ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно-інтегрованих технологій, автоматизації, електроінженерії та радіоелектроніки
(повне найменування інституту, назва факультету)

Автоматика та телекомунікації
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

_____

_____
(підпис) (ініціали, прізвище)

"___"_____20_ р.

Випускна кваліфікаційна робота

магістра
(освітньо-кваліфікаційний рівень)

на тему «Дослідження механізмів балансування навантаження в центрах обробки даних»

Виконав : студент 2 курсу, групи ТКРм-16
(шифр групи)

спеціальності 172
(шифр і назва напряму підготовки, спеціальності)

«Телекомунікації та радіотехніка»

Вдовенко Олена Андріївна _____
(прізвище та ініціали) (підпис)

Керівник к.т.н., доц., проф. каф. АТ Воропаєва В.Я. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Покровськ – 2018 р.

ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно- інтегрованих технологій, автоматизації, електроінженерії та радіоелектроніки
(повне найменування інституту, назва факультету)

Автоматика та телекомунікації
(повна назва кафедри)

Допустити до захисту:
Декан фКІТАЕР
_____В.П. Тарасюк
(підпис та дата)

Захист відбувся_____
(дата)
з оцінкою _____
секретар ДЕК_____
(підпис)

Випускна кваліфікаційна робота

магістра
(освітньо-кваліфікаційний рівень)

на тему «Дослідження механізмів балансування навантаження в центрах обробки даних»

Виконав студент групи ТКРм–16 _____ О.А. Вдовенко
(підпис, дата) (ініціали, прізвище)

Керівник _____ к.т.н., доц. В.Я.Воропаєва
(підпис, дата) (ініціали, прізвище)

Зав. каф. автоматики та телекомунікацій _____ к.т.н., доц. В.В.Поцепаєв
(підпис, дата) (ініціали, прізвище)

Консультанти _____.
(підпис, дата) (ініціали, прізвище)

_____
(підпис, дата) (ініціали, прізвище)

_____
(підпис, дата) (ініціали, прізвище)

Нормоконтролер _____ асистент Д.О.Жуковська
(підпис, дата) (ініціали, прізвище)

Покровськ – 2018 р.

ДВНЗ «Донецький національний технічний університет»

Факультет комп'ютерно- інтегрованих технологій, автоматизації електроінженерії та радіоелектроніки
Кафедра автоматика та телекомунікації
Освітній ступінь магістр
Спеціальність 172 «Телекомунікації та радіотехніка
(шифр і назва)

ЗАТВЕРДЖУЮ:
Завідувач кафедри
_____ /_____/
"____" _____20___року

З А В Д А Н Н Я
НА ВИПУСКНУ КВАЛІФІАЦІЙНУ РОБОТУ СТУДЕНТУ

Вдовенко Олені Андріївні
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження механізмів балансування навантаження в центрах обробки даних»

керівник роботи: Воропаєва Вікторія Яківна, к.т.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом від " 12 " жовтня 2017 року № 439-08
2. Строк подання студентом роботи _____
3. Вихідні дані до роботи: результати науково-дослідної роботи, магістерської практики
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
1) аналіз предметної області та існуючі рішення;
2) поняття балансування навантаження та її особливості;
3) створення алгоритму керування вхідним потоком та моніторінгу;
4) створення динамічного алгоритму балансування навантаження;
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
- блок-схема алгоритму управління вхідним потоком;
- блок-схема алгоритму моніторингу;
- блок-схема динамічного алгоритму балансування навантаження.

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання_____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
	Аналіз предметної області	01.09-31.10.2016	
	Постановка задачі	01.11-30.11.2016	
	Вивчення теоретичного матеріалу	01.12.2016-31.03.2017	
	Розробка алгоритму управління вхідним потоком	01.04-31.08.2017	
	Розробка алгоритму моніторінгу	01.09.2017-30.11.2017	
	Збір даних	01.12.2017-10.01.2018	
	Розробка динамічного алгоритму балансування навантаження	11.01-31.01.2018	
	Оцінка та опис отриманих результатів	01.02-28.02.2018	
	Оформлення пояснювальної записки до роботи, графічного матеріалу, доповіді	01.03-10.05.2018	
	Нормоконтроль пояснювальної записки та матеріалів дипломування	10.05-01.06.2018	
	Рецензування роботи	01.06-13.06.2018	
	Захист роботи	14.06.2018	

Студент _____ _____
(підпис) (прізвище та ініціали)

Керівник проекту (роботи) _____ _____
(підпис) (прізвище та ініціали)

Лист зауважень

Посада П.І.Б.	Суть зауваження, оцінка та підпис

АНОТАЦІЯ

Вдовенко О.А. Дослідження механізмів балансування навантаження в центрах обробки даних / Випускна кваліфікаційна робота на здобуття освітнього ступеня «магістр» за спеціальністю 172 «Телекомунікації та радіотехніка» – ДВНЗ ДонНТУ, Покровськ, 2018.

Робота містить 92 сторінок, складається з вступу, чотирьох розділів, висновків, переліка використаних джерел з 40 найменувань, 16 рисунків, 10 таблиці.

Відповідно до поставленого завдання в роботі була вирішена задача розробки динамічного алгоритму балансування навантаження в мережах центрів обробки даних; наведена модель динамічного алгоритму балансування навантаження та розроблено алгоритми керування вхідним потоком, адаптивний алгоритм моніторингу.

Ключові слова: БАЛАНСУВАННЯ НАВАНТАЖЕННЯ, ЦЕНТР ОБРОБКИ ДАНИХ, АЛГОРИТМ, СЕРВЕР, МЕТОД, ЗАВАНТАЖЕННЯ.

Список публікацій здобувача:

1. «Балансування навантаження на прикладі дата-центрів соціальної мережі»
2. «Використання мережевих сховищ в мобільних додатках»
3. «Використання хмарних технологій в контексті нових послуг в 3g мережах операторів мобільного зв'язку».

ABSTRACT

Vdovenko O.A. Research of the load balancing mechanisms in data centers / Graduation qualifying work for obtaining an educational degree "Master" in specialty 172 "Telecommunications and Radio Engineering" - SHEE DonNTU Pokrovsky, 2018.

The work contains 92 pages, consists of an introduction, four sections, conclusions, a list of sources used from 40 titles, 16 figures, 10 tables.

In accordance with the task set in the work, the task of developing a dynamic load balancing algorithm in the networks of data centers was solved; The model of dynamical load balancing algorithm is presented and algorithms of input flow control, adaptive monitoring algorithm are developed.

Keywords: LOAD BALANCING, DATA PROCESSING CENTER, DATA CENTER, ALGORITHM, SERVER, METHOD, DELAY.

Publisher publication list:

1. " Load balancing with the example of social network data centers "
2. " Using network storage in mobile applications "
3. " The use of cloud technologies in the context of new services in 3g networks of mobile operators ".

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ЦОД – Центр обробки даних

ПЗ – Програмне забезпечення

IT – Information Technology

ОС – Операційна система

МЗД – Мережа зберігання даних

СКБД – Система керування базами даних

DNS – Domain Name System

IP – Internet Protocol

MAC – Media Access Control

ECMP – Equal-cost multi-path

OSI – Open systems interconnection

TCP – Transmission Control Protocol

DPI – Deep packet inspection

CPU – Central processing unit

ВСТУП

У зв'язку з підвищеним навантаженням і неоднорідністю трафіку стала актуальною проблема ефективного використання серверних потужностей. Одним з аспектів даної проблеми є ефективне планування і розподіл навантаження всередині розподілених обчислювальних систем ЦОД з метою оптимізації використання ресурсів і скорочення часу обчислення.

Відомі алгоритми управління вхідним потоком мають ряд недоліків: нерівномірний розподіл навантаження між серверами, низька продуктивність, недостатнє врахування динаміки системи, великі накладні витрати ресурсів, значний час відповіді. Аналізуючи недоліки класичних алгоритмів, можна зробити висновок про доцільність розробки динамічного контентозалежного алгоритму з модифікованою зворотним зв'язком. Алгоритм, буде враховувати тип запиту і динамічний стан сервера.

Метою магістерської роботи є розробка динамічного алгоритму балансування навантаження в мережах центрів обробки даних.

Об'єктом даного дослідження є системи балансування навантаження в центрах обробки даних і, зокрема, система балансування навантаження в програмно-обумовленому ЦОД.

Предметом досліджень в даній роботі є методи і алгоритми балансування навантаження додатків в центрах обробки даних, критерії їх порівняння, алгоритм балансування, який буде враховувати тип запиту і динамічний стан сервера.

РОЗДІЛ 1. ЦЕНТРИ ОБРОБКИ ДАННИХ

1.1. Загальне поняття центра обробки даних

Центр обробки даних являє собою відмовостійку комплексну централізовану систему, яка здійснює такі функції, як: зберігання, обробка та розповсюдження інформації [2].

ЦОД є окрема споруда (майданчик), обладнане відповідними інженерними системами кондиціонування, пожежогасіння, безперебійного електроживлення та ін., у якому розміщуються обчислювальні ресурси (сервери, системи зберігання і т.д.), створена телекомунікаційна інфраструктура, що забезпечує віддалений доступ до ресурсів ЦОД , створена система інформаційної безпеки тощо.

Структура ЦОД

У більшості випадків центр обробки даних складається з декількох основних інфраструктур:

- телекомунікаційна інфраструктура, дозволяє здійснити взаємозв'язок елементів ЦОД, і здійснює передачу між ЦОД і абонентами;
- інформаційна інфраструктура, складається безпосередньо з серверного обладнання, завдяки якому і забезпечується обробка і зберігання інформації;
- інженерна інфраструктура, призначена для забезпечення повноцінного функціонування головних систем ЦОД, і захисту від різних внутрішніх і зовнішніх факторів. Наприклад, в інженерну інфраструктуру входить протипожежна безпека, системи охолодження і т.д.

Обладнання, що використовується в ЦОД, розміщується в спеціальних шафах і стоках. Часто, для перешкоджання проникнення в ЦОД встановлюються камери відео-спостереження.

Ієрархічна модель центра обробки даних наведена на рисунку 1.1.

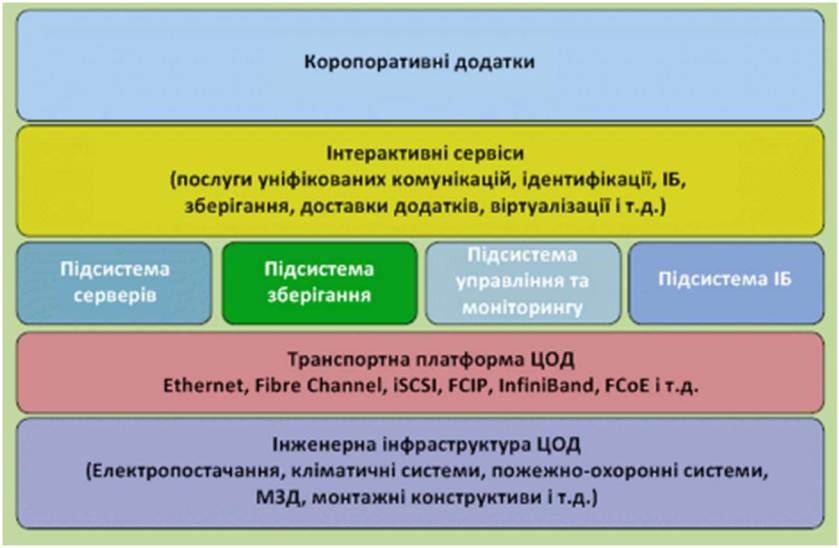

Рис.1.1. Ієрархічна модель ЦОД

Склад ЦОД. Обов'язкові компоненти, що входять до складу ЦОД, можна розділити на три основні групи:

1. Технічні компоненти. Вони створюють умови для ефективної роботи центру. До таких належать:

- серверний комплекс, включає сервери інформаційних ресурсів, додатків, надання інформації, а також службові сервери
- система зберігання даних і резервного копіювання - ядро ЦОД. Вона складається з консолідуючих дискових масивів, мережі зберігання даних, системи резервного копіювання та аварійного відновлення даних
- мережева інфраструктура забезпечує взаємодію між серверами, об'єднує логічні рівні і організовує канали зв'язку. Вона включає в себе магістралі для зв'язку з операторами загального доступу, телекомунікації, що забезпечують зв'язок користувачів з ЦОД
- інженерна система експлуатації ЦОД підтримує умови для нормального функціонування центру. До її складу входять підсистеми енергозабезпечення, кліматконтролю, пожежної сигналізації і

пожежогасіння, передачі даних, а також автоматизовані системи диспетчеризації, управління інформаційними ресурсами

- система безпеки запобігає несанкціонованому вторгненню в зони конфіденційної інформації. Вона складається з засобів захисту, системи оповіщення та системи контролю доступу

2. Програмне забезпечення. Це фактично сервіси інфраструктури ЦОД і ПЗ для коректної роботи бізнес-процесів, необхідних для конкретної організації. До компонентів інфраструктури відносяться:

- операційні системи серверів;
- програмне забезпечення баз даних;
- операційні системи робочих станцій;
- засоби кластеризації;
- засоби резервного копіювання;
- програми пристроїв зберігання даних;
- засоби адміністрування серверів і робочих станцій;
- засоби інвентаризації;
- офісне програмне забезпечення;
- електронна пошта;
- Інтернет-браузери.

До програм, які відповідають за функціонування бізнес процесів, відносяться:

- ділові додатки;
- базові корпоративні інформаційні сервіси;
- додатки для колективної роботи;
- галузеві компоненти;
- програмне забезпечення для вирішення завдань конструкторсько-технологічного плану системи електронного архіву та управління проектами;

– програми, що забезпечують сервіси файлів, друку, служби каталогів і інших прикладних задач.

3. Організаційне середовище вирішує питання, пов'язані з наданням IT-послуг. Воно повинно відповідати вимогам з надання IT-послуг, таким як ISO / IEC 20000. Тут представлені:

– процеси надання послуг, тобто якість і доступність послуг;
– процеси взаємовідносин між постачальником і клієнтом, а також з підрядними організаціями;
– процеси вирішення проблем, що виникають при функціонуванні будь-якого з компонентів системи;
– процеси управління конфігураціями, моніторинг і контроль статусу IT-інфраструктури, інвентаризація, верифікація та реєстрація конфігураційних одиниць, збір і управління документацією, надання інформації про IT-інфраструктуру для всіх інших процесів;
– процеси управління змінами, тобто визначення необхідних змін і способів їх проведення з найменшим ризиком для IT-послуг, а також проведення консультацій та координації дій з організацією в цілому;
– процеси релізу, тобто спільного тестування і введення в активну діяльність організації ряду конфігураційних одиниць.

Приблизний набір загальних вимог до дата-центру включає [1]:

– високий гарантований рівень доступності виділеного каналу зв'язку і інтернет з'єднання;
– гарантована швидкість відновлення працездатності при аваріях обладнання і збої ПЗ;
– наявність системи резервного копіювання, що відповідає вимогам по рівню збереження даних і швидкості їх відновлення;
– наявність системи моніторингу апаратно-програмних засобів (включаючи настройку, діагностування та оперативний контроль);
– цілодобова кваліфікована технічна підтримка обладнання та систем;

- наявність засобів виявлення та протидії мережевим вторгненням;
- наявність системи захисту від несанкціонованого доступу у внутрішні приміщення ЦОД;
- охорона зовнішнього периметра;
- наявність системи енергоживлення з резервуванням енерговведення, наявність джерел безперебійного живлення та дизель-генератора;
- електростатичний захист всього обладнання і підлоги;
- наявність резервованої системи кондиціонування і вентиляції;
- наявність системи виявлення спалаху і газового пожежогасіння.

Таким чином, вимоги пред'являються не тільки до інженерної інфраструктури, а й до організації ефективного управління IT-ресурсами, захисту систем, до каналів зв'язку, які повинні бути дубльовані і забезпечувати доступність даних і додатків. Всі ці вимоги спрямовані на досягнення високого рівня надійності та доступності корпоративних інформаційних систем, збереження даних, підвищення керованості IT-інфраструктури.

Принципи роботи ЦОД [9].

1. Віртуалізація (рис.1.2):

   Віртуалізація - надання набору обчислювальних ресурсів або їх логічних об'єднань, абстрагованих від апаратної реалізації і забезпечує при цьому логічний ізоляцію обчислювальних процесів, які виконуються на одному фізичному ресурсі. З її допомогою можна скоротити кількість одиниць обладнання в ЦОД та зекономити час на установку нових програм, також дозволяє ізолювати потенційно небезпечні оточення і створювати необхідні апаратні конфігурації. За рахунок віртуалізації досягається уявлення пристроїв, яких немає в ЦОД. Віртуальні машини можуть бути об'єднані в мережу.

   Віртуалізація ділиться на програмну і апаратну. При програмної віртуалізації ізоляція відбувається на рівні процесу операційної системи. Переваги програмної віртуалізації включають в себе доступність ресурсів для всіх ОС, зручний інтерфейс, можливість

швидко перемикатися між ОС, легкооновлювані гостьові ОС. Апаратна віртуалізація - віртуалізація з підтримкою спеціальної процесорної архітектури. На відміну від програмної віртуалізації, за допомогою даної техніки можливе використання ізольованих гостьових систем, керованих гіпервізором безпосередньо. Переваги апаратної - простота розробки, захищеність, ізоляція апаратних ресурсів, а також те, що гостьова операційна система не прив'язана до архітектури хост платформи.

Рис.1.2. Віртуалізація: а) програмна, б) апаратна.

2. Кластеризація:

   Необхідна основа для безперебійної діяльності бізнес процесів. Вона забезпечує об'єднання декількох серверів в єдину систему, шляхом установки програмних зв'язків між ними, координує їх роботу, виконує перерозподіл навантаження в разі збою в одному з серверів.

3. Масштабування:

   При створенні ЦОД необхідно врахувати чи змінюються з часом потреби в ресурсах. Тому або використовують максимальну потужність пристроїв, або модульну систему. Це означає, що в процесі

розвитку є можливість додавати необхідні модулі без заміни існуючого обладнання. Така система більш економічно доцільна.

