УДК 519.8

<https://doi.org/10.17721/1812-5409.2021/2.5>

Горбачук В. М.¹, д.ф.-м.н., с.н.с., зав. відділу,
Бардадим Т. О.¹, к.ф.-м.н., с.н.с., с.н.с.,
Осипенко С.П.¹, магістр, інженер-програміст

V. M. Gorbachuk¹, DSc,
T. O. Bardadaym¹, PhD,
S. P. Osypenko¹

Задача децентралізованого прийняття рішень для сучасних хмарних послуг

The problem of decentralized decision making for modern cloud services

¹ Інститут кібернетики імені В.М. Глушкова
НАН України, 03187, м. Київ, проспект
Глушкова, 40,
e-mail: GorbachukVasyl@netscape.net

¹ V.M. Glushkov Institute of Cybernetics of the NAS
of Ukraine, 03187, Kyiv, Glushkov Avenue, 40,
e-mail: GorbachukVasyl@netscape.net

Організація сучасних хмарних послуг спирається як на теоретичні результати з логістики, дослідження операцій, ланцюгів постачання, мереж передачі (транспортування) інформації, так і на практичні досягнення новітніх інформаційно-комунікаційних технологій. Оскільки всі мешканці планети стають постійними користувачами і водночас творцями таких послуг, то питання децентралізованого прийняття рішень стають повсякденними. У роботі викладається постановка проблеми таких рішень з боку постачальників (провайдерів) хмарних послуг і пропонується математичне формулювання відповідної задачі оптимізації з ресурсними обмеженнями.

Результати дослідження доповідались на Міжнародній науковій конференції "Modern Stochastics: Theory and Applications. V" (MSTA-V).

Ключові слова: багатокритеріальна оптимізація, відображення, мережа.

The organization of modern cloud services is based on theoretical results in logistics, operations research, supply chains, information transmission (transportation) networks, and on the practical achievements of the novel information and communication technologies. As all the inhabitants of the planet become regular users and at the same time creators of such services, the issues of decentralized decision making are becoming everyday problems. The paper presents the setup for the problem of such solutions by suppliers (providers) of cloud services and suggests a mathematical formulation of the corresponding optimization problem with resource constraints. It is a starting point for further mathematical elaboration of the new everyday problems.

Key words: multicriteria optimization, mapping, network.

Ефективне забезпечення ресурсами є складною проблемою в середовищах хмарних обчислень (ХО) внаслідок їхньої динамічної природи й необхідності підтримки гетерогенних застосунків [1]. Хоча технологія віртуальних машин (Virtual Machine, VMs) дозволяє паралельно виконувати кілька робочих навантажень (workloads) і використовувати спільну інфраструктуру, ця технологія не гарантує продуктивності (performance) застосунку. Тому провайдери хмарних центрів обробки даних (ЦОД) надають перевагу статичному, а не динамічному розміщенню VMs, тобто неефективному використанню ресурсів, не даючи гарантій продуктивності [2].

Крім того, робочі навантаження можуть мати різні вимоги якості послуги (Quality of Service, QoS) через виконання різнотипних застосунків (веб-застосунків або застосунків

високопродуктивних обчислень), що значно ускладнює забезпечення ресурсів [3].

Попередні дослідження зосереджувалися на єдиному типі угод про рівень обслуговування (Service Level Agreements, SLAs) чи зразків застосунків використання ресурсів (на веб-застосунках), призводячи до неефективного користування ресурсами. Можна запропонувати підходи до вирішення проблеми розподілу ресурсів у межах ЦОД з виконанням різних типів робочих навантажень застосунків, зокрема, неінтерактивних і транзакційних. Пропонуються контроль надходження і механізм планування, які не лише максимізують прибуток і використання ресурсів, але й гарантують те, що вимоги QoS користувачів задовольняються за SLAs [4]. Експериментально було доведено важливість урахування різних типів SLAs і штрафів за їх порушення, а також комбінацій

робочих навантажень для кращого забезпечення ресурсів і використання ЦОД. Запропонований механізм [4] забезпечує суттєве поліпшення порівняно зі статичною консолідацією серверів і зменшує порушення SLAs.

