ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Дослідження інструментів та методології для моніторингу та керування продуктивністю мережі в режимі реального часу, включаючи прогнозну аналітику, автоматизовану діагностику та проактивну оптимізацію мережі»

на здобуття освітнього ступеня магістра
зі 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

_____ Віталій СЛЮСАР
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав:
здобувач вищої освіти Віталій СЛЮСАР
група КСДМ-61

Керівник: Артем АНТОНЕНКО
науковий ступінь, вчене звання д.т.н., професор

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 кафедрою Комп'ютерної інженерії

_____ Ім'я, ПРІЗВИЩЕ
«_____» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Слюсар Віталій Олександрович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Дослідження інструментів та методології для моніторингу та керування продуктивністю мережі в режимі реального часу, включаючи прогнозну аналітику, автоматизовану діагностику та проактивну оптимізацію мережі

керівник кваліфікаційної роботи Артем Антоненко д.т.н., професор,
(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» 10.2023р. №145

2. Строк подання кваліфікаційної роботи «29» грудня 2023р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, зібрані дані про мережеві показники, які характеризують продуктивність мережі з точки зору додатків Grid.

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

а) Дослідження принципів моніторингу мережі;
б) Аналіз інструментів та засобів моніторингу мережі;
в) Реалізація прогнозної аналітики щодо мережі;

5. Перелік графічного матеріалу: *презентація*
   1) Системний хід діагностичного процесу;
   2) Приклад діагностики збою виклику;
   3) Приклад сценаріїв PowerShell;
   4) Збір бази даних за допомогою python;

6. Дата видачі завдання«19» жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вивчення літератури, пов'язаної з методами та інструментами моніторингу продуктивності мережі	19.10-05.11.23	
2	Складання огляду літератури на основі вивченої інформації	05.11-12.11.23	
3	Вивчення існуючих методів моніторингу та відповідних інструментів	13.11-19.11.23	
4	Розробка архітектурного дизайну системи моніторингу	20.11-25.11.23	
5	Визначення вимог до моніторингу мережі	27.11-03.12.23	
6	Розробка засобів аналізу даних для створення мережевих показників	04.12-10.12.23	
7	Оформлення роботи: вступ, висновки, реферат	11.12-20.12.23	
8	Розробка демонстраційних матеріалів	21.12-29.12.23	

Здобувач вищої освіти _____ Віталій СЛЮСАР
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ Артем АНТОНЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 84 стор., 7 табл., 44 рис., 20 джерел.

Мета роботи: Дослідження методів та інструментів моніторингу продуктивності мережі з використанням прогностичної аналітики та машинного навчання.

Об'єкт дослідження: Мережі комп'ютерних систем.

Предмет дослідження: Методи моніторингу та аналізу продуктивності мережі з використанням прогнозного аналізу та інтелектуального аналізу даних.

Короткий зміст роботи: Робота присвячена огляду літератури з питань моніторингу продуктивності мережі. В ній розглядаються класичні показники та методи моніторингу, а також архітектурний дизайн системи моніторингу. Дослідження включає використання прогнозної аналітики та машинного навчання для прогнозування продуктивності мережі. Результати роботи можуть сприяти вдосконаленню моніторингу та оптимізації роботи комп'ютерних мереж.

КЛЮЧОВІ СЛОВА: МОНІТОРИНГ ПРОДУКТИВНОСТІ МЕРЕЖІ, СИСТЕМА МОНІТОРИНГУ, ІНСТРУМЕНТИ МОНІТОРИНГУ, СХОВИЩЕ ДАНИХ, АНАЛІЗ ДАНИХ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, КЛАСИФІКАЦІЯ ТА ПРОГНОЗУВАННЯ, БЕЗПЕКА МЕРЕЖІ.

ABSTRACT

The textual part of the qualification work for obtaining an educational master's degree: 84 pages, 7 table, 44 figures, 20 sources.

Objective: Research methods and tools for monitoring network performance using predictive analytics and machine learning.

Object of research: Networks of computer systems.

Subject of study: Methods for monitoring and analyzing network performance using predictive analysis and data mining.

Summary of the work: The work is devoted to a review of the literature on monitoring network performance. It discusses classical indicators and monitoring methods, as well as the architectural design of the monitoring system. The study includes the use of predictive analytics and machine learning to predict network performance. The results of the work can contribute to improving the monitoring and optimization of computer networks.

KEYWORDS: NETWORK PERFORMANCE MONITORING, MONITORING SYSTEM, MONITORING TOOLS, DATA WAREHOUSE, DATA ANALYSIS, INTELLIGENT DATA ANALYSIS, CLASSIFICATION AND FORECASTING, NETWORK SECURITY.

ЗМІСТ

ВСТУП

Сучасний розвиток комп'ютерних мереж вимагає постійного моніторингу та підтримки їх продуктивності. Ефективне функціонування мережі є ключовим фактором для забезпечення надійності та якості обслуговування. З урахуванням швидкого зростання обсягу даних, розширення функціональності та збільшення складності мережевих інфраструктур, виникає потреба у розробці методів та інструментів моніторингу продуктивності мережі. Відповідно, актуальність дослідження полягає в пошуку ефективних методів та інструментів для забезпечення оптимальної продуктивності мереж.

Метою даного дослідження є розробка методів та інструментів моніторингу продуктивності комп'ютерних мереж. Для досягнення цієї мети передбачені такі завдання:

1) Аналіз літературних джерел та огляд існуючих методів та інструментів моніторингу продуктивності мережі;
2) Розробка архітектурного дизайну системи моніторингу мережі, який включатиме інструменти моніторингу, сховище даних, засоби аналізу та засоби доступу до показників продуктивності;
3) Вибір та реалізація інструментів моніторингу для прототипу системи;
4) Застосування прогностичної аналітики та методів машинного навчання для прогнозування продуктивності мережі;
5) Експериментальне тестування розробленої системи та оцінка отриманих результатів;

Об'єктом дослідження є комп'ютерні мережі, які використовуються для передачі даних та забезпечення зв'язку між комп'ютерами та іншими пристроями.

Предметом дослідження є методи та інструменти моніторингу продуктивності мережі, включаючи аналіз показників продуктивності, прогнозування та оптимізацію роботи мережі.

У даному дослідженні використовуються наступні методи:

1) Аналіз літературних джерел - для огляду існуючих методів моніторингу продуктивності мережі та прогнозування;
2) Розробка архітектурного дизайну системи - для створення ефективної інфраструктури моніторингу мережі;
3) Вибір і реалізація інструментів моніторингу - для забезпечення збору та аналізу даних щодо продуктивності мережі;
4) Прогностична аналітика та методи машинного навчання - для прогнозування продуктивності мережі на основі історичних даних;
5) Експериментальне тестування - для перевірки ефективності розробленої системи моніторингу;

Наукова новизна дослідження полягає у розробці комплексного підходу до моніторингу продуктивності мережі, який включає використання прогнозної аналітики та методів машинного навчання. Отримані результати дозволять покращити якість обслуговування мережі, виявляти потенційні проблеми та забезпечувати ефективне управління мережевою інфраструктурою.

Практична значущість отриманих результатів полягає в можливості використання розробленої системи моніторингу продуктивності мережі для підтримки оптимальної роботи комп'ютерних мереж. Це дозволить організаціям забезпечувати надійну передачу даних, виявляти та усувати проблеми мережі та планувати майбутні розширення з урахуванням прогнозування продуктивності.

Теоретична значущість полягає в розширенні наукових знань про методи та інструменти моніторингу продуктивності мережі з використанням прогнозної аналітики та методів машинного навчання. Результати дослідження сприятимуть розвитку теоретичних основ моніторингу мереж та їх оптимізації.

Методична значущість полягає в розробці практичних рекомендацій щодо вибору та застосування інструментів моніторингу продуктивності мережі. Це дозволить фахівцям у галузі інформаційних технологій ефективно використовувати розроблені методи та інструменти для підтримки мережевої інфраструктури.

РОЗДІЛ 1 ОГЛЯД ЛІТЕРАТУРИ

1.1 Методи та інструменти моніторингу продуктивності мережі

Вимірювання та моніторинг продуктивності мережі необхідні з двох важливих причин. По-перше, це надати інструменти, необхідні для перегляду продуктивності мережі з точки зору додатків Grid і, отже, виявлення будь-яких стратегічних проблем, які можуть виникнути (таких як вузькі місця, точки ненадійності, потреби в якості обслуговування). По-друге, це забезпечення метрик, необхідних для використання службами посередника ресурсів Grid. У цьому документі описано вимоги до моніторингу мережі. Потім описуються класичні показники, пов’язані з мережевим моніторингом, і каталогізуються існуючі методи моніторингу та відповідні інструменти. Представлено архітектурний дизайн системи моніторингу. Він складається з чотирьох функціональних блоків, а саме інструментів моніторингу або датчиків; сховище зібраних даних; засоби аналізу цих даних для створення мережевих показників; а також засоби доступу та використання похідних показників. У ньому описано інструменти, вибрані для цього випуску прототипу, і те, як вони надають метричну інформацію як проміжному програмному забезпеченню Grid, так і за допомогою візуалізації, людині-спостерігачу.

Прогностична аналітика даних і еволюція машинного навчання змінили світ. Застосування цих методів до різних аспектів життя нескінченні. Комп’ютерні мережі можуть використовувати прогнозну аналітику та машинне навчання для моніторингу продуктивності мережі та подальшого прогнозування її продуктивності [1]. Прогнозний аналіз використовує інструменти інтелектуального аналізу даних, щоб зробити прогноз і надати рекомендації на основі історичних даних [2].

Моделі прогнозної аналітики можуть виконувати статистичний і аналітичний аналіз різних типів даних. Зібрані дані потребують попередньої обробки перед застосуванням методів прогнозного аналізу, таких як алгоритми машинного

навчання. Необроблені дані обробляються та перетворюються у формат, придатний для подальшого аналізу машинного навчання або методів бізнес-аналітики.

Оброблені дані використовуються як вхідні дані для моделей машинного навчання для створення необхідного прогнозу. Необроблені дані отримують за допомогою методів аналізу даних. Інтелектуальний аналіз даних — це вилучення незрозумілої або прихованої прогнозної інформації з подальшим пошуком шаблону за допомогою методів статистичного розпізнавання шаблонів, таких як дерева рішень і нейронні мережі [2]. Дані поділяються на предиктори та змінні, які використовуються для створення прогнозної моделі. Дослідження проводилося за допомогою інтелектуального аналізу даних для вилучення та пошуку прихованих шаблонів із даних, зібраних у реальній комп'ютерній мережі.

Алгоритми машинного навчання, такі як дерева рішень, використовуються для виконання аналізу показників продуктивності мережі після поділу на предиктори та змінні відповіді [2]. Крім того, дані аналізуються за допомогою статистичних методів на основі тих самих показників продуктивності мережі, щоб порівняти результати обох методів. Прогностична аналітика поєднує кілька статистичних методів, включаючи алгоритми машинного навчання.

Використовуючи цей підхід, ми поєднуємо кілька методів, починаючи від статистики до машинного навчання та аналізу даних [3]. Прогнозування даних здійснюється шляхом знання всіх елементів рішень і використання моделей рішень для знаходження зв'язку між цими елементами [2]. Розробка алгоритмів машинного навчання дозволяє легко використовувати їх у мережевій сфері [1].

Алгоритми машинного навчання можуть вирішувати проблеми в мережевих доменах, створювати необмежений потенціал і непередбачені переваги в найближчі кілька десятиліть. Класифікація та прогнозування в машинному навчанні можуть вирішити проблеми мережі щодо продуктивності та безпеки мережі. Ці проблеми було дуже важко вирішити через складність мереж.

Застосування машинного навчання в типовій мережевій проблемі має кілька етапів. Перший крок – визначення типу проблеми, яка вимагає уваги (прогнозування, регресія або прийняття рішення). На другому етапі збираються

неупереджені дані з мережі та додатків, які містять журнали продуктивності додатків, мережеві показники тощо. На третьому етапі виконується аналіз даних, зібраних від попередньої обробки до вилучення функцій [1]. Четвертий крок — побудова моделі, яка, по суті, є моделлю навчання або алгоритмом, що вимагає навчання з використанням даних [1]. Крок п’ятий – це перевірка моделі, яка включає перевірку точності та перевірку того, чи модель переобладнана. Останній крок називається розгортанням і висновком, оскільки етап впровадження вимагає коригування та розгляду, які не застосовуються на етапі тестування.

Навчання залежить від наявності та точності даних. Відповіді моделі навчання в реальному часі вимагають наявності величезної кількості історичних даних і дуже складної моделі навчання. У дослідженні модель навчання не забезпечує відповіді в реальному часі, але вона може аналізувати історичні дані, коли вони представлені [1]. Модель навчання здатна передбачити програми, які споживають ресурси комп’ютера, такі як використання ЦП. Мета полягає в тому, щоб покращити роботу мережі та додатків, а також допомогти у проектуванні кінцевої мережі.

Інструменти моніторингу мережі використовуються для обчислення мережевих показників, які характеризують мережу. Вони будуть використовуватися проміжним програмним забезпеченням Grid для оптимізації продуктивності Grid-додатків; вони також будуть використовуватися мережевими дослідниками та розробниками, а також персоналом підтримки мережі для підтримки та управління мережею, від якої залежить функціонування Grid.

Моніторинг мережі використовується двома різними способами щодо Grid, і разом вони описують цілі моніторингу мережі.

Перш за все мережевий моніторинг використовуватиметься програмами Grid для оптимізації використання ними мереж, які складають Grid. Першочергово важливою буде публікація метрик, які описують поточну та майбутню поведінку мережі для проміжного програмного забезпечення Grid, щоб програми Grid могли коригувати свою поведінку для найкращого використання цього ресурсу.

По-друге, він використовуватиметься для забезпечення фонових вимірювань продуктивності мережі, що буде корисним для мережевих менеджерів і тих, кому доручено надавати мережеві послуги для програм Grid.

Інструмент продуктивності мережі для середовища Grid, був розроблений як частина набору обчислювальних інструментів Globus, але наразі недоступний у версії Globus, що працює в EDG Testbed1. З цієї причини WP7 розробив архітектуру моніторингу мережі для Grid.

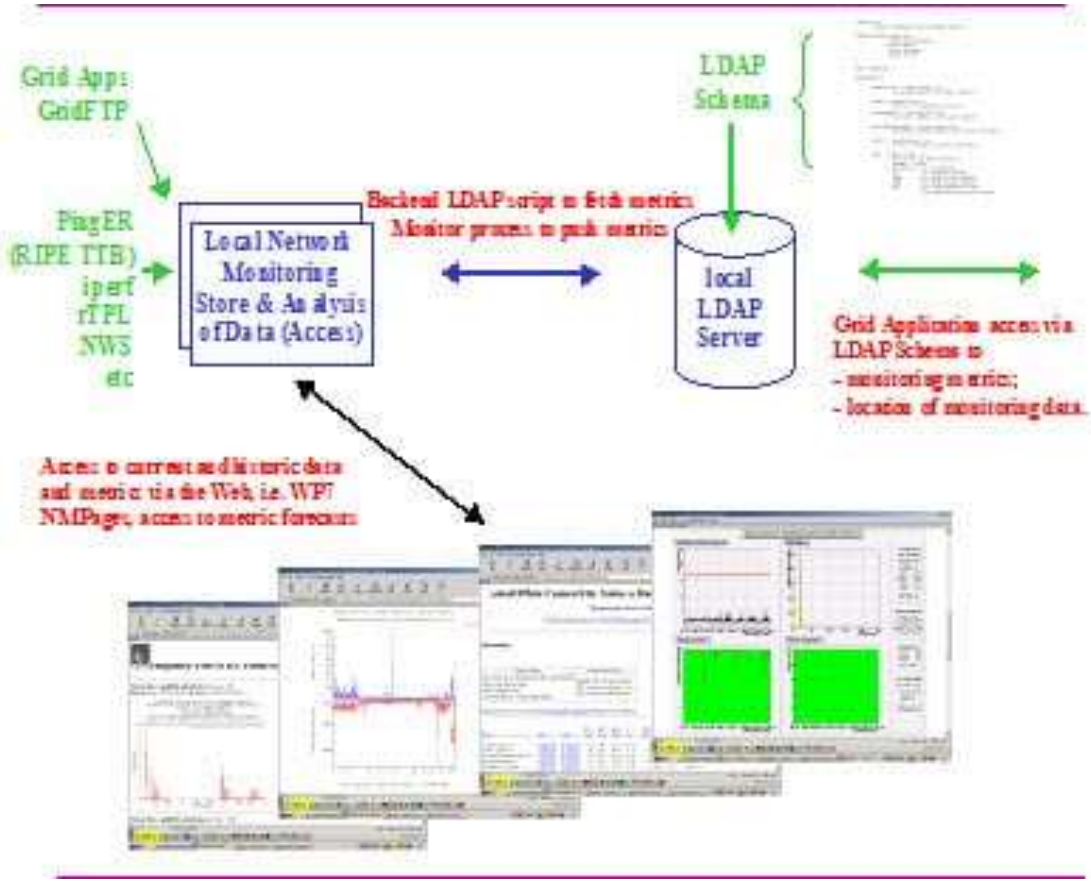

Рисунок 1.1 - Архітектура моніторингу мережі

На малюнку 1.1 схематично показано архітектуру моніторингу мережі. Він складається з чотирьох функціональних блоків, а саме інструментів моніторингу або датчиків; сховище зібраних даних; засоби аналізу цих даних для створення мережевих показників; а також засоби доступу та використання похідних показників.

Програми Grid можуть отримувати доступ до показників моніторингу мережі через служби LDAP відповідно до визначеної схеми LDAP. Сама служба LDAP збирає та підтримує дані метрики моніторингу мережі за допомогою внутрішніх сценаріїв, які отримують або надсилають поточну інформацію метрики зі сховища даних моніторингу локальної мережі. Незалежно від сайту запускається набір інструментів моніторингу мережі для збору даних, які описують локальний перегляд доступу до мережі для інших сайтів у Grid. Ці дані зберігаються в сховищі даних моніторингу мережі, яке тут моделюється як єдине ціле. Набір сценаріїв, пов’язаних із кожним інструментом моніторингу, доступний для забезпечення веб-доступу для перегляду та аналізу відповідних мережевих показників. Ця архітектура дозволяє легко додавати додаткові інструменти моніторингу, єдина вимога полягає в тому, що засоби для аналізу та візуалізації даних і або механізм push для оновлення локального сервера LDAP, або внутрішній сценарій, щоб дозволити серверу LDAP доступ до надаються конкретні показники [8].

Добре відомі інструменти – traceroute, pathchar, netperf тощо – будуть використовуватися для вимірювання основних показників.

Багато інструментів моніторингу мережі використовують ICMP, протокол рівня 3, який може підлягати стратегіям управління трафіком, відмінним від трафіку на основі TCP або UDP (протоколи рівня 4). Наприклад, в умовах перевантаження ICMP-трафік часто відкидається або обробляється зі зниженим пріоритетом. Однак була опублікована робота, у якій порівнюється використання обміну пакетами Ping і TCP Syn/Ack для характеристики визначеного мережевого маршруту. Результати були загалом еквівалентними, що свідчить про те, що в першому наближенні використання інструментів на основі ICMP забезпечує надійні вимірювання.

Використання PingER у спільноті HEP добре налагоджено. Він використовується для вимірювання часу відгуку (час проходження в обидві сторони в мілісекундах (мс)), відсотка втрачених пакетів, мінливості часу відгуку як короткочасного (шкала часу в секундах), так і більш тривалого часу, а також недостатньої доступності, тобто немає відповіді на послідовність пінгів.

Дані PingER зберігаються локально, щоб забезпечити веб-доступ для аналізу втрати пакетів, RTT і частотного розподілу вимірювань RTT як у графічному, так і в табличному форматі. Дані також збираються централізовано, щоб створити таблиці історії сайтів за місяцями або днями для всіх (або вибраних) сайтів, як видно з локальної точки моніторингу.

Проект PingER має добре налагоджену інфраструктуру, що включає сотні сайтів у багатьох країнах по всьому світу, і особливо зосереджений на спільнотах HEP/ESnet.

Очевидно, існує небезпека, що через обмеження швидкості ICMP або політику відхилення ping цей підхід або дасть недійсні результати, або не дасть жодних результатів. Це добре відомо, але поточне порівняння між PingER і Surveyor і коробкою RIPE NCC TTM [R10] свідчить про те, що занепокоєння необґрунтовані. Однак це не дає жодних гарантій у майбутньому [10].

Пакет RTPL (Remote Throughput Ping Load) використовується для періодичних тестів вимірювання продуктивності мережі між набором місць для перегляду продуктивності мережі з точки зору користувача. Вимірювання продуктивності складаються з вимірювання пропускної здатності між розташуваннями. За замовчуванням використовуються всі пари в наборі місць, але також можна вибрати пари місць для тестів. Крім того, вимірюється навантаження на задіяні системи моніторингу, щоб будь-яку зміну продуктивності можна було пов’язати з навантаженням конкретної машини. Через мету забезпечити вимірювання продуктивності мережі з точки зору користувача, а не як максимальну можливу пропускну здатність мережі, параметри вимірювання налаштовані на значення за замовчуванням, а тривалість тестування обмежена. Інакше різні довгострокові статистичні дані розмили б короткочасні коливання.

Вимірювання виконує контрольний хост. Керуючий хост запускає вимірювання продуктивності мережі в кожному з розташувань, що беруть участь, за допомогою захищеної віддаленої команди оболонки, а результати вимірювань повертаються за допомогою аналогічного механізму. Виконуються такі вимірювання продуктивності мережі,

Пропускна здатність. Як визначено: «Максимальна швидкість, з якою жоден із запропонованих кадрів не скидається пристроєм». Це спосіб кількісної оцінки потоку трафіку, який може обробити мережеве з'єднання. Пропускна здатність вимірюється загальнодоступною командою netperf.

Туди й назад. Час проходження в обидві сторони кількісно визначає відповідь, яку пропонує мережеве підключення. Він буде вимірюватися перед пропускною спроможністю через ті самі з'єднання, що й пропускна здатність. Час зворотного зв'язку вимірюється системною командою ping. Це виражається тут як кількість повністю активних процесів на хості. Це не параметр мережі, але він може допомогти пояснити несподіване зниження продуктивності. Навантаження вимірюється на поточному хості за допомогою системної команди uptime.

Презентація результатів базується на Інтернеті з використанням Java-аплету для завантаження даних із файлів у пам'ять веб-браузера від користувача, який аналізує результати.

Iperf — це інструмент для вимірювання максимальної пропускної здатності TCP, що дозволяє налаштовувати різні параметри та характеристики UDP. Він повідомляє про пропускну здатність, тремтіння затримки та втрату дейтаграм. Iperf широко використовується, однак [R17] описує результати використання iperf між декількома інститутами HEP і тому надає гарний приклад його використання [4].

IperfER було розроблено на основі програмного забезпечення PingER, але замінює вимірювання RTT і втрати пакетів на основі ping вимірюванням пропускної здатності TCP із використанням інструменту iperf. Графічний вихід iperfer дуже схожий на pinger, а показники пропускної здатності стають доступними для проміжного програмного забезпечення через LDAP у спосіб, який відповідає pinger.

UDPmon дає оцінку максимальної використаної смуги пропускання між двома кінцевими вузлами, вимірювання втрати пакетів і тремтіння пакетів або варіації часу надходження між послідовними пакетами. Це тремтіння пакетів є оцінкою варіацій в односторонніх затримках пакетів, що проходять мережею. UPDMon був розроблений Річардом Хьюз-Джонсом (PPARC) у рамках WP7, і,

оскільки немає жодного конкретного посилання, що описує його роботу, тут надається короткий опис.

UDPmon використовує дві програми: слухач під назвою udp_bw_resp, який отримує вхідні тестові дані, і програму моніторингу під назвою udp_bw_mon, яка виконує перевірку на віддаленому хості.

У тесті використовуються кадри UDP/IP із прикладним протоколом, показаним на малюнку 1.2. Тест починається з того, що вузол запиту надсилає повідомлення «Очистити статистику» відповідачу. Отримавши підтвердження ОК, запитуючий вузол надсилає серію пакетів «даних», розділених заданим фіксованим інтервалом часу між пакетами. Наприкінці тесту запитуючий вузол запитує статистику, зібрану відповідаючим вузлом. Втрата пакетів для контрольних повідомлень обробляється відповідними тайм-аутами та повторними спробами у вузлі запиту. Якщо тест уже виконується, коли до Відповідача надходить повідомлення «Очистити статистику», запитувачу надсилається повідомлення «Будь ласка, відкладіть». Процедура відкладення запобігає спотворенню вимірювань пропускної здатності кількох одночасних тестів.

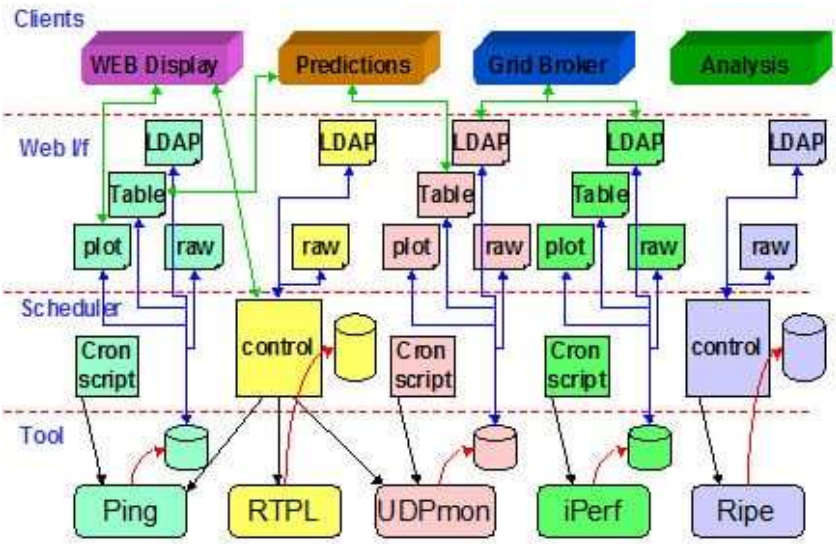

Рисунок 1.2 - Протокол для вимірювання пропускної здатності та втрати пакетів

Пропускна здатність передачі визначається за кількістю переданих даних і витраченим часом; пропускна здатність прийому розраховується на основі кількості отриманих даних і часу від першого пакета даних до останнього отриманого пакета.

Втрата пакетів вимірюється віддаленим вузлом шляхом перевірки правильності збільшення порядкових номерів у пакетах; це також виявляє пакети, що не відповідають порядку. Кількість побачених пакетів, кількість пропущених пакетів, як зазначено під час перевірки порядкового номера, і кількість непорядкових пакетів повідомляються в кінці кожного тесту.

Відповідаючий вузол також вимірює час між надходженням послідовних пакетів і створює гістограму цих часів. Цю гістограму може запитати Запитуючий вузол наприкінці тесту. Довжина черги в мережі може бути досліджена шляхом порівняння додаткового часу, необхідного для отримання пачки пакетів.

Інструмент UDPmon був повністю інтегрований в архітектуру моніторингу WP7 і забезпечений сценаріями Perl для планування тестів, створення хронологічних графіків і таблиць часової послідовності. Було надано серверний сценарій LDAP, щоб поточний знімок можна було отримати для публікації в проміжному програмному забезпеченні.

MapCenter створено для відображення гнучкого рівня представлення служб і програм у сітці. Сучасні технології моніторингу мають чудові функціональні можливості та зберігають різні та точні результати в «інформаційній системі» Grid, але загалом не існує ефективних способів графічного представлення всіх спільнот, організацій, програм, які працюють через Grid. MapCenter був розроблений, щоб заповнити цю прогалину.

MapCenter опитує обчислювальні елементи (об’єкти) різними методами, наприклад, для надсилання запитів ICMP (ping) для перевірки зв’язку обчислювальних елементів; здійснювати підключення TCP до призначених портів для перевірки служб, що працюють на обчислювальних елементах; і зможе надіслати запит до Інформаційних систем мереж, щоб перевірити наявність конкретних послуг мереж. MapCenter здатний відображати різні види сіток,

наприклад, через графічні карти; з логічними видами послуг; і з повним деревом обчислювальних елементів.

MapCenter справді зосереджений на рівні презентації у величезному та неоднорідному середовищі, як у контексті Grid. Він пропонує дуже гнучку та просту модель, яка дозволяє представляти будь-який рівень абстракції (національні та міжнародні організації, віртуальні організації, програми тощо), який необхідний таким середовищам. В принципі, MapCenter можна розширити для моніторингу та представлення інших показників. MapCenter був розроблений Franck Bonnassieux (CNRS) у рамках WP7.

Метою проекту Test Traffic є забезпечення незалежних вимірювань параметрів підключення, таких як затримки та вектори маршрутизації, в Інтернеті. Проект реалізує метрики, які обговорювалися в робочій групі IETF IPPM. Робота над цим проектом почалася в квітні 1997 року, і протягом останніх років було показано, що установка здатна регулярно вимірювати затримки, втрати та вектори маршрутизації у великих масштабах. Проект Test Traffic переміщується до послуги, яку RIPE NCC пропонує всій спільноті.

Surveyor — це вимірювальна інфраструктура, яка зараз розгортається на учасниках по всьому світу. Він базується на роботі зі стандартами, яка виконується робочою групою IETF IPPM. Surveyor вимірює ефективність Інтернет-шляхів серед організацій-учасниць. Проект також розробляє методології та інструменти для аналізу даних про продуктивність.

Одностороння затримка та втрата пакетів вимірюються для кожного шляху шляхом надсилання тестових пакетів із міткою часу з одного кінця зазначеного шляху до іншого. Затримка в один бік обчислюється шляхом віднімання мітки часу в пакеті з часу надходження на машину призначення.

Національна інфраструктура вимірювання Інтернету (NIMI) — це проект, розпочатий Національним науковим фондом США та наразі фінансований DARPA, для вимірювання глобального Інтернету. Він заснований на програмі Network Probe Daemon і розроблений як масштабований і динамічний. NIMI є масштабованим у тому сенсі, що зонди NIMI можуть бути делеговані адміністраторам для інформації

про конфігурацію та координації вимірювань. Він динамічний, оскільки інструменти вимірювання є зовнішніми, як сторонні пакети, які можна додавати за потреби.

Multi Router Traffic Grapher (MRTG) — це пасивний інструмент для моніторингу навантаження трафіку на мережеві посилання. MRTG генерує HTML-сторінки, що містять зображення, які забезпечують живе візуальне представлення цього трафіку. MRTG складається зі сценарію Perl, який використовує SNMP для зчитування лічильників трафіку маршрутизаторів, реєстрації даних трафіку та створення графіків, що представляють трафік під час контрольованого мережевого з'єднання. Ці графіки вбудовані у веб-сторінки, які можна переглядати з будь-якого сучасного веб-браузера.

Трасування використовує втрату пакетів як показник якості мережі. У кожному дослідженому випадку було виявлено, що зміни у втраті пакетів, зареєстровані traceroute і ping, а отже, і Traceping, відображають зміни в реальній продуктивності на рівні користувача. Систематичні чи випадкові помилки в даних про втрату пакетів не вимірюються та не оцінюються. Цифри слід інтерпретувати просто як якісні показники стану мережевого з'єднання: чим вище число, тим нижча якість. Наразі трасування є спеціальним інструментом VMS, тому його значення суворо обмежене, хоча є пропозиція зробити його більш загальним.

Cflowd було розроблено для збору та аналізу інформації, доступної з NetFlow flow-export. Це дозволяє користувачеві зберігати інформацію та дає змогу переглядати дані в кількох режимах. Він створює матриці портів, матриці AS, мережеві матриці та чисті структури потоку. Обсяг даних, що зберігаються, залежить від конфігурації cflowd і варіюється від кількох сотень Кбайт до сотень Мбайт за один день на маршрутизатор.

NWS — це розподілена система, яка періодично відстежує та динамічно прогнозує продуктивність, яку різні мережеві та обчислювальні ресурси можуть забезпечити протягом певного інтервалу часу. Служба керує розподіленим набором датчиків продуктивності (мережевих моніторів, моніторів процесора тощо), з яких збирає показання миттєвих умов. Потім він використовує числові

моделі для створення прогнозів того, якими будуть умови протягом заданого періоду часу. Ця функція аналогічна прогнозу погоди, тому система успадкувала назву.

NWS забезпечує інклюзивне середовище, до якого можна додавати додаткові датчики. Хоча його можна розглядати та використовувати лише як засіб надання прогнозних даних на основі поточних вимірювань метрики, насправді він може забезпечити повне, самодостатнє середовище моніторингу.

NetSaint — це програма, яка відстежуватиме хости та служби в мережі. Він має можливість надсилати електронний лист або сторінку, коли виникає проблема та коли її вирішують. NetSaint написаний на C і призначений для роботи під Linux, хоча він повинен працювати в більшості інших варіантів Unix. Він може працювати або як звичайний процес, або як демон, періодично запускаючи перевірки різних указаних служб. Фактичні перевірки служби виконуються зовнішніми «плагінами», які повертають інформацію про службу NetSaint. Кілька програм CGI включено до NetSaint, щоб дозволити переглядати поточний стан служби, історію тощо через веб-браузер.

Очікується, що грід-додатки записуватимуть дані про пропускну здатність, отримані під час нормальної роботи, і нададуть їх доступ разом із даними, зібраними іншими інструментами моніторингу мережі. Grid ftp уже має необхідні можливості для запису такої інформації під час кожної передачі. Ця інформація стане основним компонентом моніторингу мережі, зареєстрованого Grid, але інформацію, яку вона надає, потрібно ретельно інтерпретувати та порівнювати з результатами активного моніторингу Grid.

GridFTP є розширенням стандартного механізму FTP для використання в середовищі Grid. Передбачається, що він буде використовуватися як стандартний протокол для передачі даних. Пропускна здатність додатка для потоку визначає пропускну здатність, виміряну для певної передачі GridFTP між указаними кінцевими точками. Він базуватиметься на «пасивному» вимірюванні переданих даних, тобто лише переданої інформації, а не додаткових (тестових) даних. Дані, визначені для зберігання щодо передачі GridFTP, такі:

1) Джерело;
2) Пункт призначення;
3) Загальна кількість байтів/розмір файлу;
4) Кількість використаних потоків;
5) Розмір буфера TCP;
6) Сукупна пропускна здатність;
7) Швидкість передачі (Мбайт/с);
8) Перенесення часу розпочато;
9) Перенесення часу завершено;

Було запропоновано схему, яка використовує виправлену версію GridFTP, у якій записуються передачі та зберігаються підсумкові дані. Це не включено тут, оскільки дебати щодо точного формату тривають. Проте є надія, що коли буде сформовано стандартну форму схеми моніторингу мережі, обидві схеми будуть уніфіковані.

У майбутніх випусках GridFTP пропонується включити такі функції, як автоматичне узгодження розмірів TCP-буфера/вікна та паралельна передача даних, а також надійна передача даних.

Щоб забезпечити вимірювання обсягу даних, переданих у мережу, як функцію часу (щодня/щотижня/місяця/року), було запропоновано, щоб інформація від передачі даних через мережу агреговалась у міру передачі. Якщо GridFTP використовується для передачі даних, то різні змінні повинні бути отримані з вимірювань GridFTP і просто додані, інакше програмне забезпечення, яке використовується для передачі, повинно враховувати цю вимогу. Це може включати або не включати дані, передані з активних (тестових) даних.

Альтернативним підходом є визначення обсягу Grid на основі лічильників, доступних у мережевих пристроях, наприклад, через SNMP. Це вимагає ідентифікації трафіку Grid з набору трафіку, що передається мережевими пристроями, і потребує подальшого дослідження.

Розроблено та продемонстровано окремі компоненти, необхідні для архітектури моніторингу мережі. Сайти тестового стенда моніторингу мережі WP7

використовувалися для демонстрації роботи різноманітних інструментів моніторингу мережі та їх здатності збирати дані та робити метрики моніторингу мережі доступними через веб-інтерфейс. Окремо було продемонстровано сценарії, які витягують мережеві показники зі сховища даних мережевого монітора та роблять їх доступними через службу LDAP. Комбіновані можливості були продемонстровані у випуску, який було встановлено в тестовому стенді DataGrid1. У цьому випуску стали доступними показники часу проходження туди й назад і втрати пакетів.

Цей прототип поставки включає публікацію RTT, втрату пакетів і пропускну здатність TCP і UDP через служби LDAP і Інтернет. Крім того, було надано кілька інструментів, які вимірюють і звітують за тими самими показниками. У схемі LDAP буде доступним вимірювання «за замовчуванням», а також вимірювання, специфічне для конкретного інструменту моніторингу. Мета тут полягає в тому, щоб продемонструвати розширюваність архітектури; забезпечити простий засіб перевірки результатів певного інструменту; і створити інший «вигляд і відчуття» процесу візуалізації. Ця архітектура показана на малюнку 1.3, де можна побачити взаємозв’язок між компонентами продукту моніторингу мережі.

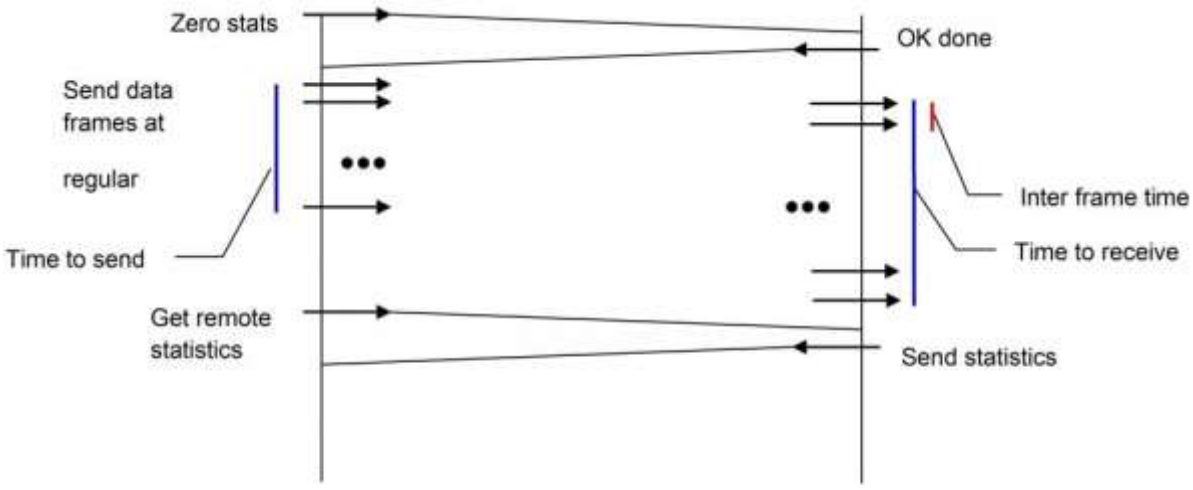

Рисунок 1.3 - Схематичне зображення архітектури моніторингу мережі, що демонструє компоненти та їх взаємозв’язок.

По суті, результати моніторингової діяльності можуть бути розділені за часом. Негайний результат моніторингу надає моментальний знімок існуючих умов

у мережі, тоді як історичні дані можна використовувати, щоб переглянути поведінку мережі за дні, тижні та місяці. Ці два способи моніторингу мережі призначені для дуже різних цілей. Можна передбачити, що перше буде використано або безпосередньо кінцевим користувачем, який бажає оптимізувати роботу певної програми, або, що більш імовірно, самою програмою через проміжне програмне забезпечення для налаштування в режимі реального часу своїх використання мережевих ресурсів; в той час як остання використовуватиметься для управління мережею, перегляду тенденцій у використанні та забезпечення наявності достатнього забезпечення для задоволення попиту.

Щоб отримати моніторингову інформацію про мережу, необхідно провести тести, які можна розділити на дві категорії.

Активний моніторинг мережі відбувається, коли тестові дані запускаються безпосередньо через мережу, щоб виявити властивості наскрізного з'єднання. Трафік, створений таким тестуванням, є доповненням до звичайного трафіку в мережі. Такий підхід використовує різноманітні інструменти моніторингу мережі та може бути належним чином запланований, щоб мінімізувати вплив на користувачів мереж, забезпечуючи при цьому точне вимірювання конкретної метрики мережі.

Пасивний мережевий моніторинг використовує реальні програми та їхній трафік для запису роботи програми під час використання мережі. Так, наприклад, Grid ftp можна використовувати для запису пропускної здатності реального трафіку Grid у мережі та аналогічно з іншими програмами. Це вигідно тим, що в мережі не вводиться додатковий трафік, але планування відображатиме досвід користувачів у виконанні певного завдання, і тому може неточно фіксувати можливості мережі.

Крім того, будь-які проблеми з обслуговуванням, пов'язані з аспектами моніторингу програми, залежать від виправлення супроводжувачів самої програми Grid.

Продукт PM12 підтримує лише активні форми моніторингу, коли інформація про мережеві показники з процесів моніторингу стає доступною у формі, яка дозволяє опублікувати її в проміжному програмному забезпеченні через службу

LDAP. WP7 тісно співпрацює з розробниками Grid ftp, щоб гарантувати, що в майбутньому такі додатки записуватимуть відповідну інформацію та, за наявності, відповідатимуть безпосередньо описаній тут архітектурній моделі.

Ця початкова реалізація використовує схему запитів/відповідей для запиту інформації про моніторинг мережі, зібраної з журналів і збереженої на сервері LDAP. Загалом це «механізм витягування», за допомогою якого інформація моніторингу зберігається локально, а служба LDAP періодично оновлює свою інформацію, вибираючи метрики моніторингу мережі. У майбутньому також буде розглянуто певну форму автоматичних потокових оновлень через «механізм натискання», коли інструмент моніторингу періодично робить інформацію про мережеву метрику доступною для служби LDAP.

Існує занепокоєння щодо доцільності використання «механізму проштовхування» в реальних GRID-середовищах, оскільки сервери можуть бути переповнені такою кількістю інформації, що у них буде мало ресурсів, щоб робити щось інше. Це було визначено як тему для майбутньої роботи.

1.2 Прогнозна аналітика для передбачення проблем мережі

Прогнозна аналітика може бути корисним інструментом для передбачення проблем мережі та запобігання їх виникненню. Вона використовує дані про продуктивність мережі, попередній досвід та статистичні методи для прогнозування майбутніх подій та проблем.

Виявлення та визначення потенційних збоїв у мережі та проблем із продуктивністю вже давно є предметом обґрунтованих припущень, але нове покоління інструментів прогнозної аналітики обіцяє забезпечити більшу точність надійних прогнозів мережі, дозволяючи персоналу вирішувати та виправляти конкретні проблеми ще до того, як вони навіть можуть почати впливають на роботу мережі. Прогнозна аналітика є потужним інструментом, який допомагає вирішувати цю задачу.

Прогнозна аналітика використовує дані про продуктивність мережі, попередній досвід та статистичні методи для прогнозування майбутніх проблем. Аналізуючи історичні дані про пропускну здатність, завантаження, затримки та інші метрики, можна виявити патерни та тренди, що покажуть, які проблеми можуть виникнути у майбутньому. Наприклад, зростання завантаження мережі протягом певного періоду часу може вказувати на можливу перевантаженість або зниження продуктивності в майбутньому.

Для передбачення проблем мережі можуть використовуватись різні методи та підходи. Статистичні моделі, такі як регресія та часові ряди, дозволяють враховувати різні фактори та залежності, що впливають на продуктивність мережі. Вони дозволяють побудувати математичні моделі, які передбачають майбутні стани мережі на основі історичних даних.

Машинне навчання також використовується для прогнозування проблем мережі. Це дає можливість аналізувати великі обсяги даних та виявляти складні залежності, які не завжди можуть бути виявлені за допомогою традиційних статистичних методів. Моделі машинного навчання, такі як класифікація та кластеризація, можуть розпізнавати патерни та залежності в даних, що допомагає передбачати проблеми мережі з високою точністю.

Крім того, автоматизовані системи моніторингу та аналітики в реальному часі дозволяють виявляти аномалії та надавати оперативну інформацію щодо проблем мережі. Вони аналізують дані мережі в реальному часі та надають рекомендації для вирішення проблем.

Прогнозна аналітика допомагає не лише передбачати проблеми мережі, але й запобігати їх виникненню. За допомогою аналізу даних та статистичних методів можна виявити потенційні проблеми та прийняти заздалегідь заходи для їх попередження. Наприклад, якщо аналітика показує, що навантаження мережі зростає, можна прийняти рішення про розширення пропускної здатності або оптимізацію ресурсів мережі.

Прогнозна аналітика також допомагає зменшити витрати на обслуговування та ремонт мережі. За допомогою передбачення проблем можна планувати технічне

обслуговування та ремонт заздалегідь, що дозволяє запобігти виникненню серйозних проблем та знизити витрати на них.

Прогнозна аналітика для передбачення проблем мережі має свої обмеження. Вона базується на аналізі даних та статистичних методах, тому завжди існує певний ризик помилки або невизначеності. Крім того, прогнозування не може враховувати непередбачувані події або зміни у зовнішніх факторах, що можуть вплинути на мережу.

У підсумку, прогнозна аналітика є важливим інструментом для передбачення та запобігання проблемам мережі. Вона використовує дані, статистичні методи та моделі машинного навчання для аналізу мережі та передбачення майбутніх проблем. Це дозволяє забезпечити більшу надійність та продуктивність мережі, зменшити витрати на обслуговування та ремонт, а також покращити задоволення користувачів.

1.3 Автоматизована діагностика проблем мережі

Існує багато систем для моніторингу та аналізу продуктивності мережевих систем. Деякі вимагають інвазивних змін у вихідному коді програмного забезпечення приладу та відстеження окремих повідомлень, які надсилаються системою. Хоча це може допомогти розробникам і операторам відстежувати незначні помилки та проблеми з продуктивністю, необхідні зміни коду створюють високий бар’єр для входу, особливо під час моніторингу системи сторонніх розробників, для якої немає вихідного коду.

Інші підходи аналізують мережеві захоплення пакетів, щоб спробувати зробити висновок про стани важливих елементів системи. Хоча захоплення пакетів можна здійснювати без впливу на продуктивність або вихідний код контрольованої системи, вони надто дорогі для безперервного запуску та аналізу, і за своєю природою вони містять мало інформації, коли система припиняє передачу даних. Коли трафік припиняється, можливо, програмне забезпечення зупинилося, кожне

з’єднання транспортного рівня (TCP) виявило перевантаження мережі та припинило повторні передачі, або система завершила всю свою поточну роботу. Без моніторингу кінцевих господарів важко достовірно розрізнити причину та наслідок.

Багато складних моніторів збирають дані з усієї мережі для виявлення системних проблем. Ці системи здатні знаходити та виявляти широкий спектр неправильної поведінки мережі, програмного забезпечення та системи, але здебільшого покладаються на складні аналізи, які важко відтворити, і малокорисні для одного хоста чи кінцевого користувача. Інший поширений підхід полягає у виконанні протокольно-специфічного аналізу для виявлення проблем із продуктивністю, що виявляються конкретними мережевими технологіями. Ці аналізи можуть бути безцінними для відстеження складних і тонких проблем у сучасних системах систем. Однак застосувати ці специфічні для протоколу знання до нових проблемних доменів непросто. Однак застосування цих конкретних ідей протоколу до нових проблемних областей може бути складним у реалізації, дорогим у виконанні та в багатьох випадках неможливим узагальненням.

Підхід DYSWIS покладається на однорангові вузли для визначення основної причини збою.

Виявивши збій, вузол запитує однорангові вузли, чи вони також спостерігають за збоєм. Однорангові вузли, ґрунтуючись на минулому досвіді роботи з тією самою службою або на основі зонду, роблять висновок, що цей збій локальний для вузла. У деяких випадках збій може бути локальним для підмережі, комутатора доступу, точки доступу або домену. Іншими словами, локальність збою може поширюватися від самого вузла до всього домену. Інфраструктура діагностики може запросити кілька однорангових вузлів про певну службу, щоб локалізувати проблему.

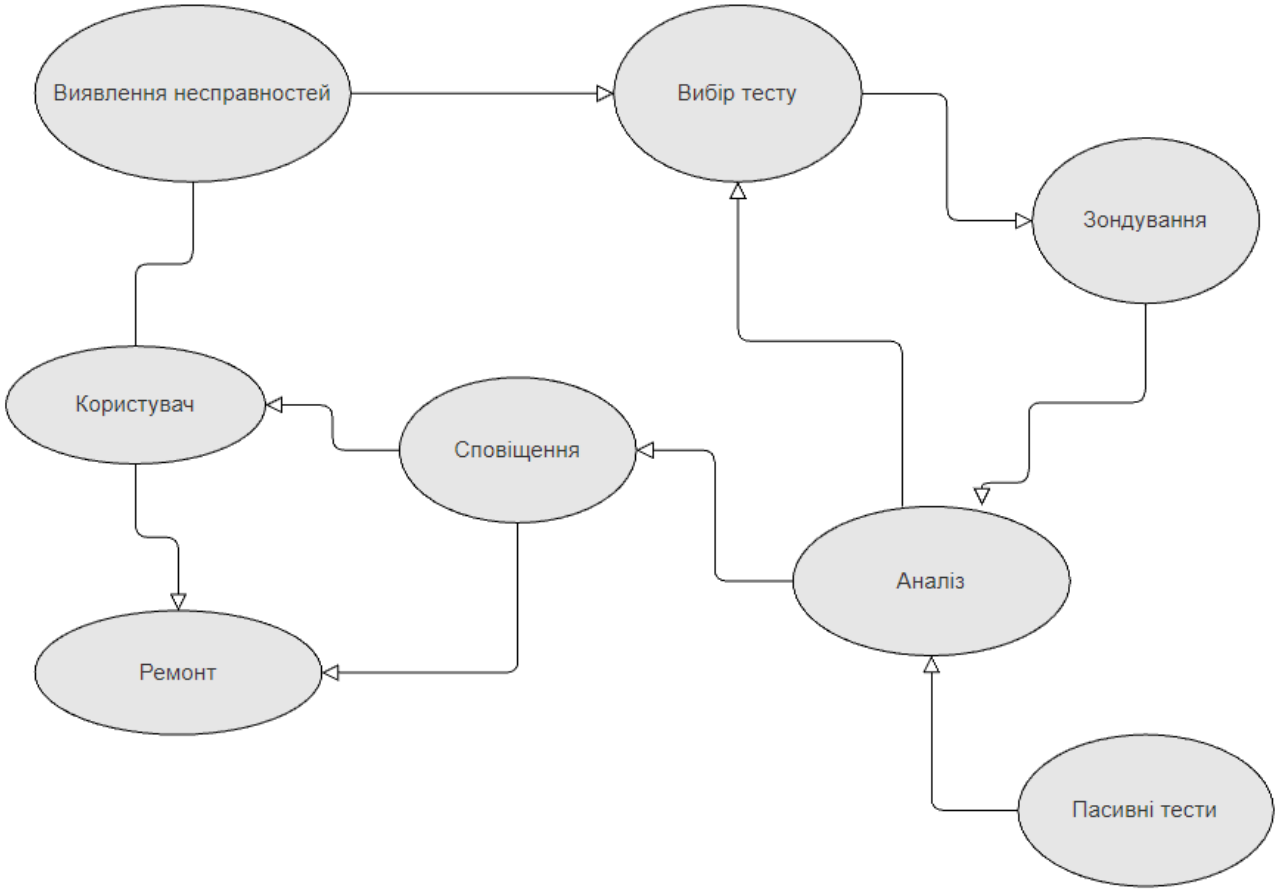

Рисунок 1.4 - Системний хід діагностичного процесу

Архітектура запропонованої системи діагностики несправностей складається з наступних основних функціональних компонентів: інфраструктура виявлення та звітування про несправності, попередня діагностична обробка та вибір діагностичних тестів і, нарешті, діагностичні тести, аналіз результатів і збереження історичних результатів. Першим кроком у процесі діагностики є виявлення несправностей і повідомлення про них. Система діагностики несправностей повідомляє про виявлені користувачем і автоматично програмно виявлені збої для аналізу в систему діагностики несправностей. Звіти про помилки включають детальну контекстну інформацію про помилку, наприклад, відстань одного стрибка на різних рівнях OSI, наприклад, інформацію про точку доступу або комутатор на рівні 2, інформацію про маршрутизатор за замовчуванням або підмережу на рівні 3, проксі-сервер SIP для першого стрибка сервер на прикладному рівні, мітка часу, коли спостерігається збій, ім’я хоста та IP-адреси вузла-учасника.

Щоб пояснити, як працює система DYSWIS, розглянемо систему VoIP (рис. 1.5). Збій, який бачить користувач, наприклад, збій встановлення виклику, може бути через збій мережі доступу, неправильну роботу NAT або збій на проксі-сервері SIP або збій автентифікації на сервері STUN. Кожна з цих помилок може бути спричинена неправильною конфігурацією, програмними помилками, перевантаженням сервера чи мережі чи іншими тимчасовими проблемами в мережі. Крім того, в мережі є повний набір допоміжних служб, таких як DHCP, ARP і DNS.

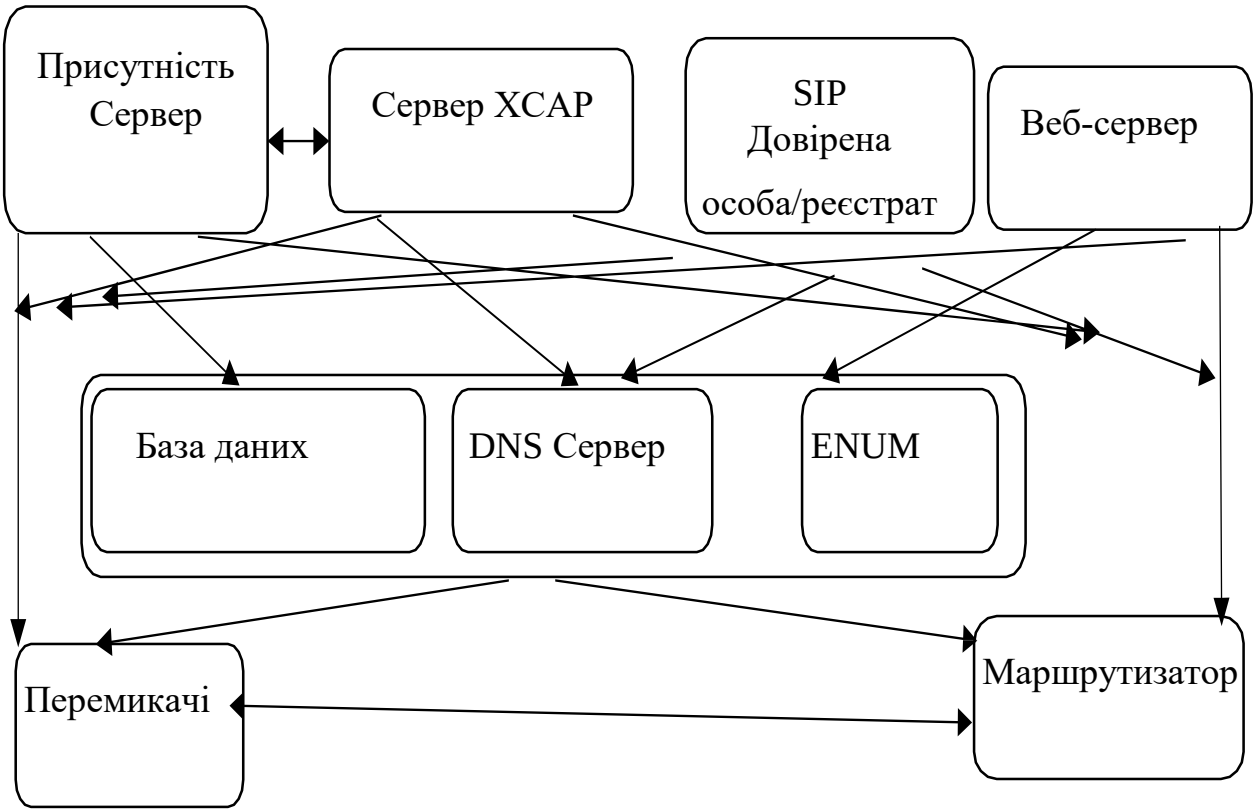

Рисунок 1.5 - Взаємодія різних компонентів мережі

Розглянемо користувача «alice@example.com» намагається зателефонувати іншому користувачеві «bob@destination.com» і дзвінок не проходить. Кінцева точка, яка намагалася здійснити виклик до віддаленої кінцевої точки, запускає систему DYSWIS для виконання діагностики, яка виконує такі дії:

1) Діагностичний вузол запитує, чи будь-який інший вузол із розташування абонента зробив виклик користувачевivian@destination.com . У цьому випадку розташування може означати ту саму підмережу, VLAN, комутатор/точку доступу або домен;
2) Відповідь може полягати в тому, що інші вузли нещодавно здійснили виклик на ту саму адресу призначення або в домен призначення, а не на ту саму адресу призначення або жоден вузол не зробив жодного виклику на адресу призначення або домен призначення. Слід зазначити, що історична інформація про успіх або невдачу запитується з урахуванням розташування спостережуваної невдачі (отже, з використанням топології) разом із функціональними залежностями;
3) На основіпісля відповіді діагностичний вузол може запросити інший вузол надіслати повідомлення SIP OPTION на адресу призначення або він може запросити здійснити виклик до тестового вузла;

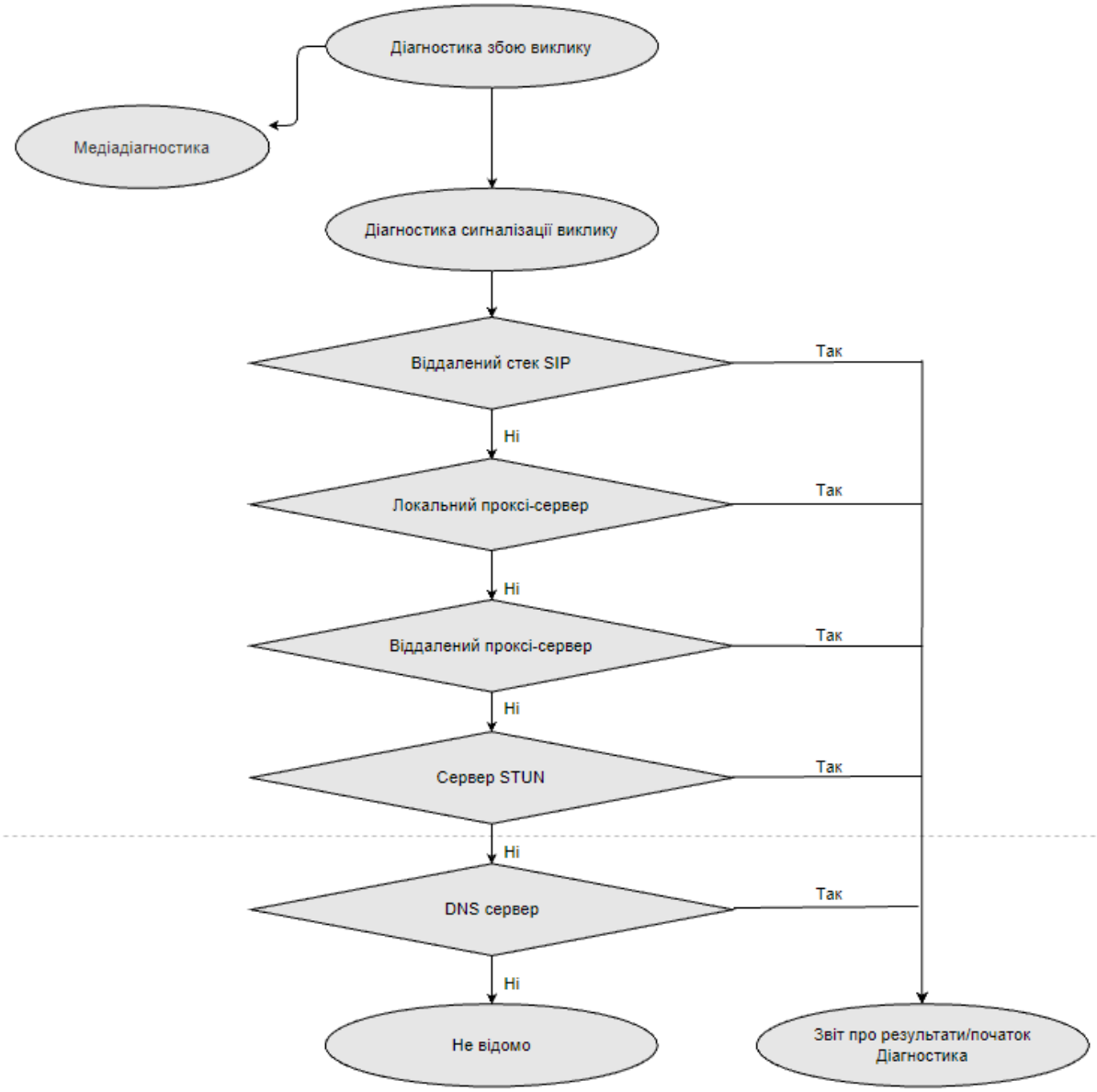

Рисунок 1.6 - Діагностика збою виклику

Діагностика збою виклику включає в себе діагностику сигналізації виклику, яка, у свою чергу, включає тестування віддаленої кінцевої точки SIP, локального проксі-сервера, сервера STUN і сервера DNS. Це може додатково ініціювати діагностику підключення до мережі та доступності підтримуваних протоколів (служб). Звіти зберігаються та використовуються для визначення порядку запитів для майбутніх помилок. Наприклад, якщо повідомляється про збій виклику до того самого адресата, і для цієї помилки вже було проведено діагностику, більше тестування проводитися не буде.

Навіть добре написане програмне забезпечення, таке як веб-сервер Apache, може відчувати раптові стрибки затримки запиту через блокування головного рядка для доступу до диска, змагання за спільні ресурси, записи на диск або запити до бази даних. Незалежно від того, чи викликані ці зупинки через помилки, неефективні механізми блокування або звернення до внутрішніх серверів баз даних, вони заважають прикладному програмному забезпеченню своєчасно відповідати на запити. Іншим поширеним джерелом падіння продуктивності є перевантаження мережі. Дослідження 2011 року щодо мережевого трафіку користувачів у двох центрах обробки даних Google виявило що втрата пакетів і повторна передача є досить поширеними: 2,5–5,6% усіх TCP-з'єднань користувача повторно передають пакети.

Хоча швидка повторна передача та вибіркові підтвердження (SACK) можуть уникнути повної зупинки пропускної здатності, коли відбувається втрата пакетів, дослідження також показало, що приблизно 1% усіх з'єднань зупиняється щонайменше на 200 мс через тайм-аут повторної передачі (RTO). Навіть якщо TCP вдається уникнути тайм-аутів повторної передачі, будь-яка повторна передача коштує дорого: короткі веб-запити займають у середньому в 7–10 разів більше часу, коли з'єднання TCP повторно передає будь-які пакети. У сучасних розподілених програмах, здавалося б, рідкісні події можуть мати значний вплив на час реакції.

Коли додатки залежать від десятків або сотень окремих служб для своєчасної відповіді, викиди в довгому хвості розподілу затримок не такі вже й рідкісні. Програми електронної комерції Amazon можуть звертатися до 150 служб, щоб відповісти на один запит, причому кожна транзакція може мати низьку пропускну здатність або тимчасову зупинку через перевантаження мережі, серверну обробку, конкуренцію за дисковий ввід/вивід, довший, ніж зазвичай, запит до бази даних. , або тимчасова проблема в мережі.

Кожна служба має відповідати угоді про рівень обслуговування (SLA), як правило, для виконання 99,9% транзакцій менш ніж за 300 мс. Навіть якщо припустити, що ці 150 транзакцій абсолютно паралельні, лише 86% запитів буде виконано протягом 300 мс; 1 серіалізована транзакція збільшує затримку.

1.4 Проактивна оптимізація мережі

Проактивна оптимізація мережі є важливим завданням для організацій та постачальників послуг зв'язку. Цей підхід передбачає вжиття заходів з покращення продуктивності та надійності мережі перед виникненням проблем. Проактивна оптимізація мережі дозволяє забезпечити кращу якість обслуговування, знизити витрати та підвищити задоволення користувачів.

Проактивна оптимізація мережі: прогнозовані мережеві показники якості (KPI) передаються до цієї функції. Оптимізаційний алгоритм передбачає налаштування параметрів заздалегідь для очікуваних змін у стані мережі з метою досягнення поставлених цілей. У статті [55] пропонується приклад оптимізації мережі у прогнозуючий та енергоефективний спосіб. Представлена рамка реалізує розумну платформу, що враховує великі обсяги даних, між основною мережею та пулом базових блоків (BBU) для аналізу поведінки користувачів та мережевих шаблонів з метою виведення стратегій керування. Слід зауважити, що ці аналітичні дані та висновки можуть бути виконані як на віддаленому хмарному рівні, так і на рівні краю мережі. Обчислення в хмарному середовищі можуть створити загальний контекст загальних поведінок, таких як рух транспорту в місті в години пік, що вказує на макроскопічну оптимізацію. У той же час, обчислення на рівні краю обробляють персоналізований контекст з допомогою краєвих дата-центрів, краєвих тензорів, керування та аналізу даних на рівні краю. Нарешті, конфігурація мережі у фізичному просторі приймається з урахуванням як загального тенденційного, так і індивідуального контексту.

Одним з основних аспектів проактивної оптимізації мережі є моніторинг та аналіз її продуктивності. Це включає збір даних про пропускну здатність, завантаження, затримки, втрати пакетів та інші показники продуктивності мережі. Ці дані дозволяють зрозуміти, як мережа працює в реальному часі та виявити потенційні проблеми, що можуть виникнути в майбутньому. Наприклад, якщо дані показують, що певні сегменти мережі перевантажені або мають високі затримки,

можна прийняти заходи для розширення пропускної здатності або оптимізації маршрутизації.

Аналіз даних також допомагає виявити патерни та залежності в роботі мережі. Застосування методів машинного навчання та штучного інтелекту дозволяє виявити складні залежності, що можуть бути непомітними для людського аналізу. Наприклад, можна використовувати класифікацію або кластеризацію для виявлення аномалій у поведінці мережі або для прогнозування майбутніх проблем на основі історичних даних. Це дозволяє вжити запобіжних заходів та оптимізувати мережу, щоб уникнути виникнення проблем у майбутньому.

РОЗДІЛ 2 ВИЗНАЧЕННЯ МЕТОДОЛОГІЇ

2.1 Вибір методів та інструментів

Продуктивність мережі та програми вимірюється кількома показниками продуктивності, такими як пропускна здатність, пропускна здатність, дисковий час, використання ЦП, кількість пакетів, що надсилаються/отримуються за секунду, і кількість байтів, які надсилаються/отримаються за надсилання.

Тестовий стенд, який використовується в реалізації, має кілька інструментів для збору показників мережі та додатків. Показники, зібрані на обох хостах і каналах зв'язку між двома хостами.

У наведених нижче таблицях показано показники продуктивності, класифіковані за використовуваним інструментом:

У малюнку нижче показано показники продуктивності, класифіковані за використовуваним інструментом:

Рисунок 2.1 - Метрики продуктивності мережі та програми для кожного інструменту

На малюнку нижче показано показники продуктивності, класифіковані за точкою вилучення показника:

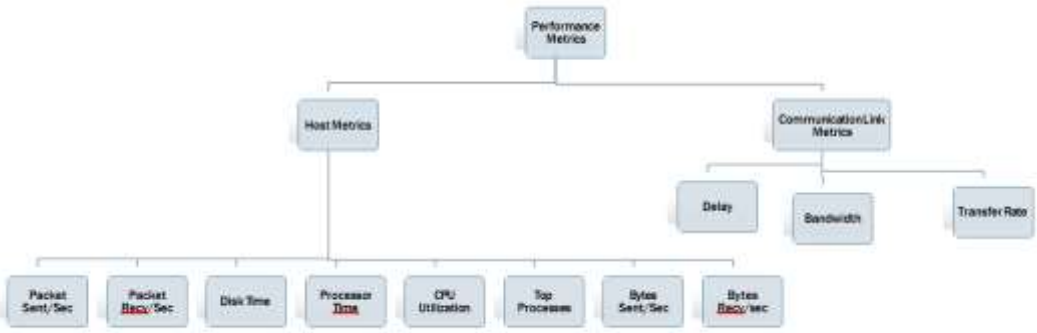

Рисунок 2.2 - Показник продуктивності для точки вилучення

У таблиці нижче наведено визначення показників продуктивності мережі та програми:

Таблиця 2.1 - Визначення показників

Назва показника	**Визначення**
% дискового часу	Відсоток часу, що минув, протягом якого вибраний диск був зайнятий обслуговуванням запитів на читання або запис.
Прийом пакетів/сек	Швидкість, з якою пакети надходять на мережевий інтерфейс.
Надіслано пакетів/сек	Швидкість, з якою пакети надсилаються через мережевий інтерфейс.
% процесорного часу	Відсоток часу, що минув, який процесор витрачає на виконання потоку, що не є Idle. Він обчислюється шляхом вимірювання відсотка часу, який процесор витрачає на виконання неактивного потоку, а потім віднімання цього значення від 100%.
Топ-процеси	Укажіть назви п’яти найпопулярніших процесів, які споживають більше ЦП.
Завантаження ЦП	Укажіть використання процесора для кожного процесу.
Затримка	Час, витрачений на те, щоб пакет пройшов від джерела до місця призначення і назад.
Пропускна здатність	Кількість біт за секунду, які надсилаються за посиланням.
Швидкість передачі	Кількість байтів, що передаються за секунду за посиланням.
Байт Recv/сек	Швидкість, з якою байти надходять через кожен мережевий адаптер, включаючи кадрові символи. Мережевий інтерфейс\Отримані байти/сек — це підмножина Мережевий інтерфейс\Загальна кількість байтів/сек.
Надіслані байти/сек	Швидкість, з якою байти надсилаються через кожен мережевий адаптер, включаючи кадрові символи. Network Interface\Bytes Sent/sec є підмножиною Network Interface\Bytes Sent/sec.

Інструмент моніторингу продуктивності, який використовується для перегляду даних продуктивності в реальному часі або створення файлів журналів. Інструмент моніторингу продуктивності є вбудованим інструментом у Windows і не потребує встановлення. Вимогою для отримання даних метрики продуктивності є налаштування набору збирача даних. Вихідні дані збирача даних налаштовано для створення файлу, розділеного комами, який використовуватиметься в подальшому аналізі. Створення набору збирача даних і його конфігурація шляхом

додавання певних показників вимагає збору. Дослідження зосереджено на зборі показників, пов’язаних із додатками та продуктивністю мережі.

Рисунок 2.3 - Інструмент моніторингу продуктивності

Метрики, додані до набору збирача даних:

1) Прийом пакетів/сек;
2) Пакетів, надісланих/сек;
3) Байт Recv/сек;
4) Надіслані байти/сек;
5) Час процесора;
6) Дисковий час.

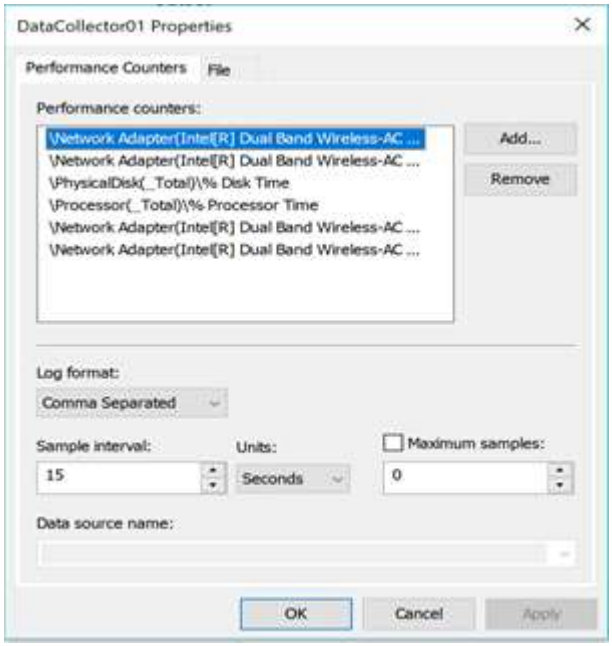

Рисунок 2.4 - Набір збирача даних

На малюнку нижче показано вихідний файл журналу з Performance Monitor Tool:

Time Stamp	Bytes Received/sec	Bytes Sent/sec	Packets Received/sec	Packets Sent/sec	% Disk Time	% Processor Time
02/19/2019 19:31:27.707	4016613.253	67625.20193	2982.737359	1062.155092	0.047004846	0.703033978
02/19/2019 19:32:27.711	3981116.276	64726.859	2953.580944	1060.251279	0.041029814	0.747058256
02/19/2019 19:33:27.715	3964752.608	63923.04068	2941.976841	1044.906257	0.058034104	0.656325714
02/19/2019 19:34:27.719	4018463.445	66674.11352	2984.048179	1068.834923	0.052372438	0.88207665
02/19/2019 19:35:27.708	4015213.942	64886.10501	2979.362963	1068.12101	0.031538576	0.64812608
02/19/2019 19:36:27.712	3997072.119	64125.03868	2964.921269	1054.998264	0.056114455	0.751712512
02/19/2019 19:37:27.715	3993494.78	63947.5344	2962.915682	1044.285993	0.033546018	0.674812307
02/19/2019 19:38:27.722	3948041.704	64075.25228	2930.762051	1050.686078	0.08205177	0.659170032
02/19/2019 19:39:27.713	4014396.783	64341.181	2978.35569	1058.590479	0.042669168	0.664776863
02/19/2019 19:40:27.717	4150444.305	35284.28375	3078.712554	563.9051056	0.036544358	0.412378615
02/19/2019 19:41:27.720	4356038.153	27843.88557	3230.55477	448.717791	0.036687239	0.394592061
02/19/2019 19:42:27.723	4364644.333	27501.61949	3236.353234	444.146525	0.030499357	0.321668916

Рисунок 2.5 - Вихідний файл монітора продуктивності

Інструмент IPerf, який використовується для тестування пропускної здатності мережі та забезпечення вимірювань [4], IPerf може генерувати трафік TCP/UDP. IPerf використовує модель клієнт/сервер. Він сумісний з декількома платформами Windows/Linux. IPerf має вбудовані функції для налаштування процесу тестування та виведення результатів у файли CSV для подальшого аналізу. IPerf підтримує різні протоколи (TCP, UDP, SCTP з IPv4 або IPV6). IPerf повідомляє про два важливі показники продуктивності мережі (швидкість передачі, пропускна

здатність). Версія IPerf, яка використовується на тестовому стенді та в реальному середовищі, — це версія iperf-3.1.3. Команда IPerf може використовувати будь-яку з наведених нижче програм для запуску своїх команд (командний рядок або PowerShell).

У таблиці нижче пояснюються параметри, які можна використовувати для налаштування команди тестування:

Таблиця 2.2 - Параметр командного рядка IPerf

Параметр командного рядка	**Опис**
-p, --порт n	Порт сервера для прослуховування сервером і підключення клієнта. Це має бути однаковим як на клієнті, так і на сервері. За замовчуванням 5201.
-i, --інтервал n	Встановлює інтервал часу в секундах між періодичною пропускною здатністю, джиттером, та звіти про збитки.
-V, --багатослівний	Дайте більш детальний вихід.
--logfile	Надіслати вихід у файл журналу.
-s, --сервер	Запустіть IPerf у режимі сервера (це дозволить лише одне підключення IPerf за раз).
-c, --клієнтський хост	Запустіть IPerf у режимі клієнта, підключившись до сервера IPerf, який працює на хості.
-t, --time n	Час у секундах для передачі.

PowerShell — це платформа командного рядка на основі завдань і мова сценаріїв від Microsoft. Для будь-якого системного адміністратора це потужний інструмент, який можна використовувати для виконання різних видів операцій на будь-якій підтримуваній платформі Windows, macOS і Linux. Раніше він автоматизував процеси та завдання, планував виконання операцій у певний час. Остання версія PowerShell є незалежною від Windows, безкоштовною та відкритою [5]. У цьому дослідженні використовується версія PowerShell 5.1

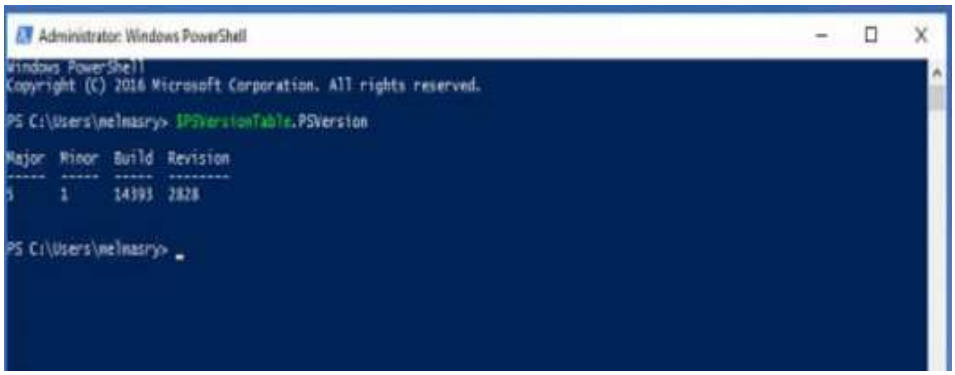

Рисунок 2.6 - Версія PowerShell

PowerShell використовується в цьому дослідженні для розробки сценаріїв, які автоматизують вилучення показників продуктивності мережі та програм із клієнтів і серверів під керуванням платформи Windows. Розроблений сценарій використовується для автоматизації запуску інструменту тестування мережі IPerf, збору основних процесів, які використовують процесор і команду ping для перевірки затримки в мережі між сервером і клієнтом.

На малюнку нижче показано сценарій Ping:

```
Test-Connection 10.5.1.229 -delay 60 -count 720 | Format-list
@{n='TimeStamp';e={Get- Date}}, SERVER,
Address,IPV4Address,ResponseTime | out-file
C:\Users\melmasry\Desktop\ThesisWork\Server-ping.txt -Append
```

Рисунок 2.7 - Сценарій Ping

На малюнку нижче показано сценарій Top-Processes:

```
function TopProcess { Param (
[Parameter(Position=1)] [Alias("l")] [int]$TotalList=5, [Parameter(Position=2)] [Alias("r")]
[int]$Invertal=60
)
Begin {} Process {
While ($true) {
$CounterSamples = Get-Counter '\Process(*)\ID Process','\Process(*)\% Processor Time','\Process(*)\Working
Set' | Select-Object -Expand CounterSamples
Clear-Host
$CounterSamples | Group-Object { Split-Path $_.Path } | Where-Object {$_.Group[1].InstanceName -notmatch
"^Idle|_Total|System$"} | Sort-Object -Property {$_.Group[1].CookedValue} -Descending | Select-Object -
First
$TotalList | Format-Table @{n='TimeStamp';e={Get- Date}},@{Name="ProcessId";Expression=
{$_.Group[0].CookedValue}},@{Name="ProcessorUsage";Expression
={[System.Math]::Round($_.Group[1].CookedValue/100/$env:NUMBER_OF_PROCESSORS,4)}},@{Name="Pr
ocessName";Expression={$_.Group[1].InstanceName}},@{Name="WorkingSet";Expression={[System.Math]::Ro
und($_.Group[2].CookedValue/1MB,4)}}
Sleep -Seconds $Invertal
}
}
End {}
}

TopProcess | Out-File C:\Users\melmasry\Desktop\ThesisWork\Server-Task.txt -Append
```

Рисунок 2.8 - Верхній процес

На малюнку нижче показано сценарій IPerf:

```
cd c:/
cd iperf-3.1.3-win64
./iperf3.exe -c 10.5.1.229 -i 60 -t 900 -p 5021 -u -V --logfile Server.csv
```

Рисунок 2.9 - Верхній процес

Планувальник завдань дозволяє користувачам Windows автоматизувати завдання та запускати сценарії в певний час, запускати програми або навіть надсилати електронні листи. Є кілька функцій, які можна використовувати та дозволяють регулювати більше параметрів керування в будь-якому запланованому виконанні завдання. Планувальник завдань за замовчуванням вбудований у кілька платформ Windows.

Рисунок 2.10 - Планувальник завдань

Основні компоненти планувальника завдань, які були використані в цьому дослідженні:

1) Завдання Дія;
2) Тригер завдання;
3) Повторення завдання;

Надбудова Ku-Tools, яка використовується з Excel Microsoft, дозволяє реалізувати розширені функції та операції у файлах Excel, CSV. Функція, яка використовується в цьому дослідженні, називається Transform Range.

Рисунок 2.11 - KU-Tools

2.2 Збір та аналіз даних

Експеримент проводиться, щоб визначити вплив програм на споживання ресурсів у комп'ютерних мережах шляхом вилучення показників продуктивності. Дані аналізуються за допомогою методів машинного навчання та аналізу даних, а також виявлення будь-якої кореляції між цими показниками. Показники продуктивності витягуються як з клієнта, так і з сервера. Клієнт отримує доступ до файлів на сервері або використовує встановлені мережеві програми.

Синхронізація між завданнями, що виконуються в цьому експерименті, є найважливішим фактором для отримання точних показників продуктивності, які перетворюються в базу даних. Показники продуктивності вимірюються та витягуються за допомогою встановленого незалежного програмного забезпечення, вбудованих програм в операційних системах Windows.

Використовується програмне забезпечення: Performance Monitor, Task Scheduler, PowerShell Scripts і інструмент тестування мережі IPerf. IPerf вводиться в експеримент для перевірки пропускної здатності мережі та швидкості передачі. Це спрямовано на збільшення обсягу даних, якими обмінюються клієнт і сервер.

Інструмент IPerf генерує протоколи трафіку UDP або TCP. У цьому експерименті інструмент IPerf генерує лише TCP-трафік. Windows PowerShell використовується для запуску команд IPerf. Сценарій PowerShell створюється та імпортується в планувальник завдань і налаштовується для запуску в певний час. Показники ефективності збираються кожні одну хвилину.

1) Команда, яка використовується для ініціювання трафіку від клієнта до сервера (запуск на клієнті);

```
iperf.exe -c 10.5.1.229 -i 60 -t 900 -p 5021 -V --logfile client-iperf.csv
```

Рисунок 2.12 – Запуск команди на рівні клієнта

2) Команда, яка використовується для ініціювання трафіку від сервера до клієнта (запуск на сервері);

```
iperf.exe -c 10.3.1.8 -i 60 -t 900 -p 5021 -V --logfile server-iperf.csv
```

Рисунок 2.13 – Запуск команди на рівні сервера

3) Команда, яка використовується для дозволу клієнту та серверу прослуховувати порт 5021 (запуск на сервері та клієнті);

```
iperf.exe -s -p 5021
```

Рисунок 2.14 – Запуск команди на рівні клієнта та сервера

Завдання збору даних повинні виконуватися точно в один і той же час на клієнті або серверах. Основна причина цього полягає в тому, щоб зробити знімок показників продуктивності як на клієнті, так і на сервері в одну секунду.

Монітор продуктивності має компонент під назвою розклад. Цей компонент використовується після створення набору збирача даних і додавання показників ефективності.

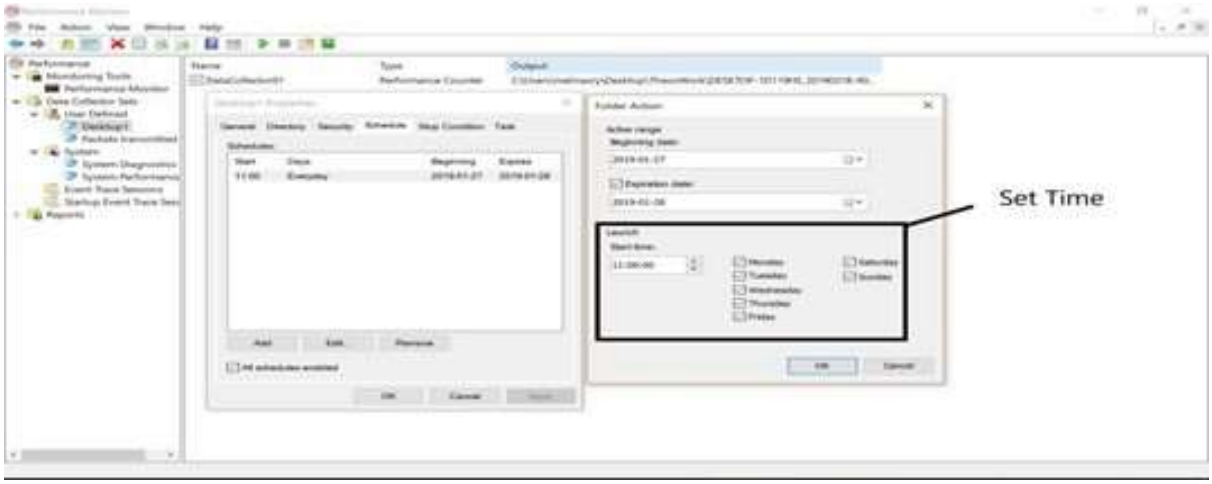

Рисунок 2.15 - Запланованe завдання монітора продуктивності

Набір збирача даних налаштовано для створення файлу CSV, який попередньо обробляється та пізніше аналізується. У таблиці нижче показано вихідний файл CSV із Performance Monitor.

(PDH-CSV 4.0) (Eastern Daylight Time)(240)	Bytes Received/sec	Bytes Sent/sec	Packets Received/sec	Packets Sent/sec	% Disk Time	% Processor Time
03/23/2019 15:01:00.683	924.566774	576.5685157	7.518429438	3.834232308	0.035911957	0.204350484
03/23/2019 15:02:00.687	902.5265613	634.8051246	7.016354882	3.91649263	0.029815403	0.2194204
03/23/2019 15:03:00.691	1275.742512	1449.353624	8.299734427	5.16650135	0.055096234	0.180583007
03/23/2019 15:04:00.696	1099.053516	675.9714102	7.483016844	4.366481989	0.086994674	0.366845533
03/23/2019 15:05:00.699	803.4012519	581.2434564	6.61640246	3.88317827	0.07667075	0.24581784
03/23/2019 15:06:00.703	1012.943248	669.7234966	7.333043138	4.249831819	0.052871442	0.204024843
03/23/2019 15:07:00.690	778.1249977	557.2919998	6.968233799	3.734173136	0.052771089	0.174651481
03/23/2019 15:08:00.694	1087.967107	558.7412144	8.099631039	3.666499647	0.038150961	0.235697007
03/23/2019 15:09:00.698	1075.416732	726.2829435	7.716308379	4.349798028	0.062595522	0.181311643
03/23/2019 15:10:00.702	1079.684063	716.0506569	7.183005542	4.116478814	0.062225282	0.178758007
03/23/2019 15:11:00.705	1057.883955	604.488451	7.282993385	3.999813301	0.106401979	0.264745522
03/23/2019 15:12:00.705	921.2916661	663.2366705	6.883022035	4.066482752	0.035093045	0.188333848
03/23/2019 15:13:00.690	1009.292889	646.6949496	7.751737467	4.184271192	0.021675269	0.098690841
03/23/2019 15:14:00.691	759.981645	643.2870225	6.649693567	3.699829503	0.045544661	0.141365877

Рисунок 2.16 - Вихідні дані монітора продуктивності

Планувальник завдань використовується для планування запуску сценаріїв PowerShell у точний час. Для виконання цієї операції використовується компонент Task Trigger.

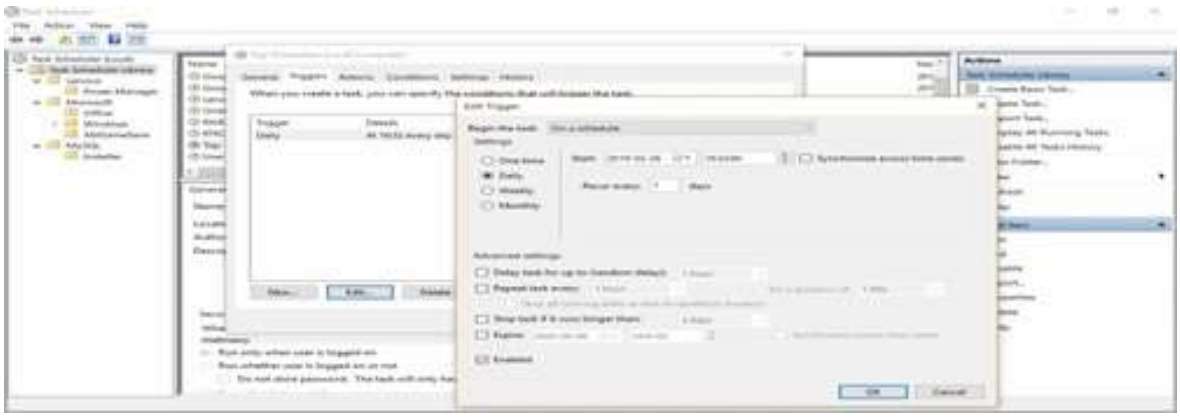

Рисунок 2.17 - Тригер завдання

Компонент планувальника завдань використовується для імпорту сценарію PowerShell для запуску.

Параметр аргументу використовується для пошуку сценарію PowerShell: «-Файл C:\Users\melmasry\Desktop\ThesisWork\iperf-test.ps1»

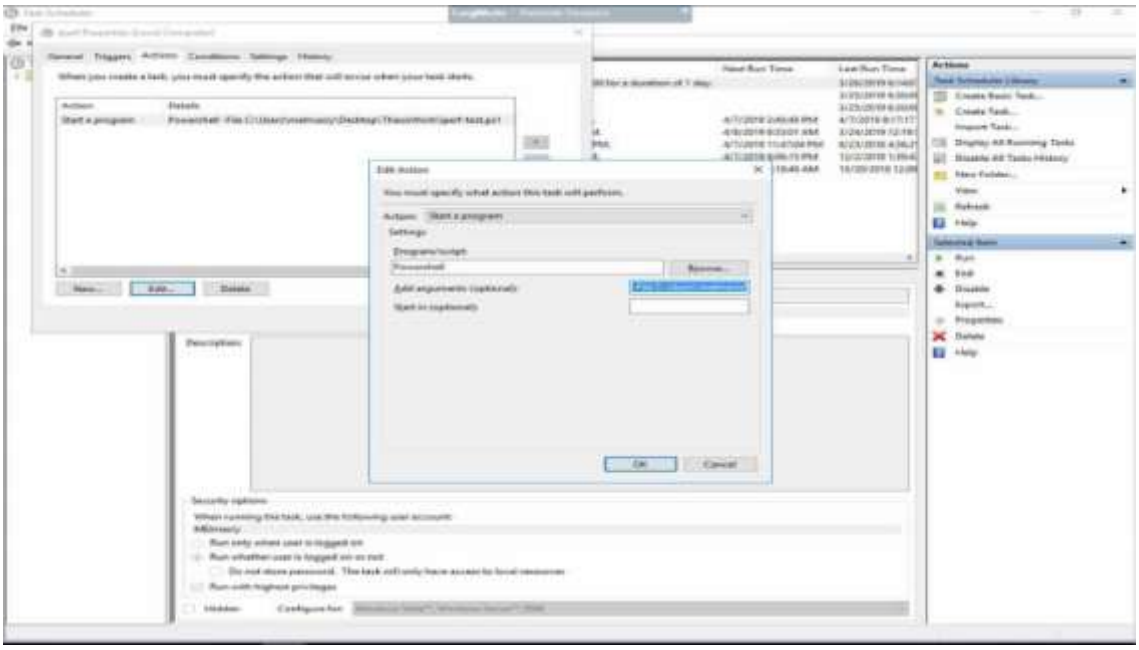

Рисунок 2.18 - Дія планувальника

Планувальник завдань використовується для запуску трьох різних сценаріїв PowerShell, запланованих для одночасного запуску як на клієнті, так і на сервері:

4) Завдання 1: Скрипт IPerf;
5) Завдання 2: скрипт команди Ping;
6) Завдання 3: сценарій Top Processes;

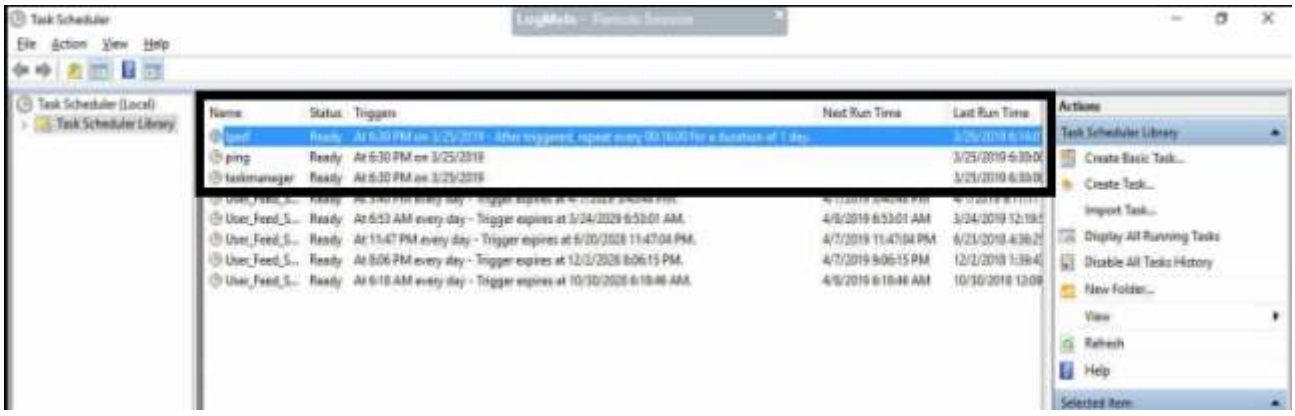

Рисунок 2.19 - Заплановане завдання

Кожен сценарій, який використовується в планувальнику завдань, надає вихідний файл. Вихідний файл є текстовим і потребує попередньої обробки перед перетворенням у базу даних. Сценарії PowerShell розроблено для надання вихідних даних із позначкою часу. Ця позначка часу використовується для вирівнювання результатів усіх сценаріїв PowerShell, монітора продуктивності та сценарію IPerf.

На малюнках нижче показано результат сценарію Ping Command PowerShell. Сценарій PowerShell, використаний у цьому експерименті, реалізував іншу команду, яка використовується як заміна, і надає той самий результат під назвою Test-Connection:

```
TimeStamp    : 7:15:01 PM
__SERVER     : PDGCUSE
Address      : 10.3.1.220
IPV4Address  : 10.3.1.220
ResponseTime : 32

TimeStamp    : 7:16:01 PM
__SERVER     : PDGCUSE
Address      : 10.3.1.220
IPV4Address  : 10.3.1.220
ResponseTime : 29
```

Рисунок 2.20 - Вихід сценарію Ping

На малюнку нижче показано результат сценарію Top Processes PowerShell:

```
TimeStamp            ProcessId ProcessorUsage ProcessName WorkingSet
---------            --------- -------------- ----------- ----------
3/23/2019 7:34:39 PM      9460         0.0044 iperf3          6.6914
3/23/2019 7:34:39 PM      5140         0.0005 powershell     71.1875
3/23/2019 7:34:39 PM      1136         0.0005 svchost        89.4258
3/23/2019 7:34:39 PM      3048         0.0005 dwm            88.1875
3/23/2019 7:34:39 PM      7508         0.0005 conhost        14.3008

TimeStamp            ProcessId ProcessorUsage ProcessName              WorkingSet
---------            --------- -------------- -----------              ----------
3/23/2019 7:35:40 PM      8028         0.0073 platform-patching-plugin    15.1523
3/23/2019 7:35:40 PM      9460         0.0063 iperf3                       6.6914
3/23/2019 7:35:40 PM      3048         0.0005 dwm                         88.1875
3/23/2019 7:35:40 PM      7508         0.0005 conhost                     14.3008
3/23/2019 7:35:40 PM      4284              0 wmiprvse                    46.0938
```

Рисунок 2.21 - Вихідні дані основних процесів

На малюнку нижче показано вихід сценарію IPerf PowerShell:

Рисунок 2.22 - Вихід сценарію IPerf

Вихідними результатами сценаріїв PowerShell є чисті текстові файли. Ці файли вимагають обробки для вилучення необхідних даних. Кожен текстовий файл імпортується в Microsoft Excel і за допомогою надбудови KuTools. Функція перетворення серед інших розширених операцій, які використовуються для отримання необхідних показників продуктивності.

База даних створюється з даних, отриманих із вихідних даних сценарію PowerShell. Стовпці бази даних складаються з двох сторін, одна сторона представляє показники продуктивності клієнта, а інша сторона сервера. Кожен рядок у базі даних представляє показники продуктивності, зібрані за одну хвилину. Після створення бази даних починається етап аналізу з використанням методів машинного навчання та аналізу даних.

Python — це потужна мова програмування, яку можна використовувати в багатьох різних сферах для вирішення реальних проблем. Python має багаті бібліотеки, які дозволяють програмістам виконувати багато дій без необхідності писати довгий код. У дипломній роботі мова програмування Python разом із кількома бібліотеками використовується для аналізу даних за допомогою машинного навчання та аналізу даних. Бібліотека Pandas надає інструменти для обробки та аналізу даних. Бібліотека Numpy надає математичні функції високого рівня, які можуть працювати з багатовимірними масивами. Бібліотека scikit-learn — це безкоштовна бібліотека машинного навчання, яка надає алгоритми класифікації, регресії та кластеризації. Він використовується в цій дослідницькій

роботі для регресії та дерев рішень. Бібліотека matplotlib — це потужна бібліотека побудови графіків для Python, яку можна об'єднати з іншими бібліотеками, такими як Pandas і Numpy.

Статистичний аналіз використовується для розуміння природи даних, що аналізуються. Це компонент аналітики даних і пов'язаний з інтелектуальним бізнесом [6]. Його застосовують до бази даних за допомогою потужних бібліотек і математичних функцій Python для виявлення тенденцій і пошуку шаблонів у структурованих і напівструктурованих даних. Базу даних аналізують, щоб знайти коефіцієнт кореляції між показником продуктивності програми та мережі. Коефіцієнт кореляції дає нам зрозуміти причинно-наслідковий зв'язок між показниками ефективності. Існує три методи коефіцієнта кореляції (Пірсона, Кендалла, Спірмена).

На малюнку нижче показано приклад коду Python, який використовується для обчислення коефіцієнта кореляції Спірмена:

```
import pandas as pd
import numpy as np
import math as m
import statistics
FileServer = pd.read_csv('FileServerMetrics.csv') Correlation_Coefficient=FileServer.corr(method='spearman')
```

Рисунок 2.23 - Код коефіцієнта кореляції Python

Статистичний аналіз продовжується для аналізу бази даних шляхом обчислення стандартного відхилення, середнього значення, кількості, процентиля, мінімального та максимального значень для показників ефективності.

На малюнку нижче показано код Python, а також результат, отриманий за допомогою методу Python під назвою describe().

```
import pandas as pd
import numpy as np
import math as m
import statistics
Client = pd.read_csv('Client Data.csv') Client.describe()
```

Рисунок 2.24 - Метод опису статистичного аналізу

Зібрану базу даних аналізують далі, щоб виявити тенденцію додатків, які споживають ресурси комп'ютерних мереж. Один стовпець метрики продуктивності «H1CPUP1» представляє найвищий процес, який використовував ЦП найдовше за хвилину. Кожне значення представляє назву процесу програми, встановленої на клієнті або сервері. Частота назви процесу в «H1CPUP1» досліджується, щоб визначити найпопулярніші програми, які використовують ЦП.

На малюнку нижче показано код Python, який використовується для пошуку найпоширеніших процесів:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline import seaborn as sns
Server = pd.read_csv('Server-2.csv') pd.set_option('display.float_format', lambda x: '%.4f' % x) plt
    .rcParams["figure.figsize"] = (20,20) plt.xlabel('occurance')
plt.ylabel('process names') Server.H2P1.value_counts().plot(kind='Barh')
```

Рисунок 2.25 - Найпопулярніші процеси Python

На графіку, створеному як вихід із наведеного вище коду Python, показано найчастіші процеси, які відбувалися як Top Process:

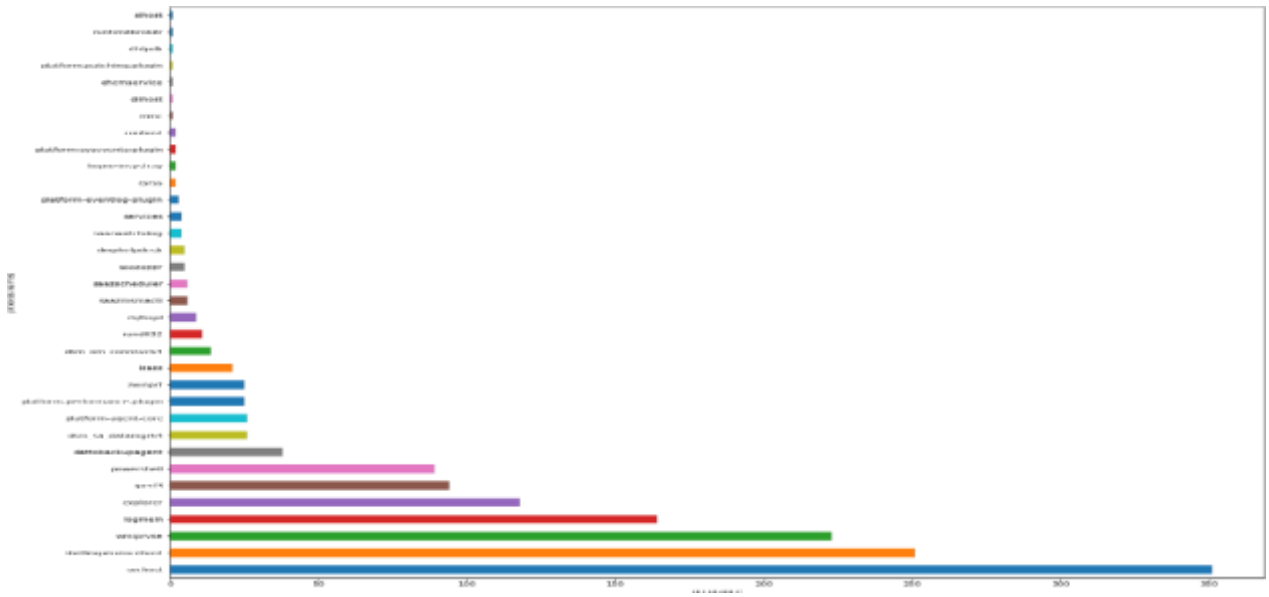

Рисунок 2.26 - Найчастіші процеси

Бібліотека seaborn надає набір інструментів для малювання графіків, наведений нижче графік представляє парний графік для показників продуктивності, щоб показати візуалізацію аналізу даних бази даних і ступінь розкиду даних.

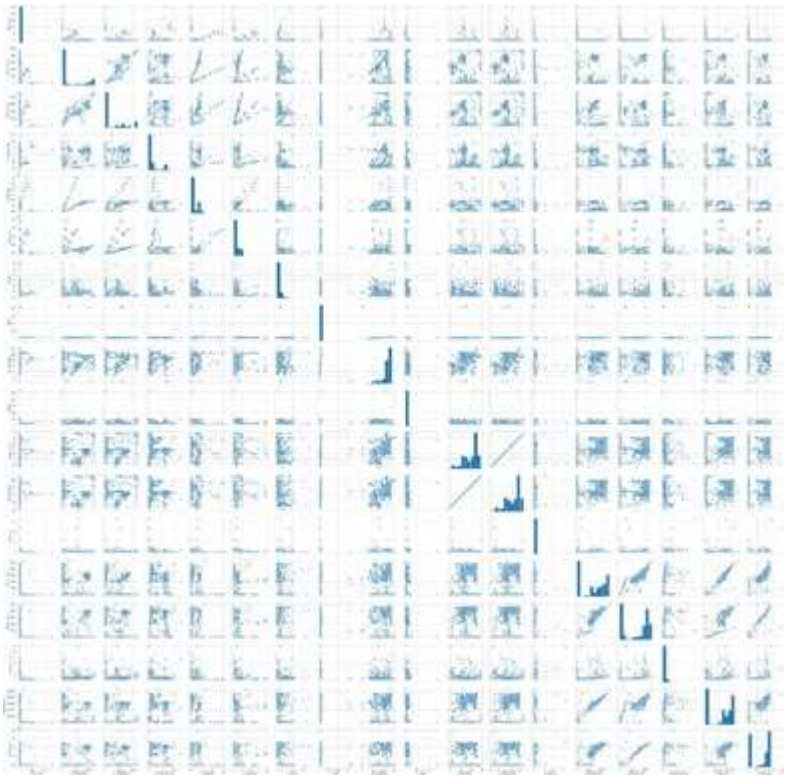

Рисунок 2.27 - Візуалізація аналізу даних

Алгоритм дерева рішень охоплює як регресію, так і класифікацію в машинному навчанні, це надає велику кількість інформації про базу даних показників ефективності. Дерево рішень забезпечує візуальне представлення рішень і прийняття рішень за допомогою моделей класифікації та дерева рішень. Він вважається інструментом інтелектуального аналізу даних, який дозволяє знаходити прихований шаблон, а також прогнозувати майбутні значення. Він полягає в постановці правильних запитань, щоб прийняти кращі рішення щодо того, які функції важливіші за інші. У цьому дослідженні дерева рішень ілюструють, які показники продуктивності мають більший вплив на споживання ресурсів і прогноз продуктивності в комп’ютерній мережі. Показники

продуктивності поділяються на групи предиктор і ціль. Базу даних також розділено на тестові та навчальні набори на 20-80 або 10-90, розділені для підвищення точності.

На малюнку нижче показана та візуалізована частина коду, розроблена для створення дерева рішень:

```
X = df1.drop(columns=['H1P1']) Y = df1['H1P1']
x_train, x_test, y_train, y_test = train_test_split(X,Y,test_size=0.1) model = DecisionTreeClassifier
    (max_depth=4) model.fit(x_train,y_train)
predications = model.predict(x_test)
score = accuracy_score(y_test, predications) import graphviz
feature_names = list(df1.drop(['H1P1'], axis=1))
dot_data = tree.export_graphviz(model, out_file=None, filled=True, rounded=True, feature_names=feature_names
    ,class_names=Y)
graph = graphviz.Source(dot_data) graph
```

Рисунок 2.28 - Код Python дерева рішень

Нижче на малюнку дерева рішень показано точність прогнозування та класифікацію для визначення майбутніх основних процесів, які використовують ресурси комп’ютерної мережі.

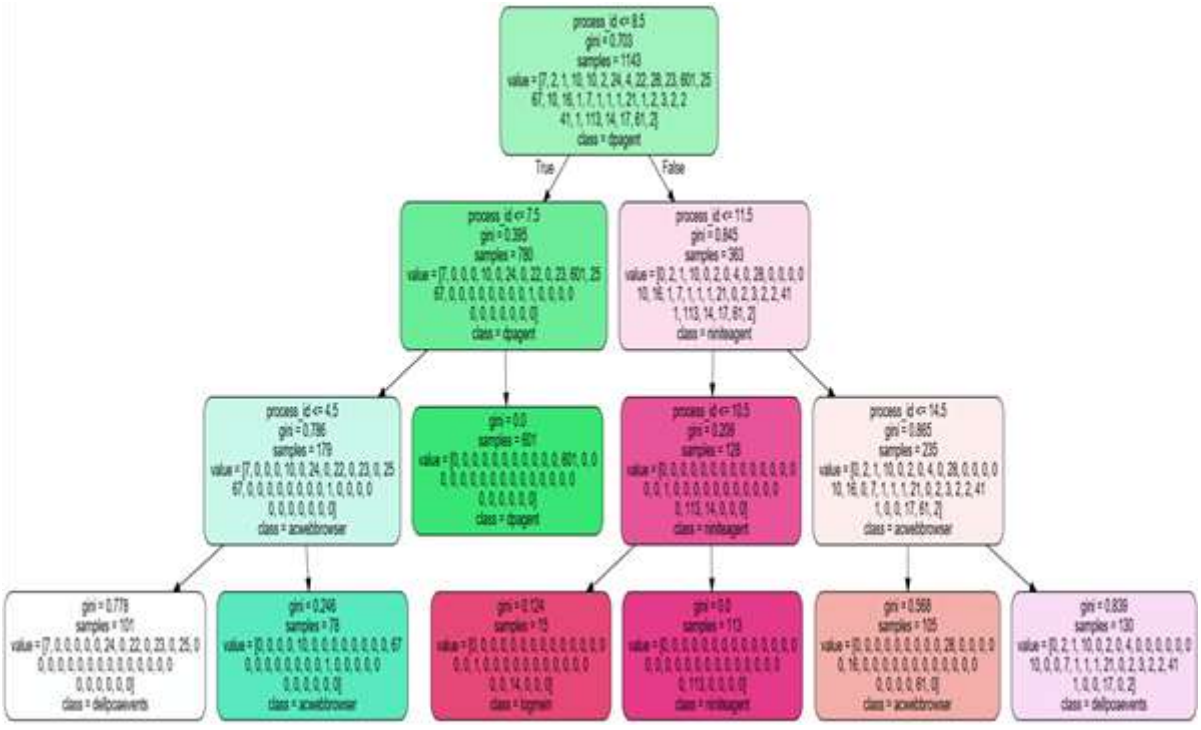

Рисунок 2.29 - Графік дерева рішень

Модель навчання та дерево рішень було створено, а потім навчальна база даних використовується для навчання моделі та тестової бази даних, які використовуються для перевірки прогнозування цільових топових процесів. Процес виконується шляхом перевірки прогнозованого результату з фактичними значеннями тестової бази даних.

2.3 Розробка моделей та алгоритмів

Розробка моделей та алгоритмів моніторингу продуктивності мережі є важливою складовою проактивної оптимізації мережі. Ці моделі і алгоритми допомагають збирати та аналізувати дані про мережу для виявлення потенційних проблем та вдосконалення її ефективності.

Одним зі способів розробки моделей моніторингу продуктивності мережі є використання статистичних методів. Наприклад, можна застосовувати методи аналізу часових рядів для прогнозування трендів та сезонності в продуктивності мережі. Це дозволяє виявити регулярні зміни в мережі та планувати запобіжні заходи заздалегідь. Крім того, можна використовувати методи регресійного аналізу для встановлення залежності між різними факторами та продуктивністю мережі. Наприклад, це може допомогти виявити, які фактори найбільше впливають на затримки або втрати пакетів в мережі.

Крім статистичних методів, для розробки моделей моніторингу продуктивності мережі можна використовувати інші підходи, такі як машинне навчання та штучний інтелект. Застосування цих методів дозволяє виявляти складні залежності та патерни у великих обсягах даних, які можуть бути складні для аналізу вручну. Наприклад, можна використовувати алгоритми класифікації для виявлення аномалій у поведінці мережі або алгоритми кластеризації для групування схожих типів проблем. Машинне навчання також може бути використане для прогнозування майбутньої продуктивності мережі на основі історичних даних.

Після розробки моделей та алгоритмів моніторингу продуктивності мережі, їх необхідно впровадити та інтегрувати з існуючими системами моніторингу. Це може включати розробку спеціального програмного забезпечення або інтеграцію з вже існуючими платформами моніторингу мережі. Після впровадження моделей та алгоритмів, вони можуть надавати регулярну інформацію про стан мережі та сповіщати про потенційні проблеми або прогнозувати майбутню продуктивність.

РОЗДІЛ 3 ПРОВЕДЕННЯ ДОСЛІДЖЕННЯ

3.1 Огляд існуючих методів та інструментів моніторингу мережі

Існує багато методів та інструментів моніторингу мережі, які допомагають збирати, аналізувати та візуалізувати дані про продуктивність мережі. Ось кілька з них:

1) SNMP (Simple Network Management Protocol): SNMP є одним з найпоширеніших протоколів для моніторингу мережі. Він дозволяє збирати дані про стан мережевих пристроїв, таких як маршрутизатори, комутатори та сервери. За допомогою SNMP можна отримати інформацію про пропускну здатність, завантаження, використання ресурсів та інші показники продуктивності;
2) Пакетний аналізатор: Пакетні аналізатори дозволяють перехоплювати та аналізувати мережеві пакети, що проходять через мережу. Вони можуть надати детальну інформацію про протоколи, затримки, втрати пакетів та інші параметри. Наприклад, популярні пакетні аналізатори включають Wireshark, tcpdump та tshark;
3) Flow-аналізатори: Flow-аналізатори збирають дані про мережевий трафік на основі потоків даних, замість аналізу окремих пакетів. Вони використовують протоколи, такі як NetFlow або sFlow, для збору інформації про пропускну здатність, навантаження, адреси вихідного та призначеного вузла тощо. Flow-аналізатори дозволяють відстежувати трафік на рівні мережі та аналізувати його характеристики;
4) Моніторинг мережевих пристроїв: Деякі виробники мережевих пристроїв надають власні інструменти моніторингу, які дозволяють збирати дані про продуктивність та стан пристроїв. Наприклад, Cisco має свою платформу моніторингу Cisco Prime, яка дозволяє відстежувати роботу мережевих пристроїв, виявляти проблеми та надавати звіти;

5) Візуалізація даних: Інструменти візуалізації даних, такі як Grafana або Kibana, дозволяють створювати графіки, діаграми та інші візуальні зображення на основі зібраних даних про мережу. Це дозволяє операторам мережі швидко сприймати інформацію та виявляти аномалії чи проблеми;
6) Аналіз логів: Лог-файли мережевих пристроїв можуть містити корисну інформацію про стан мережі. Аналіз логів дозволяє виявляти проблеми, помилки та незвичайну активність у мережі. Інструменти, такі як ELK Stack (Elasticsearch, Logstash, Kibana), можуть використовуватись для збору, аналізу та візуалізації лог-даних;
7) Алармування та сповіщення: Багато інструментів моніторингу мережі підтримують налаштування правил та сповіщень. Це дозволяє налаштувати автоматичні сповіщення про виявлені проблеми або незвичайну активність, що допомагає операторам мережі реагувати швидко на потенційні проблеми.

Моніторинг мережевих компонентів і серверів може створювати великі обсяги даних у формі файлів журналів і звітів. Щоб ця інформація була корисною, нам потрібно проаналізувати її та дійти певних висновків щодо стану та продуктивності мережі та її компонентів. Перевірка та читання файлів журналу вручну — завдання, якому не позаздрить жоден мережевий чи системний адміністратор. Це повільне й виснажливе завдання, яке піддається неправильному аналізу та висновкам через людську помилку. На щастя, існує багато інструментів і утиліт, які можна використовувати для аналізу та інтерпретації даних моніторингу та надання інформації нам у більш зручній формі. Ці інструменти та утиліти постачаються разом із програмним забезпеченням серверної операційної системи або можуть бути надані сторонніми постачальниками;

З часом інструменти та методи моніторингу розвивалися, і сьогодні моніторинг мережі включає більш проактивне вимірювання продуктивності мережевих компонентів і серверів.

Інструменти вимірювання продуктивності можуть контролювати такі показники, як використання процесора та диска, завантаження сервера, використання пам’яті, використання комутатора, маршрутизатора та мережі. Це

може включати опитування кожного обладнання в мережі для визначення справності цих компонентів. Моніторинг мережі може навіть виміряти час відповіді транзакцій і програм, які є критично важливими для компанії, або використання її пропускної здатності. Вимірювання, які виходять за межі попередньо встановлених контрольних показників продуктивності, можуть викликати попередження для моніторингового персоналу або навіть активувати попередньо визначений дії для виправлення ситуації до того, як станеться збій. Такі вимірювання можна зберігати в базі даних для аналізу тенденцій і планування потужності.

Більшість інструментів моніторингу мережі відстежуватимуть мережевий трафік і показуватимуть вам графічне представлення потоку та детальну статистику навантаження на мережу. Ви повинні мати можливість проаналізувати тип пакетів на ньому. Дані мережевого трафіку можна зберігати, а потім шукати, сортувати та фільтрувати. Ви повинні мати можливість вибирати різні протоколи, хости надсилання та отримання тощо. Це може допомогти визначити програми та/або користувачів, які можуть спричиняти проблеми з продуктивністю під час завантаження музики чи фільмів.

Великі організації з тисячами користувачів у мережі майже напевно інвестуватимуть у спеціалізовані засоби моніторингу, якими можуть бути апаратні чи програмні продукти. У них навіть може бути команда технічних спеціалістів, які лише перевіряють мережу.

Існує багато інструментів і утиліт, створених різними виробниками, які забезпечують функції моніторингу та оптимізації для різних систем. Зазвичай це комерційні продукти, які надають інформацію про моніторинг і оптимізацію в більш корисній і функціональній формі, ніж рідні інструменти. У більшості випадків інструменти сторонніх розробників зазвичай є більш спеціалізованими або призначеними для надання конкретної інформації та функцій.

Наприклад, Oracle надає утиліти налаштування системи, які працюють на серверах для моніторингу та оптимізації конфігурацій відповідно до додатків баз даних. SysInternal виробляє програмне забезпечення, яке повідомляє про те, які

файли та записи реєстру зчитуються операційною системою або до них звертаються.

Інші сторонні інструменти та утиліти можуть бути незалежними програмами, які встановлюються в мережі для спеціального моніторингу активності мережі та пристрою. Такі продукти, як OpenView , Whats UP і CAs Unicenter , забезпечують моніторинг мережі та програм у великих мережах і організаціях.

Запуск утиліт моніторингу зазвичай вимагає таких ресурсів, як пам’ять і процесорний час. Як наслідок, запущений моніторинг пов’язаний із накладними витратами та сам по собі може вплинути на продуктивність сервера чи мережевого пристрою.

3.2 Результати прогнозної аналітики для передбачення проблем мережі

Експерименти почалися з налаштування тестового середовища та сценаріїв для збору надійних даних у реальному часі з робочої мережі. Середовище тестування проводилося на окремому сервері. Вбудовані програми в операційній системі Windows і розроблені сценарії PowerShell були протестовані, щоб перевірити їхні результати, чи вони відповідають вимогам для проведення експерименту та отримання точних значень показників продуктивності.

Першою місією був вибір значущих показників продуктивності, які мають значний вплив на продуктивність програми та мережі. Наступним кроком був пошук методу отримання значень цих показників ефективності. Ряд даних показників продуктивності було отримано за допомогою вбудованих програм, а інші – за допомогою розробки сценаріїв PowerShell. Основною метою також є автоматизація цього завдання.

Отримані результати не були такими, як очікувалося на початку. Наприклад, для визначення того, які програми споживають ресурси комп’ютерної мережі, потрібен час ЦП, який використовується однією програмою за одну хвилину. Завдання, що стоїть перед цим завданням, операційна система Windows за замовчуванням дає накопичений час з моменту початку процесу. Сценарій

PowerShell, розроблений для виконання цього завдання, обчислює час процесора, який використовує програма, за одну хвилину. Результат було отримано, як очікувалося та було необхідно для успіху експерименту.

Два малюнки нижче показують різницю в результатах під час спроби отримати процесорний час, використаний однією програмою за одну хвилину:

Process Name	CPU Time
chrome	5384.59375
sqlservr	2884.859375
System	2462.625
dwm	2337.96875
chrome	1780.75

Рисунок 3.1 - Неправильний розрахунок використання часу ЦП

TimeStamp	ProcessId	ProcessorUsage	ProcessName	WorkingSet
3:00:04 PM	2692	0.001	svchost	12.3906
3:00:04 PM	11056	0.001	wmiprvse	52.8242
3:00:04 PM	10396	0.0005	logmein	59.2188
3:00:04 PM	4816	0	runtimebroker	32.6406
3:00:04 PM	6056	0	explorer	127.0625

TimeStamp	ProcessId	ProcessorUsage	ProcessName	WorkingSet
3:01:05 PM	2724	0.0034	logmeinrc	78.2109
3:01:05 PM	6680	0.0019	iperf3	6.5508
3:01:05 PM	3156	0.0015	dwm	149.0156
3:01:05 PM	12876	0.0005	powershell	69.8672
3:01:05 PM	4696	0.0005	svchost	7.2109

TimeStamp	ProcessId	ProcessorUsage	ProcessName	WorkingSet
3:02:06 PM	2724	0.0083	logmeinrc	78.2578
3:02:06 PM	6680	0.0078	iperf3	6.5508
3:02:06 PM	10396	0.001	logmein	59.2188
3:02:06 PM	12876	0.0005	powershell	72.3633
3:02:06 PM	4816	0	runtimebroker	32.625

Рисунок 3.2 - Розрахунок поправки для використання часу ЦП

Затримка є важливою метрикою для характеристики продуктивності мережі, вона дає вказівку на перевантаженість мережі через трафік, який використовується програмою, яка спілкується між клієнтом і сервером. Зазвичай у більшості операційних систем використовується команда ping. Мета полягає в тому, щоб запланувати надсилання одного пакету ping ICMP щохвилини протягом певного періоду часу, а результат затримки має супроводжуватися міткою часу початку ping.

Перший підхід полягає у використанні команди ping, але вона виявилася неефективною в цьому експерименті через відсутність налаштувань виводу. Інший підхід полягав у використанні сценарію PowerShell із командою під назвою test-connection, ця команда була доречною та задовольняла потреби цього експерименту на основі виводу та повторення. Для запуску цього сценарію використовується планувальник завдань.

3.3 Результати автоматизованої діагностики проблем мережі

Етап збору даних призвів до створення бази даних продуктивності з використанням результатів кількох програм і сценаріїв. Аналіз цієї бази даних призвів до цікавого висновку. Аналіз включав використання методів аналізу даних, а також застосування алгоритмів машинного навчання.

База даних включала п'ять найпопулярніших процесів, які використовували процесорний час за одну хвилину. Перший стовпець представляє номер один у списку найкращих процесів. Стовпець було проаналізовано, щоб визначити, які програми частіше з’являються в цьому стовпці, ніж інші, а також порівняти із загальним часом, який кожен процес використовував CPU Time. Проведемо використання бібліотеки matplotlib і опишемо метод аналізу верхнього стовпця процесу.

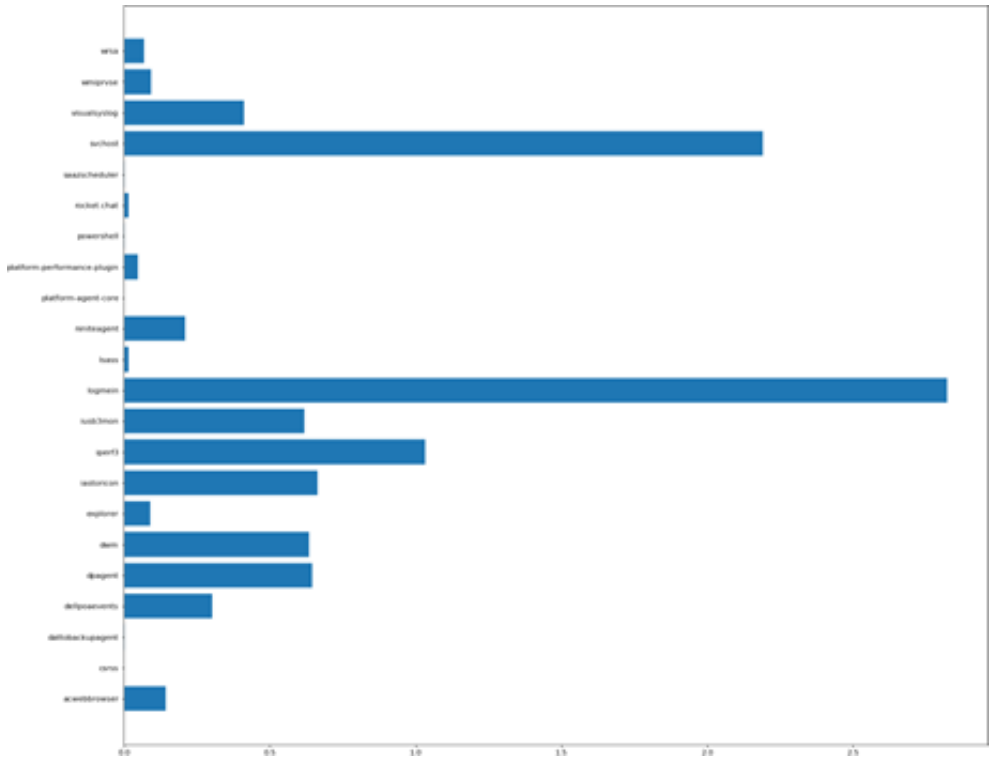

Рисунок 3.3 - Загальний час ЦП на процес

Кілька програм відображалися частіше, ніж інші, але інші програми, які відображалися менше в стовпці «Найвищий процес», споживали більше процесорного часу. Результат є логічним і доводить, що справа не в тому, як часто програма використовує ЦП, а в тому, скільки часу потрібно кожній програмі для виконання певного завдання. Наприклад, посилаючись на графік на малюнку 23 вище, найчастішим найкращим процесом є «logmein», а другим за частотою провідним процесом є «IPerf3». Досліджуючи графік на малюнку 27, ми виявили, що другим найпопулярнішим процесом використовувався процес CPU Time is svhost, що суперечить тому, що другою найчастішою програмою як найпопулярніший процес був «IPerf3».

Фільтрування бази даних для включення лише значень, що належать до найбільш часто використовуваних програм, з’являлося як головний процес і аналізувало цю частину бази даних. Використання статистичних методів для обчислення мінімального, максимального, середнього, стандартного відхилення та квартиля дає змогу зрозуміти розподіл даних у кожному стовпці. Для надання цього результату використовується метод describe().

Таблиця 3.1 - Опис результату методу

	означає	станд	хв	25%	50%	75%	макс	SD/середнє значення
H1DiskTime	0,6968	1,6632	0,0045	0,0323	0,0634	0,5982	15,4442	2,386911596
H1PacketsSent	1209,8453	1907.2108	1,2332	2,3332	36,9648	2296,6207	4817,5659	1,576408819
H1PacketsRecv	1163,8365	1834,906	1,8493	2,9665	49,6131	1822,2873	4619,0421	1,576601181
H1Час процесора	1,1159	0,7487	0,1303	0,9023	1,0226	1,1804	8,9249	0,670938256
H1BytesSent	1294659,1	1829841	122,32	5039,789	138766,58	2880897,7	6042295,7	1,413376683
H1BytesRecv	1406656,8	2336037,8	167.4	282,9439	57981,627	1856004,9	1938444	1,660702015
H1CPUP1	0,0038	0,0021	0,0005	0,002	0,0034	0,0049	0,0107	0,552631579
H1CPUP2	0,0008	0,0008	0	0,0005	0,0005	0,001	0,0078	1
H1CPUP3	0,0004	0,0006	0	0	0,0005	0,0005	0,0068	1.5
H1CPUP4	0,0003	0,0005	0	0	0	0,0005	0,0058	1,666666667
H1CPUP5	0,0001	0,0004	0	0	0	0	0,0058	4
H1Затримка	34,0239	10,7934	28	30	31	33	166	0,317229947
H1BW	33,3257	6,4725	8.9	33	35.5	37.9	38.4	0,194219476

Коефіцієнт варіації (CV) розраховується шляхом ділення стандартного відхилення на середнє значення. Якщо значення вище за 1, це означає високе стандартне відхилення, а точки даних, як правило, не розташовані надто близько одна до одної. Це міра для розподілу в точках даних показників ефективності. Показники ефективності з коефіцієнтом варіації більше 1 стосуються суттєвих змін протягом періоду збору даних. Це дозволяє зрозуміти природу показників продуктивності та в майбутньому покращити продуктивність мережі та додатків.

Кожен процес споживає певну кількість ресурсів від процесорного часу до затримки роботи мережі через обмін даними між клієнтом і сервером. Наведений нижче код розроблено для поєднання споживання ресурсів кожним процесом.

```
Client.groupby('H1P1')['H1CPUP1','H1Delay','H1BW'].sum()
```

Рисунок 3.4 – Код поєднання ресурсів

У таблиці 3.2 показано вихідні дані для кожного процесу разом із загальним процесорним часом, пропускною здатністю та затримками мережі.

Таблиця 3.2 - Підсумок споживання ресурсів

H1P1	H1CPUP1	H1 Затримка	H1BW
acwebbrowser	0,144	798	534
csrss	0,0005	39	12.9
dattobackupagent	0,0029	68	10.5
dellpoaevents	0,3036	1156	746,24
dpagegent	0,6462	2739	1789,33
dwm	0,6356	3279	5142,13
провідник	0,09	633	395,6
історікон	0,6639	2454	1567,4
iperf3	1,0323	7867	6377,7
iusb3mon	0,6186	2444	1657,17
logmein	2,8228	11482	8493,31
lsass	0,0159	209	156.9
niniteagent	0,2102	90	61.4
платформа-агент ядро	0,001	60	58.3
платформа- продуктивність-підключати	0,0468	86	62.2
Powershell	0,002	243	90.9
rocket.chat	0,0156	86	60.8
saazscheduler	0,0034	78	15.1
svchost	2.1914	1840 рік	1048,26
visualsyslog	0,4127	1438	978
wmiprvse	0,093	455	242.3
wrsa	0,0699	305	213.8

Коефіцієнт кореляції дає зрозуміти, чи існує взаємний зв’язок чи зв’язок між величинами. У цій дослідницькій роботі він використовується для виявлення причинно-наслідкового зв’язку між показниками продуктивності та вимірювання сили. Зв’язок може бути як позитивною, так і негативною кореляцією та коливатися від +1 до -1. Він використовувався для прогнозування однієї вартості на основі іншої поточної вартості.

Використання бібліотек і методів Python для розрахунку коефіцієнта кореляції між показниками ефективності. Існує кілька типів коефіцієнта кореляції, які можна розрахувати для показників ефективності. У таблиці 3.3 нижче показано коефіцієнт кореляції Пірсона між кількома показниками ефективності:

Таблиця 3.3 - Коефіцієнт кореляції Пірсона

	BytesRecv	BytesSent	PacketsRecv	PacketsSent	DiskTime	Час процесора
BytesRecv	1	0,490388	0,665255	0,550588	0,225197	0,290407
BytesSent	0,490388	1	0,926307	0,941955	-0,096438	0,073245
PacketsRecv	0,665255	0,926307	1	0,986419	-0,056017	0,116562
PacketsSent	0,550588	0,941955	0,986419	1	-0,097931	0,077764
DiskTime	0,225197	-0,096438	-0,056017	-0,097931	1	0,637152
Час процесора	0,290407	0,073245	0,116562	0,077764	0,637152	1

Для розрахунку кореляції використовувався коефіцієнт кореляції Пірсона, але результати показали, що це не найкращий тип коефіцієнта. Оскільки він обчислює лінійний зв’язок між безперервними змінними. Другим використовуваним коефіцієнтом кореляції є коефіцієнт кореляції Спірмена, він більше підходить для безперервних і дискретних даних. У таблиці нижче показано коефіцієнт кореляції Спірмена для показників ефективності:

Таблиця 3.4 - Коефіцієнт кореляції Спірмена

	H1DiskTime	H1PacketsSent	H1PacketsRecv	H1Час процесора	H1BytesSent	H1BytesRecv	H1CPUP1	H1Затримка	H1BW	H1TransferRate
H1DiskTime	1	-0,0945	-0,1498	0,2201	-0,0925	-0,1264	-0,0613	0,1219	-0,1159	-0,1152
H1PacketsSent	-0,0945	1	0,9039	0,5941	0,991	0,9061	0,4324	-0,4963	0,4989	0,4971
H1PacketsRecv	-0,1498	0,9039	1	0,6045	0,8852	0,8792	0,453	-0,4905	0,4738	0,4689
H1Час процесора	0,2201	0,5941	0,6045	1	0,5868	0,5373	0,5459	-0,282	0,331	0,3333
H1BytesSent	-0,0925	0,991	0,8852	0,5868	1	0,8705	0,4355	-0,4772	0,5023	0,501
H1BytesRecv	-0,1264	0,9061	0,8792	0,5373	0,8705	1	0,4028	-0,5273	0,4975	0,4974
H1CPUP1	-0,0613	0,4324	0,453	0,5459	0,4355	0,4028	1	-0,3558	0,4084	0,4039
H1Затримка	0,1219	-0,4963	-0,4905	-0,282	-0,4772	-0,5273	-0,3558	1	-0,4179	-0,4137
H1BW	-0,1159	0,4989	0,4738	0,331	0,5023	0,4975	0,4084	-0,4179	1	0,9978
H1TransferRate	-0,1152	0,4971	0,4689	0,3333	0,501	0,4974	0,4039	-0,4137	0,9978	1

Коефіцієнт кореляції Спірмена демонструє сильніший зв’язок і взаємозв’язок між показниками ефективності, оскільки він розглядає монотонне співвідношення, тобто змінні змінні разом, але не з постійною швидкістю. Третім коефіцієнтом кореляції, який використовується в цьому дослідженні, є коефіцієнт кореляції Кендалла. Він більше підходить для дискретної змінної.

Таблиця 3.5 - Коефіцієнт кореляції Кендалла

	H1DiskTime	H1PacketsSent	H1PacketsRecv	H1Час процесора	H1BytesSent	H1BytesRecv	H1CPUP1	H1Затримка	H1BW	H1TransferRate
H1DiskTime	1,0000	-0,2402	-0,2076	0,4187	-0,1915	-0,2041	0,3613	0,2623	-0,1292	-0,1179
H1PacketsSent	-0,2402	1,0000	0,6667	-0,2406	0,8688	0,5422	-0,1742	-0,3472	0,2858	0,2931
H1PacketsRecv	-0,2076	0,6667	1,0000	-0,2278	0,5411	0,7769	-0,1735	-0,4033	0,2833	0,2870
H1Час процесора	0,4187	-0,2406	-0,2278	1,0000	-0,1840	-0,1829	0,5124	0,2377	-0,0608	-0,0763
H1BytesSent	-0,1915	0,8688	0,5411	-0,1840	1,0000	0,4142	-0,1284	-0,2793	0,2759	0,2828
H1BytesRecv	-0,2041	0,5422	0,7769	-0,1829	0,4142	1,0000	-0,1685	-0,3929	0,3220	0,3128
H1CPUP1	0,3613	-0,1742	-0,1735	0,5124	-0,1284	-0,1685	1,0000	0,2335	-0,0586	-0,0530
H1Затримка	0,2623	-0,3472	-0,4033	0,2377	-0,2793	-0,3929	0,2335	1,0000	-0,2706	-0,2816
H1BW	-0,1292	0,2858	0,2833	-0,0608	0,2759	0,3220	-0,0586	-0,2706	1,0000	0,9628
H1TransferRate	-0,1179	0,2931	0,2870	-0,0763	0,2828	0,3128	-0,0530	-0,2816	0,9628	

Коефіцієнт кореляції Кендалла показав меншу асоціацію, ніж коефіцієнт Спірмена. Це надає інформацію про характер показників ефективності, які використовуються в цьому дослідженні.

3.4 Результати проактивної оптимізації мережі

Проактивна оптимізація — це в основному блок-схема бінарного дерева, яка використовується для поділу даних на групи, надзвичайно корисна для класифікації та регресії. Розподіл здійснюється відповідно до певних умов або питань, спрямованих на класифікацію даних зі схожими атрибутами, щоб виділити приховану інформацію, яка існує в даних. У цьому дослідженні дерева рішень використовуються для відображення та класифікації показників ефективності для прогнозів і цілей. Алгоритм дерева рішень, який використовується, є деревом рішень класифікації та регресії. Значення індексу Джіні коливається від 0 (усі цільові значення належать одній мітці) до максимального значення 1 (усі цільові значення розподіляються рівномірно). За допомогою навігації по дереву рішень можна передбачити програми, які належним чином споживають ресурси комп'ютерних мереж. Було виміряно точність моделі навчання, яка забезпечила точність між 70-80 відсотками. Наведений нижче код показує код і вихідні дані для вимірювання точності моделі навчання.

```
X = df1.drop(columns=['H1P1'])
Y = df1['H1P1']
x_train, x_test, y_train, y_test = train_test_split(X,Y,test_size=0.1)
model = DecisionTreeClassifier(max_depth=3)
model.fit(x_train,y_train)
predications = model.predict(x_test)
score = accuracy_score(y_test, predications)
score
```

```
0.8188976377952756
```

Рисунок 3.5 - Вимірювання точності моделі навчання прийняття рішень

Набір даних розділений на тренування та тестування, найвища точність досягнута при розділенні набору даних на 90% навчання та 10% тестування. Результати можна покращити за допомогою більшого набору даних. Крім того, максимальна глибина, яка використовується в дереві рішень, становить три, щоб уникнути створення більш складного дерева.

Проактивна оптимізація приймає рішення про розділення даних на основі правил, на малюнку 3.5 показано дерево рішень, що містить такі показники продуктивності (H1BytesSent, H1BytesRecv, H1DiskTime, H1ProcessorTime, H1PacketsSent, H1PacketsRecv, H1P1).

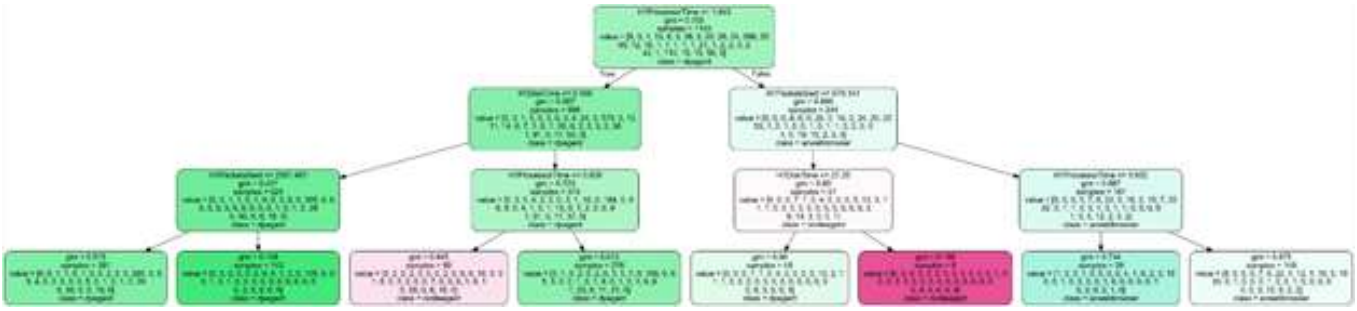

Рисунок 3.6 - Дерево рішень із сімома показниками ефективності

На малюнку нижче показано правила, створені для створення дерева рішень вище:

```
def tree(H1PacketsSent, H1PacketsRecv, H1BytesSent, H1BytesRecv, H1DiskTime, H1ProcessorTime, process_id):
  if process_id <= 8.5:
    if process_id <= 7.5:
      if process_id <= 4.5:
        return [[ 7.  0.  0.  0.  0.  0. 27.  0. 23.  0. 25.  0. 21.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]]
      else:  # if process_id > 4.5
        return [[ 0.  0.  0.  0.  8.  0.  0.  0.  0.  0.  0.  0.  0.  0. 68.  0.  0.  0.  0.
  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]]
    else:  # if process_id > 7.5
      return [[  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0. 601.   0.   0.
   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
   0.   0.   0.   0.   0.   0.]]
  else:  # if process_id > 8.5
    if process_id <= 11.5:
      if process_id <= 10.5:
        return [[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. 15.  0.  0.  0.]]
      else:  # if process_id > 10.5
        return [[  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
   0. 110.   0.   0.   0.   0.]]
    else:  # if process_id > 11.5
      if process_id <= 27.5:
        return [[ 0.  3.  1. 13.  0.  0.  0.  5.  0. 27.  0.  0.  0.  0. 11. 11.  0.  6.
  0.  1.  0. 21.  0.  0.  3.  3.  0.  0.  1.  0.  0. 19. 59.  2.]]
      else:  # if process_id > 27.5
        return [[ 0.  0.  0.  0.  0.  0.  2.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  1.  0.  1.  0.  0.  2.  0.  0.  1. 43.  0.  0.  0.  0.  0.  0.]]
```

Рисунок 3.7 - Правила дерева рішень

Раніше було проведено вимірювання моделі навчання дерева рішень, і очікувалося, що вона забезпечить точність 70-80%. Тестовий набір даних, який використовується як вхідні дані для моделі навчання після використання набору даних поїзда. Розроблено код для порівняння прогнозованого виходу з фактичним виходом тестових даних і скидання їх у файл CSV.

Вихідні дані можна побачити нижче, використовуючи простий спобіс обчислення для визначення точності виведення у відсотках:

H1P1_Predicted	H1P1_Actual		
svchost	svchost	TRUE	1
wmiprvse	lsass	FALSE	0
iperf3	iperf3	TRUE	1
iperf3	iperf3	TRUE	1
iperf3	iperf3	TRUE	1
iperf3	iperf3	TRUE	1
iperf3	iperf3	TRUE	1
wmiprvse	dattobackupagent	FALSE	0
iperf3	iperf3	TRUE	1
svchost	svchost	TRUE	1
dpagent	iusb3mon	FALSE	0
wmiprvse	dattobackupagent	FALSE	0
iperf3	iperf3	TRUE	1
iperf3	iperf3	TRUE	1
iperf3	iperf3	TRUE	1
logmein	logmein	TRUE	1
iperf3	iperf3	TRUE	1
visualsyslog	visualsyslog	TRUE	1
iperf3	iperf3	TRUE	1
iperf3	iperf3	TRUE	1
iperf3	iperf3	TRUE	1

Sum	99
total	127
	0.779528

Рисунок 3.8 - Розрахунок точності дерева рішень

Розрахунок показує, що точність становить 77,95%, що входить в діапазон прогнозованого значення.

РОЗДІЛ 4. ПОРІВНЯННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

4.1 Порівняння різних інструментів та методів моніторингу мережі

Інструменти моніторингу мережі мають життєво важливе значення, оскільки вони дають змогу організаціям та IT-командам:

1) Проактивне виявлення проблем: інструменти моніторингу мережі постійно взаємодіють із вашими мережевими пристроями та виявляють аномалії або проблеми зі зниженням продуктивності. Вони допомагають виявити потенційні проблеми до їх загострення. Проактивний підхід до моніторингу мережі позбавляє організації від кількох годин простою та дорогого ремонту;
2) Оптимізація продуктивності мережі. Інструменти моніторингу мережі виділяють критичні показники продуктивності, які допомагають налаштувати конфігурацію мережі. Такі показники, як доступна пропускна здатність, використання та моделі трафіку, дозволяють мережевим адміністраторам приймати обґрунтовані рішення щодо планування пропускної здатності;
3) Захистіть мережу: порушення безпеки та загрози є більш поширеними. Організації повинні бути готові до виявлення та пом’якшення таких ризиків. Інструменти моніторингу мережі вирішують цю важливу функцію;
4) Зниження витрат: моніторинг мережі допомагає відділу мережевих операцій здійснювати оптимізацію, належне планування та скорочувати витрати на усунення несправностей. Команди служби підтримки можуть заощадити витрати на години підтримки, дозволяючи технічним спеціалістам збирати всі дані, необхідні для швидкого розгляду скарги або запиту клієнта;

Вибираючи правильний інструмент моніторингу мережі, подумайте, який інструмент підходить для моніторингу мережевої інфраструктури, а який слід використовувати для моніторингу цифрового досвіду кінцевих користувачів. Однак, перш ніж вирушити в джунглі інструментів моніторингу мережі, майте на увазі, що існує три основні категорії інструментів моніторингу мережі:

а) SNMP;

б) пасивний;

в) Активний;

Кожне з них має певну функцію з відповідними перевагами та обмеженнями.

На основі SNMP. Ці засоби моніторингу мережі використовують протокол SNMP для моніторингу мережевих пристроїв. Сервери, які часто називають опитувальниками SNMP, активно опитують стан пристроїв і відстежують споживання ресурсів, включаючи використання ЦП, використання пам’яті та дані, що передаються та приймаються через їхні інтерфейси. Інструменти SNMP генерують сповіщення, коли мережевий вузол стає недоступним або відчуває перевантаження ресурсів.

На наступному малюнку (Рис. 4.1) показано, як працює SNMP. У верхній частині запитувач SNMP, або сервер, періодично запитує агента SNMP (контрольований пристрій), щоб зібрати його статус і доступні ресурси. У цьому конкретному прикладі опитувальник запитує в агента SNMP ім’я хоста локальної машини. Агент SNMP повертає ім’я хоста у вигляді рядка (gonzo). Таким чином сервери на основі SNMP контролюють мережу. Унизу ми бачимо іншу функціональність SNMP. Цього разу агент агента SNMP надсилає пастку SNMP на приймач (сервер) SNMP, щоб попередити про збій. Цей тип транзакцій дозволяє мережевим пристроям повідомляти сервер SNMP про те, що щось не так, щоб він міг сповістити адміністратора мережі.

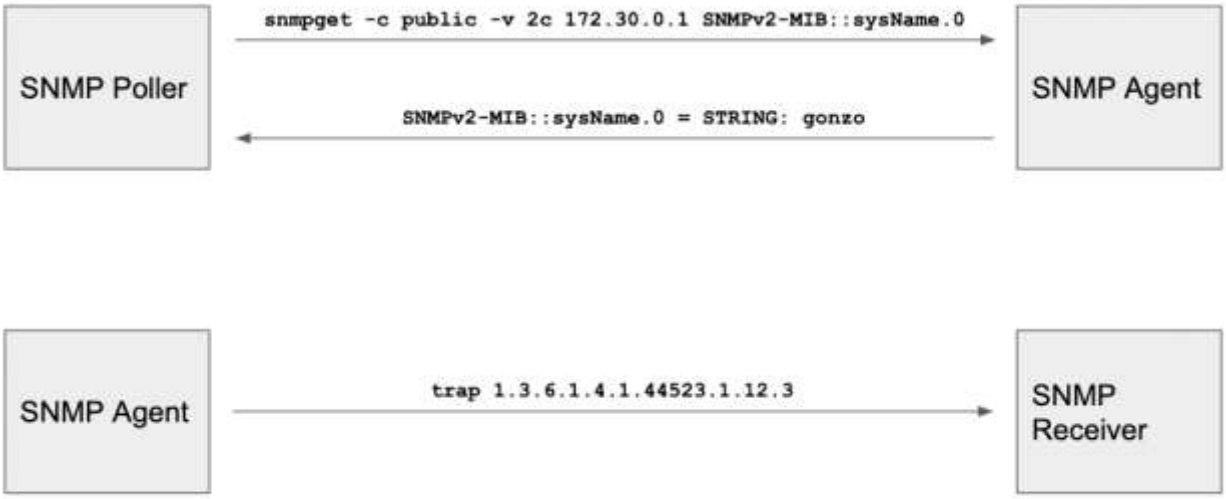

Рисунок 4.1 – Схема роботи SNMP

Найпоширеніші команди SNMP:

1) snmpget – запит на отримання інформації про об’єкт мережі. Як аргументи команди можна вказати один або кілька OID;
2) snmpbulkget – Подібно до команди snmpget, але більш ефективний спосіб запиту групи OID;
3) snmpwalk – запит на отримання піддерева значень керування від об’єкта мережі. OID може бути надано команді, яка визначає, яка частина простору ідентифікатора об’єкта буде шукатися за допомогою запитів GETNEXT;
4) snmpbulkwalk – Подібно до команди snmpwalk, але більш ефективний спосіб запиту групи піддерев;
5) snmpset – запит на встановлення інформації про мережевий об’єкт. У командному рядку потрібно вказати один або кілька OID як аргументи;
6) snmptrap – операція надсилання інформації менеджеру мережі. Як аргументи команди можна надати один або кілька OID;

Інструменти пасивного моніторингу мережі обробляють і збирають реальні дані користувача (також звані потоком трафіку). Ці інструменти генерують сукупну статистику трафіку, розбиваючи її за протоколами та хостами. Мережевий адміністратор може ідентифікувати машини (найбільш розмовні пристрої), які

споживають пропускну здатність певної мережі. Пасивні інструменти також можуть перевіряти певну послідовність пакетів, щоб точно визначити проблеми продуктивності між клієнтом і сервером.

Існує три способи пасивного моніторингу мережі: мережевий TAP, порт SPAN і протоколи на основі потоку. Коротко представимо кожну з них:

1) Мережні TAP – це спеціальні пристрої, які встановлюються в лінію, де потрібно аналізувати трафік. Мережевий TAP відтворює весь вміст кадрів (рівень 2), що протікають. З цієї причини мережеві точки доступу є найкращим варіантом для глибокої перевірки пакетів (DPI). Багато інструментів SIEM (Інформація про безпеку та керування подіями) використовують цю технологію для проактивного виявлення та пом'якшення потенційних порушень безпеки або атак;
2) Порти SPAN (Switch Port for ANalysis) налаштовуються на комутаторі шляхом надсилання копії трафіку з певного порту або VLAN до іншого порту самого комутатора або іншого комутатора, підключеного до мережевого колектора. Порти SPAN можуть копіювати весь пакет, включаючи його корисне навантаження;
3) Аналіз потоку за допомогою протоколу, реалізованого постачальником мережевого обладнання, наприклад NetFlow і s-Flow, або відкритого стандарту, такого як IPFIX (експорт інформації про потік IP). Цей тип пасивного аналізу фіксує не весь вміст пакета, а лише його заголовок, тому він переважно підходить для статистики протоколів і трафіку;

Інструменти активного моніторингу мережі вводять реальні пакети в мережу для вимірювання наскрізної доступності, часу проходження, втрати пакетів, пропускної здатності, використання каналу та інших властивостей мережі. Цей тип моніторингу мережі також перевіряє програми з точки зору користувача. Агент моніторингу виконує реальні транзакції щодо програми, а потім вимірює продуктивність, як-от виконання та час відповіді. Ця техніка дає змогу перевірити кінцевий результат мережі та додатків без необхідності відстежувати окремі компоненти та робити висновок про їх доступність і продуктивність. Зворотній

зв'язок і виявлення збоїв і проблем із погіршенням продуктивності набагато швидші та надійніші.

У таблиці 4.1 наведено список плюсів і мінусів, які слід враховувати для моніторингу мережі:

Таблиця 4.1 – Порівняння засобів моніторингу мережі

ТИП	ПЛЮСИ	МІНУСИ
SNMP	Виявляти апаратні збої та перевантаження системних ресурсів	Відсутність наскрізної видимості
	Надання байтів вхідних/вихідних мережевих інтерфейсів	Не підходить для вирішення проблем продуктивності мережі
	Завжди на моніторингу (24×7)	Практично обійти складність мережі та віртуалізацію
	Досить простий у налаштуванні	Не вдається виявити помилки конфігурації програмного забезпечення (політики маршрутизації, ACL, …), які впливають на трафік користувачів
	Виявляти апаратні збої та перевантаження системних ресурсів	Відсутність наскрізної видимості
	Надання байтів вхідних/вихідних мережевих інтерфейсів	Не підходить для вирішення проблем продуктивності мережі
		Практично обійти складність мережі та віртуалізацію
		Не вдається виявити помилки конфігурації програмного забезпечення (політики маршрутизації, ACL, …), які впливають на трафік користувачів
		Відсутність наскрізної видимості
		Не підходить для вирішення проблем продуктивності мережі
		Практично обійти складність мережі та віртуалізацію
Пасивний	Облік і статистика транспортних потоків	Велике споживання дискового простору
	Порушення протоколів між мережевими з’єднаннями. Визначте найкращих розмовників	Обмежені історичні дані Вбудовані пристрої (відводи) створюють ще одну точку відмовиТ
	Аналіз глибокої перевірки пакетів	Програми, як правило, дорогі, тому їх не можна встановити всюди
		Дзеркальні порти споживають системні ресурси та не можуть охопити всі потоки, що проходять через вузол
		Вимагають досвіду та навчання Реактивне усунення несправностей
Активний	Виявляти зниження продуктивності та тенденції	Виконуючи реальні транзакції, ці інструменти споживають ресурси мережі та/або програми
	Завжди на моніторингу (24/7)	Для успішної реалізації в мережі має бути розгорнуто кілька апаратних або програмних агентів
	Може зберігати велику кількість історичних даних	
	Для створення KPI не потрібен реальний трафік користувачів	
	Тестуйте мережеву інфраструктуру на етапі перед розгортанням	
	Перевірте зміни конфігурації	

Перш за все, функціональність є важливим аспектом при виборі інструменту моніторингу мережі. Деякі інструменти, такі як SNMP (Simple Network Management

Protocol), дозволяють збирати дані про стан мережевих пристроїв, такі як пропускна здатність, завантаження та використання ресурсів. Вони також можуть підтримувати сповіщення про проблеми або незвичайну активність. З іншого боку, пакетні аналізатори, такі як Wireshark, дозволяють детально аналізувати мережеві пакети, виявляти проблеми з протоколами, затримки та втрати пакетів. Flow-аналізатори, які використовують протоколи NetFlow або sFlow, збирають дані про трафік на рівні мережі та надають інформацію про пропускну здатність, адреси вихідного та призначеного вузла та інші параметри.

Другий критерій, який можна використовувати для порівняння, - складність налаштування. Деякі інструменти, наприклад, SNMP, можуть бути легкими у встановленні та налаштуванні, оскільки вони використовують стандартний протокол, який підтримується багатьма мережевими пристроями. З іншого боку, деякі пакетні аналізатори або flow-аналізатори можуть вимагати більшої експертизи та налаштування для ефективного використання. Вони можуть потребувати встановлення агентів або моніторингових точок на різних вузлах мережі.

Масштабованість є ще одним важливим критерієм. Деякі інструменти моніторингу мережі можуть бути легко масштабовані для великих мереж з великою кількістю пристроїв і вузлів, таких як SNMP або flow-аналізатори, які можуть збирати дані від багатьох пристроїв одночасно. Інші інструменти, наприклад, пакетні аналізатори, можуть бути обмежені у своїй масштабованості через високі вимоги до обробки даних та ресурсів.

Доступність також грає важливу роль у виборі інструменту моніторингу мережі. Деякі інструменти можуть бути безкоштовними та відкритими для використання, наприклад, Wireshark або Zabbix. Вони надають можливість здійснювати базовий моніторинг та аналіз мережі без додаткових витрат. У той же час, існують комерційні інструменти, які можуть мати розширену функціональність та підтримку, але вимагають платних ліцензій або підписок.

Останній критерій - вартість. Вартість може включати як вартість самого інструменту, так і витрати на налаштування, підтримку та навчання персоналу.

Безкоштовні інструменти, які надають базовий моніторинг, можуть бути більш доступними для менших підприємств з обмеженим бюджетом. Однак, для більших організацій, які потребують розширеної функціональності та підтримки, комерційні рішення можуть бути більш вигідними, забезпечуючи високу якість та надійність.

У підсумку, порівняння різних інструментів та методів моніторингу мережі залежить від потреб та вимог вашої організації. Важливо враховувати функціональність, складність налаштування, масштабованість, доступність та вартість, щоб знайти оптимальний інструмент моніторингу мережі, який задовольнятиме ваші потреби і відповідатиме обмеженням вашого бюджету.

4.2 Аналіз ефективності прогнозної аналітики

Використання машинного навчання та аналізу даних у комп’ютерних мережах має величезний потенціал. У дипломній роботі було досліджено конкретний алгоритм машинного навчання та методи аналізу даних, щоб знайти переваги застосування цих методів і алгоритмів у реальній проблемі. У цьому випадку метою було передбачити продуктивність програми та мережі.

Робота почалася з навігації та аналізу бази даних, створеної на основі збору показників продуктивності. Показники продуктивності були отримані з реальної комп’ютерної мережі. Робота розпочалася з використання методів аналізу даних. Статистичний аналіз пролив світло на приховану інформацію, яку звичайна людина не бачить. Природу показників ефективності стає легше зрозуміти завдяки статистиці та цифрам.

Розрахунок коефіцієнта кореляції показав причинно-наслідковий зв’язок між показниками продуктивності, який проявляється у високій кореляції між процесорним і дисковим часом серед інших виявлених кореляцій і асоціацій. Як наслідок, інтерпретація поведінки програм неможлива навіть у складних комп’ютерних мережах. Складність мережі більше не є проблемою чи фактором, тому що важливо збирати точні показники та дані, а потім аналізувати їх.

Регресія та класифікація дерева рішень допомогли у прогнозуванні майбутніх значень споживання ресурсів у комп'ютерній мережі. Процесорний час, споживаний програмою, а також обсяг трафіку, який відправляється або отримується крайовими вузлами, можна передбачити з достатньою точністю.

Виміряна точність склала 70-80 відсотків для дерева рішень. Точність результатів залежить від того, наскільки велика база даних.

4.3 Оцінка точності автоматизованої діагностики

У цьому дослідженні мережа приватної компанії використовувалася для збору показників продуктивності її клієнтів і серверів, а також каналів зв'язку між філіями. Цікавими були результати аналізу даних. Нам вдалося знайти програми, які споживають більшість ресурсів у мережі компанії. Наприклад, два процеси під назвами logmein і svchost споживали більшу частину процесорного часу.

Нам вдалося розрахувати відповідний коефіцієнт кореляції, щоб виявити зв'язок між показниками ефективності. На продуктивність впливає обсяг даних, якими обмінюється клієнт/сервер. Була виявлена позитивна кореляція між кількістю пакетів, надісланих/отриманих за секунду, і часом процесора. Відповідно, час процесора збільшується у відповідь на збільшення кількості пакетів, якими обмінюються клієнти/сервер.

Розрахунок стандартного відхилення, середнього та квартиля пояснював вибірки точок даних, взятих із кожного показника ефективності. Процесорний час мав коефіцієнт варіації менше одиниці (0,67), що вказує на те, що різниця між даними процесорного часу мінімальна. З іншого боку, коефіцієнт варіації дискового часу був більшим за одиницю (2,386), що інформує нас про те, що зібрані точки даних мають великий запас.

Ми змогли створити дерево рішень, використовуючи показники, зібрані з мережі компанії, і метою було передбачити програму, яка споживає процесорний час більшу частину часу. Потім модель було навчено за допомогою набору даних,

використовуючи тестовий зразок, зібраний зі 127 зразків. Рівень успішності цього експерименту досяг 77,9%.

Висновки можуть допомогти будь-якому IT-фахівцю проаналізувати продуктивність програми та мережі та отримати інформацію з прийнятною точністю про те, як програма споживає ресурси, і передбачити майбутні значення показників продуктивності.

4.4 Оцінка результатів проактивної оптимізації

Оцінка результатів проактивної оптимізації мережі є важливим етапом у впровадженні та вдосконаленні технологій мережевого управління. Проактивна оптимізація передбачає використання прогнозних алгоритмів та аналіз даних для налаштування параметрів мережі заздалегідь, з метою досягнення поставлених цілей та покращення її продуктивності.

Один з основних показників успішності проактивної оптимізації - це досягнення поставлених цілей у контексті мережевих KPI. Це може включати зниження затримок передачі даних, підвищення пропускної здатності мережі, збільшення якості обслуговування для користувачів тощо. Вимірювання цих показників дозволяє зрозуміти, наскільки ефективно були налаштовані параметри мережі та чи вдалося досягти бажаних результатів.

Окрім того, важливо проводити порівняльний аналіз між результатами проактивної оптимізації та станом мережі до її застосування. Це дозволяє виявити поліпшення, які були досягнуті завдяки оптимізації, а також виявити потенційні проблеми або недоліки, які можуть бути виправлені у подальших ітераціях процесу оптимізації.

Оцінка результатів проактивної оптимізації також повинна враховувати економічну ефективність застосування таких технологій. Це означає аналізувати витрати на впровадження та підтримку оптимізованої мережі порівняно з отриманими перевагами. Якщо вартість впровадження і операцій виправдовується

покращенням продуктивності та якості обслуговування, то можна говорити про успішність проактивної оптимізації.

Перш за все, для оцінки результатів проактивної оптимізації необхідно визначити ключові показники ефективності (KPI), які відображають стан мережі та досягнення цілей. Це можуть бути такі показники, як швидкість передачі даних, пропускна здатність, рівень задоволеності користувачів тощо.

Після цього проводиться порівняльний аналіз між результатами проактивної оптимізації та початковим станом мережі. Цей аналіз дозволяє виявити поліпшення, які були досягнуті завдяки оптимізації, а також визначити можливі проблеми або недоліки, які потребують уваги.

Крім того, важливо враховувати економічну ефективність проактивної оптимізації. Це означає оцінювати витрати на впровадження та підтримку оптимізованої мережі порівняно з отриманими перевагами. Якщо вартість впровадження виправдовується покращенням продуктивності та якості обслуговування, то можна говорити про успіх проактивної оптимізації.

Для більш точної оцінки результатів проактивної оптимізації можна використовувати різні методи аналізу, такі як статистичні методи, моделювання випадкових процесів, або експертні оцінки. Також можна залучати користувачів та фахівців для отримання об'єктивного відгуку та оцінки впливу оптимізації на їхнє досвідчення та задоволеність.

Узагальнюючи, оцінка результатів проактивної оптимізації мережі є складним процесом, який включає в себе аналіз KPI, порівняння з початковим станом мережі та оцінку економічної ефективності. Цей аналіз допомагає виявити успіхи та недоліки виявити успіхи та недоліки проактивної оптимізації. Успіхи можуть бути пов'язані з поліпшенням продуктивності мережі, збільшенням пропускної здатності, зниженням затримок передачі даних або покращенням якості обслуговування для користувачів. Ці позитивні результати свідчать про те, що проактивна оптимізація була ефективною та вдалою.

Однак, під час оцінки результатів також необхідно виявити можливі недоліки або проблеми, які можуть виникнути внаслідок проактивної оптимізації.

Наприклад, певні налаштування можуть призвести до нестабільності мережі або збільшення кількості помилок передачі даних. Такі негативні наслідки потрібно виявити та вирішити, щоб покращити ефективність мережі та забезпечити задоволення користувачів.

Для оцінки результатів проактивної оптимізації можна використовувати різні методи та інструменти. Один з них - аналіз даних, зокрема збір інформації про мережеві KPI та порівняння їх зі станом до оптимізації. Це дозволяє вимірювати конкретні покращення в продуктивності мережі та встановлювати, наскільки ефективно були налаштовані параметри.

Крім того, можна провести опитування або спілкування з користувачами, щоб отримати їхній відгук та оцінку проактивної оптимізації. Це дозволить отримати спрямовану інформацію про їхнє задоволення від роботи мережі та виявити можливі проблеми, з якими вони стикаються.

Також важливо враховувати економічну ефективність проактивної оптимізації. Це означає оцінювати витрати на впровадження та підтримку оптимізованої мережі порівняно з отриманими перевагами. Якщо вартість впровадження виправдовується поліпшенням продуктивності та ефективністю мережі, то можна говорити про успішність проактивної оптимізації.

Узагальнюючи, оцінка результатів проактивної оптимізації мережі включає в себе виявлення успіхів та недоліків. Це оцінювання допомагає визначити, наскільки ефективно була виконана оптимізація та які кроки можна підійняти для подальшого вдосконалення мережі. Цей процес важливий для забезпечення стабільної та надійної роботи мережі, а також для задоволення потреб користувачів.

Оцінка успіхів проактивної оптимізації допомагає виявити сильні сторони та досягнення в процесі налаштування мережі. Наприклад, якщо після оптимізації було досягнуто значного покращення швидкості передачі даних або зниження затримок, це може свідчити про успішну роботу. Також, якщо збільшення пропускної здатності мережі призвело до збільшення обсягу передачі даних або покращення якості обслуговування для користувачів, це також вважається успіхом.

З іншого боку, оцінка недоліків допомагає виявити проблеми або слабкі місця, які потребують уваги та виправлення. Наприклад, якщо після оптимізації з'явилися проблеми зі стабільністю мережі або якість обслуговування погіршилася, це може бути недоліком. Також, якщо оптимізація призвела до підвищення вартості підтримки мережі без значного поліпшення продуктивності, це також може бути визнано недоліком.

Виявлення недоліків є важливим кроком для подальшого вдосконалення мережі. Після ідентифікації проблем можна вжити заходів для їх вирішення, таких як налаштування параметрів мережі, усунення помилок або впровадження додаткових заходів безпеки. Це допоможе вдосконалити продуктивність та надійність мережі, а також покращити задоволення користувачів.

Отже, оцінка успіхів та недоліків проактивної оптимізації мережі є важливим етапом у процесі вдосконалення мережевого управління. Виявлення успіхів допомагає визначити, що було досягнуто завдяки оптимізації, тоді як виявлення недоліків дозволяє виявити проблемні аспекти, які потребують уваги та виправлення. Комбінація успіхів та недоліків проактивної оптимізації мережі надає повну картину ефективності та потенційних викликів, з якими можна зіткнутися при впровадженні цих технологій. Важливо враховувати, що успіхи та недоліки є невід'ємною частиною процесу оптимізації, і вони можуть служити як джерело навчання та покращення.

Одним з ключових аспектів управління успіхами та недоліками є систематичний підхід до оцінки та зворотного зв'язку. Під час оцінки результатів оптимізації, важливо збирати дані, проводити аналіз та використовувати цю інформацію для прийняття рішень. Зворотний зв'язок з користувачами та зацікавленими сторонами також є критичним елементом для забезпечення повного розуміння їхніх потреб і вимог.

Крім того, під час оцінки успіхів та недоліків, важливо визначити пріоритети та встановити метрики успішності. Це допоможе визначити, які аспекти оптимізації є найважливішими для досягнення поставлених цілей. Наприклад, якщо одним із головних завдань є покращення якості обслуговування для

користувачів, то метрикою успішності може бути зниження середньої затримки передачі даних або підвищення рівня задоволеності користувачів.

При оцінці успіхів та недоліків проактивної оптимізації мережі також варто звернути увагу на довгострокові наслідки та потенційні ризики. Наприклад, впровадження нових технологій може вимагати додаткових інвестицій або збільшення вартості підтримки мережі в майбутньому. Такі ризики потрібно аналізувати та враховувати при прийнятті рішень щодо оптимізації.

Оцінка успіхів та недоліків проактивної оптимізації мережі також допомагає виявити обмеження та можливості для подальшого розвитку. Наприклад, виявлення недоліків може підказати про необхідність вдосконалення алгоритмів або впровадження нових технологій для подальшого покращення продуктивності. Успіхи, з свого боку, можуть вказувати на потенційні можливості для розширення оптимізації та використання її в інших сферах або ситуаціях.

Однак, на шляху до досягнення успіху з оптимізації мережі можуть виникати виклики і перешкоди. Наприклад, одним з найбільш поширених недоліків є можливість зниження надійності та стабільності мережі. Впровадження нових технологій може вимагати додаткового навчання персоналу та зміни існуючих процесів, що можуть бути складними та витратними.

Крім того, оптимізація мережі може стикатися з проблемами з приватністю та безпекою даних. Збільшення обсягу передачі даних та використання нових технологій може створювати нові потенційні точки вразливості, які можуть бути використані зловмисниками для несанкціонованого доступу до інформації або злому мережевої інфраструктури.

Більше того, оптимізація мережі може мати вплив на соціальний аспект, такий як зміна робочих місць або вплив на існуючі бізнес-моделі. Впровадження нових технологій може призвести до автоматизації або змін у робочих процесах, що може вплинути на зайнятість та вимагати перекваліфікації працівників.

Отже, комбінація успіхів та недоліків проактивної оптимізації мережі відкриває нові перспективи, але вимагає уважного підходу та систематичного аналізу. Оцінка результатів, залучення зворотного зв'язку та визначення

пріоритетів та метрик успішності є важливими кроками для підтримки ефективної оптимізації мережі. Крім того, необхідно враховувати потенційні ризики та виклики, такі як зниження надійності, проблеми з приватністю та безпекою даних, а також соціальні наслідки. З ретельним аналізом та здійсненням необхідних заходів можна досягти оптимального балансу між ефективністю мережі та забезпеченням безпеки, якості та стабільності.

Експеримент проводився на окремому сервері. Мета полягала в тому, щоб виділити будь-який збій, який може призвести до переривання бізнесу у випадку використання реальної комп’ютерної мережі. Це подовжило часові рамки створення робочого середовища. Кілька одночасних дій і синхронізація між завданнями є важливими факторами успіху експерименту. Експеримент вимагає наявності величезної бази даних, щоб отримати хороший результат.

Отже, фаза збору даних зайняла багато часу, щоб підготувати великий набір даних, щоб забезпечити надійні результати.

ВИСНОВКИ

Продуктивність програми та мережі є головною проблемою для мережевого фахівця. У роботі розглянуто декілька проблем, пов’язаних із визначенням додатків, які можуть зловживати ресурсами в комп’ютерних мережах. Запропоновані методи вирішення цієї проблеми базуються на використанні сучасних технологій машинного навчання та аналізу даних.

Обидві методики мають репутацію вирішення багатьох проблем у різних галузях. Результатом роботи стало розуміння показників продуктивності мережі та додатків. Крім того, природа метрик, якщо вони є категоричними, неперервними чи дискретними. Крім того, причинно-наслідковий зв'язок між показниками продуктивності шляхом розрахунку коефіцієнта кореляції. Крім того, дерева рішень допомогли передбачити продуктивність програми шляхом аналізу та класифікації показників продуктивності.

Показники продуктивності, які відповідають за вплив або споживання ресурсів комп’ютерної мережі, можна передбачити із задовільною точністю в межах 70-80%. Графіки дерева рішень змогли показати класифікацію показників продуктивності, а зміни в значеннях вплинуть на цільовий показник ефективності.

Застосування цієї роботи дозволить скоротити витрати та забезпечить значні переваги IT-фахівцям і середнім великим компаніям для оптимізації мережі та продуктивності додатків. У наш час мережі стають складнішими, а програми потребують набагато більше ресурсів. Традиційних підходів до роботи з мережею та продуктивністю програм недостатньо. Дипломну роботу можна розширити, щоб отримати більше переваг за допомогою використання інших алгоритмів, які забезпечують кращі результати.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Wang, M.; Cui, Y.; Wang, X.; Xiao, S.; Jiang, J. Machine learning for networking: Workflow, advances and opportunities. IEEE Netw. 2017, 32, 92–99;
2) Nishchol Mishra, Dr.Sanjay Silakari, Predictive Analytics: A Survey, Trends, Applications, Oppurtunities & Challenges, Nishchol Mishra et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 , 2012, 4434- 4438, ISSN : 0975-9646;
3) Social Network Analysis 101: Ultimate Guide [Електронний ресурс] // Visible Network Labs. – 2023. – Режим доступу до ресурсу: https://visiblenetworklabs.com/guides/social-network-analysis-101/;
4) Schroder C. Iperf [Електронний ресурс] / Christian Schroder // wikipedia. – 2021. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Iperf;
5) Maurer T. How to Install and Update PowerShell 6 [Електронний ресурс] / Thomas Maurer – Режим доступу до ресурсу: https://www.thomasmaurer.ch/2019/03/how-to-install-and-update-powershell-6/;
6) Rouse M. Statistical analysis [Електронний ресурс] / Mark Rouse – Режим доступу до ресурсу: https://whatis.techtarget.com/definition/statistical-analysis;
7) A comprehensive survey on machine learning for networking: evolution, applications and research opportunities / R.Boutaba, N. Limam, S. Ayoubi, S. Shahriar., 2018. – 157 с;
8) Data Mining. Practical Machine Learning Tools and Techniques / M.Hall, I. Witten, E. Frank, C. Pal. – Берлінгтон: Morgan Kaufmann, 2011. – 664 с;
9) Cisco Systems, Inc., "Enterprise Campus 3.0 Architecture: Overview and Framework," 2008. [Електронний ресурс] – Режим доступу до ресурсу: http://www.cisco.com/en/US/docs/solutions/Enterprise/Campus/campover.html;
10) J. Case, M. Fedor, M. Schoffstall and J. Davin, "RFC1157: A Simple Network Management Protocol (SNMP)," 5 1990. [Електронний ресурс] – Режим доступу до ресурсу: http://www.ietf.org/rfc/rfc1157.txt

11) Ed Wilson - Network Monitoring and Analysis: A Protocol Approach to Troubleshooting. Prentice Hall 2000 – 350;
12) Dr. Moustafa Elshafei - Modern Distributed Control Systems: A comprehensive coverage of DCS technologies and standards. CreateSpace Independent Publishing Platform; 1ie ed. 2016 – 478;
13) Барабаш О. В. Побудова функціонально стійких розподілених інформаційних систем : монографія. Київ : НАОУ, 2004. 226 с;
14) Chopra K. Future Internet: The Internet of Things-A Literature Review [Текст] / K. Chopra, K. Gupta, and A. Lambora // Proc. Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. Trends, Prespectives Prospect. – 2019;
15) The IoT in 2030: 24 billion connected things generating $1.5 trillion [Електронний ресурс] / iot business news. – 2020. - Режим доступу до ресурсу: https://iotbusinessnews.com/2020/05/20/03177-the-iot-in-2030-24-billion-connected-things-generating-1-5-trillion/;
16) Хайкін С. Нейронні мережі. Повний курс, 2019. 1104 с;
17) Shahraki A., Taherkordi A., Haugen Ø., Eliassen F. A survey and future directions on clustering: From WSNs to IoT and modern networking paradigms, 2020. С. 2242 – 2274;
18) Корнієнко О. О., Петров К. Е. Методи моніторингу та аналізу мережевого трафіку в веб-орієнтованих системах: матеріали Міжнар наук. конф., м. Харків, Баку, Жиліна, 2022 р. С. 18;
19) Mitchell T. Machine Learning Definition Science/Engineering/Math. National Academy of Engineering, 1997. С. 23–55;
20) Організація операційних процесів в галузі електрозв'язку. Мод. 1. Ч. 1, Організація операційних процесів в галузі радіозвязку, радіомовлення та телебачення. [Текст] : навч. посібник / С. Б. Горелкіна, Є. М. Стрельчук, Н. К. Заборська ; каф. менеджменту та маркетингу. – Одеса : ОНАЗ ім. О. С. Попова, 2007. – 95 с;