

ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ

КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

## КВАЛІФІКАЦІЙНА РОБОТА

на тему: «ДОСЛІДЖЕННЯ МЕРЕЖ CISCO ДЛЯ ЗАБЕЗПЕЧЕННЯ  
ВИСОКОЇ ПРОДУКТИВНОСТІ ТА НИЗЬКОЇ ЗАТРИМКИ ЗА  
ДОПОМОГОЮ ТЕХНОЛОГІЇ CISCO CATALYST DATA CENTER  
SWITCHES»

на здобуття освітнього ступеня магістр

за спеціальності 123 Комп'ютерна інженерія

(код, найменування спеціальності)

освітньо-професійної програми Комп'ютерні системи та мережі

(назва)

*Кваліфікаційна робота містить результати власних досліджень.  
Використання ідей, результатів і текстів інших авторів мають посилання на  
відповідне джерело*

\_\_\_\_\_

(підпис)

Павло ДУТКО

(ім'я, ПРІЗВИЩЕ здобувача)

Виконав: здобувач вищої освіти гр.КСДМ-62

Павло ДУТКО

(ім'я, ПРІЗВИЩЕ)

Керівник:

к.т.н., доцент

В'ячеслав ЧЕРЕВИК

(ім'я, ПРІЗВИЩЕ)

Рецензент:

науковий ступінь,  
вчене звання

\_\_\_\_\_

(ім'я, ПРІЗВИЩЕ)

Київ 2023

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**Навчально-науковий інститут інформаційних технологій**

Кафедра Комп'ютерної інженерії  
Ступінь вищої освіти «Магістр»

Спеціальність 123 Комп'ютерна інженерія  
Освітньо-професійна програма Комп'ютерні системи та мережі

**ЗАТВЕРДЖУЮ**

Завідувач кафедру Комп'ютерної інженерії  
Наталія ЛАЩЕВСЬКА  
*(ім'я, ПРІЗВИЩЕ)*  
“    ”      2023 року

**З А В Д А Н Н Я  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Дутко Павлу Марковичу

*(прізвище, ім'я, по батькові здобувача)*

1. Тема кваліфікаційної роботи: Дослідження мереж Cisco для забезпечення високої продуктивності та низької затримки за допомогою технології Cisco Catalyst Data Center Switches

керівник роботи В'ячеслав ЧЕРЕВИК к.т.н., доцент  
*(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)*

затверджені наказом Державного університету інформаційно-комунікаційних технологій від “19” 10 2023 р. №145

2. Строк подання кваліфікаційної роботи \_\_\_\_\_

3. Вихідні дані кваліфікаційної роботи:

3.1. Інтернет ресурси стосовно засобів даних центрів.

3.2. Технологічна документація стосовно мереж Cisco.

3.3. Науково-технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

4.1. Технології та інфраструктура великих даних.

4.2. Системи управління великими даними.

4.3. Проектування багаторівневої моделі ЦОД.

5. Перелік ілюстраційного матеріалу: *презентація*

6. Дата видачі завдання “19” жовтня 2023р.

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Підбір технічної літератури	.2023р. .2023р.	Виконано
2.	Технології та інфраструктура великих даних	.2023р. .2023р.	Виконано
3.	Системи управління великими даними	.2023р. .2023р.	Виконано
4.	Проектування багаторівневої моделі ЦОД	.2023р. .2023р.	Виконано
5.	Оформлення роботи, висновки	.2023р. .2023р.	Виконано
6.	Розробка демонстраційного матеріалу, доповідь	.2023р. .2023р.	Виконано

Здобувач вищої освіти

( підпис )

Керівник кваліфікаційної роботи

( підпис )

Павло ДУТКО

(ім'я, ПРІЗВИЩЕ)

В'ячеслав ЧЕРЕВИК

(ім'я, ПРІЗВИЩЕ)





## РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття ступеня магістр: 77 стор., 21 рис., 2 табл., 22 джерел.

*Мета роботи* – забезпечення високої продуктивності та низької затримки мережі Cisco завдяки використанню технології Cisco Catalyst Data Center Switches.

*Об'єкт дослідження* – технології Cisco Catalyst Data Center Switches.

*Предмет дослідження* – мережі Cisco.

*Короткий зміст роботи:* У дипломній роботі розглянуті основні поняття і технології, пов'язані з Великими даними. У роботі розглянуті основні характеристики технології Cisco Catalyst Data Center Switches, такі як швидкість передачі даних, пропускна здатність, масштабованість та надійність. Також проаналізовані різні сценарії використання цих комутаторів у різних типах дата-центрів, включаючи великі корпоративні мережі, хмарні сервіси та інші. Наведено детальну інформацію про багаторівневий дизайн, який Cisco рекомендує для центрів обробки даних.

**КЛЮЧОВІ СЛОВА:** КОМП'ЮТЕРНІ МЕРЕЖІ, МЕРЕЖІ CISCO, CISCO CATALYST DATA CENTER SWITCHES, ПРОДУКТИВНІСТЬ, ПРОПУСКНА ЗДАТНІСТЬ, МАСШТАБОВАНІСТЬ, НАДІЙНІСТЬ, ДАТА-ЦЕНТРИ.

## ABSTRACT

The text part of the qualification work for obtaining a master's degree: 77 pages, 2 table, 21 figures, 22 sources.

The purpose of the work is ensuring high performance and low latency of the Cisco network thanks to the use of Cisco Catalyst Data Center Switches technology.

The object of research is Cisco Catalyst Data Center Switches technology.

The subject of research is Cisco networks.

*Summary of the work:* The main features of the Cisco Catalyst Data Center Switches technology, such as data transfer speed, bandwidth, scalability and reliability, are considered in the work. Different scenarios of using these switches in different types of data centers, including large corporate networks, cloud services and others, are also analyzed. Provides details on the tiered design that Cisco recommends for data centers.

**KEYWORDS:** COMPUTER NETWORKS, CISCO NETWORKS, CISCO CATALYST DATA CENTER SWITCHES, PERFORMANCE, BANDWIDTH CAPACITY, SCALING, RELIABILITY, DATA CENTERS.

## ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1 ТЕХНОЛОГІЇ ТА ІНФРАСТРУКТУРА ВЕЛИКИХ ДАНИХ .....	11
1.1 Дані. Підходи та визначення.....	11
1.2 Життєвий цикл даних .....	12
1.2.1. Генерація даних/Збір даних .....	12
1.2.2 Обслуговування даних.....	13
1.2.3 Синтез даних.....	14
1.2.4 Використання даних.....	14
1.2.5 Публікація даних.....	15
1.2.6 Архівування даних.....	15
1.2.7 Очищення даних.....	16
1.3 Поняття метаданих .....	16
1.3.1 Життєвий цикл метаданих.....	16
РОЗДІЛ 2 СИСТЕМИ УПРАВЛІННЯ ВЕЛИКИМИ ДАНИМИ .....	20
2.1 Розподілені файлові системи.....	23
2.2 Розподілені фреймворки .....	23
2.3 Бенчмаркінг .....	23
2.4 Серверне програмування.....	24
2.5 Планування.....	24
2.6 Системи розгортання.....	24
2.7 Інтеграція даних.....	25
2.8 Інформаційна безпека.....	25
2.9 Архітектура системи обробки Великих даних.....	27
РОЗДІЛ 3 ПРОЕКТУВАННЯ БАГАТОРІВНЕВОЇ МОДЕЛІ ЦОД .....	40
3.1 Потік трафіку в ядрі дата-центру.....	44
3.2 Рівень агрегації центру обробки даних .....	46
3.3 Потік трафіку на рівні агрегації ЦОД .....	49
3.4 Вибір шляху за наявності сервісних модулів.....	50



3.4.1 Потік трафіку серверної ферми з сервісними модулями.....	51
3.4.2 Потік трафіку серверної ферми без сервісних модулів.....	52
3.5 Масштабування рівня агрегації.....	53
3.6 Розмір домену несправності рівня 2.....	55
3.7 Масштабованість остовного дерева.....	55
3.8 Резервування шлюзу за замовчуванням з HSRP.....	57
3.9 Рівень доступу до центру обробки даних.....	58
3.10 Спільне використання послуг на рівні агрегації.....	65
3.11 Рівень послуг центру обробки даних.....	66
3.12 Потік трафіку через сервісний рівень.....	70
ВИСНОВКИ.....	73
ПЕРЕЛІК ПОСИЛАНЬ.....	75

## ВСТУП

У сучасному світі, де швидкість передачі даних та низька затримка є критичними факторами для ефективної роботи бізнесу, мережеві рішення стають все більш важливими. Одним з провідних постачальників мережевих рішень є компанія Cisco, яка пропонує широкий спектр продуктів, спрямованих на забезпечення високої продуктивності та низької затримки.

Метою даної магістерської роботи є дослідження мереж Cisco для забезпечення високої продуктивності та низької затримки за допомогою технології Cisco Catalyst Data Center Switches. Основним завданням дослідження є вивчення можливостей та переваг використання цих комутаторів у сучасних дата-центрах, а також аналіз їх впливу на продуктивність та затримку мережі.

У роботі розглянуті основні характеристики технології Cisco Catalyst Data Center Switches, такі як швидкість передачі даних, пропускна здатність, масштабованість та надійність. Також проаналізовані різні сценарії використання цих комутаторів у різних типах дата-центрів, включаючи великі корпоративні мережі, хмарні сервіси та інші. Наведено детальну інформацію про багаторівневий дизайн, який Cisco рекомендує для центрів обробки даних. Модель багаторівневого дизайну підтримує багато архітектур веб-сервісів, у тому числі на основі Microsoft.NET і Java 2 Enterprise Edition. Ці прикладні середовища веб-служб використовуються для звичайних рішень ERP, таких як PeopleSoft, Oracle, SAP, BAAN і JD Edwards; і рішення CRM від таких постачальників, як Siebel і Oracle.

Баторівнева модель спирається на багаторівневу мережеву архітектуру, що складається з ядра, агрегації та рівня доступу. У цій дипломній роботі більш детально описано апаратне забезпечення та рекомендації щодо проектування для кожного з цих рівнів.

# 1 ТЕХНОЛОГІЇ ТА ІНФРАСТРУКТУРА ВЕЛИКИХ ДАНИХ

## 1.1 Дані. Підходи та визначення

Відповідно до ГОСТ Р 52653-20061, дані - це представлення інформації у формалізованій формі, придатній для передачі, інтерпретації та обробки.

Згідно з ГОСТ 7.0-992, дані - це інформація, оброблена і представлена в формалізованому вигляді для подальшої обробки.

Кембриджський словник дає такі визначення даних: дані - це інформація, особливо факти і цифри, зібрана для подальшого використання при прийнятті рішень.

Дані - це інформація в електронному вигляді, яка може зберігатися і використовуватися комп'ютером.

Сьогодні також досить поширено говорити, що дані – це нафта цифрової економіки.

Спочатку поняття даних філософське, воно виникає в гносеології, коли розглядається основа проблеми гносеології - пізнаваність світу, пошук і досягнення істини. Процедури верифікації або фальсифікації даних створюють інформацію, досягнення істини створює знання.

Філософія займається перетворенням інформації в дані, даних в інформацію, а інформації в знання. Правдивість інформації суб'єктивна. Інформація, виражена в офіційному представленні, є даними. Обробка даних дозволяє визначити, скільки інформації вони містять. Коли експерт осмислює інформацію, створюються знання.

## **1.2 Життєвий цикл даних**

Життєвий цикл даних – це послідовність кроків, які проходить певний фрагмент даних від початкового етапу створення або отримання до моменту архівування або видалення.

Основні етапи життєвого циклу даних показані на рис. 1.1. Розглянемо ці етапи докладніше.

### **1.2.1 Генерація даних/Збір даних**

На цьому етапі дані генеруються або фіксуються. Цей етап прийнято ділити на три види збору даних:

- збір даних. Організація отримує дані, які вже створені за межами підприємства.
- введення даних. Створення нових даних оператором або комп'ютером. Дані мають цінність для підприємства.
- прийом сигналу. Захоплення даних пристроєм. Це особливо важливо в системах управління, але останнім часом це особливо цінно при використанні такого підходу, як Інтернет речей.

Всі три варіанти збору даних дуже важливі при розгляді процесу управління даними.

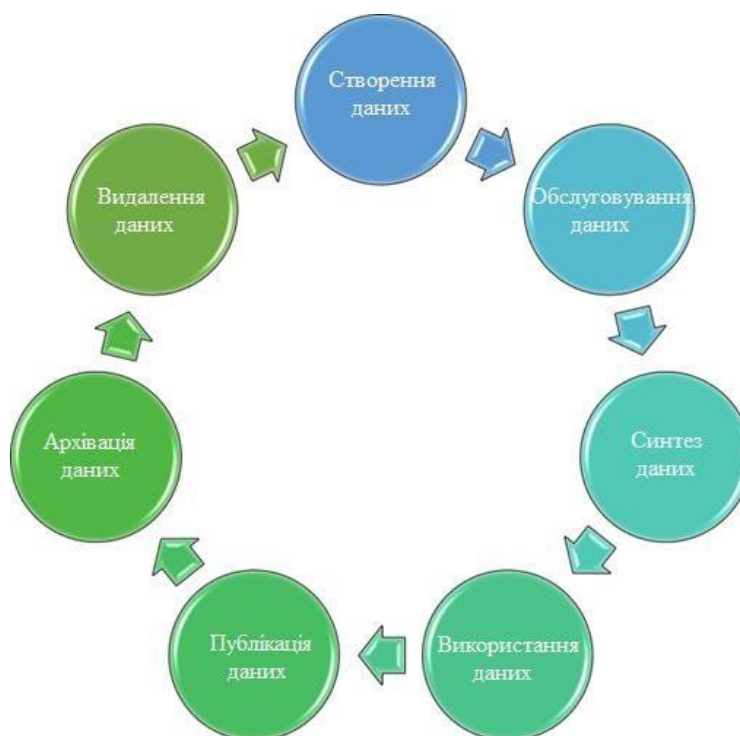


Рисунок 1.1 – Життєвий цикл даних

### 1.2.2 Обслуговування даних

Після того, як дані були створені, їх потрібно зберігати та підтримувати. Дані повинні бути доставлені в точку, де вони будуть використовуватися або оброблятися (наприклад, синтез).

Можна сказати, що ведення даних - це обробка даних без отримання корисної для підприємства інформації від її вилучення з них.

Часто обслуговування даних включає в себе дії з маніпулюванням даними, такі як переміщення, інтеграція, очищення, збагачення та процеси ETL (вилучення, перетворення, завантаження).

Обслуговування даних, як правило, передбачає використання широкого спектру методів управління даними.

### 1.2.3 Синтез даних

Це відносно недавній етап життєвого циклу даних. Використовується не в усіх моделях життєвого циклу даних.

Синтез даних - це процес вилучення додаткової цінності з даних за допомогою використання індуктивної логіки і сторонніх джерел інформації.

Це етап, на якому аналітики працюють з даними, і вони можуть використовувати методи моделювання ризиків, моделювання для прийняття інвестиційних рішень тощо.

На цьому етапі використовується індуктивна, а не дедуктивна логіка. Індуктивна логіка вимагає використання експертної думки, так як саме компетенції експертів необхідні для побудови скорингових моделей і т.д.

### 1.2.4 Використання даних

До цього часу йшлося про використання даних у межах одного підприємства, які, можливо, були очищені та збагачені на етапі обслуговування даних і використані разом із додатковими сторонніми джерелами даних на етапі синтезу даних.

На етапі використання даних дані використовуються як корисна інформація для завдань, які потрібно виконувати та керувати ними на основі даних.

Ці завдання можуть бути поза життєвим циклом даних. Однак дані стають все більш важливою частиною корпоративних бізнес-процесів. Самі дані можуть бути продуктом або послугою (або бути частиною продукту чи послуги), які пропонує компанія.

Використання даних має особливі завдання в рамках управління даними. Одним із завдань є правомірне використання даних у необхідній формі. Це називається «дозволене використання даних». Можуть існувати нормативні або договірні обмеження щодо того, як дані можуть фактично використовуватися, і

частина ролі управління даними полягає в забезпеченні дотримання цих обмежень.

### **1.2.5 Публікація даних**

При використанні даних можлива відправка даних за межі підприємства. В даному випадку мова йде про публікацію даних. Публікація даних – це акт переміщення даних за межі підприємства.

Прикладом цього процесу може бути брокер, який надсилає клієнтам щомісячні звіти. Всі надіслані дані більше не можуть бути відкликані. Якщо були відправлені дані з неправильними значеннями, дані не можуть бути виправлені, оскільки вони більше не доступні підприємству. Управління даними може знадобитися, щоб допомогти прийняти рішення про те, як оброблятимуться неправильні дані, надіслані з підприємства.

### **1.2.6 Архівування даних**

Дані можна використовувати один раз або кілька разів.

Але потім, рано чи пізно, життєвий цикл даних починає добігати кінця. Першим етапом цього стану є архівування даних.

Архівування даних — це акт копіювання даних у пасивне середовище, в якому вони зберігаються, на той час, коли вони знову знадобляться в активному виробничому середовищі, і видалення цих даних з усіх активних виробничих середовищ.

Архів даних – це просто місце, де зберігаються дані, без їх обслуговування, використання чи публікації. При необхідності дані можна відновити з архіву.

### 1.2.7 Очищення даних

Знищення даних – це послідовність операцій для виконання остаточного видалення даних, що унеможливорює відновлення даних або отримання інформації про них. Це одна з найскладніших процедур управління даними.

Навіть з теоретичної точки зору є команда на запис значення в ділянку пам'яті, але немає команди на стирання значення як такого. Для знищення даних необхідно виготовити високопродуктивне джерело випадкових чисел і перезаписати ними весь носій інформації (перезапису області зберігання недостатньо, так як зберігається інформація про вихідний обсяг даних).

Іншими словами, при знищенні даних необхідно зробити недоступними для відновлення на фізичному рівні не тільки самі дані, але і пов'язану з ними інформацію в інших наборах даних.

### 1.3 Поняття метаданих

Під час збору даних генеруються метадані, які містять певну інформацію про зібрані дані. Наприклад, час створення набору даних, авторство та походження, розмір та кодування даних є метаданими.

Метадані - це інформація про дані. Структуровані метадані називаються онтологією або схемою метаданих. Вона включає в себе машинно-інтерпретовані формулювання основних понять предметної області і взаємозв'язків між ними.

Онтології зазвичай використовуються в таких галузях, як штучний інтелект, семантична мережа, системна інженерія, біомедична інформатика, бібліотекознавство, інформаційна архітектура тощо.

#### 1.3.1 Життєвий цикл метаданих

Життєвий цикл метаданих можна розділити на чотири етапи, що показані на рисунку 1.2:



- оцінка вимог та контент-аналіз;
- специфікація системних вимог;
- система метаданих;
- сервіс та оцінка.

Розглянемо ці етапи докладніше.

*Оцінка вимог та контент-аналіз.* Цей етап складається з чотирьох частин:

- визначення та виведення базових вимог до метаданих;
- огляд відповідних стандартів метаданих і проєктів;
- вивчення глибоких вимог до метаданих;
- визначення стратегії для схем метаданих.

*Визначення та отримання базових вимог до метаданих.* На цьому етапі життєвого циклу метаданих експерти та профільні експерти обпитуються щодо вимог до метаданих конкретного проєкту та аналізу атрибутів. Метою опитування експертів є отримання попередньої інформації про проєкт та встановлення контакту між фахівцями з метаданих та постачальниками даних (або контенту). На цьому етапі слід уточнити та уточнити сферу застосування метаданих, контекст метаданих, поточну систему метаданих, а також роль і функції метаданих.