4. Резервування (рис.1.3):

   Запорука того, що при виході з ладу однієї підсистеми центру, функцію її візьме на себе інша, резервна. Є також поняття резервний ЦОД. Він починає працювати тоді, коли основний ЦОД повністю припинив роботу.

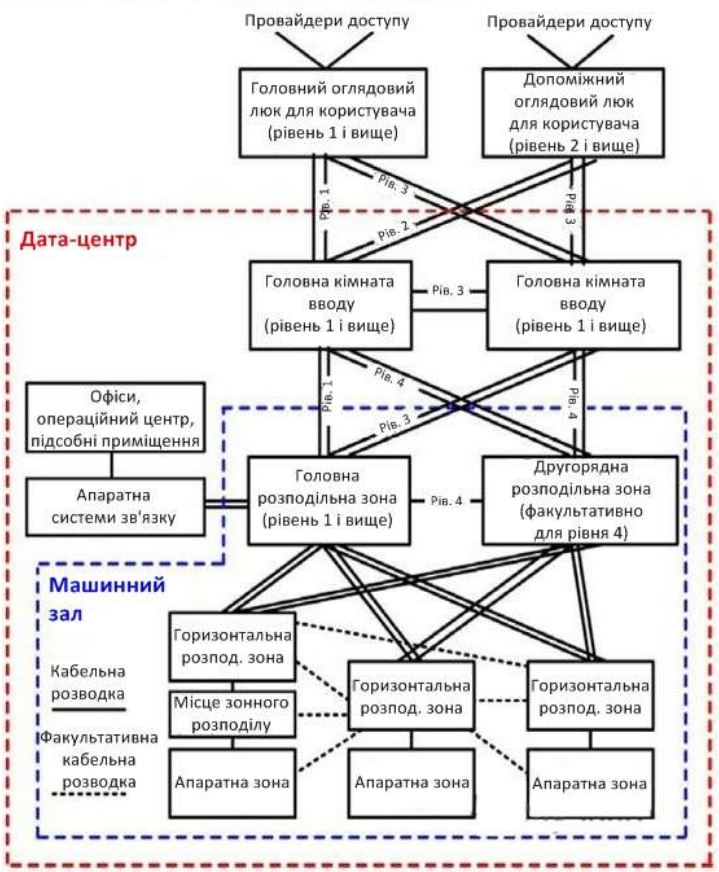

Рис.1.3. Резервування систем ЦОД

Класифікация ЦОД

За розмірами ЦОДи можна розділити на кілька типів:

– великі ЦОДи. Такі ЦОД мають власну будівлю, яке створено спеціально для створення ідеальних умов, необхідних для ЦОД. У більшості випадків великі ЦОД володіють власними каналами зв'язку, яким здійснюється підключення серверів.

– Середні ЦОДи. Даний тип ЦОД в більшості випадків бере в оренду приміщення необхідного розміру, а також канали певної пропускною спроможністю;
– Малі ЦОДи. Вони розташовуються в погано пристосованих для цього приміщеннях. У таких ЦОД дуже часто використовується обладнання низької якості, а перелік послуг представлений по мінімуму.
– Так само можна виділити контейнерні центри обробки інформації. Головна їхня особливість полягає в тому, що такі ЦОД можливо перевозити за допомогою автомобільного та залізничного транспорту.

За характером використання ЦОД можна розділити на

– Корпоративні. Розгортаються конкретною організацією і використовуються для вирішення її власних завдань, наприклад, для розміщення вузлів IT або системи резервного зберігання даних. Як правило, їх створення вимагає великих витрат і пов'язане з необхідністю обслуговувати всі ресурси самостійно.
– Комерційні. Призначені для обслуговування як телекомунікаційних компаній, що розвивають послуги хостингу / Colocation, так і організацій, що бажають скористатися послугами аутсорсингу.

1.2. Послуги, які надають ЦОД

На базі комерційних дата-центрів організовано надання послуг з розміщення інформаційних ресурсів російських і зарубіжних компаній, корпорацій і відомств. Абонентам дата-центрів зазвичай надається наступний типовий набір послуг [8]:

– Colocation (розміщення обладнання) - клієнт використовує технічні ресурси дата-центру для розміщення своїх серверів, систем зберігання даних, телекомунікаційного та іншого обладнання. Клієнт здійснює підтримку своїх ресурсів самостійно, а установку устаткування в

стійки і підключення виробляють фахівці дата-центру. Дана послуга позбавляє орендарів або власників офісних площ від необхідності нарощування власних IT-потужностей: установки серверного обладнання, організації локальної мережі і зовнішніх каналів зв'язку і проведення інших робіт, а також від утримання кваліфікованого обслуговуючого персоналу. В цьому випадку всі питання, пов'язані з експлуатацією корпоративної інформаційної системи, клієнт віддає на аутсорсинг спеціалізованим компаніям, розміщуючи свої сервери на технічних майданчиках оператора дата-центру. Постачальники послуги, в свою чергу, гарантують постійну, надійну і безперебійну роботу інформаційної системи, безпеку збережених даних, швидкісний Інтернет та інші необхідні послуги;

- Dedicated server (виділений сервер) - надання серверів в оренду. Клієнт може орендувати сервер у власника дата-центру. Послуга широко поширена серед клієнтів, які прагнуть уникнути великих разових вкладень в устаткування на початковому етапі розвитку IT-системи. Цей сервіс дозволяє на етапі становлення компанії не ризикувати інвестиціями, необхідними для створення власної IT-інфраструктури;
- Telehousing - послуги з розміщення телекомунікаційної та IT-інфраструктури замовника. Оператори зв'язку можуть розмістити свій вузол зв'язку в дата-центрі, встановлюючи обладнання повністю або частково, організувати підключення мережі до мереж інших операторів;
- Shared - оренда дискового простору на обладнанні власника дата-центру з заздалегідь узгодженої платформою і конфігурацією операційної системи. Дата-центр забезпечує підтримку (технічну і програмну) додатків клієнта, наприклад, підтримку скриптів. Корпоративні клієнти можуть використовувати послуги для повного або часткового розміщення обладнання, систем і інфраструктури.

Також можлива організація підключення обладнання замовника до мереж операторів зв'язку.

На базі дата-центрів можуть надаватися і більш складні послуги, наприклад, такі як:

- аутсорсинг інформаційних систем - послуга, яка надається за довготривалою угодою, за якою постачальник послуг одержує в повне управління або у власність всю IT-інфраструктуру клієнта або значну її частину, включаючи обладнання та встановлене на ньому програмне забезпечення. Це проекти з залученням в роботу виконавця, які передбачають відповідальність за системи, мережу і окремі програми, що входять в IT-інфраструктуру. IT-аутсорсинг оформляється у вигляді довгострокового контракту;
- хостинг обслуговування і адміністрування ПЗ - має на увазі централізоване управління тиражованим програмним забезпеченням за умови, що додатки знаходяться на території постачальника послуг, а замовник має віддалений доступ до програмного забезпечення. Основні відмінності між хостингом обслуговування і адмініструванням ПЗ і традиційним управлінням додатками полягають в наступному:
  - додатки замовника працюють, перебуваючи в інформаційному центрі постачальника послуг,
  - інформаційний центр постачальника послуг працює за схемою «один - багатьом»,
  - постачальник послуг не несе відповідальності за апаратні засоби замовника;
- хостинг інфраструктурних послуг - це сервіс з надання стандартних елементів IT-інфраструктури в віддалене користування на певний період. Мається на увазі управління серверами і мережевими рішеннями в інформаційному центрі іншої компанії для управління виділеним або розділеним веб-доступом, підтримки електронної

комерції, управління складовими і системами безпеки. Включає послуги з розгортання системи з її подальшим віддаленим керуванням.

Найперспективнішою є послуга впровадження хмарних технологій. Це дуже різнопланова і перспективна галузь, яка набуває все більшої популярності. Так, за даними дослідження, проведеного українським підрозділом IDC [10]., якщо в 2014 році 60% респондентів стверджували, що відмінно або добре знайомі з «хмарами», то в 2015 таких уже майже 80% (рис.1.4). Сам факт зростання популярності «хмарних» сервісів говорить про позитивне сприйняття таких послуг більшістю користувачів.

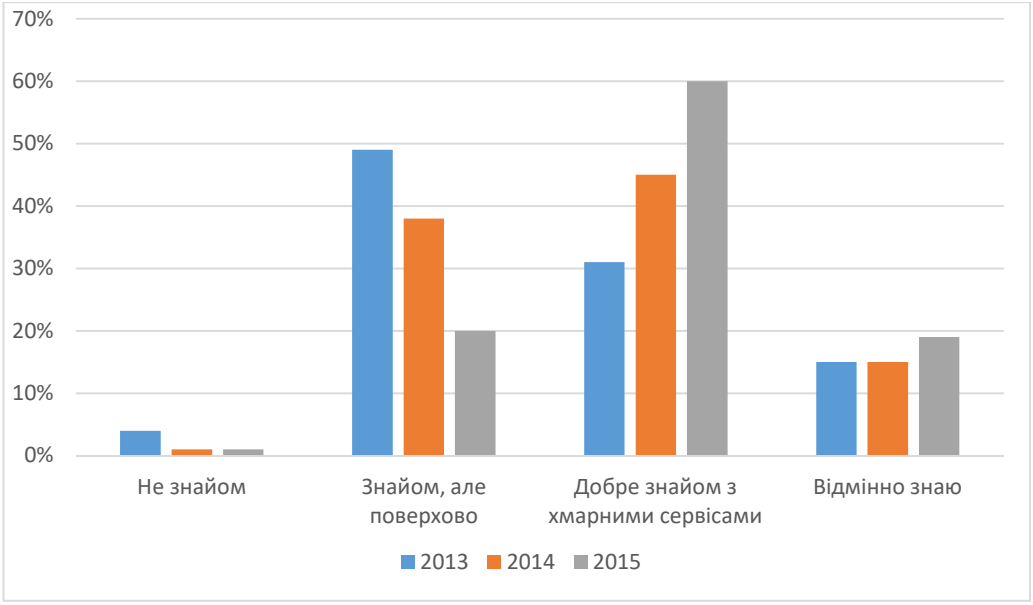

Рис. 1.4. Тенденції зростання ознайомлення населення з хмарними сервісами

Основні характеристики хмарних обчислень:

- самообслуговування / каталог IT послуг;
- доступ через мережу з різних пристроїв;
- миттєва еластичність;
- вимірність IT-послуг;
- на багато користувачів (multimenant) загальний пул ресурсів.

«Хмара» ділиться на три моделі [12].:

- SaaS. Software as a Service, «Оренда». Додаток для кінцевих користувачів, що надаються як сервіси, а не як традиційні встановлюємі ПЗ;
- PaaS. Platform as a Service, «Розробка». Програмна платформа або проміжне ПЗ, за допомогою якого розробники можуть будувати і розгортати користувацькі додатки;
- IaaS. Infrastructure as a Service, «Експлуатація». Обчислення, сховище і інша IT-інфраструктура, що надаються як сервіси, а не як виділені ресурси.

Розглядаючи дані моделі з точки зору балансування навантаження, для постачальника послуги важливіше балансувати SaaS, так як за всі налаштування і стабільне надання додатків відповідає постачальник, в IaaS користувач відповідає за всі процеси, тому відповідальність за балансування полягає на ньому.

Статистика використання: 75-80% всіх спожитих сервісів припадає на модель IaaS, 12-13% - на SaaS і 6-7% - на PaaS. Зовсім невелика частка - на додаткові і сервіси, що розвиваються (на нашому ринку), такі як Storage as a Service («МЗД як послуга»), Backup as a Service (створення та зберігання резервних копій в «хмарі»), Recovery as a Service («аварійне відновлення як послуга»), Database as a Service (підтримка СКБД організації в «хмарі оператора») і деякі інші.

Існує ряд похідних типів послуг, приклади яких наведені в таблиці 1.1.

Таблиця 1.1. Похідні типи послуг

Назва послуги (англ.)	Назва послуги (укр.)	Короткий опис
1	2	3

Workplace as a Service (WaaS)	Робоче місце як послуга	Дозволяє компанії використовувати «хмарні» ресурси (в т.ч. ПЗ, віртуальні ПК, сервери, МЗД) для організації робочих місць співробітників

Продовж. табл.1.1

1	2	3
Storage as a Service (STaaS)	МЗД як послуга	Надають користувачеві дисковий простір в «хмарі», яке може використовуватися як в якості простого диска (Disc as a Service), так і для створення повноцінних віртуальних мереж зберігання даних, SAN
Storage as a Service (STaaS)	МЗД як послуга	Надають користувачеві дисковий простір в «хмарі», яке може використовуватися як в якості простого диска (Disc as a Service), так і для створення повноцінних віртуальних мереж зберігання даних, SAN
Security as a Service (SECaaS)	Інформаційна безпека (і / або шифрування даних) як послуга	Забезпечує безпечне використання веб-технологій, в тому числі шифрування з'єднання та / або даних, які передаються клієнтом за допомогою «хмарного» сервера
Backup as a Service (BaaS)	Відновлення даних як послуга	Дозволяє зберігати резервні копії даних на «хмарних» дисках і МЗД, а при необхідності швидко їх відновлювати
Recovery as a Service (RaaS)	Відновлення IT-інфраструктури як послуга	Дозволяє створити повну віртуальну копію IT-інфраструктури в хмарі провайдера. При аварії основного майданчика можна швидко перевести її

		завдання на резервну «хмарну» інфраструктуру
Logging as a Service (LaaS)	Журнал роботи як послуга	Фіксація і зберігання інформації про будь-які дії, що відбуваються з важливими даними. Всі зміни записуються в лог-файли і зберігаються в «хмарі»

Продовж. табл.1.1

1	2	3
Database as a Service (DBaaS)	СКБД як послуга	Розгортання, підтримка і адміністрування СКБД в «хмарі» силами фахівців провайдера послуги
Load ballancing as a Service (LBaaS)	Баланс навантаження як сервіс	Розподіл навантаження між декількома дата-центрами, контроль політики забезпечення SLA додатків клієнтів і прискорення роботи додатків. Дозволяють клієнту вивільнити до 30-50% серверних потужностей і заощадити на закупівлю або оренду обладнання.
CRM as a Service	CRM як послуга	Використання ресурсів хмарних CRM-систем в «хмарі» провайдера
ERP as a Service	ERP як послуга	Використання ресурсів «хмарних» ERP-систем в хмарі провайдера

Virtual PBX as a Service	Віртуальна АТС як послуга	Надання в оренду ресурсів «хмарних» IP-АТС
Unified Communications as a Service (UCaaS)	Уніфіковані комунікації як послуга	Надання в оренду систем Unified Communications, розгорнутих в «хмарі» провайдера
Video Conferencing as a Service	Відеоконференцзв'язок як послуга	Надання в оренду систем ВКЗ, розгорнутих в «хмарі» провайдера

Продовж. табл.1.1

1	2	3
Document automation as a Service	Електронний документообіг як послуга	Надання в оренду систем електронного документообігу, розгорнутих в «хмарі» провайдера
CCTV as a Service	Охоронне відеоспостереження як послуга	Надання платформи з «хмари» для організації віддаленої системи відеоспостереження через мережу Інтернет. Також забезпечується зберігання і захист відеоданих
Bookkeeping as a Service	ПЗ для бухгалтерського обліку як послуга	Надання в оренду програм бухгалтерського обліку, розгорнутих в «хмарі» провайдера
Inventory accounting as a Service	Складський облік як послуга	Надання в оренду програм складського обліку, розгорнутих в «хмарі» провайдера

1.3 Висновки

Отже ЦОД – це відмовостійка комплексна централізована система, яка здійснює такі функції, як: зберігання, обробка та розповсюдження інформації. Серед послуг, які може надавати центр обробки даних, найобширнішими є “хмарні” послуги. Найпопулярнішими моделями “хмари” є IaaS, Paas та SaaS. Для того, щоб послуги надавалися безперебійно та швидко треба впроваджувати балансування навантаження.

РОЗДІЛ 2. ВИМОГИ ДО ПОБУДОВИ ЦЕНТРІВ ОБРОБКИ ДАНИХ

2.1. Стандарти ЦОД

У багатьох країнах існують стандарти, призначені для обладнання, що знаходяться у центрах обробки даних. За допомогою даних стандартів з'являється можливість дати оцінку рівня сервісу ЦОД. Тому в США існує спеціальний стандарт TIA-942, в який входять рекомендації по створенню центрів обробки даних, а також ділить їх за рівнем надійності.

У світі TIA-942 прийнято вважати єдиним стандартом для центрів обміну даними, але його мінусом є те, що він протягом довгого часу не отримав оновлення. На даний момент отримав активний розвиток стандарт BICSI 002 2010 Data Center Design and Implementation Best Practices, який вперше з'явився в 2010 році, але був оновлений вже в 2011 році. BISCI створений з метою доповнення вже існуючих стандартів.

Рівні надійності ЦОД

Головна характеристика рівня роботи ЦОД - відмовостійкість, але не менш важливими є вартість експлуатації, показник витрати електроенергії і регулювання температури.

Розроблена кілька років тому компанією Uptime Institute система стандартів на базі рівнів відмовостійкості Tier знайшла відображення в стандарті TIA / EIA-942 і широко використовується у всьому світі. Класифікація Tier враховує найдрібніші деталі: згідно з нею рівень надійності ЦОД визначається топологією, потужністю, схемами резервування інфраструктури та безліччю інших чинників.

Стандарт визначає чотири рівні відмовостійкості ЦОД - чим вище рівень, тим вище вимоги. Доступність рівня Tier I, концепція якого була розроблена в 60-і роки, становить 99,671%; це означає, що запланований час простою не перевищує 28,8 години на рік. Рівень Tier IV, що з'явився в 90-і

роки, - 99,995%; регламентні роботи можуть проводитися без відключення обладнання, а допустима тривалість зупинки - 0,4 години на рік. На відміну від першого рівня, четвертий передбачає повне резервування, в тому числі фідерів електроживлення і вхідних магістралей від провайдерів послуг зв'язку. Мова йде не тільки про можливості проведення запланованих профілактичних робіт без припинення роботи майданчика. Об'єкт може витримати і незаплановані інциденти, операційні помилки, вихід з ладу компонента або перебій в живленні без нанесення шкоди IT. Архітектура Tier IV дозволяє автоматично нейтралізувати наслідки збою і стримувати його подальший вплив. Вартість ЦОД четвертого рівня надзвичайно висока, що обумовлено критичністю вирішуваних завдань. Нерідко четвертий рівень доступності забезпечується шляхом створення резервного ЦОД.

