAC-адрес. Кожен комутатор

має кілька портів, які підключаються інших комутаторів у мережі. Вони також підтримують VLAN для логічного розділення трафіку.

На третьому рівні знаходяться 8 комутаторів які забезпечують з'єднання в межах підмережі. Кожен з цих комутаторів підключений до двох комп'ютерів.

На четвертому рівні знаходяться 16 комп'ютерів. Кожен комп'ютер є кінцевим пристроєм мережі. Вони є робочими станціями, які надають або використовують різноманітні послуги та ресурси в мережі.

Існування багатошляхових маршрутів полегшує наявність різних маршрутів до одного і того ж пункт призначення, і це допоможе мати кращі можливості варіанти балансування. Хоча в мережі Fat Tree існує більше одного маршруту до певного пункту призначення однак, класичні протоколи маршрутизації на основі вектора відстані та зв'язку класичні протоколи маршрутизації за вектором відстані та станом зв'язку не можуть використовувати цю властивість багатошляховості. Протоколи маршрутизації в Інтернеті зазвичай маршрутизують і пересилають пакети на основі IP-адреси призначення. Як наслідок, пакети з однаковою адресою призначення будуть прокладені тим самим шляхом.

Безсумнівно, існують деякі протоколи маршрутизації які мають характеристику багатошляхової маршрутизації з рівною вартістю однак, вони розділяють трафік статично в залежності від інформації, отриманої з заголовка пакета. Як наслідок, не буде ніякого врахування вимог потоку трафіку до параметрів QoS, крім того, не враховується стан загального навантаження на мережу не береться до уваги стан загального навантаження на мережу. Іншими словами, такі алгоритми алгоритми ECMP здатні лише вибирати серед декількох шляхів, які мають однакову найменшу вартість. Проблема балансування навантаження є однією з основних проблем в MCS в їх різних формах, незалежно від того, чи є вони фізичними чи це фізична MCS, чи віртуальна MCS. MCS зазвичай дозволяють маршрутизацію декількома шляхами з метою покращення стійкості до збоїв, а також збільшення пропускної здатності мережі за рахунок вирішення проблеми перевантаження. Рівень 2 і Рівень 3 здатні виконувати

					КвРКІ190247.19.02.26 ПЗ	Арк. 47
Зм.	Арк.	№ докум.	Підпис	Дата		

багатошляхову маршрутизацію; однак, кожен рівень розгортає її на основі своїх протоколів. Наприклад, spanning tree використовується на рівні 2, отже, для пари вузлів відправника-одержувача буде доступний лише один шлях для пари вузлів відправника-одержувача буде доступний лише один шлях

Мережа Fat-Tree має розмір $(K/2)^2$ це означає, що між будь-якими двома серверами мережі існує чотири шляхи маршрутизації мережі. Крім того, всі всі комутатори повинні бути з'єднані у $(K/2)^2$ комутаторів ядра (4 комутатори ядра)

У топології є K комутаторів у кожному PoD, розташованих у два шари по $K/2$ комутаторів, один шар для листових комутаторів, а інший для корневих комутаторів. Кожен комутатор 3 ряду з'єднано з $K/2$ корневими комутаторами.

3.2 Реалізація програмного забезпечення

Програмний код, написаний в ході цієї роботи, має вигляд проміжного програмного забезпечення. Він виконує процес передавання даних між комп'ютерами у даній системі. Розбиває конкретний потік даних на порції, та додає до кожної з них заголовки з номером послідовності. Далі кожний сегмент інкапсулюється в IP-пакет і передається через IP-протокол до хоста-отримувача.

Основними елементами коду є:

- Створення TCP/IP-сокета:

```
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

- Використання bind() для зв'язування сокета з адресою сервера. Адреса – 0.0.0.0., що вказує на поточний сервер, а номер порту - 10000.

```
server_address = ('0.0.0.0', 10000)
```

```
print >>sys.stderr, 'starting up on %s port %s' % server_address
```

```
sock.bind(server_address)
```

- Виклик listen() який переводить сокет у режим сервера, а accept() очікує на вхідне з'єднання.

```
sock.listen(1)
```

```
while True:
```

```
print >>sys.stderr, 'waiting for a connection'
```

```
connection, client_address = sock.accept()
```

– accept() повертає відкрите з'єднання між сервером і клієнтом разом з адресою клієнта. Дані зчитуються з з'єднання за допомогою recv() і передаються за допомогою sendall().

```
try:
```

```
print >>sys.stderr, 'connection from', client_address
```

```
while True:
```

```
    data = connection.recv(16)
```

```
    print >>sys.stderr, 'received "%s"' % data
```

```
    if data:
```

```
        print >>sys.stderr, 'sending data back to the client'
```

```
        connection.sendall(data)
```

```
    else:
```

```
        print >>sys.stderr, 'no more data from', client_address
```

```
        break
```

```
finally:
```

```
    connection.close()
```

– використовуючи зручну функцію create_connection() для підключення до сервера, ми заощаджуємо час. Функція приймає один аргумент, кортеж з двох значень, що містить адресу сервера, і виводить найкращу адресу для з'єднання.

```
sock = socket.create_connection(('0.0.0.0', 10000))
```

```
print >>sys.stderr, 'Family  :', families[sock.family]
```

```
print >>sys.stderr, 'Type   :', types[sock.type]
```

```
print >>sys.stderr, 'Protocol:', protocols[sock.proto]
```

```
print >>sys.stderr
```

– Багато серверів мають більше одного мережевого інтерфейсу, а отже, більше однієї IP-адреси. Замість того, щоб запускати окремі копії сервісу,

Зм.	Арк.	№ докум.	Підпис	Дата

прив'язані до кожної IP-адреси, використовуємо спеціальну адресу INADDR_ANY.

На рисунку 3.4 зображена UML діаграма цієї програми.

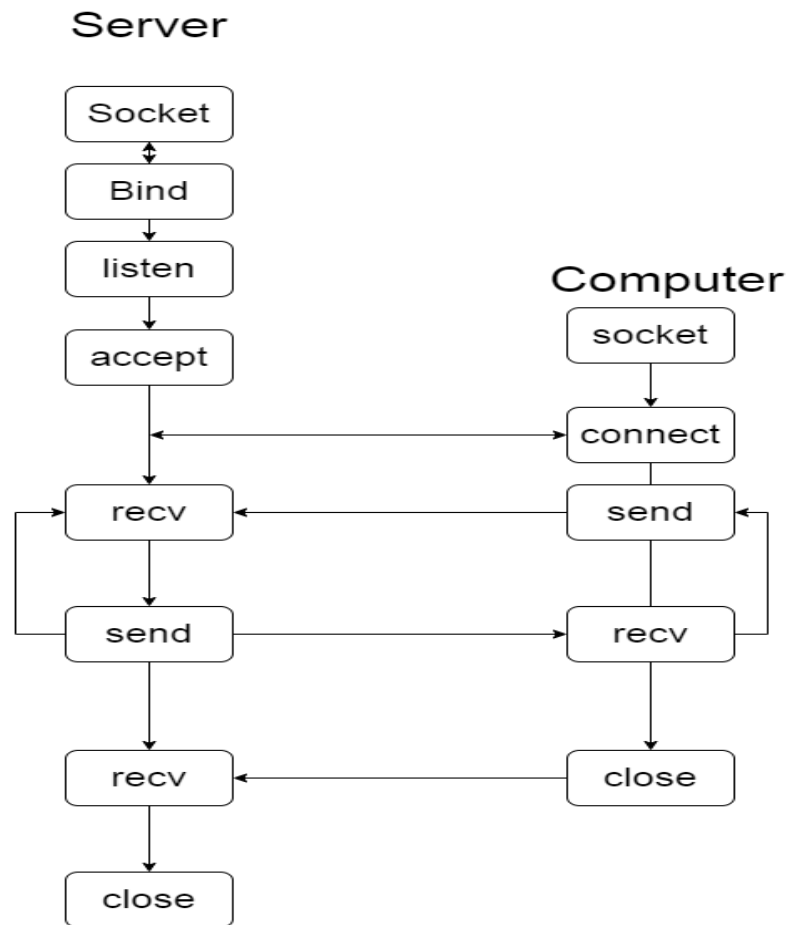


Рисунок 3.4 UML діаграма

3.3 Протокол Cisco Spanning Tree

При проектуванні мультикомп'ютерної мережі однією з найбільших проблем, яку необхідно вирішити, є запобігання утворенню петель. Якщо в комутованій мережі утворюється петля, обсяг трафіку, який може передаватися між комутаторами, швидко використовує всю смугу пропускання, доступну для кожного з них. Один з методів запобігання виникненню петель полягає в тому, щоб забезпечити тільки один шлях між комутаторами і переконатися, що в усій комутованій мережі немає надлишкових шляхів. Хоча це може спрацювати в

невеликих розгортаннях, при роботі з комутованими мережами трохи більшого розміру необхідно розгортання надлишковості.

Одним з найкращих варіантів програмного забезпечення в цьому контексті є протокол Cisco Spanning Tree Protocol, який є широко використовуваним стандартом для мережевої комутації. Протокол Cisco Spanning Tree Protocol (STP) - це мережевий протокол, який допомагає запобігти виникненню петель у мережах Ethernet(рис3.5).

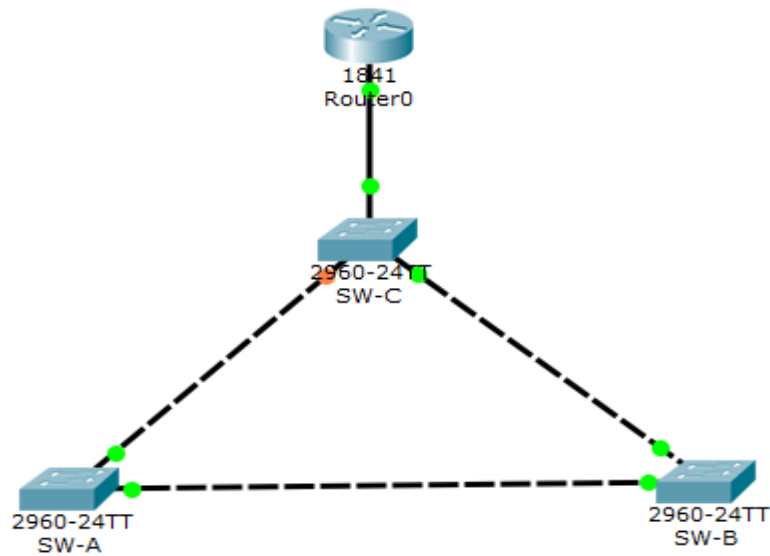


Рисунок3.5 - Приклад протоколу Cisco Spanning Tree Protocol

Він працює шляхом виявлення надлишкових каналів і їх блокування, що запобігає утворенню петель і гарантує, що між будь-якими двома мережевими пристроями є тільки один активний шлях. Це допомагає підвищити надійність і доступність мережі. STP реалізований в пристроях Cisco у вигляді стандарту IEEE 802.1D, який визначає протокол і його роботу. Він використовує процес під назвою "вибір кореневого моста" для визначення центрального комутатора в мережі, який стає кореневим мостом.

Кожен комутатор в мережі обчислює найкоротший шлях до кореневого моста і визначає, який з його портів є "кореневим портом" - портом, який з'єднує його з сегментом мережі, що забезпечує найкоротший шлях до кореневого моста. Якщо

існує декілька шляхів до кореневого моста, STP вибере той, який має найнижчу вартість, яка базується на пропускній здатності каналу.

Spanning Tree Protocol також визначає будь-які надлишкові канали в мережі і блокує їх, запобігаючи утворенню петель. Для цього в кожному сегменті мережі вибирається "призначений порт" - порт, який забезпечує найкоротший шлях до кореневого моста. Всі інші порти в цьому сегменті блокуються.

В цілому, STP від Cisco забезпечує надійний спосіб запобігання утворенню петель в мережах Ethernet і гарантує, що мережевий трафік проходить ефективно і надійно.

Основними компонентами Cisco STP є:

1. Кореневий міст (Root Bridge): Це комутатор, який має найнижчий Bridge ID (BID) у мережі. Він визначає деревоподібну топологію і слугує як початкова точка для вибору найкоротших шляхів для кожного комутатора у мережі.

2. Мости (Bridges): Кожен комутатор, що бере участь у Cisco STP, називається мостом. Вони обмінюються Bridge Protocol Data Units (BPDU) для обчислення шляхів до кореневого моста і визначення ролей портів (root port, designated port, non-designated port).

3. Порти: Кожен порт комутатора може мати одну з трьох ролей - кореневий порт (root port), призначений порт (designated port) або непризначений порт (non-designated port). Кореневі порти є шляхами найближчими до кореневого моста, призначені порти використовуються для передачі трафіку, а непризначені порти заблоковані для уникнення петель.

4. Bridge Protocol Data Units (BPDU): Це пакети, що використовуються для обміну інформацією між комутаторами. BPDU містять інформацію про Bridge ID, позицію порту, статус порту і т.д. Комутатори використовують BPDU для визначення шляхів і ролей портів.

5. Таймери: Cisco STP використовує різні таймери для виявлення змін в мережі і перерахунку шляхів. Наприклад, Hello Timer використовується для

					КвРКІ190247.19.02.26 ПЗ	Арк. 52
Зм.	Арк.	№ докум.	Підпис	Дата		