ХО призвели до зміни парадигми, коли підприємства, замість підтримки своєї власної інфраструктури, почали здійснювати аутсорсинг своїх обчислювальних потреб та ІТ-потреб за рахунок сторонніх провайдерів послуг [1–5]. Хмари – це великомасштабні аутсорсингові ЦОД, які розташовують тисячі серверів, які можуть одночасно запускати багато VMs. Тому вони розташовують широкий спектр застосунків і створюють у користувачів уявлення необмежених обчислювальних ресурсів на основі плати відповідно до обсягу використання [2].

Одним з наслідків Інтернету речей (Internet of Things, IoT) є щоденна поява великих об'ємів даних, які не завжди знаходять належне застосування через істотні відстані чи затримки до відповідної хмари аналізу цих даних. Тому розробляється нова модель туманних (fog) і периферійних (edge) обчислень, здатна своєчасно аналізувати (сприймати) чутливі дані (в режимі реального часу) на мережевій периферії, суміжній з місцем появи даних, і підтримувати дії (реакції): у такій моделі лише відібрані дані надсилаються до хмари аналізу та тривалого зберігання. Крім того, хмарні послуги, які надаються великими компаніями (скажімо, Google), можна також локалізувати, щоб мінімізувати час відгуку і збільшувати гнучкість обслуговування. Такої локалізації можна досягати шляхом розгортання маломасштабних ЦОД (cloudlets; хмарок), де це потрібно, близьких до клієнтів (пристроїв IoT) і з'єднаних з централізованою хмарою через мережі, формуючи периферійну хмару з множинним доступом (multi-access edge cloud, MEC). MEC включає принаймні три сторони – провайдерів Інтернет-інфраструктури (Internet as a service, IaaS), провайдерів застосунків (software as a service, SaaS), провайдерів мережі (network as a service, NaaS).

Ці сторони можуть мати різні цілі, ускладнюючи інтегрований менеджмент ресурсів MEC. Для такого менеджменту пропонуються різні методи в залежності від того, якого типу послуги треба розміщувати та як наявні ресурси слід розподіляти до застосунків клієнтів, зокрема мобільних застосунків. Тому замість централізованого менеджменту ресурсів припускається децентралізований менеджмент

кожною із сторін IaaS, SaaS, NaaS, який мінімізує енергоспоживання інфраструктури та витрати споживачів, забезпечуючи роботу застосунків. За емпіричними оцінками (слідів реального робочого навантаження від кластера Google), підхід [4] дозволяє знижувати енергоспоживання на майже 12 % (в основному, за рахунок IaaS), а витрати користувачів – на близько 18 % (в основному, за рахунок NaaS), забезпечуючи досить якісну роботу застосунків. Такі застосунки в режимі реального часу, як онлайн-ігри та відеоконференції, мають обов'язкові вимоги (on-demand requirements) забезпечення високоякісних результатів у межах домовленого терміну, наприклад, вимоги коротшого часу відгуку через зв'язок з найближчим сервером застосунків.

Використання хмарної платформи для розгортання застосунків у реальному часі пропонує ряд переваг, включаючи зазвичай менші експлуатаційні витрати (operational expenditures, OpEx) та розподіл ресурсів на вимогу відповідно до потреб застосунку. Однак застосунки в режимі реального часу можуть бути чутливими до якості мережі, наприклад, до затримок (зв'язку) між користувачами та сервісами. Тому вимоги таких застосунків можуть задовольнятися через поєднання новопосталої технології периферійних обчислень з MEC та туманних обчислень, уможливорюючи здійснення операцій на периферії мережі. Сенс введення в експлуатацію цієї технології полягає в розподілі послуг або запуску застосунків ближче до клієнтів і до місць попиту на результати операцій. Це досягається через розгортання маломасштабних або мікромасштабних ЦОД близько до клієнтів і підключення до регіональних хмарних ЦОД. Зазначимо, систем менеджменту для запуску та практичного застосування таких інфраструктур на 2021 р. майже не було, за винятком AWS Outposts (<https://aws.amazon.com/outposts/>), що вийшли на ринок у 2019 р., та переглянутого OpenStack [3].