За стандартом TIA-942 виділяють 4 рівня надійності ЦОД:

- Tier 1 (N) - найбільш ненадійний ЦОД, в якому, у випадку виникнення неполадок і відмови устаткування або на початку ремонтних робіт, відбувається зупинка всього центру обміну даними. Більш того в такому ЦОД немає фальшпідлоги і додаткових джерел живлення, а інженерна інфраструктура не зарезервована.
- Tier 2 (N + 1) - з'явився маленький рівень резервування, так само в такому ЦОДі присутні фальшпідлоги і додаткові джерела живлення, але у випадку початку ремонтних робіт ЦОД призупинить свою роботу.
- Tier 3 (N + 1) - в цьому типі ЦОДа вже з'явилася можливість здійснення ремонтних робіт без припинення роботи ЦОД. Так само варто відзначити, що інженерні системи одноразово зарезервовані і існує кілька каналів службовців для розподілу електроживлення та охолодження, але варто відзначити, що постійно працює лише один з них.
- Tier 4 (2 (N + 1)) - в даному типі ЦОДа стало доступно здійснення всіх типів робіт без скасування роботи ЦОДа. Більш того інженерні

системи тут зарезервовані дворазово, це означає, що продубльована основна і додаткова системи.

Таблиця 2.1. Рівні надійності ЦОД

Параметр / Клас ЦОД (рівень)	**Tier I**	**Tier II**	**Tier III**	**Tier IV**
Тип будівлі	С сусідами	С сусідами	Окремо розташована	Окремо розташована
Кількість енерговведення	1	1	Один активний, другий резервний	Два активних
Схема резервування компонентів	N	N + 1	N + 1	2 (N + 1) або S + S
Початкова потужність з розрахунку Вт на м ^{2}	215-323	430-537	430-645	537-860
Максимальна потужність Вт на м ^{2}	215-323	430-537	1075-1615	1615+
Безперебійне кондиціонування	немає	немає	можливо	є
Нормативне навантаження на фальшпідлогу, кг на кв.м.	415	488	732	732+ (за стандартом 2005 - 1000 +)
Загальна тривалість відмов за рік, годин	28,8	22	1,6	0,4
Доступність ЦОД	99,671%	99,749%	99,982%	99,995%
Обслуговування	ЦОД повинен повністю зупинятися для регламентних робіт	Допускаються перерви в роботі на технічне обслуговування і ремонт	Цілодобова зміна по буднях, у вихідні за викликом	Цілодобова чергова зміна

2.2. Етапи створення ЦОД

Процес створення центру обробки даних багатоступінчастий. Він включає 5 основних етапів:

- Планування - складання технічного завдання і вироблення концепції будівництва. Організаторами детально опрацьовуються мети, методи і засоби для створення цього проекту, визначається внутрішня структура комплексу. При плануванні враховуються регіональні особливості та спектр діяльності організації, проводиться економічна оцінка та обгрунтування інвестицій. В кінцевому підсумку вибирається майданчик для будівництва, що відповідає всім необхідним вимогам.
- Конструювання - узгодження всіх складових ЦОД з метою створення єдиного комплексу і введення його в експлуатацію. Іншими словами, це рішення задач, поставлених на етапі планування ЦОД. Також проводиться випробування компонентів з урахуванням навантаження, визначення можливості задоволення поточних і майбутніх запитів бізнесу, робиться прогноз здатності IT-інфраструктури відповідати запитам діяльності організації, яка розвивається, в цілому. Крім того, розраховуються різні ризики при введенні ЦОД в експлуатацію та шляхи їх зниження. Триває підготовка необхідної документації, пов'язаної з функціонуванням окремих компонентів: обладнання, програмного забезпечення, інженерних систем. Увесь процес конструювання закінчується створенням проекту, який узгоджується з замовником і з контролюючими органами.
- Реалізація - найбільш трудомісткий етап в процесі створення ЦОД. Він зводиться до комплектації та постачання необхідного обладнання, монтажу, впровадження основних компонентів, тестування їх і створення організаційної структури. На цьому етапі обирається та впроваджується балансування навантаження систем.
- Експлуатація - це безпосереднє обслуговування комплексу та організація роботи підрозділів. На цьому етапі регламентуються перевірки всіх систем. Терміни встановлюються в залежності від типу компонентів. Проводиться технічне обслуговування серверів. В

результаті планових перевірок заповнюється журнал регламентних робіт.

- Модернізація - структурні і технічні зміни ЦОД, спрямовані на поліпшення його роботи. За статистичними даними новітнє обладнання морально застаріває в термін до 5 років. За цей період організація розвивається, прогресує її діяльність, розширюється мережа філій. Відповідно зростають і IT-потреби. Для цього необхідно вчасно проводити процес модернізації IT-інфраструктури. Також необхідно переглядати методи, які використовуються для балансування навантаження

2.3. Тенденції розвитку

Під впливом нових можливостей можна очікувати наступні технологічні зміни глобального масштабу в області ЦОД [5]:

- ЦОД скористаються технологіями інтернету речей

  Використання технологій Інтернету Речей у центрах обробки даних дозволить IT-менеджерам відстежувати в реальному часі стан компонентів і фізичних показників навколишнього середовища для забезпечення безперебійного виконання обчислень. Датчики, що вимірюють температуру, вологість і електрику, будуть комбінуватися з мережевим обладнанням моніторингу центрів обробки даних для підтримки високої доступності, зниження капітальних і експлуатаційних витрат. У ЦОД з'явиться більше доступних платформ, в тому числі інтегруючі дані з безлічі різних джерел для забезпечення оптимальних режимів функціонування обчислювальних потужностей. Близько 95% представників західного бізнесу очікують отримання переваг від використання Інтернету Речей в найближчі два роки.
- Гіперконвергенція

Гіперконвергентні системи, які об'єднують ключові компоненти IT-систем в єдиному рішенні, керованому за допомогою ПЗ, будуть відігравати все більш важливу роль в інфраструктурі ЦОД. Вимоги бізнесу сприяють спрощення інфраструктури і зростання окупності інвестицій в IT-департаменти. Віртуалізація стає рушійною силою гіперконвергенції, особливо через неефективність класичних систем зберігання даних SAN у віртуальному середовищі, - вона формує потребу в віртуалізації МЗД та мережевої інфраструктури. Як тільки конвергенція стане швидкозростаючим ринковим трендом, для управління конвергентної інфраструктурою будуть потрібні платформи, які дозволять забезпечити єдиний інтегрований підхід і подолати розрив між віртуальними мережами і фізичної інфраструктурою.

- Програмно-керована інфраструктура

  Інфраструктура стає все більш програмно-керованої (наприклад, програмно-керовані мережі): операції автоматизуються, що виключає ручні настройки і переналаштування на рівні апаратних компонентів. Це забезпечує зростання гнучкості, мінімізацію кількості помилок і зниження операційних витрат. Дані будуть представлені таким чином, що вони стануть максимально доступні. У зв'язку з цим очікується підвищена увага до програмних платформ, які можуть забезпечити комплексний моніторинг пристроїв і з'єднань для збільшення ефективності управління та обслуговування. При наявності бази даних про всі компоненти IT-інфраструктури, точної і актуальної, з оновленням в режимі реального часу, функціонування інформаційної системи можна буде набагато швидше відновити після системних збоїв, не тільки в разі проблем з віртуальними сервісами, а й у разі поломок фізичних елементів або пристроїв.
- Масштабованість

Підприємства будуть прагнути копіювати те, що сьогодні спроектовано і використовується Facebook, Google і Amazon: досить гнучку IT-середовище, яке може легко розширюватися і стискатися залежно від вимог бізнесу. Але вони не готові розлучитися з відмовостійкістю, до якої звикли, з традиційними архітектурами і технологіями. Легкомасштабовані інфраструктури, з традиційно високим рівнем резервування та відмовостійкості, на базі програмно-визначених підходів, стають все більш популярними.

- Автоматизація для підвищення ефективності праці

  Автоматизація управління центром обробки даних стане нормою для зниження робочого навантаження і людських помилок, а також для прискорення реакції на поломки обладнання. З введенням системи, наприклад, використання контролю IP для автоматичної перевірки, відбудеться зміна підходу до адміністрування IT. У новому середовищі IT-менеджери будуть використовувати нові навички і зможуть легко позбутися рутинних і повторюваних завдань. Організації будуть наймати фахівців з абсолютно новим набором навичок в галузі управління ЦОД, фокусуючись на автоматизації, інтеграції технологій за допомогою API, віддачі від користувацького досвіду і знань в області інтеграції нового зі старим.

2.4. Висновки

В даному розділі наведені стандарти, етапи створення та тенденції розвитку центрів обробки даних. Балансування навантаження є одним з найважливіших етапів при реалізації ЦОД, треба враховувати завантаженість ЦОД, яка очікується, та тенденції його розвитку. Існуючи методи та алгоритми балансування навантаження представлені у наступному розділі.

РОЗДІЛ 3. БАЛАНСУВАННЯ НАВАНТАЖЕННЯ. МЕТОДИ ТА АЛГОРИТМИ

3.1. Балансування навантаження

В обчислювальній техніці балансування навантаження розподіляє навантаження між декількома обчислювальними ресурсами, такими як комп'ютери, комп'ютерні кластери, мережі, центральні процесори або диски. Цілі балансування навантаження:

- оптимізація використання ресурсів,
- максимізація пропускної здатності,
- зменшення часу відгуку
- запобігання перевантаження будь-якого одного ресурсу.

Використання декількох компонентів балансування навантаження замість одного може підвищити надійність і доступність за рахунок резервування. Балансування навантаження передбачає зазвичай наявність спеціального програмного забезпечення або апаратних засобів, таких як багаторівневий комутатор або система доменних імен, як серверний процес.

Проблема розподілу обчислювального навантаження виникає через те, що:

- різні обчислювальні потужності потрібні для різних логічних процесів, тобто розподілений додаток має неоднорідну структуру
- обчислювальний комплекс (наприклад, кластер) також має неоднорідну структуру - різна продуктивність у різних обчислювальних вузлів
- неоднорідна структура міжвузлових взаємодій, через це лінії зв'язку, які з'єднують вузли, можуть володіти різними характеристиками пропускної спроможності

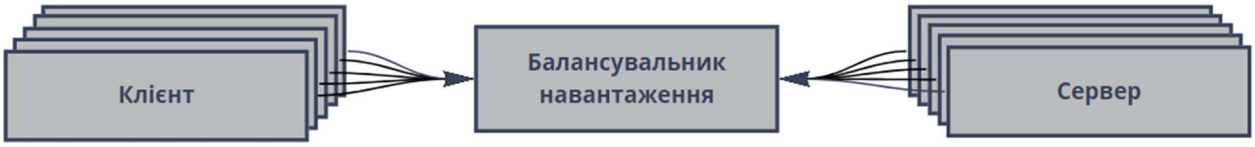

Рис.3.1. Балансування мережевого навантаження

Деяка кількість клієнтів запитує ресурси з певної кількості серверів (рис.3.1.). Балансувальник навантаження знаходиться між клієнтами і внутрішніми компонентами і на високому рівні виконує кілька важливих завдань:

- Виявлення служб: Які сервери доступні в системі? Які їх адреси (наприклад, як повинен працювати балансувальник навантаження)?
- Перевірка на працездатність: Які сервери в даний час працездатні і доступні для прийому запитів?
- Розподіл навантаження: Який алгоритм слід використовувати для балансування окремих запитів по працездатних серверах?

Переваги правильного використання балансування навантаження в центрах обробки даних:

- **Іменування абстракції**: Немає необхідності в тому, щоб кожен клієнт знав про кожен сервер (service discovery). Клієнт може звертатися до балансувальника навантаження через визначений механізм, а потім дію дозволу імен можна делегувати в балансувальник навантаження. Зумовлені механізми включають вбудовані бібліотеки і DNS / IP / port.
- **Відмовостійкість**: Завдяки перевірці працездатності та різним алгометричним методам балансувальник навантаження може ефективно маршрутизувати навколо непрацездатного чи перевантаженого сервера. Це означає, що адміністратор може спокійно, без поспіху усунути проблему на нездоровому вузлі та користувачі ЦОД не відчують проблем в роботі .

– **Вартість і продуктивність**: Розподілені системні мережі рідко бувають однорідними. Найбільш ймовірно, що система буде охоплювати кілька мережевих зон і регіонів. Інтелектуальне балансування навантаження може максимально збільшити трафік запитів в зонах, що збільшує продуктивність (менше латентності) і знижує загальну вартість системи (менше смуги пропускання і прокладки оптоволокна, необхідного між зонами).

Кроки балансування навантаження

У більшості випадків для практичного і повного вирішення задачі балансування навантаження виконуються чотири пункти [13]:

– Оцінити завантаження обчислювальних вузлів
– Ініціювати балансування навантаження
– Прийняти рішення про балансування
– Перемістити об'єкти.

Для умов ЦОД ці кроки виконуються наступним чином:

Оцінка завантаження

На цьому етапі приблизно оцінюється завантаження кожного процесора. Інформація про завантаження, яка була отримана, використовується як база даних для процесу балансування. Це робиться для того, щоб визначити виникнення дисбалансу і новий розподіл об'єктів імітаційної моделі, обчисливши обсяг робіт, який необхідний для переміщення об'єктів. Виходячи з цього, якість роботи балансування навантаження безпосередньо залежить від точності і повноти інформації в базі даних.

Два типу даних, в основному, складають таку базу даних:

– Інформація про роботу розподіленого додатка. У такі дані входять час, який потрібен для виконання окремого завдання, час простою, інтенсивність обміну інформацією і т.д.

– Інформація про роботу процесора (інформація рівня процесора). Включає в себе завантаження процесора, фонове завантаження процесора, час, який процесор простоює, швидкість передачі інформації по лініях зв'язку і т.д.

Ініціалізація балансування навантаження

Якщо занадто часто виконувати балансування навантаження, то може відбутися уповільнення виконання імітаційної моделі. Можлива вигода від проведення балансування може не перевершити витрати на один її процес.

Тобто для того, щоб балансування було продуктивне, необхідно знайти спосіб для визначення моменту його ініціалізації.

А саме:

– Визначити момент, коли виникає дисбаланс навантаження.
– Визначити наскільки необхідне балансування, порівнявши можливу користь від його проведення і витрати на нього.

Визначення дисбалансу навантаження може бути синхронним і асинхронним.

Синхронне визначення дисбалансу: порівнюється навантаження окремого процесора із загальним середнім завантаженням, при перериванні роботи в певні моменти синхронізації

Асинхронне визначення дисбалансу: обсяг дисбалансу обчислює фоновий процес, який працює паралельно з додатком (кожен процесор зберігає історію свого завантаження, момент синхронізації визначення ступеня дисбалансу відсутній).

Прийняття рішень в процесі балансування

Велика частина стратегій динамічного балансування навантаження відноситься або до класу повністю розподілених, або до класу централізованих.

У централізованій стратегії глобальна інформація про стан всієї обчислювальної системи збирається спеціальним комп'ютером, він вирішує куди переміщати завдання для кожного з комп'ютерів.

У повністю розподіленої стратегії кожен процесор обмінюється інформацією про стан з іншими і виконується алгоритм балансування навантаження. Переміщення можливо тільки між сусідніми процесорами.

Переміщення об'єктів

Після того, як рішення про балансуванню прийнято, об'єкти переміщаються серед процесорів, щоб був досягнутий новий баланс навантаження. Під час переміщення об'єкт повинен зберігати цілісність свого стану. Допоміжні функції програми використовуються при переміщенні даних об'єкта, особливо коли маються на увазі складні структури даних, такі як списки посилань або покажчиків. В "хмарних" обчисленнях цей крок замінюється на переадресацію запитів, які надходять в систему на відповідний екземпляр.

3.2. Види балансування навантаження

Локальне балансування

У разі, якщо балансування навантаження потрібне в межах одного центру обробки даних, то використовується локальне балансування навантаження. Воно здійснюється на таких рівнях моделі OSI:

- канальний (L2)
- мережевий (L3)
- транспортний (L4)

Балансування на канальному рівні. Певному спеціалізованому інтерфейсу всіх серверів присвоюється одна IP-адреса ресурсу, на який будуть приходити запити, і з якого будуть йти відповіді. На ARP-запит з цієї IP-адреси

сервери не повинні відповідати, тому така же IP-адреса присвоюється балансувальнику, відповідно, на нього будуть приходити запити, і відправлятися відповіді з нього, і він же буде відповідати на ARP запити. Тобто отримуючи запит від клієнта, балансувальник вибирає за певним алгоритмом той чи інший сервер, який буде обробляти цей запит, підміняє destination MAC і відправляє його на обробку на даний сервер. Сервер його у себе обробляє, і так як підміна заголовків на мережевому рівні не робилася, то, минаючи балансувальник, відразу відповідає клієнту через шлюз.

Для реалізації без балансувальника необхідно перетворити вхідний unicast запит в broadcast. Тобто запит приходить на ресурс, шлюз його розмножує до всіх серверів, запити надходять на всі сервери одночасно, і кожен сервер повинен сам розуміти, чи повинен він відповідати на запит чи ні (наприклад, можна поставити ділення на нуль srcIP)

Балансування на мережевому рівні. Механізм схожий з балансуванням на канальному рівні, тільки в даному випадку, отримуючи вхідний запит, балансувальник підміняє destination IP, відповідно, застосовує його на той сервер, який буде обробляти запит. Сервер отримує його, обробляє і повинен передати його назад на балансувальник, щоб той виконав зворотну заміну.

Балансування на транспортному рівні. Відрізняється від мережевого тим, що в даному виді використовуються вхідні порти джерела і адресата.

Приклад реалізації даного виду балансування - це балансування за допомогою так званого алгоритму ECMP (equal-cost multi-path). Всі сучасні роутери можуть розподіляти, балансувати навантаження самі. Для цього достатньо на роутер анонсувати одну і ту ж підмережу за різними маршрутами. Балансувальник в даному випадку, маючи два однакових маршруту, по своїх загальних метриках буде розподіляти навантаження рівномірно. Є два нюанси даного способу:

– Роутер повинен розподіляти навантаження таким чином, щоб пакети в рамках однієї TCP сесії потрапляли на один і той же сервер.
– Якщо прописати на роутері статичні маршрути, то необхідно автоматизувати процес додавання і видалення серверів з кластера. Для того, щоб уникнути цієї проблеми, можна використовувати різні протоколи маршрутизації, такі як BGP. Тобто на кожен сервер встановлюється якийсь програмний BGP роутер, який буде анонсувати серверну мережу на роутер, приймаючий запити.

Переваги та недоліки локальних методів наведені в таблиці 3.1.

Таблиця 3.1. Переваги та недоліки локальних методів

<table>
<tr><td></td><td>Канальний</td><td>Мережевий</td><td>Транспортний</td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td rowspan="5">Плюси</td><td colspan="3">Не залежить від протоколів верхніх рівнів</td></tr>
<tr><td colspan="3">Використовується тільки один публічний адресу</td></tr>
<tr><td>Скорочення витрат за рахунок відмови від виділеного балансувальника</td><td rowspan="3">Повна прозорість роботи для серверів</td><td rowspan="3">Відсутність необхідності купувати додаткове обладнання</td></tr>
<tr><td>Зворотний трафік в сторону клієнта не навантажує балансувальник</td></tr>
<tr><td>Швидке додавання і відключення сервера в кластері</td></tr>
<tr><td rowspan="4">Мінуси</td><td rowspan="3">Необхідно розміщувати сервера в одному сегменті</td><td rowspan="4">Підвищене навантаження на балансувальник за рахунок зворотного трафіку</td><td>Необхідно ставити на сервера додатковий софт</td></tr>
<tr><td>Відсутня server-affinity (всі з'єднання розриваються при додаванні / виключення сервера)</td></tr>
<tr><td>Обмеження кількості ECMP на роутерах</td></tr>
<tr><td>Обмеження по вхідній</td><td>Рівномірність розподілу</td></tr>
</table>

	смузі (у випадку з shared address вхідний трафік потрапляє на всі сервери одночасно)		навантаження потребує вимогу до вирівнювання продуктивності серверів

Продовж. табл.3.1

1	2	3	4
			Таймаути BGP призводять до розподілу навантаження на недоступний сервер
Реалізації	Linux Virtual	Linux Virtual ServerServer,різні апаратні реалізації	Cisco 3750X, 4500-X, 6500 + Sup2T

Глобальне балансування

Для географічно розподілених центрів обробки даних використовується глобальне балансування навантаження. З методів глобальної балансування можна виділити наступні найбільш поширені методи:

- балансування на рівні DNS,
- балансування на прикладному рівні - це проксінг і redirect запитів,
- балансування на мережевому рівні.

Балансування на рівні DNS. На одне доменне ім'я виділяється кілька IP-адрес. Сервер, на який буде спрямований клієнтський запит, зазвичай визначається за допомогою алгоритму Round Robin. Суть його полягає в наступному: на DNS сервер додається кілька A-записів з різними IP-адресами всіх серверів, і сервер сам буде в циклічному порядку видавати ці адреси. Тобто перший запит отримає перший сервер, другий запит - другий сервер, третій запит - третій сервер, четвертий запит - перший сервер і т.д.

Балансування на прикладному рівні, Full Proxy (алгоритм проксінг). Як балансувальник застосовується т.зв. розумний проксі. Тобто якщо балансувальник отримує запит до ресурсу, він аналізує заголовки прикладного рівня, розуміє, запит до якого ресурсу прийшов на балансувальник, і направляє запит на той чи інший сервер, на якому цей ресурс міститься. При отриманні цього запиту балансувальник може додавати до заголовків HTTP, наприклад, інформацію про те, з якого IP прийшов клієнт, для того, щоб сервер знав, куди його потім відправляти, і з ким він працює. Виконавши запит, сервер передає його назад на балансувальник, той виконує необхідні маніпуляції з новими заголовками або третього рівня, або сьомого рівня і віддає його клієнту.

Redirect запитів. Балансувальник приймає запит від клієнта, відповідає йому перенаправленням на сервер, який містить ресурси. Наприклад, отримуючи запит по HTTP, балансувальник відповідає йому у відповідь - видає помилку 302 move temporary із зазначенням адреси того сервера, на який далі буде ходити наш клієнт.

Балансування на мережевому рівні. Розглядатиметься алгоритм Anycast. Цей алгоритм балансування не вимагає ніякого налаштування з боку клієнта, і суть його полягає в наступному: з різних географічних ділянок анонсується один і той же префікс мережі. Таким чином, кожен запит клієнта маршрутизується на найближчий до нього сервер, який буде його обробляти.

Переваги і недоліки глобальних методів балансування наведені в таблиці 3.2.

Таблиця 3.2. Переваги та недоліки глобальних методів

	DNS	Прикладний рівень		Мережевий рівень
1	2	3	4	5
	Round Robin	Full Proxy	Redirect запитів	Anycast

Переваги	Абсолютна незалежність від протоколу високого рівня	Server-affinity (за певним налаштуванням cookie)	Розподіл запитів по різних серверів за рахунок аналізу запиту	Мінімальні затримки при обробці запитів
	Незалежність від навантаження сервера завдяки кешуючим DNS-серверам	Розподіл різних типів запитів до різних серверів		Доставка трафіку через немагістральні канали (здешевлення)

Продовж. табл.3.2

1	2	3	4	5
	Універсальність (як локальне, так і глобальне балансування)	Можливість аналізувати і модифікувати запити		Розподілом навантаження займається сама мережа
	Дуже низька вартість рішення і швидкий старт	Фільтрація запитів по URL (захист від різних атак)		Висока відмовостійкість
		Самостійно визначають працездатність кожного вузла		Легко додавати і виводити з роботи сервера
Недоліки	Складно відключати сервера (необхідно доп. резервування по типу CARP)	Необхідно балансувати навантаження на самі балансувальники	Досить мала застосуваність до протоколів високого рівня	Можливість пербудовування маршрутів (критично для TCP-сесій)
	Складно розподіляти навантаження в потрібній пропорції	Додаткова точка відмови	Збільшення часу відгуку за рахунок звернення до редиректору	Відсутня можливість проконтролювати, з якого вузла обслуговується користувач (цим керує мережа)
	Обмеженість IP-адрес (тому що кожен сервер повинен мати свій глобальний IP)	Велике споживання ресурсів	Два запита до сервісу на кожен запит клієнта	Дороге обладнання
	Необхідність тримати	Свій проксі для		Інтереси ISP

	подвійний запас серверної потужності	кожного протоколу		
Рішення	Будь-який DNS сервер (наприклад, Named)	HAProxy, nginx	nginx	

3.3. Алгоритми балансування навантаження

Round Robin

Round Robin – метод балансування навантаження, який заснований на алгоритмі кругового обслуговування, який дозволяє відправляти запити «по колу», тобто циклічно. Є певний перелік всіх серверів на основі якого відбувається перебір для відправки запиту.

Після виконання процедури чергового запиту система запам'ятовує, який сервер в даний момент робить обчислення. При цьому, наступний запит на обробку буде відправлений на наступний сервер в списку.

Метод Round Robin зображено на рисунку 3.2.

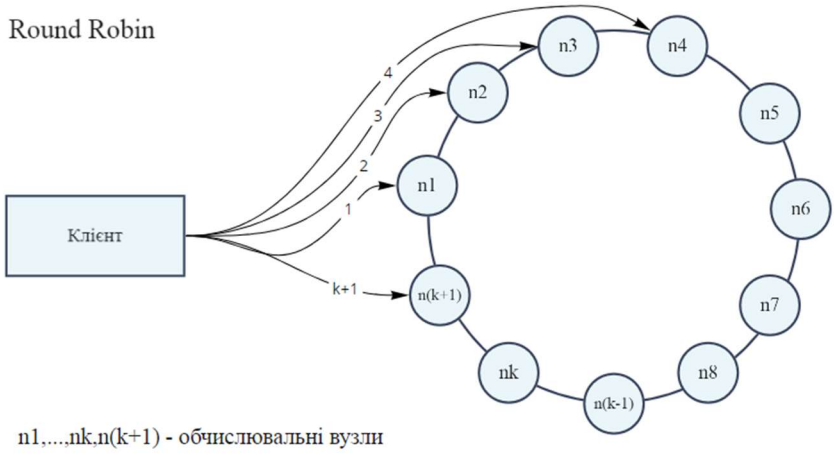

Рис.3.2. Алгоритм балансування Round Robin

Виходячи з рисунка вище, можна переконатися, що перший запит буде відправлений на сервер n1, другий - на сервер n2, третій - на сервер n3 і т.д. виходячи зі списку обчислювачів.

Після того, як останній сервер в списку почав обробку, наступний запит, що надійшов подається знову на початок списку, тобто розподіл серверів щодо запитів, які відправляються, починається спочатку.

Даний алгоритм дозволяє підвищити рівномірність розподілу навантаження між серверами, з огляду на загальну їх кількість. Однак він не враховує характеристики системи, приймаючи те, що сервери мають однакову продуктивністю.

Прикладом застосування алгоритму є Round Robin DNS, який вживає цей метод балансування навантаження для управління відповідями DNS-серверів. Round Robin DNS використовується в серверах компанії Google. Кожен DNS-сервер надає ідентичний сервіс. Однак при відправлення запиту на сервера Google видається список з 11-ти DNS-серверів (рис.3.3).

Рис.3.3. Відправлення запитів на google.com

Принцип роботи заснований на тому, що з кожним запитом змінюється DNS-сервер, який відповідає (який стоїть на вершині списку адрес).

Weighted Round Robin

Weighted Round Robin (Зважений Round Robin). Метод є доопрацьованою версією Round Robin, яка дозволяє враховувати обчислювальну потужність кожного сервера. Таким чином, на сервери з більшою продуктивністю будуть відправлятися запити частіше, ніж на сервери з меншою. Це дозволяє підвищити рівномірність розподілу і ефективність обробки запитів, що надходять.

Метод Weighted Round Robin зображено на рисунку 3.4.

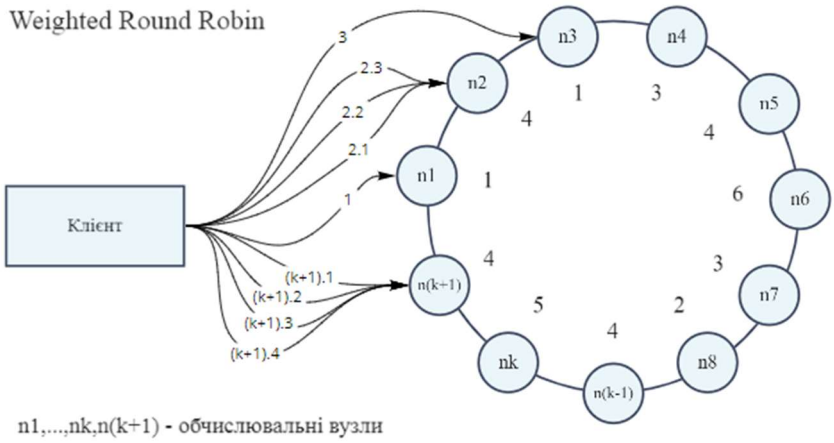

Рис.3.4. Алгоритм балансування навантаження Weighted Round Robin

Сервер n1 має вагу 1, а значить, припустимо, в даному циклі на нього може надійти тільки один запит на обробку, сервер n2 має вагу 3 - в циклі на нього може надійти три запити і т.д.

При проектуванні будь-якого зваженого методу балансування навантаження можна використовувати різні вагові системи - цілі числа; десяткові числа, які має в сумі становити 1.0 (100%) і т.д..

Черга надання серверів на обробку може виглядати наступним чином: n1, n2, n2, n2, n3, n4, n4, n4 і т.д.

Якщо на сервер n1 відправляти за цикл два запити, то він не встигне обробити їх і відправити дані. Другий запит буде очікувати виконання і виконається вже в іншому циклі, коли надійдуть нові два запити. Поступово з кожним новим циклом, чергу на обробку запитів в сервері n1 буде зростати, що може привести до втрати запитів і зниження ефективності обробки всієї системи.

Зниження ефективності також відбудеться і під час вступу на сервер меншої кількості запитів за цикл, ніж даний сервер може обробити. Наприклад, на сервер n6 буде надходити не 6 запитів за цикл, а 3.

Недоліком методу є те, що ваги для серверів встановлюються вручну і при змінах в системі (поліпшення характеристики певного вузла шляхом додавання нових апаратних рішень або різкому несподіване зниження обчислювальної потужності іншого вузла через виниклі апаратні або програмні несправності), кількість відправлених запитів на обробку залишиться незмінною.

Зниження продуктивності одного сервера може привести до зниження продуктивності і швидкості обробки запитів всієї системи, а також втрату запитів, що надходять, які будуть накопичені в черзі несправного обчислювача.

Agent-Based Adaptive

Agent-Based Adaptive - метод балансування навантаження, який дозволяє в реальному часі стежити за навантаженням кожного обчислювального вузла. Для цього кожен обчислювальний вузол має спеціальний агент, який повідомляє про поточне навантаження диспетчер балансування навантаження.

Geo-locating load balancing

Geo-locating load balancing - метод балансування навантаження, який, дозволяє обробляти запити тим обчислювальним вузлам в розподіленої обчислювальної системи, які знаходяться найближче до клієнта, який відправив запит на обробку.

Метод ефективно працює в масштабі континентів і країн, проте має певні складності в визначення місцезнаходження клієнта, що відправив запит в межах країни. Це обумовлено відсутністю точної геобази DNS-адрес, а також тим, що географія міст і місцевостей часто не корелює з тим, яким чином відбувається зв'язок між сусідніми місцевостями, тобто сусідні населені пункти можуть бути пов'язані через великий центр, який знаходиться в тисячах кілометрах від них.

Dynamic Weighted Round Robin

Dynamic Weighted Round Robin є поліпшенням методу Weighted Round Robin (додатково використовується метод Agent-Based Adaptive). Завдяки методу Agent-Based Adaptive вага динамічно присвоюється кожному серверу на основі даних в реальному часі про його поточному навантаженні і пропускної здатності.

При зниженні обчислювальної потужності обчислювального вузла диспетчер балансування навантаження відправляє менше запитів на даний вузол. При збільшенні обчислювальної потужності диспетчер балансування навантаження подасть на нього більше запитів на обробку.

Data Center Aware

Data Center Aware - метод балансування навантаження, який передбачає поділ обчислювальних вузлів на два або більше кластера. Диспетчер балансування навантаження визначає по різним характеристикам системи (кількість оброблюваних запитів в поточний момент, завантаженість кластера, кількість обчислювальних вузлів в кластері, час обробки запитів і т.д. залежно

від методу балансування навантаження) той обчислювальний кластер, куди слід відправити запит на обробку. Кожен обчислювальний кластер може обробляти запити відповідно до свого метода балансування навантаження.

На основі даного методу є можливість розподілити обчислювальні вузли між кластерами таким чином, щоб, наприклад, один обчислювальний кластер обробляв тільки швидкі і легкі запити, а інший - важкі, що вимагають високу продуктивність від обчислювачів. Метод дозволяє розподілити обов'язки між обчислювальними кластерами на основі різних обчислювальних задач і різних функцій певного проекту.

Приклад використання методу балансування Data Center Aware наведено на рисунку 3.5.

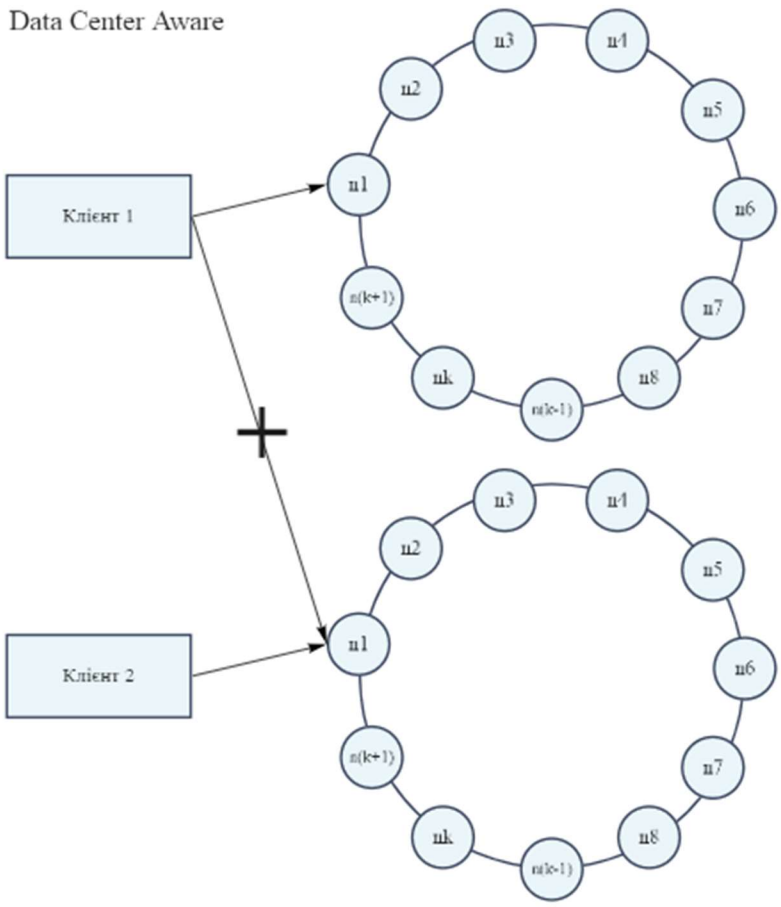

Рис.3.5. Алгоритм балансування Data Center Aware

На рисунку 3.5 можна побачити, що кліент 1 відправляє запити на перший обчислювальний кластер, а кліент 2 - на другий. Причиною може бути те, що перший кластер призначений тільки для складних типів запитів, а другий - тільки для легких. Алгоритм балансування навантаження в кожному обчислювальному кластері може відрізнятися. Однак, при відмові першого обчислювального кластера запити від кліента 1 будуть перенаправлені на другий обчислювальний кластер, що значно знизить швидкість обробки запитів системою, проте підвищена відмовостійкість дозволить системі залишитися в працездатному стані.

Метод ефективно застосовується в високонавантажених веб-проектах, де повна відмова обчислювальної системи, навіть на невеликий відрізок часу, може серйозно вдарити як по іміджу компанії, так і по користувачах, які використовують сервіс.

Token Aware

Token Aware - метод балансування навантаження, що дозволяє порівняти token (маркер), витягнутий з запиту, з маркером який знаходиться в обчислювальному вузлі, що дозволяє відправляти запити на обробку саме на той вузол, якому вони належать. Метод діє як оболонка або фільтр, обгортає інший будь-який алгоритм балансування навантаження, і дозволяє скоротити відстань, яку проходять пакети до потрібного обчислювального вузла.

Алгоритм використовується рішенням від компанії Citrix - NetScaller, яке дозволяє прискорити роботу додатків, а також оптимізувати мережі з доставки даних. NetScaller дозволяє проводити пошук маркера в перших 24-х кілобайтах передачі даних по протоколу HTTP. Для передачі даних по протоколах, відмінним від HTTP, проводиться пошук маркера в перших 16-ти пакетах або поки розмір 16-ти пакетів не перевищить 24 кілобайт. Після вилучення маркера проводиться пошук відповідного обчислювального вузла, який виконає обробку запиту.

Підбір обчислювального вузла для запиту по маркеру є ефективним способом підвищити швидкість обробки даних і збалансувати навантаження на обчислювачі.

Least Connections

Least Connections - метод балансування навантаження, при якому диспетчер балансування аналізує кожен обчислювальний вузол на предмет кількості з'єднань [55]. Чим менше з'єднань у обчислювальному вузлі, тим менше він завантажений.

Недоліком методу є те, що кількість з'єднань не враховує можливості обчислювального вузла. Вузол в системі може володіти низькою продуктивністю і, відповідно, малою кількістю з'єднань. Володіючи малим числом з'єднань, він може бути повністю навантаженим і не в змозі прийняти нові запити на обробку.

Метод в першу чергу призначений для однорідних обчислювальних середовищ.

Weighted Least Connections

Weighted Least Connections - алгоритм, робота якого схожа на роботу метода Least Connections, однак, алгоритм також дозволяє враховувати обчислювальні характеристики вузлів системи, привласнюючи кожному вузлу певну вагу, що аналогічно методу Weighted Round Robin.

Недоліком даного методу є те, що він не дозволяє визначити час, протягом якого підтримується зв'язок, що дозволило б визначити завантаженість обчислювального вузла. Обчислювач може володіти високою продуктивністю, але в даний момент до нього підключена велика кількість легких запитів і він не братиме інші, хоча для обробки легких запитів він може не використовувати всі свої можливості.

І навпаки, високопродуктивний вузол приймає лише кілька важких запитів, але цього достатньо, щоб повністю завантажити його обчислювальні потужності. Алгоритм буде передавати нові запити на обробку саме на цей, повністю навантажений, обчислювач.

Dynamic Weighted Least Connection

Dynamic Weighted Least Connection - метод балансування навантаження поєднує в собі метод Weighted Least Connection і Agent-Based Adaptive. Принцип роботи схожий з методом Dynamic Weighted Round Robin - продуктивність кожного обчислювального вузла, за допомогою спеціального агента, в реальному часі аналізується і відправляється в диспетчер балансування навантаження, який на основі останніх даних розподіляє запити.

Метод значно підвищує ефективність і гнучкість Weighted Least Connection з огляду на те, що алгоритм підлаштовується під будь-які зміни характеристик обчислювачів.

Locality-Based Least Connection Scheduling

Locality-Based Least Connection Scheduling - алгоритм балансування навантаження, заснований на методі Least Connection і призначений для використання в кешуючих проксі-серверах. За кожним клієнтським сервером закріплюється певна кількість клієнтських IP-адрес. Запити відправляються на цей закріплений за клієнтами сервер, якщо він не завантажений повністю. Якщо він завантажений, то вибирається будь-який інший сервер, який завантажений менш ніж наполовину.

Алгоритм вирішує проблему, описану в методі Weighted Least Connections. Тепер диспетчер навантаження не відправлятиме нові запити на обчислювальний вузол з малою кількістю важких запитів, який повністю завантажений.

Locality-Based Least Connection Scheduling with Replication Scheduling

Locality-Based Least Connection Scheduling with Replication Scheduling - метод балансування навантаження, який заснований на тому, що кожна IP-адреса або група адрес закріплюються за одним або групою обчислювальних вузлів. Запити передаються найменш завантаженому обчислювачу. При виникненні ситуації перевантаження всіх обчислювальних вузлів в групі відбувається резервування нового обчислювача за межами групи. Обчислювач додається в групу перевантажених обчислювачів. Одночасно з додаванням обчислювача проводиться перевірка на завантаженість вузлів і видалення найбільш завантаженого обчислювача, що дозволяє уникнути надмірної реплікації.

Приклад використання алгоритму наведено на рисунку 3.6.

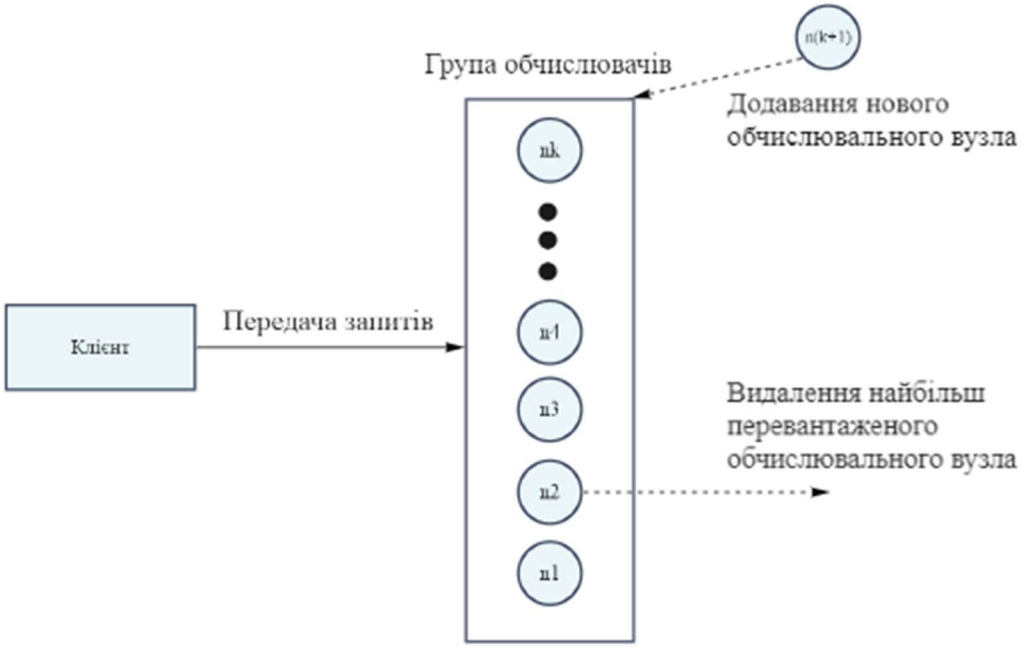

Рис.3.6. Алгоритм балансування Locality-Based Least Connection Scheduling with Replication Scheduling

Least Response Time

Least Response Time - метод балансування навантаження, який оцінює час відповіді на запит. Відповідно, чим менше час відповіді, тим продуктивніше

обчислювальний вузол, тому що обробляє він запити швидше або тим менше з'єднань в даний момент у обчислювального вузла і він менш завантажений зараз.

Диспетчер балансування навантаження вибирає обчислювальний вузол з найменшим часом відповіді і ставить найвищий пріоритет відправлення запитів даного вузла. Запити надходять на даний вузол доти, поки час відповіді не збільшиться, тобто до тих пір, поки вузол не завантажиться.

Недоліком методу є те, що диспетчер балансування не оцінює складність надійшовших запитів. Таким чином, обчислювальний вузол, володіючи невисокою обчислювальною потужністю, з найменшим часом обробив надійшовших найлегший запит на обробку. Диспетчер балансування навантаження на основі даного методу буде вважати поточний обчислювач найпродуктивнішим і відправить на нього один або кілька складних запитів, обчислення яких вузлом займе набагато більше часу, ніж, якби ці запити обробив найпотужніший обчислювальний вузол в системі. Однак найпотужніший обчислювальний вузол в системі в даний момент обробляє складні запити і час відповіді на запит у нього становить далеко не найкращий час. Недоліком також є те, що при відправці однакових за складністю запитів, високопродуктивні обчислювальні вузли прийматимуть всі запити на себе, залишаючи простоювати менш потужні обчислювачі.

З огляду на явний недолік методу Least Response Time його, як і метод Least Connections, ефективно застосовують для однорідних обчислювальних середовищ.

Load-Based

Load-Based - метод балансування навантаження, при якому диспетчер навантаження знімає певні метрики і на основі їх аналізує завантаженість обчислювального вузла.

Приклад методу Load-based наведено на рисунку 3.7.

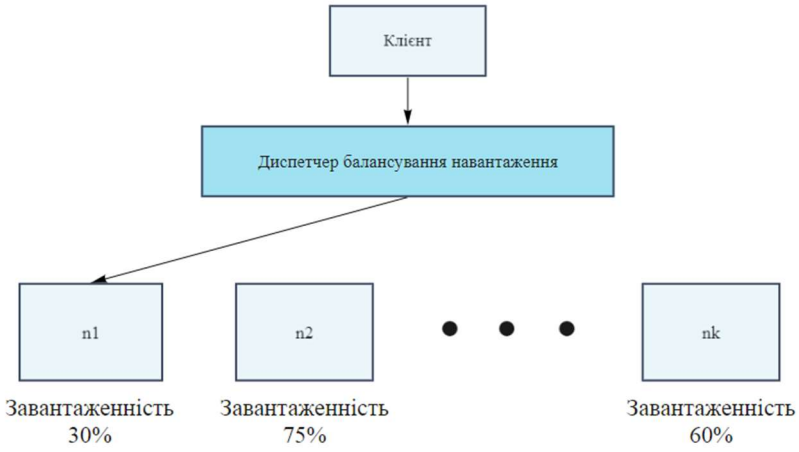

Рис.3.7. Алгоритм навантаження Load-Based

Недоліком є несвоєчасне отримання метрик диспетчером балансування навантаження, яке пов'язане як з каналами зв'язку, так і з тим, що часу на обробку запиту може бути набагато менше, ніж період збору метрик з обчислювальних вузлів. Тобто диспетчер балансування навантаження буде відправляти все нові і нові запити на обробку на обчислювальний вузол, вважаючи, що він практично не навантажений на основі застарілих метричних даних, коли як обчислювальний вузол давно завантажений повністю і запити накопичуються в чергу, що знижує швидкість обробки даних для всієї системи. Інші ж вузли, які були сильно або повністю завантажені на момент зняття метричних показників системи, на поточний момент є вільними і ненавантаженими, проте диспетчер балансування навантаження продовжує не посилати на них запити на обробку, тому що на основі його застарілих метрик дані обчислювальні вузли є навантаженими.

Рішення полягає в підвищенні частоти збору метричних показників системи, проте даний спосіб не завжди є ефективним з огляду на те, що постійним збором метрики засмічується канал зв'язку, а також це вимагає додаткового процесорного часу кожного вузла, що може знизити швидкість обробки запитів у всій системі і лише погіршити проблему ефективного балансування навантаження.

Fastest

Fastest - метод балансування навантаження, який заснований на мінімальному часу відгуку обчислювального вузла. Клієнт відправляє запит на обробку і диспетчер балансування навантаження дивиться час відгуку кожного підключеного обчислювального вузла. Чим менше час відгуку, тим менше завантажено обчислювальний вузол.

Perceptive

Perceptive - метод балансування навантаження, який об'єднує в собі методи Least Response Time і Least Connections.

Метод ефективний в гетерогенних обчислювальних системах, коли вузли між собою значно відрізняються по продуктивності.

У методі першим спрацьовує Least Connections, який дозволяє рівномірно (щодо кількості запитів, що надходять) навантажити як високо-, так і малопродуктивні вузли. У міру збільшення трафіку, потужні обчислювальні вузли будуть обробляти запити швидше малопродуктивних, які можуть «загрузнути» в важких запитах. Далі спрацьовує метод Least Response Time, який дозволяє побачити час обробки запиту обчислювальним вузлом і в першу чергу навантажувати ті вузли, які обробляють запити швидше, поступово розвантажуючи малопродуктивні обчислювачі.

Таким чином, метод дозволяє ліквідувати недоліки кожного з двох методів: для Least Response Time - навантажувати також малопродуктивні обчислювальні вузли, завдяки методу Least Connections; для Least Connections - враховувати обчислювальні можливості кожного вузла на основі часу обробки запитів, завдяки методу Least Response Time.

Observed

Observed - метод балансування навантаження, який засновано на комбінації методів Least Connections і Fastest. Таким чином, обчислювальний вузол для обробки запиту, який надходить, вибирається на підставі комбінації кількості поточних з'єднань і часу відгуку обчислювального вузла.

Метод схожий з методом Perceptive. Поєднання двох алгоритмів дозволяє поліпшити ступінь розподілу запитів на обчислювачах. Постійне сканування часу відгуку дозволяє отримати ступінь завантаженості кожного обчислювача не тільки по характеристиці кількості запитів, що надходять на вузол, тому що складність запиту може значно відрізнятися.

Chained Failover

Chained Failover - метод балансування навантаження, який має сильну схожість з методом Weighted Round Robin тим, що відбувається формування списку обчислювальних вузлів. Однак на відміну від вищезгаданого методу, запити відправляються на обробку не в залежності від ваги кожного обчислювального вузла, а повністю навантажуючи кожен обчислювальний вузол. Якщо обчислювальний вузол не може більше прийняти запитів, то відбувається вибір наступного обчислювального вузла в списку. Таким чином, для даного алгоритму немає необхідності збору інформації про вагу кожного обчислювача.

Source IP Hash

Source IP Hash - метод балансування навантаження, який використовує вихідний і цільовий IP-адреси клієнта і сервера і виконує їх об’єднання, яке необхідне для створення унікального ключа хеш функції. Цей ключ використовується для розміщення запитів клієнта на певному обчислювачі. Оскільки ключ може бути відновлений після розриву зв'язку, то цей метод гарантує те, що запит клієнта буде відправлений саме на той сервер, який був використаний раніше.

Таким чином, кожен клієнт підключається до певного обчислювального вузла, що дозволяє розподілити навантаження між обчислювачами, однак за певних обставин (наприклад, велика кількість важких запитів) ефективність алгоритму знижується.

Software Defined Networking (SDN) Adaptive

Software Defined Networking (SDN) Adaptive - метод балансування навантаження, який проводить збір і аналіз даних про мережу з верхніх і нижніх шарів мережевий моделі OSI. Це дозволяє отримувати інформацію про статус обчислювальних вузлів в системі, про стан виконуваних запитів на них, стан мережевої інфраструктури і рівні завантаженості мережі.

На основі всіх отриманих даних відбувається прийняття рішення про розподіл запитів між вузлами системи.

Least Traffic / Weighted Least Traffic

Least Traffic / Weighted Least Traffic - алгоритм балансування навантаження, в якому диспетчер контролює бітрейт, що виходить із кожного обчислювального вузла системи і відправляє запити на обчислювач з найменшим вихідним трафіком.

Недоліком є те, що диспетчеру балансування навантаження невідома продуктивність сервера. Таким чином, невеликий трафік може виходити з малопродуктивного вузла системи і при його додатковому навантаженні, запити будуть накопичуватися в черзі. Тому, для гетерогенної обчислювальної системи застосовуються зважений алгоритм.

Зважений алгоритм відрізняється від алгоритму Least Traffic лише додаванням ваг обчислювальних вузлів, що було описано в таких алгоритмах, як Weighted Round Robin, Weighted Random і Weighted Least Connections.

Weighted Least Traffic подає запити на обчислювальні вузли з урахуванням, як ваги кожного вузла, так і вихідного трафіку, який дозволяє визначити їх завантаженість.

Представлено різноманітність існуючих алгоритмів, але розглянемо недоліки основних (табл.3.3):

Таблиця 3.3. Недоліки основних алгоритмів

Назва	Опис	Недоліки

Round Robin	Відправляє запити циклічно	Не враховує характеристики системи, приймаючи те, що сервери мають однакову продуктивністю.
Weighted Round Robin	Відправляє запити циклічно, враховує обчислювальну потужність кожного сервера (статично)	Ваги для серверів встановлюються вручну і при змінах в системі (поліпшення характеристики певного вузла шляхом додавання нових апаратних рішень або різкому несподіване зниження обчислювальної потужності іншого вузла через виниклі апаратні або програмні несправності), кількість відправлених запитів на обробку залишиться незмінною.
Geo-locating load balancing	Дозволяє обробляти запити тим обчислювальним вузлам в розподіленої обчислювальної системи, які знаходяться найближче до клієнта, який відправив запит на обробку	Має певні складності в визначення місцезнаходження клієнта, що відправив запит в межах країни.
Least Connections	Диспетчер балансування аналізує кожен обчислювальний вузол на предмет кількості з'єднань	Кількість з'єднань не враховує можливості обчислювального вузла. Вузол в системі може володіти низькою продуктивністю і, відповідно, малою кількістю з'єднань. Володіючи малим числом з'єднань, він може бути повністю навантаженим і не в змозі прийняти нові запити на обробку

Продовж. табл.3.3

1	2	3
Weighted Least Connections	Диспетчер балансування аналізує кожен обчислювальний вузол на предмет кількості з'єднань, враховує обчислювальні характеристики вузлів системи, привласнюючи кожному вузлу певну вагу	Не дозволяє визначити час, протягом якого підтримується зв'язок, що дозволило б визначити завантаженість обчислювального вузла.
Least Response Time	Оцінює час відповіді на запит	Диспетчер балансування не оцінює складність запитів, що надійшли.
Load-Based	Диспетчер навантаження знімає певні метрики і на основі їх аналізує завантаженість обчислювального вузла.	Несвоєчасне отримання метрик диспетчером балансування навантаження, яке пов'язане як з каналами зв'язку, так і з тим, що часу на обробку запиту може бути набагато менше, ніж період збору метрик з обчислювальних вузлів.
Least Traffic	Диспетчер контролює бітрейт, що виходить із кожного обчислювального вузла системи і відправляє	Диспетчеру балансування навантаження невідома продуктивність сервера. Таким чином, невеликий трафік може виходити з малопродуктивного вузла

	запити на обчислювач з найменшим вихідним трафіком	системи і при його додатковому навантаженні, запити будуть накопичуватися в черзі.

3.4. Висновки

Представлений в главі перелік методів та алгоритмів балансування навантаження не є вичерпним, однак, він дозволяє побачити різноманіття різних підходів і методів до розподілу навантаження у ЦОД. Сервіси, механізми, апаратні і програмні технології, методи і алгоритми балансування навантаження дозволили розглянути всю гнучкість при побудові систем і обробці запитів. Балансування навантаження значно розширює можливості і підвищує ефективність центрів обробки даних. Залежно від системи і характеристик обчислювачів, що надходять запитів і вимог до системи, завжди є можливість ефективно збалансувати навантаження, використовуючи описані методи або поєднуючи їх в різні комбінації.

РОЗДІЛ 4. ДИНАМІЧНИЙ АЛГОРИТМ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В МЕРЕЖАХ ЦОД

4.1. Модель динамічного алгоритму балансування навантаження

Аналізуючи недоліки класичних алгоритмів, можна зробити висновок про доцільність розробки динамічного контентозалежного алгоритму з модифікованим зворотним зв'язком. Концепцію пропонованого алгоритму можна представити у вигляді схеми, представленої на рисунку 4.1. Пропонується розробити алгоритм, який буде враховувати тип запиту і динамічний стан сервера. Під час отримання запиту на балансувальник навантаження визначається тип трафіку, і направляється на сервер, який обробляє запити даного класу (типу). При перевантаженні одного з серверів, запит направляється на низькозавантажений сервер, якщо такий існує, або на найменш завантажений.

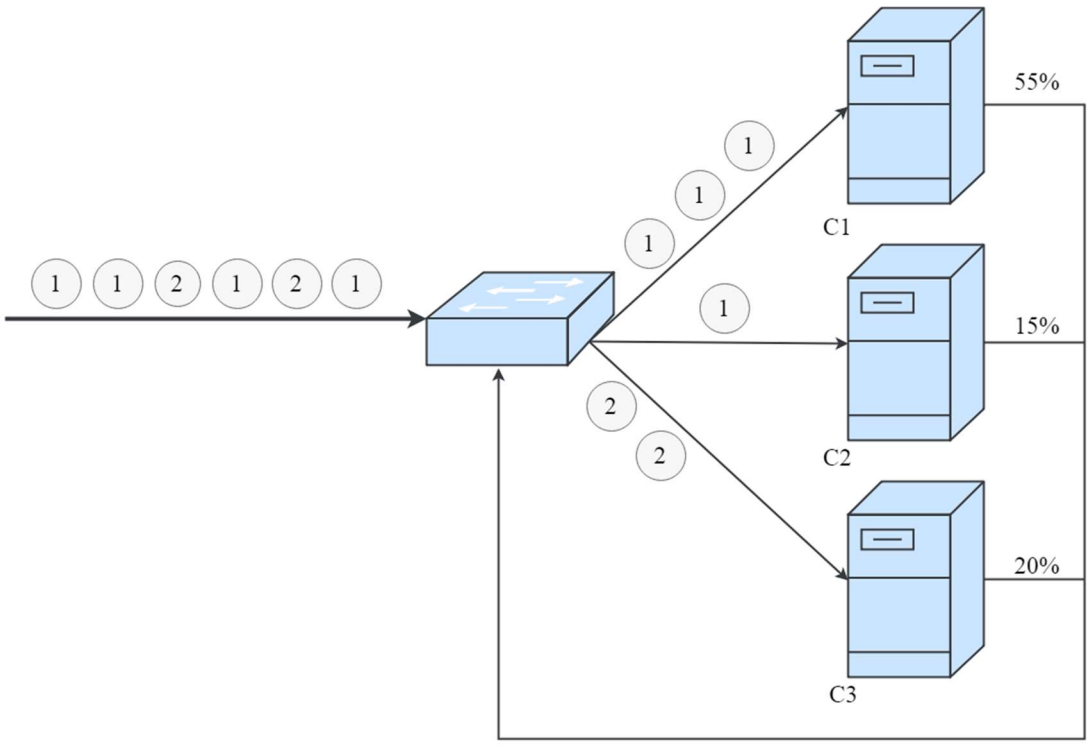

Рис.4.1. Схема розподілу вхідних запитів

Стан і доступність мережевих вузлів (серверів) контролюється шляхом періодичного обміну службовою інформацією між усіма серверами і контролером з додатком систем балансування навантаження. Вже згадана система складається з групи серверів, комутатора OpenFlowSwitch і контролера. Попередньо трафік потрапляє на систему DPI (від англ. Deep packet inspection - технологія накопичення статистичних даних, перевірки і фільтрації мережевих пакетів по їх вмісту), яка ідентифікує тип сесії в рамках з'єднання; комутатор виконує тільки функції передачі даних. Підсистема балансування навантаження і підсистема управління і моніторингу, які тісно взаємодіють один з одним в рамках даної концепції, реалізовані у вигляді відповідних додатків на контролері.

Динамічний алгоритм управління вхідним потоком, який розробляється в роботі, вміє використовувати прогностичні оцінки інтенсивності вхідного потоку запитів кожного класу. Це дозволить запобігти перевантаження і відстежувати тенденції завантаження окремих вузлів. Удосконалення методики короткострокового прогнозу дозволить системі швидше реагувати на флуктуації у вхідному потоці. Для скорочення кількості службової інформації і більш точного відстеження змін у вхідному потоці необхідно розробити адаптивний алгоритм моніторингу.

Алгоритм повинен забезпечувати високі показники продуктивності, пропускної здатності та відмовостійкості (автоматично виявляючи збої вузлів і перерозподіляючи потік даних серед решти) і низький час відгуку.

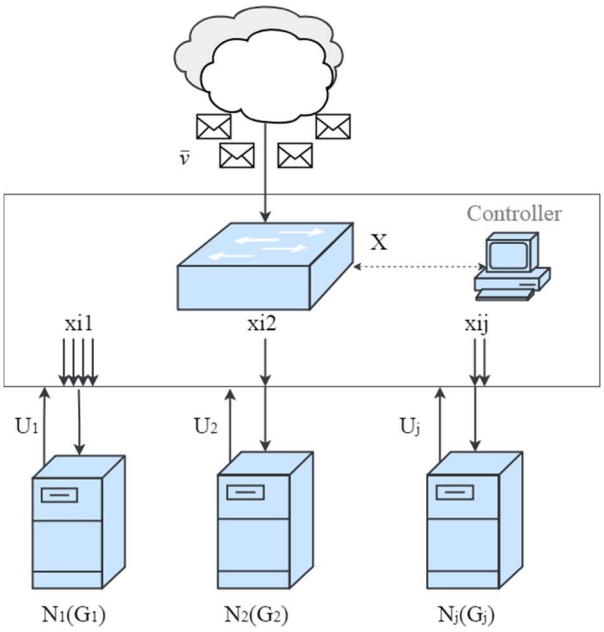

Рис.4.2. Динамічне управління потоком запитів

Математично завдання динамічного управління потоком запитів можна представити наступною залежністю (рис. 4.2):

$$X = f(N, G, \bar{v}), \tag{4.1}$$

де X – матриця розподілу запитів в мережі; N – множина серверів кластера $N = \{N_j\}$; G – множина характеристик серверів $G = \{U_{j\max}, U_j\}$, $U_{j\max}$ – максимальна завантаженість сервера, U_j – поточна завантаженість сервера; $\bar{v}$ – характеристика вхідного потоку запитів.

Телекомунікаційну мережу ЦОД можна передставити у вигляді відкритої системи масового обслуговування з наступною сукупністю характеристик:

1) множиною серверів мережі $\{N_1, N_2, \ldots, N_j\}$, які утворюють кластер, де $j = 1, N$;
2) інтенсивністю вхідного потоку $\lambda_i (i = 1, M)$, де i – клас запиту, $\lambda_i^*(k)$ – прогнозована кількість запитів i-го класу на k-му кроці;
3) матрицею розподілу запитів $X(k) = [x_{ij}]$, $(i = 1, M; j = 1, N)$, де x_{ij} – частка $\lambda_i^*(k)$-х запитів, які обробляються на j-му сервері;

4) законами розподілу часу обслуговування $F_1(t)$, $F_2(t)$,..., $F_j(t)$ і дисциплінами обслуговування запитів на серверах;
5) довжиною черги на серверах Q_j.

Запити описуються вектором:

$$\bar{v} = \left(t, i, j, z_{ij}, m_{resource_i}\right),$$

де t - час надходження запиту до контролера, і - клас запиту, j - номер сервера, який обробляє запит, z_{ij}- завантаженість j-го сервера (створюється і-м класом запитів), $m_{resource\_i}$ - кількість звернень і-го запиту до ресурсу, який обробляє.

Як тільки вхідний потік буде створювати навантаження, які будуть перевищувати максимальну завантаженість сервера, тобто не буде виконуватися умова, то на сервері будуть виникати черги і пов'язані з ними затримки:

$$U_j > U_{jmax}, \tag{4.2}$$

де U_{jmax} – максимальна завантаженість сервера.

З метою впровадження верхньої межі подібних затримок на вузлах мережі загальну буферну ємність обмежують, тобто для кожного сервера визначають поточну і максимальну ємність, позначивши їх q_j і $q_{j\max}$ відповідно. Як тільки $q_j > q_{j\max}$, запити вважають втраченими.

Для забезпечення динамічного розподілу запитів по серверах на k-му кроці визначається матриця розподілу запитів в мережі:

$$X(k) = \left[x_{ij}\right], \qquad (i = 1, M;\ j = 1, N). \tag{4.3}$$

За результатами розрахунку коефіцієнтів матриці X(k) розраховують значення завантаження j-го сервера на k-му кроці – $U_j(k)$:

$$U_j(k) = U_j(k-1) + \sum_{i=1}^{M} \lambda_{ij}^{*}(k) \cdot x_{ij}(k) \cdot z_{ij}. \tag{4.4}$$

Алгоритм управління вхідним потоком повинен розподіляти запити по серверах так, щоб відхилення по завантаженості серверів було мінімальним, тобто:

$$s = \frac{\sum_{j=1}^{N}(\overline{U} - U_j(k))^2}{N} \to min, \tag{4.5}$$

де

$$\begin{cases} U_1(k-1) + \sum_{i=1}^{M} \lambda_i^*(k) \cdot x_{i1}(k) \cdot z_{i1} = U_1(k) \\ U_2(k-1) + \sum_{i=1}^{M} \lambda_i^*(k) \cdot x_{i2}(k) \cdot z_{i2} = U_2(k) \\ \dots \\ U_j(k-1) + \sum_{i=1}^{M} \lambda_i^*(k) \cdot x_{ij}(k) \cdot z_{ij} = U_j(k) \\ \overline{U} = \frac{\sum_{j=1}^{N} U_j(k)}{N} \end{cases}$$

$$\sum_{j=1}^{N} x_{ij}(k) = 1, \qquad i = \overline{(1, M)};$$

$$\sum_{j=1}^{N} \lambda_{ij}^*(k) = \lambda_i^*, \qquad i = \overline{(1, M)};$$

$$x_{ij}(k) > 0;$$

$$z: i \times j, \qquad i = \overline{(1, M)}, \quad j = \overline{(1, N)}.$$

4.2. Алгоритм керування вхідним потоком

Основною ідеєю розробляємого алгоритму є розподіл запитів користувачів на підставі прогнозу вхідного потоку по його класах. При цьому горизонт прогнозу на кожному кроці змінюється в залежності від змін вхідного

потоку. В роботі алгоритму враховується не тільки поточний стан серверів, але і їх продуктивність. Ідею розробляємого алгоритму можна представити у вигляді блок-схеми, наведеної на рисунку 4.3.

Рис. 4.3. Блок-схема алгоритму управління вхідним потоком

Алгоритм виконується циклічно: в результаті оптимізації в кожному циклі визначається матриця розподілу запитів по вузлах мережі. Алгоритм являє собою послідовність наступних кроків.

Крок. 1. Збирається статистична інформація:

а) про інтенсивність вхідного потоку запитів $\lambda(k - 2)$, $\lambda(k - 1)$;

б) про стан серверів (утилізація CPU – U_j; завантаженість j-го сервера, створювана і-м класом запитів – Z_{ij}; середній час обслуговування запитів і-го класу – W_{ij}).

Крок 2. На підставі статистичних даних кроку 1 розраховується коефіцієнт пульсації вхідного потоку запитів – $b_m(k - 1)$, $b_m(k)$.

Примітка. Коефіцієнт пульсації $b_m \in [0; 1]$ – це частка часу, протягом якого миттєва інтенсивність надходження запитів перевищує середню інтенсивність.

Крок 3. Розраховується довжина кроку (далі – інтервал) моніторингу та довжина горизонту прогнозу d(k) за формулою:

$$d(k+1) = \frac{b_m(k-1)}{b_m(k)} \cdot d(k). \qquad (4.6)$$

Крок 4. Прогнозується інтенсивність вхідного потоку для кожного класу запиту λ_i^* на довжину горизонту d(k).

Крок 5. Визначається матриця розподілу запитів по вузлах мережі $\bar{x}(k)$ за формулою (4.3) з урахуванням класу запиту і завантаженості сервера. На підставі отриманих даних прогнозується завантаження серверів на наступному інтервалі.

Крок 6. Запити направляються на сервери відповідно до заданого алгоритму в межах кожного класу запитів.

Крок 7. Здійснюється ревізія прогнозованої кількості запитів $\lambda_i(k)$ – $\lambda_i^*(k)$, які розподіляються згідно з алгоритмом на найменш завантажений

сервер. При цьому переоцінка прогнозу $\lambda_i^*(k)$ – $\lambda_i(k)$ не враховується алгоритмом, як та, що не вносить суттєвих змін в навантаження.

Крок 8. Знімаються дані про завантаженість серверів $U_j(k)$ і передаються на контролер, для розрахунку нового розподілу запитів $\bar{x}(k+1)$.

Для ефективної роботи розробляємого алгоритму необхідно забезпечити моніторинг стану серверів. Основною з існуючих проблем моніторингу є його точність: при спробі досягнення високої точності службова інформація набуває надлишковий характер.

Крок 9. Моніторинг стану серверів, який можна здійснити трьома способами:

1) після кожного запиту, що поступив;
2) в фіксовані проміжки часу, які визначаються статичним алгоритмом;
3) в нефіксовані проміжки часу, які визначаються динамічним алгоритмом.

Інформація, отримана першим способом, є найбільшою за обсягом, т.к. вимірювання проводяться після кожного запиту, що поступив. При другому способі кількість інформації стала, але потрібно визначити інтервал знімання інформації, щоб обсяг інформації не був надмірним і недостатнім. При третьому способі кількість інформації залежить від частоти інтервалів контролю, який повинен пристосовуватися до потоку запитів, що надходять.

На підставі вищесказаного, використання динамічно мінливого інтервалу є найбільш прийнятним з точки зору зменшення надмірності даних. При такому способі частота моніторингу буде залежати від кількості сплесків (пульсацій) вхідного потоку. Наявність окремих сплесків в інтернет-трафіку є однією з його особливостей. Інтервал моніторингу повинен скорочуватися, якщо у вхідному потоці виявлено сплеск.

4.3. Адаптивний алгоритм моніторингу

Для розв'язання задачі оцінки інтервалу моніторингу запропоновано відповідний алгоритм, блок-схема якого приведена на рисунку 4.4.

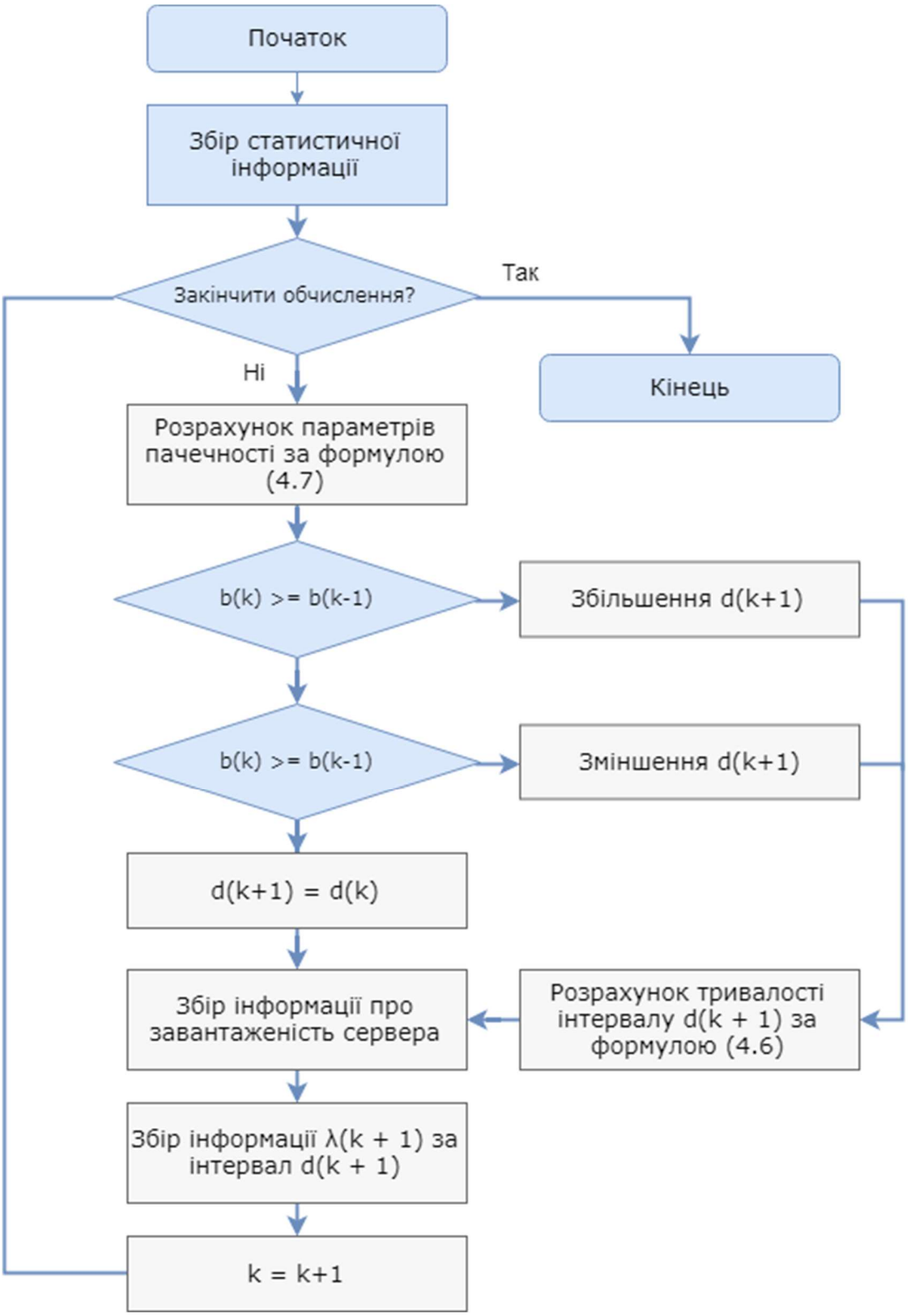

Рис.4.4. Блок-схема алгоритму моніторингу

Алгоритм представляє собою послідовність наступних кроків:

Крок 1. Збирається статистична інформація про інтенсивність вхідного потоку запитів $\lambda(k-2)$, $\lambda(k-1)$ за два попередніх інтервалу і про довжину минулого інтервалу моніторингу $d(k)$.

Крок 2. Визначається коефіцієнт пульсації $b_m(k-1)$, $b_m(k)$, використовуючи формулу:

$$b_m = \frac{g}{n} \cdot \left(1 + \frac{\lambda(k) - \lambda(k-1)}{\lambda(k-1)}\right). \qquad (4.7)$$

Крок 3. Для розрахунку тривалості інтервалу моніторингу порівнюється $b_m(k-1)$ і $b_m(k)$. Якщо $b_m(k) > b_m(k\text{-}1)$, то тривалість інтервалу зменшується, в разі $b_m(k) < b_m(k-1)$ – тривалість інтервалу збільшується в порівнянні з минулою ітерацією.

Крок 4. Розраховується тривалість чергового інтервалу d(k + 1) за формулою (4.6).

Крок 5. Збирається статистична інформація про інтенсивність вхідного потоку запитів $\lambda(k-1)$, $\lambda(k)$ за два попередніх інтервалів і про довжину минулого інтервалу моніторингу $d(k-1)$.

Крок 6. Визначається коефіцієнт пульсації $b_m(k-1)$, $b_m(k)$ за формулою (4.7), використовуючи наступну методику:

1) Підраховуємо кількість http або інших запитів – L, що потрапили в інтервал довжиною d.
2) Інтенсивність надходження запитів в інтервалі визначаємо, як $\lambda = \frac{L}{d}$.
3) Інтервал розбиваємо на n рівних підінтервалів, довжиною $h = \frac{d}{n}$.
4) Визначаємо L(h) – кількість запитів, які потрапили в інтервал h.

5) Знаходимо λ_h – інтенсивність надходження запитів за інтервал довжиною h, як $\lambda_h = \frac{L(h)}{h}$.
6) Обчислюємо L^* – загальну кількість запитів в інтервалах, що задовольняють умові $\lambda_h > \lambda$.
7) Визначаємо g – кількість інтервалів, які відповідають умові $\lambda_h > \lambda$.
8) Розраховуємо коефіцієнт пульсації за формулою (4.7).
9) Так як тривалість кожного слота d змінюється в залежності від значення пульсацій вхідного потоку, то тривалість інтервалу моніторингу d(k +1) визначаємо за формулою (4.6)

Крок 7. Для розрахунку тривалості інтервалу моніторингу порівнюємо $b_m(k-1)$ і $b_m(k)$. Якщо $b_m(k) > b_m(k-1)$ – тривалість інтервалу зменшується, в разі $b_m(k) < b_m(k-1)$ – тривалість інтервалу збільшується в порівнянні з минулою ітерацією.

Крок 8. Розраховується тривалість наступного інтервалу d(k) за формулою (4.6).

Запропонований алгоритм адаптований до відхилень у вхідному потоці, що дозволяє скоротити кількість службової інформації і запобігти перевантаженню за рахунок якості оцінок прогнозування. Горизонт прогнозування на k-ой ітерації залежить від значень пульсацій, тобто, тривалість наступного слота зменшується, якщо було виявлено сплеск.

4.4. Динамічний алгоритм балансування навантаження в мережах ЦОД

Наведені вище алгоритми і методику необхідно об'єднати в єдиний динамічний алгоритм балансування навантаження (з урахуванням специфіки реалізації на базі інфраструктури програмно-обумовленого ЦОД). При цьому варто відзначити, що аналіз стану завантаженості серверів і прийняття рішення

по адресації трафіку лежить на контролері, який повинен бути встановлений в ЦОД і мати інтерфейси моніторингу до всіх активних його серверів.

Алгоритм, блок-схема якого наведена у додатку В, являє собою послідовність наступних кроків:

Крок 1. Клієнт, знаючи IP-адресу одного з серверів (наприклад, IP1), звертається до нього.

Крок 2. На комутатор, який граничить з серверами, приходить пакет від клієнта.

Крок 3. Комутатор з'ясовує, що клієнт звертається до одного з серверів вперше, тобто сесії з сервером у нього ще немає.

Крок 4. Комутатор передає пакет клієнта на контролер.

Крок 5. Контролер бачить, що IP-адреса призначення – один з балансуємих серверів, тому передає обробку пакета додатку по балансуванню навантаження.

Крок 6. Додаток моніторингу проводить збір і аналіз статистичної інформації: про завантаженість j-го сервера, створювана i-м класом запитів – Z_{ij}; про довжину минулого інтервалу моніторингу $d(k)$; інтенсивність вхідного потоку запитів $\lambda(k-2)$, $\lambda(k-1)$, про стан серверів (утилізація CPU – U_j; середній час обслуговування запитів i-го класу – W_{ij}) за декілька ітерацій.

Крок 7. Обчислюється коефіцієнт пульсацій вхідного потоку запитів $b_m(k-2$, $b_m(k-1)$ за формулою (4.7) на підставі статистичних даних, які були зібрані на кроці 6.

Крок 8. Розраховується інтервал моніторингу завантаженості серверів і горизонт прогнозу d(k) за формулою (4.6).

Крок 9. Прогнозується інтенсивність вхідного потоку по кожному типу запитів λ_i^* на довжину горизонту d(k), яка визначається на кроці 8.

Рис.4.5. Блок-схема динамічного алгоритму балансування навантаження

Крок 10. Додаток балансування проводить розрахунок матриці розподілу запитів по вузлах мережі $x(k)$ з урахуванням класу запиту і завантаженості сервера. На підставі отриманих даних прогнозується завантаження серверів на наступній ітерації.

Крок 11. Здійснюється розподіл запитів по серверах згідно з алгоритмом в межах кожного класу.

Крок 12. Здійснюється ревізія прогнозованої кількості запитів $\lambda_i(k)$ – $\lambda_i^*(k)$, які розподіляються згідно з алгоритмом балансування на найменш завантажений сервер.

Крок 13. Додаток визначає, на який з серверів передати обробку запиту клієнта (наприклад, IP2).

Крок 14. На граничному комутаторі задається правило, що якщо від клієнта приходить пакет з адресою призначення IP1, то адреса призначення підміняється на IP2, і пакет передається на порт, відповідний сервера IP2.

Крок 15. Також на комутаторі задається правило, що якщо від сервера IP2 приходить пакет з адресою призначення клієнта, то адреса джерела підміняється на IP1 і передається на порт клієнта.

Крок 16. У додатку оновлюються значення лічильників сесій серверів для збереження актуальності завантаженості серверів $U_j(k)$ і розрахунку нового розподілу запитів $\bar{x}(k+1)$.

Крок 17. Якщо на граничний комутатор приходить пакет про закриття з'єднання клієнтом або сервером, то він також передається контролеру, який передає управління додатку по балансуванню навантаження, яке декрементує значення лічильника активованих сесій на сервері.

Алгоритм балансування навантаження повинен розподіляти запити по серверам так, щоб відхилення завантаженості серверів від середнього значення було мінімальним, тобто:

$$s = \frac{\sum_{j=1}^{N}(\overline{U} - U_j(k))^2}{N} \to min, \qquad (4.8)$$

Алгоритм обчислює завантаженість j-го сервера на k-му кроці, грунтуючись на сумі його завантаженості до цього моменту $U_j(k$ - 1) і навантаженні, створюваної часткою x_{ij}(k) прогнозованої інтенсивності потоку λ_i^*(k) і-го класу, яка буде оброблятися на даному сервері.

У якості шуканої матриці виступає матриця розподілу запитів:

$$X(k) = [x_{ij}], \qquad (i = 1, M;\ j = 1, N).$$

В результаті розрахунку забезпечується динамічний розподіл навантаження по серверах в кожному циклі.

4.5. Висновки

В даному розділі була наведена модель динамічного алгоритму балансування навантаження та розроблено алгоритми керування вхідним потоком, адаптивний алгоритм моніторингу та динамічний алгоритм балансування навантаження в мережах ЦОД.

Розроблений динамічний алгоритм балансування навантаження в мережах ЦОД відповідає вимогам до балансуванню і адаптований до змін інтенсивності вхідного потоку для зменшення часу відповіді, ймовірності втрат і підвищення продуктивності системи.

РОЗДІЛ 5. ОХОРОНА ПРАЦІ

5.1. Аналіз умов праці

Основний робочий процес при написанні диплома – це робота на персональному комп'ютері. Тривала робота за комп'ютером може мати шкідливий вплив на організм людини і привести до небезпечних захворювань.

Серед шкідливих факторів виділяються три основні групи, здатні вплинути на здоров'я людини:

- санітарно-гігієнічні фактори;
- ергономічні чинники;
- психофізіологічні чинники.

Аналіз умов праці допоможе визначити, які заходи необхідно провести для того, щоб привести умови праці до нормативних, відповідають закону про безпеку.

Санітарно-гігієнічні фактори

До санітарно-гігієнічних факторів відносяться всі елементи виробничого середовища, в якій протікає трудовий процес: мікроклімат, освітлення, Електронебезпека, шум, вібрація і електромагнітні випромінювання (ЕМВ).

Мікроклімат

Мікроклімат - кліматичні умови, створені в обмеженому просторі штучно або обумовлені природними особливостями.

Основні параметри мікроклімату: температура, відносна вологість, швидкість повітря.

Норми виробничого мікроклімату встановлені системою стандартів безпеки праці ГОСТ 12.1.005-88 «Загальні санітарно-гігієнічні вимоги до

повітря робочої зони» і Санітарними правилами і нормами СанПіН 2.2.4.548-96 «Гігієнічні вимоги до мікроклімату виробничих приміщень». Параметри мікроклімату робочого приміщення повинні відповідати вищевказаним нормам в рамках холодної пори року. Робота над дипломним проектом відноситься до легких робіт (категорія Іа), так як виконується сидячи і вимагає невеликої кількості енергії: 75 ккал / год. Інформація про те, чи відповідає робоче приміщення зазначеним нормам, наведена в таблиці 5.1.

Таблиця 5.1. Оптимальні, допустимі і фактичні значення параметрів мікроклімату

Показник	Оптимальна величина	Допустима величина		Фактичне значення
Температура повітря, °С	22-24	> опт.вел	≤ опт.вел	24
		20,0-21,9	24,1-25	
Вологість, %	40-60	15-75		54
Швидкість руху повітря, м/с	0,1	0,1		0,1

Згідно з показниками таблиці 1 можна сказати, що робоче приміщення повністю задовольняє оптимальним умовам мікроклімату.

Освітлення

Робота з комп'ютером супроводжується тривалими зоровими навантаженнями і негативно позначається на здоров'я очей. Правильно виконане освітлення робочого місця робить позитивний психофізіологічний вплив на людину, сприяє підвищенню ефективності і високої працездатності. Досягнення оптимальних умов роботи досягається шляхом забезпечення природного освітлення в світлий час доби і сприятливого штучного освітлення в темний час доби.

Для того щоб забезпечити умови, необхідні для зорового комфорту, в системі освітлення повинні бути реалізовані наступні попередні вимоги:

– рівномірне освітлення;
– оптимальна яскравість;
– відсутність відблисків;
– відповідний контраст;
– правильна колірна гамма;
– відсутність стробоскопічного ефекту або пульсації світла.

Відповідно до СНіП 23-05-95 робота над дипломним проектом відноситься до III розряду зорових робіт (мінімальний розмір об'єкта розрізнення-товщина штриха букви - 0.3 мм, звідси розряд зорової роботи - робота високої точності) при великому контрасті і світлому тлі (подрозряд зорової роботи «г»).

Робоче місце висвітлюється в світлий час доби через вікно (природне бічне освітлення), яке виходить на південну сторону, і сонячний світло не перегороджує сторонніми об'єктами. У темний час доби штучне освітлення забезпечується світильником з п'ятьма лампами розжарювання потужністю 40Вт, що не забезпечує необхідну освітленість для даного типу зорових робіт в 300 лк. Часта переадаптація очей до різної яскравості і відстаней є одним з головних негативних чинників при роботі з дисплеями. Несприятливим фактором світлового середовища є невідповідність нормативним значенням рівнів освітленості робочих поверхонь столу, екрану, клавіатури. Нерідко на екранах спостерігається дзеркальне відображення джерел світла та оточуючих предметів. Все вищевикладене утрудняє роботу і призводить до порушень основних функцій зорової системи.

Електронебезпека

Згідно з правилами улаштування електроустановок існує три класи приміщень, що розрізняються за ступенем ризику ураження електричним струмом: приміщення без підвищеної небезпеки, приміщення з підвищеною небезпекою і приміщення особливо небезпечні.

Робоче приміщення, в якому пишеться дипломна робота, відноситься до приміщень без підвищеної небезпеки, так як воно не сире (вологість повітря не перевищує 75%), температура в ньому не перевищує +35 °С (середнє значення +24 °С), і регулярно проводиться прибирання приміщення, що не дозволяє утворюватися струмопровідному пилу.

Персональний комп'ютер захищений від перепадів електроенергії запобіжником. В даному приміщенні проведена однофазова електрична мережа з ізольованою нейтраллю. Робоча напруга в мережі 220 В. Провід ізольовані і розташовані таким чином, що ймовірність випадкового контакту людини з проводами значно знижена.

Шум

Шум визначають як сукупність аперіодичних звуків різної інтенсивності і частоти. Шуми різняться за різними параметрами, бувають такі, як:

- низько-, середньо-, і високочастотні;
- постійні і непостійні;
- тривалі і короткочасні.

Велике значення надається амплітудно-тимчасовим, спектральним і імовірнісним параметрами непостійних шумів, які характерні для сучасного виробництва. Інтенсивний шум сприяє зниженню працездатності, знижує концентрацію і швидкість роботи, є причиною накопичення втоми.

У робочому приміщенні джерелами шуму є електричні прилади, а саме персональний комп'ютер і його периферійні пристрої. Згідно з нормами шуму ГОСТ 12.1.003-83 написання дипломної роботи відноситься до наступної категорії: "Творча діяльність, керівна робота з підвищеними вимогами, наукова діяльність, конструювання та проектування, програмування, викладання і навчання, лікарська діяльність: робочі місця в приміщеннях дирекції, проектно-конструкторських бюро; розраховувачів, програмістів обчислювальних машин,

в лабораторіях для теоретичних робіт і обробки даних, прийому хворих у медпунктах". Згідно нижченаведеної таблиці 5.2 з ГОСТ 12.1.003-83 "Шум і загальні вимоги безпеки» допустимий рівень звуку для такого типу приміщень – 50 дБА, а нашому випадку рівень звуку 30 дБА, отже, можна зробити висновок, що рівень шуму знаходиться в нормі.

Таблиця 5.2. Шум і загальні вимоги безпеки

Вид трудової діяльності, робочі місця	Рівні звукового тиску, дБ, в складових смугах з середньогеометричними частотами, Гц									Рівні звуку і еквівалентні рівні звуку, дБА
	31,5	63	125	250	500	1000	2000	4000	8000	
Підприємства, установи та організації										
Творча діяльність, керівна робота з підвищеними вимогами, наукова діяльність, конструювання та проектування, програмування, викладання і навчання, лікарська діяльність: робочі місця в приміщеннях - дирекції, проектно-конструкторських бюро; розраховувачів, програмістів обчислювальних машин, в лабораторіях для	86	71	61	54	49	45	42	40	38	50

теоретичних робіт і обробки даних, прийому хворих у медпунктах										

Вібрація

Вібрація – малі механічні коливання, що виникають в тілах під впливом фізичного поля. Сильна вібрація негативно позначається на здоров'ї людини, і слід обладнати своє робоче місце таким чином, щоб уникнути її впливу. В іншому випадку при впливі вібрації погіршується зір, координація, робота внутрішніх органів.

Норми вібраційної безпеки описані в наступних документах: ГОСТ 12.1.012-90 «Вібраційна безпека» і СН 2.2.4 / 2.1.8.566-96 «Виробнича вібрація, вібрація в приміщеннях житлових і громадських будівель». Основними нормованими параметрами вібрації є середні квадратичні величини рівнів віброшвидкості і віброприскорення в октавних смугах частот. Загальна вібрація, категорія 3, тип «в» (вібрація на робочих місцях працівників розумової праці і персоналу, який не займається фізичною працею).

Електромагнітні випромінювання

Комп'ютер, як і всі прилади, що споживають електроенергію, випускає електромагнітне випромінювання, яке має більший вплив зі зменшенням відстані від джерела до об'єкта. Вважається, що електромагнітне випромінювання сприяє розладу нервової системи, зниження імунітету і негативно впливає на серцево-судинну систему.

Розрізняють чотири види опромінення:

- професійне;
- непрофесійне;
- опромінення в побуті;

– опромінення в лікувальних цілях.

Ступінь впливу ЕМІ визначається частотою випромінювання, інтенсивністю і тривалістю впливу, а також розміром і положенням опромінюваної поверхні тіла, режиму опромінення і т.д. Допустимі тимчасові рівні електромагнітних полів нормуються в додатку 2 до СанПіН 2.2.2 / 2.4.1340-03 «Гігієнічні вимоги до персональних електронно-обчислювальних машин та організації роботи». Таблиця 2 показує допустимі і фактичні значення тимчасових рівнів електромагнітних полів, що створюються персональним комп'ютером.

Таблиця 5.3. Фактичні і допустимі значення тимчасових рівнів електромагнітних полів

Параметри		Допустимі значення	Фактичні значення
Напруженість електричного поля	в діапазоні частот 5 Гц – 2 кГц	25 В/м	7 В/м
	в діапазоні частот 2 кГц – 400 кГц	2,5 В/м	0,9 В/м
Щільність магнітного потоку	в діапазоні частот 5 Гц – 2 кГц	250 нТл	90 нТл
	в діапазоні частот 2 кГц – 400 кГц	25 нТл	9 нТл
Напряженность электростатического поля		15 кВ/м	6 кВ/м

Дані таблиці показують, що робоче приміщення відповідає нормам СанПіН 2.2.2 / 2.4.1340-03, за рахунок того, що були виконані наступні заходи при роботі:

– використовувався рідкокристалічний монітор, оскільки його випромінювання значно менше, ніж у ЕПТ моніторів (монітор з електронно-променевою трубкою);
– комп'ютер не залишався включеним на тривалий час;
– монітор розташований в кутку, так що випускається їм випромінювання частково поглиналося стінами;
– по можливості сеанси роботи за комп'ютером були не дуже тривалими;

– дотримувалися рекомендації Гост Р 52324-2005 «Ергономічні вимоги до роботи з візуальними дисплеями, заснованими на плоских панелях».

5.2. Ергономіка робочого місця

Правильна організація робочого місця може мати значний вплив на продуктивність праці і концентрацію і дозволити більш ефективно вирішувати поставлені завдання.

Організація робочого місце обумовлюється рекомендаціями СанПіН 2.2.2 / 2.4.1340-03, "Гігієнічні вимоги до персональних електронно-обчислювальним машинам і організації роботи". Основну увагу приділено ергономічним характеристикам робочого стола і крісла.

а) Вимоги до приміщень для роботи з ПЕОМ. З урахуванням техніки безпеки при поводженні з ПЕОМ приміщення повинні бути просторими, добре провітрюваних і освітленими (таблиця 5.4).

Таблиця 5.4. Вимоги до приміщень для роботи з ПЕОМ

Параметр	Фактичні значення	Необхідні значення
Відстань від екрана до очей користувача, мм	620	600-700, але не менше 500
Площа на одне робоче місце (при використанні РК-моніторів), $м^2$	12	4,5
Обсяг повітря на одне робоче місце, $м^3$	33	19,5

Всі вимоги для комфортної та безпечної роботи над дипломним проектом виконані.

б) Вимоги до робочого столу (таблиця 4). Конструкція робочого столу повинна відповідати сучасним вимогам ергономіки і забезпечувати оптимальне розміщення на його робочої поверхності обладнання з урахуванням його

кількості і конструктивних особливостей (розмірів системного блоку, монітора, клавіатури та ін.) і виходячи з характеру виконуваної роботи.

Таблиця 5.5. Вимоги до робочого столу

Параметр	Фактичні значення	Необхідні значення
Ширина столу, мм	1350	800-1400
Глибина столу, мм	870	800-1000
Висота столу, мм	730	725
Висота простору для ніг, мм	695	не менше 600
Ширина простору для ніг, мм	900	не менше 500
Глибина простору для ніг на рівні колін, мм	750	не менше 450
Глибина простору для ніг на рівні витягнутих ніг, мм	970	не менше 650

Ергономічні характеристики робочого столу відповідають вимогам.

в) Вимоги до робочого стільця.

Тривале перебування в сидячому положенні приводить до негативних наслідків для організму. З огляду на особливості конструкції робочого стільця можна усунути більшість негативних наслідків. Конструкція робочого стільця (крісла) повинна забезпечувати підтримку раціональної робочої пози під час роботи на ПЕОМ, дозволяти змінювати позу з метою зниження статичного напруження м'язів шийно-плечової області і спини для попередження розвитку втоми. Тип робочого стільця (крісла) слід вибирати з урахуванням зростання користувача, характеру і тривалості роботи з ПЕОМ. Робочий стілець (крісло) повинен бути підйомно-поворотним, регульованим по висоті і кутам нахилу сидіння і спинки, а також відстані спинки від переднього краю сидіння, при цьому регулювання кожного параметра повинна бути незалежною, легко здійснюваної плюс надійну фіксацію.

5.3. Розрахунок штучного освітлення

Розрахунок освітленості робочого місця зводиться до вибору системи освітлення, визначенню необхідного світлового потоку.

Для визначення світлового потоку будемо використовувати формулу:

$$\Phi_л = \frac{E \cdot S \cdot К_з}{U \cdot n \cdot N} \quad (5.1)$$

де

E – необхідна горизонтальна освітленість, лк;

S – площа приміщення, м2;

$К_з$ – коефіцієнт запасу;

U – коефіцієнт використання;

n – кількість ламп в світильнику;

N – кількість світильників;

$Ф_л$ –світловий потік однієї лампи, лм

Згідно з гігієнічними вимогами, мінімальна освітленість повинна становити не менше 300 лк.

Площа кімнати S складає 15м^2.

Коефіцієнт запасу залежить від ступеня забруднення приміщення, частоти технічного обслуговування світильника, інтенсивності експлуатації світильників і приймає значення від 1,2 до 2. У іноземних нормах використовується коефіцієнт експлуатації maintenance factor (MF), зворотний коефіцієнту запасу MF = 1/Кз.

В даному випадку $К_з$ = 1,2.

Для визначення коефіцієнта використання попередньо визначимо індекс приміщення:

$$\varphi = \frac{S}{(h_1 - h_2)(a + b)},$$

де

S – площа приміщення, м2;

a – довжина приміщення, м;

b – ширина приміщення, м;

h_1 – висота, на якій знаходиться світильник (розташований на відстані 20 см від стелі), м;

h_2 – висота розрахункової поверхні, м.

В даному випадку

$$\varphi = \frac{15}{(2.55 - 0.8)(3.5 + 4.3)} \approx 1.0989$$

Також згідно з таблицею з коефіцієнтами відбиття маємо:

- коэффициент отражения потолка – 0.7;
- коэффициент отражения стен – 0.5;
- коэффициент отражения пола – 0.3.

Далі, знаючи індекс приміщення, а також коефіцієнти відображення стелі, стін і підлоги, за допомогою таблиць знаходимо коефіцієнт використання (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп; залежить від характеристик світильника, розмірів приміщення, фарбування стін і стелі).

В даному випадку коефіцієнт використання дорівнює 0,52.

Кількість ламп в світильнику дорівнює чотирьом.

Таким чином, отримуємо

$$Ф_л = \frac{300 \cdot 12 \cdot 1,2}{0,52 \cdot 4 \cdot 1} \approx 2596 \text{ лм}$$

Але в даний час є лампи з $Ф_л$ = 420 лм. Звідси можна зробити висновок про те, що потрібні інші лампи з великим світловим потоком. Так як неможливо знайти лампи, що використовуються на поточний момент для

освітлення, зі світловим потоком більш 2596 лм, то слід доповніті світильник і, наприклад, використовувати люмінесцентній світильник D-TEC Denta Hybrid 24 зі світловім потоком 3400 лм.

Слід зазначити, що для правильного висвітлення крім використання правильної системи освітлення необхідно розміщуваті комп'ютер так, щоб світло (природне або штучне) падало збоку зліва.

5.4. Висновки

В даному розділі були розглянуті шкідливі фактори, що впливають на користувача комп'ютера, вимоги до організації робочого місця і рівню освітленості.

В ході роботи було з'ясовано, що за багатьма параметрами наявне робоче місце не підходить для тривалої роботи з комп'ютером: недостатня освітленість, погана ергономічність робочого місця. Також було з'ясовано, що слід доповнити світильник і використовувати, наприклад, люмінесцентний світильник D-TEC Denta Hybrid 24.

Однак при дотриманні зазначених норм і рекомендацій можна забезпечити безпечні умови праці при роботі з ПК, при яких рівні перерахованих шкідливих впливів зводяться до мінімуму. Це дозволяє зберегти здоров'я і високу працездатність при регулярній тривалій роботі з ПК.

ВИСНОВКИ

Центр обробки даних – це відмовостійка комплексна централізована система, яка здійснює функції зберігання, обробки та розповсюдженні інформації. При проектуванні і створенні хмарних рішень важливе місце займає забезпечення надійності та оптимальності використання ресурсів - моніторинг, або отримання інформації про доступність і завантаженність апаратних ресурсів платформ з копіями розподіленого додатка, і балансування, тобто розподіл користувацьких запитів, що надходять, між наявними апаратними платформами.

У другому розділі були розглянуті основні стандарти центрів обробки даних, етапи будування, розібравши які було виявлено, що впровадження балансування навантаження відбувається на етапі реалізації, та має бути взято до уваги на етапі модернізації. Також були наведені вимоги по експлуатації ЦОД. Для побудови дата-центру необхідно пройти всі розглянуті етапи будівництва, а також повинні бути у наявності всі наведені критерії.

У третьому розділі представлено перелік методів та алгоритмів балансування, який дозволяє побачити різноманіття різних підходів і методів до розподілу навантаження у ЦОД. Сервіси, механізми, апаратні і програмні технології, методи і алгоритми балансування навантаження дозволили розглянути всю гнучкість при побудові систем і обробці запитів. Балансування навантаження значно розширює можливості і підвищує ефективність центрів обробки даних. Залежно від системи і характеристик обчислювачів, що надходять запитів і вимог до системи, завжди є можливість ефективно збалансувати навантаження, використовуючи описані методи або поєднуючи їх в різні комбінації. Також були наведені недоліки основних алгоритмів, для того, щоб розуміти, що слід враховувати при розробці власного алгоритму.

В четвертому розділі була наведена модель динамічного алгоритму балансування навантаження та розроблено алгоритми керування вхідним потоком, адаптивний алгоритм моніторингу та динамічний алгоритм балансування навантаження в мережах ЦОД.

Розроблений динамічний алгоритм балансування навантаження в мережах ЦОД відповідає вимогам до балансуванню і адаптований до змін інтенсивності вхідного потоку для зменшення часу відповіді, ймовірності втрат і підвищення продуктивності системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Харатишвили Д. Дата-центры в цифрах и фактах / Давид Харатишвили. // Компьютер пресс. – 2009. – №8.
2. Что такое ЦОД? [Електронний ресурс] // Ксайберри. – 2015. – Режим доступу до ресурсу: http://www.xyberry.com/customer-area/knowledgebase.php?action=displayarticle&id=73.
3. Кириллов И. "Облака" в Украине: как меняется рынок / Игорь Кириллов. // Сети и бизнес. – 2015. – №5. – С. 42–52.
4. Кириллов И. Облачные услуги: как заработать больше / Игорь Кириллов. // Сети и бизнес. – 2015. – №2. – С. 56–59.
5. Йоси бен Харош. 5 тенденций развития ЦОД в 2016 году [Електронний ресурс] / Йоси бен Харош // Мир ЦОД. – 2016. – Режим доступу до ресурсу: http://www.osp.ru/dcworld/2016/01/13048167.html.
6. Кириллов И. Украинский рынок коммерческих ЦОД: в поисках новых возможностей / Игорь Кириллов. // Сети и бизнес. – 2015. – №3.
7. Архитектура ЭВМ [Електронний ресурс] – Режим доступу до ресурсу: https://sites.google.com/site/architectevm/#TOC--68.
8. Услуги ЦОД (дата-центров) [Електронний ресурс] // РБК.Research. – 2014. – Режим доступу до ресурсу: http://marketing.rbc.ru/reviews/it-business/chapter_6_1.shtml.
9. Центр обработки данных (дата-центр) [Електронний ресурс] // Tadviser. – 2014. – Режим доступу до ресурсу: http://www.tadviser.ru/index.php/Статья:Центр_обработки_данных_(дата-центр)#
10. Облачные технологии [Електронний ресурс] // РБК.Research – Режим доступу до ресурсу: http://marketing.rbc.ru/reviews/it-business/chapter_6_2.shtml.

11. С. С. Баранова Исследования тенденций развития облачных сервисов // Cloud of science . 2014. №3 С.517-523.
12. Монахов Д. Н., Монахов Н. В.Облачные Технологии. Теория и практика/МАКС Пресс Москва, МГУ,2013–128с.
13. Балансировка нагрузки в распределенных системах. [Електронний ресурс] / Интуит // – Режим доступу до ресурсу: http://www.intuit.ru/studies/courses/1146/238/lecture/6153?page=2
14. Распределенные облачные дата-центры. Главы из книги. Глава 1. [Електронний ресурс] // Телеком и ИТ – Режим доступу до ресурсу: https://shalaginov.com/2014/11/30/распределенные-облачные-дата-центры/
15. Балансировка в облаках. [Електронний ресурс] – Режим доступу: http://www.osp.ru/os/2011/09/13011562/
16. Балансировка нагрузки: основные алгоритмы и методы. [Електронний ресурс] – Режим доступу: http://habrahabr.ru/company/selectel/blog/250201/
17. Load Balancing - Compute Engine - Google Cloud Platform. [Електронний ресурс] – Режим доступу: https://cloud.google.com/compute/docs/load-balancing/
18. Cardellini V., Colajanni M., Yu P. S. Dynamic load balancing on web-server systems //IEEE Internet computing. – 1999. – Т. 3. – №. 3. – С. 28-39.
19. Sharma S., Singh S., Sharma M. Performance analysis of load balancing algorithms //World Academy of Science, Engineering and Technology. – 2008. – Т. 38. – №. 3. – С. 269-272.
20. Как балансировать на «сетевом» канате под куполом тяжелой нагрузки? / Сергей Зубов (CDNvideo) [Електронний ресурс] – Режим доступу: https://www.slideshare.net/profyclub_ru/cdnvideo-57577239
21. Ю.Б. Айзенберг "Справочная книга по светотехнике" М.: Энергоатомиздам, 1983.

22. Системы балансировки нагрузки Web-серверов. – Режим доступу: http://citforum.ru/internet/webservers/websbal.shtml
23. DNS балансировка. – Режим доступу: http://ruhighload.com/post/DNS+балансировака+
24. Балансировка масштабируемых приложений. – Режим доступу: http://www.osp.ru/os/2012/08/13019244/
25. LoadMaster For Azure. – Режим доступу: http://kemptechnologies.com/solutions/microsoft-load-balancing/loadmaster-azure/
26. AWS | Elastic Load Balancing– сетевой балансировщик нагрузки в облаке. – Режим доступу: http://aws.amazon.com/ru/elasticloadbalancing/
27. LVS Introduction - Load Balancing Server Cluster. – Режим доступу: http://www.linuxvirtualserver.org/whatis.html
28. Load Balancing - High Availability and Performance - NGINX. – Режим доступу: http://nginx.com/solutions/load-balancing/
29. CloudStack: Configure Load Balancer. – Режим доступу: https://support.getcloudservices.com/entries/22002577-CloudStack-Configure-LoadBalancer
30. Managing Networks and Traffic - Apache CloudStack Administration Documentation 4.5.0 documentation. – Режим доступу: docs.cloudstack.apache.org/projects/cloudstackadministration/en/4.5/networking_and_traffic.html
31. Cisco ACE - балансировка приложений. – Режим доступу: http://habrahabr.ru/post/143564/
32. Network Load Balancing technical Overview. Microsoft Developer Network white paper. – Режим доступу: https://msdn.microsoft.com/enus/library/bb742455.aspx

33. Load Balancing, Citrix Systems white paper. – Режим доступу: http://docs.citrix.com/content/dam/docs/en-us/netscaler/9-3/downloads/en.netscaler.ns-lb-wrapper-con-93.pdf
34. Load Balancing and Redundancy. Giritech A/S white paper. – Режим доступу: http://www.giritech.de/downloads/pdf/Load%20Balancing_whitepaper.pdf
35. William H. Press, Numerical Recipes: The Art of Scientific Computing. Third Edition – Cambridge University Press, 2007.
36. ДСанПіН 3.3.2.007-98 Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. – К.: Постанова Головного державного санітарного лікаря України від 10.12.1998 р. № 7.
37. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень. – К., 2000.- 16 с.
38. ДБН В.2.5-28:2006 Природнє і штучне освітлення. – К. : Міністерство будівництва, архітектури та житлово-комунального господарства України, 2006. – 68 с.
39. НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації ЕОМ. – Держгірпромнагляд, № 65 від 26 березня 2010 р.
40. НАПБ Б.03.002-2007 Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою