

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень


Мультипроцесорна система загального призначення на основі топології «тор»
Назва теми


КвРКІ. 170144.17.01.15 ПЗ
Шифр


Галузь знань 12 «Інформаційні технології»
Шифр, назва


Спеціальність 123 «Комп'ютерна інженерія»
Шифр, назва

Освітня програма «Комп'ютерна інженерія»
Назва

Виконав: студент IV курсу, група KI-17-1  А.І.Молодницький
Ініціали, прізвище

Керівник  О.М.Березький
Підпис, дата Ініціали, прізвище

Нормоконтролер  С.М. Лисенко
Підпис, дата Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної інженерії та системного програмування  Т.О. Говорущенко
Підпис Ініціали, прізвище

« 29 » червня 2021 р.

Хмельницький 2021
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНИХ І ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА СИСТЕМОГО ПРОГРАМУВАННЯ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЯ ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

ЗАТВЕРДЖУЮ

Зав. Кафедри Т.О.Говорушенко


11 " 01 2021 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

Молодніцькому Андрію Ігорович

Прізвище, ім'я, по батькові студента

Тема проекту (роботи) Мультипроцесорна система загального призначення на основі топології «тор»

1. Керівник проекту (роботи) Березький О.М. д.т.н., проф.

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

Затверджена наказом ректора університету від 05.02.2021 р. № 11

2. Строк подання студентом проекту (роботи) на кафедру 07.06.2021 р.

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

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

Загальні відомості

Проектування програмно-технічного засобу

Програмно-апаратна реалізація та тестування програмно-технічного засобу

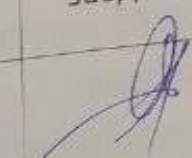
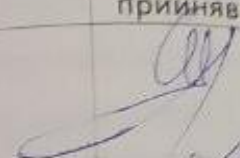


5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Логічна топологія кластерної комп'ютерної системи

Фізична топологія кластерної комп'ютерної системи

Схема розподілу пам'яті

6. Консультанти розділів дипломного проекту (роботи)

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

2021 р.

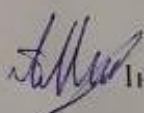
7. Дата видачі завдання

« 11 » 01

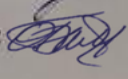
КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником		виконано
	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження		виконано
	Робота над розділом 1 – дослідження предметної області та постановка задачі		виконано
	Робота над розділом 2 – моделювання та проектування мультипроцесорної комп'ютерної мережі		виконано
	Робота над розділом 3 – реалізація комп'ютерної мультипроцесорної мережі		виконано
	Оформлення пояснювальної записки згідно вимог		виконано
	Попередній захист ВКР		виконано
	Захист ВКР на засіданні ЕК	Червень 2021 року	

Студент

Підпис  Ініціали Молодніцький А.І.

Керівник

Підпис  Ініціали Березький О.М.

№ р я д к а	Ф о р м а т	Позначення	Найменування	К і л л и с т і в	№ ек з	П р и м і т к а
			<u>Текстові документи</u>			
1		КвРКІ 170144.17.01.13 ПЗ	Пояснювальна записка	60		
			<u>Графічні матеріали</u>			
2		КвРКІ 170144.17.01.13 Е8	Логічна топологія Мультипроцесорної комп'ютерної мережі	1		
		КвРКІ 170144.17.01.13 Е8	Фізична топологія мультипроцесорної комп'ютерної мережі	1		
3		КвРКІ 170144.17.01.13 Е8	Схема розподілу пам'яті	1		

КвРКІ 170144.17.01.15 ВП

Зм	Арк	Недокум	Підпис	Дата
Розробив		Молодницький		
Перевір.		Березький		
Н.контр.		Лисенко		
Затв.		Говорущенко		29.06

Відомість проекту

Літера	Аркуш	Аркушів
У	1	1

ХНУ, КІ-17-1

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Мультипроцесорна система загального призначення на основі топології «тор»».

Автор роботи: Молодніцький А.І.

Керівник роботи: Березький О.М.

Пояснювальна записка: 60 с., 17 рис., 10 табл., 3 дод., 51 джерел. Графічна частина: 7 презентаційних слайдів.

Мультипроцесорна система загального призначення на основі топології «тор»

Метою роботи є створення Мультипроцесорної системи загального призначення

Об'єктом дослідження є програмно-технічні засоби для створення мультипроцесорної мережі

Предметом дослідження є мультипроцесорна система її можливості та її недоліки.

Практичне значення має створення незалежної Мультипроцесорної системи для прискорення обробки інформації та за для забезпечення втрати файлів та збільшення їх надійності.



Підпис студента

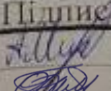
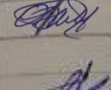


24.06.2021

Дата

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	3
ВСТУП.....	4
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	6
1.1 Аналіз задачі, обґрунтування вибору моделі життєвого циклу для реалізації проекту	6
1.2 Мультипроцесори та мультикомп'ютери	6
1.3 Переваги та недоліки мультипроцесорів	14
1.4 Висновки	15
2 ТОПОЛОГІЯ TOR	17
2.1 Створення топології	20
2.2 Висновок	24
3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ	25
3.1 Розподільна система	25
3.2 Ієрархія пам'яті та створення кластера	45
ВИСНОВКИ.....	58
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	59
Додаток А «Копія креслення логічної топології мультипроцесорної комп'ютерної мережі».....	63
Додаток Б «Копія креслення фізичної топології мультикомп'ютерної мережі».....	64
Додаток В «Копія креслення схема розподілу пам'яті»	65

КвРКІ. 170144.17.01.15 ПЗ

Зм.	Арк.	№докум.	Підпис	Дата	Мультипроцесорна система загального призначення на основі топології «тор»	Літера	Аркуш	Аркушів
Виконав		Молодніцький		24.06.21				
Перевір.		Березький О.М.		24.06.21			2	65
Н.контр.		Лисенко С.М.		24.06.21				
Затвер.		Говорушенко Ф.О.		24.06.21				

ХНУ. КІ-17-1

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ОЗУ - оперативний запам'ятовувальний пристрій

TLB – технологія, що оптимізує перетворення віртуальних адрес в фізичні

ПЗ - програмне забезпечення

БД - база даних

БПР - блок прийняття рішень

ГА - генетичний алгоритм

ПК – персональний комп'ютер WS2016-windows server 2016

IP –Intellectual property (інтелектуальна власність) Вузол – ПК який підключений до мережі

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		3

ВСТУП

Історія інформатики та промисловості налічує понад 50 років. Ідея створення програмованих обчислювальних пристроїв, які працюють із програмою, що зберігається в пам'яті, була відома в середині XIX століття. завдяки роботі англійського дослідника Чарльза Беббіджа. Успішна реалізація цих ідей стала можливою лише в перших електронних комп'ютерах (КОМП'ЮТЕРИ), що з'явилися після Другої світової війни

Загальні принципи. Побудова електронних комп'ютерів із програмою, що зберігається в пам'яті, була сформульована американським математиком Дж. Фон Нейманом. (1946). Ці принципи, або, як їх прийнято сьогодні визначати, - архітектура - були принципами послідовних обчислень і забезпечували: 1) послідовну організацію пам'яті; 2) процес послідовного виконання інструкцій (програмних операцій) у порядку їх розташування в пам'яті машини; 3) послідовна робота пристроїв (арифметичні та логічні пристрої, реєстри, пристрої оперативної пам'яті та довготривалої пам'яті), робота яких контролюється одним пристроєм управління. Але вже тоді розробники комп'ютерів зрозуміли, що наявність одного приводу та контролера не є принциповим обмеженням. Крім того, оскільки комп'ютер складався з пристроїв з різною швидкістю, це призвело до значної втрати потенційної продуктивності комп'ютера. Методи паралельної обробки інформації з'явилися та вдосконалилися під час загального розвитку комп'ютерних архітектур. У комп'ютерах другого покоління, крім операційних систем та систем програмування, існували методи мультипрограмування, тобто виконання декількох програм на одному пристрої. Однак комп'ютери залишалися практично однопроцесорними.

Сучасні комп'ютерні системи - це складні комп'ютерні системи, які насправді є системами паралельної роботи (використовуються в найпростіших версіях, наприклад, настільних), що дозволяють одночасно (паралельно) виконувати операції над однією або кількома програмами. Зазвичай сучасний комп'ютер під'єднується до локальної та глобальної мережі. Тому традиційна концепція операційної системи для такого комп'ютера перетворюється на більш

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		4

складне утворення - операційне середовище, в якому обчислювальні процеси відбуваються, розвиваються та зникають. Таким чином, архітектура та робоче середовище комп'ютерної системи тісно пов'язані між собою і є її фундаментальними характеристиками, що визначають принципи побудови та визначають роботу всієї надбудови програмного забезпечення, включаючи його прикладну частину. Тому ґрунтовне знання цих категорій є необхідною умовою для того, щоб програміст, і зокрема системи, мали успіх у створенні, налагодженні та керуванні комп'ютерними системами будь-якої складності.

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		5

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз задачі, обґрунтування вибору моделі життєвого циклу для реалізації проекту

Більшість комп'ютерних систем - це однопроцесорні системи, що означає, що у них є лише один процесор, який дозволяє виконувати справи. Однак наразі зростає потреба в багатопроцесорній або паралельній системі. Ці системи мають паралельні процесори трасування, які спільно використовують годинник комп'ютера, пам'ять, шину та периферію. Багатопроцесорні системи використовуються для прискорення обробки завдання, забезпечуючи при цьому єдиний доступ до кожного процесора в системі, тим самим оптимізуючи використання ресурсів (приклад архітектури багатопроцесорної системи показаний на рисунку 1.).

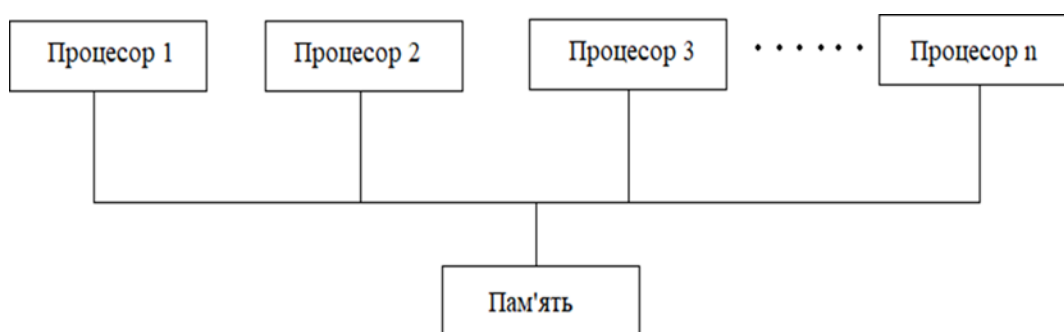


Рисунок 1.1. – Архітектура багатопроцесорної системи

1.2 Мультипроцесори та мультикомп'ютери

Порівняно з багатопроцесорними та однопроцесорними системами, перша має переваги перед останньою з точки зору швидкості та надійності. Надійність багатопроцесорної системи полягає в тому, що якщо один процесор виходить з ладу, система буде продовжувати функціонувати і виконувати завдання, хоча, звичайно, система гальмує, але все одно залишається швидшою, ніж одна процесорна система. З цією перевагою ми відразу знаходимо ще одну, яка

впливає з необхідності багатопроцесорної системи. При використанні ресурсів декількох процесорів завдання буде виконано в рази швидше, що дозволить раціонально використовувати час для виконання завдання. Використання багатопроцесорної системи збільшує продуктивність. Пропускна здатність відповідає за збільшення продуктивності. Він збільшується так часто, як кількість процесорів, що працюють паралельно, тобто пропускна здатність кожного процесора додається, і це додавання призводить до швидшої обробки інформації. Перевагою багатопроцесорної системи в довгостроковій перспективі є також ціна, оскільки процесори користуються периферійними пристроями, джерелами живлення та пам'яттю. Крім того, ця система матиме потужність, яка покриватиме попит довше, ніж ремонт. Цей параметр мінімізує подальші витрати на технічне обслуговування та модернізацію заліза. Основним недоліком багатопроцесорної системи є ціна, хоча багатопроцесорна система в перспективі дешевша, але початкова ціна висока,

і в цьому відношенні придбання однопроцесорної системи виглядає більш привабливим. У будь-якій системі паралельної обробки процесори, що виконують різні частини одного завдання, повинні якимось чином взаємодіяти один з одним для обміну інформацією. Для організації такого обміну було запропоновано та впроваджено дві розробки: - багатопроцесорні; - Кілька комп'ютерів. У першій розробці всі процесори мають спільну фізичну пам'ять. Така система називається багатопроцесорною або системою спільної пам'яті.

Мультипроцесорна модель застосовується до програмного забезпечення. Усі процеси, що працюють разом на мультипроцесорі, можуть спільно використовувати єдиний віртуальний адресний простір, який відображається у спільній пам'яті. Будь-який процес може використовувати звичайні команди читання / запису для читання слова з пам'яті або запису слова в пам'ять. Два процеси можуть обмінюватися інформацією, якщо один з них просто записує дані в пам'ять, а інший зчитує ці дані. Через цю здатність двох або більше процесів взаємодіяти мультипроцесори дуже популярні. Ця модель зрозуміла програмістам і може застосовуватися до найрізноманітніших завдань. Прикладом такої проблеми є програма, яка обробляє зображення. Копія зображення зберігається в

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		7

пам'яті. Кожен з 16 процесорів виконує процес, присвоєний одному з 16 розділів. Коли процес виявляє, що один з його об'єктів перетинає межу розділу, процес просто переходить за об'єктом до наступного розділу і читає слова в цьому розділі. У прикладі деякі об'єкти одночасно обробляються декількома процесами, так що зрештою потрібно багато координації, щоб визначити кількість будинків, дерев та рівнів. У другому типі паралельної архітектури кожен процесор має власну пам'ять, яка доступна лише для цього процесора. Ця розробка відома як багатокомп'ютерна або розподілена система пам'яті. Багатокомп'ютери - це, як правило, слабо зв'язані системи (але не обов'язкові). Основна відмінність між мультикомп'ютером і мультипроцесором полягає в тому, що кожен процесор у мультикомп'ютері має власну локальну пам'ять, до якої цей процесор може отримати доступ, виконуючи команди читання / запису, але жоден інший процесор не може отримати доступ до цієї пам'яті за допомогою тих самих команд. Таким чином, мультипроцесори мають фізичний адресний простір, який спільно використовують усі процесори, а мультикомп'ютери мають окремий фізичний адресний простір для кожного процесора.

Оскільки процесори в мультикомп'ютері не можуть просто взаємодіяти один з одним через спільну пам'ять, потрібен інший механізм взаємодії. Процесори надсилають повідомлення один одному через мережу взаємозв'язку. За відсутності спільної пам'яті в апаратному забезпеченні система забезпечує певну програмну структуру. Багатокомп'ютер не може мати єдиного віртуального адресного простору, з якого всі процеси можуть зчитувати інформацію і в який всі процеси можуть записувати інформацію. Наприклад, якщо процесор (у верхньому лівому куті) на рис. 1.2 b) виявляє цю частину свого об'єкта потрапляє в пам'ять іншого процесора (літак перемістився), він може не тільки зчитувати потрібну йому інформацію, як це могло б бути у мультипроцесорі. Він повинен зробити щось інше, щоб отримати потрібні йому дані.

Зокрема, йому потрібно якимось чином визначити, який процесор зберігає потрібні йому дані, і надіслати йому повідомлення з проханням про копію даних. Потім процесор блокується до отримання відповіді. Коли процесор даних отримує повідомлення, програмне забезпечення має проаналізувати його та відправити

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		8

назад необхідні дані. Коли процесор, який запитував дані, отримує повідомлення про це, програмне забезпечення розблоковується і починає працювати знову. У мультикомп'ютері деякі примітивні примітиви надсилання та прийому часто використовуються для взаємодії між процесорами. Тому багатокомп'ютерне програмне забезпечення має більш складну структуру, ніж багатопроцесорне програмне забезпечення. Основною проблемою є правильний розподіл даних та розумне їх розміщення. У мультипроцесорі розміщення деталей не впливає на правильність завдання, але може вплинути на продуктивність.

Таким чином, мультикомп'ютер набагато складніше програмувати, ніж мультипроцесор. Складність створення багатокомп'ютерного програмного забезпечення жодним чином не обмежує його використання, навпаки, кількість таких систем у світі постійно зростає. Це тому, що набагато простіше і дешевше побудувати великий мультикомп'ютер, ніж мультипроцесор з такою ж кількістю процесорів.

Впровадження спільної пам'яті, що ділиться сотнями процесорів, є досить складним завданням, а створення суперкомп'ютера з до десяти тисяч і більше процесорів відносно просто. Це створило дилему: мультипроцесори важко будувати, але «легко» програмувати, а мультикомп'ютери «легко» будувати, але важко програмувати.

В результаті дизайнери апаратного забезпечення почали створювати гібридні системи, відносно прості в проектуванні та відносно прості в програмуванні. Це призвело до усвідомлення того, що спільна пам'ять може бути реалізована по-різному, і в кожному випадку це матиме деякі переваги та недоліки. Майже всі дослідження в галузі архітектур з паралельною обробкою спрямовані на створення гібридних форм, що поєднують переваги обох архітектур. Тут важливо отримати систему, що розширюється, тобто, додаючи більше нових і нових процесорів, вона буде продовжувати працювати належним чином.

Багатопроцесори, як і всі комп'ютери, повинні містити пристрої введення / виводу (жорсткі диски, мережеві адаптери тощо). З деякими мультипроцесорами лише певні визначені процесори мають доступ до пристроїв вводу-виводу, в

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		9

інших системах кожен процесор має доступ до будь-якого пристрою. Якщо всі процесори мають однаковий доступ до всіх модулів пам'яті та всіх пристроїв вводу-виводу, і кожен процесор може обмінюватися іншими процесорами, то така система називається SMP (симетричний багатопроцесор). Оскільки системи SMP є найбільш поширеними, тільки цей тип системи буде розглянуто в майбутньому.

Хоча всі мультипроцесори надають процесорам відображення загального використовуваного адресного простору, часто існує багато модулів пам'яті, кожен з яких містить частину фізичної пам'яті. Процесори та модулі пам'яті з'єднані складною мережею зв'язку. Можливо, кілька процесорів намагаються прочитати слово з пам'яті, тоді як багато інших процесорів намагаються написати те саме слово, і деякі повідомлення можуть не доставлятися в тому порядку, в якому вони були відправлені. Якщо взяти до уваги існування численних копій деяких блоків пам'яті (наприклад, у кеші) і не вжити певних заходів, система створює абсолютно незрозумілу хаотичну ситуацію. Тільки неоднозначний семантичний порядок операцій читання / запису в системі може усунути двозначність у положенні даних.

Семантика пам'яті може розглядатися як контракт між програмним та апаратним забезпеченням системи. Якщо програмне забезпечення підтримує певні правила, пам'ять може давати чітко визначені результати. Ці правила відомі як шаблон узгодженості пам'яті. Багато таких моделей було запропоновано та розроблено. Суворе домовленість - найпростіша модель. У такій моделі значення останнього запису даних завжди зчитується з будь-якої пам'яті.

У разі декількох запитів на читання та запис порядок обслуговування всіх запитів визначається виключно апаратним забезпеченням, але всі процесори дотримуються однакового порядку запитів.

Консистенція процесора є більш збитковою моделлю, але її легше застосувати на великих мультипроцесорах. Він має дві характеристики:

- 1) усі процесори сприймають записи кожного процесора в тому порядку, в якому вони починаються;
- 2) усі процесори бачать записи в кожному слові пам'яті в тому самому порядку, в якому вони з'являються.

Ці два правила дуже важливі. Перший полягає в тому, що коли один процесор починає записувати в комірки пам'яті в певному порядку, всі інші процесори бачитимуть ці записи лише в тому самому порядку, і жоден інший процесор ніколи не бачитиме іншого порядку. Друге правило полягає в тому, що кожне слово в пам'яті має унікальне значення після того, як процесор зробив кілька записів у цьому слові, а потім зупинився.

Усі інші процесори повинні приймати лише останнє значення. Більш складний випадок, коли одночасно починає записувати не лише один, а, наприклад, два процесори. У цьому випадку інші процесори, що виконують операції зчитування, можуть спостерігати будь-яку послідовність дій цих двох процесорів, але таким чином, що послідовність дій будь-якого з них ніколи не переривається. У наступній моделі з низькою консистенцією записи, зроблені одним процесором, сприймаються строго послідовно. Але для усунення двозначності в послідовності, в якій записи даних спостерігаються іншими процесорами, пам'ять тут містить змінні синхронізації або операції синхронізації (апаратний аналог бар'єрної синхронізації потоків).

Коли відбувається синхронізація, усі неповні записи припиняються, і жоден новий запис не може бути запущений, поки всі старі записи не будуть завершені та синхронізовані. Синхронізація призводить пам'ять до стабільного стану, коли більше немає незавершених операцій. Самі операції синхронізації координуються одна за одною, тобто якщо їх викликає кілька процесорів, вибирається чітка послідовність і всі процесори сприймають одну і ту ж послідовність. Якщо координація слабка, час ділиться на послідовні періоди, які обмежуються моментами синхронізації.

Програмне забезпечення може навести порядок у процесі за допомогою операцій синхронізації, але це займає трохи більше часу. Погана послідовність - не дуже ефективна практика, оскільки всі операції з пам'яттю повинні бути завершені, а нові операції затримуються до завершення старих.

З вільною координацією справи йдуть набагато краще, оскільки вона використовує апаратний аналог критичних розділів програмного забезпечення.

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		11

Якщо процес виходить за межі такої критичної області, це не означає, що всі записи повинні бути завершені негайно.

Все, що вам потрібно зробити, це заповнити всі записи, перш ніж процес знову потрапить у цю критичну область. У цій моделі операція синхронізації розділена на дві окремі операції:

Для того, щоб прочитати або записати спільну змінну, програмне забезпечення процесора має виконати операцію, щоб отримати ексклюзивний доступ до спільних даних.

Потім процесор може використовувати (читати або писати) ці дані на власний розсуд. Коли ви використовуєте комірку пам'яті, процесор виконує операцію звільнення змінної синхронізації.

Процес випуску не вимагає заповнення незавершених записів, але його не можна завершити, поки не будуть завершені всі раніше розпочаті записи. Перш за все, нові операції з пам'яттю можуть починатися негайно. Коли розпочинається наступна монопольна операція з ресурсами, перевіряється, чи всі попередні операції з випуску були завершені. Якщо ні, операція захоплення завершиться, поки не будуть виконані всі операції звільнення. Якщо наступний процес запису з'являється досить довго після останнього процесу випуску, йому не потрібно чекати, і він може затримати критичну область без затримки.

Ця схема трохи складніша, ніж слабка координація, але вона має одну головну перевагу: виконання команд не повинно затримуватися так часто, як при слабкій координації. Найпростіші мультипроцесори засновані на мережевому з'єднанні із загальною топологією шини. Два і більше процесори та один або більше модулів пам'яті використовують одну і ту ж шину для зв'язку. Коли процесору потрібно прочитати слово з пам'яті, він спочатку перевіряє, чи вільна шина. Коли шина звільняється, процесор вводить адресу потрібного слова на шину, подає кілька керуючих сигналів і чекає, поки пам'ять помістить потрібне слово на шину. Коли шина зайнята, процесор просто чекає її звільнення. Ця подія має проблему. З двома або трьома процесорами доступ до загальної шини повністю контролюється, але виникають труднощі з 32 або 64 процесорами.

Продуктивність системи буде повністю обмежена пропускнуою здатністю шини, і значна частина процесорів буде більшу частину часу простоювати.

Для вирішення цієї проблеми до кожного процесора додається кеш-пам'ять. Кеш-пам'ять може бути структурно розташована в мікросхемі процесора поруч із мікросхемою процесора на платі процесора.

Поєднання цих варіантів також дозволено. Оскільки тепер слова можна читати з кеш-пам'яті, загальне навантаження на шину менше, і система може підтримувати більше процесорів. через місцевий автобус. Для оптимального використання цієї конфігурації компілятор повинен мати всі тексти програм, константи та зберігати інші дані, стеки та локальні змінні лише для читання в модулях локальної пам'яті. Спільна пам'ять використовується лише для спільних змінних. У більшості випадків таке розміщення значно зменшує обсяг даних, що передаються по шині, і не вимагає активного втручання компілятора. Для всіх інших процесорів у системі включення кеш-блоків у систему створює механізм відстеження змін даних у кеші. Це тому, що кеш-блоки, що належать різним процесорам, можуть одночасно зберігати копії одного і того ж слова з основної пам'яті. І якщо один із власників такої копії вносить зміни, всі інші власники копії втрачають право користуватися нею, оскільки копія стає недійсною. Проблема, яка виникає, відома як послідовність кешу, є дуже важливою. Якщо ви не вирішите це, ви не зможете використовувати кеш, і кількість процесорів на загальній шині доведеться зменшити до двох або трьох. У наш час існує багато різних рішень проблеми. Хоча всі алгоритми, відомі як кешування, протоколи когерентності відрізняються деякими деталями, вони не всі дозволяють різні варіанти одного рядка в різних блоках кешу одночасно. У всіх випадках контролер кеш-пам'яті сконструйований таким чином, що вимоги до кеш-пам'яті перехоплюють шину і контролюють усі запити до неї від інших процесорів та інших кеш-блоків та вживають відповідних заходів для вирішення виникаючих проблем. Ці пристрої називаються кешами відслідковування або сховищами, оскільки вони відстежують шину.

Набір правил, що виконуються кешем, процесорами та основною пам'яттю, щоб запобігти появі різних варіантів даних у декількох блоках кешу, утворює

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		13

протокол когерентності кешу. Найпростіший протокол когерентності кешу називається наскрізним кешуванням.

Коли процесор намагається прочитати слово, якого він не має в кеші, контролер кеш-пам'яті завантажує рядок, що містить це слово, з основної пам'яті.

Основна пам'ять, яка надала рядок у цьому журналі, завжди оновлюється. Потім інформацію можна прочитати з кешу.

1.3 Переваги та недоліки мультипроцесорів

В основному існує два типи мультипроцесорів, симетричний та асиметричний.

Симетричні системи: у симетричних типах систем кожен процесор містить однакову копію операційної системи, і вони взаємодіють між собою. Усі процесори є одноранговими, тобто між ними не існує взаємовідносин ведучий-підлеглий. Версія Unix для комп'ютера Multimax є прикладом симетричної багатопроцесорної системи

Асиметричні системи: в асиметричних системах кожен процесор має певне завдання. Ця система має основний процесор, який передає інструкції всім іншим процесорам у мережі.

Асиметрична багатопроцесорна система включає підпорядковане з'єднання шаблону, тобто всі процесори в системі підпорядковані головному процесору. Порівняно з багатопроцесорними та однопроцесорними системами, перша має переваги перед останньою з точки зору швидкості та надійності. Надійність багатопроцесорної системи полягає в тому, що якщо один процесор виходить з ладу, система буде продовжувати функціонувати і виконувати завдання, хоча, звичайно, система гальмує, але все одно залишається швидшою, ніж одна процесорна система. З цією перевагою ми відразу знаходимо ще одну, яка впливає з необхідності багатопроцесорної системи.

При використанні ресурсів декількох процесорів завдання буде виконано в рази швидше, що дозволить раціонально використовувати час для виконання завдання. Використання багатопроцесорної системи збільшує продуктивність.

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		14

Пропускна здатність відповідає за збільшення продуктивності. Він збільшується так часто, як кількість процесорів, що працюють паралельно, тобто пропускна здатність кожного процесора додається, і це додавання призводить до швидшої обробки інформації.

Перевагою багатопроцесорної системи в довгостроковій перспективі є також ціна, оскільки процесори користуються периферійними пристроями, джерелами живлення та пам'яттю. Крім того, ця система матиме потужність, яка покриватиме попит довше, ніж ремонт. Цей параметр мінімізує подальші витрати на технічне обслуговування та модернізацію заліза. Основним недоліком багатопроцесорної системи є ціна, хоча багатопроцесорна система в перспективі дешевша, але початкова ціна висока, і в цьому відношенні придбання однопроцесорної системи виглядає більш привабливим.

1.4 Висновки

Під час перевірки параметрів багатопроцесорної системи було вирішено, що для складних обчислень система з багатьма процесорами та системами на базі одного процесора для простих завдань та завдань, що базуються на повсякденному використанні, тобто простої офісної роботи, використовується для входу в Інтернет перегляд відео-аудіофайлів та розваги за типом гри, оскільки розробники можуть краще оптимізувати свої проекти для однопроцесорних комп'ютерів, оскільки вони складають більшість комп'ютерного ринку.

Системи MPP або системи масової одночасності - це багатопроцесорні комп'ютерні системи (КС) з локальною пам'яттю. Їх головна перевага - необмежена масштабованість (можливість збільшення кількості процесорів у системі без зміни їх характеристик). У зв'язку з цим системи MPP можуть складатися з сотень, тисяч і навіть десятків тисяч процесорів. Отже, сьогодні найпродуктивніші системи, такі як Roadrunner, включають понад 192 тис. Процесорів, а Blue Gene / L - близько 213 тис. Процесорів.

Найважливішим аспектом систем MPP є те, як окремі процесори взаємодіють між собою. Конфігурація міжпроцесорної системи зв'язку суттєво

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		15

впливає на послідовність з'єднань ліній зв'язку, що з'єднують окремі процесори. Організація внутрішнього спілкування КС називається топологією. Залежно від того, чи конфігурація взаємозв'язку залишається незмінною (поки виконується завдання), існують CS зі статичною та динамічною топологією. У статичних CS структура взаємозв'язків фіксована. У CS з динамічною топологією конфігурацію взаємозв'язків можна швидко змінити за допомогою програмного забезпечення.

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		16

2 ТОПОЛОГІЯ ТОР

Топологія багатопроцесорної системи - це метод підключення вузлів процесора один до одного за допомогою каналів передачі даних. Топологію системи корисно подати у вигляді графіка, вершини якого відповідають вузлам процесора або ребрам - каналам зв'язку. Традиційно топології можна розділити на фіксовані, переконфігуровані, з одного боку, регулярні та нерегулярні - з іншого. Топологію системи корисно подати у вигляді графіка, вершини якого відповідають вузлам процесора, а ребра каналам зв'язку. З одного боку, умовні топології можна розділити на фіксовані, переконфігуровані.

Системи з топологією тора, які можуть бути побудовані на основі перетворювачів, можуть, однак, бути викликані до певної міри: для реалізації топології тора потрібно рівно чотири ланки, якщо це не призводить до посилення для підключення системи до комп'ютера, який контролює процес обчислень. При такій комбінації процесорів однорідність процесів передачі даних між ними порушується під час обчислень. Тому важко побудувати топологію "гіперкуба" з більш ніж 16 ($2^4 = 16$) процесорами або топологію "тривимірний куб" без групування процесорів у блоки, наприклад, якщо ми поєднуємо два процесори в блоки на основі спільної пам'яті, ми отримуємо обчислювальні вузли з вісьмома підключеннями .

Передачі даних між двома процесорами в мережі це важкий процес складність цього комунікаційного процесу може бути через заміну максимальної довжини шляху (діаметру мережі) у виразах на час передачі даних різними методами зв'язку

Решітка-тор

$$t_n + 2mtk[p/2] \quad t_n + mtk + 2tc[p/2]. \quad (2.1)$$

Передача повідомлень. Загальна схема алгоритму циклічного зсуву для топології мережевих воріт така. Нехай процесори мають нумерацію від 0 до $p-1$ на стовпчиках сітки. На першому етапі організовується циклічний зсув із кроком q

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		17

mod p, зокрема для кожної смуги (якщо повідомлення передаються через праві межі смуг, поки такий зсув здійснюється, компенсаційний зсув на 1 потрібен після кожна така передача вгору для процесорів у першій колонці сітки). На другому етапі здійснюється циклічний зсув вгору з кроком $[q / p-1 /]$ для кожної колонки сітки. Загальна тривалість усіх процесів доставки впливає із співвідношення:

$$t_{пд} = (t_n + mtk)(2[p-1/2/2] + 1). \quad (2.2)$$

Архітектура будь-якої багатопроцесорної комп'ютерної системи базується на можливості обміну даними між компонентами цієї операційної системи. Система зв'язку ОС - це мережа, вузли якої з'єднані шляхами передачі даних - каналами. Вузлами можуть бути процесори, модулі пам'яті пристроїв вводу-виводу, комутатори або більше цих елементів, згрупованих разом. Організація внутрішнього зв'язку комп'ютерної системи називається топологією. Будь-які два вузли, з'єднані каналом зв'язку, називаються сусідніми вузлами або сусідами. Кожен канал $c = (x, y) \in C$ з'єднує вихідний вузол (вихідний вузол) x з вузлом приймача (вузол приймача) y , де x, y дорівнює N . Вихідний вузол, який діє як початок каналу c , позначає s_c , а приймальний вузол - інший кінець каналу - r_c . Часто пари вузлів з'єднують два канали - по одному в кожному напрямку. Канал $c = (x, y)$ характеризується шириною (w_c або w_{xy}) - кількістю сигнальних ліній; Частота (f_c або f_{xy}) - бітрейт на кожній сигнальній лінії; Затримка (t_c або t_{xy}) - час, необхідний для передачі бітів від вузла x до вузла y .

Для більшості каналів затримка безпосередньо залежить від фізичної довжини лінії зв'язку (l_c) та швидкості поширення сигналу (v): $l_c = v t_c$. Пропускна здатність каналу b_c визначається виразом $b_c = w_c f_c$.

Залежно від того, чи конфігурація взаємозв'язків залишається незмінною хоча б під час виконання певного завдання, існують мережі зі статичною та динамічною топологіями.

У статичних мережах структура взаємозв'язків є фіксованою. У мережах з динамічною топологією в обчислювальному процесі конфігурацію взаємозв'язків

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		18

можна швидко змінити за допомогою програмного забезпечення або поєднати обидві ролі. У прямих мережах кожен вузол є одночасно кінцевим вузлом і комутатором, і повідомлення надсилаються безпосередньо між кінцевими вузлами.

У непрямих мережах вузол може бути терміналом або комутатором, але не одночасно, так що повідомлення передаються опосередковано за допомогою виділених вузлів комутатора. (Далі для простоти обидва варіанти називаються "прямими" та "непрямими" мережами, також скороченими замість кінцевого вузла, ми називаємо "терміналом".)

Кожен прямий MMZ може бути представлений як непрямий через поділяючи кожен вузол на дві частини - кінцевий вузол і комутаційний вузол.

Сучасні прямі мережі реалізовані таким чином - комутатор відокремлений від кінцевого вузла і розміщений у виділеному маршрутизаторі.

Головною перевагою прямого MMZ є те, що комутатор може використовувати ресурси термінальної частини свого вузла, що стає значним, якщо врахувати, що останні, як правило, включають комп'ютер або процесор. Три найважливіші атрибути MMZ: {1} • Стратегія синхронізації

- 1) стратегія комутації;
- 2) стратегія управління.

Дві можливі стратегії синхронізації мережі - синхронна та асинхронна. В синхронному MMZ всі дії жорстко координуються у термінах, що гарантується за рахунок рівномірного генератора тактових імпульсів (ГТІ), сигнали якого передаються на всі вузли одночасно.

Асинхронні мережі не мають єдиного генератора, а функції синхронізації розподілені по системі, а локальні ГТІ часто використовуються в різних частинах мережі. І в першому, і в другому варіанті здійснення інформація передається одним пакетом. Пакет представляє групу бітів, для яких також використовується термін повідомлення.

2.1 Створення топології

У мережах з комутацією зв'язку шлях від вузла джерела до вузла прийому формується шляхом відповідного встановлення елементів комутації мережі. доставлений, не досягне пункту призначення. Надсилання повідомлень між заданою парою вузлів завжди здійснюється поодиноці та за одним маршрутом.

Мережі з комутацією пакетів припускають, що повідомлення знайде шлях до місця призначення. На відміну від мереж із комутацією зв'язку, маршрут від початкової точки до пункту призначення може щоразу відрізнятись. Пакет послідовно рухається через мережеві вузли. Наступний вузол відзначає отриманий пакет в буфері проміжного сховища, аналізує його і робить висновки про те, що з ним робити далі. Залежно від завантаження мережі приймається рішення про можливість негайної переадресації пакета до наступного вузла і про подальший маршрут пакета на шляху до пункту призначення. Коли всі можливі шляхи зайняті для переміщення пакета на наступний вузол, у буфері вузла формується черга пакетів, яка «розсмоктується», коли лінії зв'язку між вузлами звільнюються (якщо черга також насичена, один стратегій маршрутизації зможе «Падіння хвоста, відхилення вхідних пакетів»)

MMZ також можна класифікувати відповідно до їх організації. У деяких мережах, особливо при переключенні з'єднань, використовується централізоване управління (рисунок 2.1.). Процесори надсилають запит на обслуговування одному мережевому контролеру, який арбітрує запити на основі встановлених пріоритетів і визначає бажаний маршрут. Цей тип повинен включати мережі з топологією шини. Також побудовані масиви процесорів як централізовані мережі управління, які виконуються за допомогою сигналів від центрального процесора. Наведена схема також застосовується до мереж із комутацією пакетів. Тег маршрутизації, що зберігається в заголовку пакета, визначає тут адресу вузла призначення. Більшість послідовних операційних систем мають такий тип управління.

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		20

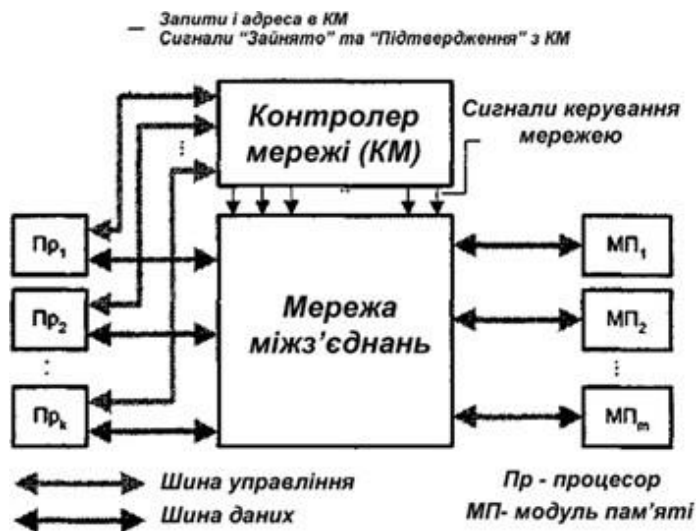


Рисунок 2.1.— Структура мережі з централізованим управлінням

У схемах з децентралізованим управлінням функції управління розподіляються між мережевими вузлами.

Централізований варіант легше реалізувати, але в цьому випадку розширення мережі пов'язане із значними труднощами. Децентралізовані мережі набагато гнучкіші щодо підключення додаткових вузлів, але взаємодія вузлів у таких мережах набагато складніша.

У деяких мережах зв'язок між вузлами здійснюється за допомогою декількох комутаторів, але є й мережі з одним комутатором. Наявність великої кількості комутаторів збільшує час передачі повідомлення, але дозволяє використовувати прості елементи комутації. Такі мережі зазвичай будуються в кілька етапів.

Методи опису властивостей мережеских з'єднань

При описі MMZ вони зазвичай характеризуються наступні параметри:

- 1) розмір мережі (N);
- 2) кількість з'єднань (I);
- 3) діаметр (D);
- 4) порядок вузлів (d);
- 5) пропускна здатність (Вт);
- 6) затримка (Т);
- 7) зв'язок (Q);

Зм.	Арк.	№докум.	Підпис	Дата

- 8) ширина вдвічі (B);
- 9) діапазони ділення (b) .

Розмір мережі чисельно дорівнює кількості вузлів, підключених мережею.

Кількість посилянь - це загальна кількість каналів між усіма вузлами мережі. Що стосується вартості, найкращою слід вважати мережу, яка вимагає меншої кількості з'єднань.

Діаметр мережі або відстань зв'язку визначають мінімальний шлях, який проходить повідомлення між двома найвіддаленішими вузлами мережі. Шлях у мережі - це впорядкований набір каналів $P = \{c_1, c_2, \dots, c_n\}$, після чого дані з вихідного вузла, які послідовно надходять від одного проміжного вузла до іншого, надсилаються до приймаючого вузла. Термін перехід використовується для позначення відрізка шляху між парою сусідніх вузлів. Мінімальний шлях від вузла x до вузла y - це шлях із мінімальною кількістю переходів. Якщо ми позначаємо кількість переходів по мінімальному шляху від вузла x до вузла y через $H(x, y)$, то діаметр мережі D є найбільшим значенням $H(x, y)$ серед усіх можливих комбінацій x і y . Таким чином, у ланцюжку з чотирьох вузлів найбільша кількість переходів відбувається між найвіддаленішими вузлами, а «діаметр» такого ланцюга - три. Зі збільшенням діаметра мережі загальний час повідомлення збільшується, тому розробники операційної системи намагаються робити якомога менше.

Порядок вузлів. Кожен вузол мережі пов'язаний з іншими вузлами набором каналів $Cx = Cx \cup C0x$, де $Cx = \{C: c_x = x\}$ - набір вхідних каналів, а $C0x = \{c: c_x = x\}$ - набір виходів це канали. Порядок вузла x - це сума кількості входів $|Cx|$ та джерело $|C0x|$ Канали вузлів, тобто рівні кількості вузлів мережі, до яких вузол безпосередньо підключений. У мережі, організованій у формі матриці, в якій кожен вузол має канали лише до найближчих сусідів (лівого, правого, верхнього та нижнього), порядок вузлів - чотири. Збільшення вартості цієї властивості додає складності мережевим комутаторам і, як наслідок, додаткові затримки в передачі повідомлень. З іншого боку, збільшуючи порядок вузлів, ви можете реалізувати топології з меншим діаметром мережі і тим самим скоротити час повідомлення. Якщо мережа з 4096 вузлів (212 чи 642) побудована за матричною схемою, її

діаметр становить 126, а для гіперкуба лише 12. Дизайнери операційних систем зазвичай віддають перевагу топологіям, в яких порядок усіх вузлів однаковий, що дає можливість модульної побудова мереж.

Пропускна здатність мережі характеризується кількістю інформації, яка може передаватися по мережі за одиницю часу. Зазвичай вона вимірюється в мегабайтах в секунду або гігабайтах в секунду, без витрат на перенесення надмірної інформації, такої як біти парності.

Затримка мережі - це час, протягом якого повідомлення проходить мережу. У мережах, де час передачі залежить від маршруту, вказується середня, мінімальна та максимальна затримка мережі.

Мережеве з'єднання можна визначити як мінімальну кількість вузлів або ліній, які повинні залишити систему, щоб мережа поділилася на дві різні мережі. Отже, підключення до мережі характеризує стійкість мережі до пошкодження, тобто її здатність забезпечувати роботу операційної системи у разі виходу з ладу мережевих компонентів ширини бісекції. Спочатку визначте поняття розділу мережі (розділу мережі). Зріз мережі $C(N_1, N_2)$ - це набір каналів, частка яких ділить набір вузлів мережі N на два різних набори вузлів N_1 і N_2 . Кожен елемент $C(N_1, N_2)$ є каналом, що з'єднує вузол набору N_1 з вузлом з N_2 . Бісекція мережі - це зріз мережі, який приблизно ділить її навпіл, тобто такий, що $|N_2| \approx |N_1| \approx |N_2| + 1$. Ширина бісекції B характеризується мінімальною кількістю каналів, які розриваються при всіх можливих бісекціях мережі:

Широта розподілу діляниць дозволяє оцінити кількість повідомлень, які можуть бути передані по мережі одночасно, за умови, що відсутні конфлікти через спроби використання одних і тих же вузлів або ліній зв'язку.

Пропускна здатність ділення - це найменша пропускна здатність на всіх можливих половинах мережі. Він характеризує пропускну здатність ліній зв'язку, які порушені в половині мережі, і дозволяє оцінити найгіршу пропускну здатність мережі при спробі одночасно передавати кілька повідомлень, коли ці повідомлення потрібно передавати з однієї половини мережі в іншу. Смуга поділу b визначається виразом $b = \min B(N_1, N_2)$. Для мереж з однаковою пропускну здатністю у всіх каналах b_s справедливим коефіцієнтом є: $b = b_s \times B$.

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		23

Невелике значення діапазону ділення вказує на можливість конфліктів при одночасному відправленні декількох повідомлень.

Статичні топології від MMZ включають ті, у яких можливий лише прямий фіксований шлях між двома вузлами, тобто для статичних топологій не потрібне комутаційне обладнання в мережі. Якщо такі пристрої існують, вони використовуються лише до виконання певного завдання, і протягом усього часу обчислень топологія MMZ залишається незмінною

З можливих критеріїв класифікації статичних мереж часто вибирають їх розмірність. Розрізняють ці положення:

- 1) одновимірні топології (лінійний масив);
- 2) двовимірні топології (кільце, зірка, дерево, сітка, масив систоли);
- 3) тривимірні топології (повністю пов'язана топологія, акордне кільце).

2.2 Висновок

Ми вияснили що багатопроцесорна система тор кращі є досконаліша ніж однопроцесорні системи, так як швидкість та якісна праця більша у декілька разів. За допомогою цього пам'ять комп'ютера не перегружається і система спокійно виконує свої функції тому, що за допомогою такої топології комп'ютеру легше опрацьовувати задачі.

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ

3.1 Розподільна система

Розподілені системи - це мережі, основною характеристикою яких є наявність декількох центрів обробки даних, за допомогою яких можна виконувати паралельні обчислення. До розподілених систем належать багатопроцесорні комп'ютери, багатомашинні системи, комп'ютерні мережі.

Багатопроцесорний комп'ютер - це комп'ютер із декількома процесорами, кожен з яких самостійно запускає власну програму, а взаємодія між ними здійснюється за допомогою спільної пам'яті.

Усі процесори мають загальну операційну систему, яка контролює навантаження між процесорами. Мультипроцесор не підтримує територіальний розподіл - усі його блоки були б в одному дизайні. Найголовніша особливість - висока продуктивність завдяки паралельній роботі процесорів, відмовостійкість.

Багатомашинна система - це комп'ютерна система, яка містить кілька комп'ютерів, кожен з яких працює зі своєю операційною системою.

Комп'ютерне програмне та апаратне забезпечення забезпечує роботу всіх комп'ютерів в цілому. Ці інструменти дуже швидкі і містять програмні модулі, які розподіляють обчислювальне навантаження та синхронізують розрахунки. У багатомашинних системах досягається високий рівень відмовостійкості та висока продуктивність (але менша, ніж у багатопроцесорних через слабкіші зв'язки між процесорами).

Комп'ютерна мережа - група комп'ютерів, пов'язаних за допомогою передачі даних.

За допомогою зв'язку комп'ютер отримує доступ до інших ресурсів, таких як дані, периферійні пристрої (принтери, модеми, факсимільні апарати тощо); ця властивість відома як прозорість мережі. Використання комп'ютерних мереж дає компанії переваги, такі як розподіл високоякісних ресурсів, покращений зв'язок,

покращений доступ до інформації, швидке та якісне прийняття рішень, територіальний розподіл комп'ютерів.

Залежно від місця розташування комп'ютерів мережі поділяються на локальні та глобальні мережі.

Глобальна мережа може включати інші такі мережі, локальні мережі, окремі віддалені комп'ютери. Глобальні мережі в свою чергу можуть бути міськими, регіональними, національними або транснаціональними. Елементи таких мереж можуть розташовуватися на значній відстані один від одного.

Локальні мережі (LAN) об'єднують комп'ютери, що знаходяться на відстані не більше кількох кілометрів, пов'язані високошвидкісними лініями зв'язку, які обмінюються програмними та апаратними ресурсами. Як правило, локальні мережі створюються всередині організації. Такі мережі також відомі як корпоративні системи.

Локальні обчислювальні мережі. Обладнання локальної мережі. Найпростіший тип локальної мережі - це комп'ютер, який має вбудовану мережеву карту і пов'язаний між собою кабелем. Складніша структура - це зв'язок між комп'ютерами (робочими станціями) та слабким комп'ютером (сервером). Основними компонентами комп'ютерної мережі є робочі станції, сервери, інтерфейсні карти та кабелі.

Комп'ютерні мережі складаються з вузлів (окремої фізичної одиниці мережі), які можуть бути комп'ютером, принтером або іншим пристроєм, підключеним до мережі. Комп'ютери поділяються на два типи: робочі станції, на яких працюють користувачі, та сервери, що обслуговують ці станції.

Сервери використовуються для об'єднання ресурсів комп'ютерної мережі та їх розподілу між клієнтами. Це може бути досить потужний комп'ютер або кілька комп'ютерів, один з яких є основним сервером, а решта - резервною копією або логічним розширенням основного сервера. Група серверів і підключених до них комп'ютерів є доменом.

Існують файлові сервери (для роботи з файлами комп'ютерної мережі), сервери друку, сервери зв'язку (для функцій факсу або модему) тощо.

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		26

Файловий сервер обслуговує всю комп'ютерну мережу, тому його диски повинні мати досить якісні характеристики - великий об'єм пам'яті, тривалий термін служби. Файлові сервери можуть використовувати іншу операційну систему, ніж робочі станції, наприклад NetWare Novell. Комп'ютери, які можуть одночасно виконувати функції файлового сервера і робочої станції під час роботи в мережі, утворюють однорангову комп'ютерну мережу, тобто таку, в якій всі вузли мережі отримують однаковий пріоритет, а ресурси кожного вузла доступні. пов'язані з іншими вузлами мережі. У комп'ютерних мережах цей тип файлового сервера може бути виділеним (якщо він виконує лише серверні функції). Клієнт-серверна мережа - це мережа, в якій одні комп'ютери виконують роль серверів, а інші - клієнтами.

Суперсервер комп'ютерної мережі має систему дублікатів дисководів для забезпечення надійного зберігання даних. У таких системах файли, записані на одному томі, автоматично відображаються на резервному томі. Якщо один жорсткий диск виходить з ладу, інформація негайно зчитується іншим, створюючи високий рівень надійності.

Для проксі-сервера, який виконує кілька функцій, передбачений потужний комп'ютер з великою ємністю оперативної пам'яті та жорстким диском. З одного боку, це функція кешування, що означає, що сервер зберігає веб-сайт на жорсткому диску і більше не отримує доступу до нього наступного разу, що значно прискорює його роботу. По-друге, це функція обмеження доступу користувачів мережі до певних веб-сайтів (фільтрація). Ще однією функцією проксі-сервера є ведення журналу, що зберігає дату відвідування веб-сайту, ім'я користувача, який його активував, назву сайту та дзвінок тощо.

Мости використовуються для об'єднання двох окремих комп'ютерних мереж або підмереж в одну мережу - пристроїв, що складаються з апаратного та програмного забезпечення і є необхідним елементом для підвищення ефективності, безпеки та діапазону мережі. Мости можуть фільтрувати пакети даних відповідно до визначених критеріїв оптимальності, підключати дві комп'ютерні мережі одного типу з різною швидкістю передачі даних, використовуватись у мережах з різною топологією та носіями передачі та

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		27

обмежувати кількість робочих станцій, які отримують та надсилають інформацію з інших мереж.

У великих комп'ютерних мережах доцільно використовувати маршрутизатор, щоб вибрати найкращий спосіб передачі пакетів даних. Існують статичні маршрутизатори, в яких маршрут вибирається в таблицях маршрутизації, створених адміністратором комп'ютерної мережі, а динамічні маршрутизатори, які за допомогою спеціальних алгоритмів обчислюють кількість переходів (тобто кількість маршрутизаторів), вибирають найкращий і постійно аналізують стан каналу, відправляти пакети з описом їх підключень до інших маршрутизаторів, адаптувати власні таблиці маршрутизації.

LAN-кабель. Комп'ютер і мережевий сервер з'єднані кабелями, які можуть бути:

Багатоядерні, найчастіше використовуються в мережевих конфігураціях, таких як "кільце" та "зірка". Недоліки таких кабелів:

- 1) захист від електромагнітного випромінювання;
- 2) низька швидкість передачі даних;
- 3) висока вартість.

У вигляді крученої пари - екранованої оболонки, яка містить одну або кілька пар провідників і, у поєднанні з коаксіальним кабелем, кабель може передавати високі швидкості передачі даних на короткі відстані.

Коаксіальний кабель, що складається з центрального провідника, ізолюючого шару, екрану (шари металевого циліндра або дротяної сітки), ізолюючого шару, працює швидше. Залежно від товщини розрізняють товсті і тонкі коаксіальні кабелі. Усі вони характеризуються низькою вартістю, відносно високою швидкістю передачі даних та безпекою та використовуються в мережах Ethernet.

Скляні волокна, які забезпечують передачу даних на дуже високих швидкостях на великі відстані (до 4,5 км), мають високий рівень безпеки, але коштують досить дорого.

Мережевий адаптер. Комп'ютер підключений до кабелю через інтерфейсні карти - мережеві адаптери, призначені для використання одного з низькорівневих

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		28

протоколів (Ethernet, Token Ring, FDDI, ARCnet тощо). Вони можуть бути двох типів: виявлення зіткнень та передача маркера. Мережеві адаптери, що підтримують програмне забезпечення, виконують такі основні операції під час передачі (зворотна дія - під час отримання) повідомлень.

Передача даних. Дані передаються з оперативної пам'яті в адаптер або навпаки через канал довільного доступу, програмований канал вводу-виводу;

Буферизація. Коли дані обробляються в мережевому адаптері, вони зберігаються в буфері, що дозволяє адаптеру отримати доступ до цілого пакета і повинен мати таку ж ємність, як і весь пакет даних. Використання буферів необхідно для координації швидкості обробки інформації різними компонентами локальної мережі.

Формування пакетів або мережевий адаптер ділить дані на частини (при отриманні - з'єднує їх), розмір яких залежить від типу мережі (в Ethernet - 1 КБ, у Token Ring - 4 КБ), додає заголовок і кінець до пакетів, тобто створює пакет так, готовий до передачі:

- 1) кабельний доступ. Перед передачею даних адаптер чекає отримання маркера або гарантує, що лінія не зайнята;
- 2) перетворення даних. Дані передаються послідовно, побітно, по кабелю, перетворюючи їх з паралельних на послідовні;
- 3) кодування (декодування) даних, тобто формування електричних сигналів, які використовуються для передачі даних;
- 4) прийом (передача) імпульсів. Дані у вигляді закодованих електричних сигналів передаються по кабелях.

Адаптери відрізняються методами доступу до середовища та протоколами і мають такі основні характеристики:

- 1) швидкість передачі даних;
- 2) тип комп'ютерної шини;
- 3) розряд бітів (8, 16, 32, 64 біт);
- 4) ємність буфера для пакетів даних;
- 5) сумісність з різними мікропроцесорами;
- 6) топологія - конфігурація з'єднання елементів у мережі.

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		29

У комп'ютерну мережу також можуть бути включені такі елементи:

- 1) повторювач - для підключення окремих мережевих компонентів;
- 2) джерело безперебійного живлення - для забезпечення відключення та зберігання даних у разі аварійного збою живлення;
- 3) трансивер - використовується для підключення комп'ютера до товстого коаксіального кабелю;
- 4) роз'єми - для підключення мережевих адаптерів за допомогою тонкого коаксіального кабелю.

Протоколи, мережеві інтерфейси. Мережеві комп'ютери обмінюються інформацією, організованою в пакети повідомлень, які складають основу, на якій побудована локальна мережа. Мережевий адаптер приймає та відправляє пакети даних під контролем відповідного програмного забезпечення, потім пакети адресуються робочим станціям, кожна з яких повинна мати унікальну адресу в мережі. У різних системах комп'ютерних мереж пакети даних визначаються по-різному, але вони мають такі спільні елементи:

- 1) унікальна адреса відправника;
- 2) унікальна адреса одержувача;
- 3) функція, що характеризує зміст пакету;
- 4) дані або повідомлення.

Контрольна сума (CRC) на виявлення помилок передачі даних. Вузол, який отримав повідомлення, повинен обчислити цю суму і порівняти результат із вмістом пакета.

Основна схема пакету повідомлень така:

Адреса відправника Адреса одержувача Тип пакету Дані повідомлення CRC

Протоколи та інтерфейси використовуються для опису взаємодії між програмними та апаратними елементами.

Протокол - набір правил взаємодії об'єктів одного рівня при обміні даними між станцією, що відправляє, та станцією, що приймає; формати блоків переданих даних; Контроль помилок, методи кодування інформації тощо.

Інтерфейс - опис процесу взаємодії об'єктів на сусідніх рівнях (наприклад, системах і середовищах) з метою контролю фізичної передачі даних, а також переданих форматів інформації.

Довідкова модель взаємодії відкритих систем OSI.

Для забезпечення обміну даними між комп'ютерними мережами були розроблені міжнародні багаторівневі стандарти протоколів, які відомі як еталонна модель Open System Interconnection (OSI). Основним завданням цієї моделі є спрощення та полегшення обміну інформацією з різним програмним та апаратним забезпеченням.

Довідкова модель визначає сім функціональних рівнів, кожен з яких відповідає окремій фізичній або логічній частині комп'ютерної мережі і підтримує роботу вищих рівнів.

Фізичний рівень описує фізичне середовище мережі (мідні лінії, волоконна оптика, космічні супутники тощо). Цей рівень приймає дані без поділу на кадри, тобто лише послідовність бітів. Залежно від типу мережі цей потік даних може бути паралельним або послідовним, а передача інформації в каналі зв'язку - дуплексною (одночасно в обидва боки), напівдуплексною (поперемінно у двох напрямках), симплексною (в одному напрямку). На цьому рівні встановлюються ретранслятори для скасування сигналу, що представляють собою інтерфейс між мережевими комп'ютерами та дискретним середовищем передачі сигналу, для якого визначаються початок і кінець кадру та формуються і приймаються сигнали певної фізичної природи:

1) каналний рівень перетворює бітовий потік фізичного рівня у кадри або пакети, що містять інформацію про адресу. На цьому рівні також перевіряється правильність передачі даних, яка може бути повторно передана;

2) мережевий рівень забезпечує передачу мережних пакетів інформації між вузлами мережі, при цьому виконуються проблеми вибору маршруту з числа можливих рішень, управління вхідним потоком і буферизація пакетів. Маршрутизатори використовуються для вибору оптимального типу доставки інформації.

Транспортний рівень відповідає за доставку повідомлень згідно з логічними адресами протоколу і ділить інформацію на менші пакети під час передачі та збирає повідомлення з пакетів під час прийому. На цьому рівні різні рівні мережі координуються за допомогою шлюзів (для мереж різного типу) та мостів (для мереж одного типу).

Рівень сесії дозволяє організувати сеанси спілкування між об'єктами вищого рівня з повною передачею інформації. Цей рівень визначає спосіб взаємодії двох функцій між двома віддаленими системами:

- 1) контроль діалогу, що визначає початок сеансу, передачу повідомлення і після закінчення сеансу - переривання спілкування;
- 2) розділення даних, що дозволяє вставляти покажчики в повідомлення для кожної робочої станції про початок і кінець повідомлення;
- 3) рівень даних описує методи перетворення інформації (з одного формату в інший, шифрування, кодування, стиснення тощо), яка передається на рівень програми.

Прикладний рівень відповідає за передачу інформації від користувача до будь-якого мережевого ресурсу, який потребує цього. Повідомлення, що передається мережею, надходить у модель OSI, надходить на фізичний рівень, надсилається на іншу робочу станцію, передається з фізичного рівня в зворотному напрямку, щоб через свій прикладний рівень дійти до прикладної програми іншої робочої станції. Залежно від параметрів встановлених протоколів на цьому рівні визначається продуктивність комп'ютерної мережі в цілому.

Топологія локальної мережі. Топологія - спосіб підключення комп'ютерів до мережі. Фізична топологія описує фізичну структуру компонентів локальної мережі, логічна - характеризує метод передачі пакетів даних, метод зв'язку, принцип управління помилками в мережі тощо.

Існує три найпоширеніші типи топологій локальної мережі: "загальна шина", "зірка", "кільце", і вибір буде залежати від структури офісу, засобів усунення несправностей, витрат на встановлення та типу використовуваного кабелю.

При організації локальної мережі за топологією "Загальна шина" всі комп'ютери з'єднуються послідовно кабелем, як правило, коаксіально. Коли використовується товстий коаксіальний кабель, мережа має центральну шину, до якої комп'ютери підключаються тонким коаксіальним кабелем з трансивером (пристрій, що з'єднує тонкі і товсті кабелі). При використанні тонкого коаксіального кабелю всі мережеві пристрої підключаються безпосередньо, що значно спрощує роботу. Такі локальні мережі є високоефективними, полегшують підключення нових станцій, керують процесом передачі даних, але переривання зв'язку між двома каналами призводить до виходу з ладу всієї мережі.

Мережа із зірковою топологією містить центральний вузол (концентратор), до якого підключені всі комп'ютери. Кабелі до робочих станцій проходять радіально від маточин, створюючи зіркоподібну конструкцію. Ці локальні мережі також відомі як пасивні зірки. Вони мають високий рівень експлуатаційної надійності; якщо одна робоча станція виходить з ладу, інші можуть обійти її через концентратор або багатопортовий трансивер, від якого виходять кабелі для цих станцій. Однак такі мережі досить дорогі, оскільки використовують багато кабелю. Варіант зіркоподібної топології локальної мережі - це "пов'язана або розподілена" зірка, в якій кілька пасивних зірок з'єднані концентраторами, що забезпечують обмін інформацією.

Топологія "кільця" складається з локальних комп'ютерів, укладених у кільце і з'єднаних парою кабелів. Вартість таких мереж досить висока.

Як правило, локальні мережі створюються за допомогою комбінації розглянутих вище топологій.

Методи доступу до каналів мережевого зв'язку реалізують функції рівня каналу моделі OSI (їх також називають протоколами низького рівня).

Метод доступу до Ethernet. Підтримує топологію шини LAN. Шина - це канал передачі даних, окремі частини якого називаються сегментами. Сегмент мережі може містити до 100 робочих станцій; кілька таких сегментів можна з'єднати повторювачами. В одній мережі Ethernet можна експлуатувати до 1024 робочих станцій. За допомогою цього методу організації доступу передача інформації не розпочинається, поки не буде доступний безкоштовний канал; в

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		33

інших випадках передача затримується, тобто реалізується процедура трансляції. Швидкість такої мережі буде знижена, якщо працює більше 100 комп'ютерів.

Метод доступу до Arcnet. Використовується в топології зірок. Повідомлення надсилаються на комп'ютери за допомогою маркерів. Коли повідомлення надсилаються, до них додаються маркери, і ця інформація надсилається через мережу. Після отримання повідомлень аналізується заголовок маркера, і якщо адреса збігається, повідомлення відокремлюються і залишаються на комп'ютері.

Метод доступу до токен-кільця. Використовується в топології "Кільце". Подібно до методу Arcnet, але тут ви можете встановити пріоритет для окремого комп'ютера для отримання маркера.

Метод доступу до FDDI. Інтерфейс розподілу волоконно-оптичних даних працює за схемою передачі маркера в логічному кільці з волоконно-оптичними кабелями і є високопродуктивним.

Вищезазначені протоколи працюють на найнижчому рівні локальної мережі, тобто вони реалізують передачу повідомлень (пакетів) між комп'ютерами, але вони не контролюють процес пересилання файлів, не мають засобів для забезпечення правильного порядку даних передача, можливість зворотного виявлення додатків - грамів, що вимагають обміну даними. Існують також протоколи середнього рівня, які відповідають транспортному рівню моделі. Сюди входять такі протоколи, як NetBIOS, IPX / SPX, TCP / IP, які виконують такі загальні функції.

Ініціюйте спілкування. Кожен протокол має можливість ідентифікації робочої станції за іменем, номером або іменем та номером, після чого дані обмінюються між вузлами. Для цього використовуються наступні типи діалогів; Дейтаграми - повідомлення, які не потребують підтвердження від сторони, яка отримує дані; Сеанси - повідомлення, для яких встановлений логічний зв'язок для обміну і отримання яких гарантовано.

Надсилання та отримання даних. Протокол забезпечує засоби надсилання та отримання повідомлень робочими станціями одержувача та відправника, в той час як можна використовувати обмеження щодо тривалості повідомлень;

Завершення обміну. Найпоширенішим протоколом (набором протоколів) є TCP / IP (Transmission Control Protocol / Internet Protocol). Протокол IP забезпечує обмін дейтаграмами між мережевими вузлами, а протокол TCP встановлює зв'язок між двома вузлами (це гарантує доставку повідомлення). Існують також стандартні утиліти, такі як передача файлів (FTP), дистанційне керування програмами (TELNET), обмін електронною поштою (SMTP).

Програмне забезпечення локальної мережі. Системне програмне забезпечення, яке управляє процесами в локальній мережі та поєднує їх із загальною архітектурою, протоколами зв'язку та механізмами взаємодії обчислювальних процесів, називають мережевими операційними системами.

Розглянемо властивості деяких таких систем.

Microsoft Windows NT Advanced Server - це середня мережева операційна система, заснована на структурі домену. Домен - це набір, який може містити до 10 000 об'єктів (комп'ютери, користувачі, групи користувачів, файлові об'єкти, принтери). Локальна мережа може складатися з одного або декількох доменів, між якими встановлюються довірчі відносини, тобто один домен може використовувати ресурси іншого. Ця система характеризується підтримкою веб-серверів, засобами ефективного архівування даних, розподілом (тиражуванням) даних на інші мережеві сервери, захистом доступу за допомогою ідентифікаційного коду.

Novell NetWare - це операційна система на основі служб каталогів (NDS), яка забезпечує глобальну основу для всіх мережевих ресурсів, які в свою чергу організовані як багаторівнева ієрархія. База даних логічно розділена на розділи, які розподіляються між мережевими серверами, забезпечуючи більш надійний доступ до мережевих ресурсів. Також існує система підтримки веб-серверів, засоби управління каталогами, реальна підтримка зв'язку з UNIX, архівування, тиражування даних, доступ до яких організований за допомогою кодів безпеки. За допомогою цієї операційної системи (версія 4.0) до 1000 користувачів можуть одночасно працювати та отримувати доступ до 54000 файлових серверів. Система захисту організована чотирма кільцями, в кожному з яких програмне забезпечення має певні привілеї.

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		35

UNIX - це гнучка, стабільна та безпечна операційна система для великих локальних мереж із серверною системою, розробленою для багатьох користувачів. Мережева файлова система дозволяє обмінюватися файлами та ресурсами з декількох серверів на різних платформах. Ще однією особливістю UNIX є Мережева інформаційна система (NIS), яка управляє мережевими каталогами та містить паролі, списки груп користувачів, фізичні адреси, IP-адреси тощо. Система також забезпечує вбудовану підтримку інтрамережі, надмірного захисту на рівні об'єкта та інструментів обслуговування файлів. При виборі операційної системи слід враховувати наступні параметри:

- 1) продуктивність залежить від кількості комп'ютерів в мережі;
- 2) надійність мережі;
- 3) ємність оперативної пам'яті та місця на жорсткому диску, необхідних мережевим пристроям;
- 4) підтримувана топологія мережі;
- 5) захист інформації від несанкціонованого доступу;
- 6) взаємодія з інтерфейсами інших мереж;
- 7) можливість використання декількох серверів у мережі.

Робота в інтернет-середовищі. Основні положення. Інтернет - це мережа комп'ютерів та комп'ютерних мереж, джерело інформації, яке постійно змінюється та розширюється. З часу свого створення в 60-х роках ХХ ст. Кількість комп'ютерів, підключених до Інтернету, продовжує зростати, і зараз ця мережа використовується мільйонами людей у комерційних та державних організаціях, а також окремо.

Для роботи в Інтернеті розроблені спеціальні програми (браузери). Найчастіше використовуються браузери Internet Explorer, Netscape Communicator та інші.

Комп'ютери підключаються до Інтернету телефонними лініями за допомогою модему, який перетворює цифрові комп'ютерні сигнали в аналогові телефонні лінії та навпаки. Модем характеризується швидкістю передачі даних, що вимірюється в бодах, і добуток цих параметрів називається пропускною

здатністю модему, яка виражається в бітах в секунду (bps). Сучасні модеми мають пропускну здатність більше 56 Кбіт / с.

Кожному комп'ютеру в Інтернеті присвоюється унікальна мережева адреса - IP-адреса, що складається з 32-розрядних цифр. Він перетворюється на доменне ім'я для зручності використання.

Адреса користувача складається з двох частин: їх ідентифікатора та доменного імені, розділених символом @. Доменне ім'я, у свою чергу, може складатися з кількох частин, розділених крапкою. Починаючи з найнижчого рівня, ці частини утворюють ієрархічну структуру. Наприклад IP-адреса:

- 1) Іваненко@kpei.kiev.ua;
- 2) виступає за;
- 3) Іваненко - ім'я комп'ютера користувача;
- 4) kpei - назва організації;
- 5) Київ - назва міста;
- 6) ua - скорочена назва країни.

Ви також можете використовувати такі функції в Інтернеті:

- 1) електронна пошта - обмінюйтесь електронними листами з іншими користувачами, до яких ви можете додавати файли, якщо це необхідно;
- 2) група новин або конференц-дзвінок (UseNet);
- 3) знаходити та передавати файли між двома віддаленими комп'ютерами (FTP);
- 4) віддалене керування комп'ютером (TelNet);
- 5) пошук і передача текстових файлів за допомогою гіперпосилань (WWW).

Робота з модемом. Модем використовується для роботи в мережі комп'ютерів, які обмінюються інформацією по телефонних лініях. Модем може бути зовнішнім пристроєм, що з'єднує телефонну мережу з комп'ютером, а може бути вбудованою в комп'ютер друкованою платою.

Зовнішній модем підключений до комп'ютера через кукурудзяний порт. Після цього вам потрібно налаштувати програмне забезпечення, виконавши наведені нижче дії:

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		37

1) активуйте папку Панель управління (двічі клацніть піктограму Мій комп'ютер або скористайтесь командами Пуск, Налаштування, Панель управління);

2) двічі клацніть піктограму установки обладнання, у наступних вікнах увімкніть опцію автоматичного пошуку модему (після аналізу наявного обладнання система розпізнає встановлений модем за замовчуванням);

3) у наступному вікні за допомогою кнопки Edit 1 активуйте вікно, в якому спочатку вибираєте виробника, а потім відповідний тип існуючого модему зі списку;

4) наступним кроком є встановлення програми віддаленого доступу, яка є в системі, але встановлення операційної системи не автоматично встановлює Windows. Отже, у папці Панель керування, вкладці Встановлення Windows, списку компонентів, категорії Зв'язок та кнопки Композиція активуйте вікно Додавання та видалення програм.

5) у наступному вікні, що відображає список відповідних компонентів, вам слід активувати компонент віддаленого доступу до мережі;

6) після того, як ви активуєте кнопку ОК і встановите на свій комп'ютер дистрибутив Windows, обрана програма стане доступною для роботи з модемом;

7) наступним кроком є увімкнення протоколу TCP / IP в системі підтримки Інтернету. Для цього активуйте вкладку Конфігурація у вікні Мережа папок на Панелі керування та натисніть кнопку Додати, щоб встановити необхідний протокол. У наступному вікні Вибір типу компонента потрібно вибрати компонент Pro-Protocol та активувати кнопку Додати.

В результаті на екрані з'являється вікно вибору:

1) мережевий протокол який ліворуч містить список виробників (активуйте Microsoft), виберіть зі списку праворуч протокол TCP / IP та активуйте кнопку ОК;

2) наступним кроком є присвоєння комп'ютеру IP-адреси, що складається з чотирьох цифр плюс адреса DSN сервера. Для цього активуйте вікно Мережа, виберіть протокол TCP / IP і активуйте кнопку Властивості;

3) вкладка IP-адреса дозволяє автоматично отримати IP-адресу (потрібно активувати відповідний перемикач). У цьому випадку користувач отримує динамічну адресу, тобто будь-яку адресу, яка є безкоштовною на сервері постачальника на момент з'єднання з користувачем. Якщо ви використовуєте постійну IP-адресу, надану провайдером, потрібно встановити прапорець Явно вказати IP-адресу та ввести адресу;

4) на вкладці Конфігурація DSN введіть адресу сервера DSN, який браузер використовує для пошуку адрес в Інтернеті.

5) наступним кроком є встановлення зв'язку між сервером провайдера та комп'ютером RAS. Для цього потрібно активувати створене з'єднання у вікні віддаленого доступу (кнопка «Пуск», «Налаштування», «Панель управління»);

6) введіть ім'я користувача (логін), надане постачальником, для входу на сервер провайдера;

7) створіть пароль (введені символи у вікні будуть замінені зірочками, тому ви повинні зробити символи чутливими до регістру та розкладкою клавіатури - українською, англійською тощо);

8) перевірити (або ввести новий) номер телефону, за яким програма буде підключатися;

9) активуйте кнопку Connect.

Під час з'єднання телефонний номер набирається автоматично, що позначається блимаючими індикаторами модему, а на екрані з'являється вікно.

У разі успішного з'єднання з провайдером, коли на екрані з'являється вікно Підключений потрібно активувати кнопку Закрити.

Після встановлення бажаного з'єднання на панелі завдань з'являється символ підключення, який активується для:

1) закінчити з'єднання - за допомогою кнопки Кінець з'єднання;

2) відображення поточної інформації (швидкість передачі даних, тривалість програми, кількість отриманої та надісланої інформації).

Електронна пошта. Outlook Express, Netscape Messenger, Internet Mail використовуються для роботи з електронною поштою.

Створюйте та надсилайте повідомлення. Щоб створити електронне повідомлення в Netscape Messenger, потрібно активувати команди Повідомлення, Скласти повідомлення. В результаті на екрані з'являється вікно креслення, в якому:

- 1) у полі До 1 введіть адресу електронної пошти одержувача повідомлення;
- 2) визначити тему повідомлення в полі Тема 4, що спрощує перегляд отриманих повідомлень користувачем;
- 3) унизу вікна 3 складіть текст повідомлення;
- 4) у вікні є панель 2 для простого форматування тексту (вирівнювання, розмір шрифту, створення списку тощо);
- 5) потім активуйте команду Надіслати повідомлення з меню Файл.

Перевірте свою поштову скриньку. Щоб отримувати сповіщення в Netscape Messenger, потрібно активувати команди Файл, Отримати нову пошту або Кнопка Отримати повідомлення. Потім на екрані з'являється вікно зі списком отриманих повідомлень у верхньому правому куті (якщо таких немає, відображається відповідне повідомлення). Кожен запис у цьому списку містить заголовок листа, дату та час його відправлення та тему листа. Усі непрочитані повідомлення виділено жирним шрифтом. Щоб переглянути лист, потрібно двічі клацнути на ньому, після чого ви можете переглянути його в окремому вікні.

Щоб скласти відповідь автору листа, потрібно активувати команду Повідомлення, Відповісти автору та відповісти всім, хто отримав такий лист - команду Повідомлення, Відповісти всім.

Створення адреси в адресній книзі. Для полегшення роботи з електронною поштою існує функція ведення адресної книги, в якій зберігаються найбільш часто використовувані адреси. Цю адресу можна використовувати під час складання повідомлень.

Щоб зберегти адресу для подальшого використання, відкрийте адресну книгу (команда Communicator, Address book). З'являється на екрані дисплея.

Потім відкрийте потрібну адресну книгу зі списку «Каталог» 1 або створіть нову, активуйте команди Файл, Створити адресну книгу. Створити нова адреса (картка) активує файл команд, створює картку або карту кнопок.

У вікні Новая карточка послідовно заповнюють усі потрібні атрибути (наприклад, прізвище, ім'я, по батькові, адреса електронної пошти, робочий телефон, домашня адреса тощо).

Для використання збереженої адреси активізують команди Сообщение, Выбрать получателей. У вікні Выбор получателей, що з'являється на екрані дисплея, із переліку вибирають потрібну адресу й активізують кнопку ОК.

Передача файлів по мережі (FTP). FTP (File Transfer Protocol) – це протокол передачі файлів, що дає змогу передавати файли з одного комп'ютера на інший. Файлами можуть бути текстові документи, електронні таблиці, програми, графічні та звукові файли.

Ця процедура виконується за допомогою програми FTP, яка дає можливість здійснювати пошук файлів, пересилати їх, виконувати інші дії на комп'ютерах. Файли зберігаються на FTP-серверах, клієнт-програма або Web-браузер за допомогою запиту дістає доступ до потрібних файлів. Щоб мати доступ до FTP-сервера, користувачеві необхідно зареєструватися, одержати своє вхідне ім'я (login) і пароль (password). Доступ до багатьох FTP-серверів, які містять архіви файлів, дозволено всім бажаючим. Ці сервери називаються серверами з анонімним доступом. Для реєстрації на такому сервері у полі із вхідним іменем (login) треба ввести ідентифікатор anonymous, а як пароль задати e-mail адресу.

Після реєстрації на екрані дисплея з'являється вікно програми WS_FTP, в якому в лівій частині виводяться зміст комп'ютера користувача та набір керуючих кнопок, а в правій – зміст FTP-сервера і також набір керуючих кнопок. У нижній частині вікна виводяться повідомлення, одержані з сервера.

Керуючі кнопки забезпечують такі дії:

- 1) ChgDir – перейти в іншу папку;
- 2) Mkdir – створити нову папку;
- 3) View – переглянути вибраний файл;
- 4) Exec – виконати вибраний файл;

- 5) Rename – перейменувати вибраний файл;
- 6) Delete – вилучити файл (або виділені файли);
- 7) Refresh – повторно активізувати файл;
- 8) DirlInfo – переглянути список файлів.

Після завершення роботи з FTP-сервером слід активізувати кнопку Close вікна WS_FTP, при цьому встановлений зв'язок буде розірвано.

Пошук інформації. Із збільшенням обсягу інформаційного наповнення Internet проблема пошуку інформації стає все більш актуальною. Потужним інструментом для вирішення цієї проблеми є створені пошукові системи, які допомагають знайти потрібні сервери або окремі Web-сторінки.

Існуючі пошукові системи різняться підходами, які застосовуються для збирання та оброблення інформації, організації БД із можливостями, які надаються користувачам для формування запитів і проведення пошуку в БД.

Можна виділити дві основні групи таких систем. До першої належать спеціальні пошукові сервери (роботи), які автоматично за заданим алгоритмом обходять Web-сервери і вибирають Web-сторінки, використовуючи знайдені на них гіперпосилання, після чого відбуваються індексування всіх знайдених документів та формування БД, в якій зберігаються відомості про слова й URL-адреси, вміщені в документах. Користувач, задаючи в запиті ключові слова, дістає в результаті підбір посилань на документи, зміст яких задовольняє критерій пошуку. Найпотужніші пошукові системи дають змогу формулювати досить складні запити з об'єднанням ключових слів у логічні вирази, створенням додаткових обмежень (наприклад, на дату формування документів, доменні імена серверів, серед яких здійснюється пошук, та ін.).

Найвідомішими пошуковими серверами є: www.euroseek.net, www.excite.com, www.infoseek.com, www.qp.dp.ua, www.meta.kharkiv.net.

Друга група пошукових систем - це каталоги Інтернет-ресурсів, до яких інформація вводиться фахівцями після її аналізу та класифікації. Такі системи містять тематично структуровані анотації на веб-серверах та окремих веб-сайтах або списки адрес серверів з їх іменами. Існуючі каталоги відрізняються між собою призначенням, структурою та обсягом наданої інформації, можливостями

підтримки та додатковими даними. Деякі довідники охоплюють якомога більше ресурсів Інтернету з різних предметних областей, інші обмежуються ресурсами країни або регіону або мають певну тематичну спрямованість. Інформацію в каталогах можуть вільно вводити всі користувачі мережі або спеціальні редакційні групи, які перевіряють отримані адреси та шукають корисні сторінки. Записи зазвичай містять назву веб-сайту, його URL-адресу, опис теми веб-сайту та примітку про доступну інформацію, а також додаткову інформацію для комерційного використання (країна, місто, адреса електронної пошти, відповідальна за веб-сайт) . людина). Відомі каталоги - www.rambler.ru, www.topping.com.ua, www.atlas-ua.net.

Щоб розпочати процес пошуку, потрібно активувати веб-браузер і ввести адресу сервера пошуку. Після завантаження програми введіть ключове слово та натисніть Enter. Автоматично підключена пошукова машина здійснює пошук на сервері бази даних і, якщо знаходить ключове слово на наступній веб-сторінці, вводить його адресу в список результатів пошуку. Тепер вам потрібно двічі клацнути на гіперпосиланні, щоб переглянути вміст знайдених файлів.

Ви також можете шукати у вікні браузера. Для цього виберіть відповідний сервер пошуку зі списку, активуючи кнопку пошуку в лівій частині вікна.

Створення та робота з веб-сайтами. Веб-сторінка - це текстовий файл, створений за допомогою команд HTML. Для роботи з такими файлами використовується спеціальна програма - браузер. Це може бути Internet Explorer, Netscape Communicator тощо.

Файл HTML має структурований вигляд, упорядкований за тегами (інструкції браузера для визначення тієї чи іншої частини файлу). Мітки парні - для позначення початку і кінця фрагмента, а елемент - для визначення дії. Створюйте такі файли в будь-якому текстовому редакторі; Під час запису на жорсткий диск виберіть Звичайний текст у полі Тип файлу, введіть ім'я файлу в поле Ім'я файлу та додайте розширення dot-ku - html.

Основні команди (теги), що використовуються при створенні веб-сторінок:

- 1) `<html> </html>` - початок і кінець документа;
- 2) `<head> </head>` - область заголовка документа;

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		43

- 3) `<body> </body>` - зміст документа;
- 4) `<title> </title>` - заголовок вікна браузера з відкритим файлом;
- 5) `<frameset rows = "200, "> ... </frameset>` - зарезервувати частину вікна для кадру;
- 6) `<frame name = center src = file.html>` - визначення імені (розташування у вікні та імені на жорсткому диску) кадру;
- 7) ` text ` - гіперпосилання, що визначає частину вікна (фрейму), що містить файл;
- 8) `<body bgcolor = "silver" text = "black" link = "# ff0000">` - визначення кольорів тла вікна, тексту, гіперпосилань; колір можна вказати за іменем (срібло) або шістнадцятковим кодом (# FF0000);
- 9) `<!--...-->` - створити коментар;
- 10) `<marquee bgcolor = "# 6defff"> text </marquee>` - Створити рухомий рядок;
- 11) `` - ви визначаєте тип шрифту, його розмір, колір;
- 12) `` - вставити графічний об'єкт із визначенням його ширини, висоти та з'явиться коментар, який буде відображатися на екрані - Миша Миша до об'єкта;
- 13) `<bgsound = "14.rmi" loop = "infinite">` - вставляє аудіофайл, де цикл визначає тривалість звуку;
- 14) ` ` - відступ абзацу;
- 15) `<p> </p>` - абзац;
- 16) `
` - перейти до нового рядка;
- 17) ` ` - форматування тексту жирним шрифтом;
- 18) `<i>>` - форматування тексту курсивом;
- 19) `<u> </u>` - форматування тексту підкресленим шрифтом;
- 20) ` ` - початок (кінець) списку переліку;
- 21) ` ` - початок (кінець) нумерованого списку;
- 22) `<li / li>` - елемент списку;
- 23) ` Текст ` - гіперпосилання на інший файл;

- 24) `<p align = center>` - визначає тип вирівнювання абзацу;
- 25) `<table> ... </table>` - початок (кінець) таблиці;
- 26) `<Кадр таблиці>` - створити фрейм таблиці;
- 27) `<td> </td>` - початок (кінець) комірки таблиці;
- 28) `<tr> </tr>` - початок (кінець) рядка таблиці;
- 29) `<th> </th>` - заголовок стовпця таблиці.

3.2 Ієрархія пам'яті та створення кластера

Лінійність - це найсильніша гарантія узгодженості, коли всі клієнти гарантовано бачать найновіші оновлені дані. Забезпечення лінеаризованості поряд з відмовостійкістю потребує консенсусних алгоритмів, таких як Raft, Zab або Paxos, що мають бути впроваджені на серверах.

Хоча консенсусний алгоритм є важливою вимогою для реалізації послідовного ядра, існують різні аспекти взаємодії з клієнтом, такі як як клієнт знаходить лідера, як обробляються дублікати запитів тощо - що є важливими рішеннями щодо впровадження. Є також кілька важливих міркувань щодо впровадження щодо безпеки та життєдіяльності.

ахос визначає лише алгоритм консенсусу, але ці інші аспекти реалізації недостатньо добре задокументовані в літературі Paxos. Raft дуже чітко документує різні аспекти реалізації, поряд із еталонною реалізацією, і тому є найбільш широко використовуваним алгоритмом сьогодні.

Коли кластер повинен обробляти багато даних, йому потрібно все більше серверів.

Для кластера серверів існують деякі загальні вимоги, такі як вибір конкретного сервера, який буде головним для конкретного завдання, управління інформацією про членство в групі, зіставлення розділів даних із серверами тощо. Ці вимоги потребують надійної гарантії узгодженості, а саме лінійності . Впровадження також повинно бути стійким до несправностей. Типовим підходом є використання відмовостійких алгоритмів консенсусу, заснованих на Кворумі.

Але в системах, що базуються на кворумах, пропускна здатність погіршується залежно від розміру кластера.

Рішення. Впровадьте менший кластер від 3 до 5 вузлів, який забезпечує гарантію лінійності та відмовостійкість. [1] Окремий кластер даних може використовувати невеликий послідовний кластер для управління метаданими та для прийняття рішень на рівні кластера з такими примітивами, як Оренда.

Таким чином, кластер даних може зростати до великої кількості серверів, але все одно може робити певні дії, які потребують надійних гарантій узгодженості, використовуючи менший кластер (рисунок 3.2).

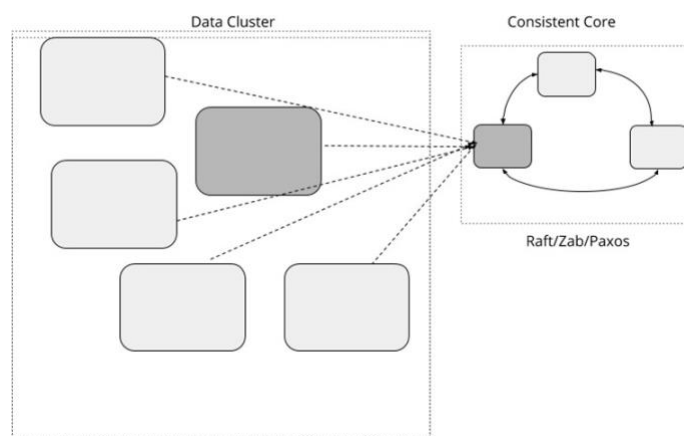


Рисунок 3.2 - Послідовне ядро

Типовий інтерфейс послідовного ядра виглядає так:

```
public interface ConsistentCore {  
    CompletableFuture put(String key, String value);  
  
    List<String> get(String keyPrefix);  
    CompletableFuture registerLease(String name, long ttl);  
    void refreshLease(String name);  
    void watch(String name, Consumer<WatchEvent> watchCallback);  
}
```

Як мінімум, Consistent Core забезпечує простий механізм зберігання значення ключа. Він використовується для зберігання метаданих.

Зберігання метаданих

Сховище реалізоване за допомогою консенсусних алгоритмів, таких як Raft. Це приклад реалізації журналу реплікаційного запису вперед, де реплікація обробляється Leader and Followers, а High-Water Mark використовується для відстеження успішної реплікації за допомогою Quorum

Підтримка ієрархічного сховища

Послідовне ядро, як правило, використовується для зберігання даних для таких речей, як: членство в групі або розподіл завдань між серверами. Типовий шаблон використання полягає у застосуванні типу метаданих із префіксом. напр. для членства в групі всі ключі зберігатимуться як / сервери / 1, сервери / 2 тощо. Для завдань, призначених серверам, ключами можуть бути / задачі / завдання1, / завдання / завдання2. Ці дані зазвичай зчитуються з усіма клавішами з певним префіксом. Наприклад, щоб отримати інформацію про всі сервери в кластері, зчитуються всі ключі з префіксом / серверами.

Прикладом є наступне:

Сервери можуть зареєструватися в системі Consistent Core створивши власний ключ із префіксом / серверами.

```
client1.setValue("/servers/1", "{address:192.168.199.10, port:8000}");
client2.setValue("/servers/2", "{address:192.168.199.11, port:8000}");
client3.setValue("/servers/3", "{address:192.168.199.12, port:8000}").
```

Потім клієнти можуть дізнатися про всі сервери в кластері, прочитавши з префіксом ключа / сервери наступне:

```
assertEquals(client1.getValue("/servers"),
Arrays.asList("{address:192.168.199.12, port:8000}",
"{address:192.168.199.11, port:8000}",
"{address:192.168.199.10, port:8000}"));
```

Через таку ієрархічну природу зберігання даних, такі продукти, як [zookeeper], [chubby], надають файлову систему на зразок інтерфейсу, де користувачі створюють каталоги та файли, або вузли, з концепцією батьківського та дочірнього вузлів. [etcd3] має плоский простір ключів із можливістю отримати набір ключів.

Обробка взаємодій з клієнтом. Однією з ключових вимог до функціональності послідовного ядра є взаємодія клієнта з ядром. Наступні аспекти є критично важливими для роботи клієнтів із послідовним ядром:

- 1) пошук лідера;
- 2) серійність і лінійність.

Коли запити на читання обробляються серверами-послідовниками, можливо, що клієнти можуть отримати застарілі дані, оскільки останні комітети від лідера не дійшли до послідовників.

Порядок отримання оновлень клієнтом зберігається, але оновлення можуть бути не останніми. Це гарантія [серіалізації] на відміну від лінійності. Лінійність дозволяє гарантувати, що кожен клієнт отримує найновіші оновлення. Клієнти можуть працювати з гарантією серіалізації, коли їм просто потрібно прочитати метадані та можуть терпіти застарілі метадані деякий час.

Для таких операцій, як Оренда, строго потрібна лінійність.

Якщо лідер розділений з решти кластера, клієнти можуть отримувати застарілі значення від лідера, Рафт описує механізм забезпечення лінійного читання. Дивіться, наприклад, реалізацію `etcd readIndex`.

Подібна ситуація може статися з послідовниками, які розділені. Послідовник може бути розділений і може не повертати клієнту останні значення. Щоб переконатися, що послідовники не розділені та в курсі лідера, їм потрібно запитати лідера, щоб знати останні оновлення, і дочекатися, поки вони отримають останні оновлення, перш ніж відповісти клієнтові. Перегляньте запропонований `kafka` наприклад, дизайн.

Важливо, щоб усі операції виконувались над лідером, тому клієнтській бібліотеці спочатку потрібно знайти сервер лідера. Існує два підходи для виконання цієї вимоги:

1) servers Сервери послідовників у послідовному ядрі знають про поточного лідера, тому, якщо клієнт підключається до послідовника, він може повернути адресу лідера. Потім клієнт може безпосередньо підключитися до лідера, визначеного у відповіді. Слід зазначити, що сервери можуть перебувати в середині виборів лідера, коли клієнт намагається підключитися. У цьому випадку сервери не можуть повернути адресу лідера, і клієнту потрібно почекати та спробувати інший сервер;

2) сервери можуть реалізувати механізм переадресації та пересилати всі запити клієнта лідеру. Це дозволяє клієнтам підключатися до будь-якого сервера. Знову ж таки, якщо сервери знаходяться в середині виборів лідера, тоді клієнтам потрібно повторити спробу, поки вибори лідера не пройдуть успішно і не буде встановлено законного лідера.

Такі продукти, як zookeeper та інше, реалізують цей підхід, оскільки вони дозволяють лише для читання запити, які повинні обробляти сервери послідовників; це дозволяє уникнути вузького місця у лідера, коли велика кількість клієнтів доступна лише для читання.

Це зменшує складність підключення клієнтів до керівника або послідовника залежно від типу запиту.

Простим механізмом пошуку лідера є спроба підключитися до кожного сервера та спробувати надіслати запит, сервер відповідає переадресацією, якщо це не лідер.

```
private void establishConnectionToLeader(List<InetAddressAndPort> servers) {
    for (InetAddressAndPort server : servers) {
        try {
            SingleSocketChannel socketChannel = new SingleSocketChannel(server,
10);

            logger.info("Trying to connect to " + server);
            RequestOrResponse response = sendConnectRequest(socketChannel);
            if (isRedirectResponse(response)) {
                redirectToLeader(response);
            }
        }
    }
}
```

Зм.	Арк.	№докум.	Підпис	Дата

```

        break;
    } else if (isLookingForLeader(response)) {
        logger.info("Server is looking for leader. Trying next server");
        continue;
    } else { //we know the leader
        logger.info("Found leader. Establishing a new connection.");
        newPipelinedConnection(server);
        break;
    }
} catch (IOException e) {
    logger.info("Unable to connect to " + server);
    //try next server
}
}

private boolean isLookingForLeader(RequestOrResponse requestOrResponse) {
    return requestOrResponse.getRequestId() ==
RequestId.LookingForLeader.getId();
}

private void redirectToLeader(RequestOrResponse response) {
    RedirectToLeaderResponse redirectResponse = deserialize(response);
    newPipelinedConnection(redirectResponse.leaderAddress);

    logger.info("Connected to the new leader "
        + redirectResponse.leaderServerId
        + " " + redirectResponse.leaderAddress
        + ". Checking connection");
}

private boolean isRedirectResponse(RequestOrResponse requestOrResponse) {

```

Зм.	Арк.	№докум.	Підпис	Дата

```

        return requestOrResponse.getRequestId() ==
RequestId.RedirectToLeader.getId();
    }

```

Просто встановлення TCP-з'єднання недостатньо, нам потрібно знати, чи може сервер обробляти наші запити. Отже, клієнти надсилають спеціальний запит на підключення до сервера, щоб підтвердити, чи може він обслуговувати ці запити або перенаправляти на ведучий сервер.

```

private RequestOrResponse sendConnectRequest(SingleSocketChannel
socketChannel) throws IOException {
    RequestOrResponse request
        = new RequestOrResponse(RequestId.ConnectRequest.getId(),
"CONNECT", 0);
    try {
        return socketChannel.blockingSend(request);
    } catch (IOException e) {
        resetConnectionToLeader();
        throw e;
    }
}

```

Якщо існуючий лідер не вдається, використовується той самий спосіб, щоб ідентифікувати новообраного лідера з кластера.

Після підключення клієнт підтримує канал Single Socket до сервера лідера

Обробка повторюваних запитів $\{\{ 1 \}\}$ У випадках невдачі клієнти можуть спробувати підключитися до нового керівника, повторно надіславши запити. Але якщо ці запити вже оброблялися невдалим керівником до відмови, це може призвести до дублікатів. Тому важливо мати на серверах механізм ігнорування дублікатів запитів. Шаблон Idempotent Receiver використовується для реалізації виявлення дублікатів.

Координацію завдань на наборі серверів можна здійснити за допомогою оренди. Те саме може бути використано для реалізації механізму членства в групі та виявлення помилок.

Державний годинник використовується для отримання сповіщень про зміни в метаданих або оренді в часі.

Приклади

Google відомий використовувати службу блокування [пухкі] для координації та управління метаданими.

[kafka] використовує [zookeeper] для управління метаданими та прийняття рішень, таких як вибори лідера для майстра кластера. Запропонована зміна архітектури в Kafka замінить zookeeper на власний кластер контролерів на основі [raft].

[bookkeeper] використовує Zookeeper для управління метаданими кластера.

[kubernetes] використовує [etcd] для координації, управління метаданими кластера та інформація про членство в групах.

Усі системи зберігання та обробки великих даних, такі як [hdfs], [spark], [flink], використовують [zookeeper] для високої доступності та координації кластерів. Реалізація кластера - це дуже простий спосіб. Ви не можете повідомити, що це вірогідність Linux-дистрибуції Ihrer Wahl nähern. Welche der Linux-Distributionen Sie als Basisbetriebssystem verwenden - egal, Ubuntu, Mandriva, Alt Linux, Red Hat, SuSE. Вибір висить лише спереду ab. Критичне представлення кластерів із загальним розподілом:

1) встановіть операційну систему на комп'ютер, який працює як консоль кластера. Паралельна, паралельна програма може бути скомпільована та скомпільована на цьому комп'ютері. Іншими словами, людину, яка сиділа за цим комп'ютером, запускала програму і дивилася, пропустили;

2) після встановлення операційної системи на консолі кластера неможливо при першій установці, необхідно встановити компілятор (Fortran, C) та всі необхідні бібліотеки, доступ до робочого столу (установка GNOME або KDE). Texteditoren usw. Комп'ютер, комп'ютер у розробнику працюють;

3) встановіть пакет MPICH або OpenMPI зі сховища або вихідного коду;

- 4) напишіть у / etc / підключені кластери Knoten Ihres, один із кластерів консолі;
- 5) встановлено NFS, і для всіх кластерних ручок створюється обліковий запис, в якому виконуваний модуль повністю паралельно виконує програму та;
- 6) файли даних, ці програми доступні для своєї роботи;
- 7) встановіть клієнт ssh (необов'язково) та сервер ssh на консолі кластера, якщо ми отримуємо доступ до консолі кластера через мережу. Встановіть усі кластери кластерів операційної системи та бібліотек, які для визначення користувацької паралельної програми є необов'язковим;
- 8) встановити MPICH, NFS-клієнт, SSH-сервер. Ресурс для збереження, читання Cluster Knoten у Runlevel 3 доступний, але ви можете використовувати GNOME у KDE. Як максимум, Darn, Gnome або локальні бібліотеки можна видалити, але це недоступно для користувача програми. опишіть у / etc / hostet alle Knoten des Clasters zukünftige Knoten usseres Clasters, einschließlich der Cluster-Konsole;
- 9) на всіх вузлах кластера невідомо, у пункті 5 є незалежний ресурс для автоматичного завантаження. Через те, що Pfad до цього ресурсу можна знайти на консолі кластера, а також на їх ідентичності Knoten. Якщо ми приймаємо дані / home / mpiuser / data на консолі кластера, ми повинні повернути цей ресурс кнопкам кластера під / home / mpiuser / data;
- 10) на всіх регуляторах кластера надається прохідний доступ до консолі кластера.

Кластер зібраний і повністю готовий до використання. Щоб забезпечити кластер, нам довелося встановити операційну систему, перейти до так званої «Установки та видалення програм», позначити інсталяційні пакети SSH та MPICH, заборонити запит на віддалений доступ до вузлів кластера, звільнити каталог вузлів у штаб-квартира, де зберігаються наші паралельні програми та дані та встановлюється автоматичне підключення до цього каталогу на вузлах кластера під час завантаження. Ви можете побачити, як компілювати та запускати паралельні програми, в інших розділах цього сайту та в документації MPICH. Це дуже просто, і все, що вам потрібно, - це сховище із сховищами.

Тепер я хочу обговорити ще одну тему, яка полягає у побудові мережі кластерів. Оскільки мережа є найвужчою точкою, і ефективність кластера безпосередньо залежить від неї, я хотів би зробити наступне. Було б бажано, щоб робота мережевої файлової системи NFS не перешкоджала обміну даними, що здійснюється частинами паралельної програми, що працює на різних вузлах. Для цього вам потрібно фізично розділити мережу на два сегменти. NFS працюватиме як в одному сегменті, так і в іншому - відбувається обмін даними між частинами програми. Отже, і консоль кластера, і її вузли повинні мати два мережеві інтерфейси (дві мережеві карти). Відповідно, вам потрібні два набори комутаторів, які не взаємопов'язані, і два набори деталей мережі для цих інтерфейсів. Це означає, що NFS працює, наприклад, у мережі 192.168.1.0/24, а обмін даними відбувається в мережі 192.168.2.0/24. Відповідно, адреси з першої мережі, а також у файлах / тощо `hosts` і у файлових машинах.LINUX опис кластера - адреси з другого. Файли, що зберігаються на диску в контексті паралельного завдання, що виконується на кластері, можуть знадобитися лише для збереження стану завдання на контрольних пунктах. Звичайно, дискові ресурси можна використовувати для організації віртуальної пам'яті, завантаження даних в оперативну пам'ять за потреби, і тим самим збільшити розмір сітки. Однак, якщо є кластер і розмір сітки збільшується, логічніше та ефективніше використовувати додаткові обчислювальні вузли на кластері. Якщо дискові ресурси використовуються лише для зберігання контрольних точок, і ці контрольні точки містяться не на кожній ітерації (а на кожній десятій чи сотій), то розподіл кластерної локальної мережі на два незалежних сегменти (NFS та міжпроцесорна мережа) не є необхідним. Можна працювати лише з одним сегментом, використовувати його для NFS та для обміну даними. Оскільки NFS використовується рідко, його негативний вплив на ефективність кластеру мінімальний.

Рекомендується використовувати гігабітну мережу для університету на найкращому (фінансовому) рівні. Генауер отримав підтримку Gigabit Ethernet NICHT, що працює за межами Wahl або Cluster-Netzwerk, а також Netzwerk eine

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		54

ziemlich hohe Latenz. Але це серйозне рішення. Якщо фінанси будуть доступні, ви зможете використовувати Myrinet або 10 Гбіт Ethernet.

Netzwerkeinstellungen sind nie sehr gut. Daher sollten wir, wenn möglich, versuchen, sie zu verbessern. Коли Myrinet або 10GbE недоступні, ви можете спробувати скористатися мережею Gigabit. насправді, це зробити дві гігабіти. Крім того, у нас є мережа з двох мережевих карт, і у нас є спеціальний канал для віртуального каналу в поєднанні з подвійною пропускнуою здатністю. Цієї осені можна підключити карти з двома окремими ядрами, а також знайти рідини на цих картах, навчити NFS та передачу даних. Відкритий канал для створення тривалого відкриття. В принципі, цей принцип не відомий, але про ефект можна дискутувати. Не дуже, у нас кластер капітально відремонтований. Справа в тому, що виставка "Je mehr Knoten im Cluster, desto schneller funktioniert es" - im allgemeinen Fall nicht zutrifft. Подивіться лише, восени ми розберемо наші програми в кластері. Для використання кластера існує лише два Цвеке:

1) існує сітка розміру R , обчислення для використання звичайного комп'ютера, що час T . Час T є критичним параметром. Ми хочемо зменшити кількість разів, тому маємо R як константу;

2) це растр великого R , то обчислення в момент нормального використання комп'ютера T часу.

3) час T не критичний. Ми зацікавлені, і Растр збирається збільшити розмір. Усі обчислення в растрі мають для нас загальний і важливий параметр: час - це ітерація. На момент кластеру цей час базується на двох частинах: кількості разів на сітці титрів і часу текстового вузла, встановленого датою. Заголовок висить лише від потужності процесора ab . Aber $Texch$ залегає від розміру сіток, кількості вузлів у кластері та пропускнуої здатності спільноти. Загалом, є також усі категорії: Комп'ютерне самообслуговування, мережеві продажні матеріали, фізичне зберігання комп'ютерів, обладнання для розподілу енергії, консолі для доступу до комп'ютера та запасні частини.

Комп'ютер - Кластер можна створити із звичайних комп'ютерів. На додаток до великого кластера, комп'ютером можна користуватися, розробляються спеціальності для бухгалтерії високого класу, а компанія supermicro продається.

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		55

Diese Systeme sind besser als einfache PC, da sie in Gehäusen mit hoher Dichte geliefert werden, z. B. 1 або 2 ВІН, в одній із стійок встановлені (в інтерпретації вертикальна стійка-платформа 1,75 дюйма). На повністю опущеному комп'ютері з Linux це працює добре. Для масштабних ваг обладнані ручки управління PXE-Boot та IPMI. Плата повинна бути обладнана RAID-масивами і повинні бути встановлені всі порти Ethernet, один для підключення до зовнішньої мережі, а інший для ізольованої внутрішньої мережі. Залежно від застосування кластерів можна поєднати потужну комбінацію енергетичних процесів, великі метри оперативної пам'яті, великі материнські плати або графічні процесори. . Додаткові порти завжди доступні, і кластер Іхрем в майбутньому можна знайти в районі Кнотен. Мережевий кабель підключений безпосередньо.

Фізична конфігурація - кластер можна створити з ПК на полиці. Професійно побудований кластер продає лише звичайний спеціальний комп'ютер. Комп'ютер можна зняти з рейок, але його також можна опустити, можна зняти і фізично зняти зі стійки. На задній панелі калькулятора рекомендується звільнити вільний простір для кабелю, який буде використовуватися для діагностики пристрою для діагностики. Скупчення може дуже вам подякувати за повітря, але вам не доведеться їх залишати, але білий раушен не зупиняється. Скупчення також може виробляти багато тепла. Коли контингент великий, у вас є спеціальна пропозиція або спеціальний пристрій, система кондиціонування.