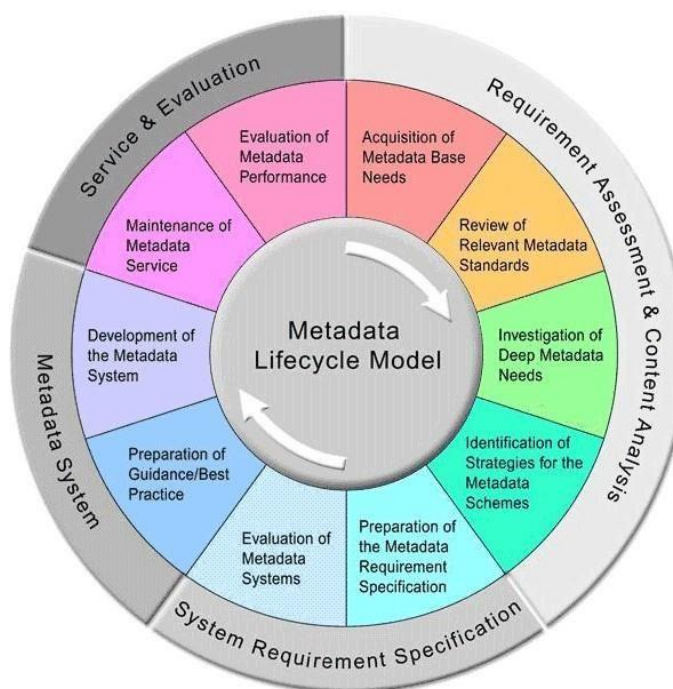


Рисунок 1.2 – Життєвий цикл метаданих

Огляд відповідних стандартів метаданих і проектів. Цей етап включає визначення стандартів метаданих, які потенційно можуть бути корисними для проекту, вивчення існуючих схем метаданих та варіантів використання. Цей етап необхідний для того, щоб можна було краще зрозуміти, які відмінності існують серед інших подібних проектів, а також переглянути цілі проекту.

На цьому етапі вимоги до метаданих визначаються більш детально та глибоко. Для цього використовується поняття контент-аналізу. Тут важливо уточнити сферу застосування та зміст метаданих, а також питання планованих СУБД та інформаційних систем, в яких планується використовувати метадані. Надалі ці розробки можуть бути використані як метод масштабування.

*Визначення стратегій для схем метаданих.* На цьому етапі формулюється стратегія метаданих на основі всіх попередніх результатів. Стратегія включає прийняття одного або кількох існуючих стандартів метаданих і розробку схеми метаданих на основі цих стандартів.

*Специфікація системних вимог.* Цей етап складається з двох частин:

- підготовка специфікації вимог до метаданих;
- оцінка систем метаданих.

Специфікація вимог до метаданих (MRS) є сполучною ланкою між зацікавленими сторонами проекту, фахівцями з метаданих і розробниками системи.

*Оцінка систем метаданих.* На цьому етапі оцінюються системи метаданих, які потенційно можуть бути використані в проекті в майбутньому. Учасники проекту можуть обирати з існуючих систем метаданих.

*Система метаданих.* Цей етап складається з двох частин:

- підготовка посібника з найкращої практики;
- розробка системи метаданих.

*Підготовка посібника з найкращої практики.* Цей етап включає створення документації та розробку настанов щодо «найкращої практики» для окремих елементів метаданих. Посібник можна використовувати як контрольний список або як засіб забезпечення контролю якості записів метаданих у проекті.

*Розробка системи метаданих.* Розробники систем розробляють інструменти та системи метаданих на основі специфікації вимог до метаданих.

*Сервіс та оцінка.* Цей етап складається з двох частин.

- підтримка сервісу метаданих;
- оцінка якості метаданих. Підтримка сервісу метаданих.

Метою розробки сервісів метаданих є забезпечення якості метаданих та механізмів роботи з ними. Модель сервісу метаданих складається з трьох основних елементів: механізму корисності, ролей і зв'язків між ролями.

*Оцінка ефективності метаданих.* Останній етап життєвого циклу метаданих спрямований на розгляд результатів усього процесу метаданих і якості самих метаданих. Інтегральна оцінка складається з оцінки якості метаданих, оцінки ефективності схеми метаданих, оцінки результатів використання інструментів створення метаданих та оцінки застосування моделі життєвого циклу метаданих на кожному етапі.

## 2 СИСТЕМИ УПРАВЛІННЯ ВЕЛИКИМИ ДАНИМИ

Якщо давати стисле визначення, то Великі дані - це дані, які не поміщаються в оперативну пам'ять комп'ютера.

По суті, це визначення позначає те, що властивість "бути великим" є не самостійною властивістю даних, а залежить від характеристики системи, застосовуваної для їхнього опрацювання.

Наприклад, звичайній людині важко запам'ятати, яка саме температура була в нашому місті щодня за минулий місяць. Таким чином, три десятки значень цілком можуть бути прикладом Великих даних. Однак ось людина впевнено повідомляє "минулий місяць був холодним". Це повідомлення несе інформацію про оброблені дані: на думку співрозмовника, середня температура за минулий місяць була нижчою, ніж зазвичай у цьому місяці за кілька десятків років.

Іншим прикладом можуть бути дані про об'єкти, які теоретично несуть важливу інформацію, проте мають такий розмір, що ці дані практично неможливо не тільки обробити або зберегти, а й навіть зібрати. Розглянемо, наприклад, набір даних, що містить координати і швидкості молекул у повітряному стовпі над територією аеропорту. Є також метадані з описом, у який момент проводилося вимірювання і що це за молекула. Такий набір даних несе інформацію про погодні умови над аеропортом, включно з температурою, тиском, вологістю, хмарністю, особливими погодними умовами - торнадо, що проходить, або град, що падає. З іншого боку, для коректного опрацювання дані для всіх молекул мають бути досить повними і репрезентативними для статистичного опрацювання.

У результаті такого уявного експерименту ми розуміємо, що для ефективної роботи з великими даними потрібна модель даних, що дає змогу сформулювати методи роботи з даними.

Дані можуть бути різних типів. Інформацію, отриману в результаті обліку або вимірювання будь-яких об'єктів чи параметрів, називають майстер-даними

(Master Data). Наприклад, облік кількості, заміри координат і швидкостей конкретних молекул - це майстер-дані.

Транзакційні дані (в англійській літературі застосовують терміни Transactional Data, Application Specific Data, Operational Data) - це дані, що відображають результат виконання будь-яких операцій. Наприклад, дані про взаємодію молекул між собою, а саме про перетин меж розглянутої області, про траєкторію конкретної молекули, про випаровування крапель дощу - це транзакційні дані. Транзакційні дані описують взаємодію об'єктів один з одним або з навколишнім світом, які можна отримати за допомогою обробки майстер-даних.

Ретроспективні дані (Historical data) - це дані, забезпечені мітками часу. Наприклад, з одного боку, ми можемо зберігати дані про координату і вектор швидкості кожної молекули, але якщо у нас є набір координат залежно від часу, то швидкість молекули стає зайвою, її обчислюють, виходячи з моделі, описуваної ньютонівською механікою.

Посилальні дані (довідники, НДІ, нормативно-посилальна інформація, Reference Data, Lookup Data, Dictionaries) - це базові незмінні дані, заздалегідь відомі із зовнішніх джерел, такі як нормативи, скорочення, акроніми, словники, стандарти. Наприклад, питомі ваги молекул, залежність температури замерзання і кипіння від тиску, залежність середньої швидкості молекул (швидкості звуку) від температури.

Формат даних. Структуровані дані мають заздалегідь визначений формат.

Напівструктуровані або слабоструктуровані дані - це дані, часто зібрані з різних джерел. Структура даних документована, але залежно від джерела даних конкретний формат подання інформації може бути різним. Неструктуровані дані потребують обов'язкового опрацювання і подальшої валідації перед використанням. Наприклад, дані про координати і швидкості молекул, у яких деякі координати пропущені або деякі записи повторюються, є напівструктурованими. Нам потрібно зрозуміти, чому так сталося і перед

використанням або виключити такі дані (що може призвести до систематичної помилки), або, виходячи з моделі даних, відновити пропущені значення.

Дані, в яких координати вимірюються в різних одиницях виміру, числа іноді записані словами, іноді латинськими цифрами, а іноді у вигляді сканованого зображення почерку лаборанта, є неструктурованими даними.

Зазвичай Великі дані описують за допомогою таких характеристик:

- обсяг (Volume) - кількість згенерованих і збережених даних. Розмір даних визначає значущість і потенціал даних, а також те, чи можуть вони бути розглянуті як Великі дані;

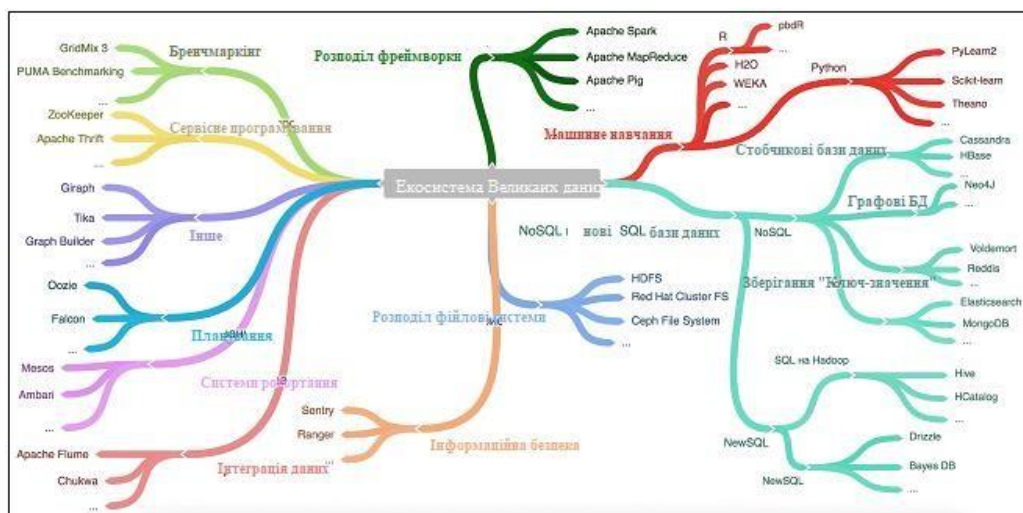
- різноманітність (Variety) - тип даних. Великі дані можуть складатися з тексту, зображень, аудіо, відео. Великі дані під час зіставлення один з одним можуть доповнювати відсутні дані;

- швидкість (Velocity) - швидкість. Тут мається на увазі швидкість, з якою дані генеруються й обробляються. Дуже часто Великі дані використовуються в режимі реального часу;

- мінливість (Variability) - суперечливість наборів даних може перешкоджати їхньому опрацюванню та управлінню ними;

- достовірність (Veracity) - якість даних безпосередньо впливає на точність проведення аналізу даних.

Великі дані можуть бути класифіковані відповідно до кількох головних компонентів. Інтелект-карта, представлена на рис. 2.1.



## Рисунок 2.1 - Інтелект-карта екосистеми Великих даних

На рис. 2.1 за допомогою інтелект-карти показано компоненти екосистеми Великих даних. Розглянемо ці компоненти докладніше.

### 2.1 Розподілені файлові системи

Для зберігання й оброблення Великих даних створено розподілені системи зберігання даних, зокрема розподілені файлові системи, що дають змогу використовувати зовнішній файловий простір системи зберігання для опрацювання даних на нодах, які входять до обчислювальних кластерів.

Найчастіше зручно використовувати розподілені файлові системи, орендовані як окремий хмарний сервіс, наприклад, Google Colossus, Amazon S3.

### 2.2 Розподілені фреймворки

Оброблення даних, що перебувають на розподілених системах зберігання, ведеться паралельно на комп'ютерах, що становлять вузли (nodes) обчислювального кластера. Для організації обчислень розробники систем обробки використовують розподілені фреймворки. Більшість фреймворків доступні за ліцензією Apache і орієнтовані на роботу в кластерах на базі Linux. Існують також хмарні фреймворки, орендовані як окремий хмарний сервіс.

### 2.3 Бенчмаркінг

Цей клас інструментів було розроблено для оптимізації інсталяції Великих даних за допомогою використання стандартизованих профілів (Profiling suites). Бенчмаркінг та оптимізація інфраструктури Великих даних часто не є сферою відповідальності дата-науковців, це сфера відповідальності для окремих професіоналів, що спеціалізуються на ІТ-інфраструктурі. Використання

оптимізованої інфраструктури може істотно знизити вартість використовуваного обладнання.

## **2.4 Серверне програмування**

Припустимо, що ви зробили додаток для прогнозування результатів футбольних матчів світового класу на платформі Hadoop, і ви хочете дозволити іншим використовувати прогнози, зроблені вашим додатком. Тим не менш, ви не маєте уявлення про архітектуру або технології всіх, хто прагне використовувати ваші прогнози. Сервісні інструменти дають змогу надавати додатки на Великих даних іншим додаткам як службу. Дана-вченим іноді доводиться надавати свої моделі через служби. Найбільш відомим прикладом тут є REST-сервіс; REST означає репрезентативну передачу стану (REpresentational State Transfer, REST). Вона часто використовується як обмін даними з веб-сайтами.

## **2.5 Планування**

Інструменти планування дають змогу автоматизувати повторювані завдання і запускати завдання на основі таких подій, як додавання нового файлу в папку. Вони схожі на такі інструменти, як CRON у Linux, але спеціально розроблені для роботи у відмовостійкому кластері. Ви можете використовувати їх, наприклад, для запуску завдання MapReduce щоразу, коли в каталозі є новий набір даних.

## **2.6 Системи розгортання**

Налаштування інфраструктури Великих даних - непросте завдання, і розгортання нових додатків у кластері Великих даних - це зона відповідальності інженерів з Великих даних. Вони значною мірою автоматизують встановлення та налаштування компонентів Великих даних.



## 2.7 Інтеграція даних

Припустимо, що вже є розподілена файлова система, і тепер необхідно перенести дані з одного джерела в інше. У таких випадках використовують фреймворки для інтеграції даних, такі як Apache Sqoop і Apache Flume. Цей процес схожий на процес вилучення, перетворення та завантаження (Extract, Transform and Load, ETL) у традиційному сховищі даних.

## 2.8 Інформаційна безпека

Засоби забезпечення безпеки Великих даних дають змогу здійснювати централізований контроль доступу до даних. Безпека Великих даних стала самостійною дисципліною, і дата-науковці зазвичай стикаються з нею лише як споживачі даних. Безпекою Великих даних займаються експерти з інформаційної безпеки.

Якщо у вас є Великі дані, то було б непогано отримати з них корисний контент. Це можна зробити за допомогою використання методів машинного навчання, статистики та прикладної математики.

Ще перед Другою світовою війною багато трудомістких обчислень проводили вручну, що природним чином обмежувало можливості аналізу даних. Після Другої світової війни стала активно розвиватися обчислювальна техніка і наукові обчислення. З'явилася можливість писати програми з формулами й алгоритмами, а потім завантажувати в програми різні дані.

На сьогоднішній день, коли з'явилася величезна кількість даних, один комп'ютер уже не в змозі впоратися із завданням їх обробки. Деякі алгоритми, розроблені в минулому столітті, на жаль, не зможуть впоратися з цим завданням, навіть якщо теоретично можна було б підключити до вирішення завдання всі комп'ютери Землі. Це пов'язано з часовою складністю алгоритму.

Одна з найбільших проблем зі старими алгоритмами полягає в тому, що вони недостатньо масштабуються. З огляду на обсяг даних, які необхідно

аналізувати сьогодні, це стає проблематичним. Для обробки цього обсягу даних потрібні спеціалізовані структури та бібліотеки. Наприклад, у мові Python є такі бібліотеки: Scikit-learn (бібліотека машинного навчання), PyBrain (для роботи з нейронними мережами), NLTK (для обробки природної мови), Pylearn2 (ще одна бібліотека машинного навчання), TensorFlow (бібліотека глибокого навчання, є програмний інтерфейс API для мови Python), Keras (бібліотека для роботи з нейронними мережами) тощо.

Існує також Apache Spark - програмний каркас з відкритим вихідним кодом для реалізації розподіленого оброблення неструктурованих і слабоструктурованих даних.

Використання реляційних баз даних для опрацювання Великих даних вкрай неефективне через високі накладні витрати. Традиційно для обробки Великих даних використовуються бази даних типу "ключ - значення" (Key value database або HashDB). Одну з перших баз цього типу DBM було реалізовано Ken Thompson для AT&T Unix 7 у 1979 році.

База даних виду "ключ - значення", по суті, являє собою асоціативний масив (Hash, Dict), тобто множину, що складається з пар (Key, Value). У деяких реалізаціях на множині ключів вводиться відношення порядку, і ми можемо отримати значення послідовно в міру зростання ключа. В інших випадках сортування за ключем нестійке за однакових ключів і за неодноразових вибірках можна отримати різну послідовність пар.

Велику кількість баз даних можна розділити на такі типи:

- стовпчасті бази даних (Column databases). Дані зберігаються в стовпцях, що дає змогу алгоритмам виконувати набагато швидші запити;
- сховища документів (Document stores) Сховища документів більше не використовують таблиці, але зберігають кожне спостереження в документі. Це дає змогу використовувати набагато гнучкішу схему даних;
- потокові дані (Streaming data). Дані збираються, перетворюються й агрегуються не в партіях, а в реальному часі;

- сховища для ключів (Key-value stores). Дані не зберігаються в таблиці; для кожного значення призначається ключ (як розказано про це вище);
- SQL на Hadoop - пакетні запити на Hadoop, що використовують фреймворк MapReduce у фоновому режимі;
- новий SQL (New SQL). Цей тип поєднує масштабованість баз даних NoSQL з перевагами реляційних баз даних. Тут використовується інтерфейс SQL і реляційна модель даних;
- графові бази даних (Graph databases). Це тип баз даних, що використовують графові структури для семантичних запитів із вузлами і ребрами та властивостями для представлення і зберігання даних. Класичним прикладом цього типу є соціальна мережа.

## **2.9 Архітектура системи обробки Великих даних**

Для роботи з Великими даними використовуються складні системи, в яких можна виділити кілька компонентів або шарів (Layers). Зазвичай виокремлюють чотири рівні компонентів таких систем: приймання, збирання, аналіз даних і подання результатів (рис. 2.2). Цей поділ є значною мірою умовним, оскільки, з одного боку, кожний компонент, своєю чергою, може бути розділений на підкомпоненти, а з іншого - деякі функції компонентів можуть перерозподілятися залежно від розв'язуваного завдання та використовуваного програмного забезпечення, наприклад, виділяють зберігання даних в окремий шар.

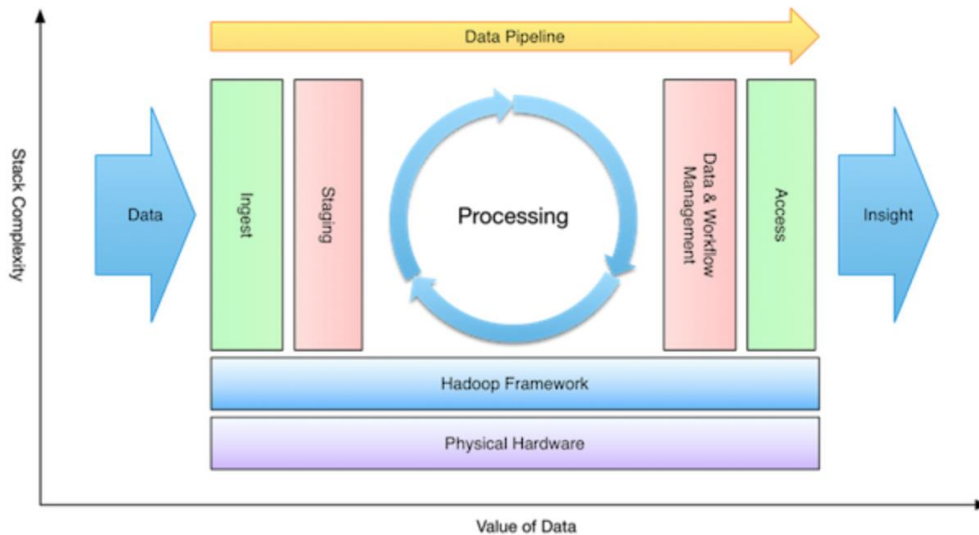


Рисунок 2.2 - Стек роботи з Великими даними

Для роботи з Великими даними розробниками систем створюються моделі даних, змістовно пов'язані з реальним світом. Розроблення адекватних моделей даних являє собою складне аналітичне завдання, яке виконують системні архітектори та аналітики. Модель даних дає змогу створити математичну модель взаємодій об'єктів реального світу і містить опис структури даних, методи маніпуляції даними та аспекти збереження цілісності даних.

Для зберігання даних використовуються розподілені системи різних типів. Це можуть бути файлові системи, бази даних, журнали, механізми доступу до загальної віртуальної пам'яті. Більшість систем зберігання орієнтовані виключно на роботу з Великими даними, вони мають вкрай обмежену кількість функцій (наприклад, може бути відсутня можливість не тільки модифікації, а й видалення даних, що надійшли), що пояснюється внутрішньою складністю створення вискоелективних розподілених систем.

Для того, щоб робота з даними відбувалася швидше, системи зберігання та обробки даних розпаралелюються в кластері (cluster, група комп'ютерів, об'єднаних мережею для виконання єдиного завдання). Однак, згідно з гіпотезою Брюера, неможливо забезпечити одночасну узгодженість (несуперечливість) даних, доступність даних і стійкість системи до відокремлення окремих вузлів.

Гіпотеза доведена для транзакцій типу ACID (Atomic, Consistent, Isolated, Durable) і відома під назвою CAP теорема (Consistency, Availability, Partition tolerance).

Джерела даних мають різні параметри, як-от частоту надходження даних із джерела, обсяг порції даних, швидкість передавання даних, тип даних, що надходять, та їхню достовірність.

Для ефективного збору даних необхідно встановити джерела даних. Це можуть бути сховища даних, постачальники агрегованих даних, API будь-яких датчиків, системні журнали, згенерований людиною контент у соціальних мережах, у корпоративних інформаційних системах, геофізична інформація, наукова інформація, успадковані дані з інших систем. Джерела даних визначають вихідний формат даних.

Наприклад, ми можемо самостійно проводити погодні на території аеропорту, використовувати дані, що надходять з літаків, які злітають і сідають, закупити дані з супутників, які пролітають над аеропортом, та в місцевій метеослужби, а також знайти їх десь у мережі в іншому місці. У загальному випадку для кожного джерела необхідно створювати власний збирач (Data Crawler для збору інформації в мережі і Data Acquisition для проведення вимірювань).

Приймання даних полягає в початковій підготовці даних від джерел з метою приведення даних до загального формату представлення даних. Цей єдиний формат вибирається відповідно до прийнятої моделі даних. Виконуються перетворення систем вимірювання, типів (типізація), верифікація. Обробка даних змістовно не зачіпає наявну в даних інформацію, але може змінювати її подання (наприклад, приводити координати до єдиної системи координат, а значення до єдиної розмірності).

Етап збирання даних характеризується безпосередньою взаємодією із системами зберігання даних. Встановлюється точка збору, в якій зібрані дані забезпечуються локальними метаданими і поміщаються в сховище або передаються для подальшої обробки. Дані, які з якихось причин не пройшли точку збору, ігноруються.

Для структурованих даних проводиться перетворення з вихідного формату за задалегідь заданими алгоритмами. Це найефективніша процедура в разі, якщо структура даних відома. Однак якщо дані подано в двійковому вигляді, структура і зв'язки між даними загублені, то розробка алгоритмів і заснованого на них програмного забезпечення для опрацювання даних може виявитися вкрай скрутною.

Для напівструктурованих даних потрібна інтерпретація даних, що надходять, і використання програмного забезпечення, яке вмiє працювати з використовуваною мовою опису даних. Істотним плюсом напівструктурованих даних є те, що в них часто містяться не тільки самі дані, а й метадані у вигляді інформації про зв'язки між даними і способи їхнього отримання.

Розробка програмного забезпечення для обробки напівструктурованих даних являє собою досить складне завдання. Однак є значна кількість готових конвертерів, які можуть, наприклад, витягти дані з формату XML у сформоване табличне представлення.

Найбільшого обсягу робіт вимагає обробка неструктурованих даних. Для їхнього переведення до заданого формату може знадобитися створення спеціального ПЗ, складне ручне опрацювання, розпізнавання і вибіркового ручний контроль.

На етапі збирання проводиться контроль типів даних і може виконуватися базовий контроль достовірності даних. Наприклад, координати молекул газу, що містяться в будь-якій області, не можуть лежати за межами цієї області, а швидкості - істотно перевищувати швидкість звуку. Для того, щоб уникнути помилок типізації, необхідно перевіряти, чи правильно задано одиниці виміру. Наприклад, в одному наборі даних висота може вимірюватися в кілометрах, а в іншому - у футах. У цьому разі необхідно здійснити перетворення висоти в ті одиниці виміру, які прийняті у використовуваній моделі.

Під час збирання дані систематизуються і забезпечуються метаданими, що зберігаються у пов'язаних метаданих. За наявності великої кількості джерел даних

може знадобитися управління збором даних для того, щоб збалансувати обсяги інформації, що надходять із різних джерел.

Зібрані дані або зберігаються в системах зберігання, або (особливо, для поточкових даних) передаються для аналізу в реальному часі.

Аналіз даних, на відміну від збирання даних, використовує інформацію, що міститься в самих даних. Аналіз може проводитися як у реальному часі, так і в пакетному режимі. Аналіз даних становить основне за трудомісткістю завдання під час роботи з Великими даними.

Існує безліч методик опрацювання даних: предиктивний аналіз, запити та звітність, реконструкція за математичною моделлю, трансляція, аналітичне опрацювання та інші. Методики використовують специфічні алгоритми залежно від поставлених цілей. Наприклад, аналітичне опрацювання може бути аналізом зображень, соціальних мереж, географічного розташування, розпізнавання за ознаками, текстовим аналізом, статистичним опрацюванням, аналізом голосу, транскрибуванням.

Алгоритми аналізу даних також, як і алгоритми обробки даних, спираються на модель даних. При цьому під час аналізу може бути використано кілька моделей, що задають загальний формат даних, але по-різному моделюють змістовні процеси, дані про які ми обробляємо. У разі використання під час аналізу методів штучного інтелекту, зокрема нейронних мереж, проводиться динамічне навчання моделей на різних наборах даних.

Під час аналізу даних здійснюють ідентифікацію сутностей, що описуються даними, на підставі наявної в даних інформації та використовуваних моделей. Сутністю аналізу є аналітичний механізм, що використовує аналітичні алгоритми, управління моделями та ідентифікацію сутностей для отримання нової змістовної інформації, яка є результатом аналізу.

Результати аналізу даних надаються на рівні споживання. Є кілька механізмів, що дають змогу використовувати результати аналізу великих даних:

- моніторинг метаінформації. Підсистема відображення в реальному часі істотних параметрів роботи системи, завантаженості обчислювачів, розподілу

завдань у кластері, розподілу інформації в сховищах, наявності вільного місця у сховищах, надходження даних від джерел, активності користувачів, відмов обладнання тощо;

- моніторинг даних. Підсистема відображення в реальному часі процесів приймання, збирання та аналізу даних, навігація за даними;

- генерація звітів, запити до даних, подання даних у вигляді візуалізації на дашбордах (Dashboard), у форматі PDF, інфографіці, зведених таблицях і коротких довідках;

- перетворення даних та експорт в інші системи, інтерфейс з ВІ-системами.

Для паралельного опрацювання даних в інтерфейсі MPI (Message Passing Interface), що є поширеним стандартом для обміну даними під час організації паралельних обчислень, було запропоновано нині широко використовувану в безлічі реалізацій парадигму паралельного опрацювання наборів даних за допомогою використання операторів Map і Reduce.

Згортка в найпростішому випадку отримує на вході функцію від двох параметрів із набору даних, що складається з елементів одного типу. На виході вона повертає один елемент такого самого типу. У загальному випадку від функції вимагається комутативність і асоціативність на множині визначення. У цьому разі порядок обчислень несуттєвий, алгоритм дає змогу ефективно розпаралелюватися, адже є можливість вибрати довільні пари елементів набору даних, від обраних пар за допомогою переданої в Reduce функції паралельно обчислити значення і вихідні пари елементів замінити на обчислені значення. Алгоритми, в яких необхідна нам функція згортання є комутативною та асоціативною, високоефективні під час опрацювання Великого неупорядкованого (або впорядкованого, для обчислення це несуттєво) набору даних.

Для асоціативної, але некомутативної функції згортка визначена для набору даних, елементи якого впорядковані. У цьому разі ми можемо розбити вихідний набір на кілька впорядкованих між собою послідовних піднаборів, наприклад, за кількістю наявних обчислювачів у кластері. З математичної точки зору ця процедура відповідає розстановці дужок. Потім, отримавши по одному елементу



даних з кожного піднабору, ми маємо проміжний упорядкований набір даних, і виконуємо згортку над ним і далі рекурсивно до отримання підсумкового значення.

У складнішому випадку ми хочемо під час роботи Reduce отримати дані іншого типу, наприклад, на вході ми маємо набір з масою і швидкостями молекул, а на виході нам потрібно отримати сумарну масу й імпульс (щоб дізнатися середню швидкість вітру, розділивши імпульс на масу). Для розв'язання такої задачі функція, що передається у згортку, істотно видозмінюється так: вона отримує один параметр результуючого типу (званий акумулятором) і один параметр вихідного типу (ітератор). На самому початку акумулятор має деяке початкове значення. Для нашого прикладу з молекулами початкове значення, що передається у згортку, - це структура з чотирьох елементів: маса дорівнює нулю і три компоненти (за координатами  $x$ ,  $y$  і  $z$ ) імпульсу теж дорівнюють нулю.

Потім елементи даних перебираються і проводяться обчислення, визначені функцією. У нашому прикладі масу молекули, переданої в ітераторі, додають до маси в акумуляторі, далі обчислюють імпульс молекули (вектор із добутків маси на компоненти швидкості) і додають до компонентів імпульсу акумулятора.

Розбивши вихідний набір даних на піднабори ми для кожного обчислимо масу та імпульс. Однак далі у нас з'являється проблема: наявна функція вміє додавати до даних, що мають тип акумулятора (маса, імпульс), дані, що мають тип ітератора (маса, швидкість). Щоб зробити підсумкові обчислення, нам необхідно провести перевизначення функції: у тому випадку, якщо передається два параметри, що мають тип акумулятора, то функція починає поводитися зовсім по-іншому. А саме складає маси та імпульси. Інакше кажучи, по суті у нас з'являються дві функції: одна працює з майстер-даними, а інша з проміжними даними. Таким чином, ми можемо провести ефективне розпаралелювання. Звернемо увагу, що цей метод також допускає роботу з даними, що не комутують, але впорядкованими, у разі асоціативних функцій.

Іноді для зручності обчислень, окрім описаного вище прямого (лівого) згортання, за якого функцію обчислюють від перших двох елементів (або від

початкового значення та першого елемента даних), потім від отриманого акумулятора та наступного елемента даних (ітератора) тощо до завершення обчислення, вводять також праве (зворотне) згортання. У разі зворотного згортання функція обчислюється спочатку від початкового значення (якщо воно задане) і останнього елемента, потім від акумулятора і передостаннього елемента, і так далі, поки не дійдемо до обчислення від акумулятора першого елемента.

Ефективне розпаралелювання обчислення згортки від некомутативних неасоціативних функцій у загальному випадку неможливе. Зауважимо, що Гвідо ван Россум (BDFL, Benevolent Dictator For Life, довічний диктатор мови Python) свідомо перевів згортку із загального простору імен до модуля `functools`, оскільки в більшості застосунків її використовували або для таких тривіальних речей, як підсумовування масивів, або для отримання абсолютно незрозумілого нечитабельного коду.

Цей оператор відомий у багатьох мовах програмування під різними іменами, окрім `Map` використовується також назви `Select` і `Transform`.

У найпростішому випадку `Map` створює з переданої йому функції та впорядкованого набору даних інший впорядкований набір даних, у якому кожен елемент є значенням функції від відповідного елемента вихідного набору даних.

Існують розширення оператора `Map` на кілька наборів даних, при цьому функція має бути визначена від відповідної кількості елементів. Повертається набір даних, елементи якого є значеннями функції від відповідних наборів. При цьому є два способи визначити поведінку розширеного оператора `Map` у разі відмінності в розмірах переданих наборів даних - або результуючий набір має розмір, що дорівнює розміру найменшого набору з переданих, або розміру максимального, при цьому функції передаються невизначені значення.

Архітектура, що дає змогу проводити роботу з даними за допомогою еквівалентних алгоритмів одночасно в пакетному режимі і в реальному часі. Аналітика в реальному часі може бути неточною, виконується в оперативній пам'яті, але надаються швидко. Розрахунки в пакетному режимі забезпечують

зберігання отриманих результатів і видають достовірні дані, але виконуються довго.

Нині використовується значна кількість платформ і систем Великих даних. Системи обробки великих даних є фреймворками, тобто каркасами, для використання яких необхідно зістикувати їх з іншими фреймворками, прикладним програмним забезпеченням користувача і системою зберігання даних.

В аналітичному звіті Big Data Analytics Market Study, 2017 Edition наводиться така діаграма інфраструктур Великих даних, впроваджених на підприємствах, подана в розрізі розмірів підприємств (рис. 2.3).

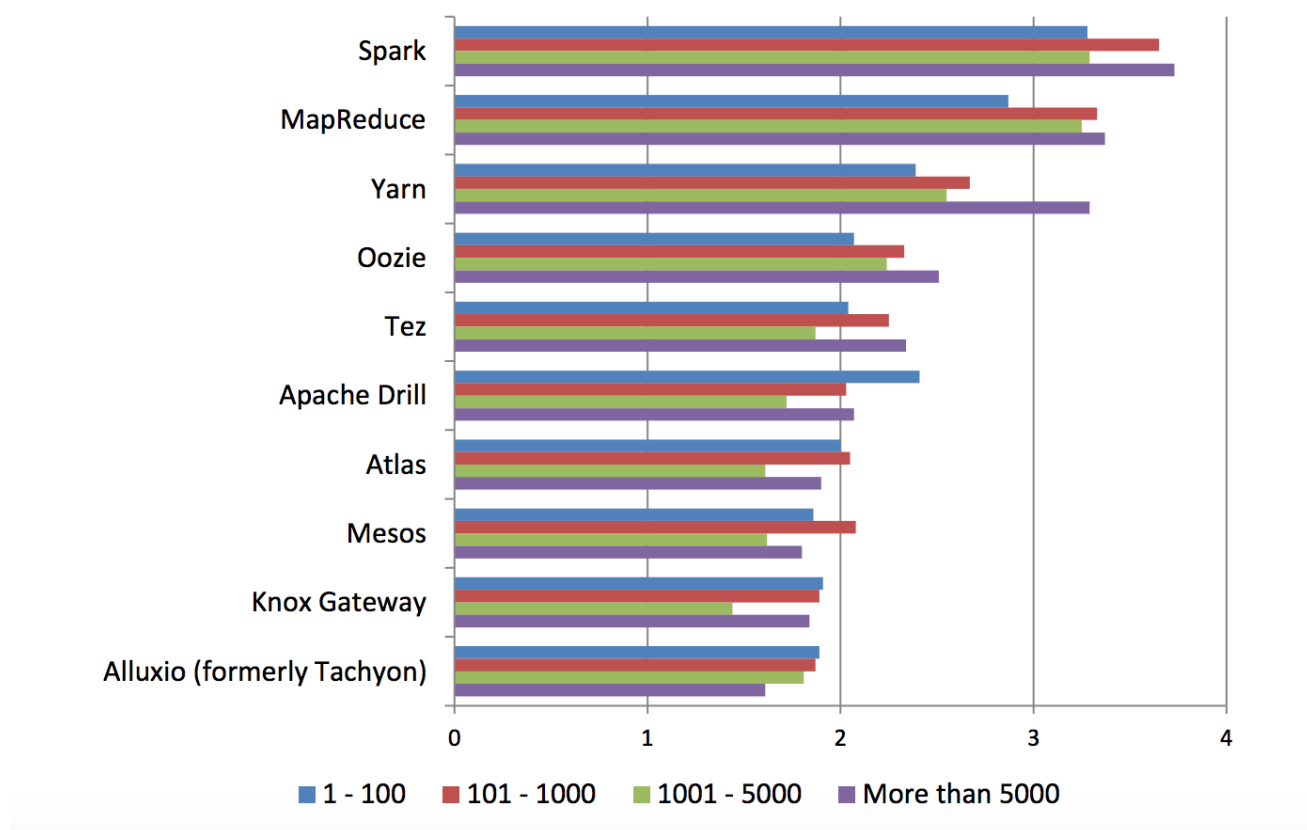


Рисунок 2.3 - Інфраструктури Великих даних у розрізі розміру підприємств

Більшість використовуваних платформ доступні за ліцензією Apache 2.0 і розташовані на сайті фонду програмного забезпечення Apache.

Комплект обладнання для обробки Великих даних монтується в ЦОД. Основними компонентами системи є система управління, обчислювальні ресурси, система зберігання даних, локальна мережа. Електроживлення, моніторинг,

доступ до інтернету та інші зовнішні ресурси надають центри обробки даних (ЦОД).

Система управління призначена для загального управління системою, зовнішнього доступу, забезпечення автентифікації, авторизації та акаунтингу і представлення результатів користувачам. Виконується на базі звичайної серверної платформи, специфічних вимог не має.

На рис. 2.4 представлено кластер Hadoop від Yahoo 10 річної давності.



Рисунок 2.4 - Кластер Hadoop від Yahoo 10 річної давності

Обчислювальні ресурси кластера складаються з вузлів (nodes), основними з яких є параметрами яких є обсяг оперативної пам'яті з контролем парності (ECC) і максимальна кількість ядер CPU. Розрахунковими параметрами є розмір оперативної пам'яті на ядро і швидкість роботи процесора. Висока відмовостійкість вузлів бажана, але загалом не потрібна, деяка кількість відмов є нормальною і компенсується за допомогою програмного забезпечення - відмовлений вузол автоматично виводять з експлуатації, і його роботу перерозподіляють на інші вузли.

У деяких випадках для опрацювання Великих даних, особливо під час використання нейронних мереж, використовують обчислювачі на базі GPU, аналогічні тим, що використовуються для HPC.

Розподілена система зберігання даних складається з дискових полиць, що забезпечують максимально швидкий доступ до даних. Резервування виконується за допомогою програмного забезпечення систем зберігання і загалом випадку не потрібно. Найчастіше також використовуються комп'ютери, в яких під'єднано велику кількість локальних дисків.



Рисунок 2.5 - Центр зберігання даних виробництва Titan Power

Мережева інфраструктура пов'язує обчислювальні ресурси, систему зберігання і систему управління в єдине ціле. Основна вимога до локальної мережі - низькі затримки. За великої кількості вузлів використовуються мережеві комутатори, що починають передачу пакета даних відразу після обробки заголовка пакета даних. За малої кількості вузлів поширене пряме з'єднання комп'ютерів за схемою гіперкуб, де розмірність куба відповідає числу портів на інтерфейсних платах. Раніше масово використовувалися мережі на базі різних варіантів інтерфейсу Infiniband, зараз, в основному, використовується 100 Gigabit Ethernet. Порівняно недавно з'явився 200 і 400 Gigabit Ethernet<sup>13</sup>.

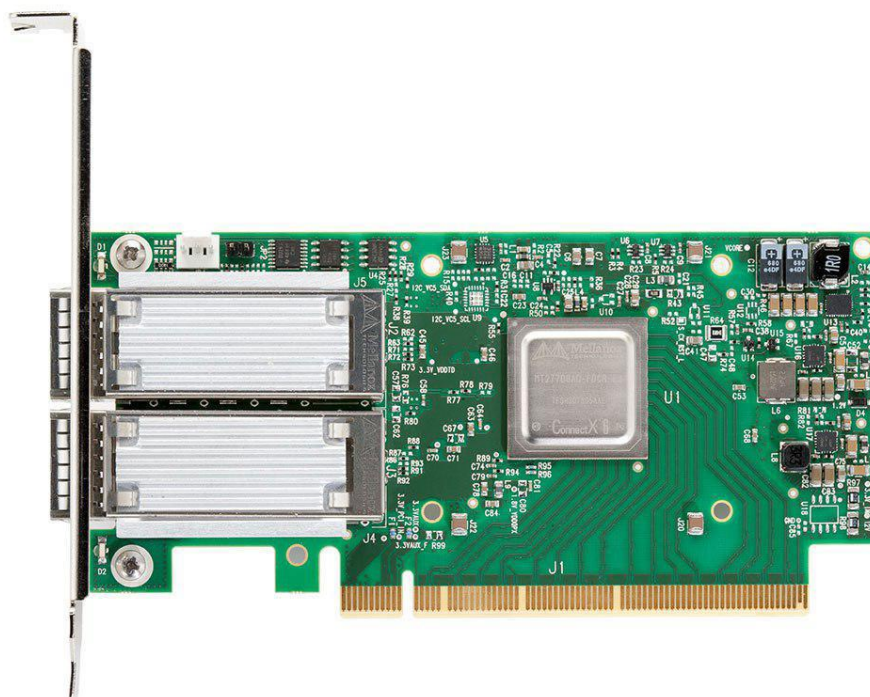


Рисунок 2.6 - ConnectX®-6 EN двопортовий мережевий адаптер 200Gb/s Ethernet

Великі дані обробляють у центрах обробки даних (ЦОД). У 2005 році асоціація телекомунікаційної індустрії (TIA) та інженерний комітет з телекомунікаційних кабельних систем (TR-42) випустили стандарт ANSI/TIA-942 на телекомунікаційну інфраструктуру центрів оброблення даних (Telecommunications Infrastructure Standards for Data Centres), який регламентує вимоги до інфраструктури ЦОД.

Поточна версія стандарту TIA-942-A встановлює вимоги до мережевої інфраструктури, організації електропостачання, файлового зберігання, збереження резервних копій архівування, відмовостійкості систем, контролю доступу до мережі та мережевої безпеки, управління базами даних, веб-хостингу, хостингу додатків, розподілу контенту, управління кліматом, захисту від зовнішніх фізичних руйнувань, як-от пожежі, повені, погодним явищам, управління енергоживленням.

При побудові датацентрів також застосовуються стандарти ANSI/TIA/EIA-568-B.1, ANSI/TIA/EIA-568-B.2, ANSI/TIA/EIA-568-B.3, ANSI/TIA-569-B, ANSI/TIA/EIA-606-A, ANSI/TIA/EIA-J-STD-607, ANSI/TIA-758-A, національні правила з безпечного влаштування електроустановок NESC IEEE C2, національний електричний код NEC NFPA 70 (у РФ їхню роль відіграють правила влаштування електроустановок 6 і 7 редакції, ПУЕ-6 і ПУЕ-7), захист ІТ-обладнання, стандарт NFPA 75, інженерні вимоги до універсальної телекомунікаційної стійки ANSI T1.336, рекомендовані правила заживлення та заземлення електронного обладнання IEEE 1100, рекомендовані правила для систем аварійного та резервного енергопостачання промислового та комерційного застосування IEEE 446, технічні вимоги Telcordia GR-63- CORE (NEBS) та GR-139-CORE та інші галузеві стандарти.

У даному розділі були розглянуті основні поняття і технології, пов'язані з Великими даними. Обробка Великих даних є складним технологічним процесом, що вимагає глибоких програмно-інженерних знань для розроблення моделі даних, вибору відповідних програмно-апаратних засобів і оцінки сукупної вартості управління даними. У багатьох випадках обробка даних може бути проведена досить скромними коштами, за допомогою оренди систем зберігання та обробки в хмарному середовищі, в інших випадках потрібна оренда або навіть будівництво власного ЦОД і установка власного обладнання, в третій випадках - вартість роботи з даними може перевищити дохід від їх обробки і обробка даних своїми силами недоцільна, проте може бути виконана за допомогою підрядника.

### 3 ПРОЕКТУВАННЯ БАГАТОРІВНЕВОЇ МОДЕЛІ ЦОД

У цьому розділі наведено детальну інформацію про багаторівневу структуру, яку Cisco рекомендує для центрів обробки даних. Багаторівнева модель проектування підтримує безліч архітектур веб-сервісів, в тому числі заснованих на Microsoft.NET і Java 2 Enterprise Edition. Ці середовища додатків веб-сервісів використовуються для поширених ERP-рішень, таких як PeopleSoft, Oracle, SAP, BAAN і JD Edwards; і CRM-рішення від таких вендорів, як Siebel і Oracle.

Багаторівнева модель спирається на багаторівневу мережеву архітектуру, що складається з ядра, агрегації та рівнів доступу, як показано на рисунку 3.1. У цьому розділі більш детально описані апаратні та конструктивні рекомендації для кожного з цих рівнів. Включені такі основні теми:

- огляд багаторівневого проектування центрів обробки даних;
- основний рівень центру обробки даних;
- рівень агрегації центру обробки даних;
- рівень доступу до центру обробки даних;
- рівень послуг центрів обробки даних.

Багаторівнева модель є найбільш поширеною моделлю, яка використовується на підприємстві на сьогоднішній день. Ця модель складається в переважно з рівнів веб-серверів, додатків і баз даних, що працюють на різних платформах, включаючи блейд-сервери, сервери з однією стійкою (1RU) і мейнфрейми.

На рисунку 3.1 показана топологію багаторівневої моделі центру обробки даних.



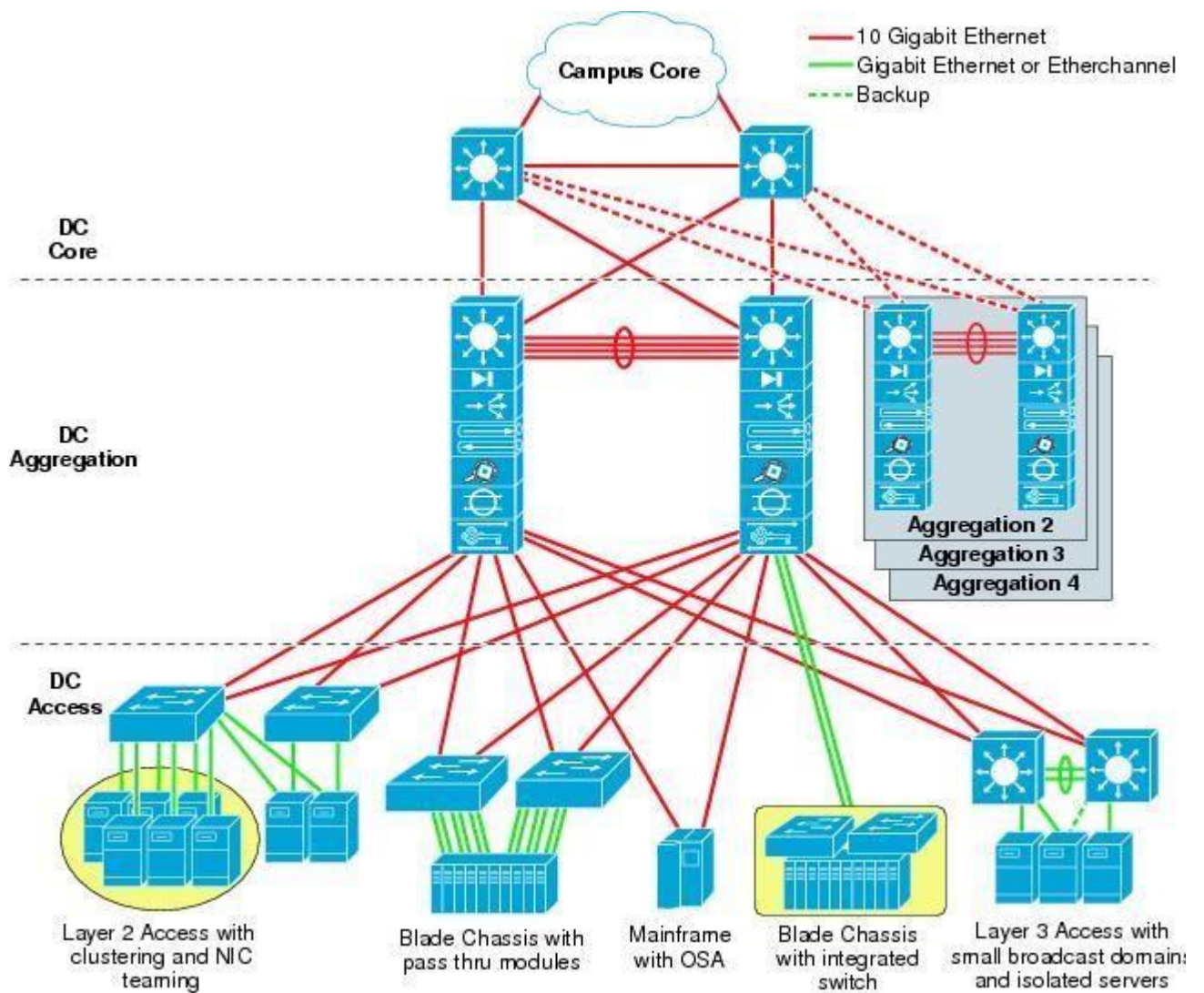


Рисунок 3.1 - Топологія багаторівневої моделі центру обробки даних

Рівень ядра центру обробки даних забезпечує високошвидкісну комутацію пакетів між декількома модулями агрегації. Цей рівень слугує шлюзом до ядра кампусу, до якого підключаються інші модулі, включаючи, наприклад, екстранет, глобальну мережу та периферію Інтернету. Всі лінії, що з'єднують ядро центру обробки даних, закінчуються на рівні 3 і зазвичай використовують інтерфейси 10 GigE для підтримки високого рівня пропускної здатності, продуктивності і відповідності рівням надлишкової підписки.

Ядро центру обробки даних відрізняється від ядра кампусу, оскільки має іншу мету та обов'язки. Ядро центру обробки даних не є обов'язковим, але рекомендується, якщо використовується декілька модулів агрегації для забезпечення масштабованості. Навіть якщо використовується невелика кількість

модулів агрегації, може бути доцільно використовувати ядро кампусу для з'єднання структури центру обробки даних.

Визначаючи, чи варто впроваджувати ядро центру обробки даних, треба врахувати наступне:

- адміністративні домени і політики - окремі ядра допомагають ізолювати рівні розподілу кампусу і рівні агрегації ЦОД з точки зору адміністрування і політик, таких як QoS, списки доступу, усунення несправностей і обслуговування;

- щільність портів 10 Gigabit - одна пара основних комутаторів може не підтримувати кількість портів 10 Gigabit, необхідних для підключення розподільчого рівня кампусу, а також комутаторів рівня агрегації центру обробки даних;

- майбутні очікування - вплив на бізнес впровадження окремого рівня ядра центру обробки даних на більш пізній термін може зробити доцільним його впровадження на початковому етапі реалізації.

У великому центрі обробки даних одна пара комутаторів ядра центру обробки даних зазвичай з'єднує кілька модулів агрегації за допомогою інтерфейсів 10 GigE Layer 3.

Рекомендованою платформою для ядра корпоративного центру обробки даних є Cisco Catalyst 6509 з процесорним модулем Sup720. Висока швидкість комутації, велика комутаційна структура і щільність портів 10 GigE роблять Catalyst 6509 ідеальним рішенням для цього рівня. Велика кількість портів 10 GigE необхідна для підтримки декількох модулів агрегації. Catalyst 6509 може підтримувати модулі 10 GigE у всіх позиціях, оскільки кожен слот підтримує два канали до комутаційної матриці (Catalyst 6513 не може підтримувати цю функцію). Ми не рекомендуємо використовувати в базовому шарі модулі, що не кріпляться до тканини (класичні).

Ядро дата-центру з'єднане з ядром кампусу та рівнем агрегації з резервуванням за допомогою каналів 10 GigE 3-го рівня. Це забезпечує повністю надлишкову архітектуру і усуває єдиний вузол ядра як єдину точку відмови. Це також дозволяє розгортати вузли ядра лише з одним модулем супервізора.

Лінійні карти Cisco серії 6700 підтримують додатковий модуль дочірньої карти, який називається розподіленою картою переадресації (Distributed Forwarding Card, DFC). DFC дозволяє приймати рішення про локальну маршрутизацію на кожній лінійній карті через локальну інформаційну базу переадресації (FIB). Таблиця FIB на картці функцій політики Sup720 (PFC) підтримує синхронізацію з кожною таблицею FIB DFC на лінійних картах, щоб забезпечити точну цілісність маршрутизації по всій системі. Без карти DFC компактний пошук заголовка повинен бути відправлений на PFC на Sup720, щоб визначити, куди в комутаційній структурі переслати кожен пакет, щоб він досяг місця призначення. Це відбувається як для пакетів 2-го, так і для 3-го рівнів комутації. При наявності DFC лінійна карта може перемикає пакет безпосередньо через комутаційну структуру на лінійну карту призначення, не звертаючись до таблиці FIB Sup720 на PFC. Різниця в продуктивності може становити від 30 Мбіт/с для всієї системи до 48 Мбіт/с на слот з DFC.

З DFC або без них доступна пропускна здатність системи така ж, як і в комутаторі Sup720. У таблиці 2-1 наведені дані про пропускну здатність і смугу пропускання для модулів з підтримкою DFC і старіших модулів CEF256, а також класичних шинних модулів для порівняння.

Таблиця 3.1 - Порівняння продуктивності з розподіленою переадресацією

Конфігурація системи з Sup720	Пропускна спроможність в Мбіт/с	Пропускна здатність в Гбіт/с
Модулі серії CEF720 (6748, 6704, 6724)	До 30 Мбіт/с на систему	2 x 20 Гбіт/с (виділено на кожен слот) (6724=1 x 20 Гбіт/с)
Модулі серії CEF720 з DFC3 (6704 з DFC3, 6708 з DFC3, 6724 з DFC3)	Підтримує до 48 Мбіт/с (на слот)	2x 20 Гбіт/с (виділено на один слот) (6724=1 x 20 Гбіт/с)
Модулі серії CEF256 (FWSM, SSLSM, NAM-2, IDSM-2, 6516)	До 30 Мбіт/с (на систему)	1x 8 Гбіт/с (виділено на один слот)
Модулі серії Classic (CSM, 61xx-64xx)	До 15 Мбіт/с (на систему)	Спільна шина 16 Гбіт/с (класична шина)

Використання DFC на рівні ядра багаторівневої моделі не є обов'язковим. Аналіз потоків сеансів додатків, які можуть проходити через ядро, допомагає визначити максимальні вимоги до пропускної здатності та доцільність використання DFC. Якщо використовується декілька модулів агрегації, існує велика ймовірність того, що велика кількість потоків сеансів буде поширюватися між рівнями серверів. Взагалі кажучи, базовий рівень виграє завдяки меншій затримці та вищій загальній швидкості пересилання, якщо включити DFC на лінійних картах.

### **3.1 Потік трафіку в ядрі дата-центру**

Основний рівень з'єднується з кампусним і агрегаційним рівнями за допомогою кінцевих каналів 10 GigE третього рівня. З'єднання 3-го рівня необхідні для досягнення масштабованості пропускної здатності, швидкої конвергенції та уникнення блокування каналів або ризику неконтрольованої трансляції, пов'язаної з розширенням доменів 2-го рівня.

Потік трафіку в ядрі складається переважно з сеансів, що проходять між ядром кампусу та модулями агрегації. Ядро агрегує потоки трафіку модулів агрегації на оптимальні шляхи до ядра кампусу, як показано на рисунку 3.2. Трафік між серверами зазвичай залишається в межах модуля агрегації, але трафік резервного копіювання та реплікації може переміщатися між модулями агрегації через ядро.

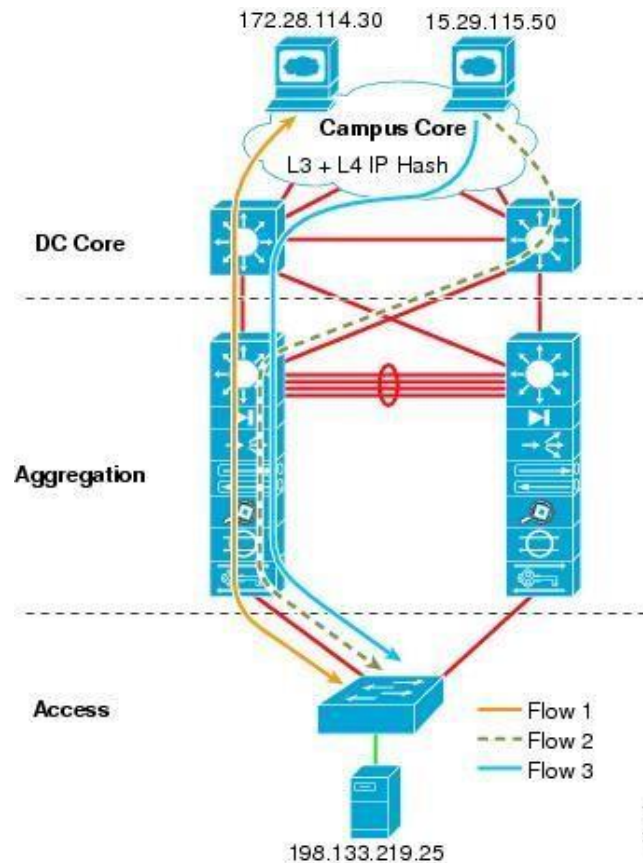


Рисунок 3.2 - Потік трафіку через основний рівень

Як показано на рисунку 3.2, на вибір шляху може впливати наявність сервісних модулів і топологія рівня доступу, що використовується. Маршрутизацію від ядра до рівня агрегації можна налаштувати так, щоб весь трафік надходив до певного вузла агрегації, де розташовані основні сервісні модулі.

З точки зору ядра кампусу, існує щонайменше два маршрути до серверних підмереж з однаковою вартістю, що дозволяє ядру розподіляти навантаження на кожен комутатор агрегації в конкретному модулі. За замовчуванням це виконується за допомогою балансування навантаження на основі CEF на основі хешування IP-адрес джерела/призначення на рівні 3. Також можна використовувати алгоритми хешування CEF-балансування навантаження на основі IP-адрес 3-го рівня та портів 4-го рівня. Це зазвичай покращує розподіл навантаження, оскільки надає алгоритму хешування більше унікальних значень.

Щоб увімкнути алгоритм хешування CEF на основі Layer 3 плюс Layer 4,

необхідно скористатися наступною командою на глобальному рівні:

```
CORE1(config)#mls ip cef load full
```

Більшість IP-стеків використовують автоматичну рандомізацію номерів вихідних портів, що сприяє кращому розподілу навантаження. Іноді, з політичних або інших причин, номери портів переводяться брандмауерами, балансувальниками навантаження або іншими пристроями. Необхідно завжди тестувати певний алгоритм хешування, перш ніж впроваджувати його у виробничій мережі.

### **3.2 Рівень агрегації центру обробки даних**

Рівень агрегації, до якого підключено багато висхідних каналів рівня доступу, несе основну відповідальність за об'єднання тисяч сеансів, що виходять і входять в центр обробки даних. Агрегаційні комутатори повинні підтримувати безліч з'єднань 10 GigE і GigE, забезпечуючи при цьому високошвидкісну комутаційну структуру з високою швидкістю переадресації. Рівень агрегації також надає додаткові послуги, такі як балансування навантаження на сервер, брандмауер і розвантаження SSL для серверів на комутаторах рівня доступу.

Комутатори рівня агрегації несуть навантаження з обробки основного дерева та обробки протоколів резервування шлюзів за замовчуванням. Рівень агрегації може бути найбільш критичним рівнем в центрі обробки даних, оскільки щільність портів, надлишкові значення підписки, процесорна обробка і сервісні модулі вносять унікальні зміни в загальну конструкцію.

Корпоративний центр обробки даних містить принаймні один модуль агрегації, який складається з двох комутаторів рівня агрегації. Пари комутаторів рівня агрегації працюють разом, щоб забезпечити надмірність і підтримувати стан сеансу, надаючи при цьому цінні послуги на рівні доступу.

Рекомендованими платформами для рівня агрегації є комутатори Cisco Catalyst 6509 і Catalyst 6513, оснащені процесорними модулями Sup720. Важливими вимогами до рівня агрегації є висока швидкість комутації, велика

структура комутатора і здатність підтримувати велику кількість портів 10 GigE. Рівень агрегації також повинен підтримувати пристрої і сервіси безпеки та прикладні пристрої і сервіси, в тому числі наступні:

- сервісні модулі брандмауера Cisco (FWSM);
- Cisco Application Control Engine (ACE);
- виявлення вторгнень;
- модуль мережевого аналізу (NAM);
- розподілений захист від атак на відмову в обслуговуванні (Guard).

Хоча Cisco Catalyst 6513 може здатися гарним рішенням для рівня агрегації через велику кількість слотів, зверніть увагу, що він підтримує суміш одноканальних і двоканальних слотів. Слоти з 1 по 8 є одноканальними, а слоти з 9 по 13 - двоканальними (рисунок 3.3).

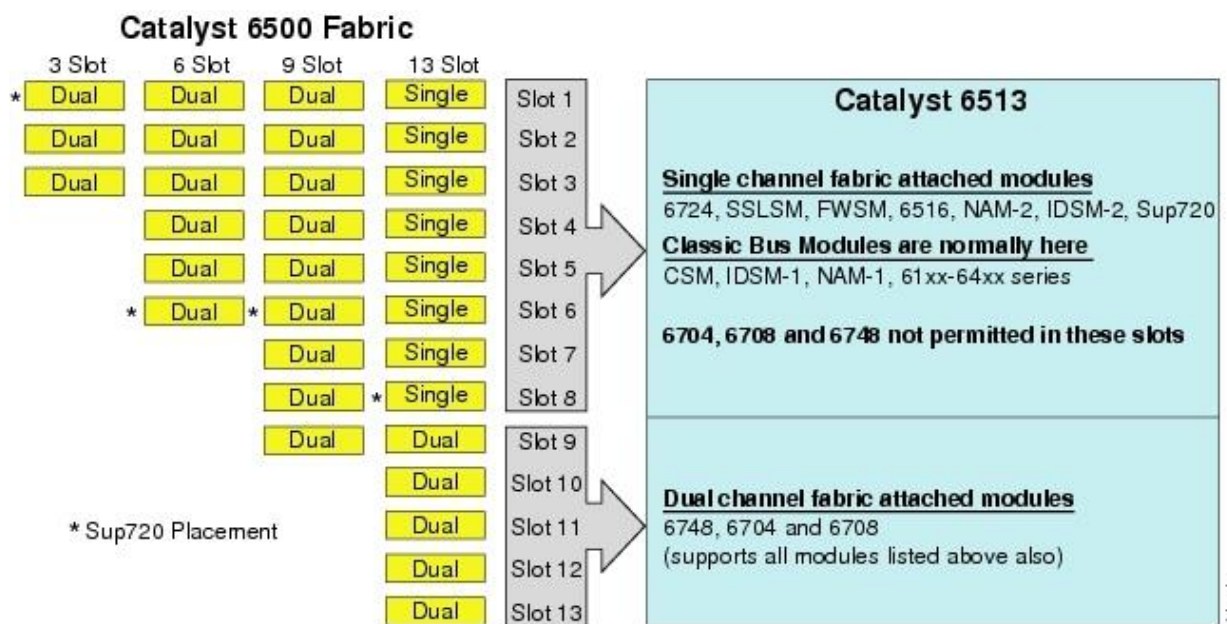


Рисунок 3.3 - Канали комутаційної мережі Catalyst 6500 по шасі та слотам

Двоканальні лінійні карти, такі як 6704-10 GigE, 6708-10G або 6748-SFP (TX), можна розмістити в слотах 9-13. Одноканальні лінійні карти, такі як 6724-SFP, а також старі одноканальні або класичні шинні лінійні карти можна використовувати і найкраще підходять для слотів 1-8, але також можна використовувати і в слотах 9-13. На відміну від Catalyst 6513, Catalyst 6509 має менше доступних слотів, але він може підтримувати двоканальні модулі в

кожному слоті.

Двоканальний слот підтримує всі типи модулів (CEF720, CEF256 і класичну шину). Одноканальний слот може підтримувати всі модулі, за винятком двоканальних карт, які наразі включають лінійні карти 6704, 6708 і 6748.

Вибір між Cisco Catalyst 6509 або 6513 можна зробити, проаналізувавши наступні вимоги:

- Cisco Catalyst 6509 - коли рівень агрегації вимагає багато каналів 10 GigE з невеликою кількістю сервісних модулів або без них і дуже високої продуктивності;

- Cisco Catalyst 6513 - коли рівень агрегації вимагає невеликої кількості 10 GigE каналів з великою кількістю сервісних модулів.

Якщо на рівні агрегації потрібна велика кількість сервісних модулів, комутатор рівня послуг може допомогти оптимізувати використання слотів рівня агрегації та продуктивність.

Інші міркування пов'язані з повітряним охолодженням і використанням простору шафи. Catalyst 6509 можна замовити в шасі, сумісному з NEBS, яке забезпечує вентиляцію повітря спереду назад, що може знадобитися в певних конфігураціях центрів обробки даних. Версію Cisco Catalyst 6509 NEBS також можна встановлювати у два блоки в одній шафі центру обробки даних, що дозволяє більш ефективно використовувати простір.

Використання DFC на рівні агрегації багаторівневої моделі не є обов'язковим. Аналіз потоків сеансів додатків, які можуть проходити через рівень агрегації, допомагає визначити максимальні вимоги до переадресації та доцільність використання DFC. Наприклад, якщо рівні серверів через комутатори рівня доступу призводять до великого обсягу міжпроцесного обміну даними (IPC) між ними, рівень агрегації може виграти від використання DFC. Взагалі кажучи, рівень агрегації виграє за рахунок меншої затримки і вищої загальної швидкості переадресації при включенні DFC на лінійних картах.



### 3.3 Потік трафіку на рівні агрегації ЦОД

Рівень агрегації з'єднується з основним рівнем за допомогою кінцевих каналів 10 GigE третього рівня. З'єднання 3-го рівня необхідні для досягнення масштабованості пропускну здатності, швидкої конвергенції, а також для уникнення блокування шляху або ризику неконтрольованої трансляції, пов'язаної з транскордонними доменами 2-го рівня.

Трафік на рівні агрегації в основному складається з наступних потоків:

- від рівня ядра до рівня доступу - потоки трафіку від ядра до рівня доступу зазвичай пов'язані з клієнтськими HTTP-запитами до ферми веб-серверів. До підмереж веб-серверів існує щонайменше два маршрути з однаковою вартістю. Алгоритм хешування L3 плюс L4 на основі CEF визначає, як сеанси розподіляються між рівноцінними маршрутами. Спочатку веб-сесії можуть бути спрямовані на VIP-адресу, яка знаходиться на балансувальнику навантаження на рівні агрегації, або відправлені безпосередньо на серверну ферму. Після того, як клієнтський запит пройде через балансувальник навантаження, він може бути перенаправлений на модуль розвантаження SSL або прозорий брандмауер, перш ніж продовжити роботу на реальному сервері, розташованому на рівні доступу;

- від рівня доступу до рівня доступу - модуль агрегації є основним транспортом для трафіку між серверами на рівні доступу. Сюди входить багаторівневий трафік між серверами (веб-додаток або додаток до бази даних) та інші типи трафіку, зокрема трафік резервного копіювання або реплікації. Сервісні модулі на рівні агрегації дозволяють трафіку між серверами використовувати балансувальники навантаження, розвантажувачі SSL і служби брандмауерів для покращення масштабованості та безпеки серверної ферми.

Вибір шляху, що використовується для різних потоків, залежить від різних вимог до проектування. Ці відмінності залежать, в першу чергу, від наявності сервісних модулів і топології рівня доступу, що використовується.

### 3.4 Вибір шляху за наявності сервісних модулів

Коли сервісні модулі використовуються в активно-резервному режимі, вони розміщуються на обох комутаторах рівня агрегації з резервуванням, з первинними активними сервісними модулями на комутаторі рівня агрегації 1 і вторинними резервними сервісними модулями на комутаторі рівня агрегації 2, як показано на рисунку 3.4.

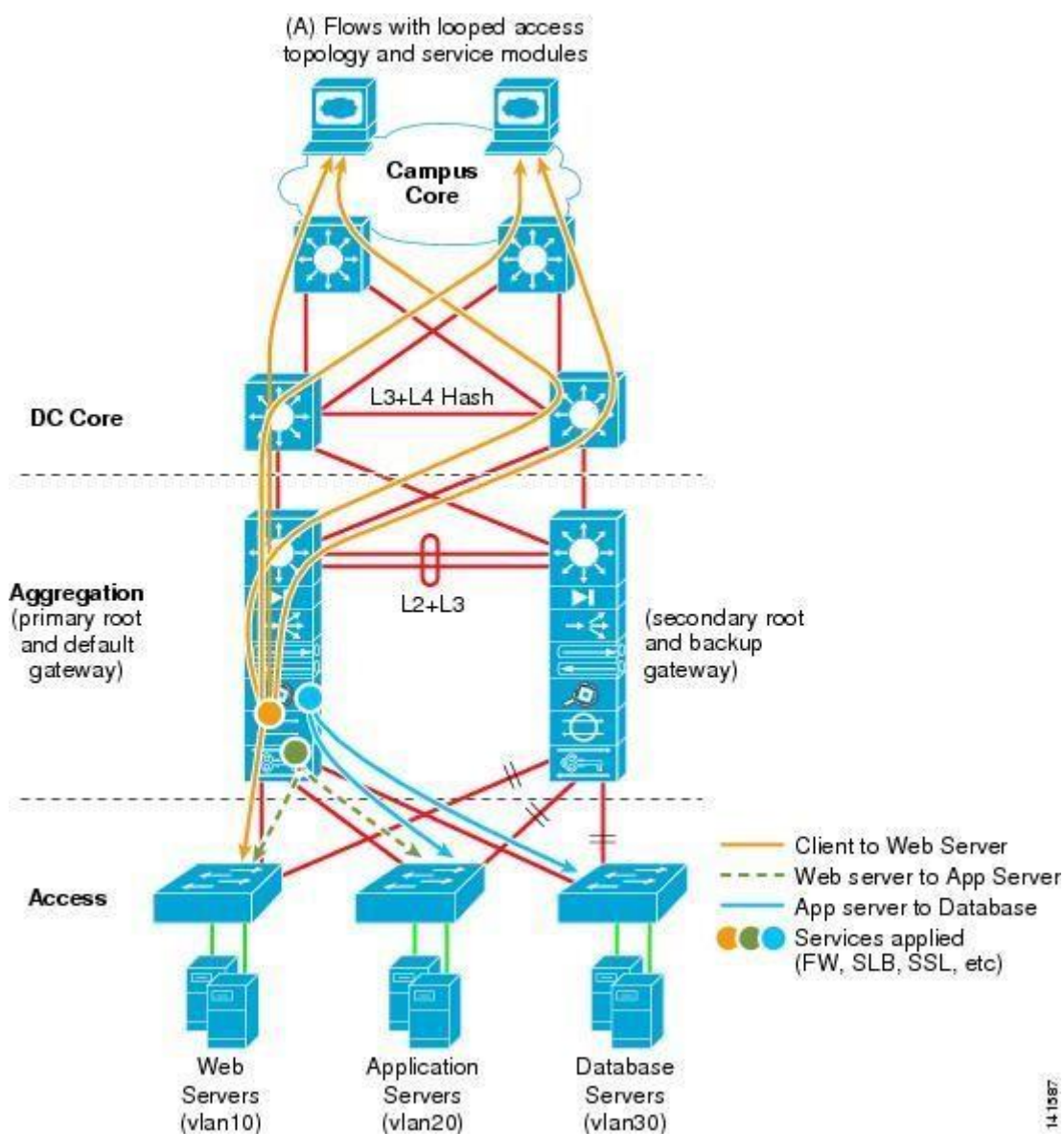


Рисунок 3.4 - Потік трафіку з сервісними модулями в топології закільцьованого доступу

У схемі з сервісними модулями може знадобитися налаштувати протокол маршрутизації так, щоб основний шлях трафіку був спрямований до активних сервісних модулів на комутаторі Aggregation 1, а в разі збою - до резервних сервісних модулів на комутаторі Aggregation 2. Це забезпечує конструкцію з передбачуваною поведінкою і шаблонами трафіку, що полегшує пошук і усунення несправностей. Крім того, завдяки вирівнюванню всіх активних сервісних модулів в одному комутаторі, потоки між сервісними модулями залишаються на локальній комутаційній шині, не перетинаючи магістраль між комутаторами агрегації.

Без налаштування маршрутів ядро має два маршрути до підмережі серверної ферми з однаковою вартістю, тому сеанси розподіляються по каналах до обох комутаторів рівня агрегації. Оскільки агрегація 1 містить активні сервісні модулі, 50 відсотків сеансів без потреби проходять через міжкомутаційне з'єднання між агрегацією 1 і агрегацією 2. Налаштувавши конфігурацію маршрутизації, сеанси можуть залишатися на симетричних шляхах у передбачуваний спосіб. Налаштування маршрутів також допомагає в певних сценаріях збоїв, які створюють сценарії активно-активних сервісних модулів.

### **3.4.1 Потік трафіку серверної ферми з сервісними модулями**

Потоки трафіку в серверній фермі складаються переважно з багаторівневих зв'язків, зокрема клієнт-мережа, мережа-до-додаток і додаток-до-бази даних. Інші типи трафіку, які можуть існувати, включають доступ до сховища (NAS або iSCSI), резервне копіювання та реплікацію.

Як описано в попередньому розділі, рекомендовано вирівнювати активні сервіси на спільному комутаторі рівня агрегації. Це утримує потоки сеансів на одній високошвидкісній шині, забезпечуючи передбачувану поведінку і спрощуючи пошук і усунення несправностей. Закільцьована топологія рівня доступу, як показано на рисунку 3.4, забезпечує перевірену модель для підтримки реалізації активного/резервного сервісного модуля. Шляхом вирівнювання служб

первинного кореня остовного дерева і первинного шлюзу за замовчуванням HSRP на комутаторі Aggregation 1 створюється симетричний шлях трафіку.

Якщо в парі комутаторів агрегації використовується кілька пар сервісних модулів, можна розподілити активні сервіси, що дозволяє використовувати обидва висхідні канали рівня доступу. Однак це зазвичай не є життєздатним рішенням через необхідність використання додаткових сервісних модулів. Майбутні можливості "активний-активний" повинні дозволити такий розподіл без необхідності в додаткових сервісних модулях.

Сервісні модулі CSM та FWSM-2.x наразі працюють в активному/резервному режимах. Ці пари модулів вимагають ідентичних конфігурацій. Конструкція рівня доступу повинна гарантувати, що шляхи з'єднання залишаються симетричними до активного сервісного модуля. Cisco Application Control Engine (ACE) - це новий модуль, який впроваджує кілька удосконалень щодо балансування навантаження та послуг безпеки. Ключовою відмінністю між CSM, FWSM версії 2.x і Cisco ACE є можливість підтримки активно-активних контекстів у модулі агрегації з обходом відмови для кожного контексту.

### **3.4.2 Потік трафіку серверної ферми без сервісних модулів**

Якщо сервісні модулі не використовуються в комутаторах рівня агрегації, можна використовувати кілька топологій рівня доступу. На рисунку 3.5 показані потоки трафіку з топологіями з петлею і без петлі.

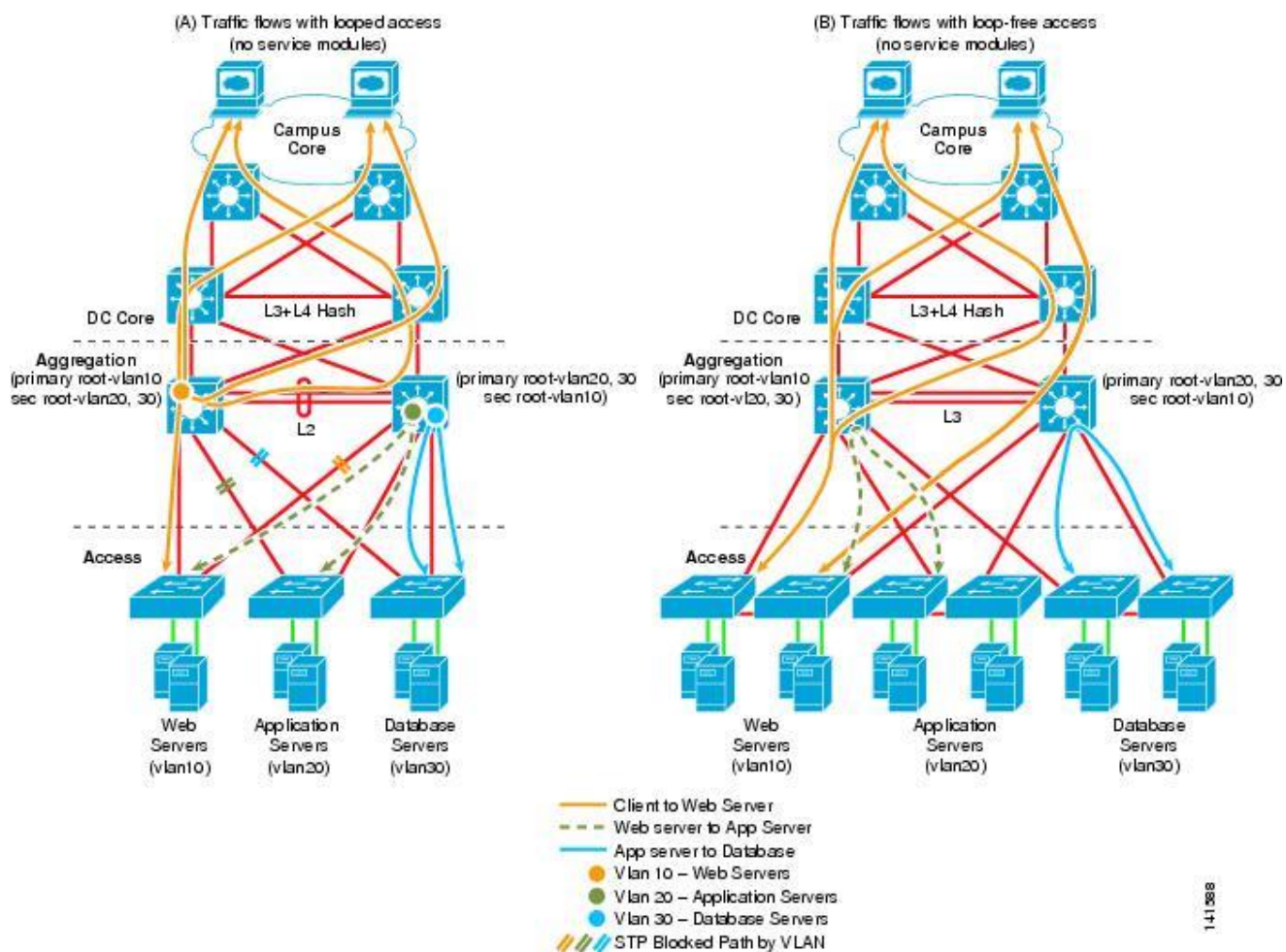


Рисунок 3.5 - Потік трафіку без сервісних модулів

Якщо сервісні модулі відсутні, можна розподілити кореневий шлюз і шлюз за замовчуванням HSRP між комутаторами агрегації, як показано на рисунку 3.5. Це дозволяє збалансувати трафік як між комутаторами агрегації, так і між висхідними лініями рівня доступу.

### 3.5 Масштабування рівня агрегації

Конструкція рівня агрегації має вирішальне значення для стабільності та масштабованості всієї архітектури центру обробки даних. Весь трафік, що входить і виходить з центру обробки даних, не тільки проходить через рівень агрегації, але й залежить від сервісів, вибору шляхів і надлишкової архітектури, вбудованих в конструкцію рівня агрегації. У цьому розділі описані наступні

чотири області, які мають вирішальне значення і впливають на проектування рівня агрегації:

- розмір домену несправності 2-го рівня;
- масштабованість основного дерева;
- щільність 10 GigE;
- масштабування резервування шлюзу за замовчуванням (HSRP).

Рівень агрегації складається з пар з'єднаних між собою комутаторів агрегації, які називаються модулями. На рисунку 3.6 показано конструкцію з декількох модулів агрегації, що використовують спільне ядро.

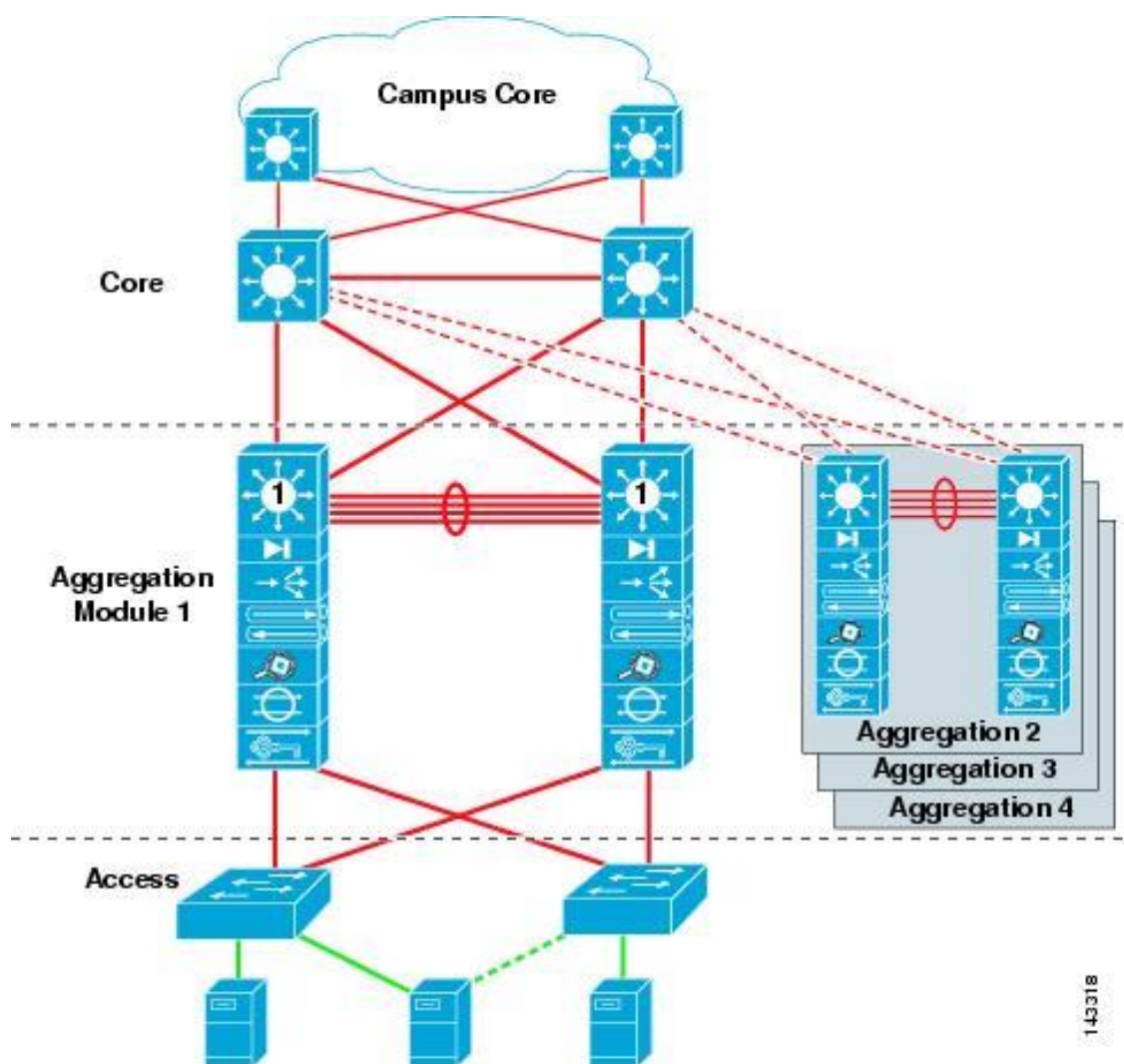


Рисунок 3.6 - Кілька модулів агрегації

Використання модулів агрегації допомагає вирішити проблеми масштабування, пов'язані з чотирма перерахованими вище сферами.

### 3.6 Розмір домену несправності рівня 2

Оскільки домени рівня 2 продовжують збільшуватися в розмірах через кластеризацію, об'єднання мережевих карт та інші вимоги додатків, діаметри рівня 2 масштабуються ще більше, ніж будь-коли раніше. Рівень агрегації несе найбільше навантаження в цьому відношенні, оскільки він встановлює розмір домену рівня 2 і керує ним за допомогою протоколу основного дерева, такого як Rapid-PVST+ або MST.

Першою проблемою, пов'язаною з великими діаметрами Layer 2, є розмір домену збоїв. Незважаючи на те, що технології продовжують покращувати надійність і стабільність доменів рівня 2, все ще залишається певний рівень вразливості до ширококомовних штормів, які можуть бути спричинені несправним обладнанням або людськими помилками. Через наявність петлі всі канали не можуть постійно перебувати в стані переадресації, оскільки ширококомовні/багатоадресні пакети будуть рухатися по нескінченній петлі, повністю насичуючи VLAN, що негативно вплине на продуктивність мережі. Протокол основного дерева, такий як Rapid PVST+ або MST, необхідний для автоматичного блокування певного каналу і розриву петлі.

Великі центри обробки даних повинні розглянути можливість встановлення максимального розміру домену рівня 2, щоб визначити максимальний рівень вразливості до цієї проблеми. Використовуючи кілька модулів агрегації, розмір домену рівня 2 можна обмежити; таким чином, можна заздалегідь визначити вразливість до збоїв. Багато клієнтів використовують значення "максимальної кількості серверів" для визначення максимального домену збоїв на рівні 2.

### 3.7 Масштабованість основного дерева

Розширення віртуальних локальних мереж в центрі обробки даних необхідно не тільки для задоволення вимог додатків, таких як суміжність на рівні 2, але і для забезпечення високого рівня гнучкості в адмініструванні серверів.

Багатьом клієнтам потрібна можливість групувати і обслуговувати сервери відділів разом в загальному адресному просторі VLAN або IP-підмережі. Це полегшує управління середовищем центру обробки даних щодо додавання, переміщення та змін.

При використанні петлевої топології 2-го рівня необхідний механізм захисту від петель, такий як протокол Spanning Tree. Огинаюче дерево автоматично розриває петлі, запобігаючи безперервній циркуляції широкомовних пакетів і руйнуванню мережі. Протоколи покривного дерева, рекомендовані при проектуванні центрів обробки даних, - це 802.1w-Rapid PVST+ і 802.1s-MST. І 802.1w, і 802.1s мають однакові характеристики швидкої конвергенції, але відрізняються гнучкістю та роботою.

На рівень агрегації припадає основне навантаження, пов'язане з обробкою остовного дерева. Кількість віртуальних локальних мереж і межі їх розширення безпосередньо впливають на масштабованість і збіжність остовного дерева. Впровадження модулів агрегації допомагає розподілити і масштабувати обробку остовного дерева.

Оскільки вимоги до пропускної здатності та інтерфейсу сервера на рівні доступу зростають, висхідні канали до рівня агрегації виходять за межі швидкостей GigE або гігабітного EtherChannel і переходять на 10 GigE. Очікується, що ця тенденція буде посилюватися і може створити проблему щільності в існуючих або нових конструкціях рівня агрегації. Хоча в довгостроковій перспективі рішенням можуть стати лінійні карти 10 GigE з більшою щільністю і більші комутатори, поточним перевіреним рішенням є використання декількох модулів агрегації.

В даний час максимальна кількість портів 10 GigE, які можна розмістити в комутаторі рівня агрегації, становить 64 при використанні лінійної карти WS-X6708-10G-3C в Catalyst 6509. Однак, з урахуванням брандмауера, балансувальника навантаження, мережевого аналізу та інших модулів, пов'язаних зі службами, це число, як правило, є меншим. Використання рівня ядра центру обробки даних і впровадження декількох модулів агрегації забезпечує більш



високий рівень щільності 10 GigE.

Також важливо розуміти потік трафіку в центрі обробки даних при розгортанні цих модулів 10 GigE підвищеної щільності через їх перенасиченість підпискою.

Конструкція рівня доступу також може впливати на щільність 10 GigE, що використовується на рівні агрегації. Наприклад, топологія квадратної петлі дозволяє використовувати вдвічі більше комутаторів рівня доступу, ніж топологія трикутної петлі.

### **3.8 Резервування шлюзу за замовчуванням з HSRP**

Рівень агрегації надає адресу "шлюзу за замовчуванням" первинного і вторинного маршрутизатора для всіх серверів на всьому рівні доступу, використовуючи протоколи резервування шлюзу за замовчуванням HSRP, VRRP або GLBP. Це можна застосувати лише до серверів з топологією доступу 2-го рівня. Процесор на модулях Sup720 в обох комутаторах агрегації несе навантаження на обробку даних для підтримки цієї необхідної функції. Навантаження на процесор лінійно залежить від конфігурації таймера оновлення і кількості VLAN, які поширюються на весь рівень доступу, що підтримується цим модулем агрегації, оскільки між ними підтримується стан кожного активного шлюзу за замовчуванням. У разі виходу з ладу обладнання або середовища агрегації один процесор повинен стати основним шлюзом за замовчуванням для кожної налаштованої VLAN.

HSRP є найбільш широко використовуваним протоколом для резервування шлюзу за замовчуванням. HSRP надає найбагатший набір функцій і гнучкість для підтримки декількох груп, регульованих таймерів, відстеження і великої кількості екземплярів. Поточні результати тестування рекомендують обмежити максимальну кількість екземплярів HSRP в модулі агрегації до ~ 500, з рекомендованими таймерами односекундного привітання і трисекундного утримання.

Слід враховувати інші процеси, що керуються перериваннями процесора, які можуть бути запущені на комутаторі рівня агрегації (наприклад, тунелювання та SNMP-опитування), оскільки вони можуть зменшити це значення ще більше. Якщо потрібно більше екземплярів HSRP, ми рекомендуємо розподілити це навантаження між кількома комутаторами модулів агрегації.

### **3.9 Рівень доступу до центру обробки даних**

Рівень доступу забезпечує фізичний рівень підключення до ресурсів сервера і працює в режимах 2-го або 3-го рівня. Цей режим відіграє вирішальну роль у виконанні певних вимог до сервера, таких як об'єднання мережевих карт, кластеризація та стримування ширококомовлення. Рівень доступу є першою точкою надлишкової підписки в центрі обробки даних, оскільки він агрегує трафік сервера на гігабітному EtherChannel або 10 GigE/10 Gigabit EtherChannel до рівня агрегації. Протоколи маршрутизації Spanning Tree або Layer 3 поширюються з рівня агрегації на рівень доступу, в залежності від того, яка модель рівня доступу використовується. Cisco рекомендує використовувати комутатори рівня доступу, логічно об'єднані в групи по два для підтримки резервних з'єднань серверів або для підтримки різноманітних з'єднань для виробничих, резервних і керуючих Ethernet-інтерфейсів.

Рівень доступу складається в основному з трьох моделей: Рівень 2 з петлею, Рівень 2 без петлі та Рівень 3.

Рисунок 3.7 ілюструє рівень доступу з використанням петлевої моделі рівня 2 у топології трикутної та квадратної петлі.

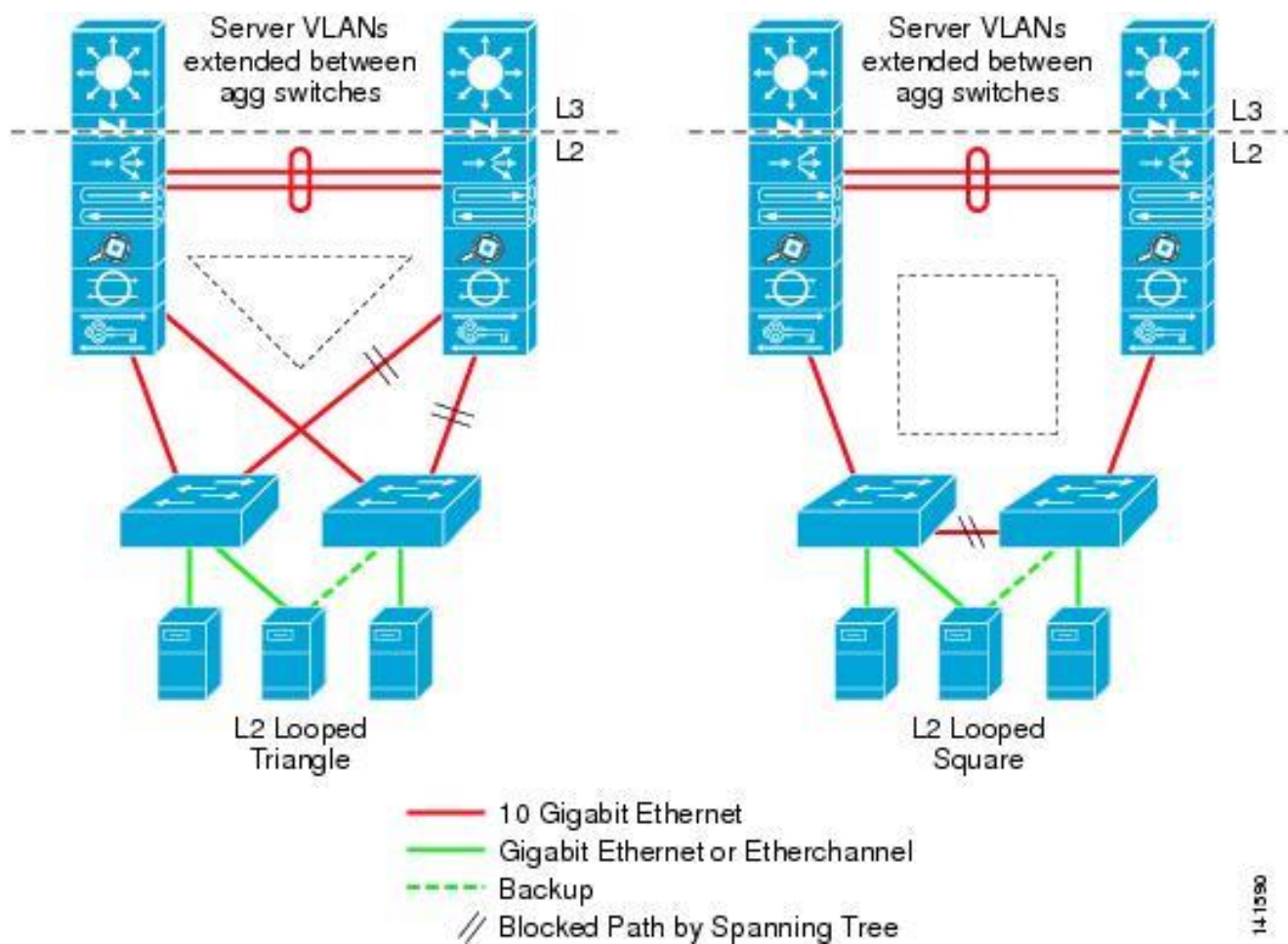


Рисунок 3.7 - Закільцьовані топології рівня доступу

На сьогоднішній день найбільш поширеною є петлеподібна топологія на основі трикутника. Петльова топологія є найбільш бажаною на рівні доступу до центру обробки даних з наступних причин:

- розширення VLAN - можливість додавання серверів до певної VLAN на всьому рівні доступу є ключовою вимогою в більшості центрів обробки даних;
- топології із зацикленням на відмовостійкості за своєю суттю є надлишковими;
- інтероперабельність сервісних модулів - сервісні модулі, що працюють в режимах "активний-резервний", вимагають суміжності на рівні 2 між своїми інтерфейсами;
- вимоги до серверів для суміжності на рівні 2 для підтримки об'єднання мережевих карт і кластеризації високої доступності.

Рисунок 3.8 ілюструє рівень доступу з використанням безшлейфової моделі

рівня 2 у топологіях безшлейфової U і безшлейфової перевернутої U.

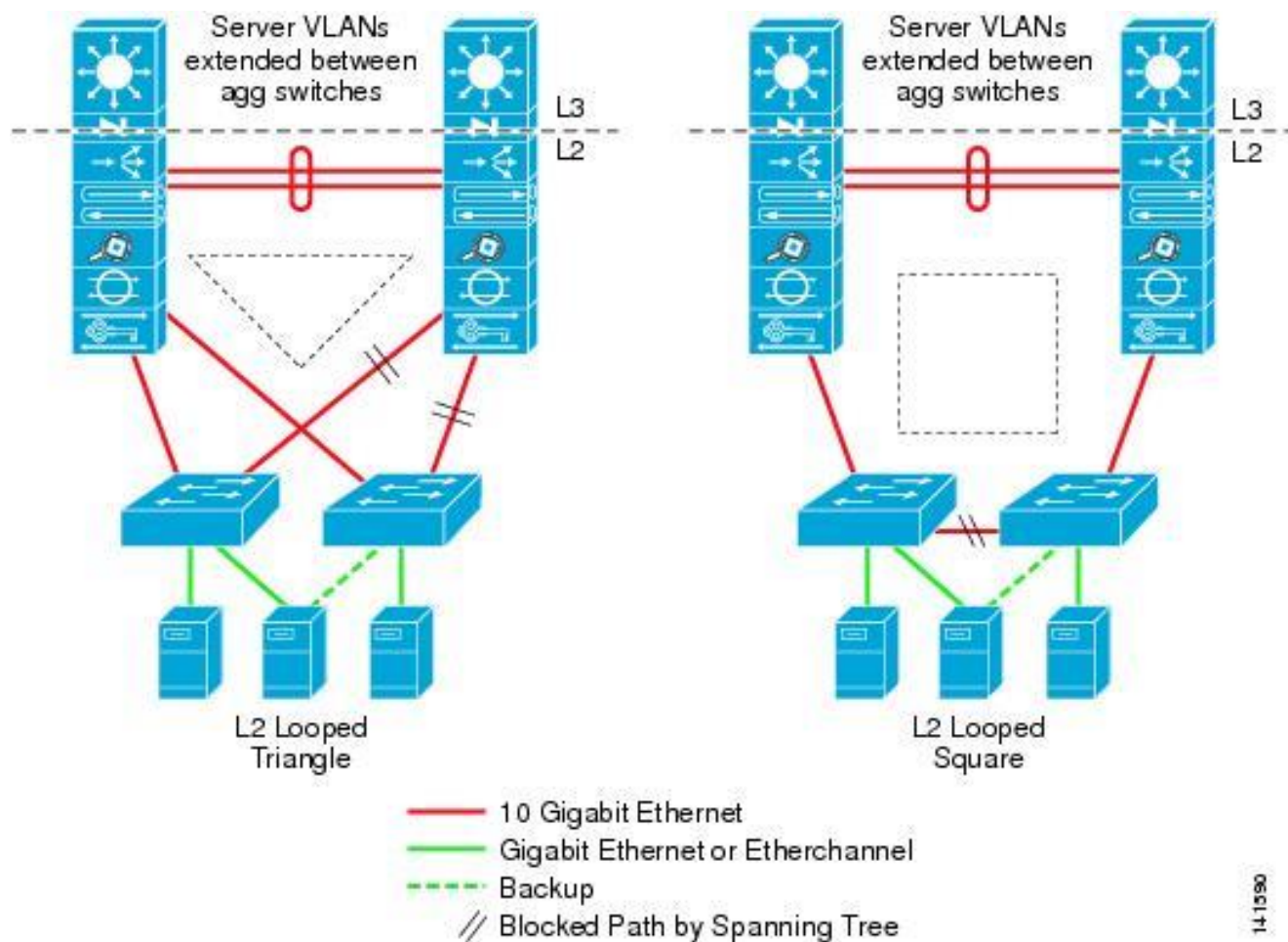


Рисунок 3.8 - Топології без петель на рівні доступу

Модель без петель на рівні 2 зазвичай використовується, коли петлеподібна топологія є небажаною. Це може бути пов'язано з недосвідченістю в роботі з протоколами остовного дерева рівня 2, необхідністю, щоб всі висхідні лінії були активними, або негативним досвідом, пов'язаним з реалізацією STP. Нижче наведено основні відмінності між топологією з петлями та без петель:

- немає блокування на висхідних посланнях, всі послання активні в безпетльовій топології;
- сусідство на рівні 2 для серверів обмежується однією парою комутаторів доступу в безпетльовій топології;
- розширення VLAN в центрі обробки даних не підтримується в топології U без петель, але підтримується в топології U з перевернутою петлею.

При використанні моделі без петель все одно необхідно запускати STP як

інструмент запобігання петлям.

Сервісні модулі можуть не працювати належним чином під час певних збоїв у разі використання безпетльової U-подібної топології.

Рисунок 3.9 ілюструє рівень доступу з використанням моделі рівня 3.

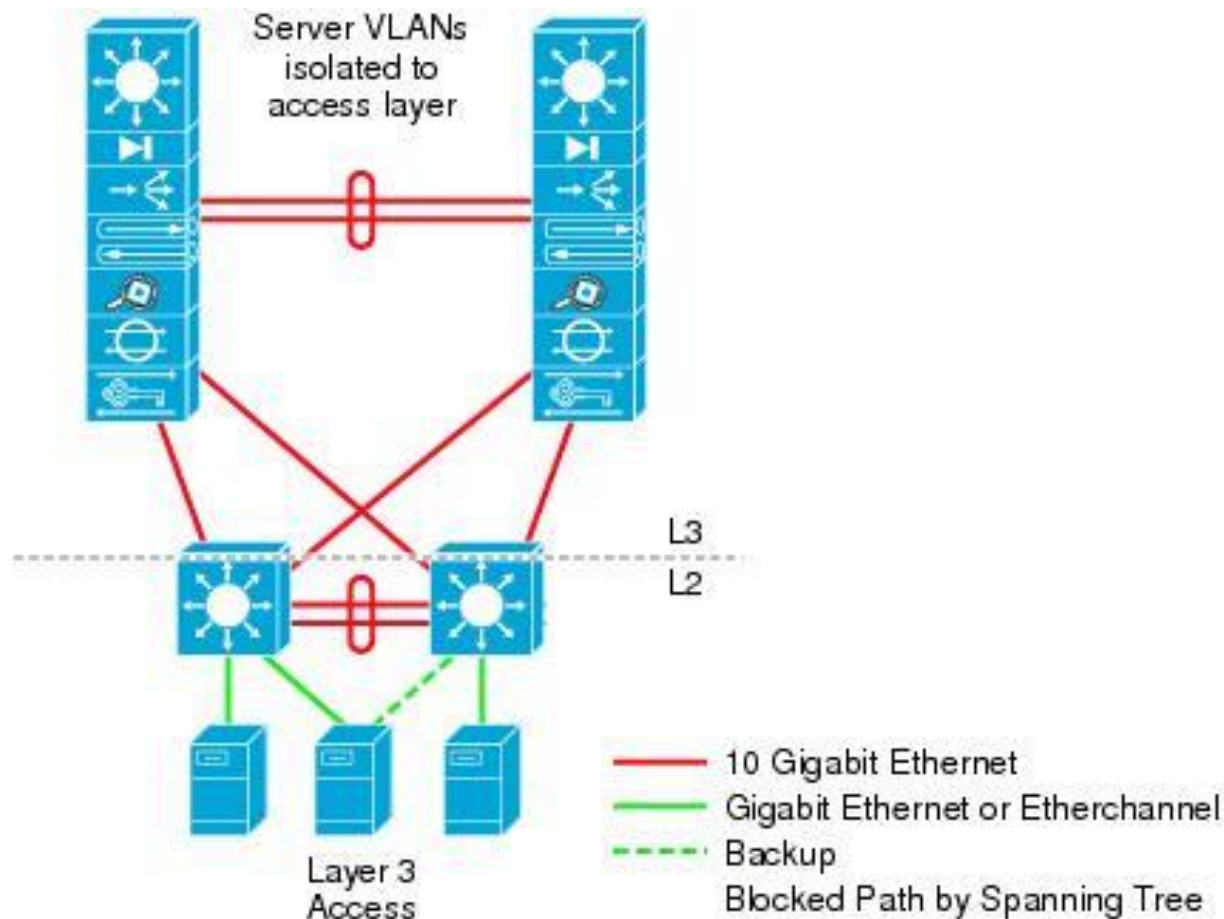


Рисунок 3.9 - Топологія 3-го рівня доступу

Модель доступу 3-го рівня зазвичай використовується для обмеження або утримання широкомовних доменів до певного розміру. Це може бути використано для зменшення впливу проблем широкомовних доменів або для захисту певних серверів, на які може негативно вплинути певний рівень широкомовності. Доступ третього рівня має такі характеристики:

- всі висхідні канали активні і використовують балансування навантаження CEF до максимуму ECMP (наразі 8);
- суміжність на рівні 2 для серверів обмежується однією парою комутаторів доступу в топології рівня 3;

- розширення VLAN у центрі обробки даних неможливе.

При використанні моделі доступу 3-го рівня Cisco все ще рекомендує використовувати STP як інструмент запобігання петель. Протокол STP буде активним лише на міжкомутаційних магістральних та серверних портах.

Оскільки сервісні модулі активного та резервного режимів вимагають суміжності на рівні 2 між своїми інтерфейсами, конструкція доступу рівня 3 не дозволяє розміщувати сервісні модулі на рівні агрегації і вимагає їх розміщення в кожній парі комутаторів доступу. Конструкція доступу 3-го рівня, яка використовує VRF-Lite, може забезпечити рішення для сервісних модулів рівня агрегації, але вона не була протестована для включення в цей посібник.

Рекомендовані платформи для рівня доступу включають всі комутатори серії Cisco Catalyst 6500, оснащені процесорними модулями Sup720 для модульних реалізацій, і Catalyst 4948-10GE для реалізації в верхній частині стійки.

Модульний комутатор доступу Catalyst 6500 забезпечує високу щільність портів GE, 10GE(C) uplink, резервні компоненти і функції безпеки, а також надає високошвидкісну комутаційну структуру, необхідну для серверної ферми. Catalyst 4948-10GE має два 10GE uplink, резервне живлення, а також 48 серверних портів GE у форм-факторі 1RU, що робить його ідеальним рішенням для верхньої частини стійки. Комутатори серії Catalyst 6500 і Catalyst 4948-10GE використовують образ IOS, що забезпечує однаковий зовнішній вигляд конфігурації, спрощуючи розгортання серверних ферм.

Нижче наведені деякі з найбільш поширених міркувань при виборі платформ рівня доступу:

- щільність - щільність розміщення серверів разом з максимальною кількістю інтерфейсів, що використовуються в одній стійці/рядку, може допомогти визначити, яке рішення - модульне чи 1RU - є кращим. Якщо в одній стійці використовується велика кількість портів, для їх підтримки може знадобитися багато комутаторів висотою 1RU в кожній стійці. Модульні комутатори, розташовані в ряд, можуть зменшити складність з точки зору кількості комутаторів і забезпечити більшу гнучкість у підтримці різної кількості

серверних інтерфейсів;

- підтримка висхідного каналу 10 GigE/10 Gigabit EtherChannel - важливо визначити, який коефіцієнт перепідписки для кожної програми. Коли це значення відоме, його можна використовувати для визначення правильної пропускну здатності висхідного каналу, яка необхідна на комутаторі рівня доступу. Вибір комутатора з підтримкою 10 GigE і 10 Gigabit EtherChannel може бути важливим варіантом при розгляді поточних або майбутніх коефіцієнтів перевищення підписки;

- функції відмовостійкості - коли сервери з'єднані однією інтерфейсною картою NIC на рівні доступу, комутатор доступу стає єдиною точкою відмови. Це робить такі функції, як резервне живлення і резервний процесор, набагато більш важливими для комутатора рівня доступу;

- виробництво в порівнянні з використанням для розробки - мережа для розробки може не вимагати надмірності або багатих функцій програмного забезпечення, які необхідні для виробничого середовища;

- дизайн кабелів/вимоги до охолодження - щільність кабелів у серверній шафі та під підлогою може бути складною в управлінні та підтримці. Нагромадження кабелів також може створювати проблеми з охолодженням, якщо повітряні канали заблоковані. Використання комутаторів доступу 1RU може поліпшити конструкцію кабелів.

Рекомендовані платформи рівня доступу включають наступні моделі Cisco Catalyst:

- всі платформи серії Catalyst 6500 з процесором Sup720;
- каталізатор 4948-10G

Використання DFC на рівні доступу багаторівневої моделі не є обов'язковим. Вимоги до продуктивності більшості комутаторів доступу корпоративних центрів обробки даних задовольняються без використання DFC, а в багатьох випадках вони і не потрібні.

Якщо очікується інтенсивний трафік між серверами на одному модульному шасі, як, наприклад, у високопродуктивних обчислювальних системах, DFC,

безумовно, може підвищити продуктивність. У таблиці 3.1 наведено порівняння продуктивності.

Сервери, що підключаються на рівні доступу, можуть бути однодомними або дводомними для резервування. Сервер з одним підключенням не має захисту від відмов мережевої карти або комутатора доступу і являє собою єдину точку відмови. Резервування процесора і живлення є критично важливим на рівні доступу, коли використовуються окремі підключені сервери, оскільки відмова комутатора доступу може мати значний вплив на доступність мережі. Якщо окремі підключені сервери створюють велику вразливість, слід звернути увагу на платформи, які забезпечують повне резервування джерел живлення, резервування процесорів і перемикання за станом.

Додатки, які працюють на окремих підключених серверах, можуть використовувати балансувальники навантаження серверів, такі як CSM, для досягнення надмірності. У цьому випадку сервери певної групи додатків (VIP) розподіляються між двома або більше комутаторами рівня доступу, що усуває комутатор доступу як єдину точку відмови.

На рисунку 3.10 показано, як використовувати балансувальники навантаження для досягнення надмірності з однопідключеними серверами у фермі веб-серверів.



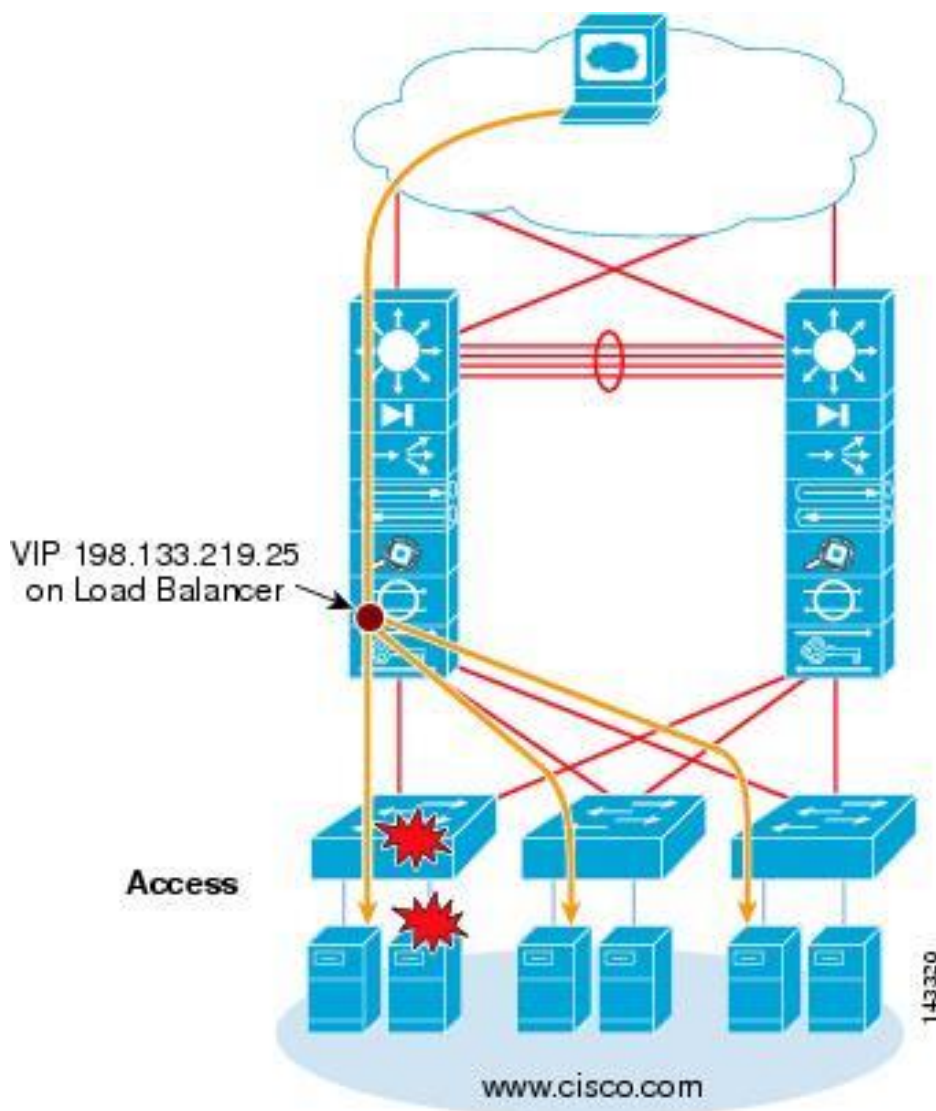


Рисунок 3.10 - Резервування серверів за допомогою балансувальників навантаження

У цьому прикладі відмова мережевої карти сервера або комутатора доступу призводить до того, що балансувальник навантаження автоматично виводить сервери з роботи, а сесії перенаправляються на сервери, що залишилися. Це досягається завдяки використанню функцій моніторингу стану CSM.

### 3.10 Спільне використання послуг на рівні агрегації

Топологія 2-рівневого закільцьованого доступу має унікальну перевагу - можливість використовувати послуги, що надаються сервісними модулями або

пристроями, розташованими на рівні агрегації. Інтегровані сервісні модулі на рівні агрегації оптимізують простір в стійці і кабелі, спрощують управління конфігурацією і підвищують загальну гнучкість і масштабованість.

Сервіси рівня агрегації, які можуть використовуватися серверами рівня доступу, включають наступні:

- балансування навантаження;
- брандмауер;
- розвантаження SSL (сеансове шифрування/дешифрування);
- моніторинг мережі;
- виявлення та запобігання вторгненням;
- кеш-двигуни.

### **3.11 Рівень послуг центру обробки даних**

Комутатор рівня послуг забезпечує метод масштабування послуг за допомогою сервісних модулів без використання слотів у комутаторі рівня агрегації. Переміщення певних сервісних модулів з комутатора рівня агрегації збільшує кількість доступних слотів і покращує продуктивність рівня агрегації. Наприклад, це корисно, коли потрібна ферма сервісних модулів Cisco ACE і Cisco Firewall Service Modules (FWSM).

На рисунку 3.11 показано топологію з дизайном комутатора сервісного рівня.

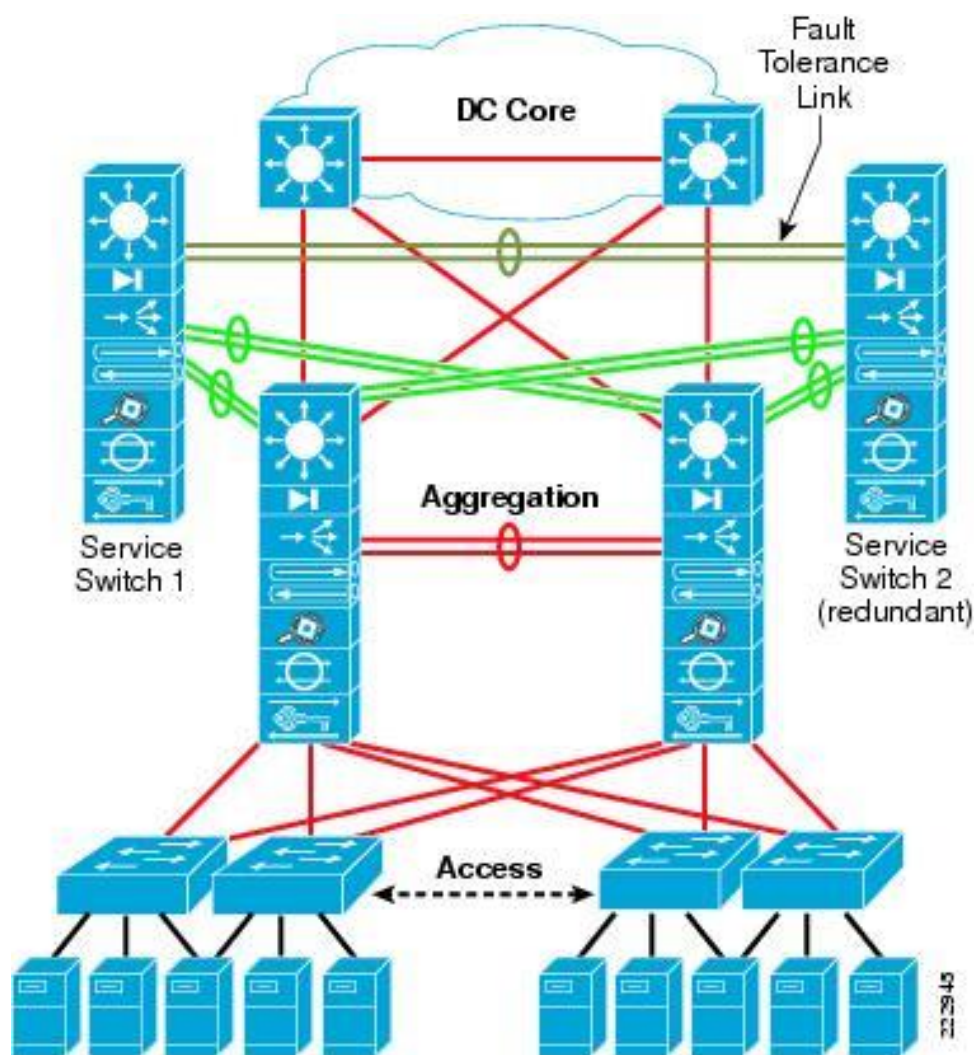


Рисунок 3.11 - Комутатор рівня обслуговування центру обробки даних

Як правило, сервісні модулі Cisco ACE і Cisco FWSM розташовуються в комутаторі агрегації для надання послуг серверам через комутатори рівня доступу. Розміщення цих сервісних модулів в окремому автономному комутаторі, підключеному за допомогою магістралей 802.1Q, дозволяє агрегаційному рівню підтримувати вищу щільність висхідних каналів на рівні доступу. Це особливо корисно, коли на рівні агрегації зростають вимоги до щільності портів 10 GigE.

Змішування старих лінійних плат або сервісних модулів з інтегрованою архітектурою комутаційної матриці Sup720 може обмежити загальну комутаційну спроможність і може не відповідати вимогам продуктивності комутаторів рівня агрегації. У цьому розділі розглядаються наслідки, пов'язані з розміщенням класичних лінійних карт в комутаторі рівня агрегації.

