МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кваліфікаційна наукова
праця на правах рукопису

ПЕЛЕХ НАЗАР ВОЛОДИМИРОВИЧ

УДК 621.391

ДИСЕРТАЦІЯ

ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ФУНКЦІОНУВАННЯ ХМАРНИХ СИСТЕМ ДЛЯ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СЕРВІСНО-ОРІЄНТОВАНИХ МЕРЕЖ

172 – Телекомунікації та радіотехніка
(шифр і назва спеціальності)

17 «Електроніка та телекомунікації»
(галузь знань)

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело _____ / Пелех Назар Володимирович /

Науковий керівник Климаш Михайло Миколайович д.т.н., професор
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

ЛЬВІВ – 2022

АНОТАЦІЯ

Пелех Н.В. Підвищення ефективності функціонування хмарних систем для інформаційно-комунікаційних сервісно-орієнтованих мереж. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 172 – Телекомунікації та радіотехніка. – Національний університет «Львівська політехніка» Міністерства освіти і науки України, Львів, 2022.

Стрімкий розвиток і популяризація інформаційних сервісів вимагають нових підходів до систем, які забезпечують їх функціонування. Здавалося б сучасні технології, такі як хмарні обчислення, мають на меті задовольнити доступ до великої кількості обчислювальних потужностей, адже функціонують в умовах агрегації ресурсів та надання єдиного системного підходу по їх управлінню. Особливо актуальним є їх інтеграція у мережі провайдерів, коли хмарні системи стають складовими інформаційно-комунікаційних мереж. З одного боку – провайдер має можливість розширити набір послуг, які надає користувачам, з іншого – зростає потік трафіку, який передається через таку мережу. Відповідно, необхідні не лише нові підходи по удосконаленню, зберіганню та управлінню, а й до аналізу трафіку і, як наслідок обробки службових даних, оскільки її кількість збільшується в геометричній прогресії. Крім того, надмірне використання віртуальних серверів може ще більше ускладнити задачу, оскільки потребуватимуть залучення додаткових канальних ресурсів на рівнях комунікацій адміністратор додатків - сервер обробки – сервер зберігання даних. Крім того, управління такою інтегрованою мережевою інфраструктурою зводиться до проблеми підвищення захищеності даних від несанкціонованого доступу до них. Для цього необхідні певні системи моніторингу стану мережі, яка буде аналізувати можливі проблеми, що можуть виникати під час роботи, і запускати системи захисту в самому сервісі або повідомляти адміністраторів про можливу загрозу. Такі системи повинні в дуже

короткий час відслідковувати всі зміни, які відбуваються із системою - атаки, технічні збої, при яких сервіс перестає нормально функціонувати.

В дисертаційній роботі розв’язано науково-практичне завдання підвищення ефективності функціонування хмарних систем при їх інтеграції у сучасні інформаційно-комунікаційні сервісно-орієнтовані мережі шляхом розроблення методу кластеризації серверних вузлів, удосконалення моделей оцінки ефективності функціонування хмарних систем в умовах обслуговування групових потоків запитів та розвитку систем інтелектуального виявлення мережевих атак.

Метою представленої дисертаційної роботи є підвищення ефективності функціонування хмарних систем при їх інтеграції у сучасні інформаційно-комунікаційні сервісно-орієнтовані мережі із реалізацією інтелектуального виявлення кіберзагроз.

Об'єктом дослідження є процес надання хмарних веб сервісів в інформаційно-комунікаційних сервісно-орієнтованих мережах.

Предметом дослідження є моделі, методи та алгоритми підвищення ефективності функціонування хмарних систем.

В процесі проведення досліджень використано методи теорії систем масового обслуговування, нечіткої логіки, теорії ймовірності та математичної статистики та імітаційного моделювання. Для підтвердження теоретичних результатів застосовано експериментальні методи дослідження.

У вступі обґрунтовано актуальність теми дисертаційної роботи, сформульовано мету і визначені завдання для її досягнення, наукову новизну та практичне значення отриманих результатів. Наведено дані про впровадження результатів роботи, їх апробацію, публікації та особистий внесок здобувача.

У першому розділі **«Підходи та особливості інтеграції cloud архітектур в інформаційно-комунікаційні мережі»** розглянуто основні принципи побудови, підходи та особливості інтеграції хмарних систем і інформаційно-комунікаційні мережі провайдерів телекомунікацій. Проведено аналіз основних

вимог щодо якості надання послуг в сервісно-орієнтованих телекомунікаційних мережах. Встановлено, що попри усі переваги, які підприємства і провайдери отримують при інтеграції хмарних систем, їх ефективність функціонування погіршується за рахунок динамічного характеру хмарного середовища та різноманітністю запитів користувачів. Під'єднання хмарної інфраструктури займає досить тривалий час та потребує відповідної кваліфікації системних адміністраторів, а реалізація та налаштування Оркестратора, для управління таким компонентом мережі є складним процесом. Аналіз основних підходів, щодо інтеграції хмарних систем, показав необхідність врахування ступеню віртуалізації та їх величини, особливостей обробки потоків запитів, при оцінці ефективності функціонування хмарних систем. Надмірне використання віртуальних серверів привело до того, що для їх розгортання та підтримки працездатності необхідне залучення додаткових канальних ресурсів на рівнях комунікацій адміністратор додатків - сервер обробки – сервер зберігання даних. Динамічність розгортання як фізичних так і віртуальних машин, породжують проблему доступності та відмовостійкості обчислювальних ресурсів для обслуговування потоків запитів.

Динамічна конфігурація мережі за допомогою контролера мережі без оцінки ефективності функціонування апаратного і програмного забезпечення мережевих пристроїв привела до того, що сьогодні більшість провайдерів телекомунікаційних мереж частково впроваджують цю технологію, але при цьому виникають складнощі з управлінням інтегрованою мережевою інфраструктурою. У зв'язку з цим проблема зводиться не лише до підвищення ефективності функціонування інформаційно-комунікаційних сервісно-орієнтованих мереж, а й підвищення захищеності даних від несанкціонованого доступу, що є особливо актуальним на сьогоднішній день.

У другому розділі **«Моделі та методи підвищення ефективності функціонування хмарних систем як компонентів сервісно-орієнтованих мереж»** удосконалено математичну модель оцінки ефективності

функціонування хмарних систем в умовах групового надходження запитів від користувачів. Модель базується на двоступеневій техніці апроксимації та забезпечує повний імовірнісний розподіл часу очікування запитів, часу відгуку щодо виконання запитів і кількості запитів у системі. Удосконалена модель враховує можливість обслуговування при груповому надходженні запитів та розподілений час обслуговування запитів, що дасть змогу в моменти пікових навантажень не порушувати працездатність мережі з високим ступенем віртуалізації за рахунок зменшення тривалості очікування запитів в черзі. Модель дозволяє знаходити баланс між кількістю віртуальних машин, що здійснюють обслуговування групових потоків запитів та кількістю запитів, що вже знаходяться на обслуговуванні в системі.

Доведено адекватність моделі оцінки ефективності функціонування хмарних систем в умовах обслуговування групових потоків запитів. Визначено, що продуктивність роботи хмарних систем значно залежить від коефіцієнту варіації (CoV), часу обслуговування запитів і розміру групового потоку (тобто кількості запитів у груповому потоці запитів). Чим більший розмір потоку та/або значення коефіцієнта варіації часу обслуговування запитів, тим довша тривалість відгуку. Встановлено, що обидва показники ефективності можуть бути поліпшені за рахунок гомогенізації, отриманої шляхом поділу вхідних групових потоків на основі кількості запитів та/або коефіцієнту варіації часу обслуговування запитів та обслуговування підпотоків окремими хмарними центрами. Встановлено, в умовах надходження великих групових потоків запитів та/або великого значення CoV можна досягти підвищення ефективності функціонування хмарних центрів шляхом групування запитів за критерієм однорідності.

Удосконалено метод кластеризації вузлів хмарних систем в умовах обслуговування групових потоків запитів, що дозволить підвищити ефективність функціонування хмарних систем, в яких одночасне обслуговування групових потоків запитів забезпечує велика кількість

віртуальних машин. Такий підхід базується на групуванні в кластер пулу віртуальних машин, які будуть використовуватися під обслуговування чітко визначеного групового потоку із врахуванням географічної близькості вузлів між собою та моніторингу доступних програмно-апаратних та телекомунікаційних ресурсів, що дасть змогу підвищити життєвий цикл фізичних машин за рахунок збільшення рівня залишкової енергії.

Розроблено алгоритм кластеризації і вибору головного вузла кластера, особливістю якого є визначення головного вузла кластера із врахуванням центральності по діаграмах Вороного, правил нечіткої логіки та моніторингу параметрів вузлів. Ухвалення рішення про вибір головного вузла здійснюється методами нечіткої логіки за допомогою правила Мамдані та центру ваги, а також із врахуванням достатніх програмно-апаратних ресурсів для подальшого функціонування вузла в якості центроїда кластеру.

У третьому розділі **«Моделювання та дослідження ефективності функціонування хмарних систем у інформаційно-комунікаційних сервісно-орієнтованих мережах»** проведено моделювання та дослідження ефективності запропонованого алгоритму кластеризації на основі комплексного критерію ефективності: тривалості пошуку маршрутів, споживання енергії в мережі і життєвий цикл мережі. Завдяки реалізації розробленого методу кластеризації вузлів хмарних систем вдалося досягти зменшення тривалості пошуку маршруту між довільною парою віртуалізованих вузлів в межах одного кластеру у 1,3 рази. Завдяки визначенню головного вузла кластера із врахуванням центральності по діаграмах Вороного, правил нечіткої логіки та моніторингу параметрів вузлів вдалося досягти зменшення тривалості пошуку маршруту між довільною парою віртуалізованих вузлів, які обслуговують один і той же груповий потік у 2,5 рази. Доведено, що розроблений алгоритм у порівнянні із CHS-FL-VD зменшує споживання енергії на 45% і продовжує життєвий цикл мережі на 8% в порівнянні з відомими алгоритмами. Даний

метод дозволяє підтримувати якість надання послуг користувачам, особливо в умовах групового надходження запитів.

Для доведення ефективності запропонованих методів та моделей підвищення ефективності функціонування хмарних систем проведено дослідження на основі розгорнутої моделі інформаційно-комунікаційної сервісно-орієнтованої мережі з використанням CDN серверів як PaaS сервісу та хмарної архітектури Azure Cloud. Проведено реалізацію удосконаленої математичної моделі оцінки ефективності функціонування хмарних систем в умовах групового надходження запитів, \ та методу кластеризації вузлів хмарних систем. В результаті дослідження запропонованих рішень досягнуто зменшення тривалості завантаження статичного контенту з кеш-серверів (які виступали головними вузлами кластерів, визначеними за допомогою алгоритму вибору головного кластерного вузла з використанням центральності по діаграмах Вороного, правил нечіткої логіки та моніторингу параметрів вузлів) у порівнянні із завантаженням з базового сервера в 4 рази. Використання пропускної здатності каналу при доступі до контенту із віддалених серверів зростає до 700 Кбіт/с. Дослідження показали, що контент не повинен кешуватись на POP або HTTP-клієнтом. Лише в одному випадку на 0,05 с. кешована версія запитуваного ресурсу не була знайдена на найближчому до клієнта POP.

У четвертому розділі **«Реалізація мережевого захисту хмарних веб сервісів у програмно-конфігурованих сервісно-орієнтованих мережах»** розроблено інтегровану архітектуру системи інтелектуального виявлення DDoS атак у інформаційно-комунікаційних мережах. Основною особливістю архітектури є реалізація Підсистеми аналізу лог файлів, яка є елементом інтегрованої системи управління, що аналізує потенційні проблеми і запускає запобіжні механізми, або може повідомляти системних адміністраторів про порушення безпеки мережі.

Розроблено комплексний підхід до аналізу логів, особливістю якого є проведення паралельного дослідження ентропійних властивостей потоків запитів. Для його реалізації використовується два методи: структурований та неструктурований аналіз даних.

Запропоновано неструктурований метод аналізу даних, який базується на дослідженні ентропійних властивостей записів лог журналів. Метод реалізує алгоритм знаходження ентропії записів, створених за певний проміжок часу та аналізі останніх значень ентропії для всіх файлів журналу в системі.

Розроблено структурований метод аналізу даних, який базується на визначенні метрики поведінки трафіку використовуючи підхід Кульбака — Лейблера для виявлення аномалій потоку за часом тривалості сесії.

Проведено моделювання та дослідження ефективності інтеграції архітектури інтелектуального виявлення DDoS атак на основі розробленої імітаційної моделі інформаційно-комунікаційної сервісно-орієнтованої мережі. Доведено, що використання комплексного підходу до аналізу лог журналів із використанням як структурованого так і неструктурованого аналізу службових файлів дало змогу відстежувати як наявні загрози, так і прогнозувати атаки на мережу в цілому за рахунок використання елементу машинного навчання як окремої підсистеми інтегрованої архітектури управління мережею. Реалізація машинного навчання дозволило класифікувати клієнтів по категоріям на основі значень ентропії та виявляти зловмисника. В результаті використання такого навчання та відповідно постійного моніторингу сесій, SDN контролер блокуватиме IP домени, з яких лише розпочинаються DDoS атаки.

Висновки до дисертації включають узагальнені результати дослідження та рекомендації щодо їх практичного застосування. Теоретичне значення роботи полягає в тому, що її результати дають змогу забезпечити ефективне функціонування хмарних систем при їх інтеграції у сучасні інформаційно-комунікаційні сервісно-орієнтовані мережі шляхом удосконалення методів та моделей оцінки ефективності функціонування хмарних систем в умовах

обслуговування групових потоків запитів та із реалізацією інтегрованої архітектури системи інтелектуального виявлення атак, що дасть змогу забезпечити мережевий захист хмарних веб сервісів від атак зловмисників.

Основні результати дисертаційної роботи використано і впроваджено з метою підвищення ефективності функціонування хмарних систем при їх інтеграції у сучасні інформаційно-комунікаційні сервісно-орієнтовані мережі провайдерів телекомунікацій, зокрема ТОВ «Аркада-Х» та ТзОВ «МаксіТех», що підтверджено актами впровадження.

Ключові слова: інформаційно-комунікаційна мережа, сервісно-орієнтована мережа, угода про рівень послуг, ступінь віртуалізації, груповий потік запитів, захист веб сервісів, виявлення атак.

Список публікацій здобувача:

Наукові праці, у яких опубліковані основні результати дисертації

1. N. Peleh, O. Shpur, M. Klymash, “Intelligent detection of DDoS attacks in SDN networks”, *Lecture Notes in Electrical Engineering*., Vol. 831 : Future intent-based networking. On the QoS robust and energy efficient heterogeneous software defined networks, pp. 210–222, 2022.

2. N. Peleh, S. Zhuravel, O. Shpur, O. Rybytska, “Structured and unstructured log analysis as a methods to detect DDoS attacks in SDN networks,” *Internet of Things (IoT) and Engineering Applications*, pp. 1-9, Sept. 2021.

3. Н.В. Пелех, О.М. Шпур, М.М. Климаш, “Модель оцінки ефективності функціонування хмарного центру з високим ступенем віртуалізації та в умовах групового надходження запитів,” *Проблеми телекомунікацій*, №2(27), с.82-98. 2020.

4. М. М. Климаш, О.М. Шпур, Н.В. Пелех, “Моніторинг доступності веб-сервісу в розподілених інфокомунікаційних системах,” *Вісник Університету "Україна". Серія : Інформатика, обчислювальна техніка та кібернетика.*, №1(28), с.135-150, 2020.

5. О.А. Лаврів, О.М. Шпур, Н.В. Пелех, “Забезпечення доступу до статичного контенту із використанням CDN мереж як PaaS сервісу Azure Cloud,” *Системи обробки інформації - Х: Харк. ун-т Повітр. Сил ім. Івана Кожедуба*, №3(158), с. 83-91, 2019.

Наукові праці, які засвідчують апробацію матеріалів дисертації

6. O. Shpur, Y. Pyrih, T. Havryliv, N. Peleh, O. Urikova, A. Branytskyi, "Development of a road traffic monitoring system," *2021 IEEE 16th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, Polyana, Ukraine, 2021, pp. 10-14.

7. M. Klymash, O. Shpur, N. Peleh, O. Maksysko "Concept of intelligent detection of DDoS attacks in SDN networks using machine learning," *2020 8th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*, Kharkiv, 2020, pp. 609-612.

8. M. Klymash, O. Shpur, N. Peleh, S. Hladun "Monitoring of web service availability in distributed infocommunication systems," *2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, Lviv-Slavske, Ukraine, 2020, pp. 723-728.

9. M. Klymash, O. Shpur, O. Lavriv and N. Peleh "Information security in virtualized data center network," *2019 3rd International Conference on Advanced Information and Communications Technologies (AICT)*, Lviv, Ukraine, 2019, pp. 419-422.

10. M. Klymash, O. Shpur, N. Peleh, O. Lavriv, R. Bak, O. Skybinskyi "Increasing the accessibility of static content using CDN networks as PaaS," *2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, Polyana, Ukraine, 2019, pp. 2/57–2/60.

11. M. Klymash, O. Shpur, N. Peleh, I. Lutsiuk "Clustering model of cloud centers for Big Data processing" *2018 14th International Conference on Advanced*

Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavske, Ukraine, 2018, pp. 268–271

12. М.М. Климаш, О.М. Шпур, Н.В. Пелех "Модель кластеризації хмарних дата-центрів в умовах передачі та захисту потоків Big Data," *І міжнародна науково-практична конференція "Проблеми кібербезпеки інформаційно телекомунікаційних систем"*, м. Київ, 2018 р., с. 191- 197.

ABSTRACT

Peleh N.V. Improving the efficiency of cloud systems for information and communication service-oriented networks. – Qualification research paper as a manuscript.

The thesis for the Doctor of Philosophy Degree in the specialty 172 – Telecommunications and Radio Engineering. – Lviv Polytechnic National University, Ministry of Education and Science of Ukraine, Lviv, 2021.

The rapid development and promotion of information services require new approaches to systems that ensure their functioning. It would seem that modern technologies, such as cloud computing, are intended to satisfy access to a large amount of computing power, as they operate in an environment of resource aggregation and provide a unified systems approach to their management. Their integration in the network of providers is especially important when cloud systems become part of information and communication networks. On the one hand, the provider has the opportunity to expand the range of services it provides to users, on the other hand, the provider increases the flow of traffic transmitted through such a network. Accordingly, not only new approaches to improvement, storage, and management are needed, but also to traffic analysis and, as a consequence, to the processing of service data, as its number increases exponentially. In addition, excessive use of virtual servers can further complicate the task, as they will require the involvement of additional channel resources to provide the communication levels between the application administrator, processing server and storage server. In addition, the management of such an integrated network infrastructure is reduced to the problem of increasing the security of data from unauthorized access. This requires certain systems to monitor the status of the network, that will analyze possible problems that may occur during operation, and run security systems in the service itself or notify administrators of a possible threat. Such systems must promptly monitor all the changes that occur with the system including attacks and technical failures, under which the service ceases to function properly.

The thesis solves the scientific and practical task of improving the efficiency of cloud systems in their integration into modern information and communication service-oriented networks by developing a method of clustering server nodes, improving models for assessing the effectiveness of cloud systems in servicing group requests, and developing systems for intelligent detection of network attacks.

The purpose of the presented dissertation is to increase the efficiency of cloud systems in their integration into modern information and communication service-oriented networks with the implementation of intelligent detection of cyber threats.

The object of research is the process of providing cloud web services in information and communication service-oriented networks.

The subject of research is models, methods, and algorithms to improve the efficiency of cloud systems.

The research methods of queuing systems theory, fuzzy logic, probability theory, mathematical statistics, and simulation modeling were used in the research process. Experimental research methods were used to confirm the theoretical results.

The introduction substantiates the relevance of the topic of the thesis, formulates the goal and identifies objectives for its achievement, scientific novelty and practical significance of the results. Data on the implementation of the results, their approbation, publications and personal contribution of the applicant are given.

In the first chapter **"Approaches and features of integration of cloud architectures in information and communication networks"** the basic principles of construction, approaches, and features of integration of cloud systems and information and communication networks of telecommunications providers are overviewed. An analysis of the basic requirements for the quality of services in service-oriented telecommunications networks is performed. It is established that despite all the benefits that companies and providers receive from the integration of cloud systems, their efficiency is impacted by the dynamic nature of the cloud environment and the diversity of user requests. Connecting the cloud infrastructure takes a long time and requires the appropriate qualifications of system administrators,

and the implementation and configuration of the Orchestrator to manage such a network component is a complex process. The analysis of the main approaches to the integration of cloud systems showed the need to take into account the degree of virtualization and the size of cloud systems, features of the processing of request flows when assessing the effectiveness of their operation. Excessive use of virtual servers has led to the fact that their deployment and maintaining the performance requires the involvement of additional channel resources to provide communications between application administrator, processing server and storage server. Dynamic deployment of both physical and virtual machines creates a problem of availability and fault tolerance of computing resources to service request flows.

The dynamic configuration of the network using a network controller without assessing the performance of network hardware and software has led to the fact that today most telecommunications network providers partially implement this technology, but there are difficulties with managing integrated network infrastructure. In this regard, the problem is not only to increase the efficiency of information and communication service-oriented networks but also to increase the protection of data from unauthorized access, which is especially relevant today.

In the second chapter **"Models and methods for improving the efficiency of cloud systems as components of service-oriented networks"** the improved the mathematical model for assessing the effectiveness of cloud systems in terms of group receipt of requests from users has been developed. The model is based on a two-step approximation technique and provides a full probability distribution of queue time, query response time, and the number of queries in the system. The improved model takes into account the possibility of servicing the incoming groups of requests and distributed service time of requests, which will allow during peak loads not to disrupt the network with a high degree of virtualization by reducing the waiting time for queues. The model allows to find a balance between the number of virtual machines that service group request flows and the number of requests that are already in service in the system.

The adequacy of the model for assessing the efficiency of cloud systems in terms of servicing the group request flows is proved. It is determined that the performance of cloud systems significantly depends on the coefficient of variation (CoV) of the request service time and the size of the group flow (i.e., the number of requests in the group flow of requests). The larger the flow rate and/or the value of the coefficient of variation of the service time of requests, the longer the response time. It has been found that both efficiency indicators can be improved by homogenization obtained by dividing incoming group streams based on the number of requests and/or the coefficient of variation of request service time and sub-stream service by individual cloud centers. It is established that in the conditions of large group flow of requests and/or large value of CoV it is possible to achieve an increase in the efficiency of cloud centers by grouping requests by the criterion of homogeneity.

The method of clustering nodes of cloud systems in the conditions of service of group flows of requests is improved which will allow increasing the efficiency of functioning of cloud systems in which simultaneous service of group flows of requests is provided by a large number of virtual machines. This approach is based on clustering a pool of virtual machines that will be used to serve a well-defined group flow, taking into account the geographical proximity of nodes and monitoring available software and hardware, and telecommunications resources, which will increase the life cycle of physical machines by increasing residual energy.

An algorithm for clustering and selection of the main node of the cluster has been developed. The feature of the algorithm is the definition of the main node of the cluster taking into account centrality according to Voronoi diagrams, fuzzy logic rules, and monitoring of node parameters. The decision to choose the main node is made by fuzzy logic using the Mamdani rule and the center of gravity, as well as taking into account sufficient software and hardware resources for the further functioning of the node as a centroid of the cluster.

In the third chapter "**Modeling and study of cloud system efficiency in information and communication service-oriented networks**" it has been performed the modeling and study of the efficiency of the proposed clustering algorithm based on a comprehensive criterion of efficiency: route search duration, network energy consumption, and network life cycle. Due to the implementation of the developed method of clustering nodes of cloud systems, it was possible to reduce the duration of the route search between any pair of virtualized nodes within one cluster by 1.3 times. By defining the main node of the cluster, taking into account the centrality of Voronoi diagrams, fuzzy logic rules, and monitoring node parameters, it was possible to reduce the duration of the route search between any pair of virtualized nodes serving the same group stream by 2.5 times. It is proved that the developed algorithm in comparison with CHS-FL-VD reduces energy consumption by 45% and prolongs the life cycle of the network by 8% compared to known algorithms. This method allows to maintain the quality of service for users, especially in the context of group requests.

To prove the effectiveness of the proposed methods and models for improving the efficiency of cloud systems, a study was conducted based on a developed model of information and communication service-oriented network using CDN servers as a PaaS service and cloud architecture Azure Cloud. The improved mathematical model for estimating the efficiency of cloud systems in terms of group requests, and the method of clustering nodes of cloud systems have been implemented in the scope of the network model. As a result of research of the proposed solutions the duration of loading of static content from cache servers (which acted as the main nodes of clusters defined the by the algorithm that selects the main cluster node based on the centrality of Voronoi diagrams, fuzzy logic rules, and monitoring node parameters) comparing to loading from the base server was decreased by 4 times. Channel bandwidth usage increases to 700 Kbps when accessing content from remote servers. Research has shown that content should not be cached on a POP or HTTP client.

Only in one case for 0.05 s. the cached version of the requested resource was not found on the nearest POP client.

In the fourth chapter **"Implementation of network protection of cloud web services in software-configured service-oriented networks"** an integrated architecture of the system for intelligent detection of DDoS attacks in information and communication networks has been developed. The main feature of the architecture is the implementation of the Log File Analysis Subsystem, which is part of an integrated management system that analyzes potential problems and triggers security mechanisms, or can report to system administrators about network security breaches.

A comprehensive approach to the analysis of logs, the feature of which is the parallel study of entropy properties of query flows has been developed. Two methods are used for its implementation: structured and unstructured data analysis.

An unstructured method of data analysis, which is based on the study of entropic properties of logbook records has been proposed. The method implements an algorithm for finding the entropy of records created over a period of time and analyzing the latest entropy values for all log files in the system.

A structured method of data analysis, which is based on determining the metrics of traffic behavior using the Kulbak - Leibler approach to detect flow anomalies over the duration of the session has been developed.

Modeling and research of efficiency of integration of architecture of intellectual detection of DDoS attacks on the basis of the developed simulation model of information-communication service-oriented network has been carried out. It is proved that the use of a comprehensive approach to the analysis of log logs using both structured and unstructured analysis of service files allowed to track existing threats and predict attacks on the network as a whole by using machine learning as a separate subsystem of integrated network management architecture. The implementation of machine learning allowed to classify customers into categories based on entropy values and identify the attacker. As a result of using such training

and, accordingly, constant monitoring of sessions, the SDN controller will block IP domains from which DDoS attacks are just starting.

The conclusions of the thesis include generalized research results and recommendations for their practical application. The theoretical significance of the work is that its results allow to ensure effective functioning of cloud systems when integrating them into modern information and communication service-oriented networks by improving methods and models for assessing the effectiveness of cloud systems in group request flows and integrated architecture ofintelligent attack detection systems, which will provide network protection of cloud web services from attackers.

The main results of the dissertation are used and implemented to improve the efficiency of cloud systems in their integration into modern information and communication service-oriented networks of telecommunications providers, including LLC "Arcade-X" and LLC "MaxiTech", which is confirmed by implementing acts.

Keywords: information and communication network, service-oriented network, service level agreement, degree of virtualization, group flow of requests, protection of web services, detection of attacks.

The list of author's publications:

Proceedings where basic scientific results of thesis were published

1. N. Peleh, O. Shpur, M. Klymash, "Intelligent detection of DDoS attacks in SDN networks", *Lecture Notes in Electrical Engineering*., Vol. 831 : Future intent-based networking. On the QoS robust and energy efficient heterogeneous software defined networks, pp. 210–222, 2022.

2. N. Peleh, S. Zhuravel, O. Shpur, O. Rybytska, "Structured and unstructured log analysis as a methods to detect DDoS attacks in SDN networks," *Internet of Things (IoT) and Engineering Applications*, pp. 1-9, Sept. 2021.

3. N. Peleh, O. Shpur, M. Klymash, “Model for evaluating the effectiveness of the cloud center with a high degree of virtualization and in the context of group requests,” *Problems of Telecommunications*, №2 (27), p.82-98. 2020

4. M. Klymash, O. Shpur, N. Peleh “Monitoring the availability of web service in distributed infocommunication systems,” *Bulletin of the University "Ukraine". Series: Informatics, Computing and Cybernetics.*, №1(28), p.135-150, 2020.

5. O. Lavriv, O. Shpur, N. Peleh, “Providing access to static content using CDN networks as a PaaS of the Azure cloud service,” *Information processing systems - Kh: Kharkiv. University of Air. Forces them. Ivan Kozhedub*, №3(158), p. 83-91, 2019.

Proceedings that certify an approvement of thesis materials

6. O. Shpur, Y. Pyrih, T. Havryliv, N. Peleh, O. Urikova, A. Branytskyi, "Development of a road traffic monitoring system," *2021 IEEE 16th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, Polyana, Ukraine, 2021, pp. 10-14.

7. M. Klymash, O. Shpur, N. Peleh, O. Maksysko "Concept of intelligent detection of DDoS attacks in SDN networks using machine learning," *2020 8th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*, Kharkiv, 2020, pp. 609-612.

8. M. Klymash, O. Shpur, N. Peleh, S. Hladun "Monitoring of web service availability in distributed infocommunication systems," *2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, Lviv-Slavske, Ukraine, 2020, pp. 723-728.

9. M. Klymash, O. Shpur, O. Lavriv and N. Peleh "Information security in virtualized data center network," *2019 3rd International Conference on Advanced Information and Communications Technologies (AICT)*, Lviv, Ukraine, 2019, pp. 419-422.

10. M. Klymash, O. Shpur, N. Peleh, O. Lavriv, R. Bak, O. Skybinskyi "Increasing the accessibility of static content using CDN networks as PaaS," *2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, Polyana, Ukraine, 2019, pp. 2/57–2/60.

11. M. Klymash, O. Shpur, N. Peleh, I. Lutsiuk "Clustering model of cloud centers for Big Data processing" *2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET)*, Lviv-Slavske, Ukraine, 2018, pp. 268–271

12. M.M. Klymash, O.M. Shpur, H.V. Peleh "Model of clustering of cloud data centers in terms of transmission and protection of Big Data streams," *I International Scientific and Practical Conference "Problems of cybersecurity of information and telecommunications systems"*, Kyiv, 2018, p. 191- 197.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

PaaS - Platform-as-a-Service, платформа як сервіс.

IaaS - Infrastructure-as-a-Service, інфраструктура як сервіс.

SaaS - Platform as a service, програмне забезпечення як послуга.

AI - Artificial Intelligence, штучний інтелект.

SOA – Service-Oriented Architecture, сервісно-орієнтована архітектура.

ACL – Access Control List, список контролю доступу.

VLAN – Virtual Local Area Network — віртуальна локальна комп'ютерна мережа.

QoS – Quality of Service, якість обслуговування

VM – Virtual Machine, віртуальні машини

IP – Internet Protocol,інтернет протокол, в мережах з комутаціє пакетів.

SDN – Software-Defined Networking, програмно-конфігурована мережа.

IDS – Intrusion Detection System, система виявлення вторгнень.

IPS – Intrusion Prevention System, система запобігання вторгнень.

API – Application Programming Interface, формалізований набір визначень для взаємодії програмного забезпечення.

ПКМ – Програмно-конфігурована мережа.

TCP – Transmission Control Protocol, протокол керування передачею

UDP – User Datagram Protocol, протокол датаграм користувача

ISP – Internet Service Provider, провайдер інтернет мереж

CSP – Constrained Shortest Path Problem, розширення проблеми знаходження найкоротших шляхів, на які накладаються певні обмеження.

SLA– Service-Level Agreement, угода про рівень послуг.

QoE – Quality of Experience, оцінка досвіду використання користувачем певної послуги.

SD-WAN - Software-defined networking in a wide area network, програмно визначена глобальна мережа

BSS - Business Support System, прикладного програмного забезпечення внутрішніх бізнес-процесів

OSS - Operation Support System, прикладного програмного забезпечення внутрішніх бізнес-процесів провайдерів послуг

CPE - customer premises equipment, елементи приватної хмари записуються вручну

RMS - Record Management System, система управління записами

NE - Node equipment, вузли мережі

NBI - Northbound interface, програмний інтерфейс, за допомогою якого програма представляє низькорівневі деталі додатку, серверний інтерфейс

VPN - Virtual Private Network, віртуальна приватна мережа

API VPC - Application Programming Interface Virtual Privat Cloud, сервісно-орієнтовані інтерфейси, які захищають технічні деталі, забезпечуючи швидку комунікацію між платформою агрегації кількох хмар і мережею провайдера

DSCP – Differentiated Services Code Point, поле коду диференційованої послуги.

ITU – International Telecommunication Union, Міжнародний союз телекомунікацій.

LLDP – Link Layer Discovery Protocol, протокол для обміну інформацією про конфігурацію між мережевими пристроями.

MAC – Media Access Control, адреса мережевої карти.

NFV – Network Functions Virtualization, віртуалізація мережних функцій.

ToS – Type of Service, Рівень пріоритету IP, вид послуги.

RTT – Round-Trip Time, кругова затримка пересилання пакетів.

ВСТУП

Актуальність теми.

Хмарні обчислення стали важливою інфраструктурою в галузі інформаційно-комунікаційних технологій. Останнім часом SaaS, PaaS і IaaS використовуються в основному в корпоративних компаніях. Багато хмарних сервісів вимагають сумісних послуг, щоб розширити можливості обслуговування та бізнес-ринок. Крім того, у багатьох галузях з'являються програми на основі штучного інтелекту (AI, Artificial Intelligence) де хмарні системи використовуються для швидких обчислень в процесі машинного навчання. Звичайні хмарні послуги на основі віртуальних машин стикаються з багатьма проблемами, що виникають на практиці у процесі інтеграції із інформаційно-комунікаційними мережами провайдерів послуг. Зокрема, при традиційних сервісно-орієнтованих архітектурах сьогодні неможливо контролювати та передбачити сплесковість трафіку для гарантування наскрізної якості обслуговування, забезпечити стійкість проти мережевих загроз, а також структурно адаптуватись під мінливі вимоги користувачів. Відповідно, традиційні хмарні системи характеризуються низкою проблем, вирішення яких вимагає внесення змін в існуючу архітектуру, а саме нових способів інтеграції програмно-конфігурованих мереж здатних розв'язувати задачу розподілу ресурсів оперативно, у відповідності до властивостей вхідних потоків, які передаються у конкретний момент часу через конкретні телекомунікаційні вузли. Такі підходи повинні базуватись на удосконалених методах та моделях розподілу обчислювальних ресурсів, забезпечуючи при цьому високу масштабованість, енергоефективність, швидкодію, гнучкість, низьку операційну складність та ресурсоємність.

Дослідженням методів та моделей підвищення ефективності функціонування хмарних систем, а саме підвищити надійність, продуктивність, гнучкість та визначити необхідні заходи для забезпечення якісного надання послуг активно займаються, як науковці України: Глоба Л.С., Лемешко О.В.,

Ложковський А.Г., Соловська І.М., так і іноземні: Sania Malik, Johnson Thomas, Rajkumar Buyya, Athanasios Vasilakos, зокрема, особливу увагу слід звернути на останні роботи спрямовані на дослідження програмно-конфігурованих сервісно-орієнтованих мереж.

Таким чином, зростання кількості та розмаїття контенту, розвиток веб-сервісів і масштабів, що їх охоплюють, призвели до необхідності розв’язання науково-практичного завдання підвищення ефективності функціонування хмарних систем при їх інтеграції у сучасні інформаційно-комунікаційні сервісно-орієнтовані мережі шляхом розроблення методу кластеризації серверних вузлів, удосконалення моделей оцінки ефективності функціонування хмарних систем в умовах обслуговування групових потоків запитів та розвитку систем інтелектуального виявлення мережевих атак.

Зв’язок роботи з науковими програмами, планами, темами.

Тематика дисертаційного дослідження підібрана і виконувалась у відповідності до наукового напряму кафедри телекомунікацій Національного університету «Львівська політехніка» - «Інфокомунікаційні системи та мережі»

Мета і завдання дослідження. Метою представленої дисертаційної роботи є підвищення ефективності функціонування хмарних систем при їх інтеграції у сучасні інформаційно-комунікаційні сервісно-орієнтовані мережі із реалізацією інтелектуального виявлення кіберзагроз.

Досягнення поставленої мети здійснюється розв'язанням таких завдань:

1. Аналіз існуючих методів підвищення ефективності функціонування хмарних систем при їх інтеграції у сучасні інформаційно-комунікаційні мережі для вирішення задач передавання та обслуговування групових потоків запитів.

2. Удосконалення математичної моделі оцінки ефективності функціонування хмарних систем в умовах обслуговування групових потоків запитів.

3. Удосконалення методу кластеризації вузлів хмарних систем в умовах обслуговування групових потоків запитів.

4. Розробка алгоритму кластеризації і вибору головного вузла кластера у хмарних центрах в умовах групового надходження запитів на основі діаграм Вороного, правил нечіткої логіки та моніторингу параметрів доступних ресурсів.

5. Моделювання та дослідження ефективності запропонованих рішень на основі розгорнутої моделі інфокомунікаційної сервісно-орієнтованої мережі.

6. Розробка інтегрованої архітектури інтелектуального виявлення DDoS атак у інформаційно-комунікаційних мережах із використання машинного навчання та лог аналізу.

7. Удосконалення методів структурованого та не структурованого лог аналізу для реалізації моніторингу доступності веб сервісів з метою виявлення потенційних атак.

8. Моделювання та дослідження ефективності інтеграції архітектури інтелектуального виявлення DDoS атак у сучасні інформаційно-комунікаційні сервісно-орієнтовані мережі.

Об'єктом дослідження є процес надання хмарних веб сервісів в інформаційно-комунікаційних сервісно-орієнтованих мережах.

Предмет дослідження: моделі, методи та алгоритми підвищення ефективності функціонування хмарних систем.

Методи дослідження. В процесі проведення досліджень використано методи теорії систем масового обслуговування, нечіткої логіки, теорії ймовірності, математичної статистики та імітаційного моделюванн. Для підтвердження теоретичних результатів застосовано експериментальні методи дослідження.

Наукова новизна отриманих результатів.

1. Удосконалено математичну модель оцінки ефективності функціонування хмарних систем в умовах обслуговування групових потоків запитів, яка, на відмінну від існуючих, враховує можливість обслуговування при груповому надходженні запитів та розподілений час обслуговування

запитів, що дасть змогу в моменти пікових навантажень не порушувати працездатність мережі з високим ступенем віртуалізації за рахунок зменшення тривалості очікування запитів в черзі.

2. Удосконалено метод кластеризації вузлів хмарних систем в умовах обслуговування групових потоків запитів, який відрізняється від відомих врахуванням правил нечіткої логіки та центральності по діаграмах Вороного для визначення головного вузла кластера, що дало змогу зменшити тривалість пошуку маршруту між довільною парою вузлів та підвищити ефективність інтеграції хмарних систем.

3. Вперше запропоновано інтегровану архітектуру системи інтелектуального виявлення DDoS атак у інформаційно-комунікаційних мережах на основі комплексного підходу до аналізу лог журналів із використанням як структурованого так і неструктурованого аналізу службових файлів, що дає змогу відстежувати як наявні загрози, так і прогнозувати атаки на мережу в цілому.

Практичне значення одержаних результатів полягає в тому, що:

1. Удосконалено метод кластеризації вузлів хмарних систем в умовах обслуговування групових потоків запитів, що дало можливість зменшити тривалість пошуку маршруту між довільною парою віртуалізованих вузлів, які обслуговують один і той же груповий потік у 2,5 рази.

2. Розроблено алгоритм вибору головного кластерного вузла з використанням центральності по діаграмах Вороного, правил нечіткої логіки та моніторингу параметрів вузлів, що дало змогу зменшити споживання енергії на 45% і продовжити життєвий цикл мережі на 8% та підтримувати якість надання послуг користувачам в умовах обслуговування групових потоків запитів.

3. Реалізація удосконаленої математичної моделі оцінки ефективності функціонування хмарних систем дала змогу зменшити тривалість завантаження статичного контенту з кеш-серверів (які виступали головними вузлами кластерів, визначеними з використанням алгоритму вибору головного

кластерного вузла з використанням центральності по діаграмах Вороного, правил нечіткої логіки та моніторингу параметрів вузлів) у порівнянні із завантаженням з базового сервера в 4 рази.

4. Розроблено інтегровану архітектуру системи інтелектуального виявлення загроз на основі комплексного підходу до аналізу лог журналів із використанням як структурованого так і неструктурованого аналізу службових файлів. Реалізація такої системи дозволяє відстежувати як наявні загрози, так і прогнозувати атаки на мережу в цілому за рахунок використання елементу машинного навчання як окремої підсистеми інтегрованої архітектури управління мережею, що дасть змогу проводити інтелектуальне виявлення атак використовуючи інформацію про стан потоку, тривалість сесії та джерела його походження, а також навчити SDN контроллер блокувати «подібні» потоки запитів без втручання системних адміністраторів.

Наукові та практичні результати виконаних досліджень використані в навчальному процесі кафедри телекомунікацій Національного університету «Львівська політехніка» для модернізації курсу лекцій з дисципліни «Технології інформаційно-комунікаційних мереж», «Розподілені системи та cloud технології».

Основні результати дисертаційної роботи використано і впроваджено з метою підвищення ефективності функціонування хмарних систем при їх інтеграції у сучасні інформаційно-комунікаційні сервісно-орієнтовані мережі провайдерів телекомунікацій, зокрема ТОВ «Аркада-Х» та ТзОВ «МаксіТех», що підтверджено актами впровадження.

Особистий внесок здобувача. Основні наукові результати дисертаційної роботи отримано автором самостійно. У працях, опублікованих у співавторстві, внесок Пелеха Н.В. є вирішальним, зокрема авторові належать (*нумерація згідно Додатку Б*: у роботах [1, 7, 8, 11] – розроблення інтегрованої архітектури системи інтелектуального виявлення атак у інформаційно-комунікаційних мережах на основі комплексного підходу до аналізу лог журналів із

використанням як структурованого так і неструктурованого аналізу службових файлів, [3, 6] – розроблення алгоритмів структурованого та неструктурованого лог аналізу; [2] – удосконалення математичної моделі оцінки ефективності функціонування хмарних систем в умовах обслуговування групових потоків запитів, [4, 9] – моделювання та дослідження ефективності впровадження запропонованої математичної моделі оцінки ефективності функціонування та методу кластеризації вузлів хмарних систем в умовах обслуговування групових потоків на основі розгорнутої моделі інфокомунікаційної сервісно-орієнтованої мережі з використанням CDN серверів як PaaS сервісу та хмарної архітектури Azure Cloud; [5, 12] – удосконалення методу кластеризації вузлів хмарних систем в умовах обслуговування групових потоків; [10] – розроблення алгоритму визначення головного вузла кластера із врахуванням центральності по діаграмах Вороного, правил нечіткої логіки та моніторингу параметрів вузлів.

Апробація результатів дисертації. Основні наукові результати і положення дисертації представлені, доповідались та обговорені на 8-ми міжнародних науково-технічних конференціях та наукових семінарах: Міжнародних науково-технічних конференціях «Сучасні проблеми радіоелектроніки, телекомунікацій, комп’ютерної інженерії» (м. Львів-Славське 2018, 2020 рр.); International IEEE Conferences on Advanced Information and Communication Technologie (м. Львів, 2019 р.); Міжнародних науково-технічних конференціях «Досвід розробки та застосування приладо-технологічних САПР в мікроелектроніці» (м. Львів-Поляна, 2019, 2021 рр.); 8th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (м. Харків, 2020 р.); Міжнародних науково-практичних конференціях “Проблеми кібербезпеки інформаційно телекомунікаційних систем” (м. Київ, 2018, 2020 рр.). Крім цього, дисертаційна робота у повному обсязі представлена на наукових семінарах кафедри телекомунікацій Національного університету «Львівська політехніка».

Публікації. За результатами досліджень, які викладені у дисертаційній роботі, опубліковано 12 наукових праць, з них 3 статті у наукових фахових виданнях України та 1 стаття у науковому періодичному виданні інших держав, що входять до міжнародних науково-метричних баз даних, 1 стаття у науковому періодичному виданні інших держав; 7 публікацій у збірниках тез наукових конференцій (зокрема 6 – у виданнях, які входять до наукометричних баз даних Scopus та Web of Science).

Структура та обсяг роботи. Робота складається з переліку умовних скорочень, вступу, 4 розділів, висновків, списку використаних джерел і 2 додатків. Загальний обсяг роботи складає 170 сторінок друкарського тексту, із них 6 сторінок вступу, 115 сторінок основного тексту, 65 рисунків, 6 таблиць, список використаних джерел із 120 найменувань, 2 додатки на 4 сторінках. Додатки містять обрані акти впровадження результатів дисертаційної роботи, а також список праць автора.

РОЗДІЛ 1. ПІДХОДИ ТА ОСОБЛИВОСТІ ІНТЕГРАЦІЇ CLOUD АРХІТЕКТУР В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ МЕРЕЖІ

1.1. Принципи побудови та проблематика інтеграції хмарних систем в мережі провайдерів

З точки зору комунікації, людство еволюціонувало від скромного передавання інформації по двопровідній лінії до масового зростання інформації, що зараз охоплює світ і на сьогоднішній день опинилося на порозі нової ери з повсюдними комп'ютерами, програмним забезпеченням та інтелектуальними можливостями. Після трьох промислових революцій – перша, що характеризується механізацією завдяки винаходу парової машини, друга – винаходом електрики, а третя – після винаходу комп'ютерів – тепер світ переживає четверту промислову революцію на чолі з технологіями: хмарні обчислення, великі дані, Інтернет речей і штучний інтелект. Хмарні обчислення формують основний цифровий імпульс у цій четвертій промисловій революції, забезпечуючи достатній інтелект і обчислювальну потужність для терміналів Інтернету речей, які швидко розвиваються в тисячах галузей, для отримання необхідної інформації та знань із великих даних. Подібно до ролі, яку виконують електромережі, хмарні мережі тепер необхідні для передачі обчислювальної потужності різноманітним галузям по всьому світу (рис.1.1).

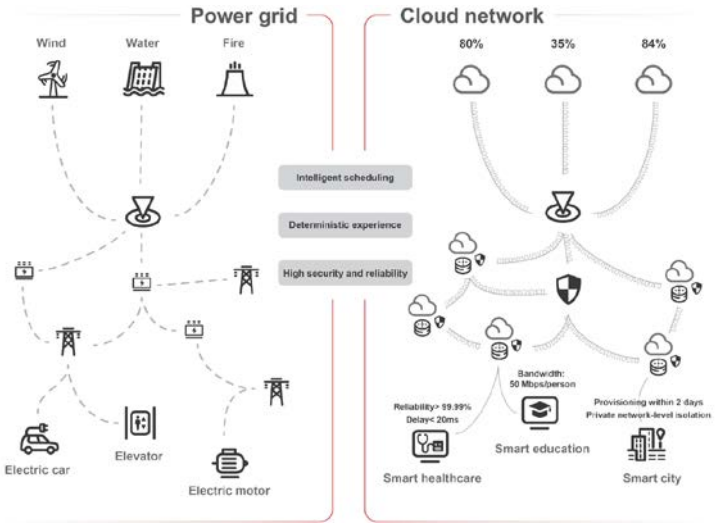


Рис. 1.1 Принцип трансформації промислових систем під впливом мережних технологій

Завдяки національним стратегіям та тенденціям цифрової трансформації все більше підприємств переносять свої послуги в хмару [1-8]. Під час заходу FutureScapes 2020 Міжнародна корпорація даних (IDC) передбачила, що 85% підприємств розгорнуть нову цифрову інфраструктуру в хмарі до 2025 року [9]. Хмарні додатки також розвиватимуться з Інтернет-додатків, таких як веб-сайти порталів та системи електронної комерції, в ключову інформацію та основні виробничі системи. Крім того, враховуючи такі фактори, як витрати на міграцію хмари, різні базові технології, що обслуговують аварійне відновлення даних та відповідають за безпеку та розгортання сервісів, одна хмара сама по собі не може задовольнити всі вимоги підприємства. Тому, багато провайдерів створюють інтегровані мережі, які дають змогу поєднувати усі переваги хмарних систем задовільняючи постійно зростаючі вимоги для кінцевих користувачів та бізнеспідприємств. До таких переваг можна віднести:

- Можливість моніторингу стану мережі в будь який час і в будь якому місці.
- Підвищення продуктивності за рахунок автоматизації та віртуалізації.
- Велика можливість для інновацій.
- Підвищення надійності і безпеки мережі.
- Програмне управління мережею.
- Покращений сервіс для користувачів.

Така інтеграція стає особливо актуальною в еру цифровізації, оскільки підприємства продовжують прискорювати свою цифрову трансформацію, а вимоги до мережі постійно змінюються. Розподіленість сервісів, сукупна взаємодія між приватними та гібридними хмарними системами спонукають провайдерів, як основних надавачів такого роду послуг, безперервно адаптуватися під швидкі зміни до потреб користувачів.

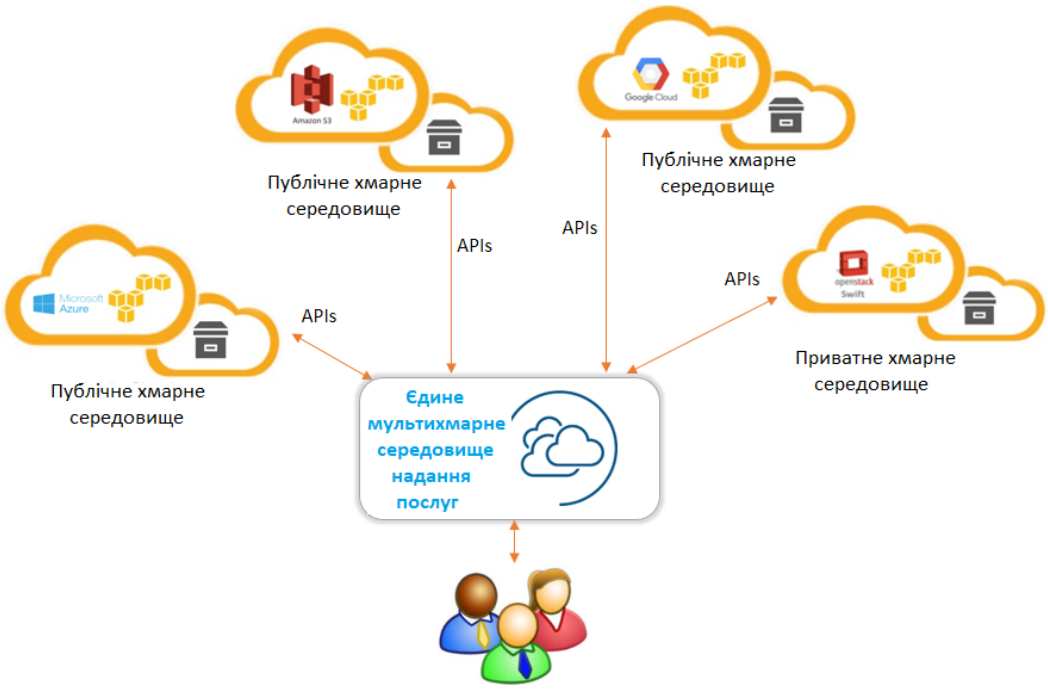


Рис. 1.2. Концепція інтеграції хмарних систем в інформаційно-комунікаційну сервісно-орієнтовану мережу провайдера

Для початку, при інтеграції хмарних систем у наявні вже інформаційно-комунікаційні сервісно-орієнтовані мережі провайдерам слід враховувати, що підприємства переважно використовують приватні хмари (рис.1.2), а автоматизація сервісів впроваджується на етапі надання користувачам онлайн-підписки. При цьому, основними вимогами є інтеграція та підключення наявної хмарної системи із забезпеченням такої ж гнучкість та швидкості надання сервісів. Для такого випадку, провайдер повинен забезпечити постійно діючий високошвидкісний канал передавання, який фактично буде активним протягом 24 годин на добу 7 днів на тиждень. Однак надання такого каналу передавання потребує від провайдера високоефективного планування та перерозподілу своїх потоків даних. З іншого боку, слід пам’ятати, що для відкриття фактично наскрізних каналів для передавання одному користувачеві, необхідно забезпечувати вимоги корпоративності надання послуг. Налаштування і конфігурація системи управління потоками трафіку потребуватиме великої кількості програмно-апаратних ресурсів провайдера та часу (тижнів або навіть місяців) для відлагодження роботи мережі. Таким чином, оператори повинні

оновити наявні інформаційно-комунікаційні сервісно-орієнтовані мережі до інтелектуальних хмарних мереж з інтегрованим плануванням системи управління потоками трафіку та постійно підвищувати ефективність їх функціонування.

З іншого боку, завдяки використанню технології віртуалізації провайдери мають можливість досягти компромісу між інтеграцією хмарних систем та адаптацією наявної інфраструктури під нові вимоги. Для цього слід належним чином контролювати роботу центрів обробки даних. Центри обробки даних з підтримкою функцій віртуалізації використовують спрощену архітектуру системи для передавання трафіку між будь-якими двома хостами, що призводить до надання сервісів з низькою затримкою, неблокуючим зв'язком і гарантованими параметрами QoS (Quality of service - якість обслуговування) відповідно до угоди про рівень обслуговування (SLA - Service-Level Agreement).

Водночас, інформаційно-комунікаційні сервісно-орієнтовані мережі із інтеграцією хмарних систем перетворилися на важливий комерційний актив та користуються великим попитом. OTT провайдери (OTT - Over-the-top) поступово переходять від постійного удосконалення роботи центрів обробки даних до хмарних глобальних мереж, маючи за мету побудувати інтегровану систему мереж, додатків і хмар. Такі постачальники послуг вважають за краще будувати власні хмарні магістральні мережі та поступово переміщувати точки присутності (POP - Point of presence) якомога ближче до користувачів або ж делегувати ці функції невеликим провайдерам. У зв'язку з цим провайдери все частіше використовують технологію віртуалізації мережі для побудови хмарної глобальної мережі розгортаючи програмні аналоги не лише програмно-апаратного забезпечення, а й каналів для передавання. У той же час, зосередившись на вимогах користувачів, POP постачальники реалізують сервіси OTT, надаючи хмарним глобальним мережам наступні функції:

- Надання послуг у режимі реального часу: послуги можна надати на хмарній платформі за лічені хвилини після оплати орендарем, а мережу орендаря можна швидко переналаштувати для підтримки міжсистемних з'єднань.
- Плата за використання: підприємства платять за фактичний трафік, який вони використовують, що дозволяє їм гнучко та ефективно використовувати пропускну здатність каналів передавання, а потім розподіляти або коригувати пропускну здатність між сайтами чи надавати її в оренду в режимі реального часу.

Порівняння можливостей інформаційно-комунікаційних мереж OTT і приватних операторів при інтеграції хмарних систем показано на рис.1.3.

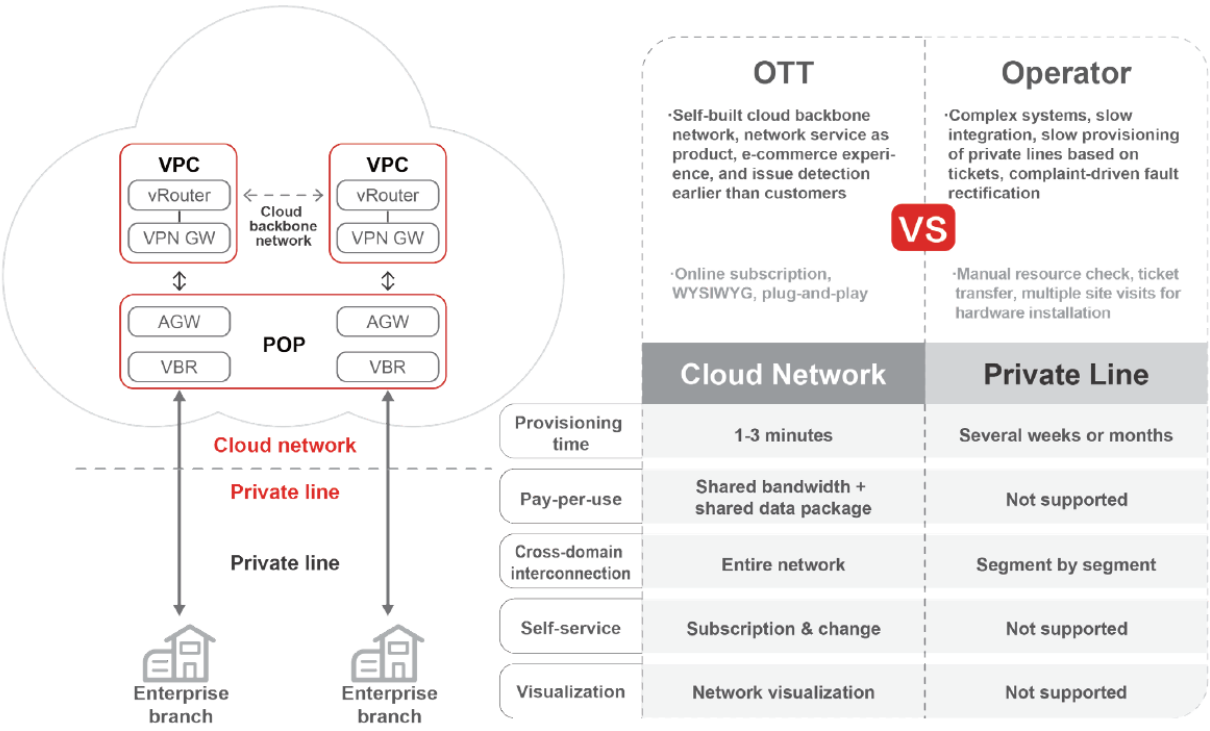


Рис. 1.3. Порівняння можливостей інформаційно-комунікаційних мереж OTT і приватних операторів при інтеграції хмарних систем

Використовуючи технологію SD-WAN (Software-defined networking in a wide area network - програмно-конфігурована глобальна мережа) OTT-провайдери мають можливість перенаправляти трафік на POP в безпосередній близькості до користувачів. Орендарям потрібно лише придбати необхідну

пропускну здатність для гнучкого міжрегіонального з'єднання. Зрештою, трафік буде поступово перенаправлятися від магістральних мереж OTT провайдерів, на рівень функціонування POP провайдерів, що породить проблему підтримки наскрізного передавання, пов'язаної із обмеженнями «останньої милі».

Таким чином високий ступінь віртуалізації для реалізації вище описаних сценаріїв надання послуг OTT провайдерами вимагає високої пропускної здатності та детермінованої затримки на рівні POP провайдерів, тоді як для роботи основних систем мережі POP провайдера потрібна наднизька затримка. Особливо важливим це є у випадку, де віртуалізація використовується для забезпечення чітко визначених обчислювальних ресурсів для користувачів. Важливе значення при цьому відіграє ступінь віртуалізації, тобто число розгорнутих і функціонуючих машин. Достатньо високий ступінь віртуалізації, і відповідно велика кількість запитів, які обслуговують віртуальні машини, може призвести до перевантаження мережі в цілому. Наприклад, підтримка диференційного захисту електромережі загального користування, яка здійснює постачання електроенергії в приватні будинки, вимагає затримки менше 2 мс.

У порівнянні з OTT-провайдерами, POP оператори стикаються з більшими проблемами щодо оновлення та синхронізації у роботі інтегрованої інформаційно-комнікаційної мережі. По-перше, мережі операторів тепер ще більші та складніші на мережевому рівні. Щоб забезпечити орендарів віртуальними мережами з гарантованою якістю, необхідно вирішити безліч технічних проблем. По-друге, традиційне управління мережною інфраструктурою в кінці-кінців зводиться до роботи саме на мережевому рівні TCP стеку, причому багато процесів потребують втручання людини. Таким чином, стає складно розгорнути ефективні системи моніторингу параметрів переданого трафіку та досягти автоматизації при внесенні конфігураційних змін. Не менш важливим є і факт, що екосистема функціонування хмарної системи в комплексі єдиної сервісно-орієнтованої мережі стикається з

проблемами, пов’язаними з надмірною кількістю постачальників BSS (Business Support System – прикладного програмного забезпечення внутрішніх бізнес-процесів), OSS (Operation Support System – прикладного програмного забезпечення внутрішніх бізнес-процесів провайдерів послуг) та контролерів, відсутністю стандартів і специфікацій інтеграції та налаштуванням інтерфейсу між системами, що призводить до складної інтеграції системи та повільного розгортання послуг.

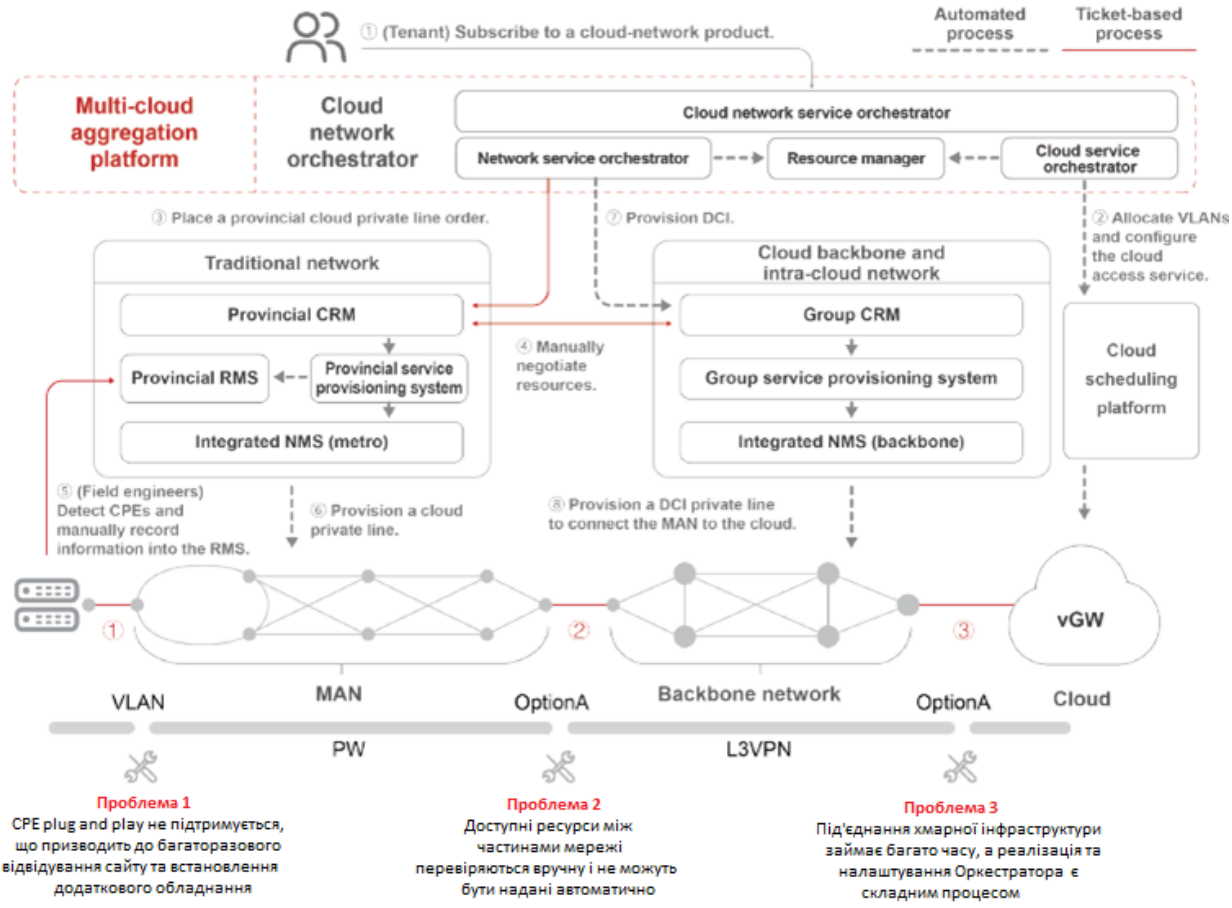


Рис. 1.4. Проблеми, пов’язані із інтеграцію хмарних систем в інформаційно-комунікаційні мережі

Як результат виникає ряд проблем пов’язаних із інтеграцію хмарних систем в інформаційно-комунікаційні мережі провайдерів (рис.1.4):

Проблема 1: конкатенація VPN з’єднань між хмарною системою та мережею провайдера. Система управління ресурсами в реальному часі (RMS) стає недоступною до моменту реконфігурації всієї мережі, а дані про CPE

(customer premises equipment) або елементи приватної хмари записуються вручну. З точки зору користувача - для встановлення з’єднання із тим чи іншим сервісом доводиться здійснювати часте оновлення сайту, через який він доступний.

Наразі система активації сервісу включає три основні компоненти: управління ресурсами, надання послуг і управління мережею. Компонент надання послуг спочатку викликає інтерфейси RMS (Record Management System - система управління записами), щоб провести перетворення 9-рівневої адреси у ідентифікатори NE (Node equipment – вузли мережі), а потім викликає інтерфейси NBI (Northbound interface - програмний інтерфейс, за допомогою якого програма представляє низькорівневі деталі додатку, серверний інтерфейс) системи керування мережею NMS (Network Manahement system) для надання послуг. NMS відповідає за VPN (Virtual Private Network — віртуальна приватна мережа) і конфігурацію тунелю для NE. Реалізація, розгортання та налаштування компонентів системи активації сервісів при інтеграції хмарних систем в мережі провайдерів призводять до складного процесу оцінки ефективності їх функціонування, стійкості роботи мережі та налаштування ефективного розподілу запитів на надання сервісів. Після встановлення CPE їм потрібно вручну ввести інформацію CPE в RMS, щоб активувати послуги. Після активації повинен знову здійснити реконфігурацію всієї мережі, щоб перевірити надання послуг. Будь-яка помилка в інформації про CPE призведе до помилки активації служби і як наслідок некоректної роботи усієї мережі. Міждоменна конкатенація сервісних VPN показано на рис.1.5.

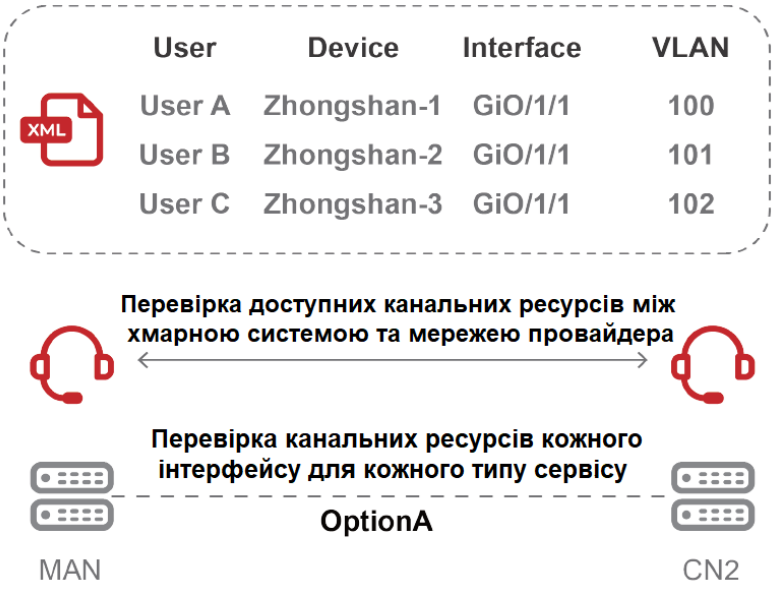


Рис. 1.5. Міждоменна конкатенація сервісних VPN

Враховуючи вищезазначені проблеми, оновлення всієї системи до можливості встановлення VPN з'єднань та надання сервісів уже існуючим користувачам є складним завданням. Автоматичне виявлення онлайнових CPE вимагає використання міжмашинних інтерфейсів між RMS та NMS. Однак RMS є автономною системою в традиційному процесі, і її дорого реконфігурувати як онлайн-систему. Якщо RMS не може бути реконфігурованою, то автоматична реконфігурація інтегрованої мережі не можлива.

Проблема 2: надання наскрізних тунелів для передавання інформації користувачам cloud системи. Послуги VPN об'єднані як в середині мережі провайдера так і в хмарній системі. При їх інтеграції надання наскрізних тунелів для передавання інформації користувачам такої мережі є досить дорогим, а ресурси координуються вручну, що робить автоматизацію надання таких каналів досить складною задачею.

Оскільки оператори традиційно будують і керують мережами за принципом міждоменної взаємодії, то надання послуг між окремими сегментами вимагає конкатенації кількох сегментів VPN. З одного боку все

відбувається відповідно до правил налаштування комунікації між сегментами мережі, однак при цьому виникає ряд супутніх завдань:

- З'єднання мережевих служб: на пограничних пристроях існує велика кількість конфігурацій VPN на рівні клієнта, і їх важко підтримувати.
- Складна автоматизація виділення VPN тунелів: системи керування вдаються до ручної перевірки канальних ресурсів через велику кількість віртуалізованих елементів хмарної системи та неможливість отримати інформацію міждоменного зв'язку, що надзвичайно ускладнює планування розподілу ресурсів для кожної служби та інтерфейсу.

Отже, надання послуг при інтеграції хмарних систем та мереж POP провайдерів із виділенням наскрізних тунелів для передавання інформації із реалізацією міждоменого управління зводиться до фактично ручного налаштування конкатенації каналів, і для цього може знадобитися проміжок часу до двох місяців (в залежності від величини мережі).

Проблема 3: синхронізація та управління інтегрованою мережею займає багато часу, а інтеграція оркестраторів хмарних систем у єдину систему управління є доволі складним процесом.

Деякі оператори зосереджуються на мультихмарній інтеграції, створюючи хмарні магістральні мережі та впроваджуючи багатохмарну агрегаційну платформу, щоб надавати клієнтам послуги взаємодії VPC між хмарними провайдерами та провайдерами POP. Оскільки хмарні магістральні мережі та мережі центрів обробки даних використовують режим конкатенації VPN, платформа агрегації кількох хмар має володіти можливостями оркестрування хмарної мережі, тоді як фізичні інтерфейси та ресурси VLAN повинні бути заплановані для взаємозв'язку хмари та мережі заздалегідь. Коли орендарі подають заявки на VPC та наскрізні тунелі для доступу до хмари онлайн, платформа агрегації кількох хмар автоматично виділяє неактивні фізичні інтерфейси та ресурси VLAN.

На сьогоднішній день інтеграція «хмара - існуюча мережа» є складним процесом, і інтеграція оркестраторів хмарних систем у єдину систему управління додає ще більше проблем. Одна з головних причин пов'язана з інтеграцією оркестраторів хмарної системи з NMS через інтерфейси на рівні мережі, які відкривають усі технічні деталі мережі. Оскільки інтерфейси мережевого рівня містять десятки тисяч параметрів, для системної інтеграції необхідна обробка великого обсягу навантаження та постійне адміністрування ресурсів. Саме тому, системна інтеграція займає в середньому шість місяців. З іншого боку, хмарна платформа надає лише десятки параметрів API VPC (Application Programming Interface Virtual Privat Cloud — це сервісно-орієнтовані інтерфейси, які захищають технічні деталі, забезпечуючи швидку комунікацію між платформою агрегації кількох хмар і мережею провайдера). Щоб досягти узгодженого функціонування хмари із мережею, необхідно прискорити їх інтеграцію, провайдер повинен забезпечити перехід до сервісно-орієнтованого інтерфейсу, що у свою чергу теж створює певні труднощі.

Беручи до уваги описану вище проблематику пов'язану із інтеграцією хмарних систем та мереж провайдерів слід детально зважити всі фактори. З одного боку – провайдер має можливість розширити набір послуг, які надає користувачам, з іншого – зростає потік трафіку, який передається через таку мережу. Відповідно, необхідні не лише нові підходи по удосконаленню, зберіганню та управлінню, а й до аналізу трафіку і, як наслідок обробки службових даних, оскільки її кількість збільшується в геометричній прогресії. Крім того, надмірне використання віртуальних серверів може ще більше ускладнити задачу, оскільки потребуватимуть залучення додаткових канальних ресурсів. Крім того, управління такою інтегрованою мережевою інфраструктурою зводиться до проблеми підвищення захищеності даних від несанкціонованого доступу до них. Такі системи захисту повинні в дуже короткий час відслідковувати всі зміни, які відбуваються із системою - атаки, технічні збої, при яких сервіс перестає нормально функціонувати.

1.2. Ефективне функціонування хмарних систем як один із методів реалізації сучасних інформаційно-комунікаційних мереж

Сучасний стан цифровізації суспільства та широке впровадження інформаційних технологій спрямований на розроблення та впровадження нових підходів до організації обчислювального процесу мереж провайдерів. Особливо актуальним таке завдання є у випадку інтеграції хмарних систем та створення єдиної інформаційно-комунікаційної сервісно-орієнтованої мережі із достатньо високим ступенем віртуалізації. Оцінка ефективності хмарних центрів є важливим напрямком досліджень, проте вона ускладнюється динамічним характером хмарного середовища та різноманітністю запитів користувачів.

В основному хмарний центр моделюється як класична відкрита мережа з одним входом, в якій розподіл часу відгуку набуває вигляду експоненти і приймає значення між часом прибуття і часом обслуговування запиту. Науковці [10] дослідили час відгуку при різних значеннях метрик, таких як накладні витрати на придбання та реалізацію віртуальних ресурсів та ін. Для зменшення часу відгуку вони розробили і впровадили портативну, розширюючу і просту в роботі платформу для генерування та відправки тестових навантажень до обчислювальних хмар. В роботі [11] хмарний центр був змодельований як система з чергою, типу $M/M/m/m+r$ (де $m+r$ - розмір буферу m з максимальною кількістю запитів **r**, які одночасно знаходяться в системі: на обслуговуванні і в черзі), в якій вже є змодельований розподіл часу відгуку. Припускається, що розподіл між надходженням і часом обслуговування - експоненціальний. Час відгуку розбивається на час очікування, час обслуговування та період виконання, за умови, що всі три періоди є незалежними (що є нереальним за словами авторів). Ієрархічний підхід до моделювання системи для оцінки продуктивності функціонування в хмарному середовищі був запропонований у роботі [12]. Суть підходу полягала у застосуванні відомих моделей - класичної формули Ерланга для оцінки втрат та системи масового обслуговування типу

M/M/m/K, як базис для моделювання системи - для оцінки пропускної здатності вихідного каналу та моделювання часу відгуку.

В роботах [13-18] науковці запропонували загальні аналітичні моделі, які базуються на підході до аналізу продуктивності хмарних сервісів. Запропонований у роботі [13] підхід зменшує складність аналізу продуктивності хмарних центрів шляхом поділу загальної моделі на простіші Марківські підмоделі, та отримання загального результату з ітерації над окремими підмоделями. Проте, запропонований авторами підхід поділу на підмоделі та вимоги, які встановлюються при проведенні досліджень, є не цілком виправдані. Часи прибуття запитів у кожній із підмоделей підпорядковуються строго експоненційному закону розподілу, а система в цілому характеризується не високим ступенем віртуалізації (в даному дослідженні кількість віртуальних машин в одній фізичній машині менша 10), що в реальних умовах надання хмарних сервісів є досить обмеженим.

У роботі [19] запропоновано математичну модель оцінки продуктивності живої міграції віртуальних машин. Ця модель показує, що ефективна оперативна міграція віртуальної машини може зменшити ймовірність помилки при наданні композитних сервісів та середню загальну затримку. Автори в [20] отримали розподіл часу відповіді за допомогою класичної відкритої мережі масового обслуговування, припускаючи, що час між надходженням і обслуговуванням є експоненційно розподіленим і фіксованим. Використовуючи отриманий розподіл, вони продемонстрували, що аналітична модель може бути використана для виявлення зв’язку між кількістю необхідних обчислювальних серверів, максимальною кількістю запитів і найвищим рівнем обслуговування. У роботі [21] досліджувалася продуктивність моделі масового обслуговування M/M/m для опису синтетичного методу оптимізації продуктивності послуг у хмарних обчисленнях. Результат моделювання показує, що запропонована модель для оцінки черги може зменшити час очікування, довжину черги, які використовують функцію синтетичної оптимізації, коли кількість серверів

збільшується. У роботі [22] запропоновано модель для кращої кількісної оцінки продуктивності великомасштабних віртуалізованих центрів обробки даних IaaS. Аналітичні результати показують вплив характеристик системи та навантаження клієнтів на два показники продуктивності: середній час відповіді та ймовірність відхилення завдання. Щодо авторів роботи [23], вони розробили систему оптимізації та аналізу витрат за допомогою стохастичних моделей продуктивності доступності хмари IaaS. Їхня модель враховує змінні вимоги до обслуговування, а саме простої та частоти відхилень, а потім розраховує тип і кількість вузлів, які знадобляться, щоб задовольнити ці вимоги з найменшими експлуатаційними витратами.

Автори роботи [24] розробили аналітичні моделі для оцінки ефективності інтеграції хмари IaaS, де запити користувачів обробляються віртуальними машинами. За основу взято аналіз ефективності реплікації віртуальних машин з точки зору продуктивності часу виконання запиту та споживання енергії. Вони запровадили підхід до моделювання енергоспоживання, який базується на вимірюванні споживання електроенергії при різних режимах роботи (холодний, теплий і гарячий) і пов’язане з цим споживання енергії. Запропонована модель була перевірена за допомогою симулятора CloudSim. Чисельні приклади показують ефективність схем реплікації віртуальних машин щодо споживання енергії та часу виконання роботи. У роботі [25] побудована комплексна модель масового обслуговування для вимірювання продуктивності гетерогенних центрів обробки даних (ЦОД). Автори проаналізували різні параметри QoS. Отримані результати демонструють точність запропонованої моделі та її зручність для аналізу неоднорідних ЦОД. У роботі [26] розроблено новий підхід, заснований на моделях черги для ефективної живої міграції кількох віртуальних машин. Автори моделювали надходження запиту за допомогою системи масового обслуговування типу *M/M/C/C,* щоб оцінити швидкість блокування запитів клієнтів. Аналогічно змоделювано запити на прибуття, використовуючи чергу *M/M/C*, щоб оцінити середню довжину черги

очікування, середню довжину черги, середній час очікування та середній час перебування. Запропонована модель оцінюється шляхом проведення математичного аналізу. Чисельні результати показують, що запропонований підхід перевершує існуючі підходи.

Автори в роботі [27] запропонували прості підходи продуктивності для оцінки часу відгуку програм, що виконуються в хмарі SaaS. Розгорнуто запропоновані моделі на реальній віртуалізованій платформі. Результати експериментів показують, що моделі дають змогу точно прогнозувати час відгуку SaaS-додатків. У роботі [28] представлено комплексне моделювання центру обробки даних, стійкого до катастроф (DTDC) з використанням стохастичних мереж винагород (SRN). Отримані результати показують прогрес доступності DTDC та реакцій системи, що відповідають вибраним змінним (аналіз вартості простою та аналізу чутливості). У [29] автори запропонували модель черги, яка може бути використана для забезпечення належної еластичності для хмарних клієнтів. Запропонована модель передбачає необхідну кількість віртуальних машин, щоб задовольнити попередньо визначену вимогу до продуктивності за угодою про рівень обслуговування.

Проте жоден із проаналізованих підходів до оцінки ефективності функціонування хмарних систем не враховує їх ступінь віртуалізації. Все вищевикладене визначає актуальність дисертаційного дослідження та його мету – розробка моделі оцінки ефективності функціонування хмарного центру з високим ступенем віртуалізації та в умовах групового надходження запитів від користувачів. У всіх проаналізованих наукових працях час обслуговування підпорядковується експоненційному закону розподілу. Крім того ступінь віртуалізації не врахований належним чином або ігнорується взагалі [30]. Для вирішення цієї проблеми необхідно запропонувати модель для оцінки ефективності хмарних центрів в умовах високого ступеню віртуалізації, яка, на відмінну від існуючих, враховує можливість обслуговування при груповому надходженні запитів та розподілений час обслуговування запитів.

1.3. Кластеризація як показник підвищення продуктивності складних сервісно-орієнтованих мереж

Хмарні обчислення забезпечують середовище для виконання та надання послуг. Вони поєднують між собою набір технологій доступу та систем, які покликані забезпечити необхідні сервіси кінцевим користувачам [31]. Все це стає можливим за допомогою технології віртуалізації. Як відомо існують три основних моделі надання послуг у хмарному середовищі – це Інфраструктура в якості сервісу (IaaS), Платформа в якості сервісу (PaaS) та Програмне забезпечення в якості сервісу (SaaS). Саме ефективне функціонування цих моделей забезпечує обробку та виконання великих масивів даних.

Підвищення рівня апаратної потужності, пришвидшення обробки і аналізу службової інформації, ефективне використання пропускної здатності каналів зв'язку вимагають від інженерів розробки нових методів та алгоритмів, які б задовольняли вимоги користувачів. Здавалося б хмарні обчислення, мають на меті задовольнити доступ до великої кількості обчислювальних потужностей, адже функціонують в умовах агрегації ресурсів та надання єдиного системного підходу по їх управлінню. Особливо актуальним є їх інтеграція у мережі провайдерів, коли хмарні системи стають складовими інформаційно-комунікаційних мереж. З одного боку – провайдер має можливість розширити набір послуг, які надає користувачам, з іншого – зростає потік трафіку, який передається через таку мережу. Відповідно, необхідні не лише нові підходи по удосконаленню, зберіганню та управлінню, а й до аналізу трафіку і, як наслідок обробки службових даних. Крім того, надмірне використання віртуальних серверів може ще більше ускладнити задачу, оскільки потребуватимуть залучення додаткових канальних ресурсів на рівнях комунікацій адміністратор додатків - сервер обробки – сервер зберігання даних. Особливо гостро це питання постає у випадках групового надходження запитів або обробки потоків типу Big Data, адже провайдерам так чи інакше необхідно підтримувати необхідний рівень QoS та умови SLA.

Важливе місце у вирішенні такого роду проблем та підтримки належного рівня якості послуг займає архітектура побудови таких мереж. Динамічність розгортання як фізичних так і віртуальних машин, породжують проблему доступності та відмовостійкості обчислювальних ресурсів для обслуговування тих чи інших потоків запитів. Як наслідок, деякі із щойно розгорнутих машин будуть одразу перевантаженні не тільки запитами, які обслуговуються в межах одного групового потоку, а й запитами інших потоків. Вирішенням цієї проблеми може стати кластеризація.

Вирішенню проблем ефективного управління обробкою групових потоків даних займається багато вчених. Ключовим аспектом в їх працях є надійне та ефективне розміщення даних, а також якомога швидша їх обробка. Так наприклад, автори статті [32] розробили механізм для функціонування розподіленої та паралельної обробки даних, в якій однорідним даним надається доступ (по певному алгоритму) до існуючих великомасштабних сховищ даних. Це забезпечить підвищення швидкості їх обробки, однак не враховуватиметься неоднорідність групових потоків реального часу. Автори публікації [33] запропонували підвищити надійність даних при їх аналізі та обробці використовуючи додаткове кодування. Однак, не враховувалося затратність ресурсів системи на проведення додаткового етапу кодування.

Якщо говорити про ефективність використання ресурсів дата центрів на обробку даних, то в цій області теж існує багато досліджень. Автори J. Cohen, B. Dolan [34] запропонували своє бачення гнучкого та глибокого аналізу даних використовуючи паралелізм системи баз даних Greenplum. Для вирішення проблеми високої складності планування задач обробки даних з використанням процесу MapReduce автори [35] запропонували удосконалення планування всіх трьох фаз процесу MapReduce, що дало змогу підвищити точність планування так зменшити енергоефективність системи.

Поняття відмовостійкості в хмарі пов’язане з механізмами, необхідними для того, щоб техніка витримувала збої при виконанні та обслуговуванні

запитів, які залишаються в системі. Однією з переваг розробки методів відмовостійкості в хмарних обчисленнях є уникнення збоїв, відновлення працездатності та підвищення показників продуктивності. На сьогодні механізми відмовостійкості тісно взаємопов'язані та узгоджуються з плануванням роботи та ефективним розгортання хмарних систем, тобто кластеризацією їх компонентів [36, 37 Fault tolerance aware scheduling technique for cloud computing environment using dynamic clustering algorithm]. Методи динамічного планування та розгортання хмарних систем призначені для створення незалежного пулу обчислювальних ресурсів відповідного розміру або кластерів обчислювальних ресурсів, які б враховували специфіку виконання запитів, їх інтенсивність поступлення та складність, а також задовольняли параметри QoS цілого потоку запитів [38–40]. Автори праць [38,41–44] застосовують розбиття запитів по кластерах відповідно до наперед визначених категорій із наданням для їх обслуговування обмежених обчислювальних ресурсів хмарного середовища. Гангешварі та ін. [45] представляють динамічну гіперкубічну однорангову хмару (HPCLOUD), яка являє собою структурований одноранговий фреймворк, реалізований в системі хмарних обчислень, у якому для планування завдань використовується принцип кластеризації схожий до MapReduce. Результати експерименту показують, що запропонований метод продемонстрував кращу ефективність з точки зору часу відповіді та дозволив збільшити кількість опрацьованих файлів. Однак динамічність розгортання середовища HPCLOUD враховується лише на початковому рівні планування, що не дозволяє проводити автоматичну реконфігурацію системи при зміні розміру завдань в межах кожного окремого кластеру. Крім того, час відгуку враховується при вимірюванні параметра відмовостійкості, тоді як інші параметри системи не враховуються.

Тауфік та ін. [46] запропонували метод динамічного планування процесу обслуговування запитів в хмарі з використанням інтелектуальної кластеризації, що базується на оптимізації колоній (ACO). Для цього підходу прийнято, що

управління вхідними запитами, які надходять в систему, відбувається шляхом знаходження оптимізаційної функції обчислювальних ресурсів, які виділяються для їх обслуговування. На основі значень цієї функції відбувається кластеризація обчислювальних ресурсів під фіксований набір запитів Експериментальні результати показують, що запропонований метод ACO перевершував алгоритми FCFS (First-Come, First-Served – перший прийшов, перший обслужено) та RR (Round Robin). Однак динамічна зміна структури мережі та розміру запитів в запропонованому методі не є пріоритетом для досягнення відмовостійкості системи. Тому через постійні зміни в структурі гетерогенних хмарних систем існує потреба в розробці методу кластеризації вузлів хмарних систем, який би ефективно функціонував в умовах обслуговування групових потоків запитів, та дозволив підвищити ефективність функціонування хмарних систем, в яких одночасне обслуговування групових потоків запитів забезпечує велика кількість віртуальних машин. Такий підхід повинен базуватися на групуванні в кластер пулу віртуальних машин, які будуть використовуватися під обслуговування чітко визначеного групового потоку із врахуванням географічної близькості вузлів між собою та моніторингу доступних програмно-апаратних та телекомунікаційних ресурсів, що дасть змогу підвищити відмовостійкість системи.

1.4. Підходи щодо забезпечення політики безпеки веб-сервісів у програмно-конфігурованих сервісно-орієнтованих мережах

Зі зростанням залежності від технологій стає все більш важливим захистити кожен аспект інформації та даних. Оскільки комп’ютерні мережі стають більшими, цілісність даних стала одним з найважливіших аспектів, які організаціям та провайдерам слід враховувати. Хоча не існує мережі, яка була б захищеною від атак, стабільна та ефективна робота системи безпеки мережі є важливою для захисту даних клієнта та допомагає мінімізувати ризик втрати конфіденційної інформації. Для цього необхідні ефективні системи моніторингу стану кожної системи, яка буде аналізувати можливі проблеми, що

можуть виникати під час роботи, і запускати системи захисту в самих сервісах або повідомляти провайдерів чи системних адміністраторів про можливу загрозу [47]. Одним із можливих варіантів розвитку таких систем є самодіагностика сервісу, коли аналізатор сам перевіряє підключені системи на вразливості в системі захисту.

Згідно аналітики компанії Cisco [48], проблема безпеки і захисту від можливих атак лише загострюється. Згідно зі статистичними даними, представленими в [48], до середини 2022 року кількість DDoS-атак подвоїться до 14,5 мільйонів у порівнянні із 2017 роком (рис.1.6). У зв'язку з цим, збільшується розмір і трафік DDoS-атак, що може сягати до 1,7 Тб/с, а втрати від порушення працездатності мережі становлять близько 221 836,80 доларів США. Порівнюючи 2017 і 2018 роки, кількість атак на пристрої IPS та брандмауери майже подвоїлася з 16% до 31% відповідно [48]. За цей час кількість DDoS-атак на хмарні сервіси і центрів обробки даних також зросла з 11% до 34% (отс.1.7).

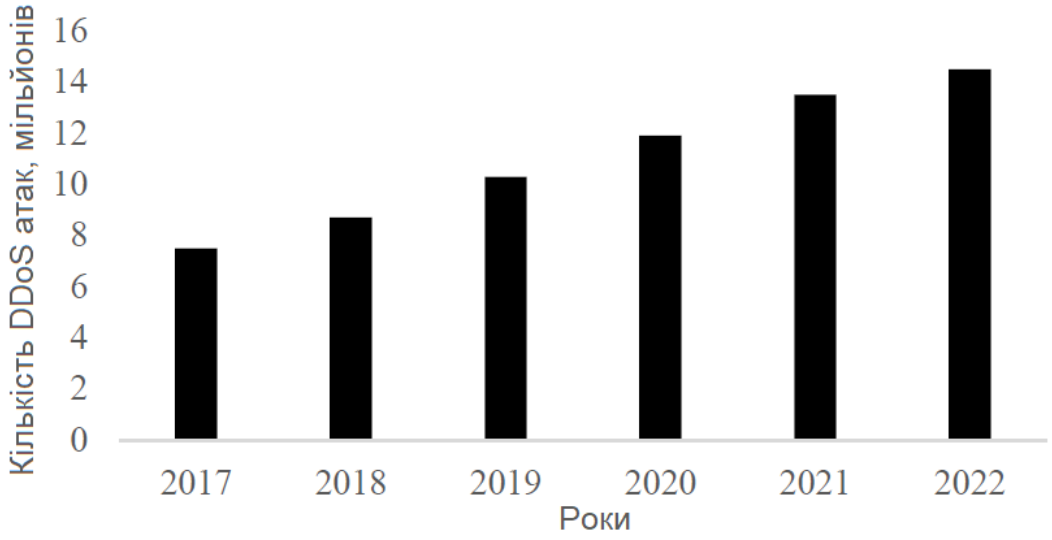


Рис. 1.6. Прогноз та аналітика компанії Cisco щодо кількості DDoS атак [48]

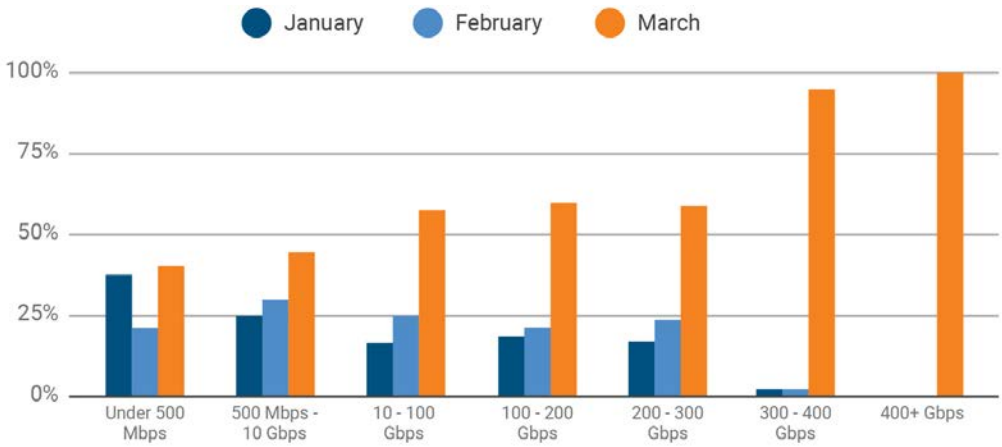


Рис. 1.7. Розподіл розмірів атак протягом першого кварталу 2021 р. [48]

Наведена аналітика підтверджує проблематику захисту мереж від атак. На сьогоднішній день не існує якогось універсального засобу для протидії атакам. Для протидії розподіленим атакам, спрямованим на відмову в обслуговуванні, слід дотримуватися двох основних завдань:

1. Діагностувати атаку на самих ранніх стадіях.

2. Друге завдання пов'язане з поділом загального потоку трафіку на шкідливий і звичайний. Зрозумівши, які з клієнтських запитів є результатом атаки, можна буде створити відповідні правила для брандмауера чи контроллера мережі.

На сьогоднішній день існує багато способів виявлення атак на відмову в обслуговуванні. Для прикладу, HADEC [49-51]– це метод для виявлення високошвидкісної DDoS-атаки, що відбувається на мережевих і прикладних рівнях, з використанням протоколів TCP-SYN, HTTP GET, UDP та ICMP. Рамка складається з двох основних компонентів: сервера виявлення та сервера захоплення. Виявлення DDoS в реальному часі починається з захоплення сервера, який відповідає за захоплення мережевого трафіку та передачу журналу на сервер виявлення для обробки. Виявлення обчислює вхідний пакет для UDP, ICMP та HTTP для виявлення атаки, якщо кількість з'єднань джерела перевищує заданий поріг. Однак такий метод виявлення потребує залучення значних як мережевих так і фінансових ресурсів. Іншим широко розповсюдженим методом виявлення атаки є MLP-GA алгоритм. Його

дослідження наведені у роботі [52]. Пропонований метод використовує чотири параметри для генерування виявлення атаки на прикладних рівнях. Метод виявлення підраховує кількість HTTP GET-запитів, отриманих веб-сервером, і обчислює кількість IP-адрес, орієнтованих на сервер протягом 20 секунд, а також перевіряє номер порту, який використовується HTTP DDoS, оскільки порти, які використовуються зловмисниками HTTP DDoS, змінюються і залишаються відкритими. Методика виявлення на основі завантаженості вихідного вузла [53]. Ця методика використовує кілька рівнів для захисту веб-сервера. Перший рівень дозволяє або відхиляє отримане з'єднання, перевіривши вихідну IP-адресу з білим списком. Зареєстрованим IP-адресам дозволено встановити з'єднання з веб-серверами для отримання сервісу, тоді як підключення незареєстрованих IP-адрес переривалося. Дозволені IP-адреси були перевірені, і якщо вони поводитимуть зловмисні дії – з'єднання буде перервано, а IP-адреси додаються у чорний список.

Використання даних із лог журналів для аналізу та моніторингу безпеки мережі та сервісів теж присвячено дуже багато наукових досліджень. Автоматизацією аналізу даних із розбиттям вмісту лог журналів на класи займалися науковці Hu Q., Tang, B., та Lin, D. Результати їх досліджень наведені у роботі [54]. Кожній події із історії журналу присвоювався свій ваговий коефіцієнт, який визначався відношенням між кількістю унікальних запитів користувача протягом години його активності у системі, що спостерігаються протягом n днів. Таким чином події класифікувалися на аномалії (коли ваговий коефецієнт мав не середньозважене значення) та звичайні.

Дослідники Max Landauer та ін **Помилка! Джерело посилання не знайдено.** пропонують виявляти аномальну поведінку на основі групування рядків журналів за подібністю, щоб встановити статичні карти кластерів. Потім рядки журналів розподіляються на існуючі карти кластерів, створені в попередніх і наступних часових вікнах. Тим самим встановлюється зв'язок між

кластерами двох сусідніх карт кластерів, які раніше не мали спільних елементів. Це дозволяє проводити обчислення метрики перекриття, яка визначає ймовірність переходу з кластеру в кластер. Однак на кожному наступному етапі роботи системи проводиться додаткова кластеризація для створення нових статичних карт.

Рішенням такого роду проблем може бути впровадження машинного навчання, особливо в рамках реалізації концепції програмно-конфігурованих дата центрів. Автори роботи [56] зосередили свої дослідження на глибокому машинному навчанні, яке працює на основі методу Restricted Boltzmann Machine (RBM). Як результат, вони представили безпечний фреймворк з SDN управлінням та IDS структурою. Дослідження проводились на основі Tensorflow з KDD99 набором вхідних даних. Запропонований алгоритм показав 94% точності. Ще один варіант IDS представлений у [57]. Дослідники моделюють захищену мережу шляхом впровадження ідентифікаторів типів атак та паралельного нейронного навчання з перехресним контекстом. Реалізацію IDS на основі машинного навчання на програмного управління проводили і в роботах [58,59], зокрема автори [59] пропонували альтернативний сценарій функціонування інтелектуальних транспортних SDN мереж.

Механізм виявлення та захисту від DDoS атак, запропонували дослідники Lin and Wang [60]. В основі їх дослідження лежить метод, який розділяє управління Openflow та sFlow для виявлення аномалії. Як результат – робота та розгортання такого захисту є досить складним завданням. Більш точний метод виявлення атак запропонований у [61]. Yang et al. запропонували метод, який базується на значенні ентропії між інформацією про потік та середнім значенням ентропії потоку. Незважаючи на те, що інформаційна ентропія є більш точнішою для виявлення аномалій, її все одно потрібно поєднувати з іншими технологіями при визначенні порогового та багатоелементного розподілу ваги. Саїд та ін. [62] дослідили, що на основі аналізу характеристик кожного протоколу TCP/UDP/ICMP метод виявлення аномального трафіку

повинен розрізняти пакетний протокол, та за допомогою навчального алгоритму ANN виявляти DDoS-атак. Проте, це є досить складно, оскільки необхідно проводити постійний аналіз всіх заголовків пакету.

З іншого боку використання статистичних підходів до виявлення аномалій, теж не є оптимальним варіантом. Автори [63] для виявлення DDoS-атак пропонують використовувати алгоритм SOM, який працює шляхом вилучення статистики потоку протягом певного виділеного часового інтервалу. Проте недоліком методу є те, що поведінка атаки виявляється не вчасно і не точно. У [64] запропоновано механізм виявлення DDoS-атак на основі аналізу аномальних характеристик вихідної IP-адреси та IP-адреси призначення. Однак метод не враховує та не коригує певний поріг значень аномальності IP адрес.

Усі проаналізовані вище дослідження базуються або на статистичних методах навчання систем захисту або потребують додаткового аналізу мережевого трафіку. Саме тому актуальним є пошук рішення щодо захисту сервісно-орієнтованих мереж та побудова інтегрованої архітектури системи інтелектуального виявлення атак, що дасть змогу відстежувати як наявні загрози, так і прогнозувати атаки на мережу.

Висновки до 1-го розділу

Проведено аналіз існуючих методів інтеграції хмарних систем у сучасні інформаційно-комунікаційні мережі. Визначено, що попри усі переваги, які підприємства і провайдери отримують при інтеграції хмарних систем, їх ефективність функціонування ускладнюється динамічним характером хмарного середовища та різноманітністю запитів користувачів. Встановдено, що під’єднання хмарної інфраструктури займає досить тривалий час та потребує відповідної кваліфікації системних адміністраторів, а реалізація та налаштування Оркестратора, для управління таким компонентом мережі є складним процесом. Ресурси, необхідні для функціонування різних частин мережі досить часто перевіряються вручну або не можуть автоматично виділятися без втручання адміністратора.

Доведено, що важливе значення при інтеграції хмарних систем відіграє ступінь віртуалізації, тобто число розгорнутих і функціонуючих машин. Достатньо високий ступінь віртуалізації, і відповідно велика кількість запитів, які обслуговують віртуальні машини, може призвести до перевантаження мережі в цілому.

Управління інтегрованою мережевою інфраструктурою зводиться до проблеми підвищення захищеності даних від несанкціонованого доступу. Для цього необхідні певні системи моніторингу стану мережі, яка буде аналізувати можливі проблеми, що можуть виникати під час роботи, і запускати системи захисту в самому сервісі або повідомляти адміністраторів про можливу загрозу.

РОЗДІЛ 2. МОДЕЛІ ТА МЕТОДИ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ФУНКЦІОНУВАННЯ ХМАРНИХ СИСТЕМ ЯК КОМПОНЕНТІВ СЕРВІСНО-ОРІЄНТОВАНИХ МЕРЕЖ

2.1. Математична модель оцінки ефективності функціонування хмарних систем в умовах групового надходження запитів

Сучасний стан розвитку інформаційних технологій спрямований на розроблення та впровадження нових підходів до організації обчислювального процесу. Оцінка ефективності хмарних центрів є важливим напрямком досліджень, проте вона ускладнюється динамічним характером хмарного середовища та різноманітністю запитів користувачів. Із аналізу нещодавніх результатів досліджень в області хмарних обчислень тільки незначна частина була присвячена оцінці ефективності.[10-18] Оцінка ефективності є особливо важливою у випадку, де віртуалізація використовується для забезпечення чітко визначених обчислювальних ресурсів для користувачів [65].

У випадку, коли ступінь віртуалізації, тобто число віртуальних машин, що одночасно працюють на одній фізичній машині, є високим в поточному стані технології, це означає більше сотні віртуальних машин, а достатньо велика кількість запитів, які вони обслуговують, може призвести до перевантаження системи в цілому.

Для вирішення цієї проблеми у дисертаційній роботі пропонується удосконалена математична модель для оцінки ефективності функціонування хмарних систем при їх інтеграції в сучасні інфокомунікаційні мережі, яка, на відмінну від існуючих, враховує можливість обслуговування під час групового надходження запитів і розподілений час обслуговування запитів.

Модель базується на двоступеневій техніці апроксимації, де основний не Марківський процес спочатку моделюється як вбудований напів марківський процес, який потім моделюється як апроксимований процес Маркова, але тільки в моменти надходження групового потоку запитів. Для побудови моделі використовується техніка побудови ланок Маркова. Спочатку моделюється не

марківська система з вбудованим напів марківським процесом, що збігається з основною системою в моменти надходження групового потоку запитів і їх виходу з системи. Для визначення ймовірності переходу до вбудованого напів марківського процесу, необхідно підрахувати кількість обслужених запитів між двома груповими потоками, які прибули один за одним.

Наступним кроком після процесу підрахунку є введення вбудованого марківського процесу, який моделює впроваджений напів марківський процес дискретним способом. Цей метод забезпечує точне обчислення важливих показників ефективності таких як середня кількість запитів в системі, довжина черги, час відгуку та час очікування, ймовірність блокування та ймовірність негайного обслуговування [65].

Вважатимемо, що хмарний центр складається з M фізичних машин, кожна з яких може розмістити m віртуальних машин, як показано на рис. 2.1. Вхідні запити направляються через сервер розподілу навантаження до одного з M фізичних серверів. Користувачі можуть здійснити запит на одну або більше віртуальних машин одночасно, тобто від одного користувача можливе групове надходження запитів, що в умовах великої кількості користувачів є типовим. Приймемо, що групове надходження запитів буде підпорядковуватися Пуасонівському закону розподілу.

Коли відбувається групове надходження запитів на сервер, сервер розподілу навантаження здійснює спробу обслужити його – тобто надати йому одну із M фізичних машин.

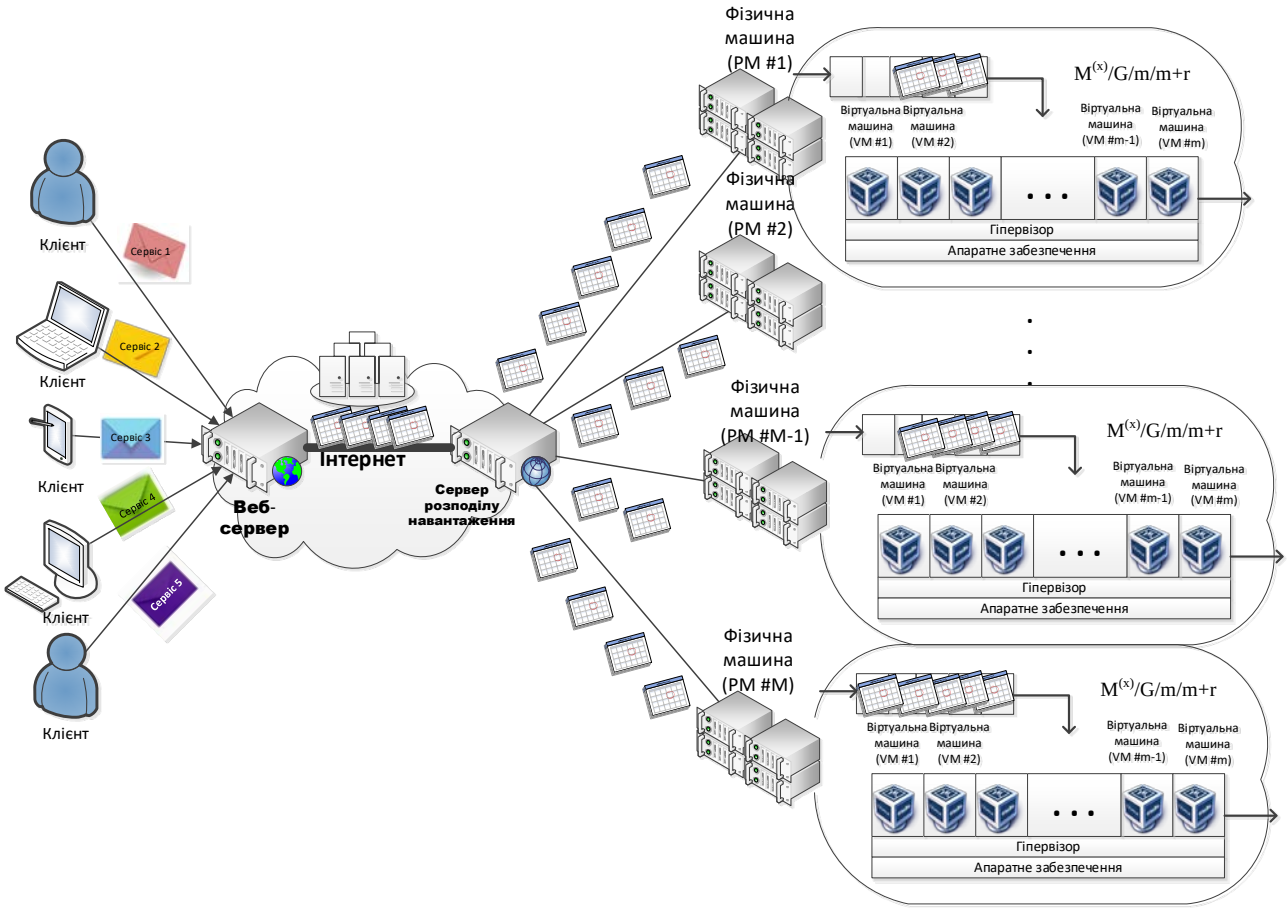


Рис.2.1. Архітектура хмарного центру

Оскільки кожна фізична машина має скінченну вхідну чергу, такі вхідні групові запити обробляються наступним чином:

- Якщо фізична машина з достатньою вільною ємністю знайдена, то групові запити розпаралелюються між доступними віртуальними машинами і виконуються негайно.
- В іншому випадку, якщо фізичну машину з достатнім розміром вхідної черги не знайдено, то запити в межах групового потоку запитів стають в чергу на виконання.
- Інакше, потік запитів відкидається.

Такий спосіб відомий як загальний спосіб ефективного прийому\відхилення користувачів, які зазвичай не приймають часткового виконання своїх запитів на обслуговування [65].

На сьогоднішній день провайдери хмарних послуг не публікують інформацію щодо дослідження таких статистичних параметрів, як час виконання сервісу, тому можна припустити, що загальний розподіл часу обслуговування запитів переважно той, що враховує коефіцієнт варіації (він визначається відношенням середньоквадратичного відхилення та середнього значення величини). Коли загальне навантаження на одну фізичну машину збільшується, то зменшується продуктивність віртуальних машин, які працюють одночасно, і фактична тривалість надання сервісу зростатиме. Запропонована у дисертаційній роботі модель враховує цю особливість.

Змоделюємо кожну фізичну машину в хмарному центрі, як систему масового обслуговування типу *M/G/m/m+r* (пуасонівським потоком вхідних запитів, довільним розподілом часу обслуговування, *m* фізичних машин та розміром буферу m з максимальною кількістю запитів **r**, які одночасно знаходяться в системі: на обслуговуванні і в черзі (тобто з урахуванням запитів, які обслуговуються віртуальними машинами). Вважатимемо, що груповий потік запитів, які прибули на обслуговування, підпорядковується Пуасонівському закону розподілу, а час прибуття описується експоненціальним законом розподілу. Вважатимемо, що потоки запитів надходять з інтенсивністю λ_j (запитів за секунду). Кумулятивна функція розподілу ймовірностей часу прибуття групового потоку запитів протягом часового інтервалу x матиме вигляд $CDF(A) = P\{A < x\}$. Враховуючи закон розподілу часу прибуття групового потоку, ймовірність того, що протягом часового інтервалу x надійдуть групові потоки запитів з інтенсивністю λ_j визначатиметься як $PDF(x) = \lambda_j e^{-\lambda x}$. Тоді час очікування i-го запиту в черзі на обслуговування із використанням перетворення Лапласа-Стілтьєса визначатиметься як:

$$A^j(S) = \int_0^\infty e^{-Sx} PDF(x)dx = \frac{\lambda_j}{\lambda_j + S} \tag{2.1}$$

Звернемо увагу, що інтенсивність поступлення запитів λ в хмарний центр визначатимуться як:

$$\lambda = \sum_{i=1}^{M} \lambda_i \tag{2.2}$$

Нехай g_k – імовірність того, що груповий потік має розмір $k = 1,2,...MRS$, де MRS - максимальна кількість запитів в потоці, $\overline{g}$ - середнє значення розміру групового потоку, а $\prod_g(z)$ - ймовірнісна твірна функція групового потоку, які визначаються відповідно::

$$\begin{aligned} \prod\nolimits_g(z) &= \sum_{k=1}^{MRS} g_k z^k \\ \overline{g} &= \prod\nolimits_g^{(1)}(1) \end{aligned} \tag{2.3}$$

Таким чином моделюється кожна фізична машина в хмарній системі, як система масового обслуговування типу M/G/m/m+r. Кожна фізична машина може запустити m віртуальних машин із розміром черги рівною m+r кількості запитам.

Припустимо, що час обслуговування запитів у межах групового запиту однаковий і рівномірно розподілений відповідно до загального розподілу. Проте, параметри загального потоку (наприклад, середнє значення, дисперсія тощо) залежать від ступеня віртуалізації фізичних машин. Вважатимемо, що зміна середнього часу обслуговування у зв’язку зі зміною робочого навантаження така ж, як у дослідженнях, проведених у статті [11]. На жаль, не можна зробити якийсь висновок щодо зміни моментів часу обслуговування, приведених у статті [12]. Наприклад, на рис. 2.2 наведені результати оцінки продуктивності засобами VMmark для сімейства фізичних машин з різною кількістю віртуальних машин. З даних на рис.2.2 можна побудувати залежність середнього часу обслуговування від кількості розгорнутих на фізичних машинах віртуальних машин, яка нормалізована за часом обслуговування, отриманого за рахунок роботи фізичних машин як окремого елементу. Слід зазначити, що середній час обслуговування об’єднує як обробку запитів, так і тривалість обслуговування запиту в груповому потоці запитів.

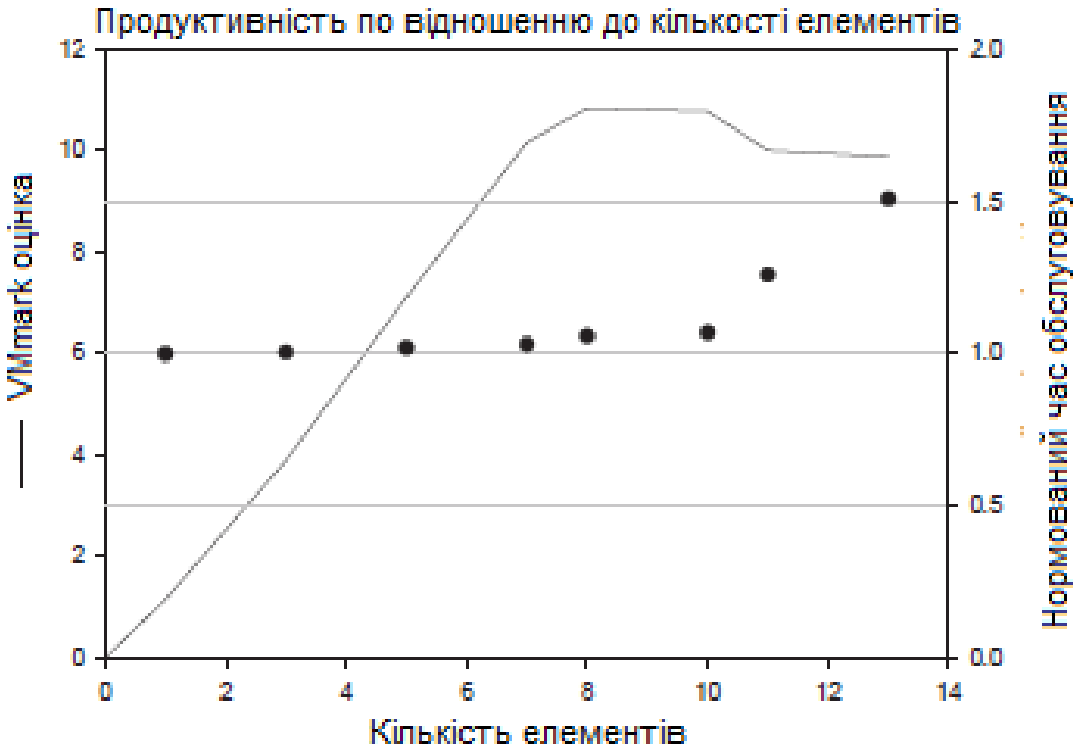


Рис.2.2. Продуктивність у порівнянні з відсутністю елементів [12]

Залежність приведена на рис.2.2 може бути апроксимованою і показувати нормалізоване значення середнього часу обслуговування $\overline{b_n}(y)$ як функцію залежну від кількості елементів та ступеня віртуалізації:

$$\overline{b_n}(y) = \frac{\bar{b}(y)}{\bar{b}(1)} = \frac{1}{0.996 - (8.159 * 10^{-6}) * y^{4.143}} \tag{2.4}$$

де y – кількість віртуальних машин на одній фізичній машині. Кумулятивна функція розподілу часу обслуговування групового потоку запитів y віртуальними машинами матиме вигляд $B_y(x) = P\{B_y(x) < x\}$, а його функція щільності $b_y(x)$ визначатиметься як $PDF(x)$. Тоді час очікування j-го запиту в черзі на обслуговування y віртуальними машинами із використанням перетворення Лапласа-Стілтьєса визначатиметься таким чином:

$$B_y^j(S) = \int_0^\infty e^{-sx} b_y(x) dx \tag{2.5}$$

Після цього час обслуговування визначатиметься як:

$$\bar{b}(y) = -B_y^j(0) \tag{2.6}$$

У випадку, якщо кожна віртуальна машина зможе одночасно обслужити максимально Ω запитів, то функція розподілу часу обслуговування j-го запиту y віртуальними машинами визначатиметься:

$$B^j(s) = \sum_{y=1}^{\Omega} p_y B_y^j(s) \tag{2.7}$$

де p_y – імовірність того, що на кожній фізичній машині розташовано y віртуальних машин. Тоді загальний середній час обслуговування дорівнює:

$$\bar{b} = \sum_{y=1}^{\Omega} p_y \bar{b}(y) \tag{2.8}$$

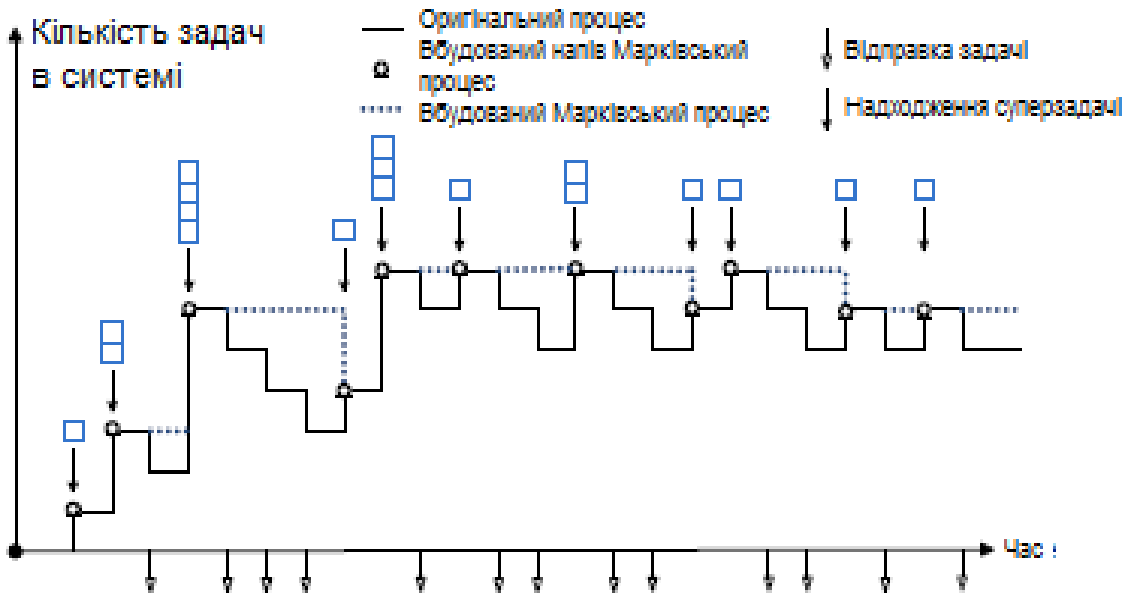


Рис. 2.3. Приклад надходження задачі від первісного процесу, вбудованого напів марківського процесу і вбудованого марківського процесу.

Розглянемо час обслуговування, як часовий інтервал протягом якого розпочинається і закінчується обслуговування запиту, що надійшов на обслуговування. Позначимо через B_+ час обслуговування запитів в довільній точці перед початком та до закінчення інтервалу обслуговування, а B_- - інтервал часу від початку обслуговування до довільної точки в середині цього

інтервалу. Вважатимемо, що вони мають однакову функцію розподілу часу обслуговування і визначатимуться:

$$B_+^j(s) = B_-^j(s) = \frac{1 - B^j(s)}{s\bar{b}} \tag{2.9}$$

Це дозволить врахувати ступінь віртуалізації при визначенні функції розподілу часу перебування запитів у кожній «ділянці» при дослідженні переходу запитів зі стану в стан (надходження запиту – обслужений запит), який описуватиметься нижче.

Система масового обслуговування типу *M/G/m/m+r* при надходженні групового потоку запитів здійснює їх паралельно-групове обслуговування. При такому способі обслуговування важливим є процес переходу запитів із стану в стан (надходження запиту – обслужений запит). Для опису таких переходів та побудови достовірної моделі оцінки ефективності функціонування хмарного центру використаємо техніку побудови ланцюгів Маркова із деякою апроксимацією.

Для початку змоделюємо не марківську систему з вкладеним напів марківським процесом, який позначимо eSMP, що збігається з основною системою в моменти надходження групових потоків запитів і їх виходу з системи. Для визначення ймовірності переходу до вкладеного напів марківського процесу, необхідно підрахувати кількість обслужених запитів між двома запитами, які прибули один за одним, оскільки процес обслуговування є паралельним (рис.2.3).

Наступним кроком після процесу підрахунку є введення вкладеного марківського процесу (позначимо як aEMP), який моделює напівмарківський процес дискретним способом (рис. 2.4). Стани нумеруються відповідно до кількості запитів, які наразі знаходяться в системі. Вважатимемо, що у aEMP розподілення вихідних запитів, котрі передаються між двома запитами, які прибули послідовно, відбувається за час надходження групового потоку.

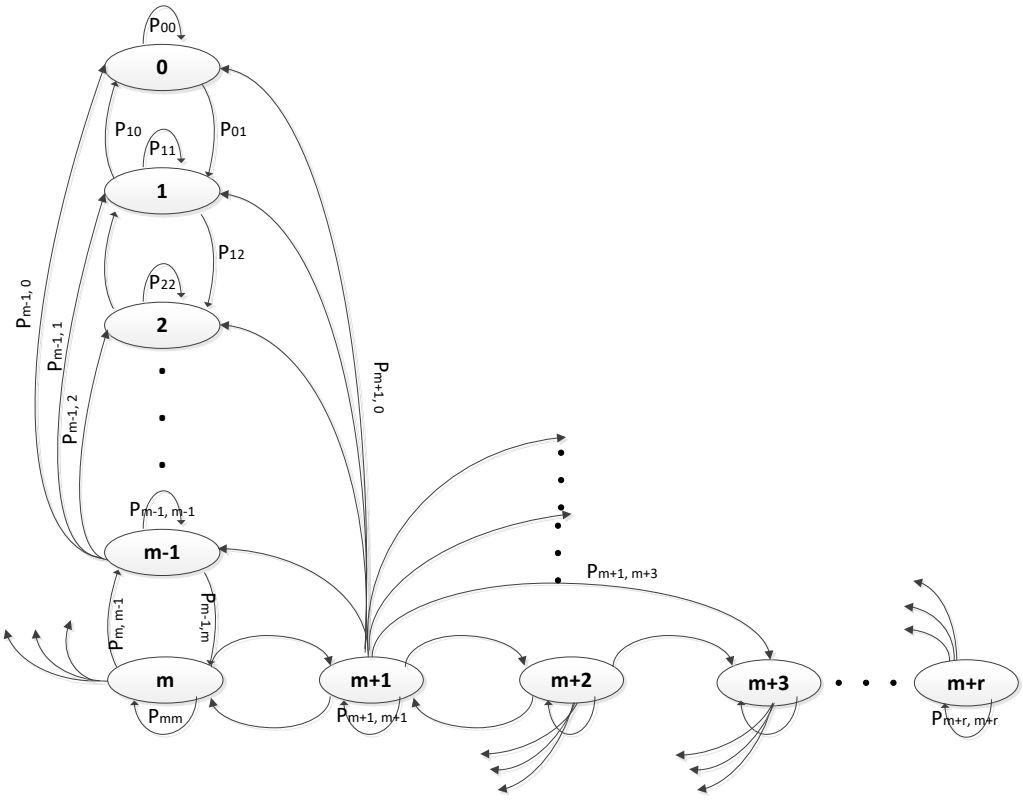


Рис.2.4. Принцип процесу переходу запитів зі стану в стан з використанням ланцюгів Маркова

Нехай маємо протікання трьох процесів: початковий, вкладений напів марківський та один апроксимований вкладений процес. Пронумеруємо ділянки залежно від кількості запитів, що надходять у систему в даний момент часу (які включають в себе і ті, які обслуговуються і ті, які очікують).

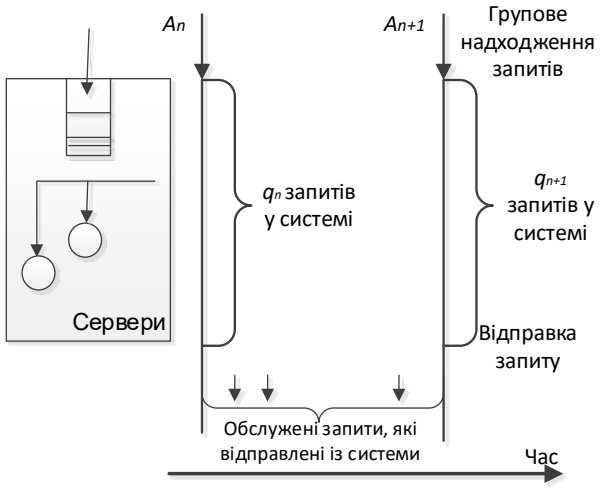


Рис.2.5. Точки спостереження процесів залежно від кількості запитів, що надходять у систему в даний момент часу

Нехай A_n та A_{n+1} вказують відповідно моменти n та $(n+1)$ надходження групового потоку запитів в систему, а q_n та q_{n+1} вказує на кількість запитів, які знайдені в системі перед тим, як прибули наступні, що схематично показано на рис. 2.5. Якщо $k-$ кількість запитів в груповому потоці, а V_{n+1} вказує на кількість запитів, які обслужені системою між A_n та A_{n+1}, то $q_{n+1} = q_n - V_{n+1} + k$.

Для знаходження вкладеного процесу переходу запитів зі стану в стан з використанням ланцюгів Маркова (позначимо його як аЕМС) необхідно розрахувати відпові дні ймовірності переходів, які можуть визначатися як:

$$P(i, j, k) \equiv P[q_{n+1} = j \mid q_n = i \text{ та } X_g = k] \qquad (2.10)$$

Іншими словами, нам необхідно знайти ймовірність того, що $i + k - j$ клієнтів обслуговуються у проміжку часу між двома послідовними надходженнями групових потоків. Такий процес знаходження ймовірності вимагає точного знання поведінки системи між двома груповими потоками запитів, які надходять. Очевидно, що для $j > i + k$, $P(j,k,i) = 0$. Звідси випливає, що існує принаймні не більше $i + k$ запитів, що потребують обслуговування між моментами надходження A_n та A_{n+1}. Для розрахунку інших імовірностей переходів у аЕМС необхідно визначити функцію розподілу часу перебування запитів у кожній «ділянці» в eSMP. Для цього необхідно розглянути декілька випадків:

- Випадок 1: Час перебування в «ділянці» на початку відправки залишається часом обслуговування сервісу $B_+(x)$, з моменту останнього надходження запитів і є випадковою величиною в момент обслуговування поточного запиту.
- Випадок 2: якщо починається надходження другого групового потоку запитів з того ж сервера, то час обслуговування запитів дорівнює $B(x)$.
- Випадок 3: ні час відправки між надходженням двох групових потоків, ні останній час відправки перед наступним надходженням запитів не відповідатиме експоненціальному закону розподілу

- Випадок 4: Якщо i - час відправки запитів з іншого серверу, то функція часу перебування ділянки CDF набуває значення $B_{i+}(x)$.

Розглянемо час відправки D_{21} (рис. 2.6), який наступає після часу відправки D_{11}. Момент часу D_{11} можна розглядати як довільну точку в моменті часу обслуговування запитів на сервері 2, а отже, кумулятивна функція розподілу часу перебування в другому моменті відправки матиме вигляд $B_{2+}(x)$. В результаті функція розподілу часу обслуговування разом із $B_{2+}(x)$ є такою ж як і $B_{+}^{j}(s)$, аналогічно (2.9), але з додатковим кроком рекурсії. Зазвичай функція розподілу часу обслуговування з урахуванням часу знаходження запитів між різними серверами, може бути рекурсивно визначена та мати такий вигляд:

$$B_{i+}^{j}(s) = \frac{1 - B_{(i-1)+}^{j}(s)}{s * \bar{b}_{(i-1)+}}, i = 1, 2, 3\ldots. \qquad (2.11)$$

де $\bar{b}_{(i-1)+} = [\frac{d}{ds} B_{(i-1)+}^{j}(s)]_{s=0}$. Для підтримки узгодженості в позначеннях приймемо, що $\bar{b}_{0+} = \bar{b}, B_{0+}^{j}(s) = B^{j}(s)$ та $B_{1+}^{j}(s) = B_{+}^{j}(s)$.

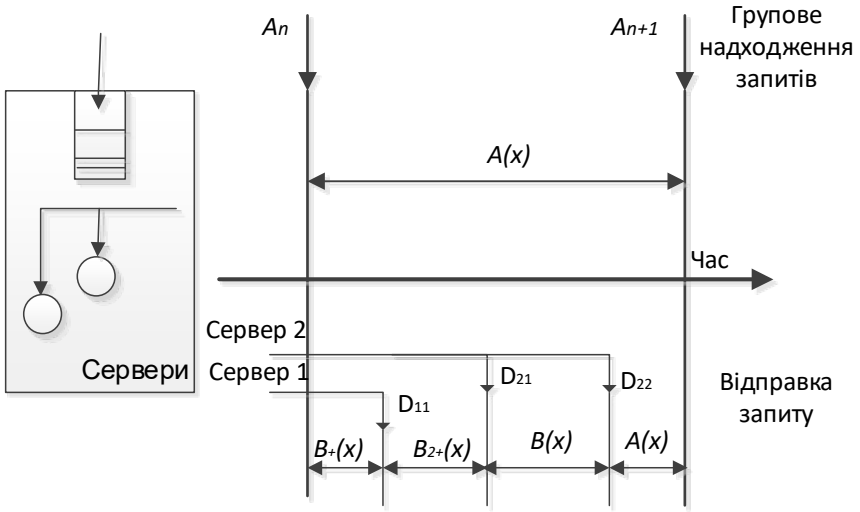


Рис.2.6. Поведінка системи між двома точками спостереження

Нехай $R_k(x)$ визначає кумулятивну функцію з часом перебування в стані k у eSMP та визначається:

$$R_k(x) = \begin{cases} B_+(x), & для\_випадку\_1 \\ B(x), & для\_випадку\_2 \\ A(x), & для\_випадку\_3 \\ B_{i+}(x), & для\_випадку\_4 \end{cases} i = 2,3,... \quad (2.12)$$

2.1.1 Визначення ймовірності стійкого стану хмарної системи та рівняння балансу

Для оцінки ефективності функціонування хмарної системи із врахуванням переходів зі стану в стан, необхідно визначити передавальну ймовірнісну матрицю, елементами якої є кількість запитів, що відправляються із системи в проміжку часу між двома прибулими послідовними груповими потоками запитів. Перш за все, для цього необхідно обчислити ймовірність існування k запитів під час перебування в кожному елементі. Нехай $N(B_+)$, $N(B)$ та $N(B_{i+})$, де $i = 2,3.....$ вказують на кількість запитів, які прибули в періоди часу $B_+(x)$, $B(x)$ та $B_{i+}(x)$ відповідно. Оскільки досліджувана система характеризується Пуассонівським потоком надходження запитів, можна визначити ці ймовірності:

$$\alpha_k \equiv P[N(B_+) = k] = \int_0^\infty \frac{(\lambda x)^k}{k!} e^{-\lambda x} dB_+(x) \quad (2.13)$$

$$\beta_k \equiv P[N(B) = k] = \int_0^\infty \frac{(\lambda x)^k}{k!} e^{-\lambda x} dB(x) \quad (2.14)$$

$$\delta_{ik} \equiv P[N(B_{i+}) = k] = \int_0^\infty \frac{(\lambda x)^k}{k!} e^{-\lambda x} dB_{i+}(x) \quad (2.15)$$

Проте, у подібних системах існує ймовірність, того, що в певний момент часу черга запитів порожня і запити на обслуговування не надходять. Відштовхуючись від цього, можна визначити ймовірність переходу в стан еЕМС наступним чином:

$$P_x \equiv P[N(B_+) = 0] = \int_0^\infty e^{-\lambda x} dB_+(x) = B_+^j(\lambda) \quad (2.16)$$

$$P_y \equiv P[N(B)=0] = \int_0^{\infty} e^{-\lambda x} dB_+(x) = B^j(\lambda) \tag{2.17}$$

$$P_{ix} \equiv P[N(B_{2+})=0] = \int_0^{\infty} e^{-\lambda x} dB_{i+}(x) = B_{i+}^j(\lambda) \tag{2.18}$$

$$P_{xy} = P_x P_y \tag{2.19}$$

Зазначимо, що $P_{ix} = P_x$, тому, завжди можна знайти ймовірність невідправки запитів між моментами надходження супер-задач. Нехай A – випадкова експоненціальна змінна з параметром λ; B_+ є випадковою змінною з розподілом відповідно до B_+(x) (час подальшого знаходження запиту). При цьому ймовірність відправки сервісу визначається:

$$\begin{aligned} P_z = P[A < B_+] = \int_{x=0}^{\infty} P\{A < B_+ \mid B_+ = x\} dB_+(x) = \int_0^{\infty} (1-e^{-\lambda x}) dB_+(x) = \\ = 1 - B_+^j(\lambda) = 1 - P_x \end{aligned} \tag{2.20}$$

Використовуючи ймовірності P_x, P_y, P_{xy}, P_{ix}, та P_z можна визначити ймовірність відправки для eEMC. Щоб зберегти модель гнучкою, вважаємо, що жодна віртуальна машина (virtual mashine – VM) не буде обробляти більше ніж три вихідні запити між двома груповими потоками, які послідовно прибули на обслуговування.

Ймовірності переходу *P* можуть бути зображені у вигляді матриці переходу (2.21), де рядки і стовпці відповідають кількості запитів у системі безпосередньо перед прибуттям групового потоку (стовпці) та відповідно відразу ж після чергового прибуття групового потоку (рядки). Можна виділити чотири робочих області, залежно від того чи вхідна черга порожня чи ні, до і після послідовних надходжень групових потоків запитів. Кожна область має яскраво виражену ймовірність переходу *P* - рівняння, яке залежить від поточного стану *i*, наступного стану *j*, кількості запитів в груповому потоці *k* і кількості вихідних запитів між надходженням двох групових потоків.

$$\left[\begin{array}{cccc|ccc}
(0,0) & (0,1) & \ldots & (0,m) & (0,m+1) & \ldots & (0,m+r) \\
\cdot & & \textit{Область_1} & & \cdot & \textit{Область_2} & \\
\cdot & & P_{LL}(i,j,k) & & \cdot & P_{LH}(i,j,k) & \\
\cdot & & & & \cdot & & \\
(m,0) & (m,1) & \ldots & (m,m) & (m,m+1) & \ldots & (m,m+r) \\
- & - & - & - & - & - & - \\
(m+1,0) & (m+1,1) & \ldots & (m+1,m) & (m+1,m+1) & \ldots & (m+1,m+r) \\
\cdot & & \textit{Область_4} & & \cdot & \textit{Область_3} & \\
\cdot & & P_{HL}(i,j,k) & & \cdot & P_{HH}(i,j,k) & \\
\cdot & & & & \cdot & & \\
(m+r,0) & (m+r,1) & \ldots & (m+r,m) & (m+r,m+1) & \ldots & (m+r,m+r)
\end{array}\right] \quad (2.21)$$

Область 1

В даній області визначається ймовірність переходу $P_{LL}(i,j,k)$, в якій вхідна черга порожня і залишається порожньою до наступного надходження запитів, тому переходи починаються і закінчуються в області запитів з мітками *m* чи без них. Позначимо кількість запитів, які виходять із системи між двома надісланими потоками запитів як $\omega(k) = i + k - j$. Для $i, j \le m$ імовірність переходу рівна:

$$P_{LL}(i,j,k) = \begin{cases} \sum_{z=0}^{\min(i,\omega(k))} \binom{i+k}{\omega(k)} P_x^{\omega(k)} (1-P_x)^j, if\_i+k \le m \\ \sum_{z=i+k-m}^{\min(i,\omega(k))} \binom{i}{z} P_x^z (1-P_x)^{i-z} \\ \binom{k}{\omega(k)-z} P_{xy}^{\omega(k)-z} (1-P_{xy})^{z-i+j}, if\_i+k > m \end{cases}, \quad (2.22)$$

Область 2

В даній області визначається ймовірність переходу $P_{LH}(i,j,k)$, в якій вхідна черга порожня та залишається порожньою до наступного надходження запитів, тобто $i \le m, j > m$. Тому переходи починаються та закінчуються в області запитів з мітками *m* чи без них, тому Це означає, що надіслані групові потоки запитів стають у чергу, так як не можуть обслуговуватися відразу через недостатню кількість вільних фізичних машин.

Відповідна ймовірність переходу матимуть вигляд:

$$P_{LH}(i,j,k) = \prod_{s=1}^{\omega(k)} \left[(i-s+1) P_{sx} \right] * (1-P_x)^{i-\omega(k)} \quad (2.23)$$

Для розрахунку ймовірності переходу для областей 3 та 4, потрібно знайти ймовірність існування n непрацюючих серверів та m простоюючих. Розглянемо надходження потоку запитів, який підпорядковується Пуассонівському закону розподілу і в якому кількість запитів випадкова. Кожен груповий потік запитів зберігається в кінцевій черзі. Зберігання групового надходження у черзі буде продовжено, поки черга або не буде повна або останнє надходження запитів не поміщається в ній.

Область 3

В даній області визначається ймовірність переходу $P_{HH}(i,j,k)$, в якій черга не порожня, тобто $i, j > m$. В цьому випадку переходи починаються і закінчуються в станах m як на рис.2.4, а ймовірність переходу в область може бути обчислено наступним добутком:

$$P_{HH}(i,j,k) \approx \sum_{\psi=(m-MBS+1)}^{m} \sum_{s1=\min(\omega(k),1)}^{\min(\omega(k),\psi)} \binom{\psi}{s_1} P_x^{s1} (1-P_x)^{\psi-s1} P_i(m-\psi) *$$
$$\sum_{\delta=\max(0,m-\psi+s1-MBS+1)}^{m-\psi+s1} \sum_{s2=\min(\omega(k)-s1,1)}^{\min(\delta,\omega(k)-s1)} \binom{\delta}{s_2} * P_x^{s2} (1-P_{2x})^{\delta-s2} P_i(m-\psi+s1-\delta) * \quad (2.24)$$
$$* \sum_{\phi=\max(0,m-\psi+s1-\delta+s2-MBS+1)}^{m-\psi+s1-\delta+s2} \binom{\phi}{\omega(k)-s1-s2} P_{3x}^{\omega(k)-s1-s2} (1-P_{3x})^{\phi-\omega(k)-s1-s2} *$$
$$* P_i(m-\psi+s1-\delta+s2-\phi)$$

Область 4

В даній області визначається ймовірність переходу $P_{HL}(i,j,k)$, у якій черга не порожня під час першого надходження запиту, але стає порожньою в момент наступного надходження потоку запитів: $i > m, j \le m$.

Ймовірність переходу для цієї області визначається:

$$P_{HL}(i,j,k) \approx \sum_{\psi=(m-MBS+1)}^{m} \sum_{s1=\min(0,\psi-j)}^{\min(\omega(k),\psi)} \binom{\psi}{s1} P_x^{s1}(1-P_x)^{\psi-s1} P_i(m-\psi) *$$

$$* \sum_{\delta=\max(0,m-\psi+s1-MBS+1)}^{m-\psi+s1} \sum_{s2=\min(\omega(k)-s1,\psi-j)}^{\min(\delta,\omega(k)-s1)} \binom{\delta}{s2} * P_{2x}^{s2}(1-P_{2x})^{\delta-s2} P_i(m-\psi+s1-\delta) * \quad (2.25)$$

$$* \sum_{\phi=\max(0,m-\psi+s1-\delta+s2-MBS+1)}^{m-\psi+s1-\delta+s2} \binom{\phi}{\omega(k)-s1-s2} P_{3x}^{\omega(k)-s1-s2} *$$

$$*(1-P_{3x})^{\phi-\omega(k)+s1+s2} P_i(m-\psi+s1-\delta+s2-\phi)$$

Тепер представимо рівняння балансу матриці переходу, яке матиме вигляд:

$$\pi_i = \sum_{j=\max[0,i-MRS]}^{m+r} \pi_j p_{ji}, 0 \le i \le m+r \quad (2.26)$$

яке доповнене рівнянням нормалізації $\sum_{i=0}^{m+r} \pi_i = 1$. Загальна кількість рівнянь становить $m+r+2$. Таким чином, необхідно видалити одне з рівнянь, щоб отримати єдине рівноважне рішення. Рівняння (2.26) цілком підходить, оскільки володіє мінімальним обсягом інформації про систему у порівнянні з усіма іншими.

2.1.2 Розподіл запитів в системі із врахуванням ступеню віртуалізації

Як тільки отримаємо ймовірності стійкого стану, встановимо значення твірної функції групового потоку для заданої кількості запитів у фізичних машинах в момент надходження групового потоку, що матиме вигляд::

$$\Pi(z) = \sum_{k=0}^{m+r} \pi_z z^k \quad (2.26)$$

Завдяки груповим надходженням не всі запити виконуються. Таким чином, функція $\prod(z)$ для розподілу кількості запитів фізичними машинами в довільний момент часу не співпадає з функцією $P(z)$ для розподілу кількості запитів у фізичних машинах у момент надходження групового потоку. Для отримання першої функції використовуємо техніку двокрокової апроксимації з вкладеним напівмарківським процесом та апроксимованим вкладеним марківським процесом, відповідно до 2.10 та 2.11.

Функція $P(z)$ з кількістю запитів в фізичній машині задається у вигляді:

$$P(z) = \sum_{i=0}^{m+r} p_i z^i \qquad (2.27)(15)$$

Це означає, що середня кількість запитів в РМ рівна $\overline{p} = P'(1)$.

Оскільки момент надходження запитів не залежить від стану буферу та розподілу кількості запитів у фізичній машині, то можна отримати ймовірність блокування групових запитів по фізичних машинах з розміром буфера r. Вона визначатиметься, таким чином:

$$P_{\text{block}}(r) = \sum_{k=0}^{MRS-1} [\sum_{i=0}^{MRS} p_{m+r-i-k}(1-G(i))] * P_i(k) \qquad (2.28)$$

Розмір буферу r_φ зобов'язаний зберігати ймовірність блокування нижче певного значення ψ, а саме:

$$r_\varphi = \{r \geq 0 \mid P_{\text{block}}(r) \leq \psi \;\; P_{\text{block}}(r-1) > \psi\} \qquad (2.29)$$

Отже, ймовірність блокування в хмарному центрі з M фізичними машинами розраховується як:

$$P_{block.cloud} = (P_{block})^M \qquad (2.30)$$

2.2. Дослідження адекватності моделі оцінки ефективності функціонування хмарних систем в умовах обслуговування групових потоків запитів

Отримані рівняння балансу розв'язані за допомогою програмного засобу Maple виробництва Maple Soft. Це система комп'ютерної алгебри, призначена для символьних обчислень, хоча має ряд засобів і для чисельного рішення диференціальних рівнянь і знаходження інтегралів. Володіє легко інтегрованими графічними засобами. Maple включає динамічну мову програмування в імперативному стилі (схожу на Паскаль), яка дозволяє визначати змінні будь якого формату. Володіє програмними інтерфейсами, які дають змогу проводити інтеграцію з системами написаними за допомого інших мов програмування(C, C #, Fortran, Java, MATLAB та Visual Basic), а також до Microsoft Excel. Maple підтримує MathML 2.0, який є форматом W3C для

представлення та інтерпретації математичних виразів, включаючи їх відображення на веб-сторінках. Існує також функція для перетворення виразів із традиційних математичних позначень у розмітку, придатну для системи LaTeX.

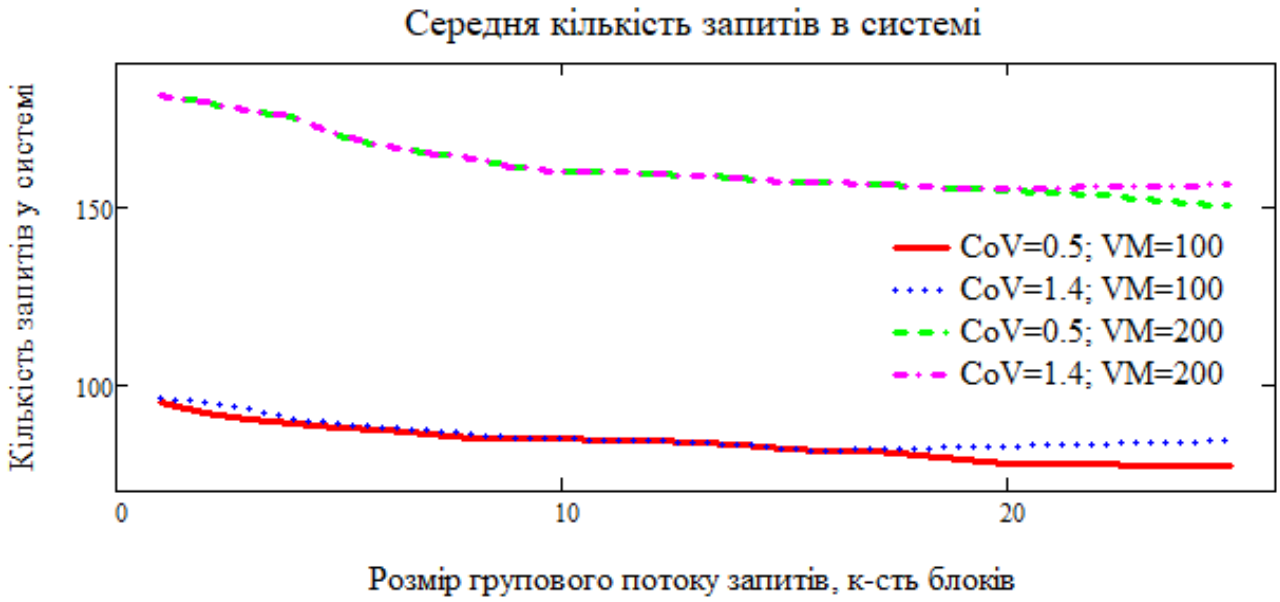
Maple заснований на невеликому ядрі, написаному мовою C, яке забезпечує інтеграцію мови Maple. Більшість функціональних можливостей забезпечують бібліотеки, які надходять із різних джерел. Більшість бібліотек написані мовою Maple; вони мають видимий вихідний код. Багато числових обчислень виконуються числовими бібліотеками NAG, бібліотеками ATLAS або бібліотеками GMP. Для різної функціональності Maple потрібні числові дані у різних форматах. Символічні вирази зберігаються в пам'яті як спрямовані ациклічні графіки. Стандартний інтерфейс та інтерфейс калькулятора написані на Java.

Для перевірки адекватності моделі оцінки ефективності функціонування хмарних систем в умовах обслуговування групових потоків запитів побудовано дискретний симулятор хмарної системи за допомогою засобів Artifex. Було налаштовано хмарну систему на роботу з M=1000 фізичних машин та двома ступенями віртуалізації: 100 та 200 віртуальних машин (VM – virtual mashine) на кожній фізичній машині. Розмір вхідної черги рівний r = 300 запитів. Припустимо, що час обслуговування запитів змінюватиметься пропорційно експоненційному закону розподілу. Важливим при такому розподілі є наявність коефіцієнту варіації, який встановлюється незалежно від середнього значення. Середнє значення залежить від кількості віртуальних машин, які здійснюють обслуговування запитів з врахуванням формули (2.4).

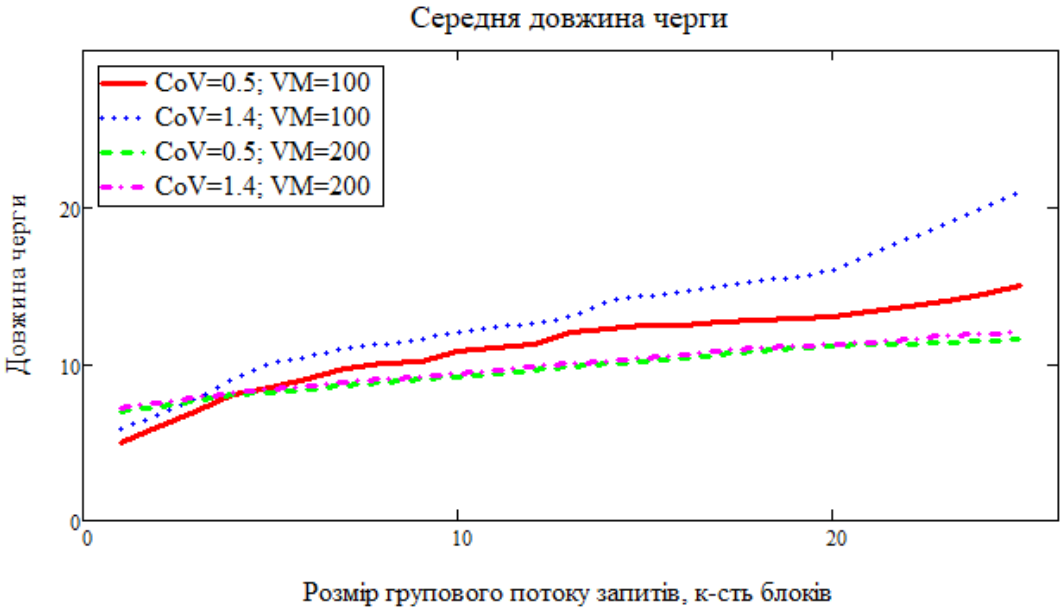
Для більшої достовірності встановимо час обслуговування запиту рівним 120 та 150 мс відповідно для помірного та високого ступеню віртуалізації. Використовуємо також два значення коефіцієнту варіації CoV=0,5 та 1,4, які відповідно призводять до гіпо- та гіперекспоненціально розподілених часів обслуговування.

В результаті на рис. 2.7 представлені результати для різних показників ефективності в залежності від розміру групового потоку запитів. В даному випадку критерієм ефективності функціонування системи будуть виступати група параметрів: ймовірність блокування, час відгуку, час обслуговування та час очікування в черзі. Червоною лінією позначено дослідження, при якому CoV=0,5, а кількість VM=100; синьою пунктирною лінією позначено дослідження, при якому CoV=1,4, а кількість VM=100; зеленою пунктирною лінією позначено дослідження, при якому CoV=0,5, а кількість VM=200; фіолетовою пунктирною лінією позначено дослідження, при якому CoV=1,4, а кількість VM=200.

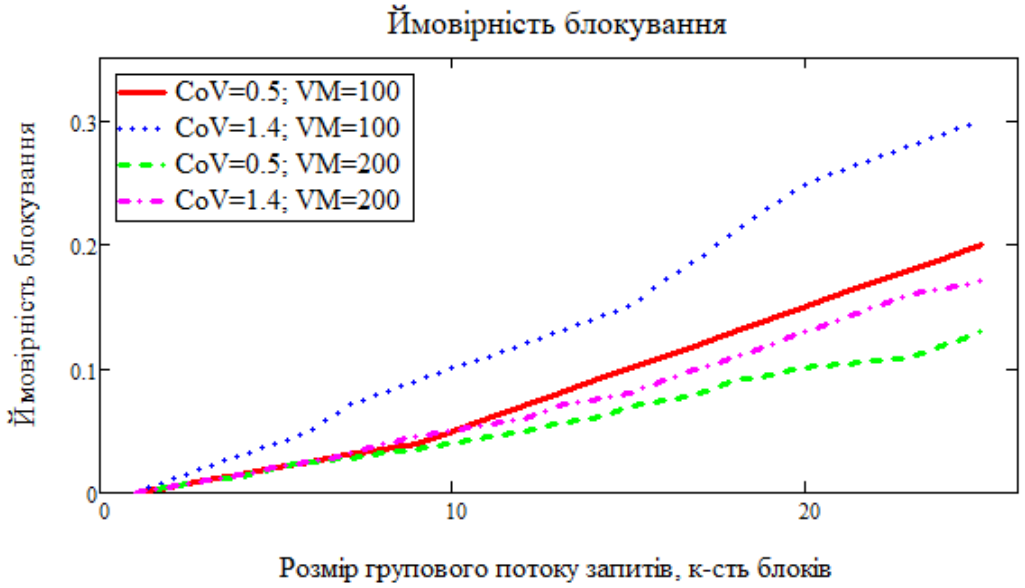
Як видно з графіків розмір черги збільшується в залежності від середнього розміру групового потоку, тоді як середня кількість запитів в системі зменшується (рис.2.7а).



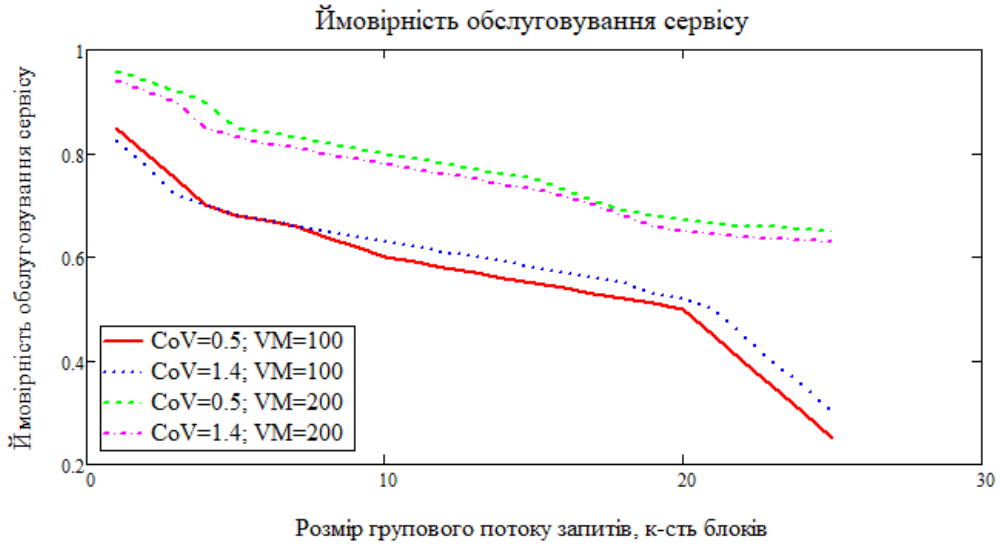
а)


Середня довжина черги
CoV=0.5; VM=100
CoV=1.4; VM=100
CoV=0.5; VM=200
CoV=1.4; VM=200
Довжина черги
0
10
20
0
10
20
Розмір групового потоку запитів, к-сть блоків

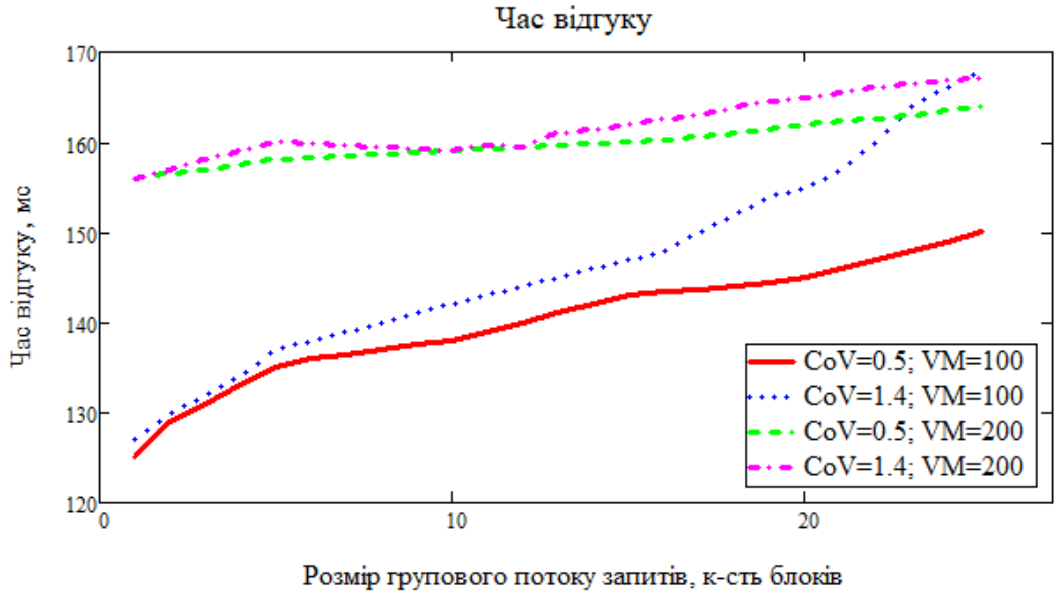
б)


Ймовірність блокування
CoV=0.5; VM=100
CoV=1.4; VM=100
CoV=0.5; VM=200
CoV=1.4; VM=200
Ймовірність блокування
0
0.1
0.2
0.3
0
10
20
Розмір групового потоку запитів, к-сть блоків

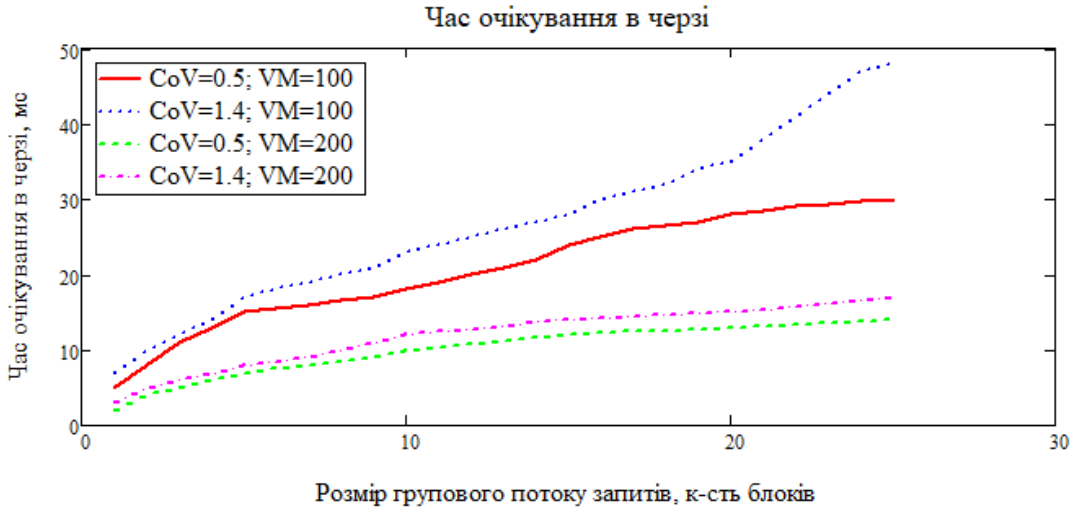
в)


Ймовірність обслуговування сервісу
CoV=0.5; VM=100
CoV=1.4; VM=100
CoV=0.5; VM=200
CoV=1.4; VM=200
Ймовірність обслуговування сервісу
0.2
0.4
0.6
0.8
1
0
10
20
30
Розмір групового потоку запитів, к-сть блоків

г)



д)



е)

Рис.2.7. Показники ефективності як функції середнього розміру групового потоку

Як видно з отриманих результатів, ймовірність блокування зростає, коли збільшується розмір групового потоку (рис. 2.7 в), а ймовірність обслуговування (яку можна назвати продуктивністю) спадає майже лінійно (рис. 2.7 г). Тим не менш, ймовірність обслуговування є вищою за 0,5 в досліджуваному діапазоні середніх розмірів пакетів, що означає, що більше 50% запитів від користувачів будуть обслуговуватися відразу ж після прибуття.

Як видно із залежностей представлених на рис.2.7д час відгуку – який є сумою часу очікування та часу обслуговування, а також час обслуговування (рис. 2.7г) повільно зростає зі збільшенням середнього розміру групового потоку. У всіх дослідах, результати моделювання для всіх випадків розгортання практично рівнозначні. Це підтверджує обґрунтованість запропонованої процедури апроксимації з використанням функцій eSMP , aEMP та aEMC.

В результаті дослідження встановлено, що використання гіперекспоненціального розподілу часу обслуговування (при CoV=1,4) є менш ефективним ніж використання гіпоекспоненціального (при CoV=0,5). Чим більший розмір групового потоку і/або значення коефіцієнта варіації часу обслуговування запиту, що не більший 1, тим довша тривалість відгуку, проте це сприяє зменшенню рівня використання ресурсів хмарними провайдерами. Обидва показники ефективності можуть бути поліпшені за рахунок гомогенізації, отриманої шляхом поділу вхідних групових потоків на основі кількості запитів і\або коефіцієнту варіації часу обслуговування запитів та обслуговування підпотоків окремими хмарними центрами. Таким чином, цей простий підхід пропонує значні переваги як для користувачів так і для провайдерів хмари.

2.3. Метод кластеризації вузлів хмарних систем в умовах обслуговування групових потоків запитів

Стрімкий розвиток і популяризація інформаційних сервісів вимагають нових підходів до систем, які забезпечують їх функціонування. Кластери забезпечують високий рівень доступності - в них відсутні єдина операційна система і спільно використовувана пам'ять, тобто немає проблеми когерентності кешу. При створенні кластерних систем використовується один з двох підходів.

Перший підхід застосовується для побудови невеликих кластерних систем, наприклад, на базі невеликих локальних мереж організацій або їх підрозділів. В

такому кластері кожна віртуальна машина продовжує працювати як самостійна одиниця, одночасно виконуючи функції вузла кластерної системи.

Другий підхід орієнтований на використання кластерної системи в ролі потужного обчислювального ресурсу. Вузлами кластера служать тільки системні блоки обчислювальних машин, які компактно розміщуються в спеціальних стійках. Управління системою і запуск завдань здійснює повнофункціональний хост-комп'ютер (орекестратор). Він же підтримує дискову підсистему кластера і різноманітне периферійне устаткування.

Кластери добре масштабуються шляхом додавання вузлів, що дозволяє досягти найвищих показників продуктивності. Завдяки цій особливості архітектури кластерів з сотнями і тисячами вузлів позитивно зарекомендували себе на практиці. За допомогою кластеризації у великих обчислювальних системах можна досягти:

- Абсолютної масштабованості. Можливе створення великих кластерів, що перевершують по обчислювальній потужності навіть найпродуктивніші поодинокі ВМ. Кластер взмозі утримувати десятки вузлів, кожен з яких представляє собою мультипроцесор.
- Нарощуваної масштабованості. Кластер будується так, що його можна нарощувати, додаючи нові вузли невеликими порціями. Таким чином, користувач або провайдер може почати з найпростішої системи, розширюючи її в міру необхідності.
- Високого коефіцієнту готовності. Оскільки кожен вузол кластера - самостійна ВМ або обчислювальна система, відмова одного з вузлів не призводить до втрати працездатності кластера. У багатьох системах відмовостійкість автоматично підтримується програмним забезпеченням.
- Хороше співвідношення ціна/продуктивність. Кластер будь-якої продуктивності можна створити, поєднуючи стандартні «будівельні блоки», при цьому його вартість буде нижче, ніж у одиночній ВМ з еквівалентною обчислювальною потужністю.

У той же час потрібно згадати і основний недолік, властивий кластерним системам, - взаємодія між вузлами кластера займає набагато більше часу, ніж в інших типах обчислювальних систем. Саме тому необхідно запропонувати новий підхід до кластеризації та вибору головного кластерного вузла, що дозволить підвищити ефективність функціонування хмарних систем, в яких одночасне обслуговування групових потоків запитів забезпечує велика кількість віртуальних машин. Такий підхід повинен базуватися на групуванні в кластер пулу віртуальних машин, які будуть використовуватися під обслуговування чітко визначеного групового потоку із врахуванням географічної близькості вузлів між собою та моніторингу доступних програмно-апаратних та телекомунікаційних ресурсів, що дасть змогу підвищити життєвий цикл фізичних машин за рахунок збільшення рівня залишкової енергії.

Розглянемо таку ж хмарну систему як на рис. 2.1, в основі якої функціонує M фізичних машин із кількістю віртуальних машин m. Груповий потік запитів, що надходить в систему на обслуговування підпорядковується тим же законам розподілу що й у п. 2.1. Завдяки удосконаленню математичної моделі оцінки ефективності функціонування хмарних систем отримаємо певний баланс між кількістю одночасно розгорнутих і функціонуючих віртуальних машин та часом обслуговування групових потоків запитів.

Визначимо межі кожного кластеру, шляхом групування віртуальних машин, які будуть використовуватися під обслуговування чітко визначеного групового потоку із врахуванням географічної близькості вузлів між собою та моніторингу доступних програмно-апаратних та телекомунікаційних ресурсів та головний вузол в межах кластеру. Для формування кластера використаємо принцип діаграм Вороного [66]. У кожному раунді хмарна система поділяється випадковим чином на кластери. Кластери формуються по принципу знаходження як найменшої відстані один до одного, а центроїд кластеру обирається випадковим чином. Після вибору головного вузла кластера в

кожному кластері обчислюється відстань між вузлами і головним вузлом кластера з використанням формули (2.31).

$$D(S_i, C_j) = \sum_{i=1}^{m} \sqrt{(S_i - C_j)^2} \quad (2.31)$$

де S_i - вузол *i* в кластері ($i = 1,...,m$) і C_j - головний вузол *j* кластера ($j = 1,...,k$).

А також для кожної фізичної машини оцінюється енергія, яка необхідна для передачі *k* бітів даних на відстань *d* визначатиметься

$$E_{T_x}(k,d) = E_{T_x-elec}(k) + E_{T_x-amp}(k,d) \quad (2.32)$$

де $E_{T_x}(k,d)$ - енергія, що витрачається на передачу, $E_{T_x-elec}(k)$ і $E_{T_x-amp}(k,d)$ - енергії, необхідні для роботи сервера (PM).

Проте при такому класичному використанні діаграм Вороного, немає можливості перевірити правильність вибору центроїда кластеру, оскільки дана методика не враховує чи достатньо в такого центроїда програмно-апаратних потужностей для подальшого функціонування. Будучи головним вузлом кластеру, крім традиційного обслуговування запитів, такий вузол повинен буде здійснювати моніторинг роботи інших вузлів кластеру та комунікувати з іншими кластерами. У дисертаційній роботі пропонуємо розробити такий алгоритм вибору головного вузла кластеру.

2.3.1. Визначення головного вузла кластеру

Для визначення головного вузла в кластері використаємо теорію нечіткої логіки. Для її реалізації реалізуємо Контролер нечіткої логіки (рисунок 2.8). Контролер нечіткої логіки FLC (Fuzzy Logic Controller) складається з наступних компонентів: блоку фазифікації, бази правил, блоку нечіткого виведення і блоку дефаззифікації [67-70]. В якості вхідних параметрів для Контролера використаємо залишкову енергія і центральність, які визначені відповідно до принципу Вороного.

Від усіх фізичних машин на вхід контролера нечіткої логіки надходять дані про стан кожного вузла.

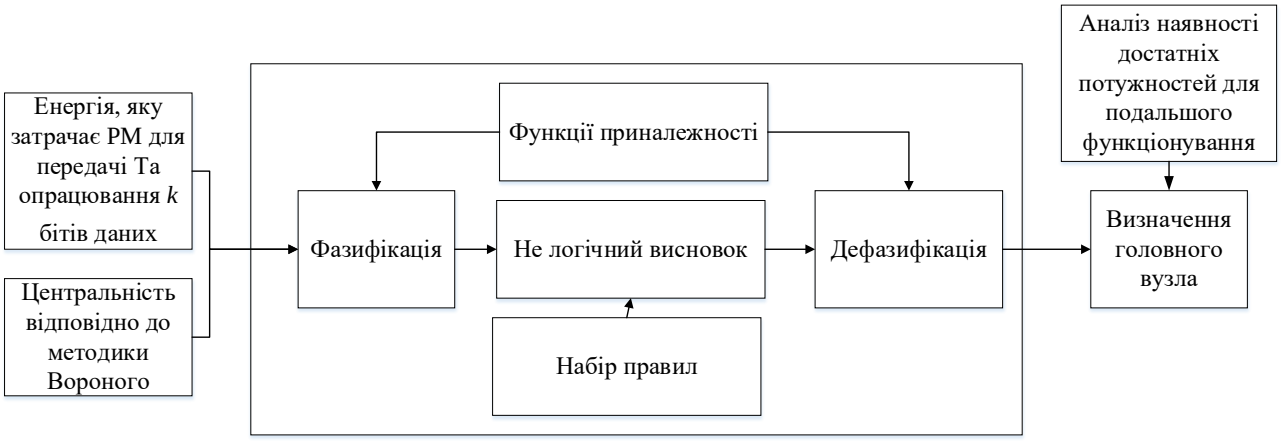


Рис. 2.8. Контролер нечіткої логіки вибору головного вузла

Фазифікація – це процес, що перетворює точні значення вхідних змінних в значення лінгвістичних (нечітких) змінних за допомогою застосування певних функцій приналежності.

У таблиці 2.1 показані параметри системи нечіткого виведення і їх нечіткі множини. На виході з FLC будемо мати ймовірність вибору головного вузла у відсотках.

Таблиця 2.1

Параметри системи нечіткого виведення і їх нечіткі множини

Тип змінної	**Змінна**	**Терм множини**	**Тип функції приналежності**	**Значення параметрів функції приналежності**
Вхідна	Залишкова енергія [0; 0.1] Дж	Низька	z-подібна	[0; 0.02; 0.05]
		Середня	Трикутна	[0.02; 0.05; 0.08]
		Висока	s-подібна	[0.05; 0.08; 0.1]
	Центральність по діаграмах Вороного [0; 100] %	Дальня	z-подібна	[0; 25; 50]
		Середня	Трикутна	[20; 50; 80]
		Близька	s-подібна	[50; 75; 100]

Продовження таблиці 2.1

Вихідна	Ймовірність вибору головного вузла [0; 100] %	Дуже низька	z-подібна	[0; 5; 15]
		Низька	Трикутна	[5; 15; 25]
		Більше низької	Трикутна	[15; 25; 35]
		Менше середньої	Трикутна	[25; 35; 45]
		Середня	Трикутна	[35; 50; 65]
		Більше середнього	Трикутна	[55; 65; 75]
		Не велика	Трикутна	[65; 75; 85]
		Велика	s-подібна	[75; 80; 95]
		Дуже велика	s-подібна	[85; 95; 100]

В якості функцій приналежності для кожного терма всіх лінгвістичних змінних вибираємо трикутні функції приналежності. Після визначення функції приналежності і вхідних параметрів необхідно визначити базу правил для відповідних параметрів.

На виході нечіткого контролера отримуємо ймовірність вибору головного вузла, яка виходить в результаті дефазифікації вихідного нечіткого рішення.

База правил, яка називається лінгвістичної моделлю, являє собою безліч нечітких правил $R^k, k = 1,..., N$ виду:

$$R^k : \begin{matrix} ЯКЩО(x_1 - A_1^k Ix_2 - A_2^k ...Ix_n - A_n^k) \\ РЕЗУЛЬТАТ(y_1 - B_1^k Iy_2 - B_2^k ...Iy_n - B_n^k) \end{matrix} \quad (2.33)$$

де n - кількість нечітких правил;

A_1^k - нечіткі множини $A_1^k \subseteq X_i \subset R,\ i = 1,..., n$

У таблиці 2.2 показані правила нечіткого виведення для вибору головного вузла.

Таблиця 2.2

Правила нечіткого виведення для вибору головного вузла

№ правила	*Якщо* Залишкова енергія	*І* Центральність по діаграмі Вороного	*Результат* Ймовірність вибору головного вузла
1	низька	Дальна	Дуже низька

Продовження таблиці 2.2

№ правила	*Якщо* Залишкова енергія	*І* Центральність по діаграмі Вороного	*Результат* Ймовірність вибору головного вузла
2	низька	Середня	Низька
3	низька	Близька	Більша низької
4	середня	Дальна	Менше середнього
5	середня	Середня	Середня
6	середня	Близька	Більше середнього
7	висока	Дальна	Невелика
8	висока	Середня	Велика
9	висока	Близька	Дуже велика

В якості Z-норми будемо використовувати операцію мінімуму (min), в якості S-норми - операцію максимуму (max):

$$\begin{aligned} \mu_A(x) *^T \mu_B(y) &= \min[\mu_A(x), \mu_B(y)] \\ \mu_A(x) *^S \mu_B(y) &= \max[\mu_A(x), \mu_B(y)] \end{aligned} \tag{2.34}$$

Правило нечіткої імплікації задамо правилом Мамдані:

$$\mu_{A\to B}(x,y) = \mu_R(x,y) = \mu_A(x) \cap \mu_B(y) = \min[\mu_A(x), \mu_B(y)] \tag{2.35}$$

де A і B - нечіткі множини $A \subseteq X, B \subseteq Y$, відношення R визначено на $X * Y$.

Оскільки на виході блоку отримання рішення отримуємо одну нечітку множину $B^{=k} \subseteq Y$ (ймовірність вибору головного вузла), то нечіткий висновок визначається:

$$\begin{aligned} &\textit{Умова}: X = (x_1, x_2)^T \textit{ це } \_\ A', \text{ де } A' = A_1' * A_2' \\ &\textit{Імплікація}: \bigcap_{k=1}^{N} R^{(k)}, R^{(k)} : A^{(k)} \to B^{(k)} \\ &\textit{Висновок}: y \to B' \end{aligned} \tag{2.36}$$

Таким чином, ймовірність вибору головного вузла обчислюється як:

$$B' = A' * \bigcup_{k=1}^{N} R^{(k)} \tag{2.37}$$

Так як правило, нечіткої імплікації ми визначили Z-нормою типу min, то виконується умова, в якій:

$$B' = A' * \bigcup_{k=1}^{N} R^{(k)} = \bigcup_{k=1}^{N} R^{(k)} * A' \tag{2.38}$$

Отже, функція приналежності нечіткої множини B' дорівнює:

$$\mu_{B'}(y) = \max_{k=1...N} \mu_{\overline{B}^k}(y) \tag{2.39}$$

де функція приналежності $\mu_{\overline{B}^k}(y)$ задається виразом:

$$\mu_{\overline{B}^k}(y) = \mu_{A^{(k)} \to B^{(k)}}(\overline{x}, y) \tag{2.40}$$

Таким чином, на вхід нечіткого контролера подається $\overline{x} = (\overline{x_1}, \overline{x_2})$. Виходячи з (2.40), отримуємо:

$$\mu_{\overline{B}^k}(y) = \mu_{A^{(k)} \to B^{(k)}}(\overline{x}, y) = \mu_{(A_1^k * A_2^k) \to B^k}(\overline{x_1}, \overline{x_2}, y) \tag{2.41}$$

Так як в якості нечіткої імплікації використовується правило мінімуму, а декартовий простір представляється у вигляді нечітких множин задається також операцією мінімум, отримуємо:

$$\mu_{\overline{B}^k}(y) = \min\{\mu_{(A_1^k * A_2^k) \to B^k}(\overline{x_1}, \overline{x_2}, y)\} = \min\{\mu_{A_1^k}(\overline{x_1}), \mu_{A_2^k}(\overline{x_2}), \mu_{B^k}(y)\} \tag{2.42}$$

В результаті, використовуючи вираз (2.45), отримуємо:

$$\mu_{B'}(y) = \max_{k=1...N}[\min\{\mu_{A_1^k}(\overline{x_1}), \mu_{A_2^k}(\overline{x_2}), \mu_{B^k}(y)\}] \tag{2.43}$$

де $\overline{x_1}$ і $\overline{x_2}$ - відповідно вхідні параметри (залишкова енергія вузла і центральність по діаграмах Вороного), A_1^k і A_2^k - відповідні їм нечіткі множини, $k = 1,..., N$ - правило нечіткого виведення, N - кількість правил нечіткого виводу y - вихідний параметр (ймовірність вибору головного вузла), B_1^k - відповідна їй множина. Відповідно до отриманої бази правил можна виконати операцію нечіткого висновку. Як висновок для кожного правила використовується лінгвістична змінна «ймовірність вибору головного вузла» y, множину значень якої складається з дев'яти термів: «дуже мала», «мала», «більше малої», «менше середньої», «середня», «більше середньої», «невелика», «велика» і «дуже велика». Значення ймовірності вибору головного вузла виходить в результаті операції дефазифікації вихідної нечіткої множини. Дефазифікація вихідного значення контролера (значення ймовірності вибору головного вузла) будемо виробляти за методом центра ваги, використовуючи формулу:

$$\overline{y} = \left(\sum_{k=1}^{N} a_k \int_y \mu_{\overline{B^k}}(y) dy \right) / \left(\sum_{k=1}^{N} \int_y \mu_{\overline{B^k}}(y) dy \right) \quad (2.44)$$

де $\mu_{\overline{B^k}}(y)$ - функція приналежності правила вихідного нечіткого множини *k*-го правила бази правил, $k = 1,...,n; a_k$ є точкою, в якій дана функція приналежності приймає значення 1.

В умовах групового надходження запитів або обробки потоків типу Big Data при виборі головного вузла кластеру слід враховувати і наявність в такого вузла достатніх апаратних та телекомунікаційних ресурсів. Розглянемо випадок, коли при виконанні умов 2.32 – 2.44 обраний вузол, в силу недостатньої програмно-апаратної потужності, не зможе належним чином виконувати свої функції. З одного боку, це вплине на значення енергії необхідної для передачі *k* бітів даних на відстань *d*, з іншого – недостатність пропускної здатності вхідного/вихідного каналу передавання вплине на якість обслуговування групових потоків. В такому випадку пропонуємо враховувати наявність вільних ресурсів. Якщо для контролю за вузлами в межах кластеру та обслуговування запитів, що надійшли в систему обраний головний вузол кластеру буде завантажений більше ніж на певне критичне значення, відповідно, визначення головного вузла кластеру розпочинається спочатку.

Для оцінки параметрів віртуальних машин, які здійснюють обробку M_i запитів із групового потоку скористаємося співвідношеннями:

$$CPU_{pr} = \frac{\sum_{i=1}^{k} M_i * CPU_i}{\sum_{i=1}^{k} M_i} \qquad RAM_{pr} = \frac{\sum_{i=1}^{k} M_i * RAM_i}{\sum_{i=1}^{k} M_i} \quad (2.45)$$

де M_i – кількість запитів групового потоку у *i* -тій VM (Virtual mashine); CPU_i – тактова частота процесора VM, яку використовує *i*-тий запит; RAM_i – об’єм оперативної пам’яті VM, яку використовує *i*-тий запит; *k* – ступінь віртуалізації, тобто кількість VM на одній PM.

Нехай $v = \overline{1,V}$ - віртуальна машина, на якій виконуються $i = \overline{1,K}$ запитів із групового потоку $j = \overline{1,S}$, а $m = \overline{1,M}$ – фізична машина, на якій знаходиться $V_m = \overline{1,V}$ віртуальних машин; $P = \{P_1,\ldots,P_n,\ldots P_N\}, n = \overline{1,N}$, $P_n = \{e_1,\ldots e_l,\ldots e_L\}, l = \overline{1,L}$ – множини шляхів, де N – кількість шляхів, які необхідні для передавання виконаних запитів i та $i+1$, L – кількість переходів (ребер) у шляху n, які з'єднують віртуальні машини VM_v та VM_{v+1}.

Спочатку перевіряється вхідна черга запитів. Якщо там міститься $f = \overline{1,F}$ запитів групового потоку j, запускається аналізатор, який визначає, які ресурси необхідні для його обслуговування. Як результат, формується таблиця значень CPU, RAM, вільної пропускної здатності, які забезпечать передавання виконаного запиту за час $t < t_{кр}$ та тип віртуальної машини. Для виконання кожного запиту використовуються апаратні потужності - CPU, та RAM як фізичної так і віртуальної машин. Перевіряємо, чи v-та віртуальна машина взмозі обслужити наявну кількість запитів i із множини $f = \overline{1,F}$ групового потоку. Якщо ресурсів достатньо, то перевіряємо, чи вистачить ресурсів каналу для того, щоб передати обслужені запити.

$$C_{P_n} > R_{i,j}, \tag{2.46}$$

де $R_{i,j}$ - швидкість інформаційного потоку, що генерує *i*-тий запит *j*-го потоку. Всі параметри, які будуть доступні у момент вибору головного вузла кластеру і будуть записані у $\overline{A}$ масив, що буде формувати вхідну чергу на обслуговування.

Нехай ваговий коефіцієнт швидкості *i*-ого запиту *j*-ого групового потоку відносно загальної кількості запитів у супер запиті, передається по *l*-ому каналу і може бути визначеним за наступним співвідношенням:

$$r_{i,j|l} = \frac{R_{i,j|l}}{\sum_{i,j} R_{i,j|l}} \tag{2.47}$$

Величина використаної пропускної здатності каналу буде пропорційною до вагового коефіцієнту швидкості *i*-ого запиту *j*-ого групового потоку відносно загальної кількості запитів у груповому потоці запитів та визначатиметься за наступним співвідношенням:

$$C_{i,j|l} = \frac{R_{i,j|l}}{C_l} \cdot 100\% \qquad (2.48)$$

де C_l – смуга пропускання *l*-ого каналу.

Якщо при оцінці вільних потужностей виконується хоча б одна з умов:

$$\begin{gathered} CPU_{available}(v) > CPU(i,j) \\ RAM_{available}(v) > RAM(i,j) \\ C_{P_n} \min\left(1 - \frac{\sum_{i,j} R_{i,j|l}}{C_l} \mid \mathrm{P}_n\right) > R_{i,j} \mid P_n \end{gathered} \quad , \qquad (2.49)$$

то вибір головного вузла в кластері починається спочатку

Блок-схема алгоритму кластеризації і вибору головного вузла кластера наведено на рис. 2.9.

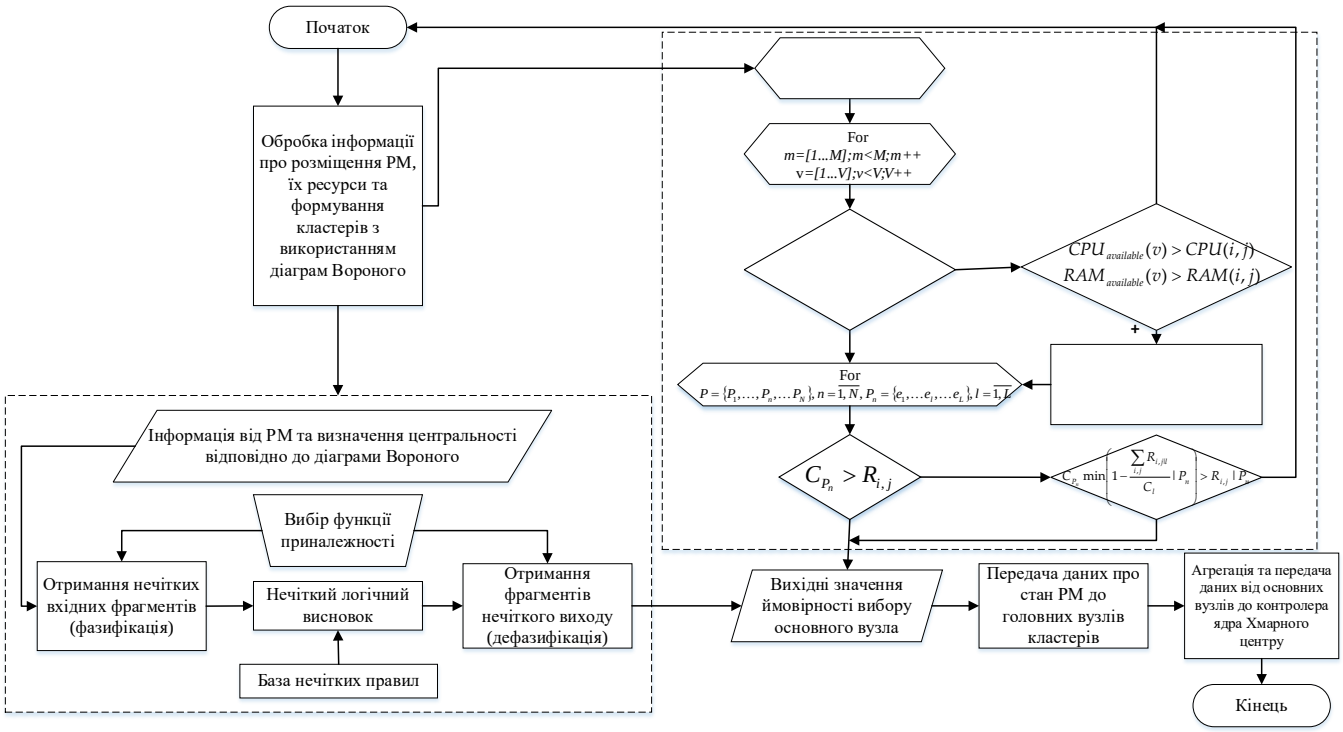


Рис. 2.9. Блок-схема алгоритму кластеризації і вибору головного вузла кластера

Висновки до 2-го розділу

Удосконалено математичну модель оцінки ефективності функціонування хмарних систем в умовах групового надходження запитів від користувачів. Модель використовує теорію масового обслуговування та ймовірнісний аналіз для отримання ряду показників ефективності, в тому числі час відгуку, час очікування в черзі, довжину черги, ймовірність блокування, ймовірність обслуговування сервісу та ймовірність розподілу кількості запитів у системі. Вона базується на двоступеневій техніці апроксимації, де основний немарківський процес спочатку моделюється як вкладений напівмарківський процес, який потім моделюється як апроксимований процес Маркова, але тільки в моменти надходження групових потоків запитів. Дана модель забезпечує повний імовірнісний розподіл часу очікування запитів, часу відгуку щодо виконання запитів і кількості запитів у системі. На відмінну від існуючих моделей запропонована модель враховує можливість обслуговування при груповому надходженні запитів та розподілений час обслуговування запитів, що дасть змогу в моменти пікових навантажень не порушувати працездатність мережі з високим ступенем віртуалізації за рахунок зменшення тривалості очікування запитів в черзі. Модель дозволяє знаходити баланс між кількістю віртуальних машин, що здійснюють обслуговування групових потоків запитів та кількістю запитів, що вже знаходяться на обслуговуванні в системі

Результати перевірки адекватності моделі оцінки ефективності функціонування хмарних систем в умовах обслуговування групових потоків запитів доводить, що продуктивність роботи хмарних систем значно залежить від коефіцієнту варіації (CoV), часу обслуговування запитів і розміру групового потоку (тобто кількості запитів у груповому потоці запитів). Чим більший розмір потоку та/або значення коефіцієнта варіації, часу обслуговування запитів, тим довша тривалість відгуку. Проте, це сприяє зменшенню рівня використання ресурсів хмарними провайдерами і як наслідок забезпечується стабільне функціонування системи. Обидва показники ефективності можуть

бути поліпшені за рахунок гомогенізації, отриманої шляхом поділу вхідних групових потоків на основі кількості запитів та/або коефіцієнту варіації часу обслуговування запитів та обслуговування підпотоків окремими хмарними центрами. Встановлено, в умовах надходження великих групових потоків запитів та/або великого значення CoV можна досягти підвищення ефективності функціонування хмарних центрів шляхом групування запитів за критерієм однорідності.

Удосконалено метод кластеризації вузлів хмарних систем в умовах обслуговування групових потоків запитів, що дозволить підвищити ефективність функціонування хмарних систем, в яких одночасне обслуговування групових потоків запитів забезпечує велика кількість віртуальних машин. Такий підхід базується на групуванні в кластер пулу віртуальньних машин, які будуть використовуватися під обслуговування чітко визначеного групового потоку із врахуванням географічної близькості вузлів між собою та моніторингу доступних програмно-апаратних та телекомунікаційних ресурсів, що дасть змогу підвищити життєвий цикл фізичних машин за рахунок збільшення рівня залишкової енергії.

Розроблено алгоритм кластеризації і вибору головного вузла кластера, особливістю якого є визначення головного вузла кластера із врахуванням центральності по діаграмах Вороного, правил нечіткої логіки та моніторингу параметрів вузлів. Ухвалення рішення про вибір головного вузла здійснюється методами нечіткої логіки за допомогою правила Мамдані та центру ваги, а також із врахуванням достатніх програмно-апаратних ресурсів для подальшого функціонування вузла в якості центроїда кластеру. Такий підхід дозволить підвищити ефективність функціонування хмарних систем шляхом збільшення життєвого циклу фізичних машин, особливо в умовах великої кількості віртуальних машин та обслуговування групових потоків запитів.

РОЗДІЛ 3. МОДЕЛЮВАННЯ ТА ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ФУНКЦІОНУВАННЯ ХМАРНИХ СИСТЕМ У ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СЕРВІСНО-ОРІЄНТОВАНИХ МЕРЕЖАХ

3.1.Дослідження ефективності кластеризації вузлів хмарних систем із застосуванням правил нечіткої логіки та моніторингу параметрів вузлів

Для дослідження ефективності кластеризації вузлів хмарних систем проведемо моделювання запропонованого алгоритму у п. 2.3.1 в програмному середовищі Matlab. Як інструмент імітаційного моделювання використовувався програмний пакет Fuzzy Logic Toolbox (пакет нечіткої логіки в середовищі MatLab) [71, 72]. Для розробки і подальшого застосування систем нечіткого виводу в інтерактивному режимі використаємо наступні графічні засоби, що входять до складу пакета Fuzzy Logic Toolbox:

- Редактор систем нечіткого вводу/виводу FIS (FIS Editor).
- Редактор функцій приналежності системи нечіткого виведення (Membership Function Editor) .
- Редактор правил системи нечіткого виведення (Rule Editor) .
- Програма перегляду правил системи нечіткого виведення (Rule Viewer) .

Крім цих графічних засобів до складу пакета Fuzzy Logic Toolbox також входять:

- Редактор адаптивних систем нейро-нечіткого виводу (Adaptive Neuro-Fuzzy Inference System Editor) .
- Програма нечіткої кластеризації методом нечітких K-середніх (fuzzy C-means clustering)

Пакет для нечіткої кластеризації методом нечітких K-середніх (fuzzy C-means clustering) будемо використовувати для порівняння розробленої нами моделі кластеризації.

Редактор систем нечіткого виводу FIS (або просто редактор FIS) є основним засобом, який ми використовуватимемо для створення або редагування систем нечіткого виводу в графічному режимі. Редактор FIS може

бути відкритий за допомогою введення функції fuzzy або fuzzy ('FISfile') в командному рядку. Ця функція надає користувачеві можливість задавати і редагувати якість системи нечіткого виведення, зокрема число вхідних і вихідних змінних, тип системи нечіткого виведення, метод дефаззифікації і т. д.

Якщо функція fuzzy викликається без аргументів, то редактор FIS викликається для новостворюваної системи нечіткого виведення за замовчуванням. За замовчуванням також задається ряд таких параметрів, таких як тип системи нечіткого виведення (Мамдані), нечіткі логічні операції, методи імплікації, агрегування і дефаззифікації.

Редактор FIS володіє графічним інтерфейсом і дозволяє викликати всі інші редактори і програми перегляду систем нечіткого висновку. Графічний інтерфейс редактора володіє достатньою зручністю і гнучкістю, необхідною для інтерактивної роботи з окремими компонентами системи нечіткого виводу.

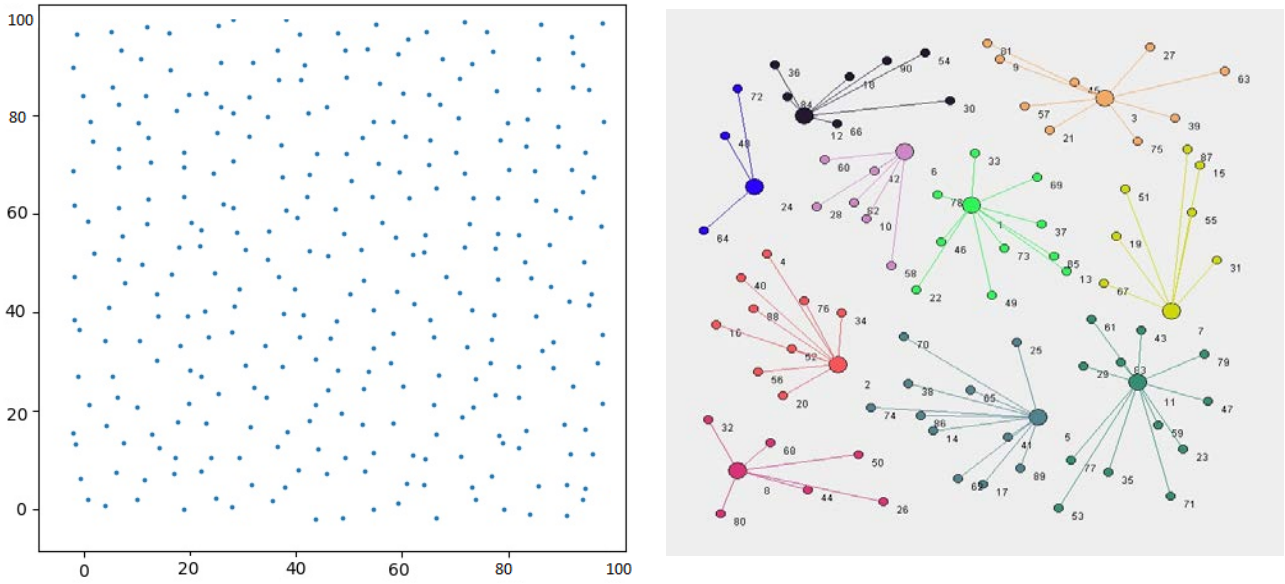
Редактор функцій приналежності призначений для завдання і редагування функцій приналежності окремих термів системи нечіткого виведення в графічному режимі. Сюди ми вносимо описані в розділі функції приналежності

Редактор правил системи нечіткого виведення - призначений для завдання і редагування окремих правил системи нечіткого виведення в графічному режимі. Ця функція, записана в форматі ruleedit ('FISfile'), і викликає редактор правил, який дозволяє користувачеві в графічному режимі аналізувати і модифікувати правила системи нечіткого виведення FIS. Функція дозволяє також виконувати граматичний аналіз правил, які використовуються в деякій системі нечіткого виведення FIS.

Перший етап, який перевіримо за допомогою даної імітаційної моделі – ефективність запропонованого алгоритму кластеризації.

В якості моделі мережі використовується модель зі 100 вузлів, розподілених випадковим чином на площині розміром 100 * 100 м^2. Після того, як вузли розподілені випадковим чином, відбувається перехід до формування

кластерів з використанням діаграм Вороного і вибору головного вузла кластера на основі нечіткої логіки відповідно до п. 2.3



а) б)

Рис. 3.1. Розподіл вузлів на площині а) до формування кластерів; б) з формуванням кластерів із використанням діаграм Вороного і вибору головного вузла кластера на основі нечіткої логіки

Кожен з вхідних параметрів моделювання в реальних системах не є сталим і змінюється в залежності від обраного програмного та апаратного забезпечення. Наприклад, розміри пакетів будуть змінюватися в залежності від протоколу та корисного навантаження, тривалість обробки від продуктивності апаратного чи програмного забезпечення та алгоритму обробки. Для максимальної наближеності до реальних умов згенеруємо потік трафіку. Повідомлення генерувалися протягом 10000 с, розмір кожного з них – 500 байт. Середня тривалість обробки повідомлення одним вузлом становить 1 мкс. Моделювання проводилось протягом 100 сек. Характеристика згенерованого навантаження наведено на рис. 3.2

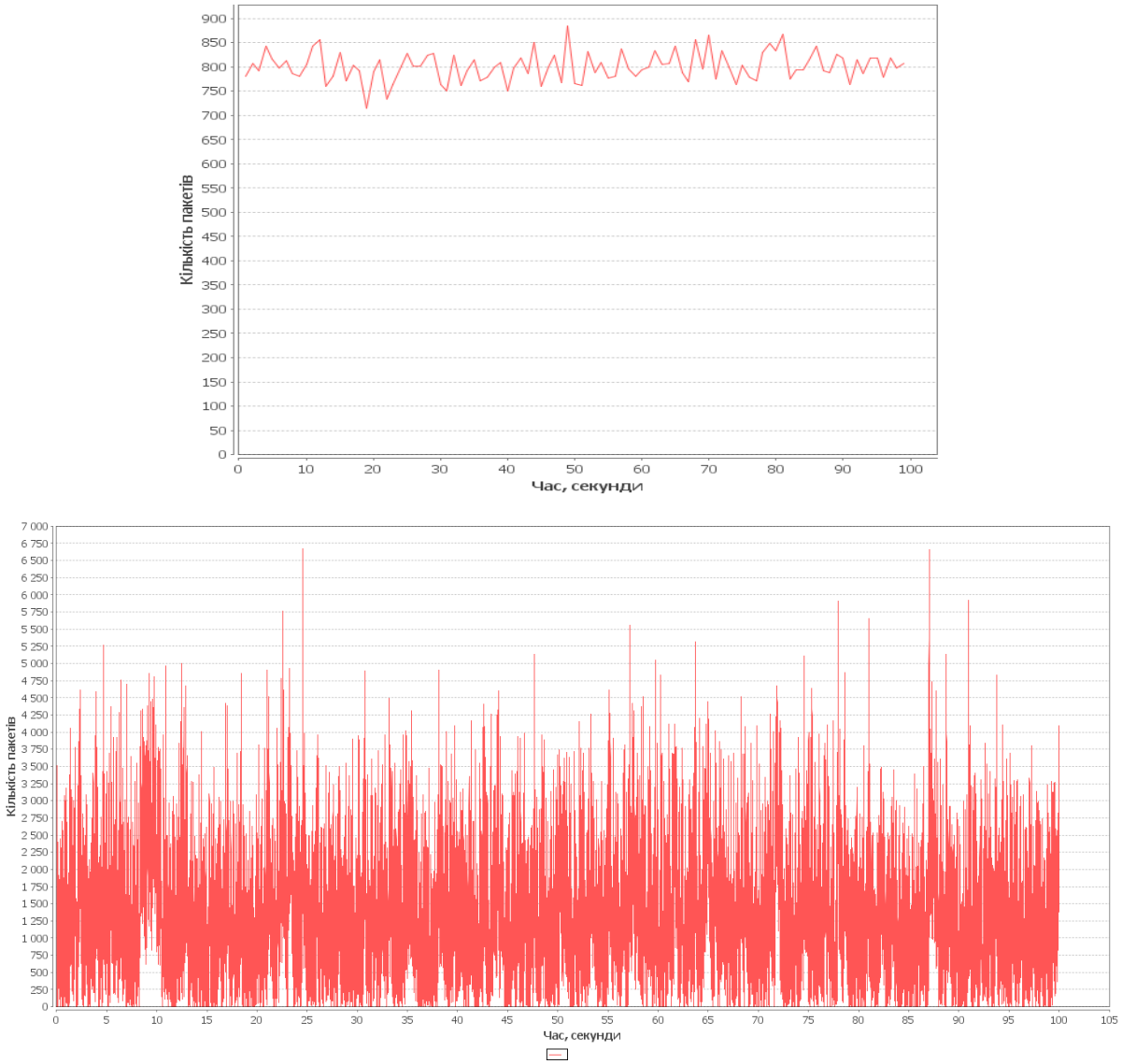


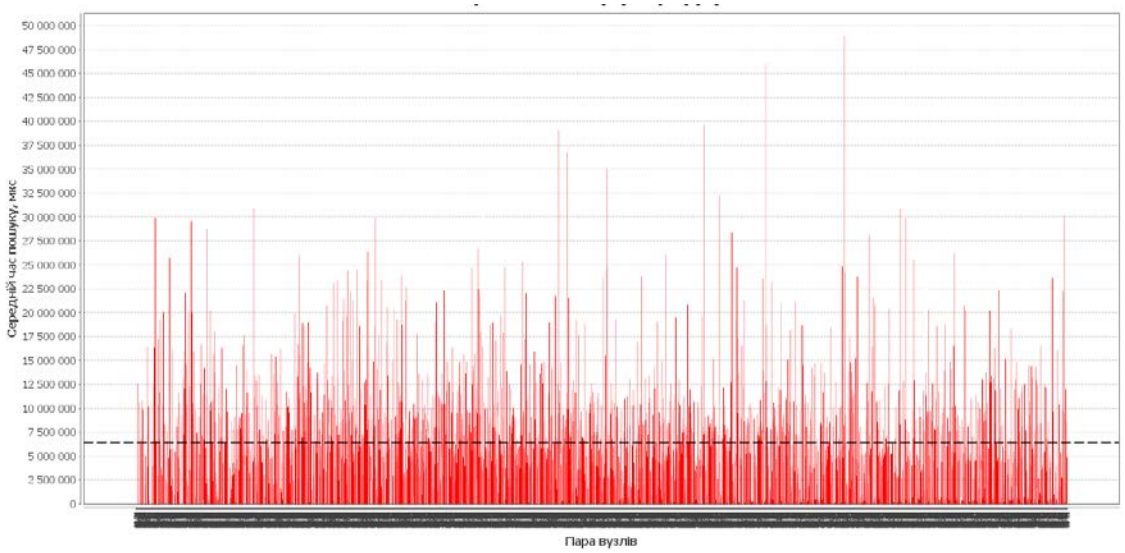
Рис. 3.2. Характеристики навантаження у досліджуваній мережі

Для того, щоб оцінити ефективність запропонованого алгоритму, скористаємося комплексним критерієм ефективності для порівняння: тривалістю пошуку маршутів, споживанням енергії в мережі і життєвим циклом мережі. Якщо говорити про загальну ефективність кластеризації, то необхідно зазначити, що чим більша кількість вузлів в кластері – тим більша тривалість пошуку маршруту, оскільки для розрахунку метрики необхідно враховувати не лише відстань між кожною окремою парою вузлів, але й тривалість обробки та передавання кожним окремо взятим вузлом.

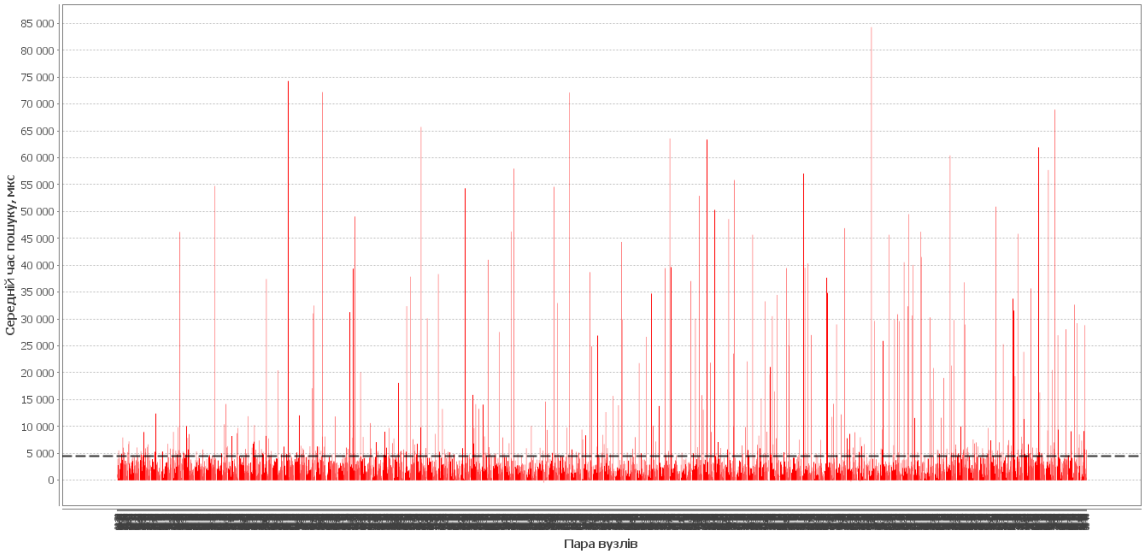
На рис. 3.3 представлено оцінку тривалості пошуку маршруту у досліджуваній мережі з використанням удосконаленого методу. Оцінка

тривалості пошуку маршруту у досліджуваній мережі без кластеризації між довільно взятою парою вузлів показала, що середній час пошуку маршруту займає близько 6500 нс, що є досить тривалим та збільшує час передачі повідомлень кінцевому користувачу (вузлу призначення) (рис. 3.3 а).

Після ввімкнення удосконаленого методу кластеризації спостерігається значне покращення часових характеристик роботи мережі.



а)



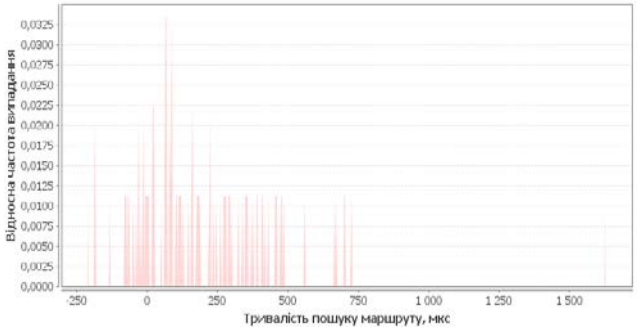
б)

Рис. 3.3.Оцінка тривалості пошуку маршруту у досліджуваній мережі між довільно взятою парою вузлів а) без кластеризації та із використанням удосконаленого методу кластеризації б)

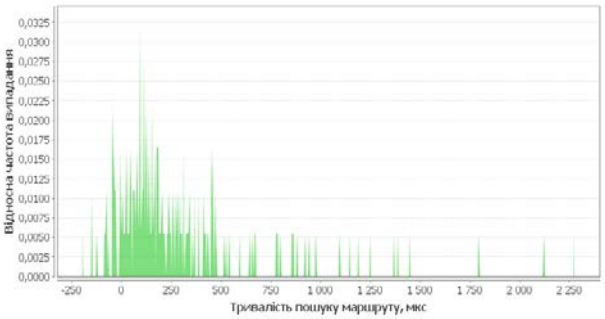
Оцінка тривалості пошуку маршруту у досліджуваній мережі з пропонованим алгоритмом кластеризації між довільно взятою парою вузлів показала, що середній час пошуку маршруту зменшився у 1,3 рази і становить

5000 нс (рис 3.3б), що підтверджує ефективність запропонованого алгоритму кластеризації.

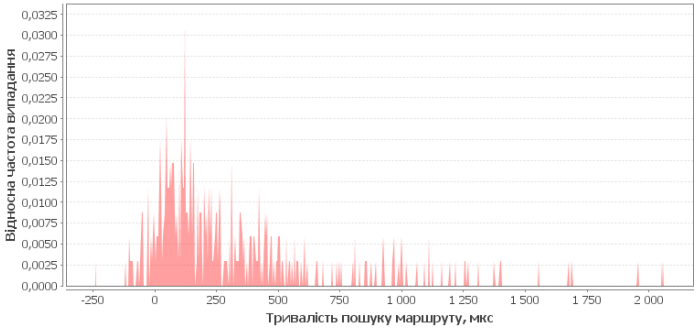
Якщо окремо розглядати ефективність кластеризації, то слід зазначити, що чим більша кількість вузлів – тим більше триває пошук маршруту, оскільки для розрахунку метрики необхідно враховувати тривалість обробки та передавання кожним окремо взятим вузлом, а також відстань між кожною окремою парою вузлів. Для дослідження ефективності кластеризації розглянемо випадок, коли у мережі в межах кластеру функціонують від 4 до 20 вузлів з такими ж параметрами. Приймемо, що таких кластерів у нас є 5. Із залежностей на рис. 3.4 видно, що при наявності 10 вузлів в мережі тривалість пошуку найкоротшого маршруту між довільно взятою парою вузлів може тривати до 0,75 с, тоді як для 19 вузлів – пошук може займати до 2 с. Це зумовлено нерівномірністю надходження запитів на вузли мережі, низькою ефективністю використання ресурсів каналів зв’язку, високим рівнем бітових помилок.



а)10 вузлів



б)14 вузлів



в) 19 вузлів

Рис. 3.4.Оцінка тривалості пошуку маршруту в залежності від кількості вузлів при стандартному алгоритмі кластеризації

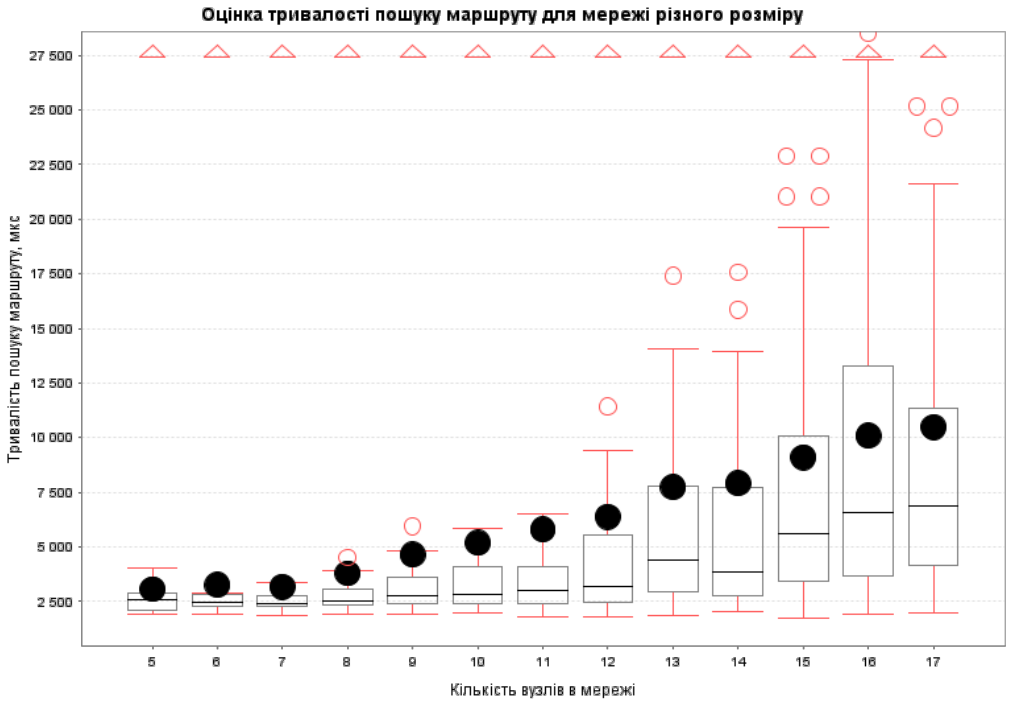


Рис. 3.5. Узагальнена характеристика тривалості пошуку маршруту

Оцінка тривалості пошуку маршруту у досліджуваній мережі без кластеризації між довільно взятою парою вузлів показала, що середній час пошуку маршруту займає до 10500 мкс. Проте із ввімкненням кластеризації тривалість пошуку найкоротшого маршруту між довільно взятою парою вузлів значно зменшується.

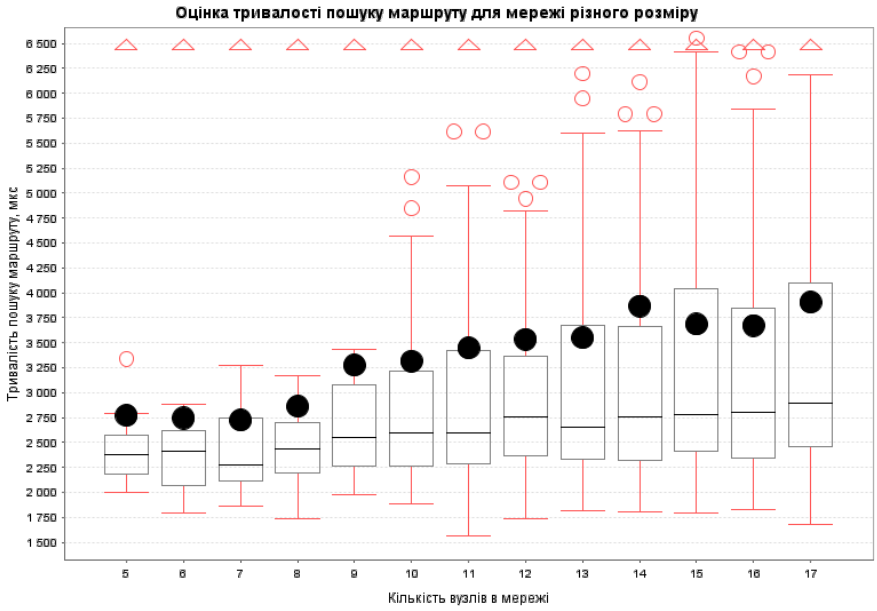


Рис. 3.6. Узагальнена характеристика тривалості пошуку маршруту із використанням удосконаленого методу кластеризації

Як видно із отриманих результатів завдяки визначенню головного вузла кластера із врахуванням центральності по діаграмах Вороного, правил нечіткої логіки та моніторингу параметрів вузлів вдалося досягти зменшення тривалості пошуку маршруту між довільною парою віртуалізованих вузлів, які обслуговують один і той же груповий потік у 2,5 рази.

Для підтвердження ефективності використання удосконаленого алгоритму кластеризації та вибору головного вузла кластеру проведемо порівняльний аналіз із загально відомими алгоритми LEACH та C-means, що є по суті базовими алгоритмами кластеризації, які для створення кластеру використовують метрику відстані між вузлами мережі, та добре відомим алгоритмом з використанням нечіткої логіки CHS-FL-VD. Результати досліджень представлені на рисунках 3.7 та 3.8

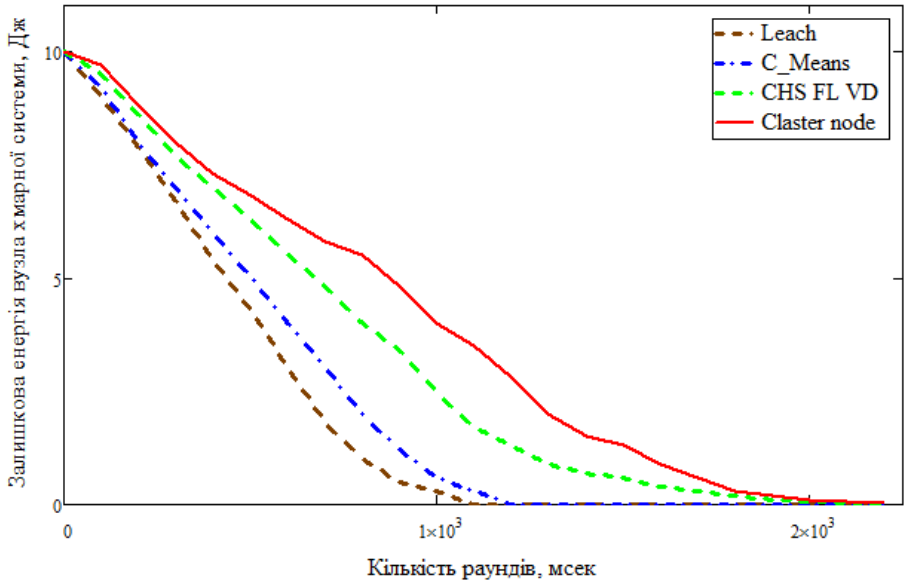


Рис. 3.7. Залежність залишкової енергії вузлів хмарної системи від числа раундів

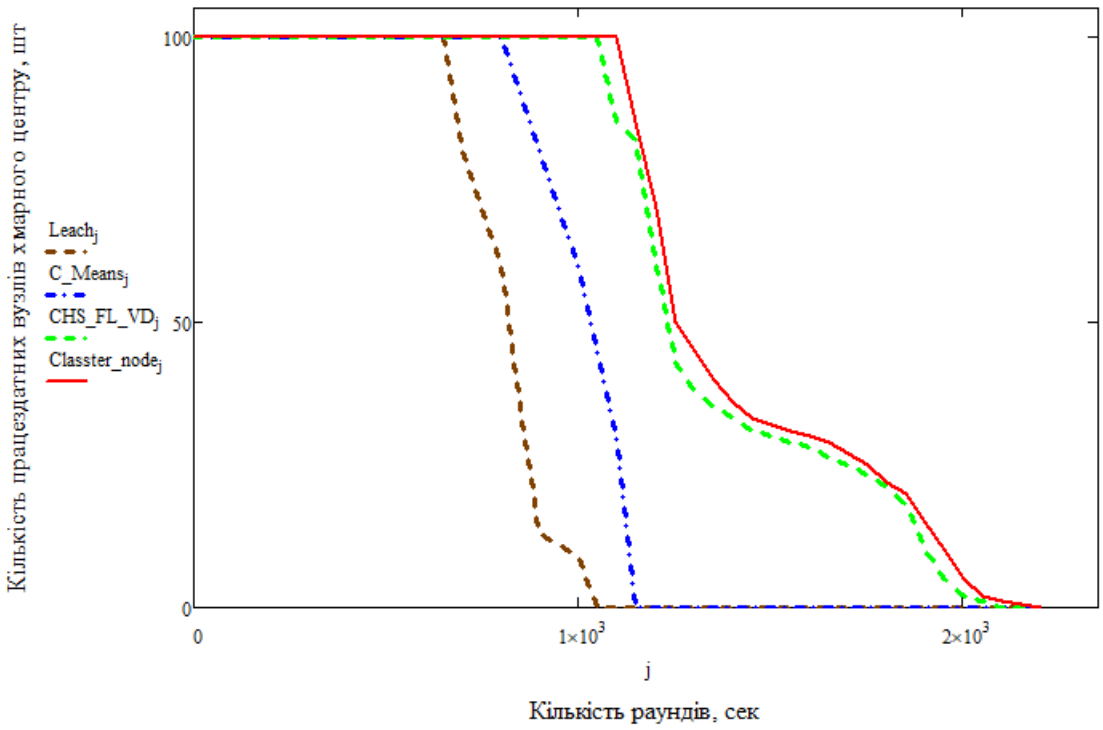


Рис. 3.8. Залежність числа працездатних вузлів хмарної системи від числа раундів

Як бачимо, використання діаграм Вороного в якості міри визначення центральності розташування вузла, нечіткої логіки та моніторингу параметрів вузла дозволило істотно збільшити життєвий цикл мережі порівняно із базовим алгоритмом LEACH, та з алгоритмом CHS-FL-VD, заснованому також на використанні методів нечіткої логіки. Порівняльний аналіз та моделювання показують, що розроблений алгоритм у порівнянні із алгоритмом CHS-FL-VD зменшує споживання енергії на 45% і продовжує життєвий цикл мережі на 8% в порівнянні з відомими алгоритмами, що підвищує і дозволяє підтримувати якість надання послуг користувачам, особливо в умовах групового надходження запитів. По суті, це є першим практичним результатом наших досліджень.

3.2. Практична реалізація моделі сервісно-орієнтованої мережі з використанням CDN серверів як PaaS сервісу.

Дослідження ефективності впровадження запропонованих рішень здійснювалося на основі розгорнутої моделі інфокомунікаційної сервісно-

орієнтованої мережі з використанням CDN серверів як PaaS сервісу та хмарної архітектури Azure Cloud. CDN сервери використовувалися в якості вузлів зберігання контенту і здійснювали віртуалізацію своїх ресурсів при збільшенні попиту на контент, який на них розташовувався. З іншого боку їх основним завданням є зменшення навантаження на сервер, на якому розміщено ядро сервісу, розміщують свої кешсервери по всьому світу. Запит клієнта направляється до кеш сервера, який географічно ближчий до нього та менш завантажений. Дані параметри будуть визначатися на основі удосконаленого методу кластеризації з використанням діаграм Вороного та моніторингу параметрів серверів. Такий метод доступу до контенту дозволить зменшити латентність при завантажені запитуваного контенту.

Мережі доставки контенту(CDN) розміщують свої кеш-сервери по всьому світу, тим самим зменшуючи навантаження на сервер походження. Запит клієнта направляється до кеш сервера, який географічно ближчий до нього та менш завантажений. Такий метод доступу до контенту призводить до зменшення латентності при завантажені запитуваного контенту.

Сьогодні великої популярності та стрімкого розвитку набирають хмарні технології. Популярність хмарних обчислень пов'язана із зниженням витрат, гнучкістю архітектури інформаційних технологій та масштабованістю.

Хмарні обчислення забезпечують середовище для виконання та надання послуг. Вони поєднують між собою набір технологій доступу та систем, які покликані забезпечити необхідні сервіси кінцевим користувачам [73-75]. Все це стає можливим за допомогою технології віртуалізації. Як відомо існують три основних моделі надання послуг у хмарному середовищі – це Інфраструктура в якості сервісу (IaaS), Платформа в якості сервісу (PaaS) та Програмне забезпечення в якості сервісу (SaaS). Саме ефективне функціонування цих моделей забезпечує обробку та виконання великих масивів даних. [76, 77] Однак, важливим аспектом залишається дослідження процесів доступу до

статичного контенту та його завантаження в умовах надання PaaS з мереж з CDN.

Статичний контент - це будь-який контент, який може бути доставлений кінцевому користувачеві без модифікації чи обробки. Сервер передає той самий файл кожному користувачеві, перетворюючи статичний контент на один з найпростіших та найбільш ефективних типів контенту для передачі через Інтернет. Ефективними мережами для передавання такого роду даних стають мережі CDN. Як відомо, мережі даного типу складаються з двох основних компонентів: PoPs або Points of Presence і крайніх серверів. PoPs - це в основному географічне розташування серверів CDN [78, 79]. Ефективне їх функціонування забезпечується шляхом їх рівномірного розподілу по всьому світу. Це важливо для охоплення величезних територій та підтримка рівня якості сервісів. Кожен PoP має свій сервер - Edge Servers - який в основному є простою проксі-кеш-пам'яттю. Він має аналогічну функціональність кешу веб-браузера. Це допомагає зберігати копію контенту в кеш-пам'яті.

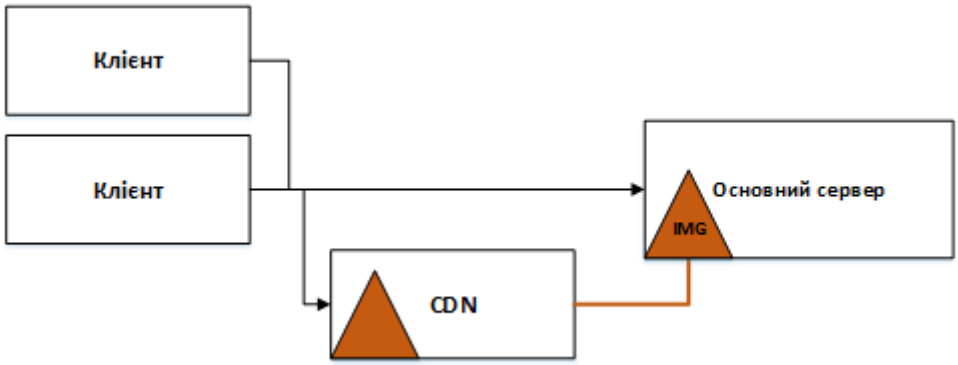


Рис. 3.9. Принцип доступу до контенту у досліджуваній мережі

Розподіл PoP залежить від CDN. Деякі постачальники послуг пропонують декілька, проте надійних, PoPs для своїх користувачів, а деякі - навпаки [78, 79]. Сторінки певного веб-сайту копіюються на сервери, які розташовані в різноманітних місцях, кешуючи контент. Після того, як запит буде ініційований зі сторони користувача, найближчий кеш сервер відповість на даний запит, щоб зменшити час відгуку. CDN діє як середовище, яке передає інформацію з

сервера веб-сайту на найближчий сервер. Це також допомагає передавати будь-який контент, який не є кешований. Наявність CDN дозволяє зберігати статичні дані веб-сторінки поблизу географічного розташування користувача. Коли вони розподіляються однаково по всьому світу, доступність стає бездоганною і оперативною.

Для дослідження ефективності доставки статичного контенту з використанням CDN мереж як PaaS сервісу використаємо сервіси Azure Cloud. Azure Cloud Services є прикладом платформи як сервісу (PaaS) [80]. Подібно до програми Azure App, ця технологія призначена для підтримки програм, які є масштабованими, надійними та недорогими для роботи. Azure Cloud Services точно так само, як служба додатків (App Service) розміщується на віртуальних машинах (VM). Можна встановити власне програмне забезпечення на віртуальні машини, які використовують Azure Cloud Services, і можна отримати доступ до них віддалено.

Коли доставка контенту забезпечується для великої аудиторії користувачів, стає надзвичайно важливо забезпечити оптимізацію доставки цього контенту. Мережа доставки контенту Azure (CDN Azure) - це глобальне рішення CDN для передачі високошвидкісного контенту. Його можна розмістити в Azure або будь-якому іншому місці. За допомогою Azure CDN можна кешувати статичні об'єкти, завантажені з накопичувача Azure Blob, веб-програми або будь-якого загальнодоступного веб-сервера, використовуючи найближчий сервер присутності (POP). Azure CDN також може прискорити передавання динамічного вмісту будь якого сервісу, який не можна кешувати, використовуючи різні способи оптимізації мережі та маршрутизації [81-83].

Мережа доставки Content Azure (CDN) включає чотири продукти: Azure CDN Standard від Microsoft, Azure CDN Standard від Akamai, Azure CDN Standard від Verizon і Azure CDN Premium від Verizon.

У наведеній нижче таблиці порівнюються функції, доступні для кожного продукту.

Таблиця 3.1.

Порівняння продуктів CDN Azure

	Standart Microsoft	Standart Akamai	Standart Verizon	Premium Verizon
Функції продуктивності та оптимізації				
Динамічне прискорення сайту		+	+	+
Динамічне прискорення сайту – адаптивне стиснення зображення		+		
Динамічне розгортання сайту – попередній вибір об'єкта		+		
Оптимізація потокового відео	*	+	*	*
Оптимізація великих файлів	*	+	*	*
Глобальне балансування навантаження сервера (GSLB)	+	+	+	+
Швидка очистка	+	+	+	+
Попереднє завантаження активів			+	+
Параметри кешу/заголовка (використовуючи правила кешування)		+	+	
Кешування строкового запиту	+	+	+	+
IPv4/IPv6 двоканальний	+	+	+	+
HTTP/2 підтримка	+	+	+	+
Безпека				
Підтримка HTTPS за допомогою кінцевої точки CDN	+	+	+	+
Користувацький домен HTTPS	+		+	+
Підтримка користувацького доменного імені	+	+	+	+
Геофільтрація	+	+	+	+
Автентифіка-ція токен				+
Захист DDOS	+	+	+	+
SSL сертифікація (власний сертифікат)	+			
Аналітика та звітність				
Діагностичні журнали Azure	+	+	+	+
Основні звіти Verizon			+	+
Користуваць-кі звіти від Verizon			+	+
Розширені HTTP-звіти				+
Статистика в реальному часі				+
Ефективність кінцевого вузла				+
Сповіщення в реальному часі				+
Простота використання				
Легка інтеграція зі службами, такими як Storage, WebApp та Служби медіа	+	+	+	+
Управління через REST API, NET, Node,js PowerShell	+	+	+	+
Переадресація/переписування URL				+

Для моніторингу параметрів трафіку використано диспетчер трафіку Azure, який підтримує шість методів маршрутизації трафіку, які визначають правила маршрутизації мережевого трафіку в різні кінцеві точки служби. Для

будь-якого профілю диспетчер трафіку застосовує пов'язаний з ним метод маршрутизації трафіку до кожного отриманого запиту DNS.

При проведенні дослідження ми зупинилися на методі маршрутизації трафіку за пріоритетом та дослідили час доставки контенту користувачеві з використанням запропонованих моделей та алгоритмів та з використанням CDN серверів, які використовуються в якості вузлів зберігання контенту і здійснюють віртуалізацію своїх ресурсів при збільшенні попиту на контент, який на них розташовувався. Запит клієнта направляється до кеш сервера, який географічно ближчий до нього та менш завантажений. Дані параметри будуть визначатися на основі удосконаленого методу кластеризації з використанням діаграм Вороного та моніторингу параметрів серверів.

Як правило, організація прагне забезпечити надійність своїх служб, розгортаючи одну або кілька резервних служб на випадок, якщо основна служба вийде з ладу. Метод маршрутизації трафіку "За пріоритетом" дозволяє клієнтам Azure легко реалізувати цю схему відпрацювання відмови. В цьому профілі диспетчера трафіку створюється впорядкований список кінцевих точок служби. За замовчуванням диспетчер трафіку направляє весь трафік в первинну кінцеву точку (з найвищим пріоритетом). Якщо основна кінцева точка недоступна, диспетчер трафіку направляє трафік на другу кінцеву точку. Якщо і первинна, і вторинна кінцеві точки стають недоступні, трафік прямує до третьої і т.д. Доступність кінцевої точки визначається за вказаним для неї станом(включена або відключена) і даними моніторингу кінцевих точок.

На рис. 3.10 зображено перелік кеш-серверів (кінцевих точок), які додано до диспетчера трафіку PelehTM. PrimaryEndpoint розташований у західній Європі та має пріоритет 1. BackupEndpoint розташований у північній Європі та має пріоритетність 2.

Для того, щоб розпочати роботу із Azure CDN потрібно оформити підписку Azure. Для даного дослідження використовувалась безкоштовно

підписка від Azure, яка надається терміном на один календарний місяць та має кредит у 100$.

Обліковий запис (Storage account, сховище) для зберігання даних дає доступ до послуг Azure Storage. Обліковий запис для зберігання являє собою найвищий рівень простору імен для доступу до кожного компонента сервісу Azure Storage: зберігання Azure Blob, Queue і Table. Для початку необхідно створити контейнер "images", де буде міститись досліджуваний контент. Для дослідження в якості контенту до контейнера завантажено зображення з іменем "брошура_фр.jpg".

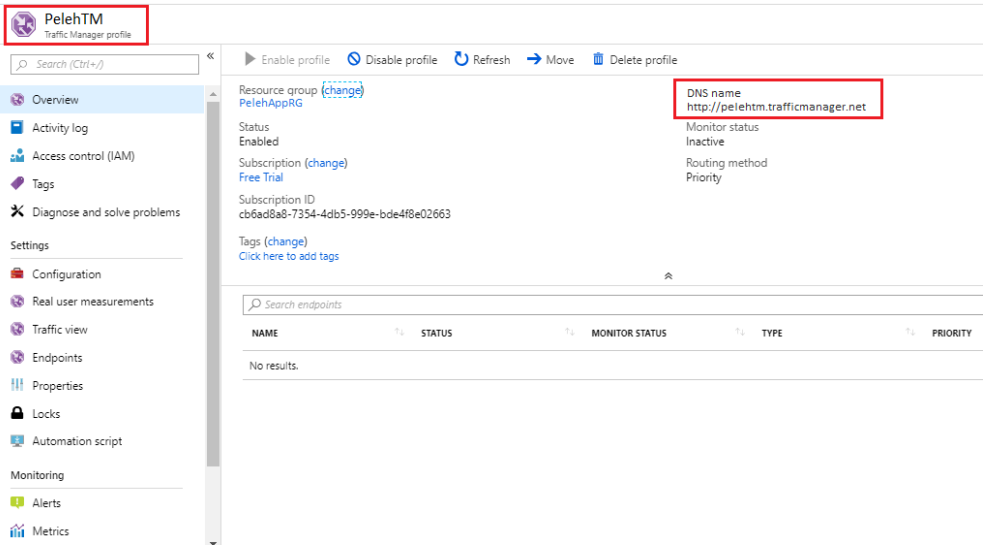


Рис. 3.10.Вигляд створеного диспетчера трафіку

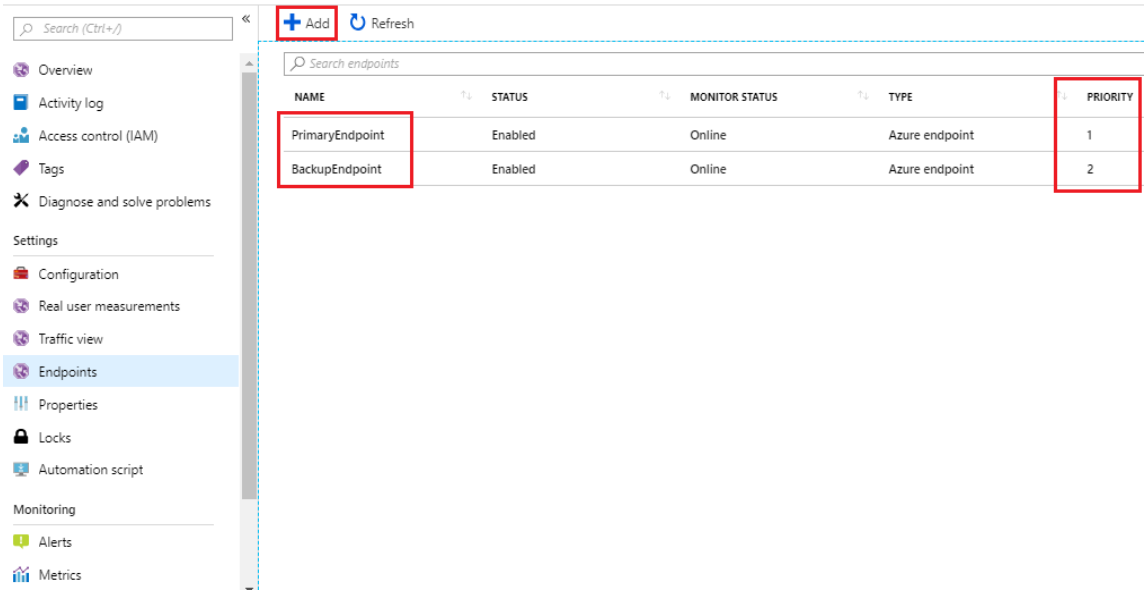


Рис. 3.11.Додавання кеш-серверів в диспетчер трафіку з пріоритетами

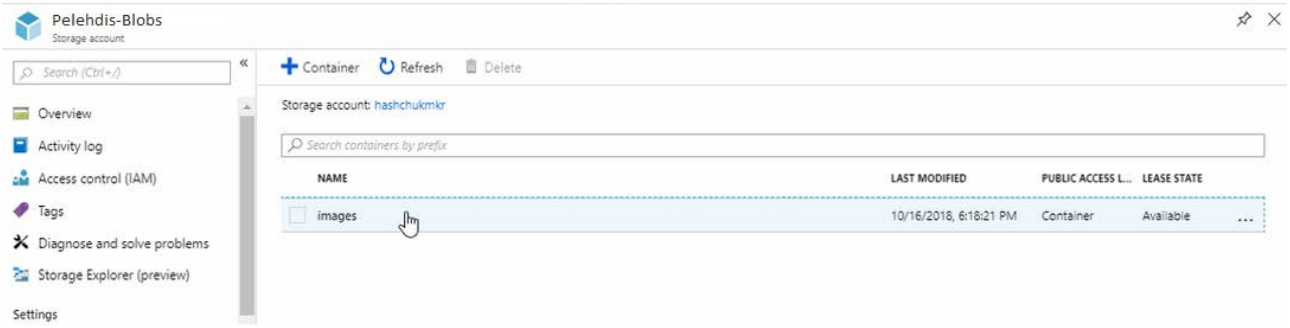


Рис. 3.12. Контейнер для досліджуваного контенту

Після завантаження контенту до контейнера, він буде відображатись як показано на рис.3.11.

Профіль CDN являє собою контейнер для кінцевих точок CDN та визначає рівень ціноутворення. При створені профілю CDN потрібно ввести унікальне ім'я, яке в даному випадку "PelehCDN". Місце розташування – "Західна Європа". Продукт/ціновий рівень – Standart Akamai. Готовий профіль CDN зображено на рис.3.13.

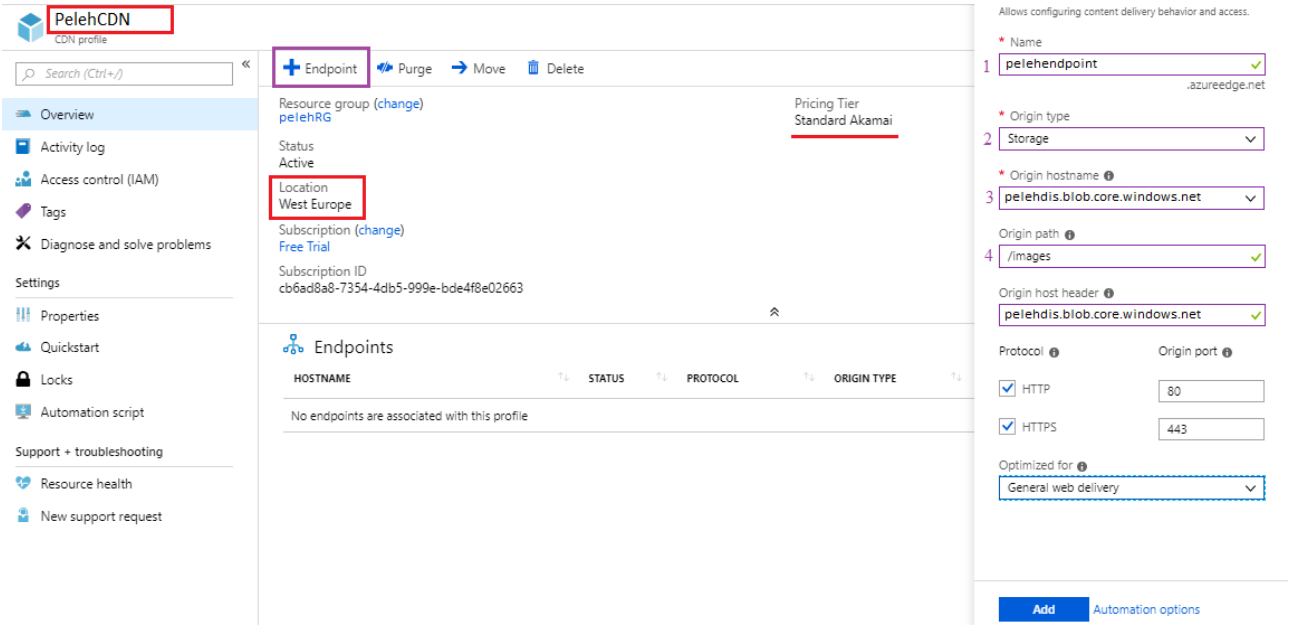


Рис. 3.13. Вигляд створеного CDN профілю та процес створення Endpoint

Створений endpoint під назвою "pelehendpoint" розташований в тій же зоні, що і профіль CDN – Західній Європі. Посилання в полі "Endpoint hostname" - це посилання на досліджуване зображення, яке раніше завантажено в storage account в контейнер "images", але доступ до нього надається з даного кеш-

серверу. Тобто pelehendpoint містить кешовану версію дослідженого зображення.

3.2.1. Завантаження та доступ до контенту при різних умовах та географічному розташуванні

Доступ до кешованого контенту відбувався з м.Львів через кеш-сервери західної Європи, центральної Америки, центральної Індії та східної Японії. Також виміряно час завантаження при доступі до оригіналу зображення, яке знаходиться в Storage account (базовому сервері), який розташований в східній Австралії.

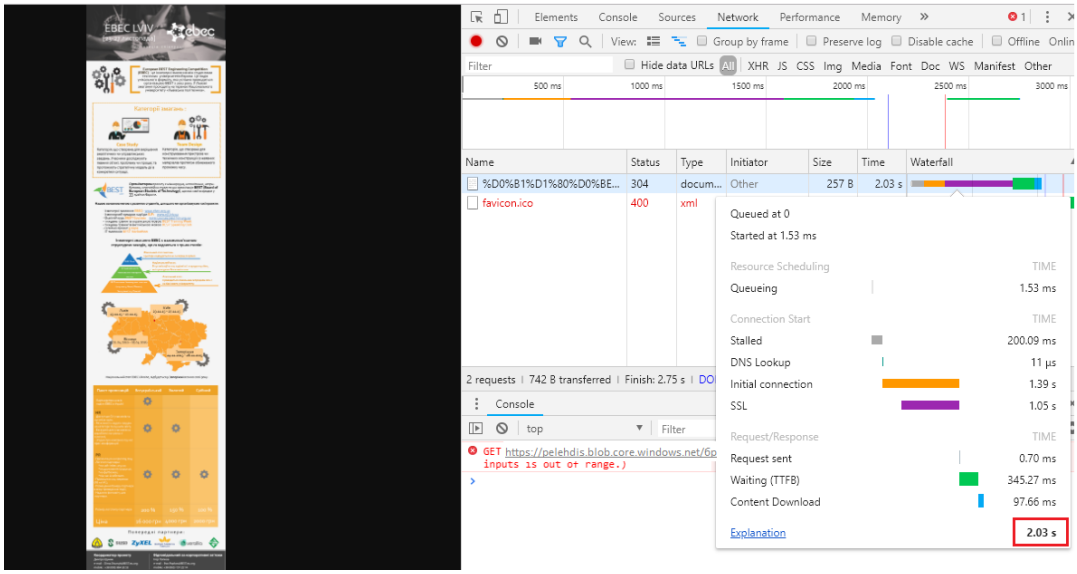


Рис. 3.14. Завантаження напряму з серверу розташованого в Східній Австралії

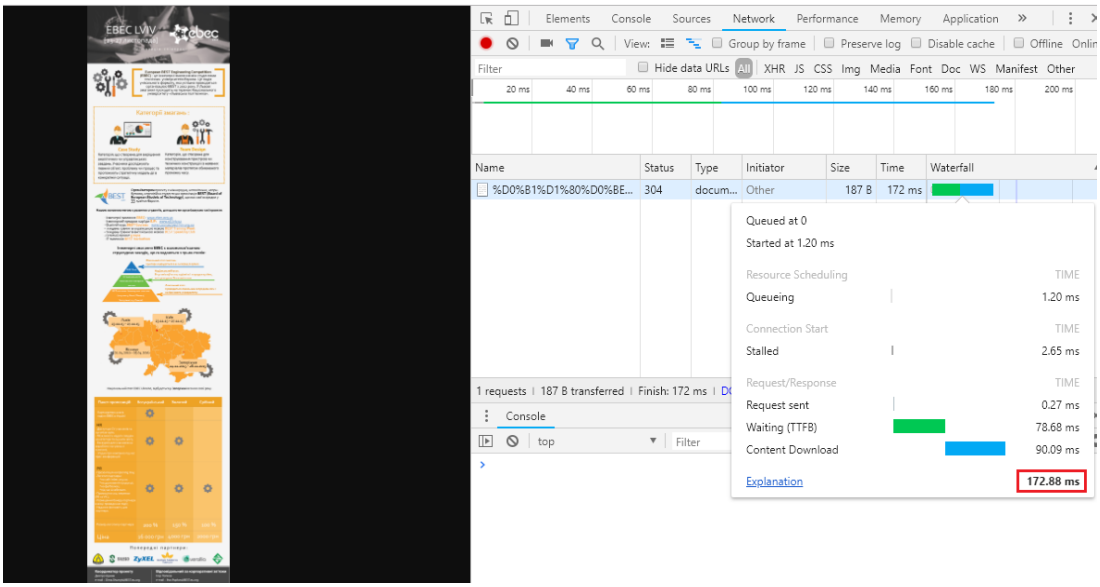


Рис. 3.15. Завантаження через кеш-сервер Західної Європи

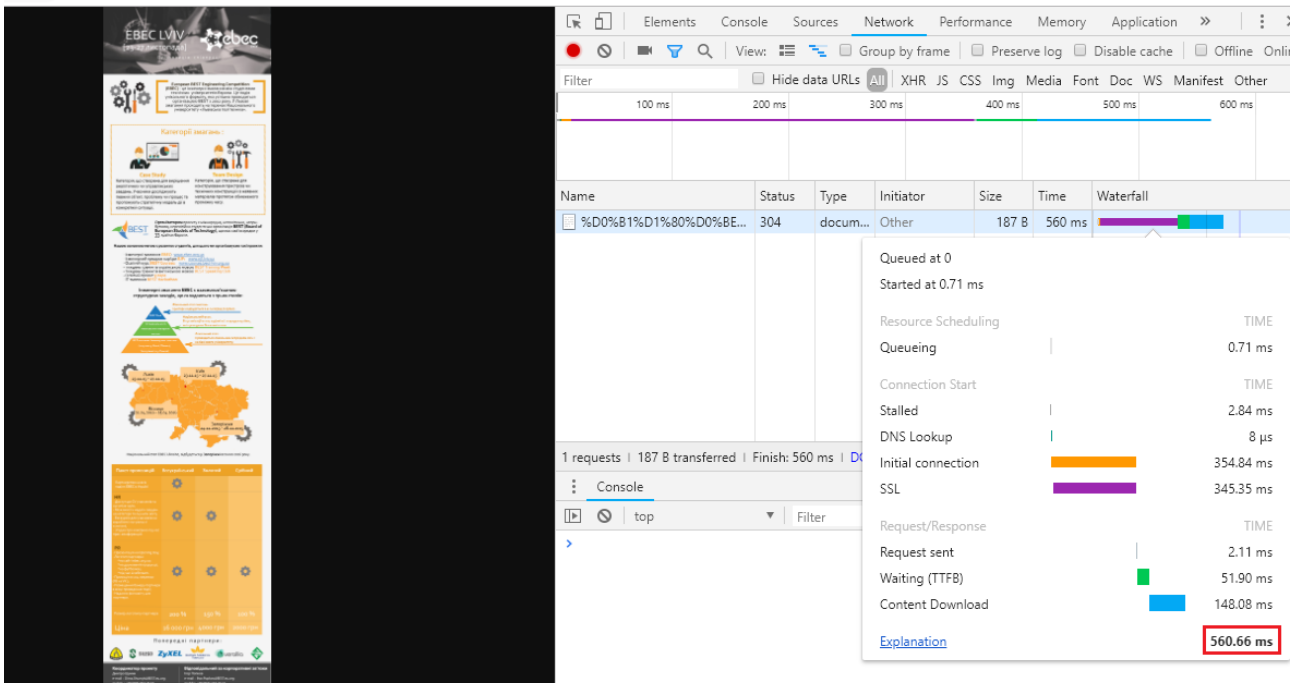


Рис. 3.16. Завантаження через кеш-сервер Центральної Америки

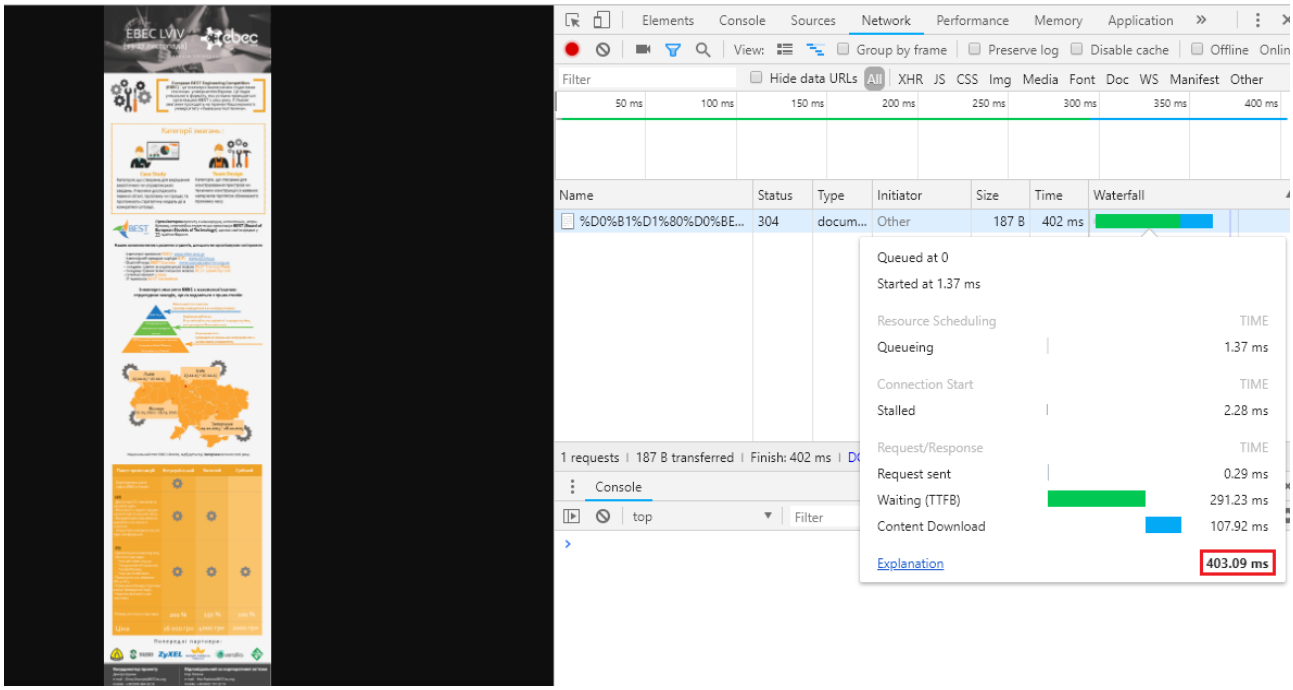


Рис. 3.17. Завантаження через кеш-сервер Центральної Індії

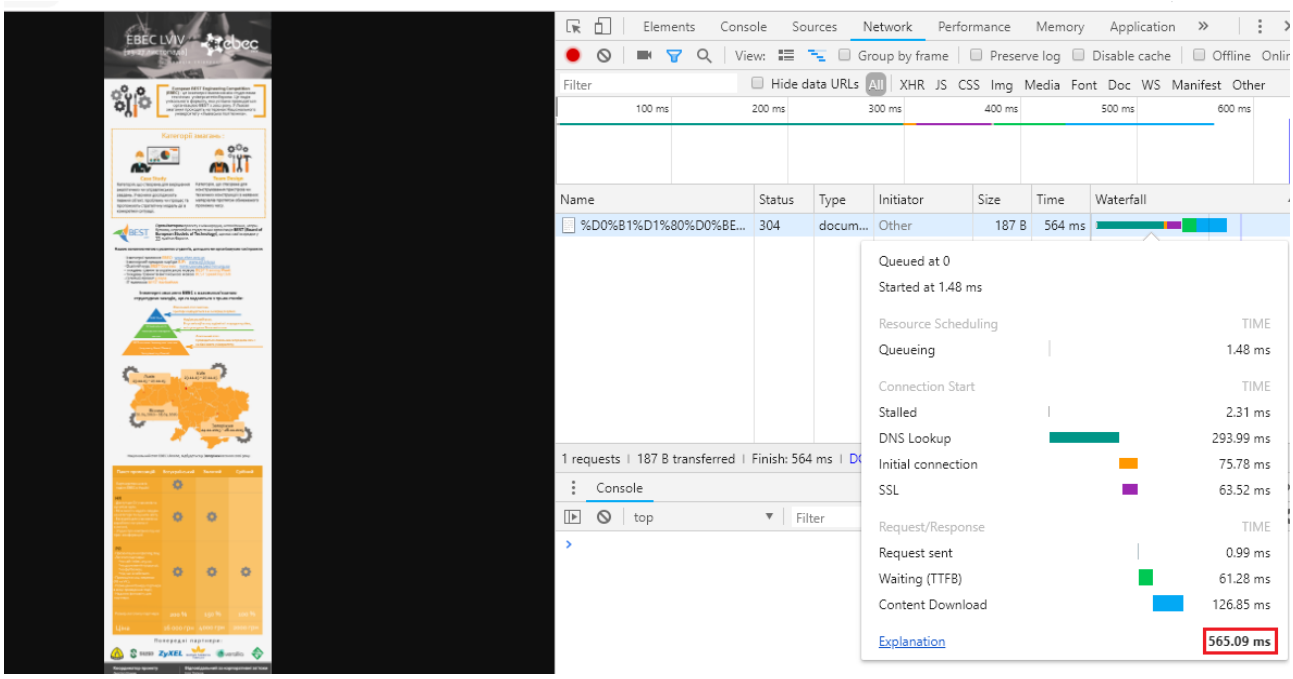


Рис. 3.18. Завантаження через кеш-сервер Східної Японії

Таблиця 3.2.

Час завантаження зображення при різних умовах та географічному розташуванні

Дата/час завантаження	Місто/країна з якої йде доступ	Blob Storage	West Europe	East Japan	Central US	Central India
19.08.2019 18:15	Львів, Україна	2.03 с	172.88 мс	565.09 мс	560.66 мс	403.09 мс
21.08.2019 17:30	Львів, Україна	2.21 с	114.28 мс	724.67 мс	1.05 с	742.13 мс
21.08.2019 19:47	Брно, Чехія	2.56 с	50.52 мс	1.598 с	1.082 с	1.603 с
22.08.2019 22:55	Токіо, Японія	2.18 с	67.588 мс	202.11 мс	517.11 мс	141.91 мс

Результати дослідження показали, що завантаження відбувалось напряму з сервера походження за 2.03с, з кеш-серверу Західної Європи завантаження відбувалось за 172.88 мс, з кеш-серверу Цетральної Америки – за 560.66 мс, з кеш-серверу Центральної Індії час завантаження становив 403.09мс, завантаження з кеш-серверу Східної Японії –565.09мс.

Завдяки допомозі людей з інших країн, таких як Чехія та Японія, з'явилась можливість дослідити час завантаження з різних кеш-серверів, доступаючись

до контенту з вищезгаданих країн. Всі результати дослідження зібрано в таблиці 3.2.

Аналізуючи результати дослідження можна легко побачити, що завантаження з кеш-серверів відбувалось набагато швидше, ніж з базового сервера, час завантаження контенту зменшився орієнтовно в 4 рази.

При доставці контенту до клієнтів, функціонал статистики в реальному часі дозволяє визначити необхідні ресурси, які затрачаються на передавання чи оригінального чи кешованого контенту: пропускна здатність, коди статусу, статуси кеш-пам'яті та паралельні підключення до профілю CDN. Це дає змогу постійно стежити за працездатністю сервісу в будь-який час, включаючи події, що відбуваються в реальному часі.

В результаті досліджень проведено моніторинг поведінки Azure CDN. Дана функція доступна тільки при використані продукту Verizon Premium, тому спеціально для цього дослідження створено окремий профіль CDN та кеш-сервер у Західній Європі.

В результаті досліджень проведено аналіз використання пропускної здатності каналу, який забезпечує доставку контенту. Як видно з рис. 3.19 використання пропускної здатності каналу при доступі до контенту із віддалених серверів зростає до 700 Кбіт/с.

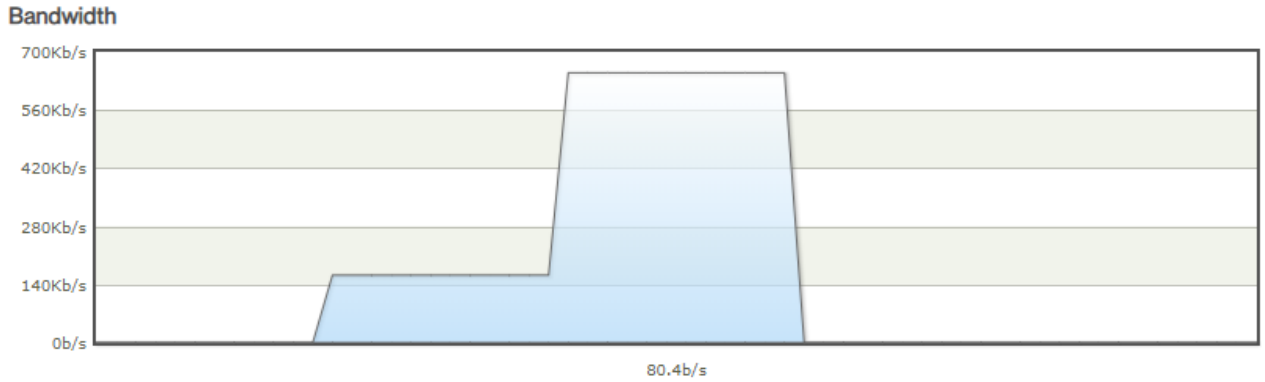


Рис. 3.19.Розподіл пропускної здатності при доступі до статичного контенту

Розподіл кодів статусу показує, як часто певні коди відповіді HTTP надходять за вибраний проміжок часу. Як видно з рис. 3.20 HTTP відповіді

надходять через однакові інтервали часу, і не залежать від території, з якої надсилаються запит на доступ до даних. Cache Statuses вказує, як часто певні типи статусів кешу поширюються протягом вибраного проміжку часу.

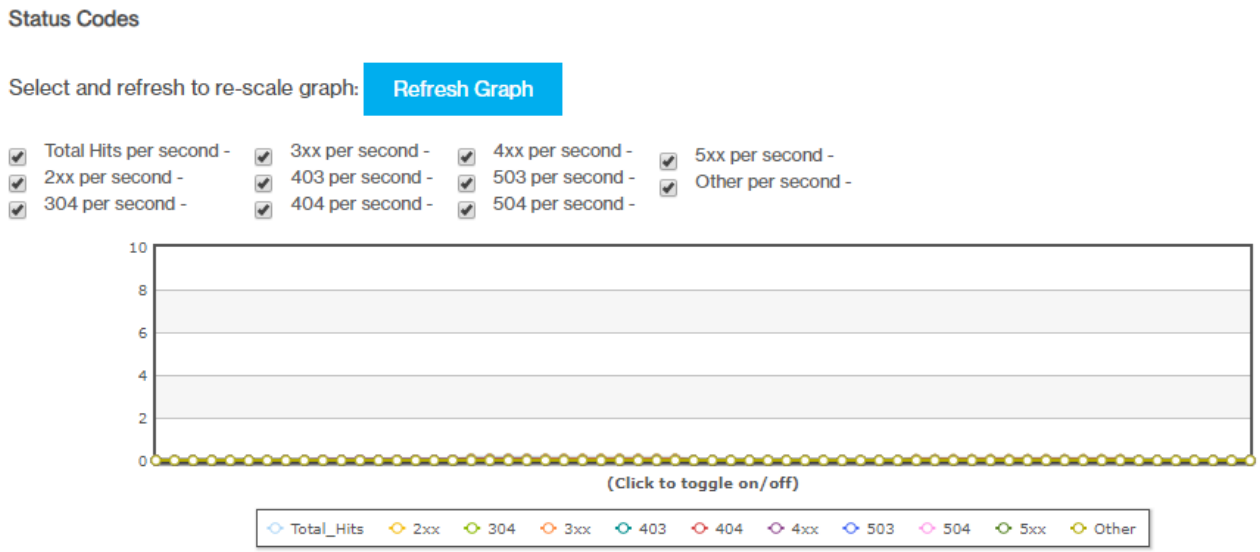


Рис. 3.20. Розподіл кодів статусу протягом проведення дослідження

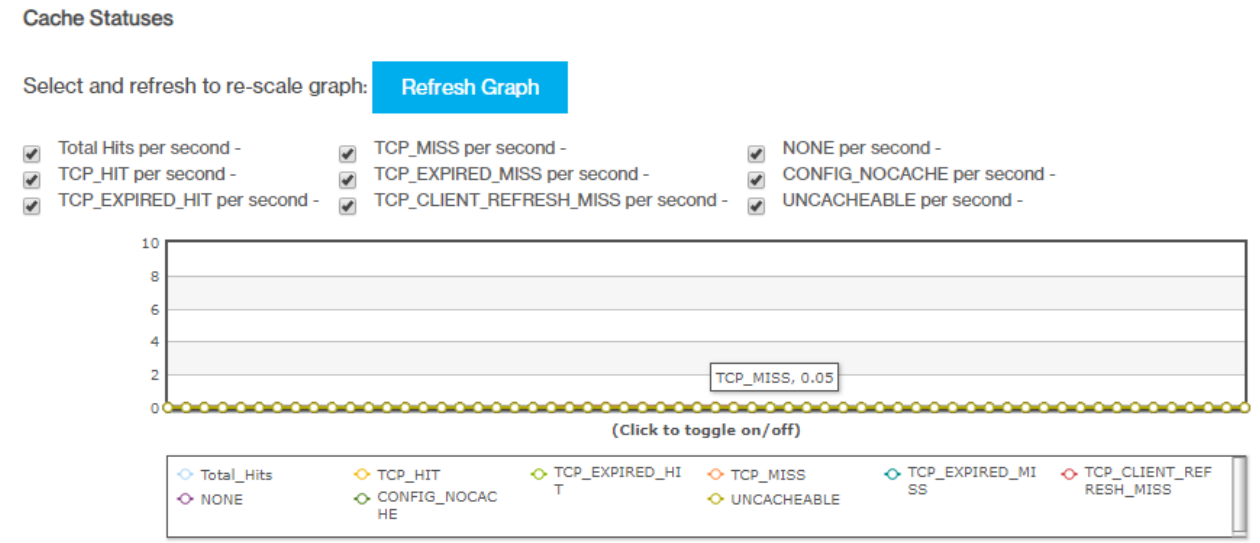


Рис. 3.21. Статус кешу

Дослідження показали, що контент не повинен кешуватись на POP або HTTP-клієнтом. Лише в одному випадку на 0,05 с. кешована версія запитуваного ресурсу не була знайдена на найближчому до клієнта POP. Контент буде запитано як з базового сервера, так і від базового сервера. Якщо

базовий сервер або щит базового сервера повертає active, він буде передаватися клієнту і кешуватись як на клієнтській стороні, так і на сервері.

Висновки до 3-го розділу

Для проведення дослідження стосовно ефективності запропонованого методу кластеризації вузлів хмарних систем із застосуванням правил нечіткої логіки та моніторингу параметрів вузлів у роботі розроблено імітаційну модель на основі програмного пакету Fuzzy Logic Toolbox.

В процесі моделювання доведено ефективність запропонованого алгоритму кластеризації на основі комплексного критерію ефективності: тривалості пошуку маршутів, споживання енергії в мережі і життєвий цикл мережі. Завдяки реалізації розробленого методу кластеризації вузлів хмарних систем вдалося досягти зменшення тривалості пошуку маршруту між довільною парою віртуалізованих вузлів в межах одного кластеру у 1,3 рази. Завдяки визначенню головного вузла кластера із врахуванням центральності по діаграмах Вороного, правил нечіткої логіки та моніторингу параметрів вузлів вдалося досягти зменшення тривалості пошуку маршруту між довільною парою віртуалізованих вузлів, які обслуговують один і той же груповий потік у 2,5 рази.

Для підтвердження ефективності використання удосконаленого алгоритму кластеризації та вибору головного вузла кластеру проведено порівняльний аналіз із загально відомими алгоритми LEACH та C-means, та добре відомим алгоритмом з використанням нечіткої логіки CHS-FL-VD з точки зору споживання енергії і працездатності вузлів мережі. Порівняльний аналіз та моделювання показують, що розроблений алгоритм у порівнянні із CHS-FL-VD зменшує споживання енергії на 45% і продовжує життєвий цикл мережі на 8% в порівнянні з відомими алгоритмами. Даний метод дозволяє підтримувати якість надання послуг користувачам, особливо в умовах групового надходження запитів.

Для доведення ефективності запропонованих методів та моделей підвищення ефективності функціонування хмарних систем проведено дослідження на основі розгорнутої моделі інформаційно-комунікаційної сервісно-орієнтованої мережі з використанням CDN серверів як PaaS сервісу та хмарної архітектури Azure Cloud. Реалізація удосконаленої математичної моделі оцінки ефективності функціонування хмарних систем в умовах групового надходження запитів, яка, на відмінну від існуючих, враховує можливість обслуговування при груповому надходженні запитів, розподілений час обслуговування запитів та методу кластеризації вузлів хмарних систем дала змогу зменшити тривалість завантаження статичного контенту з кеш-серверів (які виступали головними вузлами кластерів, визначеними з використанням алгоритму вибору головного кластерного вузла з використанням центральності по діаграмах Вороного, правил нечіткої логіки та моніторингу параметрів вузлів) у порівняні із завантаженням з базового сервера в 4 рази. В результаті досліджень проаналізовано ефективність використання пропускної здатності каналу, який забезпечує доставку контенту. Використання пропускної здатності каналу при доступі до контенту із віддалених серверів зростає до 700 Кбіт/с. Дослідження показали, що контент не повинен кешуватись на POP або HTTP-клієнтом. Лише в одному випадку на 0,05 с. кешована версія запитуваного ресурсу не була знайдена на найближчому до клієнта POP.

РОЗДІЛ 4. РЕАЛІЗАЦІЯ МЕРЕЖЕВОГО ЗАХИСТУ ХМАРНИХ ВЕБ СЕРВІСІВ У ПРОГРАМНО-КОНФІГУРОВАНИХ СЕРВІСНО-ОРІЄНТОВАНИХ МЕРЕЖАХ

4.1. Формування інтегрованої архітектури інтелектуального виявлення атак у SDN мережах із використання машинного навчання

Управління будь якою мережевою інфраструктурою зводиться до проблеми підвищення захищеності даних від несанкціонованого доступу до них. Втрата даних або несанкціоноване їх використання для компаній можуть призвести до значних фінансових затрат. Для цього необхідні певні механізми моніторингу стану кожної системи, яка буде аналізувати можливі проблеми, що можуть виникати під час роботи, і запускати системи захисту в самому сервісі або повідомляти адміністраторів про можливу загрозу. Такі системи повинні в дуже короткий час відслідковувати всі зміни, які відбуваються із системою - різноманітні ін’єкції, атаки, технічні збої, при яких сервіс перестає нормально функціонувати.

Одним із ефективних засобів відслідковування і запобігання такого роду проблемам є контроль за часом встановлення сесії кожним користувачем та аналіз і управління журналами службової інформації. Такого роду аналіз повинен проводитися на рівні ядра мережі за рахунок логування всіх запитів на етапі, як тільки вхідний пакет пройшов фаєрвол мережі і перенаправляється у Black Hole.

На сьогоднішній день важко уявити систему, яка працює з веб додатками без програмного управління. Передавання, управління та збір статистичної інформації здійснюється SDN контролером шляхом пошуку записів у групових таблицях потоків від одного або/до декількох інтерфейсів чи додатків. Коли мова іде про додатки, Контролер здійснює перенаправлення потоків запитів до необхідної аплікації. В такому випадку саме Контролер буде відповідальним за безпеку цих додатків [84].

Сучасні веб системи повинні протидіяти різним загрозам зі сторони зловмисників. Але для цього необхідно знати, які типи загроз можуть спричинити технічні проблеми на стороні серверу, де компанія буде втрачати кошти кожну секунду, чи більш поганий варіант – доступ до приватних даних користувачів. Тож, є кілька типів загроз веб сервісу: ін’єкції, слабка авторизація, Dos/Ddos атаки, Man-in-the-Middle атака (MitM)

Ін'єкційні атаки відносяться до широкого класу векторів атак, які дозволяють зловмиснику подавати ненадійний вхід до програми, яка обробляється інтерпретатором, як частина команди або запиту, що змінює хід виконання цієї програми. Ін'єкційні атаки є одними з найдавніших та найнебезпечніших атак веб-додатків. Вони можуть спричинити крадіжку даних, втрату даних, втрату цілісності даних, відмову в обслуговуванні, а також повний системний компроміс.

Ін'єкція є головною проблемою веб-безпеки. Він занесений як ризик безпеки веб-додатків номер один у Топ-10 OWASP [85]. Ін'єкційні атаки, зокрема інжекція SQL (SQLi) та міжсайтовий сценарій (XSS), є не лише дуже небезпечними, але й дуже широко поширені, особливо у застарілих програмах. Те, що робить ін'єкційні атаки особливо небезпечними, це те, що вони можуть буди виконані в дуже великій кількості сценаріїв (особливо для SQLi та XSS).

DDOS-атаки - розподілені атаки, спрямовані на відмову в обслуговуванні, продовжують залишатися однією з найважливіших загроз в мережі. Атаки такого типу можуть швидко виснажити мережеві ресурси або потужності сервера, що призведе до неможливості отримати доступ до ресурсу і викличе серію негативних наслідків: втрачений прибуток, неможливість скористатися послугами і зробити різні транзакції і т.д. На даний момент не існує якогось універсального засобу для протидії DDOS-атакам. Для протидії розподіленим атакам, спрямованим на відмову в обслуговуванні, слід дотримуватися двох основних завдань:

1. Діагностувати DDOS-атаку на якомога ранніх стадіях. Чим раніше буде виявлена DDOS-атака, тим раніше зможе включитися в гру мережевий адміністратор і тим раніше можна буде почати проводити анти DDOS-заходи. Крім того, при виявленні DDOS-атаки можна буде, не чекаючи реагування адміністратора, автоматично запустити заходи з протидії: задіяти резервні канали зв'язку, включити фільтри і т.д.

2. Друге завдання пов'язана з поділом загального потоку трафіку на шкідливий і звичайний. Зрозумівши, які з клієнтських запитів є результатом DDOS-атаки, можна буде створити відповідні правила для брандмауера або ACL правила для маршрутизатора або ж, у разі масштабної атаки, передати ці дані на вищі маршрутизатори.

Найчастіше веб додатки піддаються саме DDoS атакам. Виведення з ладу окремих його частин може призвести до втрати працездатності системи та втрати конфіденційної інформації користувачів. З метою підвищення ефективності управління та вирішення проблеми підвищення захищеності даних від несанкціонованого доступу пропонуємо інтегровану архітектуру системи інтелектуального виявлення DDoS атак у інформаційно-комунікаційних мережах, яка реалізується на основі комплексного підходу до аналізу лог журналів із використанням як структурованого так і неструктурованого аналізу службових файлів. Запропонована архітектура управління представлена на рис 4.1. Особливістю даної архітектури є:

- Підсистема аналізу логів (Log analysis subsystem), яка на основі інформації з таблиць потоків реалізує комплексний аналіз службових файлів (структурований та неструктурований) та на основі інформації про стан потоку, тривалість сесії та джерела його походження проводить розподіл трафіку на аномальний та типовий. Результати аналізу записуються у базу даних Машинного навчання.
- База даних Машинного навчання (Mashine Learning - ML) - накопичення даних, отриманих в результаті як структурного так і неструктурного

аналізу даних, в базі ML дозволить виявляти аномалії при поступленні потоків запитів, на основі аналізу тривалості доступу до сервісу та прописувати правила роботи Контролера. Використання елементу машинного навчання як окремої підсистеми інтегрованої архітектури управління мережею, дасть змогу проводити інтелектуальне виявлення атак використовуючи інформацію про стан потоку, тривалість сесії та джерела його походження, а також навчити SDN контролер блокувати «подібні» потоки запитів без втручання системних адміністраторів.

- SDN контролер – здійснює керування потоками трафіку, які надсилаються від комутаторів чи доменів рівня доступу до площини функціонування веб сервісів (Application plane), завдяки зберіганню інформації про трафік у групових таблицях потоків. Підтримує постійний моніторинг роботи веб додатків та принципів їх розгортання, що використовує в своїй основі віртуалізацію ресурсів.

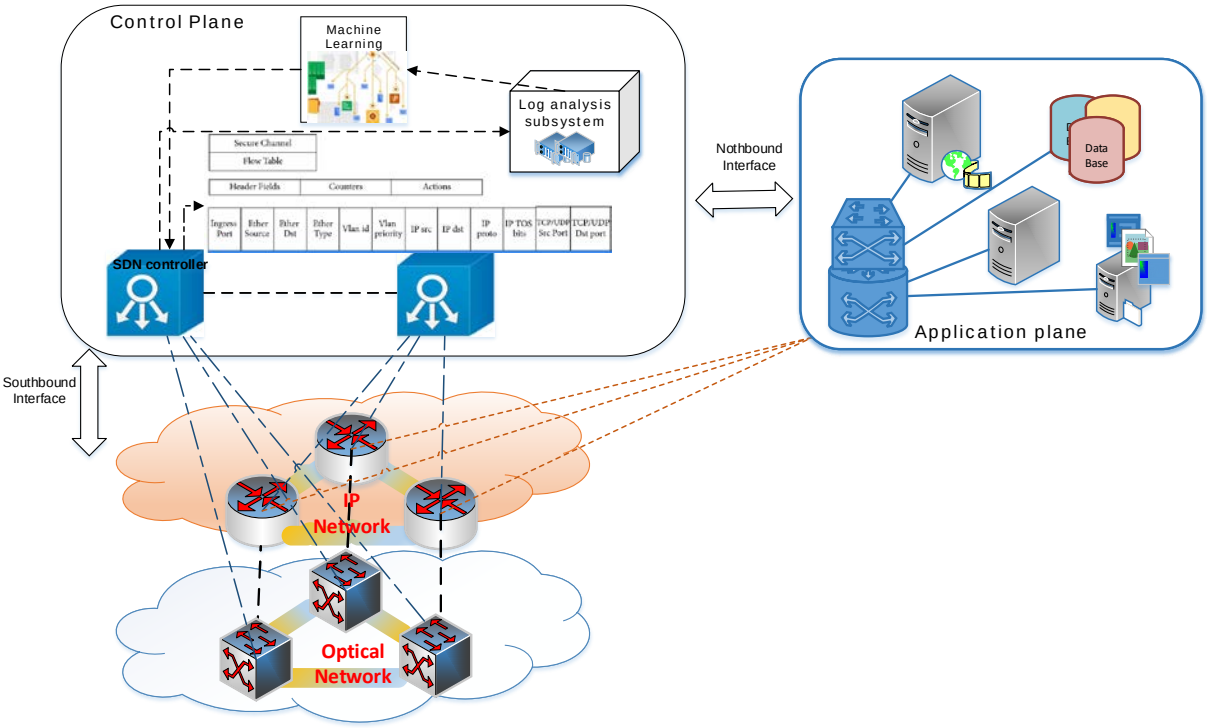


Рис. 4.1.Запропонована архітектура інтелектуального виявлення атак у SDN мережах із використання машинного навчання

- Площина функціонування веб сервісів (Application plane) – здійснює розгортання, зберігання, обробку та функціонування веб сервісів. У дисертаційній роботі не зосереджено увагу на принципі функціонування, особливостях розгортання чи інших властивостях веб сервісів, припускаємо, що дана площина для ефективного надання веб сервісів кінцевим користувачам може розгортатися і на хмарній інфраструктурі.

У запропонованій архітектурі, усі дані та події, створені користувачами записуються контролером або набором контролерів та збираються агентом керування журналами і періодично передаються на платформу керування й аналізу журналів, розгорнуту в хмарі. Той самий агент контролює продуктивність сервера, на якому працюють контролери SDN, і періодично надсилає дані в Підсистему аналізу журналів. Подібно до SDN Control Plane, усі журнали, створені однією машиною або кількома машинами, на яких працюють мережеві компоненти на рівні доступу, збираються підсистемою аналізу журналів і передаються на платформу керування журналами для аналізу в реальному часі через TCP-з'єднання. Адміністратори мережі тепер можуть переглядати, співвідносити та аналізувати мережеві журнали SDN в режимі реального часу або звертатися до історії цих даних. Підсистема аналізу журналів на основі отриманої службової інформації запускає два паралельні незалежні процеси аналізу журналів: структурований і неструктурований, які проводять ідентифікацію трафіку та відокремлюють звичайний та аномальний трафік. Поділ трафіку здійснюється на основі визначення ентропійних властивостей потоків, що дасть змогу виявляти наявність аномалій.

Більшість систем захисту функціонують на рівні ядра мережі. Частина вхідних запитів перенаправляється у Black Hole. Єдиним можливим варіантом підвищення ефективності захисту системи буде логування всіх запитів на етапі проходження фаєрволу. Дослідивши лог файл, можна дізнатися всю потрібну інформацію. Саме тому, в розробленій системі аналізатор буде функціонувати після фаєрволів.

4.2. Комплексний підхід до аналізу лог журналів як спосіб виявлення атак

На сьогоднішній день інформаційні системи стали справжніми мішенями для кіберзлочинців, оскільки від їх функціонування залежить якість нашого повсякденного життя, та робота критично важливої інфраструктури: електромереж, медичних установ, фінансові та транспортні системи. Вони складаються з різних програмних компонентів, які варіюються від програми до програми, балансують між проміжним програмним забезпеченням, базою даних та операційною системою, створюючи при цьому величезну кількість службової інформації, яка поміщається в лог журнали. Саме аналітика цих даних з точки зору реалізації політики безпеки, є новою тенденцією в галузі кібербезпеки та допомагає адміністраторам мереж захищати критично важливі елементи інфраструктури та виявляти підозрілу діяльність чи атаку.

Загроза несанкціонованого доступу впливає на вимоги до мережевої інфраструктури. Поточне управління безпекою в мережах в значній мірі залежить від управління правилами та пріоритетності потоків трафіку. Рішенням щодо запобігання несанкціонованого доступу чи атакам може стати запропонована у дисертаційній роботі Підсистема аналізу лог файлів, яка є елементом інтегрованої архітектури управління, та аналізує потенційні проблеми і запускає запобіжні механізми, або може повідомляти системних адміністраторів про порушення безпеки мережі. Така підсистем повинна в дуже короткий час відслідковувати всі зміни, які відбуваються із системою - як різноманітні ін’єкції, так і атаки, при яких порушується доступ до веб сервісу і він перестає нормально функціонувати.

Для початку пропонуємо вивчити файли лог журналів, створені в результаті діяльності користувача. Існує два можливих способи роботи з даними з лог журналу - структурований та неструктурований.

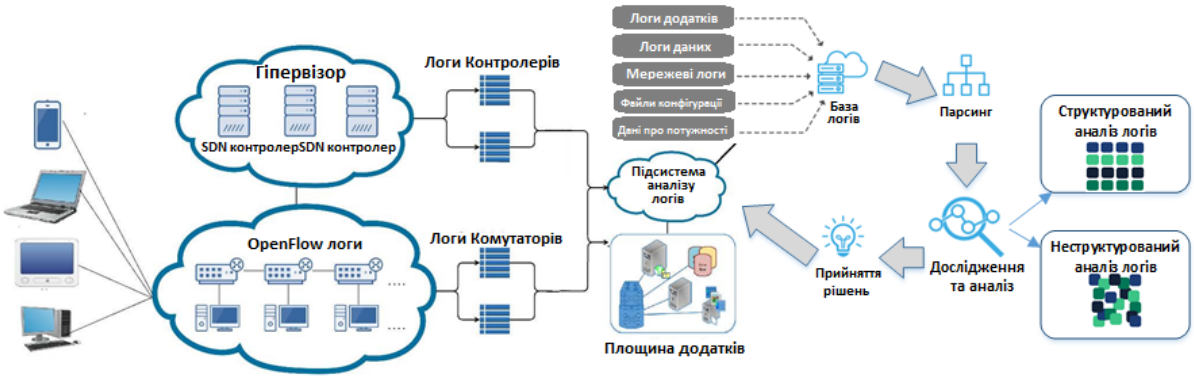


Рис. 4.2. Принцип роботи Підсистеми аналізу логів

Структуровані дані є високоорганізованими, заздалегідь визначеними та відформатованими до встановленої структури перед збереженням у сховищі даних, тоді як неструктуровані дані не визначаються заздалегідь моделлю даних, вони можуть бути створені людьми або створені машиною у текстовому чи нетекстовому форматі .

Структуровані дані можуть бути універсально зрозумілими, дають можливість використання різних засобів обробки даних, можуть бути легко засвоєні різними програмами даних, алгоритмами машинного навчання та вимагають набагато менше місця для зберігання. До недоліків можна віднести негнучкість сховища (якщо є вимога змінити структуру даних, необхідно оновити весь набір даних) та обмежені випадки використання (майже весь час можна використовувати лише для передбачуваних цілей, що спричиняє певну негнучкість).

Неструктуровані дані включають більш широкі варіанти використання, гнучке форматування та просте зберігання. Існує також можливість отримати з цього дуже цінну інформацію, яку неможливо отримати зі структурованих джерел даних. До недоліків можна віднести труднощі у підготовці та аналізі даних та вимоги до спеціальних інструментів для даних.

AlienVault USM1, IBM QRadar2, LogRhythm3 та Splunk Enterprise Security (ES) 4 - це приклади відомих програмних продуктів для аналізу службових даних. Вони покладаються на внутрішні формати подання (наприклад, MDI

Fabric та Common Information Model, що використовуються відповідно LogRhythm та Splunk ES) для імпорту та консолідації практично будь -якого джерела даних, з яким можна зустрітись у реальних системах. Згадані продукти реалізують вбудовані адаптери, які добре справляються з імпортом структурованих даних у внутрішнє представлення; проте вбудованої підтримки набагато більше і вона обмежена, особливо коли дані складаються з неструктурованих текстових журналів. Наприклад, у випадку джерела системного журналу, лише невелика кількість стандартних полів (наприклад, мітка часу, ім'я хосту, серйозність) автоматично розпізнаються згаданими продуктами, тоді як безкоштовне текстове повідомлення просто імпортується у його необробленому форматі. Крім того, у випадку застарілого або залежного від програми джерела лог журналу, аналітик повинен налаштувати свій власний адаптер.

Для комплексного аналізу лог журналів пропонуємо проводити паралельний аналіз як структурованих так і неструктурованих даних.

4.2.1 Метод неструктурованого аналізу на основі ентропії термів

У даній дисертаційній роботі пропонуємо для аналізу даних використовувати метод аналізу неструктурованих текстових журналів, особливістю яких є відсутність інформації про попередні стани системи.

Нехай система складається з вузлів N_j $(1 \le j \le S)$, кожен з яких має різноманітну структуру журналу, оскільки розділи журналів можуть створюватися різноманітним набором фреймворків, баз даних, проміжного програмного забезпечення тощо (рис.4.3). Загалом вузли видають L_i $(1 \le i \le N)$ кількість журналів подій. В якості одного з кроків можна відфільтрувати записи журналу користувачів, які виконують операції в системі. Наприклад, шукаючи ідентифікатор користувача у файлах журналу, можна знайти відповідні записи журналу, які належать користувачеві. Відповідно запропонований метод базуватиметься на знаходженні ентропії записів, створених за певний проміжок

часу та аналізі останніх значень ентропії для всіх файлів журналу в системі. Для цього необхідно провести вибірку із набору неструктурованих даних.

Вибірка - це періодичний процес, який кожні T періоду часу визначає ентропію нових L_i записів, створених протягом часу T_i: вибірка виконується індивідуально для кожного L_i запису. Вибірка даних із журналу та визначення ентропії включає два етапи: вилучення та зважування.

Враховуючи записи L_i, створені під час минулих T_i, вилучення даних витягує всі терми k, які входять у записи (терм являє собою послідовність символів $\{k_1,...k_n\}$, розділені одним або більше пробілами і може мати розмір) і проводить їх підрахунок. На етапі вилучення обчислюється ентропія на основі підрахунку термів. Результатом є ентропія E_{L_i}, яка визначатиметься:

$$E_{L_i} = -\sum_{j=1}^{n} P(k_j) \ln P(k_j) \qquad (4.1)$$

Ентропія - це числовий показник, який вимірює релевантність записів у L_i: чим вище значення ентропії, тим більш «цікавими» з точки зору безпеки, є записи.

Аналіз останніх значень ентропії для всіх файлів журналу в системі буде проводити Модератор. Модератор отримує значення ентропії для всіх журналів системи та на основі отриманих значень ентропії проводить їх порівняння. Якщо ентропія визначена в результаті зважування є більшою ніж ентропія визначена в результаті звичайного підрахунку термів, то такі записи є не типовими для лог журналу, а отже, ці записи є «цікавими» з точки зору мережевої безпеки.

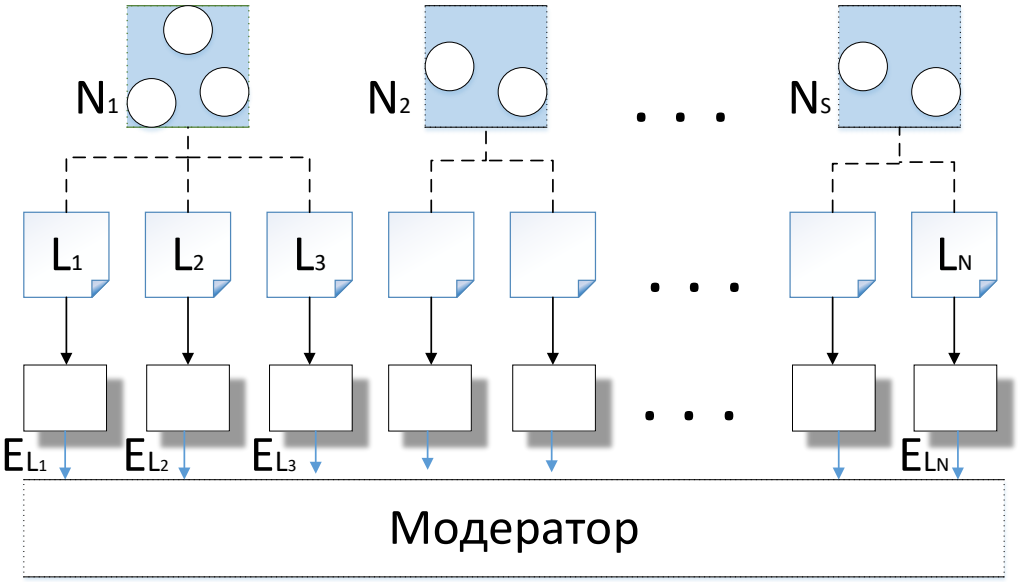


Рис. 4.3.Принцип вибірки неструктурованих даних та визначення їх ентропії

Вилучення термів застосовує деякі етапи підготовки даних до заздалегідь визначених умов. Журнали, що генеруються під час звичайної діяльності системи, місять в собі велику кількість випадкових термів, тобто таких, які зустрічаються лише один раз. Серед них можуть бути терми з мітками часу, значенням змінних, фреймворки цілих структур даних та ідентифікатори процесів. Фактично, типовий запис складається з інваріантних та змінних полів.

Інваріантні поля не змінюються у записах з однаковим шаблоном; з іншого боку, змінні поля, ймовірно, будуть зустрічатися рідко у журналі. Це може спотворити ступінь зважування, метою якого є присвоєння більших ваг нечастим термам.

Враховуючи набір записів у журналі, підготовка даних має на меті пом'якшити часту зміну термів. Наприклад, вилучення термів видаляє спеціальні буквено-цифрові символи (наприклад #,?, %), і реалізує список стоп-слів, щоб не перетинатися із звичайними термами: назви днів/місяців і т.д. Ще одним способом є метод Портера [86-88], який витягує основу кожного терма: щоразу, коли дві лексеми мають однакову основу, вони розглядаються як два входження одного і того ж терма. Що ще важливіше, кожен запис перевіряється з базою, за принципом представленим на рисунку 4.4. Запис замінюється

відповідним шаблоном, якщо шаблон є в базі. Якщо шаблон відсутній - запис залишається незмінним. Після того, як усі записи підготовлені, вилучення термів токенізує записи та підраховує кількість входів кожного терма у записах.

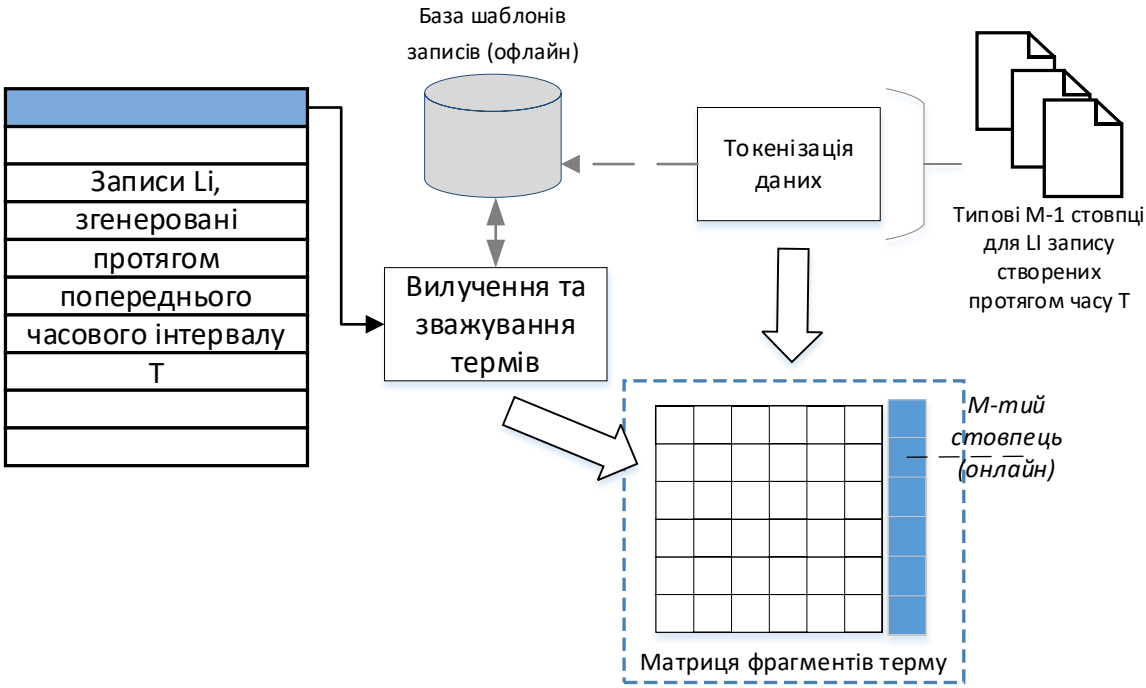


Рис. 4.4. Вибірка ентропії логів з L_i записів

Зважування термів дозволяє визначити актуальність термів у наборі даних. Для визначення того, чи «цікаві» записи, з точки зору мережевої безпеки, траплялися в L_i записі кожну T одиницю часу визначимо ентропію логів (позначимо її $\log.entropy$). Визначення цієї ентропії базується на визначенні матриці сторонніх фрагментів, тобто матриці $W * M$ з W термами (рис. 4.4). M-тий стовпець матриці містить кількість термів у L_i записах, згенерованих протягом останніх T одиниць часу; крайні ліві $(M-1)$ стовпці містять кількість термів, отриманий з $(M-1)$ L_i фрагментів тривалістю T_i, зібраних під час звичайної системної діяльності. Підрахунок термів у M-му стовпці витягується в режимі он-лайн, тоді як підрахунки в стовпцях $(M-1)$ встановлюються в автономному режимі на етапі налаштування, зробленому перед початком роботи методу. Крайні ліві $(M-1)$ стовпці матриці є постійними, тоді як M-й стовпець переписується кожні T_i одиниць часу.

Нехай, $x_{i,j}$ $(1 \le i \le W, 1 \le j \le M)$ кількість разів, коли терм i зустрічається у фрагменті j. Отримаємо $\log.\text{entropy}_M$, тобто оцінку ентропії логів M-ого стовпця:

$$\log.\text{entropy}_M = \sqrt{\sum_{i=1}^{W} \left(e_i \log_2 \left(1 + x_{i,M}\right)\right)^2} \tag{4.2}$$

де e_i $(0 \le e_i \le 1)$ представляє ентропію терму i для всіх M фрагментів і обчислюється відповідно до рівняння 4.3

$$e_i = 1 + \frac{1}{\log_2(M)} \sum_{j=1}^{M} p_{ij} \log_2(p_{ij}) \qquad p_{ij} = 1 + \frac{x_{i,j}}{\sum_{j=1}^{M} x_{ij}} \tag{4.3}$$

Як випливає з рівняння 4.3, терми, які часто зустрічаються на M - фрагментах, мають невелику вагу. Наприклад, терму в M-ому фрагменті, який ніколи не генерувався в L_i запиті під час регулярної діяльності (знову ж таки, крайні ліві стовпці $(M-1)$), надається велика вага. Результатом зважування є нова множина значень ентропії, яка надсилається Модератору.

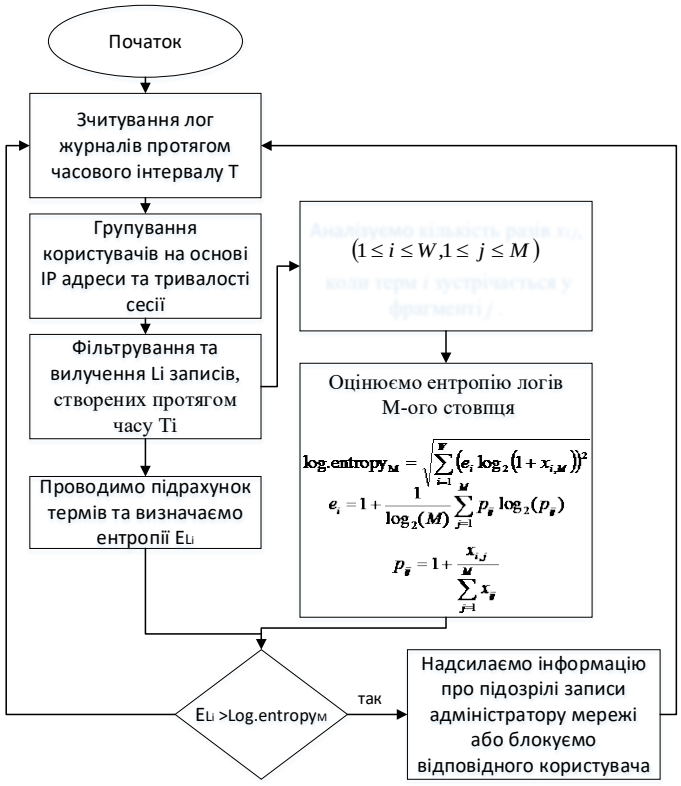


Рис. 4.5. Блок-схема алгоритму не структурованого лог аналізу

Блок схема алгоритму не структурованого аналізу наведена на рис. 4.5. Порівняння значень ентропії, визначеної в результаті зважування та в результаті звичайного підрахунку термів дозволяє Модератору отримати не типовий L_i запис серед множини записів лог журналу та класифікувати його як є «аномальний» з точки зору мережевої безпеки.

4.2.2. Структурований аналіз лог файлів як один із елементів виявлення аномалій

Крім неструктурованого аналізу даних пропонуємо в паралелі проводити аналіз структурованих даних. Проте, постійне зчитування з лог файлу структурованих даних буде доволі затратним з точки зору використання програмно-апаратних ресурсів. Саме тому, створимо базу даних та переносимо в неї всі лог файли. Це дозволить пришвидшити пошук та проводити фільтрування логів. Найбільш ідеальним варіантом для цього є використання так званої бази даних in-memory, оскільки завдяки реалізації принципу in-memory дані, що аналізуються постійно знаходяться в операційній пам'яті. Це дозволяє швидко виконувати операції запису і пошуку. На рис. 4.6 представлено блок-схему алгоритму запису/зчитування логу до бази даних.

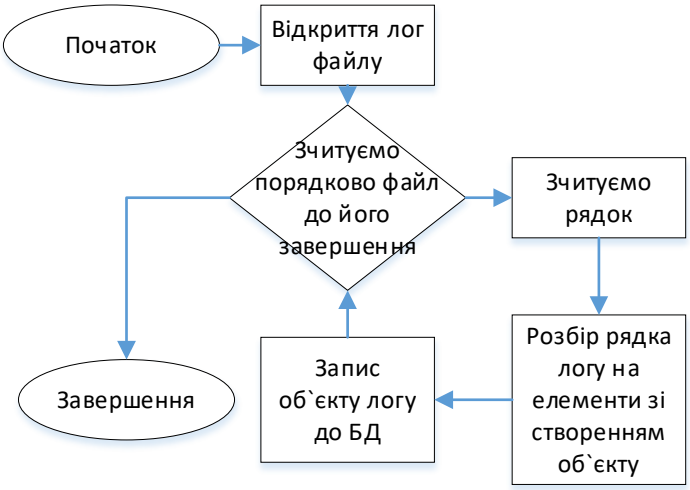


Рис. 4.6. Спрощені кроки переносу логів до бази даних

Для деякого спрощення та дослідження, за приклад візьмемо структурований лог формату:

$$[``yyyy`/`MM`/`dd`:`HH`:`mm`:`ss``]--ip\_address--ID\_port`$$

В цей лог записуються основні параметри підключення: дата і час підключення, IP адреса клієнта та номер порта (сокета), та кількість часу в секундах, яку клієнт провів підключеним до сервера.

Такий тип логу дозволяє сконцентруватися виключно на потрібній інформації. Але за потреби лог можна розширити. В якості парсера оберемо Antlr. Він дає змогу швидко змінювати лексер дуже швидко. Типовий парсер є комбінацією лексера і парсера. Для початку необхідно створити граматику. Для логу представленого формату, необхідно виділити ip адресу, а також дату і час. Все інше відкидаємо. Створимо граматичний словник для парсера (рис. 4.7).

```
grammar RQL;
leg: query;
query: ip '- - '['daceAndTime'] ' WORD*;
ip: NUMBER*' . 'NUMBER* ' . 'NUMBER*' . 'NUMBER*
daceAndT ime:dace':'time;
dace: NUMBER'/'NUMBER'/'NUMBER;
cime: hour':'minuce':'second;
hour:HUMBER;
minuce:NUMBER;
second:NUMBER;
NUMBER : [0-9]+;
WORD : [a-zA-Z_]+;
WHITESPACE : (' '|'\t'|'.')+ -> skip;
```

Рис. 4.7. Граматичний словник для парсеру

В такому вигляді граматичного словника окремо виділимо ip адресу. Більш складним є розбирання дати і часу. Оскільки може бути необхідно брати лише час або лише дату, було вирішено виділити їх в окремі лексеми, а потім об’єднати їх разом. Після цього створюється клас, який розбирає вхідний текст на лексеми. Створюється синтаксичний аналізатор. Вхідний текст розбирається, аналізується і виділяються необхідні елементи на етапі візитора. Це клас, який ходить по дереві лексеру і дозволяє обробляти лексеми на вході чи на виході.

Розглянемо приклад: для логу формату ‘52.14.235.178 - - [2019/11/06:19:40:00]’ розбір на лексичне дерево буде виглядати як представлено на рис. 4.8

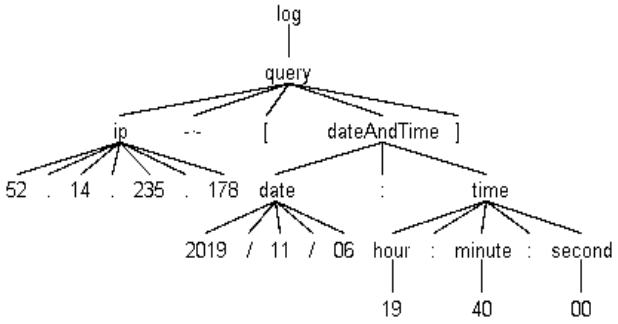


Рис. 4.8. Розклад логу на лексичне дерево

Наступним кроком після парсингу даних є створення об’єкту, в який поміщаються розспарсені дані. Після цього використовуючи Hibernate зберігаємо їх в базу даних для подальшої роботи.

Для розпізнавання DoS -атаки розробимо алгоритм, який дозволяє виділити запит користувача серед інших запитів, який атакує сервіс. Для цього Log analysis subsystem відправляє до бази даних запит, який знаходить всі звернення до веб-серверу в проміжку часу від 24 годин до запиту і до самого моменту цього ж запиту. Після цього для пришвидшення пошуку сортується список всіх запитів за IP-адресою. Далі в окремому модулі знаходиться максимальна кількість запитів для кожної окремої IP-адреси і отримане число повертається, після чого серед усіх запитів, здійснених протягом 24 годин, знаходиться IP-адреса, з якої і було відправлено найбільшу кількість запитів. Для того, щоб перевірити, чи максимальна кількість запитів не відрізняється від кількості запитів інших користувачів, треба знайти медіану кількості запитів серед всіх інших користувачів, за виключенням числа максимальної кількості запитів. Надалі проводиться порівняння, чи максимальна кількість запитів буде більшою ніж медіана, помножена на певний коефіцієнт. Таким коефіцієнтом обрано число 10, оскільки воно дозволяє відсікти сплески трафіку в пікові години навантаження від реальних DoS -атак, де з однієї ip-адреси надсилається велика кількість запитів. Оскільки це буде виділятися серед загального веб-трафіку, можна буде визначити IP-адресу, з якої ймовірно здійснюється DoS -атака.

Рис. 4.9. Загальна блок-схема алгоритму пошуку DoS -атаки за останню добу

Необхідно врахувати той факт, що атака відбувається з використанням великої кількості джерел трафіку. Для цього по світу заражається велика кількість ‘стандартних’ пристроїв, які після зараження продовжують і далі функціонувати належним чином, допоки зловмисник не задіє внесений програмний код

Тож алгоритм пошуку D DoS -атаки буде виглядати так:

- для початку ми шукаємо в базі даних запити в проміжку часу від 24 годин до запиту і до самого моменту цього ж запиту;
- сортуємо кількість запитів погодинно;
- розбиваємо список на різні списки, де кожній годині відповідає один список;
- з відсортованого списку, беремо кількість запитів, яка відповідає годині дня коли виконується аналіз;
- знаходимо медіану кількості запитів за весь день, за виключенням години, яка відповідає за запити в данну годину;

- якщо при порівнянні середньої кількості запитів і кількості запитів на поточну годину їх значення вони будуть відрізнятися більше ніж у 10 разів, то ми можемо зробити висновок, що в даний період відбувається DoS - атака;
- у випадку, коли різниця між середньою кількістю запитів та їх кількості запитів на поточну годину буде меншою ніж в 10 разів, проводиться додатковий аналіз, який дає змогу відслідкувати довготривалу Ddos-атаку. Для цього беруться до уваги всі запити, здійснені протягом 7 днів, знову розбиваються поденно і повторюється процедура знаходження різниці між їх середньою кількістю та максимальним значенням, знаходиться медіана;
- відбувається порівняння кількості запитів на поточну годину і значення медіани кількості запитів за 7 днів: якщо різниця буде значною, тобто більшою ніж в 10 разів, ми можемо зробити висновок, що в даний момент відбувається Ddos-атака, і це дозволяє задіяти систему захисту, що дає можливість продовжити коректну роботу.

Крім цього важливим для нас параметром є тривалість такої сесії. Нехай маємо множину тривалостей доступу до веб-серверу $\{T_{access}\}$ із множини IP адрес$\{P\}$. На основі цієї інформації будемо тренувати систему. Для цього визначимо відносну ентропію Кульбака-Лейблера.

Відносна ентропія Кульбака-Лейблера (KL) є мірою того, наскільки один розподіл імовірності відрізняється від іншого, еталонного розподілу ймовірності. До області її використання належать віднóсна (шеннонова) ентропі́я в інформаційних системах, випадкóвість у неперервних часових рядах, та прúріст інформáції при порівнюванні статистичних моделей висновування. На противагу до різновидності інформації, вона є асиметричною міжрозподіловою мірою, і відтак не відповідає вимогам статистичної метрики розкиду. В простому випадку нульове розходження Кульбака — Лейблера показує, що два розглянуті розподіли є ідентичними.

В нашому випадку порівнюватимемо середній час тривалості сесії із тривалістю доступу до серверу із конкретних IP адрес, які відсортувалися в наслідок роботи алгоритму, представленому на рис. 4.9. Для визначення KL потрібно обчислити інформаційну ентропію обох розподілів і знайти їх різницю. Для нашого випадку це визначатиметься:

$$KL(T_{aver} \| T_{access\_last\_hour}) = \sum T_{aver}(P) \log \frac{T_{aver}(P)}{T_{access\_last\_hour}(P)} \quad (4.4)$$

$$H(T_{access}) = \log T_{aver}(P) - KL \quad (4.5)$$

Загальну блок-схему роботи даного алгоритму наведено на рисунку 4.10.

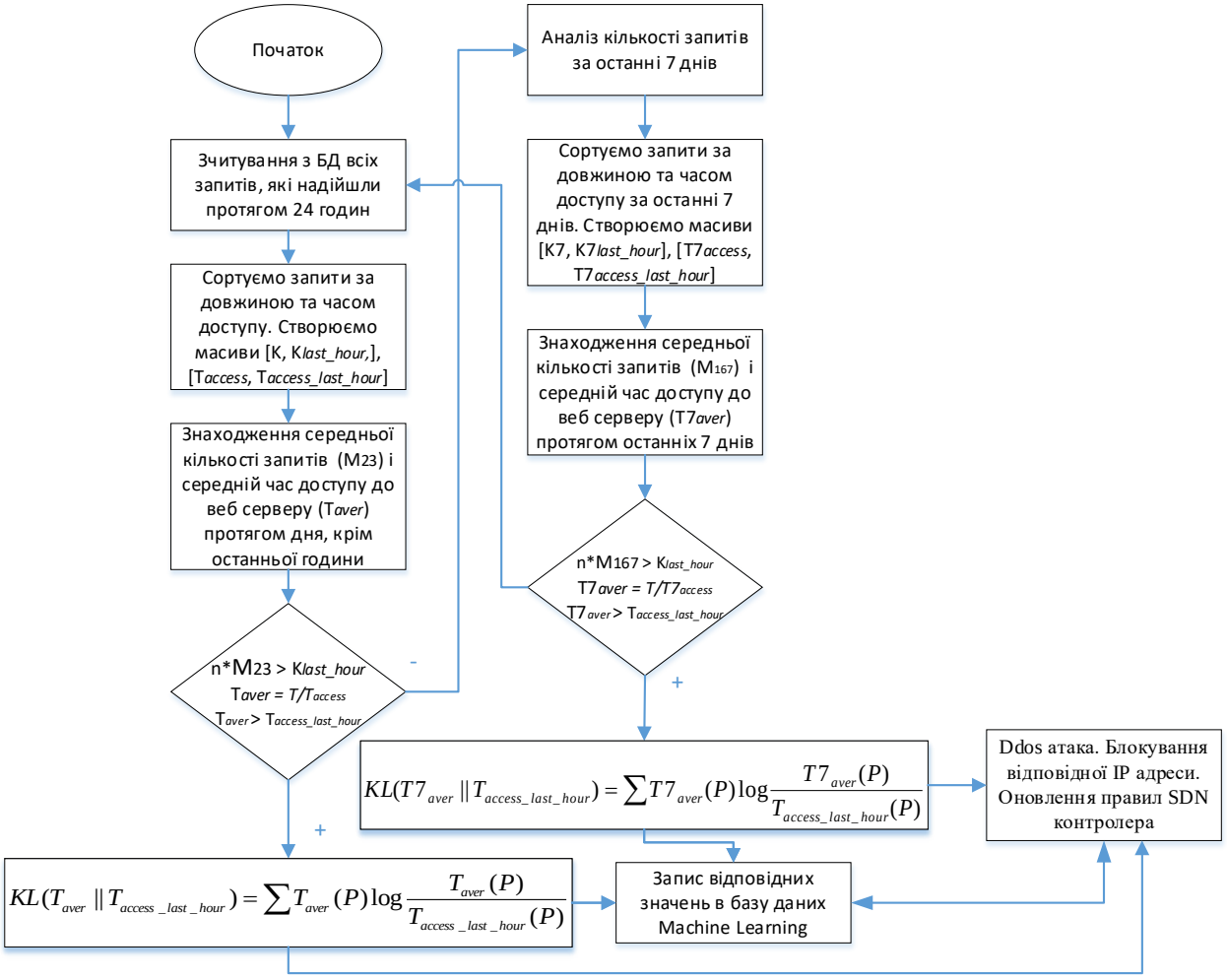


Рис. 4.10. Блок-схема алгоритму розпізнавання DDoS-атаки із використанням структурного методу аналізу лог файлів

Отримані значення записуватимуться у базу даних Машинного навчання. Для навчання та обробки інформації отриманої в результаті як структурованого

так і неструктурованого аналізу обрано алгоритм машинного навчання «Дерево рішень», через його простоту, надійність, та відповідність ситуації. В даному випадку слід класифікувати клієнтів по категоріям таким чином: якщо значення інформаційної ентропії обох розподілів і їх різниця доволі велика – зловмисника виявлено і контролер створює правило, яке заблокує відповідну IP адресу та порт. Якщо результат порівняння не приніс результату відбувається порівняння тривалостей доступу до сервісу протягом останніх семи днів. Накопичення значень KL в базі ML дозволить виявляти аномалії при поступленні потоків запитів, на основі аналізу тривалості доступу до сервісу та прописувати правила роботи Контролера. В результаті використання такого навчання та відповідно пості ного моніторингу сесій, SDN контролер блокуватиме IP домени, з яких лише розпочинаються DDoS атаки.

4.3. Дослідження ефективності інтеграції архітектури інтелектуального виявлення DDoS атак на основі розробленої імітаційної моделі інформаційно-комунікаційної сервісно-орієнтованої мережі

Для дослідження ефективності інтеграції архітектури інтелектуального виявлення DDoS атак проведено імітаційне моделювання. Дослідження проводилося на основі розробленої імітаційної моделі інформаційно-комунікаційної сервісно-орієнтованої мережі на базі НТЦ МТ Національного університету «Львівська політехніка» на одній із підмереж, яка була ізольована від всієї мережі.

Модель базується на взаємодії між клієнтами і сервером, на якому розміщено веб сервіс, а також програмного аналогу SDN контролеру. Комунікація між клієнтами і веб сервером реалізована на основі socket модулів Python.

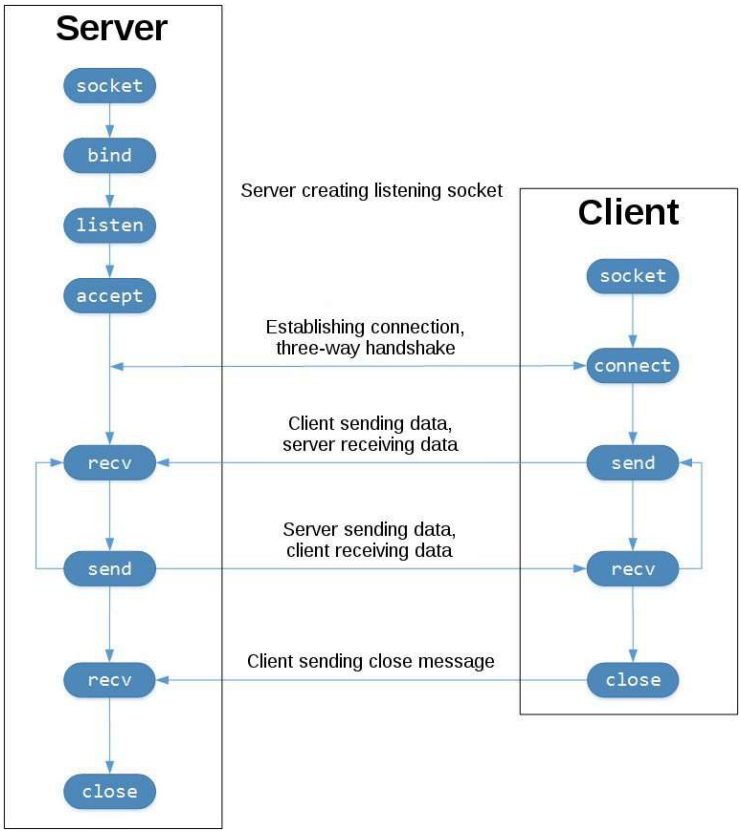


Рис. 4.11. Принцип клієнт-серверної взаємодії на базі сокетів

Перед початком роботи веб-серверу генеруються випадкові дані, які записуються у лог файл. Дані логів з файлу генерують дані за нормальним законом розподілу, проте не будуть послідовними в часі, тому вони додатково сортуються при записі в базу даних. Принцип запису в базу даних здійснюється на основі алгоритму наведеного на рис. 4.6. Реалізація взаємодії з базою даних відбувається з використанням мови програмування Java та методу ORM (Object-Relational Mapping), який реалізований через `Hibernate`. Сама база даних реалізована як MySQL база. В середині площини управління комунікація відбувається з використанням протоколів Rest API та Openflow.

Запустивши сервер модель створює відповідний вхідний сокет на довільному порті. Інформація про цей запис перенаправляється у «вхідні» та «вихідні» списки, які в свою чергу, використовуючи модуль Select, створюють списки ReadList та WriteList. Після формування цих списків відбувається опитування подій і обробка їх результатів. Під подією розуміється інформація

про підключення кожного клієнта. За допомогою парсингу записуються дані про номер порта, за допомогою якого підключається клієнт, та переносяться до основних «вхідних» та «вихідних» списків портів. Оскільки клієнт щойно підключився і його запит ще не оброблено, то «вихідні» списки порожні. У списки WriteList – записуються дані у випадку, якщо відповідь вже була надіслана, а у messageForSocket – записується безпосередньо відповідь клієнта серверу. Проте, як тільки клієнт під'єднався ці списки також порожні. Інформація із усіх «вхідних» списків дублюється у файл eventLogs.txt

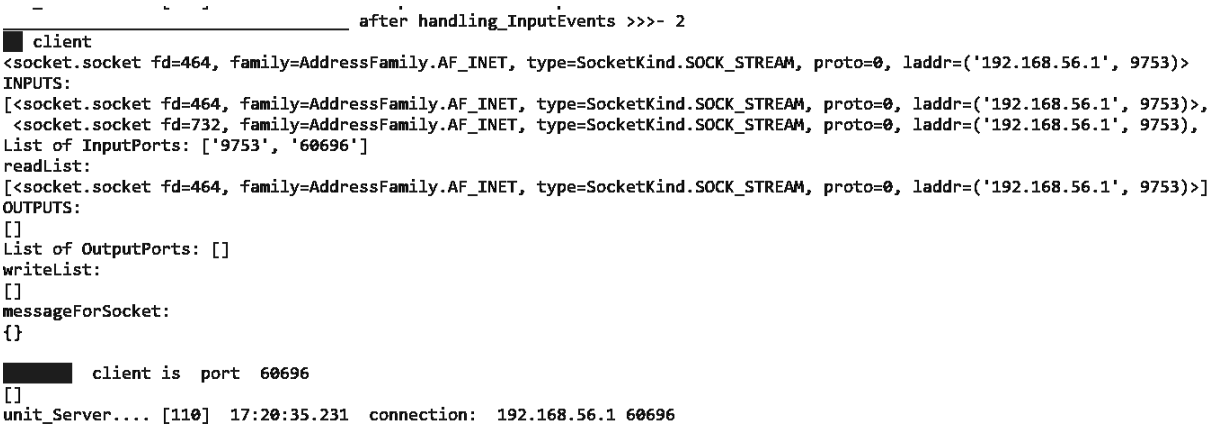


Рис. 4.12. Інформація про клієнтське підключення

Після цього система починає обробляти запит від клієнта. Додає дані про клієнта у відповідні списки, формує повідомлення-відповідь для клієнта, а також фіксує момент його створення. Цей час буде часом підключення користувача до системи і записуватись в лог файл разом з IP адресою, номером порта та ID клієнта.

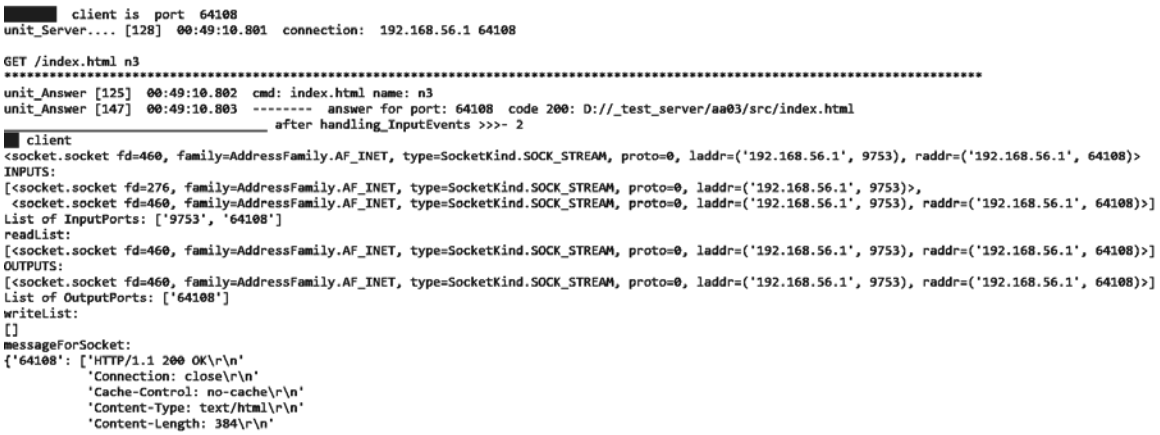


Рис. 4.13. Обробка запиту від клієнта і відправка повідомлення-відповіді

```
d:\_test_server\aa03>client_N3.py 192.168.56.1 9753
#################### Client_N3 ####################
I = 1-3 ########################################################################
'GET /index.html n3'
(b'HTTP/1.1 200 OK\r\nConnection: close\r\nCache-Control: no-cache\r\nContent'
 b'-Type: text/html\r\nContent-Length: 384\r\n\r\n<!DOCTYPE html>\n<html lang='
 b'"en">\n<head>\n    <meta charset="UTF-8">\n    <meta name="viewport" conten'
 b't="width=device-width, initial-scale=1.0">\n    <meta http-equiv="X-UA-Co'
 b'mpatible" content="ie=edge">\n    <link rel="stylesheet" type="text/css" '
 b'href="main.css">\n    <title>Hello there</title>\n</head>\n<body>\n        <'
 b'div>\n            <h1>Hello there</h1>\n        </div>\n</body>\n</html>\n')
Client was closed/
```

Рис. 4.14. Надсилання файлу користувачеві

Після отримання відповіді, клієнт може ініціювати повторні запити або відключитись. Якщо клієнт повторно ініціює запит, то сервер фіксує час і обраховує сумарну кількість часу цього підключення, та записує її в лог файл. Якщо клієнт відключається - сервер фіксує момент від'єднання клієнта і обраховує сумарну кількість часу, коли цей клієнт підключався до системи (тривалість сесії) та записує її в лог файл.

```
_____ after handling_OutputEvents ### 2
client
<socket.socket [closed] fd=-1, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0>
INPUTS:
[<socket.socket fd=600, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('192.168.56.1', 9753)>]
List of InputPorts: ['9753']
readList:
[]
OUTPUTS:
[]
List of OutputPorts: []
writeList:
[<socket.socket [closed] fd=-1, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0>]
messageForSocket:
{}

Time of Connect:  2021-05-29 18:24:06.888116
Time of Disconnect:   18:24:12.959
Time on server: 6
Connection closed, data removed.
```

Рис. 4.15. Очищення «вхідних» та «вихідних» списків і запис тривалості сесії

Після 5 хвилин роботи нашої імітаційної моделі у файл eventLogs.txt, вноситься 150000 записів. Для них IP-адреса і час генеруються за законом розподілу Гауса та у випадку помилки - прологується. Вміст лог файлу матиме вигляд представлений на рисунку 4.16.

```
[2021-05-24 22:22:20] 192.168.56.1 62758 n3     12
[2021-05-24 22:48:04] 192.168.56.1 63074 n1 af  52
[2021-05-24 22:49:54] 192.168.56.1 63108 n4 af  22
[2021-05-24 23:16:57] 192.168.56.1 63478 n1 af  10
[2021-05-24 23:18:29] 192.168.56.1 63489 n7 af  12
[2021-05-24 23:21:46] 192.168.56.1 63674 n1 af  12
[2021-05-24 23:23:29] 192.168.56.1 63705 n9 af  41
[2021-05-24 23:24:00] 192.168.56.1 63718 n2     11
[2021-05-25 00:27:31] 192.168.56.1 64388 n4     12
[2021-05-25 00:29:12] 192.168.56.1 64415 n7     42
[2021-05-25 00:40:25] 192.168.56.1 64556 n2     12
[2021-05-25 00:40:42] 192.168.56.1 64562 n3     11
[2021-05-25 00:40:56] 192.168.56.1 64563 n1     11
[2021-05-25 14:41:38] 192.168.56.1 49975 n1     55
```

Рис. 4.16. Приклад записів згенерованих логів у файлі

Для деякого спрощення задачі MAC-адреса не бралася до уваги. Для того, щоб відрізнити звичайного клієнта від зловмисника, у якості додаткової ознаки було введено кількість файлів, які клієнт отримав.

Для логування повідомлень в якості аналізатора було обрано Logback. Дана обгортка дозволяє налаштувати різноманітні варіанти запису логів у різні напрями. При цьому запис до файлу можна налаштувати таким чином, щоб розбивати логи по різних файлах. У випадку застарілості логу, він буде автоматично перезаписаний. Логування умовно, можна розділити на 2 етапи: перевірка навантаження за день у порівнянні із середнім рівнем та перевірка кількості запитів у поточну годину і в середньому за 7 днів. Це дозволить відслідкувати атаки, які тривають більше години, оскільки тримати під атакою веб-сервер довгий час - затратний процес.

На рис.4.17 наведено результат того, як відбувається перевірка системи на атаку за допомогою запропонованого у п.4.2 комплексного підходу. Як видно з рисунку на даний момент атака не відбувається

```
: if number of visits for current hour more than average number of visits for 7 day  * 10: false
: number of visits for current number 0
: average number of visits for 7 day 222
: Ddos false
```

Рис. 4.17. Перевірка системи на атаку за допомогою комплексного підходу

Для перевірки правильності роботи, змінимо дані в базі даних, що дозволить зрозуміти, чи правильно працює аналізатор. Проведемо 2 симуляції.

Перша симуляція буде відповідати за перевірку наявності атаки в поточну годину, при перевірці з запитами за один день. Друга перевірка буде відповідати ситуації, коли перша перевірка дала негативний результат. Після цього необхідно перевірити, чи не проходить атака великий проміжок часу.

Результат першої перевірки представлений на рисунку 4.18. При 260 кількості запитів за поточну годину і середньою кількістю запитів рівною 25, проводиться атака на наш веб-сервер. Логіка аналізатора виділила запити в базі даних і порахувавши їх дала відповідь, що відбувається атака.

```
number of visits for current hour 260
average number of visits for day 25
Number of visits for last hour more than average number of visits for a day
Ddos true
```

Рис. 4.18. Перший етап перевірки на атаку

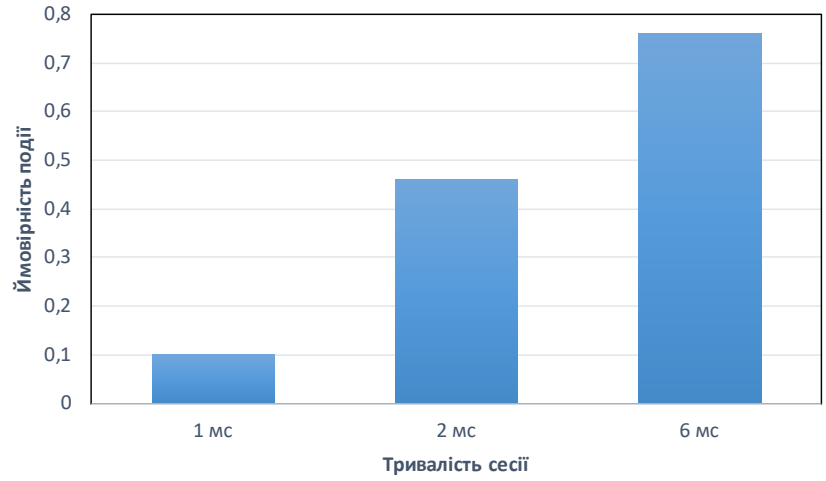
Результат другої перевірки наведено на рис. 4.19. У цьому випадку враховувалися дані за поточну годину і середнє значення запитів за тиждень. Із отриманих результатів видно, що середня кількість запитів за поточний день не дуже відрізняється від максимальної, що дозволяє пройти перевірку з негативним результатом на поточний день і перейти до перевірки за тиждень.

```
number of visits for current hour 3000
average number of visits for day 2500
if number of visits for current hour more than average number of visits for 7 day  * 10: true
number of visits for current number 3000
average number of visits for 7 day 221
Ddos true
```

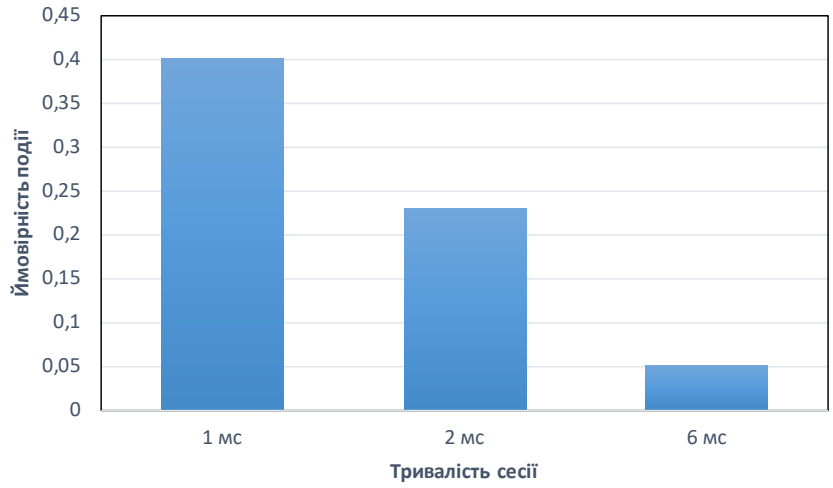
Рис. 4.19. Другий етап перевірки на атаку

Для перевірки доцільності використання проведемо визначення розбіжності між різними часами сеансу за допомогою підходу Кульбака-Лейблера. На рис. 4.20 показана ймовірність подій (позначених як P і Q) залежно від тривалості сеансу. Для цього розраховано середнє значення ймовірності кожної події та розбіжність між ними, за допомогою підходу Кульбака-Лейблера. В іншому експерименті (рис. 4.21) ймовірність двох подій майже однакова для кожного часу сеансу. Коли визначити середнє значення та

ентропію між P і Q, видно, що розбіжність зменшилася. Ці експерименти підтверджують ефективність використання запропонованого підходу для розмежування звичайних та аномальних запитів клієнтів.

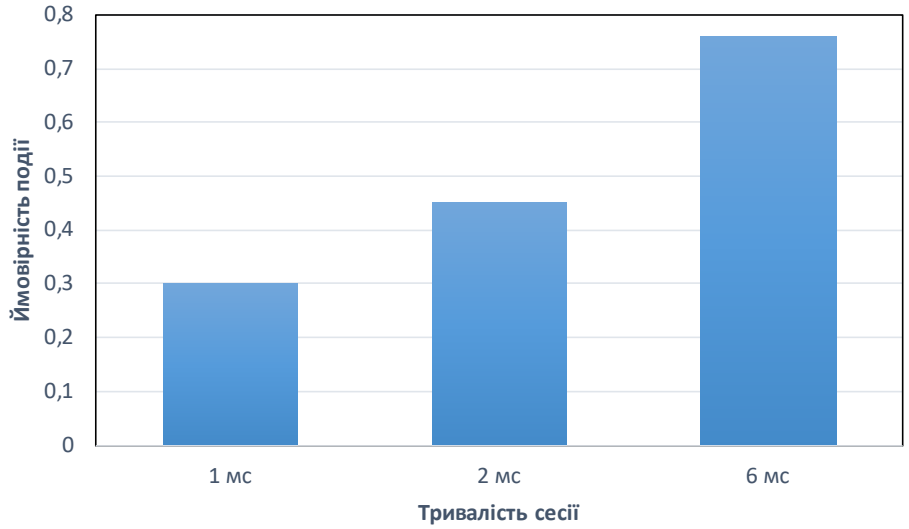


а)

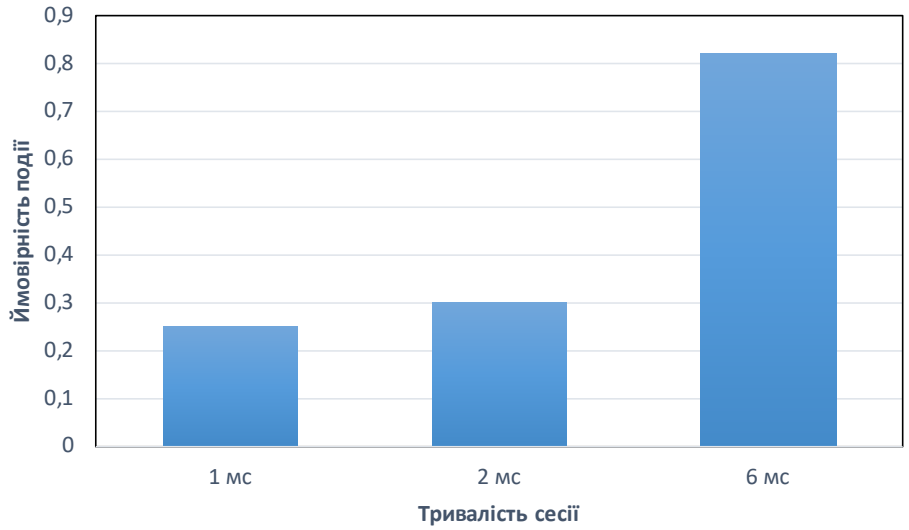


б)

Рис. 4.20. Ймовірність подій P (а) і Q (б) залежно від тривалості сесії (після першої ітерації алгоритму)



а)



б)

Рис. 4.21. Ймовірність подій P (а) і Q (б) майже однакова для кожної сесії (після другої ітерації алгоритму)

Таблиця 4.1

Середнє значення ймовірності кожної події та розбіжність між ними

Перша ітерація алгоритму		Друга ітерація алгоритму	
P=1.200 Q=0.610		P=1.400 Q=1.350	
KL(P || Q)	KL(Q || P)	KL(P || Q)	KL(Q || P)
4.490	0.539	0.110	0.036

Ці експерименти підтверджують ефективність використання запропонованого підходу для розмежування звичайних та аномальних запитів клієнтів.

Для перевірки ефективності інтеграції розробленої архітектури проведемо атаку на веб сервіс та застосуємо комплексного підхід до аналізу даних. Не зважаючи на логування, перевірка статусу роботи системи та підрахунок навантаження в різні моменти часу можуть викликати незручності. Для цього розроблено окремий модуль, який буде відповідати за відображення стану ресурсу та навантаження на нього. В якості UI платформи було взято Vaadin. Моніторинг додатка, збирання показників, поведінку трафіку або стану відповідної бази даних проводитиметься на основі Spring Actuator. Результати моніторингу наведені на рис. 4.22 – 4.24. Для спрощення представлення результатів прийнято відображати трафік у відносних одиницях. Одна умовна одиниця трафіку прирівнюється до 21 Мбайт даних.

Рис. 4.22. Показники навантаження на веб-сервіс

По замовчуванню встановлено представлення навантаження за останні 30 календарних днів. Але є можливість перемикати періоди навантаження. Для перевірки ефективності відслідковування атаки, проведемо атаку на відповідний веб сервіс.

Отримані результати представлені на рис. 4.23. З рис. 4.23 видно, що аналізатор трафіку, який працює на основі запропонованого підходу відслідкував момент, коли відбулася DDoS атака. Із отриманих результатів видно, наскільки різко змінилося навантаження на систему, порівняно з попереднім днем.

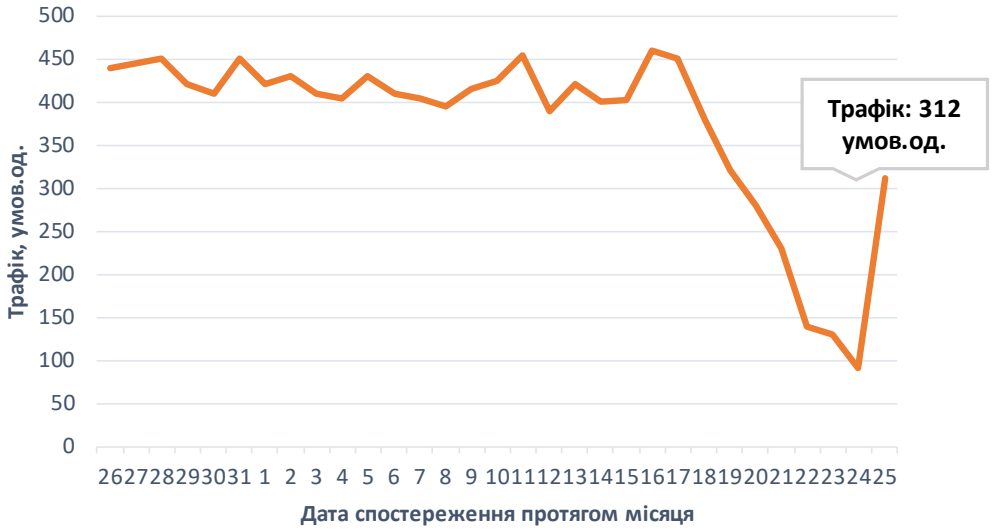


Рис. 4.23. Момент виявлення атаки на веб-сервіс

Для більш детальної перевірки, перейдемо на відображення навантаження погодинно. Результат наведений на рис.4.24. Бачимо, що в останню годину різко зросло навантаження на сервіс, хоча перед цим воно було настільки низьким, що його складно відрізнити від нуля.

Рис. 4.24. Момент виявлення атаки на веб-сервіс в погодинному відображенні

Отримані результати підтверджують ефективність використання запропонованого підходу для розмежування звичайних запитів клієнта та DDoS-атак.

Як зазначалося вище вся отримана інформація надсилається у базу даних Машинного навчання, де відбувається процес навчання на основі аглоритму «Дерева рішень». Для навчальної вибірки було створено окремий відредагований датасет, що знаходиться у файлі Log-learn.txt.

4	13	no
4	12	no
4	60	no
4	16	no
1	57	yes
1	70	yes
1	12	no
1	22	no
1	61	yes
1	73	yes
4	12	no
1	60	yes
4	16	no
1	47	no
1	70	yes
1	12	no
1	82	yes
1	61	yes

	Files	Time
0	4	13
1	4	12
2	4	60
3	4	16
4	1	57
5	1	70
6	1	12
7	1	22
8	1	61
9	1	73
10	4	12
11	1	60
12	4	16
13	1	47
14	1	70
15	1	12
16	1	82
17	1	61

	Key
0	no
1	no
2	no
3	no
4	yes
5	yes
6	no
7	no
8	yes
9	yes
10	no
11	yes
12	no
13	no
14	yes
15	no
16	yes
17	yes

Рис. 4.25. Навчальний датасет та його відображення в консолі

Також введено нову колонку Key, зі значеннями yes та no, що показують чи клієнт зловмисниий чи ні.

Модель навчання побудована за допомогою функції DecisionTreeClassifier бібліотеки sklearn. Для перевірки її адекватності і точності проведено два тести із даним з датасету. Для прикладу розглянемо два записи: перший запис - 4 записи у лог файлі, а тривалість сесії рівна 13 секунд; другий - 1 запис у файлі та 57 тривалість сесії. Припустимо, що ці записи є типовими для системи. Наступними протестуємо дані, що відсутні в датасеті (1,60;1,65;1,70). Усі вони є характеристиками зловмисних клієнтів і всіх їх програма визначила правильно. Результати тестування наведено на рис. 4.26

```
Prediction with values from logs
['no']
['yes']
******************************
```

```
Prediction with new values
['yes']
['yes']
['yes']
```

Рис. 4.26. Перевірка моделі навчання по даним з датасету і сторонніми даними

Оскільки лог аналіз успішно пройшов перевірку і довів свою ефективність, його можна використовувати для виявлення зловмисних клієнтів. Програма парсить лог і підставляє дані отримані від кожного підключення у модель, що перевіряє, зловмисний клієнт чи ні. Якщо клієнт виявився зловмисним, його ID зберігається у файл Check.txt.

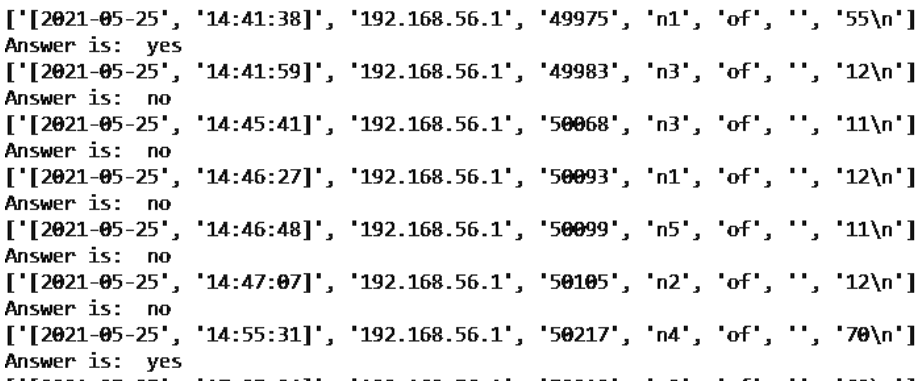


Рис. 4.27. Модель машинного навчання аналізує лог і виявляє зловмисні клієнти

Кожного разу, коли клієнт підключається до сервера, сервер проводить перевірку ID клієнта із ID зловмисних клієнтів, що зберігаються у системі. Якщо вони збігаються, Контролер блокує підключення клієнта. Результат блокування наведений на рисунку 4.28

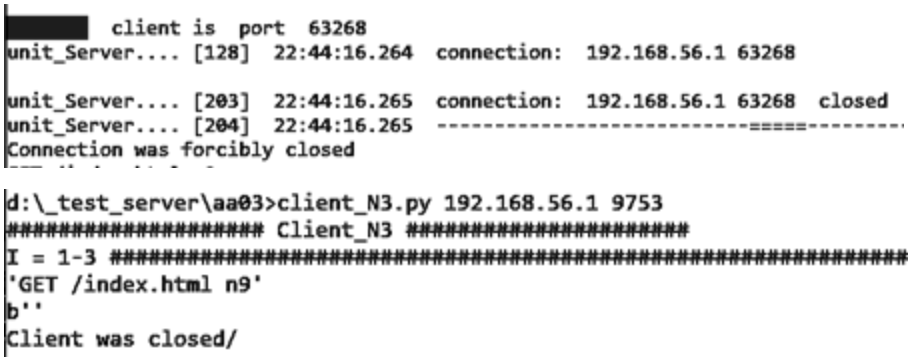


Рис. 4.28. Блокування сервером зловмисного клієнта.

Висновки до 4-го розділу

Запропоновано інтегровану архітектуру системи інтелектуального виявлення DDoS атак у інформаційно-комунікаційних мережах. Основною особливістю архітектури є реалізація підсистеми аналізу лог файлів, яка є

елементом інтегрованої системи управління, та аналізує потенційні проблеми і запускає запобіжні механізми, або може повідомляти системних адміністраторів про порушення безпеки мережі. Реалізація такої системи дасть змогу відстежувати як наявні загрози, так і прогнозувати атаки на мережу в цілому за рахунок використання елементу машинного навчання як окремої підсистеми інтегрованої архітектури управління мережею, що дасть змогу проводити інтелектуальне виявлення атак використовуючи інформацію про стан потоку, тривалість сесії та джерела його походження.

Розроблено комплексний підхід до аналізу логів, особливістю якого є проведення паралельного дослідження ентропійних властивостей потоків запитів. Для його реалізації використовується два методи: структурований та неструктурований аналіз даних. Інформація отримана в результаті комплексного аналізу лог журналів дасть змогу виявляти аномалії при поступленні потоків запитів, здійснюючи постійний моніторинг сесій та прописувати правила роботи Контролера, який відповідає за управління мережею.

Як елемент комплексного підходу до виявлення аномалій в системі запропоновано неструктурований метод аналізу даних, який базується на дослідженні ентропійних властивостей записів лог журналів. Метод реалізує алгоритм знаходження ентропії записів, створених за певний проміжок часу та аналізі останніх значень ентропії для всіх файлів журналу в системі. У випадку, коли ентропія визначена в результаті зважування є більшою ніж ентропія визначена в результаті звичайного підрахунку термів, то такі записи є не типовими для лог журналу, а отже, порушують мережеву безпеку системи.

Розроблено структурований метод аналізу даних, який базується на визначенні метрики поведінки трафіку використовуючи підхід Кульбака — Лейблера для виявлення аномалій потоку за часом тривалості сесії. В нашому випадку порівнюватимемо середній час тривалості сесії із тривалістю доступу до серверу із конкретних IP адрес. У випадку, якщо результат порівняння не

приніс результату відбувається порівняння тривалостей доступу до сервісу протягом останніх семи днів. Якщо різниця буде значною, то це свідчить, що в даний момент відбувається Ddos-атака.

Проведено моделювання та дослідження ефективності інтеграції архітектури інтелектуального виявлення DDoS атак на основі розробленої імітаційної моделі інформаційно-комунікаційної сервісно-орієнтованої мережі. Доведено, що використання комплексного підходу до аналізу лог журналів із використанням як структурованого так і неструктурованого аналізу службових файлів дало змогу відстежувати як наявні загрози, так і прогнозувати атаки на мережу в цілому за рахунок використання елементу машинного навчання як окремої підсистеми інтегрованої архітектури управління мережею. Реалізація машинного навчання дозволило класифікувати клієнтів по категоріям на основі значень ентропії та виявляти зловмисника. В результаті використання такого навчання та відповідно постійного моніторингу сесій, SDN контролер блокуватиме IP домени, з яких лише розпочинаються DDoS атаки.

ОСНОВНІ РЕЗУЛЬТАТИ ТА ВИСНОВКИ

В результаті виконання дисертаційних досліджень розв'язано науково-практичне завдання підвищення ефективності функціонування хмарних систем при їх інтеграції у сучасні інформаційно-комунікаційні сервісно-орієнтовані мережі шляхом розроблення методу кластеризації серверних вузлів, удосконалення моделей оцінки ефективності функціонування хмарних систем в умовах обслуговування групових потоків запитів та розвитку систем інтелектуального виявлення мережевих атак. В результаті проведення дисертаційних досліджень отримано наступні результати та висновки:

1. Проведено аналіз існуючих методів інтеграції хмарних платформ та сервісів з високим ступенем віртуалізації у сучасні інформаційно-комунікаційні мережі для вирішення задач передавання та обслуговування групових Big Data потоків. Встановлено, що ефективність функціонування хмарних центрів при їх інтеграції ускладнюється динамічним характером хмарного середовища та різноманітністю запитів користувачів. Достатньо високий ступінь віртуалізації, і відповідно велика кількість запитів, які обслуговують віртуальні машини, може призвести до перевантаження мережі в цілому. Крім того, управління такою інтегрованою мережевою інфраструктурою зводиться до проблеми підвищення захищеності даних від несанкціонованого доступу до них. Для цього необхідні певні системи моніторингу стану мережі, яка буде аналізувати можливі проблеми, що можуть виникати під час роботи, і запускати системи захисту в самому сервісі або повідомляти адміністраторів про можливу загрозу.

2. Удосконалено математичну модель оцінки ефективності хмарного центру з високим ступенем віртуалізації та в умовах групового надходження запитів, яка, на відмінну від існуючих, враховує можливість обслуговування при груповому надходженні запитів та розподілений час обслуговування запитів. Модель базується на двоступеневій техніці апроксимації, де основний не Марківський процес спочатку моделюється як вбудований напів марківський процес, який потім моделюється як апроксимований процес Маркова, але

тільки в моменти надходження групових потоків. Встановлено, що продуктивність роботи хмарних центрів дуже залежить від коефіцієнту варіації (CoV), часу обслуговування запитів і розміру групового потоку (тобто кількості запитів в груповому потоці запитів). Чим більший розмір потоку і/або значення коефіцієнта варіації, часу обслуговування запитів, тим довша тривалість відгуку, проте це сприяє зменшенню рівня використання ресурсів хмарними провайдерами. Як результат завдяки проведенню такої оцінки, в умовах надходження великих групових потоків запитів і /або великого значення CoV можна досягти підвищення ефективності функціонування хмарних центрів шляхом групування запитів за критерієм однорідності.

3. Удосконалено метод кластеризації вузлів хмарних центрів для зменшення тривалості пошуку маршруту між їх довільною парою. Завдяки реалізації розробленого методу кластеризації вузлів хмарних систем вдалося досягти зменшення тривалості пошуку маршруту між довільною парою віртуалізованих вузлів в межах одного кластеру у 1,3 рази. За допомогою кластеризації вдалося досягти зменшення тривалості пошуку маршруту між довільною парою віртуалізованих вузлів, які обслуговують один і той же груповий потік у 2,5 рази та підвищити ефективність інтеграції хмарних архітектур з високим ступенем віртуалізації, шляхом збільшення життєвого циклу фізичних машин, в умовах групового надходження задач або обробки потоків типу Big Data.

4. Для вирішення проблем ефективного функціонування хмарних центрів з високим ступенем віртуалізації, шляхом збільшення життєвого циклу фізичних машин, в умовах групового надходження задач або обробки потоків типу Big Data запропоновано алгоритм вибору головного кластерного вузла з використанням центральності по діаграмах Вороного, правил нечіткої логіки та моніторингу параметрів вузлів. Розроблений алгоритм вибору головного вузла кластера використовує в якості параметрів вибору центральність вузла по діаграмах Вороного і залишкову енергію. Ухвалення рішення про вибір

головного вузла здійснюється методами нечіткої логіки за допомогою правила Мамдані та центру ваги для дефазифікації, а також із врахуванням достатніх програмно-апаратних ресурсів для подальшого функціонування вузла в якості центроїда кластеру. Порівняльний аналіз розробленого алгоритму кластеризації та моделювання показують, що розроблений алгоритм зменшує споживання енергії на 8% і продовжує життєвий цикл мережі на 45% в порівнянні з відомими алгоритмами, що підвищує і дозволяє підтримувати якість надання послуг користувачам, особливо в умовах групового надходження задач або обробки потоків типу Big Data

5. Дослідження ефективності впровадження запропонованих рішень здійснювалося на основі розгорнутої моделі інфокомунікаційної сервісно-орієнтованої мережі з використанням CDN серверів як PaaS сервісу та хмарної архітектури Azure Cloud. Реалізація удосконаленої математичної моделі оцінки ефективності хмарного центру з високим ступенем віртуалізації та в умовах групового надходження запитів, та методу кластеризації вузлів хмарних центрів дало змогу зменшити тривалість завантаження статичного контенту з ендпоінтів (які виступали головними вузлами кластерів, визначеними з використанням алгоритму вибору головного кластерного вузла з використанням центральності по діаграмах Вороного, правил нечіткої логіки та моніторингу параметрів вузлів) у порівнянні із завантаженням з базового сервера в 4 рази. В результаті досліджень проаналізовано ефективність використання пропускної здатності каналу, який забезпечує доставку контенту. Використання пропускної здатності каналу при доступі до контенту із віддалених серверів зростає до 700 Кбіт/с. Дослідження показали, що контент не повинен кешуватись на POP або HTTP-клієнтом. Лише в одному випадку на 0,05 с. кешована версія запитуваного ресурсу не була знайдена на найближчому до клієнта POP.

6. З метою підвищення ефективності управління будь якою мережевою інфраструктурою та вирішення проблеми підвищення захищеності даних від

несанкціонованого доступу вперше запропоновано інтегровану архітектуру системи інтелектуального виявлення DDoS атак у інформаційно-комунікаційних мережах, яка реалізується на основі комплексного підходу до аналізу лог журналів із використанням як структурованого так і неструктурованого аналізу службових файлів. Реалізація такої системи дасть змогу відстежувати як наявні загрози, так і прогнозувати атаки на мережу в цілому за рахунок використання елементу машинного навчання як окремої підсистеми інтегрованої архітектури управління мережею, що дасть змогу проводити інтелектуальне виявлення атак використовуючи інформацію про стан потоку, тривалість сесії та джерела його походження, а також навчити SDN контроллер блокувати «подібні» потоки запитів без втручання системних адміністраторів.

7. Для реалізації комплексного підходу до аналізу лог журналів у роботі пропонується проводити паралельний аналіз як структурованих так і неструктурованих даних. Неструктурований метод аналізу даних базується на знаходженні ентропії записів, створених за певний проміжок часу та аналізі останніх значень ентропії для всіх файлів журналу в системі. Якщо ентропія визначена в результаті зважування є більшою ніж ентропія визначена в результаті звичайного підрахунку термів, то такі записи є не типовими для лог журналу, а отже, ці записи є «цікавими» з точки зору мережевої безпеки. У порівнянні з існуючими, структурований метод аналізу даних базується на визначенні метрики поведінки трафіку використовуючи підхід Кульбака — Лейблера для виявлення аномалій потоку за часом тривалості сесії. В нашому випадку порівнюватимемо середній час тривалості сесії із тривалістю доступу до серверу із конкретних IP адрес. У випадку, якщо результат порівняння не приніс результату відбувається порівняння тривалостей доступу до сервісу протягом останніх семи днів. Якщо різниця буде значною, то це свідчить, що в даний момент відбувається DDoS-атака. Інформація отримана в результаті комплексного аналізу лог журналів дасть змогу виявляти аномалії при

поступленні потоків запитів, здійснюючи постійний моніторинг сесій та прописувати правила роботи Контролера, який відповідає за управління мережею.

Проведено моделювання та дослідження ефективності інтеграції архітектури інтелектуального виявлення DDoS атак на основі розробленої імітаційної моделі інформаційно-комунікаційної сервісно-орієнтованої мережі. Доведено, що використання комплексного підходу до аналізу лог журналів із використанням як структурованого так і неструктурованого аналізу службових файлів дало змогу відстежувати як наявні загрози, так і прогнозувати атаки на мережу в цілому за рахунок використання елементу машинного навчання як окремої підсистеми інтегрованої архітектури управління мережею. Реалізація машинного навчання дозволило класифікувати клієнтів по категоріям на основі значень ентропії та виявляти зловмисника. В результаті використання такого навчання та постійного моніторингу сесій, SDN контролер блокуватиме IP домени, з яких лише розпочинаються DDoS атаки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1]. An official website of the European Union. Joint Communication: Eastern Partnership policy beyond 2020: Reinforcing Resilience – an Eastern Partnership that delivers for all [Електронний ресурс] – Режим доступу: https://eeas.europa.eu/headquarters/headquarters-homepage/76166/jointcommunication-eastern-partnership-policy-beyond-2020-reinforcing-resilience-%E2%80%93- eastern_en

[2]. An official website of the European Union. Joint Communication: Eastern Partnership: Commission proposes new policy objectives for beyond 2020 [Електронний ресурс] – Режим доступу: https://ec.europa.eu/commission/presscorner/detail/en/IP_20_452

[3]. Урядовий портал. Єдиний веб-портал органів виконавчої влади України. Постанова Кабінету міністрів України від 18 вересня 2019 р. № 856 «Питання Міністерства цифрової трансформації» [Електронний ресурс] – Режим доступу: https://www.kmu.gov.ua/npas/pitannya-ministerstvacifrovoyi-t180919

[4]. Ресурс Міністерства цифрової трансформації України. Національна стратегія розвитку широкосмугового доступу до Інтернету [Електронний ресурс] – Режим доступу: https://drive.google.com/file/d/1X9xlLCIpTaXwcOjRdK9l5Mw2cAIZryuQ/view

[5]. Офіційний веб-сайт «Міністерство та Комітет цифрової трансформації України». Повідомлення про проведення публічного громадського обговорення проєкту Національної стратегії розвитку широкосмугового доступу до Інтернету [Електронний ресурс] – Режим доступу: https://thedigital.gov.ua/regulations/povidomlennya-pro-provedennya-publichnogo-gromadskogoobgovorennya-proyektu-nacionalnoyi-strategiyi-rozvitku-shirokosmugovogo-dostupu-do-internetu

[6]. Урядовий портал. Єдиний веб-портал органів виконавчої влади України. Розпорядження Кабінету міністрів України від 28 жовтня 2020 р. №

1353-р «Про схвалення Стратегії цифрової трансформації соціальної сфери» [Електронний ресурс] – Режим доступу: https://www.kmu.gov.ua/npas/pro-shvalennya-strategiyi-cifrovoyi-transformaciyi-socialnoyi-sferi1353281020

[7]. Офіційний веб-сайт Верховної Ради України. Указ президента України від 15 березня 2016 року № 96/2016 «Про рішення Ради національної безпеки і оборони України від 27 січня 2016 року "Про Стратегію кібербезпеки України"» [Електронний ресурс] – Режим доступу: https://zakon.rada.gov.ua/laws/show/96/2016#Text

[8]. Офіційний веб-сайт Верховної Ради України. Закон України від 5 жовтня 2017 року № 2163-VIII «Про основні засади забезпечення кібербезпеки України» [Електронний ресурс] – Режим доступу: https://zakon.rada.gov.ua/laws/show/2163-19#n108

[9]. Rick Villars, Sandra Ng, Alan Webber etc. IDC FutureScape: Worldwide IT Industry 2022 Predictions. IDC FutureScape - Doc # JPJ48319521 [Електронний ресурс] – Режим доступу: https://www.idc.com/getdoc.jsp?containerId=JPJ48319521

[10]. F. Ait-Salaht and H. Castel-Taleb. Stochastic bounding models for performance analysis of clouds. In The 15th IEEE International Conference on Computer and Information Technology (CIT-2015), 26-28 October 2015, Liverpool, UK, pages 603–610. IEEE, 2015

[11]. B. Yang, F. Tan, Y. Dai, and S. Guo. Performance evaluation of cloud service considering fault recovery. First Int’l Conference on Cloud Computing (CloudCom) 2009, Beijing, China, Dec. 2009, pp. 571–576.

[12]. H. Qian, D. Medhi, and K. S. Trivedi. A hierarchical model to evaluate quality of experience of online services hosted by cloud computing,” in Integrated Network Management (IM), IFIP/IEEE International Symposium on, May. 2011, pp. 105–112.

[13]. Li K. Power and performance management for parallel computations in clouds and data centers, Journal of Computer and System Sciences, Vol. 82, 2016, No. 2, pp. 174-190.

[14]. Khazaei, H. Misic, J. Misic, V. B. Performance of an iaas cloud with live migration of virtual machines, Proceedings of the IEEE Global Communications Conference (GLOBECOM), GLOBECOM'13, Aalanta, USA, December 2013, pp. 2289-2293.

[15]. Xiong K. Perros H. Service performance and analysis in cloud computing, Proceedings of the 1st IEEE Congress on Services, SERVICES'09, Los Angeles, California, USA, July 2009, pp. 693-700.

[16]. Guo L., Yan T., Zhao S., Jiang C. Dynamic performance optimization for cloud computing using m/m/m queueing system, Journal of Applied Mathematics, Vol. 2014, 2014.

[17]. Bai W.-H., Xi J.-Q., Zhu J.-X., Huang S.-W. Performance analysis of heterogeneous data centers in cloud computing using a complex queuing model, Mathematical Problems in Engineering, Vol. 2015, 2015.

[18]. El Kafhali S., Salah K. Stochastic Modelling and Analysis of Cloud Computing Data Center, 20th ICIN Conference Innovations in Clouds, Internet and Networks, Paris, France, March 7-9, 2017, pp .122-126.

[19]. Khazaei, H. Misic, J. Misic, V. B. : Performance of an iaas cloud with live migration of virtual machines, Proceedings of the IEEE Global Communications Conference (GLOBECOM), GLOBECOM'13, Aalanta, USA, December 2013, pp. 2289-2293.

[20]. Xiong, K. Perros, H.: Service performance and analysis in cloud computing, Proceedings of the 1st IEEE Congress on Services, SERVICES'09, Los Angeles, California, USA, July 2009, pp. 693-700.

[21]. Guo, L. Yan, T. Zhao, S.Jiang, C.: Dynamic performance optimization for cloud computing using m/m/m queueing system, Journal of Applied Mathematics, Vol. 2014, 2014.

[22]. Ghosh, R. Longo, F. Naik, V. K. Trivedi, K. S.: Modeling and performance analysis of large scale iaas clouds, Future Generation Computer Systems, Vol. 29, 2013, No. 5, pp. 1216-1234.

[23]. Ghosh, R. Longo, F. Xia, R. Naik, V. K. Trivedi, K. S.: Stochastic model driven capacity planning for an infrastructure-as-a-service cloud, IEEE Transactions on Services Computing, Vol. 7, 2014, No. 4, pp. 667-680.

[24]. Mondal, S. K. Muppala, J. K. Machida, F.: Virtual machine replication on achieving energy-efficiency in a cloud, Electronics, Vol. 5, 2016, No. 3, p. 37.

[25]. Bai, W.-H. Xi, J.-Q. Zhu, J.-X. Huang, S.-W.: Performance analysis of heterogeneous data centers in cloud computing using a complex queuing model, Mathematical Problems in Engineering, Vol. 2015, 2015.

[26]. Sun, G. Liao, D. Anand, V. Zhao, D. Yu, H.: A new technique for efficient live migration of multiple virtual machines, Future Generation Computer Systems, Vol. 55, 2016, pp. 74-86.

[27]. Cheikh, H. B. Doncel, J. Brun, O. Prabhu, B.: Predicting response times of applications in virtualized environments, Proceedings of the 3rd Symposium on Network Cloud Computing and Applications, NCCA’14, Rome, February 2014, pp. 83-90.

[28]. Nguyen, T. A. Kim, D. S. Park, J. S.: Availability modeling and analysis of a data center for disaster tolerance, Future Generation Computer Systems, Vol. 56, 2016, pp. 27-50.

[29]. Salah, K. Elbadawi, K. Boutaba, R.: An analytical model for estimating cloud resources of elastic services, Journal of Network and Systems Management, Vol. 24, 2016, No. 2, pp. 285-308.

[30]. Н.В. Пелех, О.М. Шпур, М.М. Климаш, “Модель оцінки ефективності функціонування хмарного центру з високим ступенем віртуалізації та в умовах групового надходження запитів,” Проблеми телекомунікацій, №2(27), с.82-98. 2020.

[31]. Shpur O. Improving the Quality of Composite Services Through Improvement of Cloud Infrastructure Management // O. Shpur, M. Klymash, M. Seliuchenko, B. Strykhaliuk, O.Lavriv // International Journal of Computer Science and Information Security (IJCSIS). - 2015. - Vol. 13. - No. 9 – P.36-44

[32]. Heinzelman, W. R. An application–specific protocol architecture for wireless microsensor networks / Heinzelman W. R., Chandrakasan A. P., Balakrishnan H. // IEEE Transactions on Wireless Communications. – October 2002. – Vol. 1, № 4. – PP.660-670.

[33]. Koucheryavy, A. Cluster head selection for homogeneous Wireless Sensor Networks / A. Koucheryavy, A. Salim // Advanced Communicat ion Technology. ICACT 2009. 11th In-ternational Conference IEEE. – Phoenix Park, Korea. – Feb. 2009. – Vol. 03. – PP. 2141-2146.

[34]. Koucheryavy, A. Prediction-based Clustering Algorithm for Mobile Wireless Sensor Networks / A. Koucheryavy, A. Salim //International Conference on Advanced Commu-nication Technology, ICACT 2010. – Phoenix Park, Korea. – 2010. – PP. 1209-1215.

[35]. Ben Alla, S. Gateway and Cluster Head Election using Fuzzy Logic in heterogeneous wireless sensor networks / Ben Alla S., Ezzati A., Mohsen A. // International Conference on Multimedia Computing and Systems (ICMCS).10–12, May 2012. – PP. 761-766.

[36]. Kushwah VS, Goyal SK, Narwariya P (2014) A survey on various fault tolerant approaches for cloud environment during load balancing. Int J Comput Netw Wirel Mobile Commun 4(6):25–34

[37]. Yang W, Zhang C, Shao Y, Shi Y, Li H, Khan M, Hussain F, Khan I, Cui L-J, He H (2014) A hybrid particle swarm optimization algorithm for service selection problem in the cloud. Int J Grid Distrib Comput 7(4):1–10

[38]. Hussin M, Lee YC, Zomaya AY (2010) Dynamic job-clustering with different computing priorities for computational resource allocation. In: Proceedings

of the 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing, IEEE Computer Society, pp 589–590

[39]. Vidhate D, Patil A, Guleria D (2010) Dynamic cluster resource allocations for jobs with known memory demands. In: Proceedings of the international conference and workshop on emerging trends in technology, ACM, pp 64–69

[40]. SiM Abdulhamid, Latiff SMA, Bashir MB (2014) Scheduling techniques in on-demand grid as a service cloud: a review. J Theor Appl Inf Technol 63(1):10–19

[41]. Abdullahi M, Ngadi MA (2016) Symbiotic organism search optimization based task scheduling in cloud computing environment. Future Gener Comput Syst 56:640–650

[42]. Madni SHH, Latiff MSA, Coulibaly Y (2016) An appraisal of meta-heuristic resource allocation techniques for IaaS cloud. Indian J Sci Technol 9(4):1–14. doi:10.17485/ijst/2016/v9i4/80561

[43]. Chiroma H, Shuib NLM, Muaz SA, Abubakar AI, Ila LB, Maitama JZ (2015) A review of the applications of bio-inspired flower pollination algorithm. Procedia Comput Sci 62:435–441

[44]. Boru D, Kliazovich D, Granelli F, Bouvry P, Zomaya AY (2015) Energy-efficient data replication in cloud computing datacenters. Clust Comput 18(1):385–402. doi:10.1007/s10586-014-0404-x

[45]. Gangeshwari R, Subbiah J, Malathy K, Miriam D (2012) HPCLOUD: a novel fault tolerant architectural model for hierarchical MapReduce. In: 2012 international conference on recent trends in information technology (ICRTIT), IEEE, pp 179–184

[46]. G¸asior J, Seredyn´ski F (2013) Multi-objective parallel machines scheduling for fault-tolerant cloud systems. In: Joanna K, Di Martino B, Talia D, Xiong K (eds) Algorithms and architectures for parallel processing. Springer, Switzerland, pp 247–256. doi:10.1007/978-3-319-03859-9_21

[47]. Klymash Mykhailo, Shpur Olga, Lavriv Orest, Peleh Nazar. Information security in virtualized data center network // Advanced information and communication technologies, AICT–2019 : proceedings of the 3rd International conference (Lviv, Ukraine, July 2–6 2019). – 2019. – C. 419–422

[48]. Sharafaldin, I.; Lashkari, A.H.; Hakak, S.; Ghorbani, A.A. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In Proceedings of the International Carnahan Conference on Security Technology, Chennai, India, 1–3 October 2019.

[49]. Hameed, S., Ali, U. HADEC: Hadoop-based live DDoS detection framework. EURASIP J. on Info. Security 2018, 11 (2018) doi:10.1186/s13635-018-0081-z

[50]. Hameed S. Efficacy of Live DDoS Detection with Hadoop // Hameed S., Ali U. // In Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS) (Istanbul, Turkey, 25–29 April 2016). – 2016. – PP.336-340

[51]. Iswardani A. Denial of service log analysis using density k-means method // A. Iswardani and I. Riadi// Journal of Theoretical and Applied Information Technology. – 2016 . - vol. 83. - no. 2. - PP. 299–302.

[52]. Aborujilah A. Cloud-based DDoS HTTP attack detection using covariance matrix approach // A. Aborujilah and S. Musa // Journal of Computer Networks and Communications. – 2017. - vol. 1. - 8 pages. Article ID 7674594.

[53]. Girma, A. Analysis of DDoS attacks and an introduction of a hybrid statistical model to detect DDoS attacks on cloud computing environment. // Girma, A., Garuba, M., Li, J., Liu, C. // In: IEEE 12th International Conference on Information Technology-New Generations (ITNG). – 2015. - PP. 212–217

[54]. Hu Q. Anomalous user activity detection in enterprise multi-source logs // Hu Q., Tang, B., and Lin, D.// Proc. IEEE Int. Conference on Data Mining Workshops (ICDMW) (Nov. 18-21, New Orleans 2017, LA, USA). – 2017. - PP.797-804

[55]. Landauer M. Dynamic log file analysis: An unsupervised cluster evolution approach for anomaly detection // M. Landauer, M. Wurzenberger, F. Skopik, G. Settanni and P. Filzmoser // Computers & Security Journal.- 2018. - vol. 79. - PP. 94-116.

[56]. A. Dawoud, S. Shahristani and C. Raun, "Deep learning and software-defined networks: Towards secure IoT architecture", Internet of Things, vol. 3-4, pp. 82-89, 2018. Available: 10.1016/j.iot.2018.09.003.

[57]. R. Smith, A. Zincir-Heywood, M. Heywood and J. Jacobs, "Initiating a Moving Target Network Defense with a Real-time Neuro-evolutionary Detector", in In: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion, New York, New York, USA, 2016, pp. 1095–1102.

[58]. H. Zhang, Y. Wang, H. Chen, Y. Zhao and J. Zhang, "Exploring machine-learning-based control plane intrusion detection techniques in software defined optical networks", Optical Fiber Technology, vol. 39, pp. 37-42, 2017. Available: 10.1016/j.yofte.2017.09.023.

[59]. A. Raj, T. Truong-Huu, P. Mohan and M. Gurusamy, "Crossfire Attack Detection using Deep Learning in Software Defined ITS Networks", in Proceedings of 89th Vehicular Technology Conference (VTC2019-Spring), Kuala Lumpur, Malaysia, 2019. Available: 10.1109/VTCSpring.2019.8746594

[60]. H. Lin and P. Wang, "Implementation of an SDN-based security defense mechanism against DDoS attacks", in Proceedings of the 2016 Joint International Conference on Economics and Management Engineering (ICEME 2016) and International Conference on Economics and Business Management (EBM 2016), Pennsylvania, Penn, USA, 2016. Available: 10.12783/dtem/iceme-ebm2016/4183

[61]. J. G. Yang, X. T. Wang, and L. Q. Liu, “Based on traffic and IP entropy characteristics of DDoS attack detection method”, Application Research of Computers, vol. 33, no. 4, pp. 1145–1149, 2016.

[62]. A. Saied, R. Overill and T. Radzik, "Detection of known and unknown DDoS attacks using Artificial Neural Networks", Neurocomputing, vol. 172, pp. 385-393, 2016. Available: 10.1016/j.neucom.2015.04.101.

[63]. R. Braga, E. Mota and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow", in Proceedings of the 35th Annual IEEE Conference on Local Computer Networks (LCN '10), Denver, Colo, USA, 2010, pp. 408–415.

[64]. X. Wang, M. Chen, C. Xing, and T. Zhang " Defending DDoS attacks in software-defined networking based on legitimate source and destination IP address database", IEICE Transaction on Information and Systems, vol. E99D, no. 4, pp. 850–859, 2016.

[65]. Н.В. Пелех, О.М. Шпур, М.М. Климаш, “Модель оцінки ефективності функціонування хмарного центру з високим ступенем віртуалізації та в умовах групового надходження запитів,” Проблеми телекомунікацій, №2(27), с.82-98. 2020.

[66]. M. Klymash, O. Shpur, N. Peleh, I. Lutsiuk "Clustering model of cloud centers for Big Data processing" 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavske, Ukraine, 2018, pp. 268–271

[67]. Ben Alla, S. Gateway and Cluster Head Election using Fuzzy Logic in heterogeneous wireless sensor networks / Ben Alla S., Ezzati A., Mohsen A. // International Conference on Multimedia Computing and Systems (ICMCS).10–12, May 2012. – PP. 761-766.

[68]. Hadjila, M. A Routing Algorithm based on Fuzzy Logic Approach to Prolong the Life-time of Wireless Sensor Networks / Hadjila M., Guyennet H., Feham M // International Journal of Open Scientific Research IJOSR. – Oct. 2013. – Vol.1, № 5. – PP.24-35.

[69]. Jin–Shyan, L. Fuzzy–Logic–Based Clustering Approach for Wireless Sensor Networks Using Energy Predication / Jin–Shyan Lee, Senior Member, Wei–

Liang Cheng // IEEE Sensors Journal. – September 2012. – Vol. 12, № 9. – PP.2891-2897.

[70]. Karimi, A. Cluster Head Selection Using Fuzzy Logic and Chaotic Based Genetic Algo-rithm in Wireless Sensor Network / Karimi A., Abedini S. M., Zarafshan F., Al–Haddad S.A.R. // Journal of Basic and Applied Scientific Research. – 2013. – 3 (4). – PP.694-703.

[71]. Shubham Sharma & Ahmed J. Obaid (2020) Mathematical modelling, analysis and design of fuzzy logic controller for the control of ventilation systems using MATLAB fuzzy logic toolbox, Journal of Interdisciplinary Mathematics, 23:4, 843-849, DOI: 10.1080/09720502.2020.1727611

[72]. J. Zeng and Z.-Q. Liu, "Type-2 fuzzy Markov random fields and their application to handwritten chinese character recognition", IEEE Trans. Fuzzy Syst., vol. 16, no. 3, pp. 747-760, 2008.

[73]. О.А. Лаврів, О.М. Шпур, Н.В. Пелех, “Забезпечення доступу до статичного контенту із використанням CDN мереж як PaaS сервісу Azure Cloud,” Системи обробки інформації - Х: Харк. ун-т Повітр. Сил ім. Івана Кожедуба, №3(158), с. 83-91, 2019.

[74]. Strykhalyuk B., Shpur O., Demydov I., Klymash Yu. “Synthesis of distributed service-oriented structures cloud networks is based on algorithm for determining hyperbolic virtual coordinates”, Proceedings of XIIIth international conference”The experience of designing and application of CAD Systems in microelectronics” CADSM’2015. (24-27 February, Lviv-Poljana, Ukraine), pp. 231-235.

[75]. Ling Li, Xiaozhen Ma, Huang Yulan (2013), “CDN cloud: A novel scheme for combining CDN and cloud computing” Proceedings of 2nd International Conference on Measurement, Information and Control (16-18 Aug. 2013), vol.01, pp.687-690

[76]. Lavriv O., Klymash M., Grynkevych G., Tkachenko O., Vasylenko V. “Method of cloud system disaster recovery based on ‘Infrastructure as a code’ 36

concept” Modern problems of radio engineering, telecommunications, and computer science. Proceedings of the International Conference TCSET’2018 (Lviv-Slavske, Ukraine February 20 – 24, 2018), Publishing House of Lviv Polytechnic, pp. 1139-1142

[77]. Klymash M., Romanchuk V., Beshley M., Arthur P. (2017) “Investigation and simulation of system for data flow processing in multiservice nodes using virtualization mechanisms” IEEE First Ukraine Conference on Electrical and Computer Engineering UKRCON’2017 (Kyiv, May 29 – June 2, 2017), pp. 989-992

[78]. M. Klymash, O. Shpur, N. Peleh, O. Lavriv, R. Bak, O. Skybinskyi "Increasing the accessibility of static content using CDN networks as PaaS," 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), Polyana, Ukraine, 2019, pp. 2/57–2/60.

[79]. Letaifa A., Haji A., Jebalia M., and Tabbane S. (2010), “State of the art and research challenges of new services architecture technologies: virtualization, SOA and cloud computing” International Journal of Grid and Distributed Computing, Science and Engineering Research Support Center, vol. 3, No.4 , pp. 69-88

[80]. Microsoft Azure PaaS [Electronic source]. Available: https://dou.ua/lenta/articles/microsoft-azure-paas/

[81]. Chawathe Y. , McCanne S., Brewer E. (2011), “An architecture for internet content distribution as an infrstructure service” Proceedings of the 11th international workshop on Network and operating systems support for digital audio and video (Port Jefferson, NY, USA June 25 - 26, 2011), pp.11-20

[82]. Gong C., Liu J., Zhang Q., Chen H., and Gong Z. (2010), “The characteristics of cloud computing” 39th International Conference on Parallel Processing Workshops (San Diego, CA, USA, 13-16 September, 2010), pp. 275-279.

[83]. Zhang S., Chen X., and Huo X. (2010) “Cloud computing research and development trend in Future Networks” Proceedings of Second International

Conference on Future Networks (22-24 January, Sanyay, Hainan, China, 2010), pp.93-97

[84]. N. Peleh, O. Shpur, M. Klymash, “Intelligent detection of DDoS attacks in SDN networks”, Lecture Notes in Electrical Engineering., Vol. 831 : Future intent-based networking. On the QoS robust and energy efficient heterogeneous software defined networks, pp. 210–222, 2022.

[85]. https://owasp.org/

[86]. Теорія конкурентних переваг М. Портера. Зовнішньоекономічна діяльність підприємства. – [Електронний ресурс]. – Режим доступу : http://pidruchniki.com/12461220/economika/teoria_konkurentnih_perevag_portera_porters_competitive_advantage_theory

[87]. Климчук А. О. Сучасна парадигма забезпечення конкурентних переваг підприємства / А. О. Климчук // Бізнес Інформ. – 2014. – № 1. – С. 221-225. – [Електронний ресурс]. – Режим доступу : http:nbuv.gov.ua/UJRN/binf_2014_1_42.

[88]. Портер М. Конкуренция. Пер. с англ. – М. : Издательский дом «Вильямс». – 2006. – 806 с.

[89]. T. Mahmood and U. Afzal. Security analytics: Big data analytics for cybersecurity: A review of trends, techniques and tools. In Information Assurance (NCIA), 2013 2nd National Conference on, pages 129–134, Dec 2013.

[90]. A. A. Cardenas, P. K. Manadhata, and S. P. Rajan. Big data analytics for security. IEEE Security & Privacy, 11(6):74–76, 2013.

[91]. M. Cinque, D. Cotroneo, R. Della Corte, and A. Pecchia. Characterizing direct monitoring techniques in software systems. IEEE Transactions on Reliability, 65(4):1665–1681, Dec 2016.

[92]. K. M. Kavanagh, O. Rochford, and T. Bussa. Magic quadrant for security information and event management. Technical report, Gartner Reasearch, 2016.

[93]. A. A. Cardenas, P. K. Manadhata, and S. P. Rajan. Big data analytics for security intelligence. Technical report, Cloud Security Alliance - Big Data Working Group, 2013.

[94]. A. Sharma, Z. Kalbarczyk, J. Barlow, and R. Iyer. Analysis of security data from a large computing organization. In The 41nd IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2011), pages 506–517, 2011.

[95]. J. Cao, B. Yu, F. Dong, X. Zhu, and S. Xu. Entropy-based denial of-service attack detection in cloud data center. Concurrency and Computation: Practice and Experience, 27(18):5623–5639, 2015.

[96]. S. Yu, W. Zhou, R. Doss, and W. Jia. Traceback of ddos attacks using entropy variations. IEEE Transactions on Parallel and Distributed Systems, 22(3):412–425, March 2011.

[97]. K. F. Hong, C. C. Chen, Y. T. Chiu, and K. S. Chou. Scalable command and control detection in log data through of icf analysis. In Security Technology (ICCST), 2015 International Carnahan Conference on, pages 293–298, Sept 2015.

[98]. Y. Liao and V. R. Vemuri. Using text categorization techniques for intrusion detection. In Proceedings of the 11th USENIX Security Symposium, 2002.

[99]. A. D'Amico and K. Whitley. The real work of computer network defense analysts. In VizSEC 2007, pages 19–37. Springer, 2008.

[100]. G. P. Spathoulas and S. K. Katsikas. Reducing false positives in intrusion detection systems. Computers & Security, 29(1):35–44, 2010.

[101]. N. A. Bakar, B. Belaton, and A. Samsudin. False positives reduction via intrusion alert quality framework. In Networks, 2005. Jointly held with the 2005 IEEE 7th Malaysia International Conference on Communication, volume 1, pages 6–pp. IEEE, 2005.

[102]. R. Alshammari, S. Sonamthiang, M. Teimouri, and D. Riordan. Using neuro-fuzzy approach to reduce false positive alerts. In CNSR, pages 345–349. IEEE Computer Society, 2007.

[103]. T. Pietraszek. Using adaptive alert classification to reduce false positives in intrusion detection. In Erland Jonsson, Alfonso Valdes, and Magnus Almgren, editors, RAID, volume 3224 of Lecture Notes in Computer Science, pages 102–124. Springer, 2004.

[104]. X. Fu, J. Shi, and L. Xie. A novel data mining-based method for alert reduction and analysis. Journal of Networks, 5(1), 2010.

[105]. F. Valeur, G. Vigna, C. Kruegel, and R. A. Kemmerer. Comprehensive approach to intrusion detection alert correlation. Dependable and Secure Computing, IEEE Transactions on, 1(3):146–169, 2004.

[106]. D. Cotroneo, A. Paudice, and A. Pecchia. Automated root cause identification of security alerts: Evaluation in a saas cloud. Future Generation Computer Systems, 56:375 – 387, 2016.

[107]. K. Julisch and M. Dacier. Mining intrusion detection alarms for actionable knowledge. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’02, pages 366–375, New York, NY, USA, 2002. ACM.

[108]. P. Giura and W. Wang. Using large scale distributed computing to unveil advanced persistent threats. Science J, 1(3):93–105, 2012.

[109]. X. Shu, J. Smiy, D. Daphne Yao, and H. Lin. Massive distributed and parallel log analysis for organizational security. In 2013 IEEE Globecom Workshops (GC Wkshps), pages 194–199, Dec 2013.

[110]. D. Gonçalves, J. Bota, and M. Correia. Big data analytics for detecting host misbehavior in large logs. In Trustcom/BigDataSE/ISPA, 2015 IEEE, volume 1, pages 238–245, Aug 2015.

[111]. T.-F. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W. Robertson, A. Juels, and E. Kirda. Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks. In Proceedings of the 29th Annual Computer Security Applications Conference, ACSAC ’13, pages 199–208, New York, NY, USA, 2013. ACM.

[112]. C. W. Ten, G. Manimaran, and C. C. Liu. Cybersecurity for critical infrastructures: Attack and defense modeling. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 40(4):853–865, July 2010.

[113]. D. Hadiosmanovic, D. Bolzoni, P. Hartel, and S. Etalle. Melissa: Towards automated detection of undesirable user actions in critical infrastructures. In Computer Network Defense (EC2ND), 2011 Seventh European Conference on, pages 41–48, Sept 2011.

[114]. J. Timonen, L. Lääperi, L. Rummukainen, S. Puuska, and J. Vankka. Situational awareness and information collection from critical infrastructure. In Cyber Conflict (CyCon 2014), 2014 6th International Conference On, pages 157–173, June 2014.

[115]. A. J. Oliner, A. Aiken, and J. Stearley. Alert detection in system logs. In IEEE International Conference on Data Mining, 2008.

[116]. J. Stearley and A. J. Oliner. Bad words: Finding faults in spiritís syslogs. In IEEE International Symposium on Cluster Computing and the Grid, pages 765–770, 2008.

[117]. C. Lim, N. Singh, and S. Yajnik. A Log Mining Approach to Failure Analysis of Enterprise Telephony Systems. In Proc. Intl. Conf. on Dependable Systems and Networks, June 2008.

[118]. M. F. Porter. An algorithm for suffix stripping. In K. Sparck Jones and P. Willett, editors, Readings in Information Retrieval, pages 313–316, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[119]. R. Vaarandi. Mining event logs with slct and loghound. In NOMS 2008 - 2008 IEEE Network Operations and Management Symposium, pages 1071–1074, April 2008.

[120]. A. Pecchia, D. Cotroneo, R. Ganesan, and S. Sarkar. Filtering security alerts for the analysis of a production saas cloud. In Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, UCC ’14. IEEE Computer Society, 2014.

ДОДАТОК А. АКТИ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНИХ ДОСЛІДЖЕНЬ

"ЗАТВЕРДЖУЮ"
Генеральний директор ТОВ «Аркада-Х»
_____ Куранов Ю.В.
"02" _____ 2022 р.

АКТ
про використання результатів дисертаційної роботи
аспіранта кафедри телекомунікацій
Національного університету "Львівська політехніка"
Пелеха Назара Володимировича
на тему:
"Підвищення ефективності функціонування хмарних систем для інформаційно-комунікаційних сервісно-орієнтованих мереж"

Даний акт складений про те, що у ТОВ «Аркада-Х» використані результати дисертаційної роботи Пелеха Н.В. "Підвищення ефективності функціонування хмарних систем для інформаційно-комунікаційних сервісно-орієнтованих мереж ". А саме:

- використання розробленого алгоритму вибору головного кластерного вузла з використанням центральності по діаграмах Вороного, правил нечіткої логіки та моніторингу параметрів вузлів дало змогу зменшити споживання енергії на 45% і продовжити життєвий цикл мережі на 8% та підтримувати якість надання послуг користувачам в умовах обслуговування групових потоків запитів;
- реалізація удосконаленої математичної моделі оцінки ефективності функціонування хмарних систем, яка на відмінну від існуючих враховує можливість обслуговування при груповому надходженні запитів і розподілений час обслуговування запитів та методу кластеризації вузлів хмарних систем в умовах обслуговування групових потоків запитів дало змогу зменшити тривалість завантаження статичного контенту з кеш-серверів у порівнянні із завантаженням з базового сервера в 4 рази.

Результати експериментальних досліджень, виконаних на виробничих потужностях ТОВ «Аркада-Х» відповідають результатам досліджень, що представлені у дисертаційній роботі, похибка не перевищує 3%.

Генеральний директор
ТОВ «Аркада-Х» Куранов Ю. В.

"ЗАТВЕРДЖУЮ"
Директор
ТзОВ «МаксіТех»
_____ Заблоцький С.О.
"_20_" _січня_ 2022р.

АКТ

про використання результатів дисертаційної роботи
Пелеха Назара Володимировича на тему:
"Підвищення ефективності функціонування хмарних систем для інформаційно-комунікаційних сервісно-орієнтованих мереж"

Даний акт складений про те, що у ТзОВ «МаксіТех», для підвищення якості обслуговування користувачів у процесі надання мультисервісних послуг з використанням телекомунікаційних сервісно-орієнтованих мереж, використані результати дисертаційної роботи Пелеха Назара Володимировича "Підвищення ефективності функціонування хмарних систем для інформаційно-комунікаційних сервісно-орієнтованих мереж", представленої на здобуття наукового ступеня доктора філософії, а саме:

- для зменшення втрат внаслідок перевантаження каналів при неочікуваних стрибках інтенсивності навантаження використано модель оцінки ефективності функціонування хмарних систем в умовах обслуговування групових потоків запитів, яка дає змогу обслуговувати користувачів в моменти пікових навантажень на мережу;
- використано комплексний підхід до аналізу лог-журналів із використанням як структурованого так і неструктурованого аналізу службових файлів, що дало змогу виявляти атаки, які вже проводяться на мережі, використовуючи інформацію про стан потоку, тривалість сесії та джерела його походження, та блокувати «подібні» потоки запитів без втручання системних адміністраторів.

Внаслідок перевірки використаних моделей на мережевому обладнанні у ТзОВ «МаксіТех» встановлено, що результати знаходяться в межах п'ятивідсоткового середньоквадратичного відхилення від поданих у дисертаційній роботі.

Директор
ТзОВ «МаксіТех Заблоцький С.О.

ДОДАТОК Б. СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ ТА ВІДОМОСТІ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. N. Peleh, O. Shpur, M. Klymash, “Intelligent detection of DDoS attacks in SDN networks”, Lecture Notes in Electrical Engineering., Vol. 831 : Future intent-based networking. On the QoS robust and energy efficient heterogeneous software defined networks, pp. 210–222, 2022.

2. N. Peleh, S. Zhuravel, O. Shpur, O. Rybytska, “Structured and unstructured log analysis as a methods to detect DDoS attacks in SDN networks,” *Internet of Things (IoT) and Engineering Applications*, pp. 1-9, Sept. 2021.

3. Н.В. Пелех, О.М. Шпур, М.М. Климаш, “Модель оцінки ефективності функціонування хмарного центру з високим ступенем віртуалізації та в умовах групового надходження запитів,” *Проблеми телекомунікацій*, №2(27), с.82-98. 2020.

4. М. М. Климаш, О.М. Шпур, Н.В. Пелех, “Моніторинг доступності веб-сервісу в розподілених інфокомунікаційних системах,” *Вісник Університету "Україна". Серія : Інформатика, обчислювальна техніка та кібернетика.*, №1(28), с.135-150, 2020.

5. О.А. Лаврів, О.М. Шпур, Н.В. Пелех, “Забезпечення доступу до статичного контенту із використанням CDN мереж як PaaS сервісу Azure Cloud,” *Системи обробки інформації - Х: Харк. ун-т Повітр. Сил ім. Івана Кожедуба*, №3(158), с. 83-91, 2019.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

6. O. Shpur, Y. Pyrih, T. Havryliv, N. Peleh, O. Urikova, A. Branytskyi, "Development of a road traffic monitoring system," *2021 IEEE 16th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, Polyana, Ukraine, 2021, pp. 10-14.

7. M. Klymash, O. Shpur, N. Peleh, O. Maksysko "Concept of intelligent detection of DDoS attacks in SDN networks using machine learning," *2020 8th*

International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, 2020, pp. 609-612.

8. M. Klymash, O. Shpur, N. Peleh, S. Hladun "Monitoring of web service availability in distributed infocommunication systems," *2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, Lviv-Slavske, Ukraine, 2020, pp. 723-728.

9. M. Klymash, O. Shpur, O. Lavriv and N. Peleh "Information security in virtualized data center network," *2019 3rd International Conference on Advanced Information and Communications Technologies (AICT)*, Lviv, Ukraine, 2019, pp. 419-422.

10. M. Klymash, O. Shpur, N. Peleh, O. Lavriv, R. Bak, O. Skybinskyi "Increasing the accessibility of static content using CDN networks as PaaS," *2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, Polyana, Ukraine, 2019, pp. 2/57–2/60.

11. M. Klymash, O. Shpur, N. Peleh, I. Lutsiuk "Clustering model of cloud centers for Big Data processing" *2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET)*, Lviv-Slavske, Ukraine, 2018, pp. 268–271

12. М.М. Климаш, О.М. Шпур, Н.В. Пелех "Модель кластеризації хмарних дата-центрів в умовах передачі та захисту потоків Big Data," *І міжнародна науково-практична конференція "Проблеми кібербезпеки інформаційно телекомунікаційних систем"*, м. Київ, 2018 р., с. 191- 197.