Крім того, у рамках MEC або туманних обчислень є певні питання, що потребують вивчення, наприклад: а) де слід розгорнути ці малі ЦОД (хмарки); б) які сервіси слід встановлювати; в) де та які ресурси слід розподіляти застосункам користувачів; г) якою мобільністю користувачів (міграцією сервісів) слід управляти; д) як згадані рамки MEC слід оптимізувати (максимізувати чи мінімізувати різні цілі – монетарні витрати користувачів, енергоспоживання, продуктивність навантаження з погляду затримок, тривалості виконання тощо).

Хоча питання в), г), д), здебільшого, є схожими до традиційних хмар, але стратегії менеджменту з цих питань для туманної інфраструктури слід перепроектовувати.

Заслугує уваги вивчення основних проблем розподілу чи розміщення ресурсів, пов'язаних з периферійними (туманними) обчислювальними платформами: а) де слід встановлювати малі ЦОД (хмарки на периферіях мереж), щоб задовольняти попит користувачів і водночас знижувати глобальні витрати інфраструктури (суму OpEx та капітальних витрат (capital expenditures, CapEx)); б) як слід розгортати послуги глобального контролю після встановлення платформи; с) якого розміру чи масштабу має бути платформа. Вирішення цих проблем відкриватиме можливості для консультацій про загальну структуру розподілу чи розміщення для різних видів туманних або периферійних послуг, а також про їх структуру менеджменту чи міграції.

Хоча сучасна модель ХО, яку забезпечують кілька надвеликих і великих провайдерів (Google, Amazon AWS), є широко вживаною, інноваційні та новітні технології (застосунки IoT, периферійні обчислення, MEC) впливають на розвиток цієї моделі. Щоб впоратися з цими технологічними змінами, хмарові та мережеві спільноти працюють у напрямку великомасштабних розподілених інфраструктур малорозмірних ЦОД (хмарок), встановлених на периферії мережі ближче до користувачів та їхніх пристроїв, тобто туманних інфраструктур (що містять застосунки розподіленої конфігурації, які складаються з різних модулів). Туманна, периферійна та хмарна парадигма привертає більший інтерес, зважаючи також на те, що вона поліпшує рухливість (agility) і продуктивність послуг з точки зору часу відгуку. Наприклад, застосунки IoT можуть скористатися розгортанням периферійних вузлів для здійснення аналізу в режимі реального часу, зберігаючи основні ЦОД для поглибленої аналітики даних: поєднується згадана парадигма з MEC (яка дає множинний, а не лише мобільний доступ), що перебуває в зоні мережі радіодоступу (radio access network, RAN), де туман або периферія може пов'язувати споживчі пристрої, агрегуючи дані від сенсорів (до їх передачі на віддалену хмару).

Крім того, хмарка як міні-ЦОД, що має відповідати нормам великого хмарного провайдера, могла б підтримувати таку агрегацію, розташовуючись на більшій відстані

від клієнта, ніж сенсор або сенсори, без потреби перебування в зоні RAN.

Крім визначення того, де слід розгортати чи встановлювати периферійні хмари, рушії такої еволюції мають проектувати прийнятні системи менеджменту, які дозволятимуть: оператору (хмарі, мережі, периферії) агрегувати, спостерігати та виявляти такі масово розподілені ресурси; впроваджувати нові види послуг, які можуть розгортатися і керуватися оператором або користувачами. Однак проектування такої системи менеджменту є складним, бо туманні чи периферійні інфраструктури відрізняються від традиційних хмар щодо гетерогенності, динамічності, можливості великого розподілу ресурсів, економії від масштабу.

MECs пропонують обмежені ресурси на периферії мережі, уможливаючи запуск застосунку користувача в безпосередній близькості від нього через різноманітні туманні обчислення, які входять у відповідні хмарки з певної хмари. Ресурс периферії надається на хмарці чи сервері MEC. Крім того, застосунок у хмарці працює принаймні на одній логічній (hop) відстані зв'язку, на відміну від своєї початкової реалізації на мобільному (туманному) пристрої чи основному (core) Інтернеті (віддаленій хмарі). Тому затримка застосунку може потенційно залежати від послуг мережі. Скінчення кількість ресурсів хмарки теж може впливати на ефективність їх розподілу до підключених користувачів: хоча є підходи до поділу ресурсів кількох хмарок у певній географічній зоні, проблема ефективності стає складною, коли ці ресурси пропонуються різними провайдерами послуг з різними цілями й завданнями.

Аналогічно до провайдерів, користувачі мобільних пристроїв, які запускають застосунки, мають різні цілі та конкурують за ресурси: кожний користувач хоче оптимізувати свій власний вигравш або продуктивність застосунку.

Крім того, різні ресурси можуть пропонуватися при різних витратах, рівнях енергоспоживання та продуктивності. Розміщення ресурсів може істотно впливати на економіку провайдерів послуг (через IaaS) і споживачів (через SaaS): наприклад, нижча продуктивність застосунків збільшує витрати користувачів, а також енергоспоживання. Тому важливо забезпечувати відповідні ресурси для того, щоб задовольняти вимоги QoS застосунку та цілі провайдерів. Що стосується мобільності, то користувачі рухаються, а модулі застосунків явно мігрують серед хостів для підвищення

енергоефективності чи продуктивності, потенційно ускладнюючи менеджмент ресурсів. Серед відповідних методів менеджменту ресурсів для досягнення різних цілей суттєвими є: методи менеджменту інфраструктури та наявних ресурсів (IaaS, SaaS, NaaS) для підвищення рухливості послуг і продуктивності з точки зору часу відгуку, а також зниження енерговитрат; способи організації виконання навантажень користувачів, розташування та виділення необхідних ресурсів або сервісів; алгоритми ефективного розв'язання задач розміщення (мережевих та обчислювальних) ресурсів для підтримки новітніх мобільних застосунків MEC; підходи до реалізації та втілення моделей і алгоритмів у програмне забезпечення, які демонструють їх доцільність і практичність.

Проблема менеджменту ресурсів MEC складна тим, що пропонування якісних послуг кінцевим користувачам залежить від різноманітних гравців, цілі яких не є антагоністичними, але і не є однаковими. Такими гравцями є власники інфраструктури (IaaS), провайдерів застосунків (SaaS), мережових операторів (NaaS). Кожний з них контролює певну частину всієї системи. Невід'ємною складовою є те, що комунікаційні та обчислювальні спроможності потрібні для гарантування високої QoS з точки зору малого часу відгуку та високої пропускної здатності. Оскільки гравці можуть мати різні цільові функції, то поведінка одного гравця може потенційно впливати на поведінку іншого гравця. Традиційний підхід припускає, що вся інфраструктура управляється єдиним гравцем, який здебільшого забезпечує ресурси, або відділяє керівництво мережі з ресурсами застосунків від базової периферійної обчислювальної спроможності. У підході [4, 5] розміщення ресурсів у MEC вважається результатом конкуренції гравців, кожний з яких може оптимізувати свою цільову функцію. Цілі сторін певним чином узгоджуються: оскільки недостатнє забезпечення застосунків знижуватиме доходи всіх сторін (клієнти шукатимуть застосунки деінде, а провайдері шукатимуть клієнтів), то у сторін є стимули діяти у діапазоні від виграшу всіх гравців разом до програшу всіх гравців. Оскільки звичайні методи оптимізації не гарантують виграшу всіх гравців разом у подібних сценаріях розміщення ресурсів, то можна звернутися до методів теорії ігор [6].

Як правило, провайдері SaaS розміщують свої застосунки на віртуалізованих ресурсах, що

надаються провайдером IaaS. Крім того, провайдері SaaS мають дотримуватися кожної з вимог QoS застосунку, як сказано в SLA з користувачами, де визначається дохід SaaS на основі досягнутого рівня продуктивності. Однак продуктивність застосунків залежить не лише від обчислювальних ресурсів (що надаються провайдером IaaS), але також від пропускної здатності мережі (що надається провайдерами NaaS). Аналогічно провайдері послуг визначатимуть пріоритетність своїх робочих навантажень, виходячи з природи (своїх або сторонніх) застосунків. Оскільки у MEC провайдері NaaS можуть бути власниками IaaS (внутрішньої мережі) та сторонніми операторами (зовнішньої) мобільної мережі, то мережеві ресурси від різних провайдерів слід надавати за доступними цінами.

Провайдері SaaS приділяють основну увагу максимізації своїх доходів через мінімізацію SLAs, зменшуючи загальні витрати на обчислювальні ресурси (що надаються провайдерами IaaS), а також на мережеві ресурси (що надаються провайдерами NaaS і сторонніми провайдерами мережових послуг). У випадку множинних провайдерів IaaS або множинності MECs, провайдері SaaS конкуруватимуть і висуватимуть свої цінові пропозиції на використання інфраструктурних ресурсів. З іншого боку, провайдері IaaS і сторонні провайдері мережових послуг максимізуватимуть свої доходи шляхом більшого надання своїх віртуалізованих ресурсів. При цьому провайдері IaaS максимізуватимуть використання своїх ресурсів для мінімізації енергоспоживання.

У пропонованій теоретико-ігровій моделі є три гравці з різними цільовими функціями: IaaS мінімізує енергоспоживання через спрямування робочого навантаження на найменші ресурси; SaaS максимізує продуктивність обслуговування при обмеженнях SLAs, збільшуючи доходи; сторонні провайдері й оператори мережових послуг мінімізують мережевий трафік для гарантування QoS у сенсі часу відгуку. Позначимо N множину (перелік) $\{1, 2, \dots, N\}$ всіх таких гравців. Крім того, кожний гравець має набір (перелік) $K = \{1, 2, \dots, K\}$ різних наявних у себе видів ресурсів, зокрема, обчислювальних (IaaS), застосункових (SaaS), комунікаційних (NaaS). Провайдер $m \in N$ послуг пропонує ресурси $C^m = \{C_1^m, C_2^m, \dots, C_K^m\}$, де C_k^m – пропонований ним обсяг ресурсу типу k . Таким

чином, всі N провайдерів (хостів) загалом пропонують набір ресурсів (спроможностей)

$$C = \left\{ \sum_{i=1}^N C_1^i, \sum_{i=1}^N C_2^i, \dots, \sum_{i=1}^N C_K^i \right\} = \left\{ \sum_{i=1}^N C_k^i \right\}_{k=1, \dots, K}.$$

Кожне конкретне завдання (job) j для застосунок з цієї множини (модуль застосунок) потребує (коаліційного) набору ресурсів, який належить множині $M = \{M_1 \cup M_2 \cup \dots \cup M_K\}$ всіх застосунків при умові, що кожному модулю застосунок виділено неподільні ресурси кожного гравця $m \in N$: модулем застосунок вважається те, що запускається рівно один раз. Нехай кожний модуль j застосунок запускається у віртуальній машині чи контейнері. Крім того, завдання може містити велику кількість контейнерів, віртуальних машин, мережових ресурсів, які можуть мати спільне розташування; розташування лише однієї віртуальної машини чи контейнера на хості чи застосунок спрощує постановку задачі. Припустимо, що кожний модуль вимагає не більше однієї VM чи одного контейнера, а загальне число наданих VMs або контейнерів задовольнятиме ресурсні (resource) вимоги R_M застосунків.

Нехай система МЕС має один кластер (хмарний ЦОД), кілька периферійних розташувань (хмарок) і численні мобільні чи туманні пристрої. Ці ресурси з'єднані між собою через мережі так, що хмарки перебувають у безпосередній близькості до туманних пристроїв.

Модулі застосунок розподіляються і запускаються на цих ресурсах. Хмарний ЦОД складається з H гетерогенних кластерних хостів (hosts) $h \in \{1, 2, \dots, H\}$. Кожному хосту h відповідає вектор спроможності (capacity) $C^h = \{c_1^h, c_2^h, \dots, c_K^h\}$ за ресурсом типу $k \in K$: якщо ресурсами є процесор, пам'ять, сховище, мережа, то вектор $h = C^h = (4, 3, 2, 1)$ означатиме хост, який має 4 одиниці процесорів (CPU), 3 одиниці пам'яті (RAM, вимірюваної у гігабайтах), 2 одиниці дискового сховища (вимірюваного у терабайтах), мережеву карту на одиницю пропускної здатності (гігабіт в секунду). Нехай також є M периферійних розташувань (хмарок), а кожна периферійна хмарка $e \in M$ складається з множини S гетерогенних хостів; подібно до вектора C^h , кожному периферійному хосту $s \in S$ відповідає вектор $C^s \ll C^h$ спроможності.

Кожна VM може запускати конкретний модуль застосунок чи завдання. Позначимо J_u

застосунок або завдання, подане окремим користувачем $u \in \{1, 2, \dots, U\}$.

Кожний застосунок містить кілька модулів, які працюють одночасно. Крім того, різноманітні типи VMs або контейнерів наперед визначаються кожним хмарним (або хмарковим) провайдером; ресурсам кожного типу відповідає вектор спроможності $R = \{r_1, r_2, \dots, r_U\}$.

Зазначимо, що кожна VM чи контейнер може здійснювати єдине завдання (модуль застосунок) у даний момент часу як на віддаленому ЦОД, так і на хмарках.

Ресурси різного типу хоста $h \in H$ чи хмарки $s \in S$ будуть зайнятими лише тоді, коли на цьому хості чи хмарці створено конкретну VM чи контейнер. Таким чином, міграція VMs уможливорює мобільність модулів застосунків.

За вищевизначених умов, запити (requests) ресурсів для конкретного завдання j (або користувача) можна визначити як матрицю $u \times v$, де рядок задає тип VM чи контейнера, а стовпець – різні ресурси, пов'язані з типом VM чи контейнера:

$$R^j = \begin{bmatrix} r_1^j \\ r_2^j \\ \vdots \\ r_u^j \end{bmatrix} = \begin{bmatrix} r_{11}^j & r_{12}^j & \dots & r_{1v}^j \\ r_{21}^j & r_{22}^j & \dots & r_{2v}^j \\ \vdots & \vdots & \ddots & \vdots \\ r_{u1}^j & r_{u2}^j & \dots & r_{uv}^j \end{bmatrix}.$$

Тоді всі запити (від усіх провайдерів послуг) визначаються розширеною матрицею $U \times V$ вимог ресурсів

$$R = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_U \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1v} \\ r_{21} & r_{22} & \dots & r_{2v} \\ \vdots & \vdots & \ddots & \vdots \\ r_{U1} & r_{U2} & \dots & r_{Uv} \end{bmatrix}.$$

Припускаємо, що конкретне завдання можна призначити щонайбільше одному хосту. Нехай для конкретного хоста $h \in H$ можливий стан розміщення ресурсів задається матрицею розміщення (allocation) $u \times v$

$$A^h = \begin{bmatrix} a_1^h \\ a_2^h \\ \vdots \\ a_u^h \end{bmatrix} = \begin{bmatrix} a_{11}^h & a_{12}^h & \dots & a_{1v}^h \\ a_{21}^h & a_{22}^h & \dots & a_{2v}^h \\ \vdots & \vdots & \ddots & \vdots \\ a_{u1}^h & a_{u2}^h & \dots & a_{uv}^h \end{bmatrix},$$

де a_{dk}^h – обсяг ресурсів k на конкретному хості h , виділених контейнеру чи VM d . Тоді, аналогічно до матриці всіх запитів, можна ввести матрицю всіх розміщень (від усіх провайдерів)

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_U \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1V} \\ a_{21} & a_{22} & \cdots & a_{2V} \\ \vdots & \vdots & \ddots & \vdots \\ a_{U1} & a_{U2} & \cdots & a_{UV} \end{bmatrix}.$$

Для кожного хоста рішення (decision) про розміщення ресурсів є набором усіх можливостей (possibilities), оснований на матриці R :

$$D = \{ A^1, A^2, \dots, A^h, \dots, A^P \}.$$

Мета проблеми розміщення ресурсів, при даних матриці R та наборі C , – обчислити прийнятне відображення від ресурсів до завдань користувача: різні гравці (провайдери ресурсів, застосунків, мережі) спільно приходять до єдиного рішення, що описує розміщення VMs, які є найкращими узгодженими розміщеннями для всієї системи MEC у сенсі оптимізації енергоспоживання та продуктивності (часу виконання завдань).

Якщо вважати цю проблему задачею однокритеріальної оптимізації, то кожний провайдер послуг окремо бажає максимізувати свою корисність шляхом такого розміщення своїх наявних ресурсів, що: IaaS мінімізує енергоспоживання та продуктивність робочого навантаження (сподіваний час виконання); SaaS мінімізує сподіваний час виконання застосунків; NaaS мінімізує мережевий трафік [7]. Ця задача полягає у виборі такої матриці A розміщення, яка максимізує суму корисностей усіх провайдерів $\sum_{j=1}^N u_j^n(A_{i \in M}^n)$.

Крім того, для кожної VM чи застосунку i її матриця A^i розміщення на кожному провайдері послуг $N \in \{IaaS, SaaS, NaaS\}$ задається функцією $X_{iN} \in \{0, 1\}$ відображення ресурсів на

провайдери з обмеженням $\sum_{i=1}^N X_{iN} \leq 1$, яке

означає, що кожний застосунок розміщується рівно один раз на одному з провайдерів. Після розміщення застосунку i кожний користувач платить за нього ціну (price) p , яка складається з p_{in}^{IaaS} , p_{in}^{SaaS} , p_{in}^{NaaS} – вартостей (корисностей) послуг IaaS, SaaS, NaaS для застосунку, користувача, каналу i відповідно на хості (провайдері) $n \in N$. Тому корисність усієї системи MEC від конкретного користувача із застосунком i задається

$$U_i = X_{ij}(p_{ij}^{IaaS} + p_{ij}^{SaaS} + p_{ij}^{NaaS}),$$

де X_{ij} – функція відображення від застосунку i на провайдера j , а загальна корисність системи MEC (усіх провайдерів) від M застосунків задається $U = \sum_{i=1}^M U_i$.

Отже, мета всієї системи MEC полягає у максимізації корисності U при обмеженнях: розміщені ресурсні спроможності не можуть перевищувати загальних спроможностей; для кожного користувача розміщуються ресурси рівно один раз в його безпосередній близькості. Вищезазначену проблему можна сформулювати як задачу багатокритеріальної оптимізації для MEC, що складається з ЦОД, кількох периферійних розташувань (хостів) і завдань користувачів, які працюють на різноманітних VMs: знайти відображення VMs до хостів таке, що мінімізує кумулятивне енергоспоживання MEC та максимізує продуктивність VM (мінімізує час виконання на VM). Аналогічно для SaaS має максимізуватися продуктивність різних застосунків. Крім того, для NaaS має максимізуватися доступна пропускна здатність (bandwidth) B . Вважатимемо продуктивністю час виконання на VM: чим довший час виконання на VM, тим гірше, і навпаки. Таким чином, можна мінімізувати згортку кількох згаданих критеріїв

$$\sum_{i=1}^N \sum_{j=1}^M e_{ij} \sum_{i=1}^N t_{ij} x_{ij} + \sum_{k=1}^M t_k - \sum B - U - \sum_{j=1}^N u_j^n(A_{i \in M}^n),$$

де x_{ij} – функція відображення VMs на хости ($x_{ij} = 1$, якщо конкретна VM i відображається на хост j), e_{ij} – спожита енергія на хості j для VM i , t_{ij} – час виконання VM i на хості j , t_k – час виконання застосунку k . Ця мінімізація має брати до уваги обмеження, що різні ресурси k , розміщені для VMs, не можуть перевищувати спроможності хостів (провайдерів) n :

$$\sum_{i=1}^N x_{ik}^n \leq C_k^n.$$

Крім того, має братися до уваги обмеження, що для VMs пропонуються лише потрібні ресурси:

$$\sum_{l \in N} x_{ik}^l \leq r_{ik}^n.$$

Зазначимо, що час виконання застосунку обернено пропорційний продуктивності мережі, яка, в свою чергу, не пропорційна енергоспоживанню.

При оптимізації за одним критерієм провайдер SaaS максимізуватиме свій дохід шляхом

мінімізації виграшів користувачів. Проте нижчі витрати на користувачів можуть збільшувати число клієнтів і попит на послуги, які небажано впливатимуть на робочі навантаження користувачів: такі нижчі витрати опосередковано передбачають збільшення порушень SLA. Ці порушення залежать від ресурсів, придбаних у відповідних провайдерів IaaS та NaaS (які мають близькі інтереси). Якщо провайдери пропонують послуги, насамперед, клієнтам з найвищою готовністю платити, то має братися до уваги

Список використаних джерел

1. Горбачук В. Економіка Internet-застосунків і цифрового контенту / В. Горбачук, С. Гавриленко, Г. Голоцуков, Д. Ніколенко // *The Role of Technology in the Socio-Economic Development of the Post-Quarantine World*. M. Gavron-Lapuszek, A. Karpenko (eds.) Katowice: Katowice School of Technology, 2020. P. 81–88.
2. Gorbachuk V. The impact of cloud services pricing on provider profit, consumer surplus, and social welfare / V. Gorbachuk, S. Gavrylenko // *CEUR Workshop Proceedings*. – 2020. – 2866. – P. 237–245.
3. Bardadym T. On biomedical computations in cluster and cloud environment / T. Bardadym, V. Gorbachuk, N. Novoselova, S. Osypenko, V. Skobtsov, I. Tom // *Cybernetics and Computer Technologies*. – 2021. – № 2. – P. 76–84.
4. Zakarya M. epcAware: a game-based, energy, performance and cost efficient resource management technique for multi-access edge clouds / M. Zakarya, L. Gillam, H. Ali, I.U. Rahman, K. Salah, R. Khan, O. Rana, R. Buyya // *IEEE Transactions on Services Computing*. – 2020. – doi: 10.1109/TSC.2020.3005347.
5. Li K. A game theoretic approach to computation offloading strategy optimization for non-cooperative users in mobile edge computing / K. Li // *IEEE Transactions on Sustainable Computing*. – 2018. – doi: 10.1109/TSUSC.2018.2868655.
6. Моклячук М.П. Лекції з теорії вибору та прийняття рішень / М.П. Моклячук, Р.Є. Ямненко. Київ: Київський університет, 2007. 256 с.
7. Gorbachuk V.M. Wholesale price contracts under random demand / V. M. Gorbachuk // *Modern Stochastics: Theory and Applications*

додаткове обмеження: така пропозиція подібна перевазі (від провайдера IaaS) користувачів Gmail на кластері Google порівняно з кластером Azure MicroSoft .

Таким чином, проблема розміщення ресурсів MEC є досить складною і такою, що потребує новітніх підходів до децентралізованого прийняття рішень у ситуаціях перебору великої кількості варіантів – аналізу великих даних.

V. Kyiv: Taras Shevchenko National University of Kyiv, 2021. P. 67–68.

References

1. GORBACHUK, V. et al. (2020) Economics of Internet-applications and digital content. In *The Role of Technology in the Socio-Economic Development of the Post-Quarantine World*. M. Gavron-Lapuszek, A. Karpenko (eds.) Katowice: Katowice School of Technology, pp. 81–88.
2. GORBACHUK, V. and GAVRYLENKO, S. (2020) The impact of cloud services pricing on provider profit, consumer surplus, and social welfare. In *CEUR Workshop Proceedings*, 2866, pp. 237–245.
3. BARDADYM, T. et al. (2021) On biomedical computations in cluster and cloud environment. *Cybernetics and Computer Technologies*, № 2, pp. 76–84.
4. ZAKARYA, M. et al. (2020) epcAware: a game-based, energy, performance and cost efficient resource management technique for multi-access edge clouds. *IEEE Transactions on Services Computing*. doi: 10.1109/TSC.2020.3005347.
5. LI, K. (2018) A game theoretic approach to computation offloading strategy optimization for non-cooperative users in mobile edge computing. *IEEE Transactions on Sustainable Computing*. doi: 10.1109/TSUSC.2018.2868655.
6. MOKLYACHUK, M.P., and YAMNENKO, R.E. (2007) Lectures on the choice theory and decision making. Kyiv: Kyivskyj universytet. 256 p.
7. GORBACHUK, V.M. (2021) Wholesale price contracts under random demand. In *Modern Stochastics: Theory and Applications* V. Kyiv: Taras Shevchenko National University of Kyiv, pp. 67–68.

Надійшла до редколегії 30.08.21