Модуль CSM підключається до шини Catalyst 6500 за допомогою з'єднання EtherChannel зі швидкістю 4 Гбіт/с на об'єднувчій платі. Цей інтерфейс можна побачити, переглянувши зарезервовану адресу EtherChannel порту 259, як показано нижче:

```
AGG1#sh etherchannel 259 port-channel
Port-channel: Po259
-----
Age of the Port-channel      = 4d:00h:33m:39s
Logical slot/port           = 14/8           Number of ports = 4
GC                          = 0x00000000    HotStandBy port = null
Port state                   = Port-channel Ag-Inuse
Protocol                     = -
Ports in the Port-channel:
Index  Load   Port      EC state  No of bits
-----+-----+-----+-----+-----
  0     11    Gi3/1     On/FEC    2
  1     22    Gi3/2     On/FEC    2
  2     44    Gi3/3     On/FEC    2
  3     88    Gi3/4     On/FEC    2
```

Цей інтерфейс EtherChannel зі швидкістю 4 Гбіт/с використовується для всього трафіку, що входить і виходить з балансувальника навантаження, і використовує алгоритми хешування для розподілу сеансового навантаження так само, як і зовнішнє фізичне з'єднання EtherChannel. CSM також базується на класичній шинній архітектурі і залежить від Sup720 для комутації пакетів на вході і виході з інтерфейсу EtherChannel, оскільки він не має прямого інтерфейсу з інтегрованою комутаційною структурою Sup720.

Якщо один канал EtherChannel зі швидкістю 4 Гбіт/с не забезпечує достатньої пропускної здатності для задоволення проектних вимог, то для масштабування можна використовувати кілька модулів CSM. Підтримка декількох модулів CSM не створює особливих проблем, але збільшує кількість слотів, що використовуються в комутаторі рівня агрегації, що може бути небажаним.

Оскільки CSM базується на класичній шині, він повинен надсилати усічені заголовки пакетів на Sup720 PFC3 для визначення місця призначення пакетів на об'єднувчій платі. Коли в комутаторі є один модуль класичної шини, всі лінійні карти, які не підтримуютьDFC, повинні виконувати пошук усічених заголовків, що обмежує загальну продуктивність системи.

У таблиці 3.2 наведено огляд продуктивності комутації за типами модулів.

Таблиця 3.2 - Продуктивність з класичними модулями шини

Конфігурація системи з Sup720	Пропускна спроможність в Мбіт/с	Пропускна здатність в Гбіт/с
Модулі серії Classic (CSM, 61XX-64XX)	До 15 Мбіт/с (на систему)	Загальна шина 16 Гбіт/с (класична шина)
Модулі серії CEF256 (FWSM, SSLSM, NAM-2, IDSM-2, 6516)	До 30 Мбіт/с (на систему)	1 x 8 Гбіт/с (виділено на кожен слот)
Модулі серії CEF720 (6748, 6704, 6724)	До 30 Мбіт/с (на систему)	2 x 20 Гбіт/с (виділено на один слот) (6724=1 x 20 Гбіт/с)
Модулі серії CEF720 з DFC3 (6704 з DFC3, 6708 з DFC3, 6748 з DFC3, 6724 з DFC3)	Підтримувати до 48 Мбіт/с (на слот)	2 x 20 Гбіт/с (виділено на кожен слот) (6724=1 x 20 Гбіт/с)

Сервісним комутатором може бути будь-яка платформа серії Catalyst 6500, що використовує процесорний модуль Sup720. При виборі процесора супервізора слід враховувати вимоги до економного використання, майбутню міграцію на модулі наступного покоління, вимоги до продуктивності та вимоги до каналів зв'язку з модулем агрегації. Наприклад, якщо планується використання 10 GigE uplink, необхідно використовувати sup720 для підтримки модуля 6704 10 GigE. Sup32-10 GigE можна використовувати, якщо використовуються тільки класичні модулі з підтримкою шини, такі як CSM, але це не тестувалося в рамках цього посібника, і в цілому Sup32 не рекомендується для використання в центрі обробки даних через загальні високі характеристики продуктивності, які бажані для цього об'єкта.

Якщо Cisco ACE або CSM у сервісному комутаторі налаштовано на передачу даних про стан маршруту (Route Health Injection, RHI), необхідна конфігурація рівня 3 для комутатора рівня агрегації, оскільки RHI знає, як вставити маршрут хоста тільки в таблицю маршрутизації локального MSFC. З'єднання 3-го рівня дозволяє протоколу маршрутизації перерозподіляти маршрут

хоста на рівень агрегації.

### 3.12 Потік трафіку через сервісний рівень

Сервісний комутатор підключений до обох комутаторів агрегації за допомогою каналів 10 GigE, налаштованих як магістралі 802.1Q. З логічної точки зору, це можна розглядати як розширення VLAN сервісного модуля через магістраль 802.1Q. На рисунку 3.12 показано потік сеансу з використанням ACEhg в односторонньому режимі на комутаторі сервісного рівня. Самі сервісні комутатори з'єднані між собою каналом Gigabit Ethernet або 10 GigE, щоб відмовостійкі віртуальні мережі залишалися тільки на сервісних комутаторах.

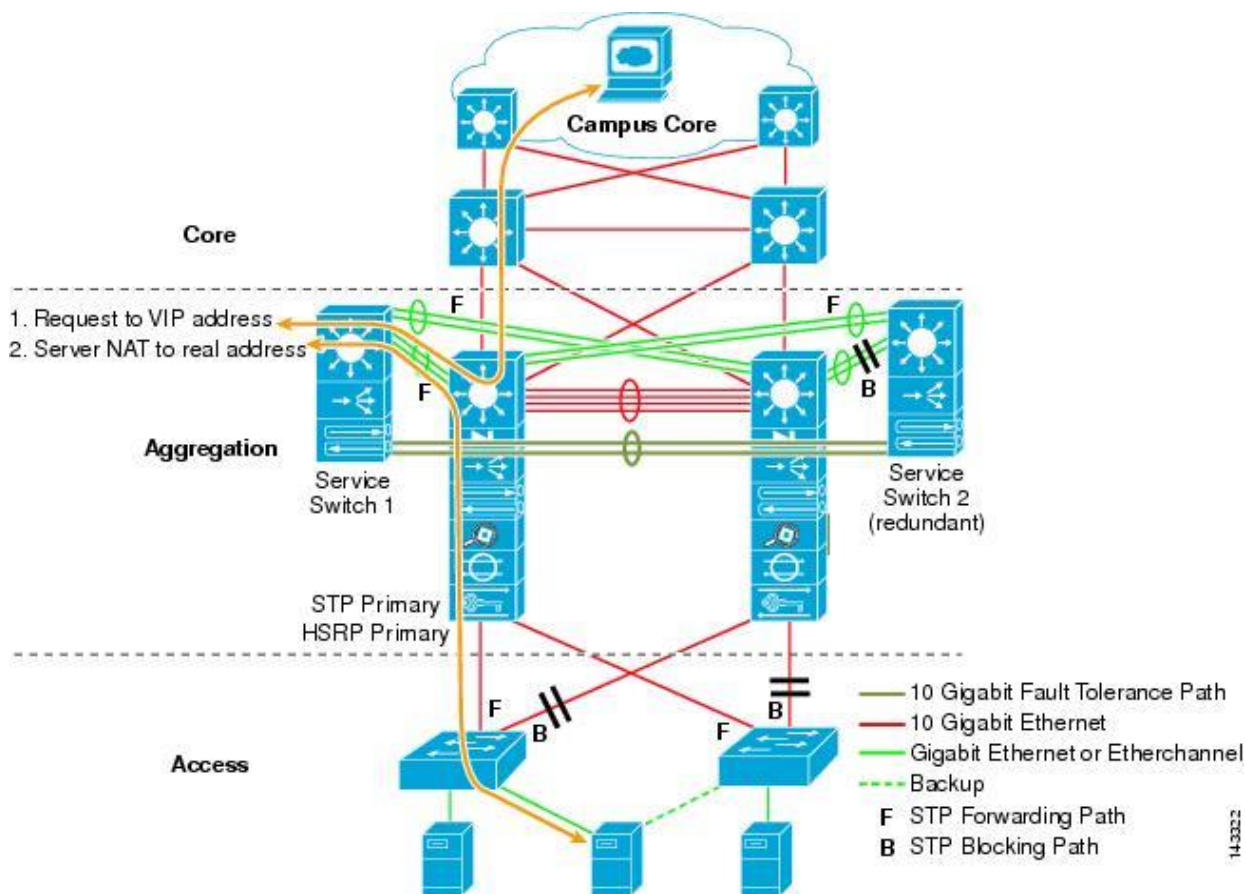


Рисунок 3.12 - Потік трафіку комутатора службового рівня

Віртуальні локальні мережі, що використовуються для підтримки конфігурації Cisco ACE, простягаються через магістралі 802.1Q (GEC або 10GigE) від комутатора рівня агрегації до кожного сервісного комутатора. Відмовостійкі

(FT) VLAN не простягаються через ці магістралі. Огинаюче дерево блокує тільки одну магістраль від вторинного комутатора послуг до комутатора рівня агрегації 2. Ця конфігурація забезпечує шлях переадресації до первинного комутатора з обох комутаторів агрегації.

Слід ретельно продумати смугу пропускання, яка використовується для підключення сервісних комутаторів. Cisco ACE здатний підтримувати до 16 Гбіт/с серверного трафіку. Щоб уникнути перевантажень, з'єднання комутатора з комутаторами агрегації повинно бути достатньо великим, щоб уникнути перевантажень.

Рівень комутаторів послуг слід розгорнути попарно, щоб підтримувати повністю надлишкову конфігурацію. Це дозволяє розміщувати основні/активні сервісні модулі на одному шасі, а резервні/резервні - на другому. Обидва сервісні комутатори повинні бути надлишково підключені до кожного комутатора агрегації, щоб виключити будь-яку єдину точку відмови.

На рисунку 3.13 комутатор послуг 1 налаштований на активні сервісні модулі, тоді як комутатор послуг 2 використовується для резервних сервісних модулів. Якщо комутатор послуг 1 виходить з ладу, комутатор послуг 2 стає активним і забезпечує обхід відмови за станом для існуючих з'єднань.

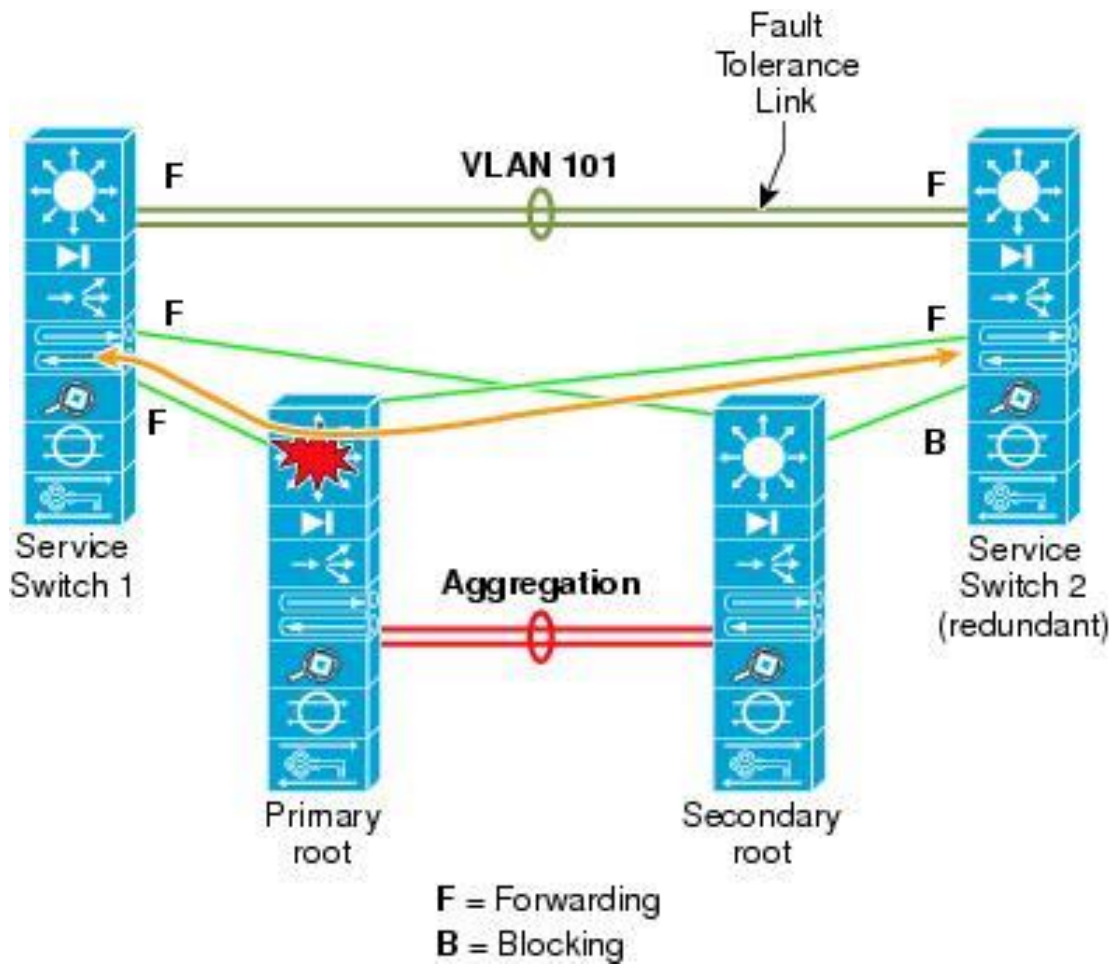


Рисунок 3.13 - Резервування з дизайном комутатора рівня обслуговування

FT VLAN використовується для підтримки стану сеансу між сервісними модулями. FT VLAN - це пряме магістральне з'єднання 802.1Q між сервісними комутаторами. FT VLAN не передаються через канали, підключені до пар комутаторів агрегації та між ними.

Сервісний комутатор призначений для участі в основному дереві і автоматично вибирає шляхи до первинного кореня в агрегації 1 для серверної VLAN. Якщо сервісний комутатор 1 виходить з ладу, зв'язок з сервісним модулем FT припиняється, і первинним стає сервісний комутатор 2.



## ВИСНОВКИ

Обробка великих даних є складним технологічним процесом, що вимагає глибоких програмно-інженерних знань для розробки моделі даних, вибору відповідних програмно-апаратних засобів та оцінки сукупної вартості управління даними. У багатьох випадках обробка даних може бути проведена досить скромними засобами, за допомогою оренди систем зберігання та обробки у хмарному середовищі, в інших випадках потрібна оренда або навіть будівництво власного ЦОД та встановлення власного обладнання, у третіх випадках – вартість роботи з даними може перевищити дохід від їх обробки та обробка даних самотужки недоцільна, проте може бути виконана за допомогою підрядника.

За останні кілька років архітектури центрів обробки даних стали більш складними через вимоги до високої доступності, низької затримки та високої продуктивності.

В результаті проведеного дослідження можна зробити висновок, що технологія Cisco Catalyst Data Center Switches надає комплексний підхід до забезпечення високої продуктивності та низької затримки в мережах Cisco, поєднуючи в собі широку пропускну здатність, низьку затримку, високу надійність, управління трафіком, масштабованість та безпеку. Ці особливості дозволяють мережам Cisco з використанням технології Cisco Catalyst Data Center Switches ефективно виконувати завдання передачі великих обсягів даних з високою швидкістю та мінімальною затримкою.

Застосування технології Cisco Catalyst Data Center Switches у різних типах дата-центрів дозволяє підвищити продуктивність роботи бізнесу, зменшити затримку та покращити якість обслуговування користувачів. Вона також дозволяє ефективно використовувати ресурси мережі та забезпечувати їх оптимальне розподілення.

Отже, використання технології Cisco Catalyst Data Center Switches є важливим фактором для досягнення високої продуктивності та низької затримки в мережах.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Blackwell M., Sen M. Large Datasets and You: <http://www.matblackwell.org/files/papers/bigdata.pdf>
2. Ross N. Faster! Higher! Stronger! – A Guide to Speeding Up R Code for Busy People: <http://www.noamross.net/blog/2013/4/25/faster-talk.html>
3. Ryan R. Rosario. Taking R to the Limit, Part II: Working with Large Datasets: [http://www.bytemining.com/wp-content/uploads/2010/08/r\\_hpc\\_II.pdf](http://www.bytemining.com/wp-content/uploads/2010/08/r_hpc_II.pdf)
4. Big Data Specialization. <https://www.coursera.org/specializations/big-data>
5. Mining Massive Datasets. Online course. <https://online.stanford.edu/course/mining-massive-datasets-self-paced>
6. G. Press. 12 Big Data Definitions. What's yours? <https://www.forbes.com/sites/gilpress/2014/09/03/12-big-data-definitions-whatsyours/>
7. Big Data awesome list. <https://github.com/onurakpolat/awesome-bigdata>
8. Keuper F., Schmidt D., Schomann M. Smart Big Data Management. ISBN-10: 3832537686. 2014. <https://www.amazon.com/Smart-Data-Management-Frank-Keuper/dp/3832537686>
9. Mayer-Schönberger V. Big Data: A Revolution That Will Transform How We Live, Work, and Think. 2013. ISBN-10: 1848547927 <https://www.amazon.com/Big-Data-Revolution-Transform-Think/dp/1848547927/>
10. Smith M.D., Telang R. Streaming, Sharing, Stealing: Big Data and the Future of Entertainment (MIT Press). 2016. <https://www.amazon.com/Streaming-Sharing-Stealing-Future-Entertainment/dp/0262034794/>
11. Karimi H.A. Big Data: Techniques and Technologies in Geoinformatics. 2014. ISBN-10: 1138073199 <https://www.amazon.com/Big-Data-Techniques-Technologies-Geoinformatics-ebook/dp/B00HZNQKMM/>
12. Marz N., Warren J. Big Data: Principles and best practices of scalable realtime data systems. 2015. ISBN-10: 1617290343 <https://www.amazon.com/Big-Data-Principles-practices-scalable/dp/1617290343/>

13. Separation of storage and compute in BigQuery.  
<https://cloud.google.com/blog/big-data/2017/11/separation-of-storage-andcompute-in-bigquery>
14. Mayer-Schonberger V., Ramge T. Reinventing Capitalism in the Age of Big Data. 2018. ISBN-10: 046509368X
15. <https://www.amazon.com/Reinventing-Capitalism-Age-Big-Data/dp/046509368X/>
16. Bahga A., Madiseti V. Big Data Science & Analytics: A Hands-On Approach. 2016. ISBN-10: 0996025537. <https://www.amazon.com/Big-Data-Science-Analytics-Hands/dp/0996025537/>
17. Marr B. Big Data: Using SMART Big Data, Analytics and Metrics To Make Better Decisions and Improve Performance. 2015. ISBN-10: 1118965833.  
<https://www.amazon.com/Big-Data-Analytics-Decisions-Performance/dp/1118965833/>
18. Marr B. Data Strategy: How to Profit from a World of Big Data, Analytics and the Internet of Things. 2017. ISBN-10: 074947985X.  
<https://www.amazon.com/Data-Strategy-Profit-Analytics-Internet/dp/074947985X/>
19. Jones H. Data Analytics: An Essential Beginner's Guide To Data Mining, Data Collection, Big Data Analytics For Business, And Business Intelligence Concepts. 2018. ISBN-10: 1985097974.
20. <https://www.amazon.com/Data-Analytics-Essential-Collection-Intelligence/dp/1985097974/>
21. Sawchik T. Big Data Baseball: Math, Miracles, and the End of a 20-Year Losing Streak. 2015. ISBN-10: 1250063507. <https://www.amazon.com/Big-Data-Baseball-Miracles-20-Year/dp/1250063507/>
22. Ferguson A.G. The Rise of Big Data Policing: Surveillance, Race, and the Future of Law Enforcement. 2017. ISBN-10: 1479892823.

## **ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ**