КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (24), 2024
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2024.24.376387
УДК 004.056:004.42:621.391.9:378.4

Лахно Валерій Анатолійович
д.т.н., професор, професор кафедри комп'ютерних систем та мереж
Національний університет біоресурсів і природокористування України, Київ, Україна
ORCID ID: 0000-0001-9695-4543
lva964@gmail.com

Москаленко Володимир Володимирович
аспірант
Державний торгово-економічний університет, Київ, Україна
ORCID ID: 0009-0009-3858-915X
vmoskalenko@knute.edu.ua

ОПТИМІЗАЦІЯ ПАРАМЕТРІВ УНІВЕРСИТЕТСЬКИХ ХМАРНИХ СИСТЕМ ДЛЯ ЗАБЕЗПЕЧЕННЯ НАДІЙНОСТІ ТА БЕЗПЕКИ ПРИ ДЕЦЕНТРАЛІЗАЦІЇ ІДЕНТИЧНОСТІ

Анотація. Показано, що мікросервісна архітектура (МСА) хмарних сервісів (ХС) приватної університетської хмари — це підхід до розроблення програмного забезпечення (ПЗ), за якого застосунок будується з невеликих незалежних сервісів, які взаємодіють між собою за допомогою API, причому кожний сервіс відповідає за виконання певної функціональності та може бути розгорнутий і масштабований окремо, що дасть можливість створювати гнучкі й масштабовані застосунки, що можуть швидко адаптуватися до умов, що змінюються, та навантаження для кожної задачі. Проведено огляд і аналіз попередніх досліджень, присвячених використанню МСА в хмарних обчисленнях (ХО) і ХС, який показав, що МСА широко застосовується в ХО і ХС завдяки своїй гнучкості, масштабованості та високій доступності. Показано, що децентралізація ідентичності дасть змогу підвищити безпеку та конфіденційність даних користувачів, тому що кожен сервіс у МСА ХС може мати свою систему управління доступом та ідентифікацією, що, з одного боку, призводить до низки переваг, як-от масштабованість, гнучкість та стійкість до збоїв, але, з іншого боку, децентралізована природа МСА також створює проблеми з управлінням ідентичностями. Отримала подальший розвиток модель для моделювання коефіцієнта завантаження системи під час МСА. Запропоноване в роботі рішення, на відміну від існуючих рішень, враховує обмеження на децентралізоване управління ідентичностями (тобто DID). Описано програмне рішення за допомогою алгоритмічної мови Python для реалізації математичної моделі для моделювання параметрів ХС. У моделі, крім іншого, враховано ваги (важливість) заявок, що надходять у систему децентралізованого управління ідентичностями.

Ключові слова: хмарні обчислення; хмарні сервіси; мікросервісна архітектура; децентралізоване управління ідентичностями.

ВСТУП

Постановка проблеми. Розвиток інформаційних технологій (далі ІТ) і хмарних обчислень (ХО) справив значний вплив на еволюцію архітектури додатків. Одним із результатів цього впливу стала мікросервісна архітектура (або далі МСА), що наразі є одним з основних підходів до побудови хмарних додатків (ХД) і сервісів (ХС) [1] – [3]. При реалізації МСА легше впроваджувати нові технології, оскільки це можна робити поступово, починаючи з одного сервісу, а не переписуючи весь монолітний код. Поступовий відхід від монолітної архітектури, що став трендом, зумовлений перевагами мікросервісів. Монолітні додатки (наприклад, такі як, системи управління ресурсами підприємства (ERP), управління взаємовідносинами з клієнтами (CRM), управління

контентом (CMS), тощо) часто стають важко підтримуваними та масштабованими зі зростанням їхнього розміру та складності. Мікросервісна архітектура, як це буде продемонстровано надалі, надає гнучкіше модульне рішення для побудови застосунків, що краще відповідає сучасним вимогам бізнесу та ринку IT.

Зі зростанням популярності МСА збільшується і важливість проблеми управління ідентичностями, оскільки децентралізований підхід до управління ідентичностями може підвищити безпеку системи і поліпшити її відмовостійкість. Крім того, розробка нових методів і технологій у цій галузі може призвести до поліпшення продуктивності та зручності використання мікросервісних систем для ХО. З огляду на ці фактори останніми роками зросла кількість наукових публікацій, присвячених цій тематиці. З огляду на вищесказане, тематика роботи видається нам актуальною.

Аналіз останніх досліджень і публікацій. Як було показано в [3] – [7], дослідження в галузі децентралізації ідентичності в хмарних обчисленнях (далі ХО) на основі МСА спрямовані на поліпшення безпеки та конфіденційності даних користувачів, зменшення залежності від централізованих служб управління ідентичністю, підвищення зручності використання для користувачів, зниження витрат на управління ідентичністю та стимулювання інновацій у цій галузі.

Згідно з [4] децентралізація ідентичності дасть змогу підвищити безпеку і конфіденційність даних користувачів, оскільки кожен сервіс може мати свою систему управління доступом та ідентифікацією. Як показують автори, у МСА додатки розбиваються на безліч незалежних, слабопов’язаних сервісів. Це призводить до низки переваг, таких як масштабованість, гнучкість і стійкість до збоїв. Однак децентралізована природа МСА також створює проблеми з управлінням ідентичностями.

Як показано в [3], [8], перехід до децентралізованої ідентичності дасть змогу зменшити залежність від централізованих служб управління ідентичністю, таких як центри автентифікації ідентичності (Identity Providers), що підвищує відмовостійкість і знижує ризики централізованих атак. На думку авторів, децентралізований підхід до управління ідентичностями дасть змогу кожному мікросервісу керувати своїми власними ідентифікаційними даними. Це підвищить безпеку і масштабованість системи.

Згідно з [9] децентралізовані системи ідентичності можуть надавати користувачам більше контролю над своїми даними та доступом до сервісів, що може поліпшити їхню зручність використання. Автори показують, що традиційні підходи до управління ідентичностями, як-от централізовані служби аутентифікації та авторизації, не підходять для МСА. Це пов’язано з тим, що вони створюють єдину точку відмови, що може призвести до серйозних проблем безпеки.

У [10], [11] автори показали, що використання децентралізованих систем ідентичності може знизити витрати на управління ідентичністю за рахунок зменшення необхідності в централізованих системах і службах.

Також зазначимо, що, як показали автори досліджень [12] – [14], децентралізація ідентичності може стимулювати розвиток нових технологій і підходів до управління ідентичністю, що може призвести до появи нових інноваційних рішень у сфері ХО.

Згідно з [12] підхід МСА дасть змогу використовувати різні технології та мови програмування для кожного сервісу залежно від його специфіки. Це збільшить гнучкість у виборі технологій і дасть змогу використовувати найбільш підходящі інструменти для кожного завдання.

У [15] автори створили схему розподіленого контролю доступу для взаємодії мікросервісів на основі блокчейна. Вони розробили два смарт-контракти для зберігання авторизованих політик мікросервісів у блокчейні для підтримки розподіленого та

узгодженого управління політиками безпеки. На основі моделі розподіленого доступу авторами було запропоновано схему ухвалення рішень на основі графів для реалізації проміжного представлення комплексною кооперацією сервісів та ефективного контролю доступу.

Мета статті. Розвиток моделі для моделювання коефіцієнта завантаження системи за використання мікросервісної архітектури.

МЕТОДИКА ДОСЛІДЖЕННЯ

Подальші міркування ми наводимо для конкретного випадку, в якому розглядаємо розв'язання задачі моделювання коефіцієнта завантаження системи при МСА хмарних сервісів для приватної університетської хмари. Такий вибір зумовлений популярністю таких хмарних рішень для університетів та інших навчальних закладів.

Будемо вважати, що підтримка мультизадачності в МСА університетської хмари посприяє вибудовуванню гнучкішої, відмовостійкої та продуктивнішої системи, що важливо для ефективної реалізації децентралізації ідентичності. Для подальших математичних викладок введемо такі позначення:

$i = 1,2,...,I$ – номери груп завдань, які вирішуються в ХС університетської хмари на основі МСА та децентралізованого підходу до управління ідентичностями. Наприклад, до таких завдань можна віднести (наведемо лише кілька прикладів): систему управління навчанням (LMS), у якій мікросервіси відповідатимуть за окремі функції, як-от реєстрація на курси, управління оцінками, надання навчальних матеріалів і проведення онлайн-тестів, причому децентралізована ідентифікація дасть змогу студентам і викладачам безпечно отримувати доступ до своїх даних та керувати ними; віртуальні лабораторії, що надають студентам доступ до віртуальних машин (ВМ) та ПЗ для проведення експериментів і виконання завдань, мікросервіси можуть керувати виділенням ресурсів, а децентралізована ідентифікація забезпечить безпечний доступ до лабораторій; індивідуальні освітні траєкторії, в яких платформа, що адаптує навчальний процес до потреб кожного студента, базується на МСА, та мікросервіси відповідають за аналіз даних, рекомендації по курсах та персоналізацію контенту, а децентралізована ідентифікація дасть змогу студентам контролювати доступ до своїх даних та ділитися ними з обраними сервісами та ін.

$j = 1,2,...,J$ – номери видів ресурсів, які необхідні для розв'язання задач $i = 1,2,...,I$, що розв'язуються в ХС університетської хмари на основі МСА та децентралізованому підході до управління ідентичностями. Наведемо кілька прикладів таких видів ресурсів: сервери, що необхідні для запуску мікросервісів та обробки запитів користувачів (можуть бути фізичними або віртуальними машинами); контейнери, що забезпечують ізольоване середовище для запуску мікросервісів та спрощують розгортання й управління мікросервісами; балансувальники навантаження, що розподіляють запити користувачів між мікросервісами; бази даних, що зберігають дані, які використовують мікросервіси, тощо.

M_{ij}^F і M_{ij}^C – відповідно, матриці, в яких відображено фактичний і розрахунковий розподіли ресурсів МСА ХС. Наприклад, матриця фактичного розподілу ресурсів M_{ij}^F, відображатиме поточний розподіл ресурсів (процесорний час, пам'ять, мережа тощо) між мікросервісами.

Для того, щоб врахувати децентралізований підхід до управління ідентичностями в матрицях M_{ij}^{F} та M_{ij}^{C}, можна додати стовпець «Ідентифікатори». Тоді, в цьому стовпчику ми вкажемо які типи ідентифікаторів (наприклад, студенти, викладачі, співробітники) мають доступ до кожного мікросервісу в ХС університетської хмари. Така модифікація матриць, на наш погляд, дасть змогу коректно оцінити навантаження на систему управління ідентифікацією. Наприклад, якщо мікросервіс LMS використовується великою кількістю студентів, це може потребувати виділення додаткових ресурсів для системи управління ідентифікацією.

Для врахування децентралізованого підходу до управління ідентичностями в матрицях фактичного та розрахункового розподілу ресурсів M_{ij}^{F} та M_{ij}^{C}, мікросервісів в ХС приватного університету, можна додати параметри, що відображають специфіку управління ідентичностями.

Зауважимо, що важливим є і врахування ваг (важливості) групи заявок, що надходять до мікросервісів — $E_i \in (0;1)$ і $E_j \in (0;1)$ — відповідно, вага (важливість) групи заявок (i) і ваги (важливість) групи ресурсів (j), (наприклад, CPU, Пам'ять, Мережа, Ідентифікатори), що визначається як відношення вартості конкретного ресурсу, наприклад, CPU, до вартості володіння всією хмарною системою). У наших міркуваннях вважаємо, що врахування ваг (важливості) заявок, що надходять до мікросервісів, зумовлене низкою причин. По-перше, мікросервіси з більш високою вагою заявок можуть потребувати більшої кількості ресурсів (наприклад, процесорного часу, пам'яті, мережі), ніж мікросервіси з більш низькою вагою заявок. Отже, врахування ваги заявки дасть змогу ефективніше розподіляти ресурси між мікросервісами. По-друге, заявки з більш високими вагами можуть бути більш критичними для користувачів або мати більш високі вимоги до продуктивності. Відповідно, врахування ваги заявок дасть змогу мікросервісам пріоритезувати опрацювання більш важливих заявок, що призведе до поліпшення якості обслуговування в університетській хмарі. І, нарешті, третій аргумент, можна сформулювати так — якщо мікросервіс з високою вагою заявок виходить з ладу, це може мати серйозні наслідки для всієї системи, відповідно, врахування ваги заявок дасть змогу вжити заходів для забезпечення відмовостійкості таких мікросервісів, наприклад, шляхом їх дублювання.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Вважаємо, що як критерій оптимальності МСА ХС доцільно використовувати коефіцієнт завантаження системи, (SL) [16] – [18], який визначається як частка ресурсів, задіяна в обслуговуванні завдань на основі МСА в ХС університетської хмари. І при цьому (SL) може врахувати і реалізацію децентралізованого підходу до управління ідентичностями, оскільки цей чинник закладено в матрицях M_{ij}^{F} і M_{ij}^{C}, в яких враховано децентралізований підхід до управління ідентичностями.

З урахуванням робіт [16], [17], ми опишемо такою аналітичною залежністю з урахуванням раніше прийнятих позначень:

$$(SL) = \left(1 / I \cdot J\right) \cdot \left(\sum_{i=1}^{I} \sum_{j}^{J} M_{ij}^{F} \cdot E_i \cdot E_j \Big/ \sum_{i=1}^{I} \sum_{j}^{J} M_{ij}^{C} \right) \to 1. \quad (1)$$

Згідно із залежністю (1) заявки поділяються на групи за пріоритетом, тобто термінові заявки, важливі заявки, звичайні заявки. Наприклад, термінові заявки вимагають миттєвого виконання через їхню критичність для функціонування системи або для забезпечення безпеки. Це можуть бути запити на аварійне відновлення доступу до даних, якщо доступ було втрачено або скомпрометовано.

Нижче наведено формалізовані обмеження, які дадуть змогу точніше врахувати ресурси, що використовуються системою.

Спочатку розглянемо обмеження, що стосується вартості володіння ХС, які базуються на МСА — *Cost*. Оскільки МСА передбачає розбиття системи на безліч незалежних сервісів (i) , кожен сервіс може мати свої вимоги до ресурсів $\left(Cost_i^r\right)$ і, отже, свою вартість. Без обмеження вартості, цільова функція (1) може призвести до неоптимального розподілу ресурсів, що збільшить загальні витрати на володіння системою. Обмеження вартості дасть змогу знайти компроміс між продуктивністю і витратами, забезпечуючи рентабельність системи. Мікросервіси дозволяють легко масштабувати окремі частини системи, наприклад, приватної університетської хмари. Однак, необмежене масштабування може призвести до неконтрольованого зростання витрат, відповідно, обмеження вартості допоможе контролювати витрати при масштабуванні системи. Зауважимо, що існує безліч ХС з різними моделями ціноутворення, відповідно, врахування обмеження вартості дасть змогу обрати найбільш рентабельні сервіси для кожного мікросервісу, оптимізуючи загальні витрати. На підставі вище викладеного можемо записати це обмеження таким чином: $Cost_i^c \le Cost_i^r$. Для цього і наступних обмежень використовуватимемо такі індекси: "c","r" — відповідно, розрахункове і необхідне значення за кожним із критеріїв.

Далі розглянемо важливість використання в системі обмежень, обмеження, що стосується децентралізованого підходу до управління ідентичностями в хмарній системі університету та її ХС, що базуються на МСА. Як було показано вище, децентралізоване управління ідентичностями (тобто DID) зменшить ризик компрометації даних за рахунок відсутності єдиної точки відмови. Замість зберігання всіх даних у централізованій базі даних, DID розподілить інформацію про користувачів університетської хмари по декількох вузлах. Це особливо важливо для університетів, які зберігають конфіденційну інформацію про студентів і співробітників. Крім того, DID краще масштабується, ніж централізовані системи, оскільки не покладається на один сервер для аутентифікації користувачів. Для університетів це також важливо, оскільки, вони мають досить велику кількість користувачів і потребують системи, здатної впоратися з піковими навантаженнями, наприклад, у період сесій. Як ми зазначали в минулому розділі роботи DID сприяє спрощенню інтеграції із зовнішніми сервісами, оскільки не вимагає від університету надавати цим зовнішнім сервісам доступ до своєї централізованої бази даних, що, відповідно, дасть змогу швидше впроваджувати нові сервіси та технології. Останнім наведемо аргумент, який стосується відповідності нормативним вимогам, оскільки DID може допомогти університетам суворо відповідати нормативним вимогам, таким як GDPR, які вимагають від організацій захисту персональних даних. Таким чином, справедливим є наступне обмеження: $DID^c \le DID^r$.

Аналогічно, проаналізуємо, чому для цільової функції важливо під час використання цільової функції (1) врахувати обмеження щодо використання ресурсів процесора $\left(CPUR\right)$ за групами завдань, зокрема з урахуванням децентралізованого

підходу до управління ідентичностями в хмарній системі університету та її ХС, які ґрунтуються на МСА.

Університетські хмарні системи часто обслуговують безліч різних завдань, таких як навчання, дослідження, адміністрування навчальним процесом тощо. Кожна група завдань може мати різні вимоги до ресурсів процесора, відповідно, обмеження використання ресурсів процесора за групами завдань дасть змогу ефективно розподілити ресурси та забезпечити належну продуктивність для всіх завдань, а додатково знизить витрати, оскільки хмарні сервіси зазвичай оплачуються залежно від використаних ресурсів. Якщо говорити про зв'язок із попереднім обмеженням, то зауважимо, що децентралізоване управління ідентичностями (DID) може не лише підвищити безпеку хмарної інфраструктури (далі — ХІнф) університету загалом, а й також збільшить навантаження на процесор. Логічно, що врахування обмеження використання ресурсів процесора допоможе на етапах моделювання впевнитися, що DID не погіршить продуктивність системи та не зробить її більш вразливою для атак зловмисників. Також слід зазначити і той факт, що обмеження використання ресурсів процесора за групами завдань може посприяти більш справедливому розподілу ресурсів між різними користувачами та завданнями в університетській хмарі, де багато користувачів можуть одночасно виконувати ресурсомісткі завдання. Тоді справедливо: $CPUR_i^c \in \left(CPUR_i^{\min} ... CPUR_i^{\max}\right) \cdot k_{res}^{CPUR}$, де max, min позначають верхню та нижню межі допустимих значень критерію, відповідно (справедливо і для інших обмежень, що будуть наведені нижче); k_{res}^{CPUR} — коефіцієнт, застосований для врахування показників резервування ресурсів, щоб гарантувати безперервну роботу ХІнф, беручи до уваги вимоги до надійності та безпеки, зокрема, з урахуванням децентралізації управління ідентичностями (аналогічно цей коефіцієнт використовують і для інших обмежень, що будуть наведені нижче); коефіцієнт, що застосовується для врахування показників резервування ресурсів, щоб гарантувати безперервну роботі, беручи до уваги вимоги до надійності та безпеки, зокрема, з урахуванням децентралізації управління ідентичностями (аналогічно цей коефіцієнт використовується і для інших обмежень, що будуть наведені нижче).

Розглянемо аргументи, які дають змогу включити до переліку обмежень обмеження за ресурсами мережі NET. Мережа є критично важливим ресурсом для будь-якої хмарної системи, і університетська мережа не є винятком. Недостатня пропускна здатність мережі може призвести до зниження продуктивності системи в цілому. Обмеження використання ресурсів мережі допоможе забезпечити належну продуктивність для всіх користувачів і додатків, задіяних у роботі відповідних процесів, які ми згадували раніше. Своєю чергою, децентралізоване управління ідентичностями (DID) може збільшити кількість мережевих з'єднань у хмарній системі університету. Відповідно, обмеження використання ресурсів мережі допоможе дослідити на стадії проєктування або модернізації мережу, з тим, щоб запобігти перевантаженню мережі та зробити її більш стійкою до атак. Тоді справедливо: $NET_i^c \in \left(NET_i^{\min} ... NET_i^{\max}\right) \cdot k_{res}^{NET}$.

Аналогічні міркування справедливі і для обмежень за ресурсами пам'яті для хмарної системи університету — HDD, RAM. Нестача пам'яті (як постійної, так і оперативної) може призвести до зниження продуктивності системи, збоїв і втрати даних. DID може збільшити кількість даних, які необхідно зберігати та обробляти в ХС. Таким чином, можна за $HDD_i^c \in \left(HDD_i^{\min} ... HDD_i^{\max}\right) \cdot k_{res}^{HDD}$, $RAM_i^c \in \left(RAM_i^{\min} ... RAM_i^{\max}\right) \cdot k_{res}^{RAM}$.

Насамкінець розглянемо два останні обмеження на цільову функцію (1). А саме — необхідний ступінь надійності системи (REL) і необхідний ступінь безпеки даних без урахування використання (DID) — (SAF).

Як було показано вище, університетські ХС зберігають і обробляють великі обсяги даних, включно з критично важливою інформацією, такою як дані про дослідження та адміністрування. Відповідно, надійність ХС має вирішальне значення для забезпечення доступності цих даних і безперебійної роботи ХІнф. Обмеження за ступенем безпеки в цільовій функції допоможе забезпечити, упевненість, що систему спроєктовано і реалізовано таким чином, щоб максимально захистити дані від загроз. Зауважимо, що ступінь безпеки даних загалом і децентралізований підхід до управління ідентичностями (тобто DID) — це не одне й те саме. DID — лише один зі способів підвищення безпеки даних, але він не є єдиним і не гарантує повну безпеку, наприклад, під час проведення атаки типу DDoS. Цільова функція (1) в ідеалі повинна враховувати всі аспекти безпеки даних, включно з такими заходами, як шифрування, контроль доступу та аудит кібербезпеки.

На підставі вищенаведеного можемо записати формалізоване подання цих обмежень так: $REL^c \le REL^r$, $SAF^c \le SAF^r$.

Для визначення необхідних критеріїв безпеки даних (SAF) можна, наприклад, використовувати такі дослідження та нормативні документи [19].

Важливо, що виконання вимог до якості системи залежить від організаційної моделі ХІнф університету та характеристик обладнання. На швидкість операцій впливають такі фактори, як метод віртуалізації, організація розподіленої файлової системи та інші.

Усі перелічені вище обмеження можна подати у вигляді такої матриці, яка дає змогу врахувати можливі варіанти розподілу ресурсів під час обслуговування заявок в ХІнф університету, зокрема й використання децентралізованого підходу до управління ідентичностями.

$$M = \begin{pmatrix} Cost_{11} & Cost_{12} & \dots & Cost_{1I} \\ DID_{21} & DID_{22} & \dots & DID_{2I} \\ CPUR_{31} & CPUR_{32} & \dots & CPUR_{3I} \\ NET_{41} & NET_{42} & \dots & NET_{4I} \\ HDD_{51} & HDD_{52} & \dots & HDD_{5I} \\ RAM_{61} & RAM_{62} & \dots & RAM_{6I} \\ REL_{71} & REL_{72} & \dots & REL_{7I} \\ SAF_{81} & SAF_{82} & \dots & SAF_{JI} \end{pmatrix}. \quad (2)$$

Для перевірки працездатності моделі ми провели серію обчислювальних експериментів. Нижче описано програмне рішення за допомогою алгоритмічної мови Python (інтерпретатор Replit) для реалізації моделювання параметрів ХС університету. У моделі, крім іншого, враховано ваги (важливість) заявок, що надходять до мікросервісів, а також додавання серверів аутентифікації.

ISSN 2663 - 4023

ОБЧИСЛЮВАЛЬНИЙ ЕКСПЕРИМЕНТ

Для додавання серверів автентифікації, наприклад, OAuth 2.0, в обробку запитів автентифікації, ми реалізували на Python наступний код, щоб врахувати цей ресурс. Результати моделювання показано на рис. 1.

```
import matplotlib.pyplot as plt
def main():

# Введення даних
      num_groups = 3
      num_resources = 4 # Додаємо сервери аутентифікації як
додатковий ресурс

# Створення матриць
      mov_matrix = [[0 for _ in range(num_resources)] for _ in
range(num_groups)]
      m_matrix = [[0 for _ in range(num_resources)] for _ in
range(num_groups)]
      v_groups = [0 for _ in range(num_groups)]
      v_resources = [0 for _ in range(num_resources)]

# Визначення обмежень щодо ресурсів
      Вартість = 1000
      Proci = [[2, 4, 6, 8], [1, 3, 5, 7], [3, 6, 9, 12]]  #
Приблизні значення для логічних ядер процесора
      RAMi = [8, 16, 32, 64] # Обсяг оперативної пам’яті в ГБ
      HDDi = [500, 1000, 2000, 4000] # Обсяг постійної пам’яті ГБ
      Neti = [100, 200, 300, 400] # Пропускна спроможність мережі
в Мбіт/с
      Reli = [0.95, 0.96, 0.97, 0.98] #Необхідний ступінь
надійності
      Safi = [1, 2, 3, 4] # Необхідний ступінь захисту

     # Введення ваги груп заявок
      v_groups = [0.5, 0.6, 0.7]

# Введення ваги типів ресурсів
      v_resources = [0.5, 0.6, 0.7, 0.8] # Враховуємо новий ресурс
      #Введення матриці фактичного розподілу
      mov_matrix = [[0.4, 0.3, 0.5, 0.2], [0.3, 0.2, 0.4, 0.1],
[0.5, 0.4, 0.6, 0.3]]
      # Введення матриці розрахункового розподілу
      m_matrix = [[0.2, 0.3, 0.4, 0.1], [0.3, 0.4, 0.5, 0.2],
[0.4, 0.5, 0.6, 0.3]]

 # Розрахунок коефіцієнтів завантаження
      k_values = []
      for i in range(num_groups):
          for j in range(num_resources):
              k_values.append(mov_matrix[i][j] / m_matrix[i][j])

# Побудова гістограми
      plt.hist(k_values, bins=10, alpha=0.75, label="Коефіцієнт
завантаження")
      plt.xlabel("Значення")
      plt.ylabel("Частота")
      plt.legend()
      plt.show()

if __name__ == "__main__":
      main()
```

ISSN 2663 - 4023

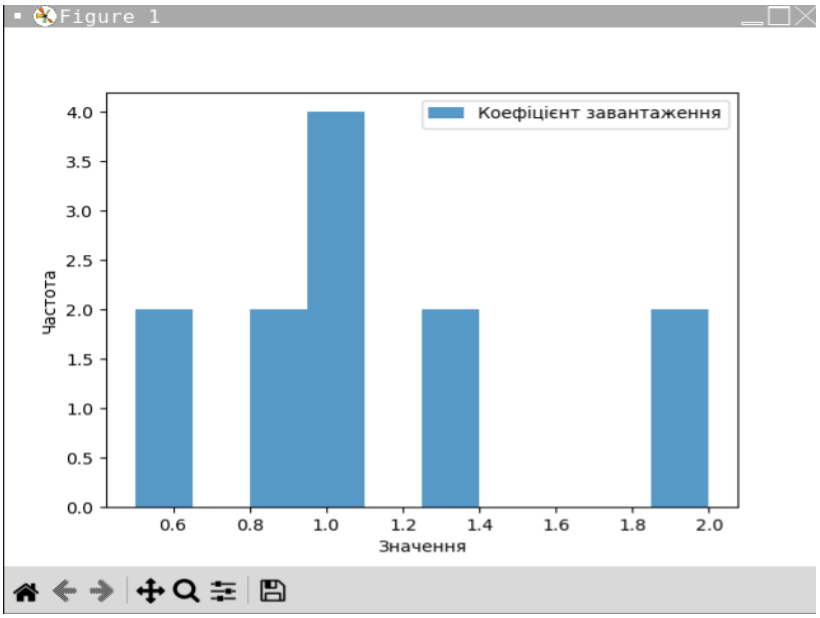

Рис. 1. Результати моделювання коефіцієнта завантаження системи (SL) з урахуванням додавання серверів аутентифікації

ОБГОВОРЕННЯ РЕЗУЛЬТАТІВ МОДЕЛЮВАННЯ

На рис. 1 на гістограмі показано розподіл коефіцієнта завантаження хмарної системи університету. Кожен стовпчик гістограми являє собою частоту повторюваності певного значення коефіцієнта завантаження з урахуванням додавання серверів аутентифікації.

Таким чином, під час дослідження обґрунтовано, що розвиток математичної моделі для моделювання коефіцієнта завантаження хмарної системи з урахуванням додавання ресурсів, таких як сервери автентифікації, є розвитком методу формалізації задачі оптимізації в контексті застосування хмарних сервісів на мікросервісній архітектурі. Показано, що такий розвиток моделі дає змогу врахувати специфіку хмарних сервісів, тому що мікросервісна архітектура характеризується високою гнучкістю та масштабованістю, а додавання нових ресурсів, як-от сервери автентифікації, може бути частим і потребуватиме швидкого й ефективного керування завантаженням системи.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Під час виконаних досліджень, набула подальшого розвитку модель для моделювання коефіцієнта завантаження хмарної системи за мікросервісної архітектури хмарних сервісів, яка, на відміну від наявних рішень, враховує обмеження на децентралізоване управління ідентичностями (тобто DID), що зменшить ризик компрометації даних за рахунок відсутності єдиної точки відмови.

Описано програмне рішення за допомогою алгоритмічної мови Python (інтерпретатор Replit) для реалізації математичної моделі для моделювання параметрів ХС. У моделі, крім іншого, враховано ваги (важливість) заявок, що надходять у систему децентралізованого управління ідентичностями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Campeanu, G. (2018). A mapping study on microservice architectures of Internet of Things and cloud computing solutions. *In 7th Mediterranean conference on embedded computing (MECO)*, 1–4.
2. Singh, V., & Peddoju, S. K. (2017). Container-based microservice architecture for cloud applications. In *International Conference on Computing, Communication and Automation (ICCCA)*, 847–852.
3. Gopalakrishnan, A. (2009). Cloud computing identity management. *SETLabs briefings, 7(7),* 45–55.
4. Yang, Y., Chen, X., Chen, H., & Du, X. (2018). Improving privacy and security in decentralizing multi-authority attribute-based encryption in cloud computing. *IEEE Access, 6,* 18009–18021.
5. Safaryan, O., Pinevich, E., Roshchina, E., Cherckesova, L., & Kolennikova, N. (2020). Information system development for restricting access to software tool built on microservice architecture. *E3S Web Conf., 224.* https://doi.org/10.1051/e3sconf/202022401041
6. Indrasiri, K., Siriwardena, P., Indrasiri, K., & Siriwardena, P. (2018). Microservices security fundamentals. *Microservices for the Enterprise: Designing, Developing, and Deploying,* 313–345.
7. Mostafa, A. M., Rushdy, E., Medhat, R., & Hanafy, A. (2023). An identity management scheme for cloud computing: Review, challenges, and future directions. *Journal of Intelligent & Fuzzy Systems,* 1–23.
8. Gopalakrishnan, A. (2009). Cloud computing identity management. *SETLabs briefings, 7(7),* 45–55.
9. Chen, J., Wu, X., Zhang, S., Zhang, W., & Niu, Y. (2012). A decentralized approach for implementing identity management in cloud computing. In *Second International Conference on Cloud and Green Computing*, 770–776.
10. Saini, S., & Mann, D. (2014). Identity management issues in cloud computing. *arXiv preprint arXiv:1406.1033.*
11. Palson Kennedy, R., & Gopal, T. V. (2010). Assessing the risks and opportunities of cloud computing—defining identity management systems and maturity models. In *Trendz in Information Sciences & Computing (TISC2010),* 138–142.
12. Samir, E., Wu, H., Azab, M., Xin, C., & Zhang, Q. (2021). DT-SSIM: A decentralized trustworthy self-sovereign identity management framework. *IEEE Internet of Things Journal, 9(11),* 7972–7988.
13. Takaoğlu, M., Dursun, T., Doğan, A., Er, H., Bozkurt Günay, B., Emeç, C., & Özcandan, N. (2023). The Impact of Self-Sovereign Identities on CyberSecurity. *IST-186-RSM, Specialist Meeting, Blockchain Technology for Coalition Operations.*
14. Čučko, Š., & Turkanović, M. (2021). Decentralized and self-sovereign identity: Systematic mapping study. *IEEE Access, 9,* 139009–139027.
15. Xi, N., Liu, J., Li, Y., & Qin, B. (2023). Decentralized access control for secure microservices cooperation with blockchain. *ISA transactions, 141,* 44–51.
16. Ramezani, F., Lu, J., Taheri, J., & Zomaya, A. Y. (2017). A multi-objective load balancing system for cloud environments. *The Computer Journal, 60(9),* 1316–1337.
17. Al-Yarimi, F. A. M., Althahabi, S., & Eltayeb, M. M. (2022). Optimal Load Balancing in Cloud Environment of Virtual Machines. Comput. *Syst. Sci. Eng., 41(3),* 919–932.
18. Adhikari, J., & Patil, S. (2012). Load balancing the essential factor in cloud computing. *International Journal of Engineering Research & Technology (IJERT), 1(10),* 1–5.
19. Documentation, T. P. S., & logical, C. (2005). *Information technology–Security techniques–Information security management systems–Requirements.*

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (24), 2024
ISSN 2663 - 4023

Valery Lakhno
Doctor of Technical Sciences, Professor of Department of Computer Systems and Networks
National University of Life and Environmental Sciences of Ukraine Kyiv, Ukraine
ORCID ID: 0000-0001-9695-4543
lva964@gmail.com

Volodymyr Moskalenko
PhD student
State University of Trade and Economics, Kyiv, Ukraine
ORCID 0009-0009-3858-915X
vmoskalenko@knute.edu.ua

OPTIMIZING PARAMETERS OF UNIVERSITY CLOUD SYSTEMS TO ENSURE RELIABILITY AND SECURITY IN IDENTITY DECENTRALIZATION

Abstract. It is shown that microservice architecture (MSA) of cloud services (CS) of a private university cloud is an approach to software development (SW), according to which an application is built from small ones independent services that interact with each other through an APIs, with each service responsible for execution specific functionality and can be deployed and scaled separately, thereby enabling the creation of flexible and scalable applications that can quickly adapt to changing conditions and workloads for each task of using microservice architecture. A review and analysis of previous studies devoted to the use of MCA in cloud computing (CC) and CS was conducted, which showed that CMA is widely used in CC and CS due to its flexibility, scalability and high availability. It is shown that the decentralization of identity will make it possible to increase the security and privacy of user data, because each service in the MSA CS can have its own access and identity management system, which, on the one hand, leads to a number of advantages, such as scalability, flexibility and resistance to failures, but on the other hand, the decentralized nature of the MSA also creates problems with identity management. The model for simulating the system load factor during the MSA was further developed. The solution proposed in the work, unlike existing solutions, takes into account the limitations of decentralized identity management (i.e., DID). A software solution using the Python algorithmic language for the implementation of a mathematical model for the simulation of CS parameters is described. The model, among other things, takes into account the weights (importance) of applications entering the decentralized identity management system.

Keywords: cloud computing; cloud services; microservice architecture; decentralized identity management.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Campeanu, G. (2018). A mapping study on microservice architectures of Internet of Things and cloud computing solutions. *In 7th Mediterranean conference on embedded computing (MECO),* 1–4.
2. Singh, V., & Peddoju, S. K. (2017). Container-based microservice architecture for cloud applications. In *International Conference on Computing, Communication and Automation (ICCCA),* 847–852.
3. Gopalakrishnan, A. (2009). Cloud computing identity management. *SETLabs briefings, 7(7),* 45–55.
4. Yang, Y., Chen, X., Chen, H., & Du, X. (2018). Improving privacy and security in decentralizing multi-authority attribute-based encryption in cloud computing. *IEEE Access, 6,* 18009–18021.
5. Safaryan, O., Pinevich, E., Roshchina, E., Cherckesova, L., & Kolennikova, N. (2020). Information system development for restricting access to software tool built on microservice architecture. *E3S Web Conf., 224.* https://doi.org/10.1051/e3sconf/202022401041
6. Indrasiri, K., Siriwardena, P., Indrasiri, K., & Siriwardena, P. (2018). Microservices security fundamentals. *Microservices for the Enterprise: Designing, Developing, and Deploying,* 313–345.
7. Mostafa, A. M., Rushdy, E., Medhat, R., & Hanafy, A. (2023). An identity management scheme for cloud computing: Review, challenges, and future directions. *Journal of Intelligent & Fuzzy Systems,* 1–23.
8. Gopalakrishnan, A. (2009). Cloud computing identity management. *SETLabs briefings, 7(7),* 45–55.

9. Chen, J., Wu, X., Zhang, S., Zhang, W., & Niu, Y. (2012). A decentralized approach for implementing identity management in cloud computing. In *Second International Conference on Cloud and Green Computing*, 770–776.
10. Saini, S., & Mann, D. (2014). Identity management issues in cloud computing. *arXiv preprint arXiv:1406.1033*.
11. Palson Kennedy, R., & Gopal, T. V. (2010). Assessing the risks and opportunities of cloud computing—defining identity management systems and maturity models. In *Trendz in Information Sciences & Computing (TISC2010)*, 138–142.
12. Samir, E., Wu, H., Azab, M., Xin, C., & Zhang, Q. (2021). DT-SSIM: A decentralized trustworthy self-sovereign identity management framework. *IEEE Internet of Things Journal, 9(11),* 7972–7988.
13. Takaoğlu, M., Dursun, T., Doğan, A., Er, H., Bozkurt Günay, B., Emeç, C., & Özcandan, N. (2023). The Impact of Self-Sovereign Identities on CyberSecurity. *IST-186-RSM, Specialist Meeting, Blockchain Technology for Coalition Operations.*
14. Čučko, Š., & Turkanović, M. (2021). Decentralized and self-sovereign identity: Systematic mapping study. *IEEE Access, 9,* 139009–139027.
15. Xi, N., Liu, J., Li, Y., & Qin, B. (2023). Decentralized access control for secure microservices cooperation with blockchain. *ISA transactions, 141,* 44–51.
16. Ramezani, F., Lu, J., Taheri, J., & Zomaya, A. Y. (2017). A multi-objective load balancing system for cloud environments. *The Computer Journal, 60(9),* 1316–1337.
17. Al-Yarimi, F. A. M., Althahabi, S., & Eltayeb, M. M. (2022). Optimal Load Balancing in Cloud Environment of Virtual Machines. Comput. *Syst. Sci. Eng., 41(3),* 919–932.
18. Adhikari, J., & Patil, S. (2012). Load balancing the essential factor in cloud computing. *International Journal of Engineering Research & Technology (IJERT), 1(10),* 1–5.
19. Documentation, T. P. S., & logical, C. (2005). *Information technology–Security techniques–Information security management systems–Requirements.*