Розподіл енергії - комп'ютер споживає багато енергії, а комп'ютер - багато енергії. Поточний ланцюг, над яким ви можете працювати, ви повинні мати можливість працювати, і ви можете знайти список штекерів, і струм можна розділити. Якщо ваш кластер невеликий, ви зможете використовувати джерело живлення для споживачів. Сонячне світло - це великий стек для будівництва граблів, поточний запас. Завдяки тому, що основний блок і блок запам'ятовування підключені до джерела вільного струму, можна уникнути живлення та захисту даних.

Der Zugriff ist impraktisch, wenn für jeden Clusterknoten eine Tastatur, ein Videomonitor und eine Maus vorhanden sind. Це гарна ідея - монітор, миша та клавіатура на основному блоці. Це можна зробити безпечно, але його можна

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		56

побудувати на вашому кластері та керувати ним адміністративно. Можна використовувати спеціальні продукти, такі як монтований ПК-монітор та клавіатура.

Після конфігурації кластерів ви можете використовувати SSH головних регуляторів на клавішах облікового запису. У звичайному режимі роботи кнопки можна складати (без роботи клавіатури, монітора та миші). Восени ненормальної роботи, як першої установки кластера або при діагностиці проблеми, ви можете знайти кошик для сміття на білбордах, мобільний пристрій, мобільний монітор та клавіатуру з довгим кабелем. Доступний інший варіант, але стандартні перемикачі кабелю E-A / A (наприклад, USB і VGA) можна встановити або використовувати через IP, але можна встановити BIOS на комп'ютерах. Ви можете використовувати більше кнопок, ви можете використовувати пристрої, монітор, клавіатуру, мишу та кабель.

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		57

ВИСНОВКИ

В ході роботи було проаналізовано існуючі методи аналізу віртуальної пам'яті, комерційні програми моніторингу, аналізу, та використання віртуального адресного простору.

Під час виконання кваліфікаційної роботи було здійснено теоретичний аналіз, вивчено недоліки існуючих рішень та сформовано задачу для дослідження шляхів їх вирішення.

Віртуальна пам'ять ефективно працює в системах з багатозадачною роботою, в яких пам'ять містить частини багатьох програм одночасно, в очікуванні читання своєї частини, процесор може бути зайнятий іншим процесом.

На першому етапі роботи відбулось ознайомлення з засобами аналізу віртуальної пам'яті. При цьому було охарактеризовано структуру предметної області та базову модель організації операційного пристрою.

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		58

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Пахмурин Д. О. Операционные системы ЭВМ. Учебное пособие. Томск: Томский государственный университет систем управления и радиоэлектроники, 2013.
2. Карачка А. Ф., Дудко О. І. Архітектура комп'ютерів: Навч. посіб. За ред. А. О. Саченка. Тернопіль: Економічна думка, 2010. 176 с.
3. Х. М. Дейтел, П. Дж. Дейтел, Д. Р. Чофнес. Операционные системы. Часть 1. Основы и принципы. М. Бином-Пресс. 2011. 1024с.
4. Операционные системы. Часть 2. Распределенные системы, сети, безопасность Х. М. Дейтел, П. Дж. Дейтел, Д. Р. Чофнес, М. Бином Пресс. 2011. 704 с.
5. Орлов С. А., Цилькер Б. Я., Организация ЭВМ и систем. СПб.: Питер, 2004. 667 с.
6. Клименко М. І., Панасенко Є. В., Стреляєв Ю. М., Ткаченко І. Г. Варіаційне числення та методи оптимізації : навч. посіб. Запоріжжя : ЗНУ, 2015. 84 с.
7. ДСТУ 3582:2013. Бібліографічний опис. Скорочення слів і словосполучень українською мовою. Загальні вимоги та правила(ISO 4:1984, NEQ; ISO 832:1994, NEQ). Вид. офіц. Київ : Мінекономрозвитку України, 2014. 15 с. (Інформація та документація).
8. Reiss, C., Tumanov, A., Ganger, G.R., Katz, R.H., and Kozuch, M.A. Heterogeneity and dynamicity of clouds at scale: Google trace analysis. Proceedings of the 3rd ACM Symposium on Cloud Computing, ACM (2012).
9. Про затвердження Вимог до оформлення дисертації: наказ Міністерства освіти і науки від 12.01.2017 р. № 40. Офіційний вісник України. 2017. № 20. С. 136–141.
10. Коноваленко І. В., Федорів П. С. Системне програмування у Windows з прикладами на Delphi, 2012.

11. Чистяков В. Д. Анатомия ПК. Все о компьютерном железе. – М.: ИТ Пресс, 2011. 160 с.
12. Ahn, J., Jin, S., and Huh, J. Revisiting Hardware-Assisted Page Walks for Virtualized Systems. Proceedings of the 39th Annual International Symposium on Computer Architecture, 2012.
13. Операционные системы, С. В. Сеницын, А. В. Батаев, Н. Ю. Налютин, 2010. 304с.
14. Современные операционные системы, Э. Таненбаум, СПб, Питер, 2010. 1120 с.
15. Иртегов Д. В. Введение в операционные системы. Д. В. Иртегов. 2-е изд. БХВ-Петербург, 2012. 1040 с.
16. Kusswurm Daniel. Modern X86 Assembly Language Programming. Daniel Kusswurm. - Apress, 2019. 604 p.
17. William Stallings. Operating Systems: Internals and Design Principles., 2018.
18. Stallings, William. Operating systems: internals and design principles William Stallings. 7 th ed. Prentice Hall, New Jersey, 2012, p.769.
19. Куклин В. М. ред. Введение в методы программных решений: учебное пособие: ХНУ им. В.Н. Каразина, 2011.
20. Текстові документи. Загальні вимоги. Красильникова, Л. Першина, Т. Касянчук., Хмельницький : ХНУ, 2017. 45 с. Текстові документи. Загальні вимоги. Красильникова, Л. Першина, Т. Касянчук., Хмельницький : ХНУ, 2017. 45 с.
21. Микитишин А.Г., Чихіра І.В. Операційні системи: конспект лекцій Тернопіль: ТНТУ, 2016. 104 с.
22. Tulloch M. Optimizing and Troubleshooting Hyper-V Networking Microsoft Press, 2013.
23. Л. Аврамов. Центры обработки данных на основе политик и АСІ: структура, концепции и методология, 2016. 384 с.
24. Christos Kozyrakis, A.K. and Vaid, K. Server Engineering Insights for Large-Scale Online Services. IEEE Micro, 2010.

					КВРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		60

25. Авраменко В. С., Салапатов В. І. Вступ до програмної інженерії. Том 1. Історія розвитку. Основні поняття. Навчальний посібник. Черкаси: Черкаський національний університет ім. Б Хмельницького, 2015. 500 с.
26. Lustig, D., Bhattacharje, A., and Martonosi, M. TLB Improvements for Chip Multiprocessors: Inter-Core Cooperative Prefetchers and Shared Last-Level TLBs. ACM Transactions on Architecture and Code Optimization, 2013.
27. Назаров С.В., Широков А.И. Современные операционные системы: учебное пособие. М.: Национальный Открытый Университет «ИНТУИТ», 2012. 367 с.
28. Srikantaiah, S. and Kandemir, M. Synergistic TLBs for High Performance Address Translation in Chip Multiprocessors. Proceedings of 43rd Annual IEEE/ACM International Symposium on Microarchitecture, (2010).
29. Вычислительные системы, сети и телекоммуникации: учебное пособие, А.П. Пятибратов, Л.П. Гудыно, А.А. Кириченко., 2013. 376 с.
30. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
31. Volos, H., Tack, A.J., and Swift, M.M. Mnemosyne: Lightweight Persistent Memory. Proceedings of the 16th International Conference on Architectural Support for Programming Languages and Operating Systems, (2011).
32. Логічні елементи то основні операційні вузли: Лабораторний практикум. В. В. Булатецький, Л. В. Булатецька. Луцьк. Вежа, 2007. 68 с.
33. Binh Pham, Viswanathan Vaidyanathan, Aamer Jaleel, and Abhishek Bhattacharjee. Colt: Coalesced large-reach TLBs. In MICRO, pages 258–269. IEEE Computer Society, 2012.
34. Устройство и функционирование ОС Windows. Национальный Открытый Университет «ИНТУИТ», 2016. 240 с.
35. Востокин С. В. Операционные системы: Учебник. Самара: Изд-во Самар, гос. аэрокосм, ун-та, 2012. 120 с.
36. Фридланд А. Я., Ханамирова Л. С., Фридланд И. А. Информатика и компьютерные технологии: Основные термины: Толковый словарь. 3-е изд., испр. и доп. М.: ООО «Издательство Астрель»: ООО «Издательство АСТ», 2013.

					КвРКІ. 170144.17.01.15 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		61

37. Рамбо Джеймс, Якобсон Айвар, Буч Грэди. Специальный справочник администратора. Питер, 2012. 656с.
38. Фролов А.В., Фролов Г.В. Аппаратное обеспечение "ДИАЛОГ-МИФИ", 2012. 208 с.
39. Lustig, D., Bhattacharje, A., and Martonosi, M. TLB Improvements for Chip Multiprocessors: Inter-Core Cooperative Prefetchers and Shared Last-Level TLBs. ACM Transactions on Architecture and Code Optimization, 2013.
40. Рассел С., Норвиг П. Искусственный интеллект: современный подход. 2-е изд. М. Вильямс, 2006. 1408 с.

Ім'я користувача:
Кафедра КІ

Дата перевірки:
29.06.2021 15:41:58 EEST

Дата звіту:
29.06.2021 15:43:01 EEST

ID перевірки:
1008364413

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005591

Назва документа: Мартинець_Мережа магазину з розмежуванням доступу на основі обладнання Mikrotik
Кількість сторінок: 66 Кількість слів: 6053 Кількість символів: 45096 Розмір файлу: 6.65 MB ID файлу: 1008434322

11% Схожість

Найбільша схожість: 2.15% з джерелом з Бібліотеки (ID файлу: 1008278827)

7.5% Джерела з Інтернету 42

Сторінка 68

4.69% Джерела з Бібліотеки 63

Сторінка 69

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи 18

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 2.0%

Словари проверки: en_US, ru_RU, ua_UA. Ошибок в документах: 17%

ID: 95532 Название: Мережа магазину з розмежуванням доступу на основі обладнання Mikrotik Добавлено в БД: 2021-06-29 Авторы: М.А. Мартиненко Руководители: Кляш Ю.П. Консультанты: Опоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	29988	337	1500 (5%)	19 (6%)

Источники плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

РЕЦЕНЗІЯ НА ДИПЛОМНИЙ ПРОЕКТ

Дипломник Молодницький Андрій Ігорович

Тема Мультипроцесорна система загального призначення на основі топології «тор»

Спеціальність 123 Комп'ютерна інженерія

Обсяг дипломного проекту:

кількість листів креслень 0; кількість сторінок записки 62

1. Висновок про відповідність ДП дипломному завданню Дипломний проект у повній мірі відповідає поставленому завданню як в теоретичній, так і в практичній частині даного бпроекту

2. Характеристика виконання кожного розділу проекту, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому, теоретичному, розділі дипломного проекту якісно та в повній мірі розглянуті методи вирішення поставленої задачі, був проаналізований кожен аспект, який стосується теми дипломного проекту. У наступному розділі було здійснено обґрунтування обраної структури мережі на основі порівняння різних можливих варіантів побудови цієї мережі. У основній проектній частині диплому була реалізована сучасними методами та рішеннями логічна структуризація мережі. Резервування каналів Internet здійснено на основі новітніх можливостей операційних систем серверного типу. Також було здійснено налаштування сервера електронної пошти у відповідному розділі дипломного проекту. Розділи «Охорона праці», «Економічна частина» та «Надійність комп'ютерної мережі» в повній мірі відповідають суті дипломного проекту. В загальному усі розділи відповідають завданню та містять сучасні методи вирішення поставлених завдань.

3. Позитивні сторони проекту Дипломний проект відповідає сучасним вимогам до проектування локальних комп'ютерних мереж та містить ряд інноваційних рішень, зокрема, з точки зору використання бездротових технологій на основі різних стандартів. Окремо можна виділити розглянуту можливість перетворення сигналів передачі даних за допомогою мобільних технологій (стандарти покоління 3G) у WLAN мережі, оскільки на сьогоднішній день таке рішення є досить затребуваним і має значні перспективи. Також важливим позитивним аспектом дипломного проекту є реалізована можливість резервування каналів Internet, що надає можливість гарантувати постійний доступ користувачів мережі до певних серверів, які мають критичне значення.

4. Негативні сторони проекту Надмірна кількість теоретичного матеріалу, відсутність початкових налаштувань комутаторів. В рамках дипломного проекту варто було приділити більшу увагу високошвидкісним способам передачі даних, оскільки потреба в даному рішенні щороку зростає.

5. Оцінка графічного оформлення та пояснювальної записки проекту Графічне оформлення виконане відповідно до суті дипломного проекту. У перших трьох (основних) листах креслення відображені основні фізичні та логічні зв'язки мережі, використання технології логічної структуризації мережі. У додаткових листах креслення розглянуті питання резервування каналів та порівняння різних варіантів побудови локальних мереж. Один із листів креслення присвячений бездротовим технологіям. В загальному графічне оформлення виконане на належному рівні, єдиним недоліком є нерівномірність розмірів шрифтів, що робить розгляд графічних креслень дискомфортним. Пояснювальна записка відповідає задекларованим нормам для її оформлення.

6. Відгук про проект в цілому В загальному дипломний проект заслуговує схвальних відгуків. Весь матеріал дипломного проекту структурований, чіткий та послідовний. Усі розділи проекту йдуть у вірній послідовності, що дозволяє чітко розуміти викладений матеріал в рамках даного дипломного проекту. Графічний матеріал дозволяє наочно побачити доцільність та ефективність рішень, які були прийняті за основу при проектуванні локальної мережі підприємства.

7. Інші зауваження _____

8. Оцінка дипломного проекту Розглянувши позитивні та негативні сторони представленого дипломного проекту, можна зробити висновок, що він заслуговує оцінку «задовільно». Турчан Кемальович доцент кафедри ІТЗ

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

« 24 » червня 2021 р.

К. Турчан (підпис)

Завідувачу кафедри КІСП
д-ру техн.наук, проф. Говорущенко Т. О.

Молодніцького Андрія Ігоровича

ІІБ здобувача вищої освіти

ФПКТС, 4 курсу, групи КІ-17-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність плагіату ознайомлений(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.

24.06.2021

дата


підпис

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА СИСТЕМНОГО ПРОГРАМУВАННЯ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

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

Назва: Синтез та моделювання операційного автомату на основі автомату Мура

Автор: Молодніцький Андрій Ігорович

Спеціальність: 123 – Компютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Березький Олег Миколайович, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

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

Підтвердження:

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

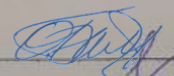
- 1) запозичення розмішені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
- 4) серед запозичень знаходяться загальновідомі терміни, скорочення та визначення.


Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 11% і адресується до 353 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

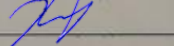
Керівник роботи

Гарант ОП

Завідувач кафедри КІСП







О.М. Березький

С. М. Лисенко

Т. О. Говорущенко