періодичного відправлення BPDU, Max Age Timer визначає час очікування на прихід BPDU, а Forward Delay Timer визначає час, який порт перебуває у стані блокування перед переходом в активний режим.

Після вибору кореневого комутатора кожному з портів призначається роль в залежності від його місця в топології STP доступні ролі портів при використанні потовщеного дерева показані нижче:

- кореневий - порт, якому призначена ця роль, є обраним найкращим шляхом для досягнення кореневого комутатора;
- призначений - для порту з цією роллю вибирається найкращий шлях до певного комутованого сегмента; для кожного комутованого сегмента є лише один призначений порт;
- альтернативний - порт, якому призначено цю роль, вибирається як резервний для кореневого порту; якщо у кореневого порту виникнуть проблеми, цей порт перейме на себе роль кореневого порту;
- резервний - порт, якому призначено цю роль, вибирається як резервний для призначеного порту; якщо у призначеного порту виникнуть проблеми, цей порт перейме на себе роль призначеного порту.

Після обчислення найкращого шляху і призначення ролі кожному з портів, всі порти з альтернативними або резервними ролями STP будуть заблоковані, щоб запобігти виникненню петель.

Стани інтерфейсу протоколу Spanning Tree

Кожен з увімкнених портів комутатора бере участь в STP, кожен з цих портів проходить через процес станів інтерфейсу, перш ніж йому буде дозволено пересилати трафік.

Як показано на рисунку 3.6, існує п'ять різних станів, в яких може перебувати порт, описаних нижче:

					КвРКІ190247.19.02.26 ПЗ	Арк. 53
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.6 - Стани потовщеного дерева

Стан блокування(Blocking) - порти, які перебувають у стані блокування, не пересилають трафік; вони просто слухають мережу, щоб переконатися, що їм слід продовжувати блокувати трафік. Якщо стан комутованої мережі змінюється, порт може перейти у стан прослуховування. Всі порти починають роботу у стані блокування після початкової ініціалізації комутатора.

Стан прослуховування(Listening) - порти, які перебувають у стані прослуховування, не пересилають трафік. Перебуваючи в цьому стані, порт буде лише слухати трафік, як і в стані блокування. Це перший стан, який настає після

Зм.	Арк.	№ докум.	Підпис	Дата

стану блокування після того, як порт налаштовано на пересилання кадрів. За замовчуванням час перебування у стані прослуховування становить 15 секунд.

Стан навчання(Learning) - порти, які перебувають у стані навчання, не пересилають трафік; перебуваючи у цьому стані, порт прослуховує трафік і починає запам'ятовувати адреси підключених пристроїв у сегменті. За замовчуванням час перебування у стані навчання становить 15 секунд.

Стан переадресації(Forwarding) - порти, які перебувають у стані переадресації, переадресовують трафік, а також продовжують вивчати адреси з сегмента.

Вимкнений стан(Disabled) - порти, що знаходяться у відключеному стані, не пересилають трафік і не прослуховують мережевий трафік.

При впровадженні RSTP змінився час, який порт витрачає на перехід, і метод, який використовується для переходу. Це дає можливість комутованій мережі швидше почати переадресацію трафіку без зайвих затримок, ці затримки є поширеною скаргою.

3.4 Тестування розробленої мультикомп'ютерної системи

Проведення різних тестових сценаріїв в мережі є корисним для перевірки функціональності, надійності та продуктивності нашої мережі. При тестуванні мультикомп'ютерної системи згідно з топологією потовщеного дерева проводять перевірку працездатності, надійності та ефективності системи. Це включає перевірку з'єднань між комп'ютерами, передачу даних та перевірку стабільності мережі. Метою тестування є впевнитися, що система працює належним чином та задовольняє вимогам перед її використанням у реальних умовах. Ось кілька варіантів проведення різних тестових сценаріїв у моделі мультикомп'ютерної мережі:

					КвРКІ190247.19.02.26 ПЗ	Арк. 55
Зм.	Арк.	№ докум.	Підпис	Дата		

3.4.1 Перевірка з'єднання: пінгування між різними комп'ютерами та пристроями в мережі

Перевірку з'єднання можна проводити шляхом використання блоку даних протоколу або пінгування.

Протокол блоку даних (PDU) це блок даних, який передається через мережу з одного вузла до іншого на певному рівні мережевого стеку. Кожен рівень мережевого стеку виконує свої функції та додає свій власний заголовок до PDU, щоб забезпечити правильну комунікацію між пристроями у мережі.

Основна ідея розподілу PDU на різні рівні полягає в тому, щоб різні протоколи та послуги мережі мали свої власні відповідні об'єкти даних та механізми керування. Це дозволяє розподілити функціональність мережі на окремі шари та спрощує розробку, розширення та управління мережею.

Важливою концепцією є процес інкапсуляції, який відбувається на кожному рівні мережевого стеку. При передачі даних з верхнього рівня до нижчого, дані обгортаються заголовком (та, можливо, підзаголовками) на кожному рівні. Коли дані досягають приймаючого пристрою, вони розпаковуються, пройшовши крізь кожний рівень стеку, досягаючи оригінального вмісту.

Цей процес інкапсуляції та розпакування дозволяє мережевим пристроям працювати на своїх рівнях, не зважаючи на деталі нижчих рівнів. Кожен рівень стеку може виконувати свої власні функції, такі як розміщення відправленого пакета на мережевій медіа, маршрутизація пакетів або забезпечення надійності передачі.

Загальна ідея PDU та розподілу функціональності на різні рівні мережевого стеку допомагає забезпечити ефективну та надійну комунікацію між пристроями у мережі. Використовуємо ці концепції для перевірки процесом передачі даних у мережі (рис 3.7).

PDU List Window						
Fire	Last Status	Source	Destination	Type	Color	Time(sec)
	Successful	PC1	PC6	ICMP		0.000
	Successful	PC2	PC15	ICMP		0.000
	Successful	PC4	PC8	ICMP		0.000
	Successful	PC15	PC9	ICMP		0.000
	Successful	PC16	PC11	ICMP		0.000
	Successful	PC9	PC14	ICMP		0.000
	Successful	PC7	PC3	ICMP		0.000
	Successful	PC11	PC13	ICMP		0.000
	Successful	PC13	PC8	ICMP		0.000
	Successful	PC8	PC14	ICMP		0.000
	Successful	PC5	PC14	ICMP		0.000
	Successful	PC14	PC3	ICMP		0.000
	Successful	PC3	PC6	ICMP		0.000
	Successful	PC6	PC10	ICMP		0.000

Рисунок 3.7 - Перевірка з'єднання за допомогою PDU

На кожному рівні мережевого стеку використовуються різні назви PDU:

1. Рівень фізичної передачі: На цьому рівні PDU називається бітами або сигналами, які передаються фізичними медіа. Наприклад, в цифрових мережах це можуть бути 0 та 1, а в аналогових - рівні сигналу;
2. Рівень каналу зв'язку: На цьому рівні PDU називається кадрами. Кадр - це блок даних, який містить дані з попереднього рівня, а також додаткову інформацію, таку як адреси MAC (Media Access Control), заголовки помилок та контрольні суми;
3. Рівень мережі: На цьому рівні PDU називається пакетами. Пакет містить дані з попереднього рівня, а також заголовки протоколу мережевого рівня, такі як IP (Internet Protocol), які використовуються для маршрутизації та адресації;
4. Рівень транспорту: На цьому рівні PDU називається сегментами (для TCP) або датаграмами (для UDP).

Пінгування (ping) - це процес перевірки доступності та відповідності мережевого пристрою, комп'ютера або сервера. Пінгування використовує протокол ICMP (Internet Control Message Protocol), який дозволяє відправляти короткі пакети даних і отримувати відповіді.

Основні етапи пінгування:

- відправлення пакета: Процес починається з відправлення пакета даних (ping-запиту) від одного пристрою до іншого. Пакет містить деякі контрольні біти, випадкові дані та мітку часу;
- прийом пакета: Цільовий пристрій отримує пакет і перевіряє його цілісність. Він також створює відповідний пакет (ping-відповідь);
- надсилання відповіді: Цільовий пристрій надсилає пакет з відповіддю назад до відправника;
- отримання відповіді: Відправник отримує пакет з відповіддю і аналізує його. Він перевіряє, чи співпадають дані відповіді з відправленими та вимірює час, який пройшов від відправлення до отримання відповіді.

Основна ідея пінгування полягає в тому, що пристрій відправляє пакет даних до цільового пристрою і очікує відповіді. Якщо цільовий пристрій доступний і працює належним чином, він прийме пакет і надішле відповідь назад. Якщо ж відповіді немає або час відповіді дуже великий, то це може вказувати на проблему зі з'єднанням або недоступність цільового пристрою.

Відправником для пінгування є 192.168.0.1(PC1)

Статистика пінгування для 192.168.0.14(PC14):

Пакети: відправлено = 4, отримано = 4, втрачено = 0 (втрат немає).

Приблизний час проходження в мілісекундах: Мінімальний = 0 мс, Максимальний = 174 мс, Середній = 47 мс.

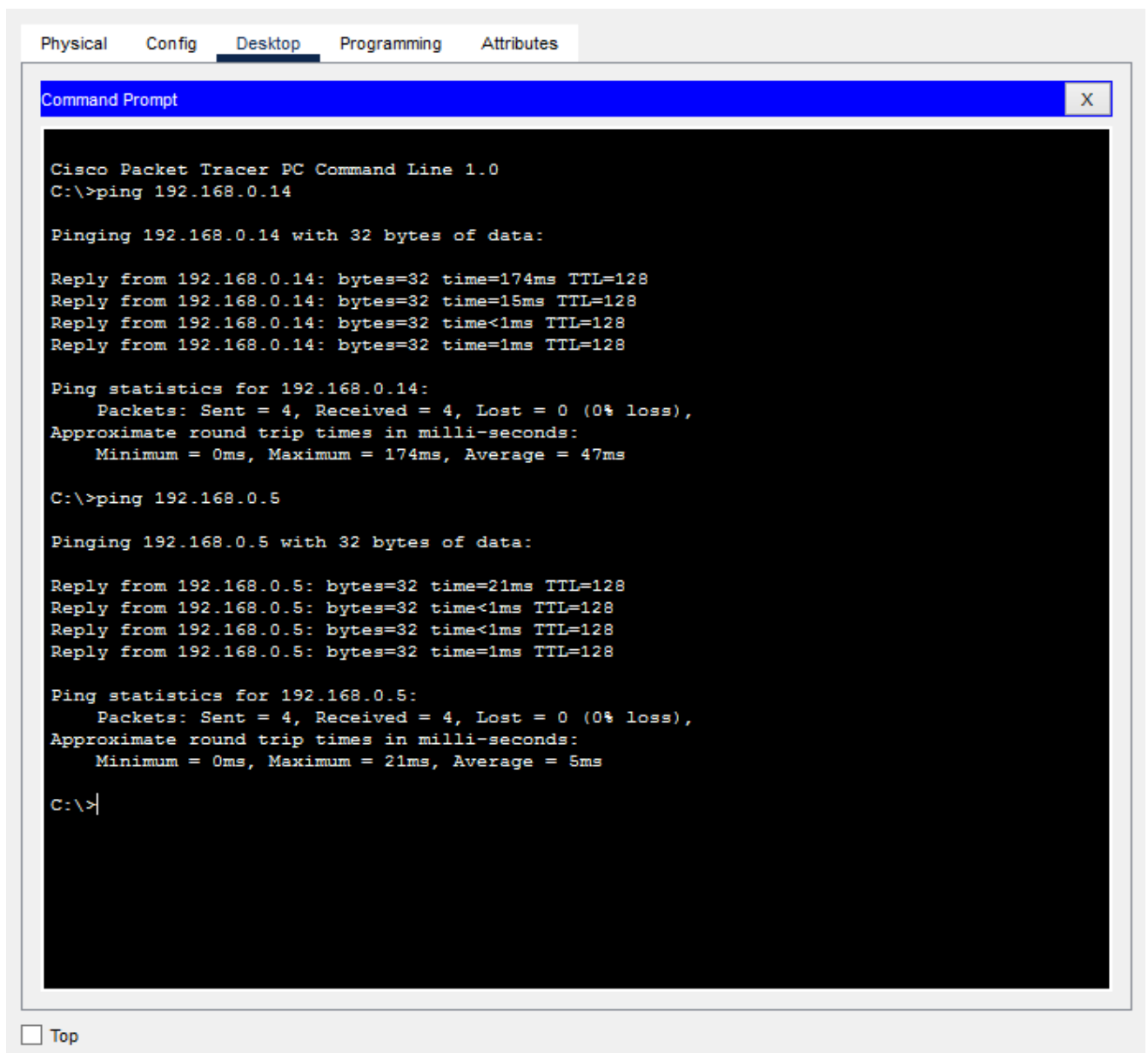
Статистика пінгування для 192.168.0.5(PC5):

Пакети: відправлено = 4, отримано = 4, втрачено = 0 (втрат немає).

Приблизний час проходження в мілісекундах: Мінімальний = 0 мс, Максимальний = 21 мс, Середній = 5 мс.

					КвРКІ190247.19.02.26 ПЗ	Арк. 58
Зм.	Арк.	№ докум.	Підпис	Дата		

На рисунку 3.8 зображено результати цього пінгування.



```
Physical  Config  Desktop  Programming  Attributes
Command Prompt X
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.0.14

Pinging 192.168.0.14 with 32 bytes of data:

Reply from 192.168.0.14: bytes=32 time=174ms TTL=128
Reply from 192.168.0.14: bytes=32 time=15ms TTL=128
Reply from 192.168.0.14: bytes=32 time<1ms TTL=128
Reply from 192.168.0.14: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.0.14:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 174ms, Average = 47ms

C:\>ping 192.168.0.5

Pinging 192.168.0.5 with 32 bytes of data:

Reply from 192.168.0.5: bytes=32 time=21ms TTL=128
Reply from 192.168.0.5: bytes=32 time<1ms TTL=128
Reply from 192.168.0.5: bytes=32 time<1ms TTL=128
Reply from 192.168.0.5: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.0.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 21ms, Average = 5ms

C:\>|
```

Top

Рисунок 3.8 - Пінгування розробленої мережі

Ці дані показують, що ми успішно пінгуємо обидві IP-адреси, отримуємо відповіді на всі відправлені пакети і втрат немає. Крім того, можемо побачити час, який потрібен для проходження пакетів від відправника до отримувача, виміряний у мілісекундах.

3.4.2 Тестування пропускної здатності

Тестування пропускної здатності мережі дозволяє виміряти максимальну кількість даних, яку мережа може передати за певний час(рис. 3.9). Це важливий критерій для оцінки продуктивності мережі і виявлення можливих обмежень або проблем з пропускною здатністю.

Vis.	Time(sec)	Last Device	At Device	Type
	0.006	Router0	Switch3	ICMP
	0.007	Switch10	PC3	ICMP
	0.007	Switch10	PC4	ICMP
	0.008	Switch3	Switch11	ICMP
	0.010	Switch11	PC5	ICMP
	0.010	Switch11	PC6	ICMP
	0.010	Switch11	Switch4	ICMP
	0.012	Switch4	Router3	ICMP
	0.015	Router3	Switch8	ICMP
	0.017	Switch8	Switch16	ICMP
	0.017	Switch8	Switch15	ICMP
	0.019	Switch16	PC15	ICMP
	0.019	Switch16	PC16	ICMP
	0.019	Switch15	PC13	ICMP
	0.019	Switch15	PC14	ICMP
	0.022	PC16	Switch16	ICMP
	0.024	Switch16	Switch8	ICMP
	0.026	Switch8	Router3	ICMP
	0.027	Router3	Switch4	ICMP
	0.029	Switch4	Switch11	ICMP
	0.031	Switch11	Switch3	ICMP
	0.033	Switch3	Router0	ICMP
	0.036	Router0	Switch1	ICMP
	0.038	Switch1	Switch9	ICMP
	0.040	Switch9	PC1	ICMP

Рисунок 3.9 - Тестування пропускної здатності протоколом ICMP

ICMP є протоколом, який використовується для передачі повідомлень про стан мережі та повідомлень про помилки. Він працює на рівні мережевого (IP) рівня в протокольному стеку TCP/IP. ICMP надає можливість обмінюватися повідомленнями між різними мережевими пристроями, такими як роутери, мережеві комутатори або комп'ютери.

Одним з основних використань ICMP є передача повідомлень про помилки. Коли виникає помилка під час обробки пакетів, наприклад, коли пакет не може

бути доставлений до призначення, ICMP генерує повідомлення про помилку та надсилає його джерелу пакету. Це дозволяє виявити проблему та вжити відповідних заходів для виправлення ситуації.

Крім повідомлень про помилки, ICMP також використовується для різних інших цілей. Наприклад, ICMP може використовуватися для відправки запитів підтвердження доступності мережевих пристроїв. Команда Ping використовує ICMP, щоб відправити Echo Request (запит підтвердження доступності) до пристрою і отримати Echo Reply (відповідь) в разі успішного отримання запиту.

ICMP також містить повідомлення про перенаправлення, які використовуються для повідомлення пристроям про більш ефективний маршрут до певної мережі. Він також має повідомлення про час життя (TTL), які використовуються для контролю часу, який пакет може перебувати в мережі перед його видаленням.

3.4 Висновки

Під час моделювання та тестування мультикомп'ютерної системи згідно топології "потовщене дерево" були проведені дослідження та виконані ряд дій для перевірки працездатності та ефективності системи

Під час тестування розробленої мультикомп'ютерної системи були проведені перевірки з'єднання та тестування пропускну здатності, розроблено проміжне програмне забезпечення для передачі даних між комп'ютерами . Перевірка з'єднання включала пінгування між різними комп'ютерами та пристроями в мережі, що демонструє правильну налагодженість комунікації. Тестування пропускну здатності дозволило оцінити швидкість передачі даних у мережі та виявити можливі обмеження.

					КвРКІ190247.19.02.26 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У даній роботі було представлено розробку мультикомп'ютерну систему згідно топології «потовщене дерево» яка складалась з 4 маршрутизаторів, 16 комутаторів та 16 комп'ютерів.

Метою даної роботи було створення мультикомп'ютерної системи на базі топології «потовщене дерево», розглянено основні типи архітектур, створено проміжне програмне забезпечення.

Програмним забезпеченням для цієї мультикомп'ютерної системи було обрано Cisco Packet Tracer.

Для підтвердження правильності розробки системи було проведено декілька тестувань. Це показало, що розроблена мультикомп'ютерна система працює досить коректно.

					КвРКІ190247.19.02.26 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Understand and Configure STP on Catalyst Switches URL: <https://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/5234-5.html> (Дата звернення:08.05.2023)
2. What is Fat Tree and how to construct it in 4-steps? URL: <https://blogchinmaya.blogspot.com/2017/04/what-is-fat-tree-and-how-to-construct.html> (Дата звернення: 20.05.2023)
3. Cisco Spanning Tree Protocol URL: <https://www.cisco.com/c/en/us/td/docs/routers/access/3200/software/wireless/SpanningTree.html> (Дата звернення:20.05.2023)
4. Introduction to TCP: Transmission Control Protocol URL: <https://internalpointers.com/post/introduction-tcp-transmission-control-protocol> (Дата звернення:23.05.2023)
5. Socket Programming HOWTO URL: <https://docs.python.org/3/howto/sockets.html> (Дата звернення 23.05.2023)
6. Y. Li; Y. Lai; G. Lu; F. Yan; P. Wei; Y. Xie. Innovative design of long-span steel–concrete composite bridge using multi-material topology optimization. *Journal. Engineering Structures*. Volume 269. 2022. 114838. ISSN 0141-0296, <https://doi.org/10.1016/j.engstruct.2022.114838>.
7. M. Aliyu; M. Murali; Z. Zhang; A Gital; S. Boukari; Y. Huang; I. Yakubu. Management of cloud resources and social change in a multi-tier environment: A novel finite automata using ant colony optimization with spanning tree. *Technological Forecasting and Social Change*. Volume 166. 2021. 120591. ISSN 0040-1625, <https://doi.org/10.1016/j.techfore.2021.120591>.
8. Teng F.; H. Zhang; C. Luo; Q. Shan. Delay tolerant containment control for second-order multi-agent systems based on communication topology design, *Journal*. Volume 380. 2020. Pages 11-19. ISSN 0925-2312 <https://doi.org/10.1016/j.neucom.2019.10.001>

					КвРКІ190247.19.02.26 ПЗ	Арк. 63
Зм.	Арк.	№ докум.	Підпис	Дата		

9. Li Y.; X. Mu. Necessary and sufficient conditions for leader-following consensus of multi-agent systems with random switching topologies. *Nonlinear Analysis: Hybrid Systems*. Volume 37. 2020. 100905. ISSN 1751-570X <https://doi.org/10.1016/j.nahs.2020.100905>.

10. Zhou L.; Zheng Y.; Zhao Q.; Xiao F.; Zhang Y.. Game-based coordination control of multi-agent systems. *Journal of Systems & Control Letters*. Volume 169. 2022. 105376. ISSN 0167-6911. <https://doi.org/10.1016/j.sysconle.2022.105376>.

11. Kapidani W.; Merkel M.; Schöps S.; Vázquez R.. Tree–cotree decomposition of isogeometric mortared spaces in $H(\text{curl})$ on multi-patch domains. *Computer Methods in Applied Mechanics and Engineering*. Volume 395 . 2022 . 114949 . ISSN 0045-7825. <https://doi.org/10.1016/j.cma.2022.114949>

12. Wang Q.; He W.; Zin L.; Tan D.; Zhong W.. Bipartite consensus for a class of nonlinear multi-agent systems under switching topologies: A disturbance observer-based approach. *Journal*. Volume 488. 2022. Pages 130-143. ISSN 0925-2312 <https://doi.org/10.1016/j.neucom.2022.02.081>

13. X. Pu; L. Ren; Y. Liu; R. Pu. Couple-group consensus for heterogeneous MASs under switched topologies in cooperative-competitive systems. *Neurocomputing*. Volume 441. 2021. Pages 335-349. <https://doi.org/10.1016/j.neucom.2020.11.013>.

14. Narayanan G.; Syed Ali M.; Alsulami H.; Stamov G.; Stamova I.; Ahmad B.. Impulsive security control for fractional-order delayed multi-agent systems with uncertain parameters and switching topology under DoS attack. *Information Sciences*. Volume 618. 2022. Pages 169-190. ISSN 0020-0255 <https://doi.org/10.1016/j.ins.2022.10.123>.

15. Gonzalez J.; Sahuquillo J.; García P.; Quiles F.; Mora G.; Towards an efficient combination of adaptive routing and queuing schemes in Fat-Tree topologies. *Journal of Parallel and Distributed Computing*. Volume 147. 2021. Pages 46-63. ISSN 0743-7315. <https://doi.org/10.1016/j.jpdc.2020.07.009>.

16.Yuyang W., Dezun D., Fei L. Understanding node connection modes in Multi-Rail Fat-tree. *Journal of Parallel and Distributed Computing*. Volume 167. 2022. Pages 199-210. ISSN 0743-7315. <https://doi.org/10.1016/j.jpdc.2022.04.019>.

17.Debobroto Das R.; Javed I.. PISA switch based traffic engineering in fat-tree data center networks. *Computer Networks*. Volume 215. 2022. 109210. ISSN 1389-1286. <https://doi.org/10.1016/j.comnet.2022.109210>

18.Jihe W.; Danghui W.; Meikang Q.; Yao C.; Bing G. A locality-aware shuffle optimization on fat-tree data centers. *Journal of Future Generation Computer Systems*. Volume 89. 2018. Pages 31-43. ISSN 0167-739X. <https://doi.org/10.1016/j.future.2018.06.016>.

19.Guelbenzu G.; Calabretta N.; Raz O.. Hybrid fat-tree: Extending fat-tree to exploit optical switch transparency with WDM. *Journal of Optical Fiber Technology*. Volume 44. 2018. Pages 89-101. ISSN 1068-5200. <https://doi.org/10.1016/j.yofte.2018.01.018>.

20.Ciznicki M.; Kurowski K.; Węglarz J.. Energy and performance improvements in stencil computations on multi-node HPC systems with different network and communication topologies . *Journal of Future Generation Computer Systems*. Volume 115. 2021. Pages 45-58. ISSN 0167-739X. <https://doi.org/10.1016/j.future.2020.08.018>

21.Qingyu S.; Fang W.; Dan F.; Weibin X.. Adaptive load balancing based on accurate congestion feedback for asymmetric topologies. *Journal Computer Networks*. Volume 157. 2019. Pages 133-145. ISSN 1389-1286. <https://doi.org/10.1016/j.comnet.2019.04.006>.

22.Jaafar A.. Sorted-GFF: An efficient large flows placing mechanism in software defined network datacenter. *Karbala International Journal of Modern Science*. Volume 4. Issue 3. 2018. Pages 313-331. ISSN 2405-609X. <https://doi.org/10.1016/j.kijoms.2018.06.003>.

23.Jianxuan L.; Yingying S.;Guangxi C.. Topology structure based on detrended cross-correlation coefficient of exchange rate network of the belt and road countries.

Physica A: Statistical Mechanics and its Applications. Volume 509. 2018. Pages 1140-1151. ISSN 0378-4371. <https://doi.org/10.1016/j.physa.2018.06.059>.

24.Sudarshan A.; Sudhakar C.; Ramesh T.. Energy efficient VM scheduling and routing in multi-tenant cloud data center. *Sustainable Computing: Informatics and Systems*. Volume 22. 2019. Pages 139-151. ISSN 2210-5379. <https://doi.org/10.1016/j.suscom.2019.04.004>.

25.Tianmin Z.; Jiamin C.; Xuancong X.; Zuhong O.; Hao Y.; Jianqiang L.; Anbo M.. A novel multi-agent based crisscross algorithm with hybrid neighboring topology for combined heat and power economic dispatch.*Journal of Applied Energy*. Volume 342. 2023. 121167. ISSN 0306-2619. <https://doi.org/10.1016/j.apenergy.2023.121167>.

26.Zhenbiao G.; Hailiang S.; Xinqing L.; Yingjun W.. Multi-resolution topology optimization using B-spline to represent the density field. *Journal Advances in Engineering Software*. Volume 182. 2023. 103478. ISSN 0965-9978. <https://doi.org/10.1016/j.advengsoft.2023.103478>.

27.Jie G.; Xiaomeng W.; Mi X.; Vinh P.; Liang G.; Timon R.. Multi-patch isogeometric topology optimization for cellular structures with flexible designs using Nitsche's method. *Journal Computer Methods in Applied Mechanics and Engineering*. Volume 410. 2023. 116036. ISSN 0045-7825. <https://doi.org/10.1016/j.cma.2023.116036>.

28.Yukun F.; Masaki N.; Yuki N.; Kei M.; Takayuki Y.. Multi-material topology optimization for additive manufacturing considering dimensional constraints. *Journal Computer Methods in Applied Mechanics and Engineering*. Volume 410. 2023. 116027. ISSN 0045-7825. <https://doi.org/10.1016/j.cma.2023.116027>.

29.Wang Y.; Sigmund O.. Multi-material topology optimization for maximizing structural stability under thermo-mechanical loading. *Journal Computer Methods in Applied Mechanics and Engineering*. Volume 407. 2023. 115938. ISSN 0045-7825. <https://doi.org/10.1016/j.cma.2023.115938>.

30.Hao D.; Jinping J.; Li Y.; Fakui W.; Weisheng C.. Event-triggered exponential synchronization of complex dynamical networks with cooperatively directed spanning

tree topology. *Journal Neurocomputing*. Volume 330. 2019. Pages 355-368. ISSN 0925-2312. <https://doi.org/10.1016/j.neucom.2018.11.013>.

31.Valuskar A.; Shandilya M.; Rajawat A.. Statistical traffic pattern for mixed torus topology and pathfinder based traffic and thermal aware routing protocol on NoC. *Journal Integration*. Volume 87. 2022. Pages 158-168. ISSN 0167-9260. <https://doi.org/10.1016/j.vlsi.2022.06.009>.

32.Lucia P.; Carlos G.. Signatures of natural selection in tree topology shape of serially sampled viral phylogenies. *Journal Molecular Phylogenetics and Evolution*. Volume 183. 2023. 107776. ISSN 1055-7903. <https://doi.org/10.1016/j.ympev.2023.107776>

33.Daifeng Z.; Haibin D.. Switching topology approach for UAV formation based on binary-tree network. *Journal of the Franklin Institute*. Volume 356. Issue 2. 2019. Pages 835-859. ISSN 0016-0032. <https://doi.org/10.1016/j.jfranklin.2017.11.026>.

34.Deborah G.; Angelo F.. Classification of microbial transglutaminases by evaluation of evolution trees, sequence motifs, secondary structure topology and conservation of potential catalytic residues. *Journal Biochemical and Biophysical Research Communications*. Volume 509. Issue 2. 2019. Pages 506-513. ISSN 0006-291X. <https://doi.org/10.1016/j.bbrc.2018.12.121>.

35.Cáceres M.; Navarro G.. Faster repetition-aware compressed suffix trees based on Block Trees. *Journal Information and Computation*. Volume 285. Part B. 2022. 104749. ISSN 0890-5401. <https://doi.org/10.1016/j.ic.2021.104749>

36.Jeannot E.. Process mapping on any topology with TopoMatch. *Journal of Parallel and Distributed Computing*. Volume 170. 2022. Pages 39-52. ISSN 0743-7315. <https://doi.org/10.1016/j.jpdc.2022.08.002>.

37.Negar R.; Somayyeh K.. Flat Ball: Dynamic topology for energy management of optical interconnection networks in data centers. *Journal Optical Switching and Networking*. Volume 48. 2023. 100730. ISSN 1573-4277. <https://doi.org/10.1016/j.osn.2023.100730>.

38.Gropp W.. Using node and socket information to implement MPI Cartesian topologies. *Journal Parallel Computing*. Volume 85. 2019. Pages 98-108. ISSN 0167-8191. <https://doi.org/10.1016/j.parco.2019.01.001>.

39.Morteza K.; Myungjin L.. A solution to MPTCP's inefficiencies under the incast problem for Data Center Networks. *Journal Computer Communications*. Volume 161. 2020. Pages 238-247. ISSN 0140-3664. <https://doi.org/10.1016/j.comcom.2020.07.034>.

40.Vesović M., Smiljanić A., Kostić D.. Fast and scalable routing protocols for data center networks. *Journal Digital Communications and Networks*. 2022 ISSN 2352-8648. <https://doi.org/10.1016/j.dcan.2022.06.010>.

					КвРКІ190247.19.02.26 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

ДОДАТОК А

Лістинг коду проміжного програмного забезпечення

```
import socket
import sys
server_address = ('0.0.0.0', 10000)
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
print >>sys.stderr, 'starting up on %s port %s' % server_address
sock.bind(server_address)
sock.listen(1)

while True:
    print >>sys.stderr, 'waiting for a connection'
    connection, client_address = sock.accept()
        print >>sys.stderr, 'connection from', client_address
            while True:
                data = connection.recv(16)
                print >>sys.stderr, 'received "%s"' % data
                if data:
                    print >>sys.stderr, 'sending data back to the client'
                    connection.sendall(data)
                else:
                    print >>sys.stderr, 'no more data from', client_address
                    break
            finally:
                connection.close()
        try:
            message = 'This is the message. It will be repeated.'
            print >>sys.stderr, 'sending "%s"' % message
            sock.sendall(message)
```

```

amount_received = 0
amount_expected = len(message)
while amount_received < amount_expected:
    data = sock.recv(16)
    amount_received += len(data)
    print >>sys.stderr, 'received "%s"' % data
finally:
    print >>sys.stderr, 'closing socket'
    sock.close()

server_name = sys.argv[1]
server_address = (server_name, 10000)
print >>sys.stderr, 'starting up on %s port %s' % server_address
sock.bind(server_address)
sock.listen(1)

while True:
    print >>sys.stderr, 'waiting for a connection'
    connection, client_address = sock.accept()
    try:
        print >>sys.stderr, 'client connected:', client_address
        while True:
            data = connection.recv(16)
            print >>sys.stderr, 'received "%s"' % data
            if data:
                connection.sendall(data)
            else:
                break
    finally:
        connection.close()

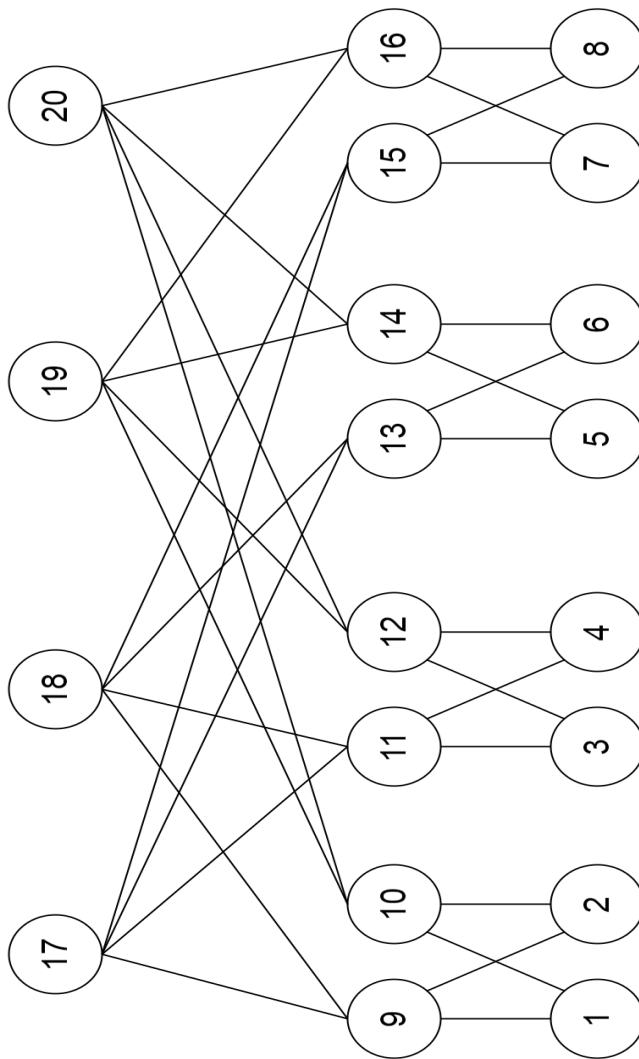
```

ДОДАТОК Б

(обов'язковий)

Копія креслення «Схема граф топології «потовщене дерево»»

КВРКІ.190247.19.02.26

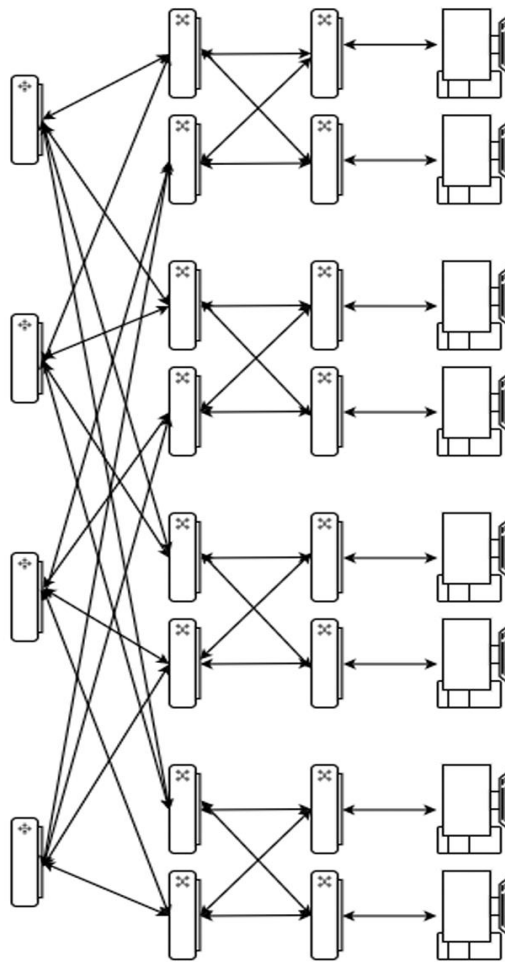


КВРКІ.190247.19.02.26		Лист	Місце	Місцевість
Схема Граф топології "потовщене дерево"		Н		
Сп. Клас.	№ докум.	Титул	Дата	
Розроб.	Виконав.			
Перевір.	Відрізняє			
Т.Вопит.				
Н.В.Стор.				
Затв.				
ХНУ КІ2-19-2		Архив 3		

ДОДАТОК В (обов'язковий)

Копія креслення «Схема топології «потовщене дерево»»

КвРКІ.190247.19.02.26



КвРКІ.190247.19.02.26		Літ.	Місяц	Матриця
		І		
Схема топології "потовщене дерево"		Дата		
Від	Для	Місце	Підпис	
Розроб.	Затверд.	Склад.	Склад.	
Технік	Проєкт.	Проєкт.	Проєкт.	
Інженер	Інженер	Інженер	Інженер	
Зам.	Зам.	Зам.	Зам.	
		ХНУ КІТ-19-2		

Ім'я користувача:
Кафедра КІ

Дата перевірки:
08.06.2023 21:30:23 EEST

Дата звіту:
08.06.2023 21:30:55 EEST

ID перевірки:
1015516044

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005591

Назва документа: Трет'яков_Мультикомп'ютерна система згідно топології «потовщене дерево»

Кількість сторінок: 65 Кількість слів: 12070 Кількість символів: 95944 Розмір файлу: 976.04 KB ID файлу: 1015170709

17% Схожість

Найбільша схожість: 4.68% з Інтернет-джерелом (<http://elar.khnu.km.ua/jspui/bitstream/123456789/10506/1/%d0%9a%...>)

16.7% Джерела з Інтернету 191 Сторінка 67

2.17% Джерела з Бібліотеки 142 Сторінка 69

1.28% Цитат

Цитати 8 Сторінка 70

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи 12

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 7.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилки в документах: 10%**

ID: 115332 Назва: БКР Мультикомп'ютерна система згідно топології «потовщене дерево» Додано в БД: 2023-06-08 Автора: Б. Р. Трет'яков Керівники: К. М. Березька Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	86783	704	9801 (11%)	83 (12%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Трет'яков Богдан Романович

Тема: Мультикомп'ютерна система згідно топології «потовщене дерево»

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 57

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є проектування та моделювання мультикомп'ютерної системи згідно топології «потовщене дерево»
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 кваліфікаційної роботи проведено дослідження предметної області (проаналізовано аспекти та особливості мультикомп'ютерних систем) та виконано постановку задачі дослідження. В другому розділі кваліфікаційної роботи проведено проектування мультикомп'ютерної системи згідно топології «потовщене дерево», а саме: виконано опис мультикомп'ютерної системи; розроблено структурну схему мережі; визначено необхідне апаратне та програмне забезпечення; В третьому розділі кваліфікаційної роботи проведено моделювання мультикомп'ютерної системи згідно топології «потовщене дерево», а саме: реалізовано модель системи в середовищі Cisco Packet Tracer; розроблено проміжне програмне забезпечення.
4. Позитивні сторони роботи: висока практична цінність роботи.
5. Негативні сторони роботи: недостатня увага опису проміжного програмного забезпечення.
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:

Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «добре» 3,75(С)

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

Гурмані Іван Васильович, доцент кафедри ІТІЗ

" 8 " червня 2023 р.

И.І.І. (підпис)

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

Трет'якова Богдана Романовича

ІІІ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-19-2

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.

22 квітня 2023 року



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: Мультимедійна система згідно топології «потовне дерево»

Автор: Трет'яков Богдан Романович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Березька Катерина Миколаївна, к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданій поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданій поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укріття 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 загальноживими фразами або виразами, про що свідчить посилання системи на збіг з 10-30 джерелами на один фрагмент речення;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 17% і адресується до 333 періодичних джерел, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС

К. М. Березька

С. М. Лисенко

Г. О. Говорущенко