

ISSN 1998-7927(print) ISSN 2664-6498 (online)

DOI: <https://doi.org/10.33216/1998-7927-2026-304-6-7-18>

УДК 004.383.8:004.421

МОДЕЛЬ І МЕТОД КОНСОЛІДАЦІЇ ВІРТУАЛЬНИХ МАШИН ХМАРНОГО ЦОД ІЗ ЗАСТОСУВАННЯМ КВАНТОВИХ АЛГОРИТМІВ

Жаріков Е.В., Куземський М.В., Полупан Ю.В., Зарічковий О.А., Ліщук К.І.

MODEL AND METHOD FOR VIRTUAL MACHINE CONSOLIDATION IN A CLOUD DATA CENTER USING QUANTUM ALGORITHMS

Zharikov E.V., Kuzemskiy M.V., Polupan Yu.V., Zarichkovyi O.A., Lishchuk K.I.

Застосування квантових та квантово-класичних алгоритмів для розв'язання широкого кола задач оптимізації набуває все більшого поширення. У цій статті запропоновано застосування квантово-класичних алгоритмів для розв'язання задачі розподілу віртуальних машин по фізичних серверах хмарного центру оброблення даних, яка відноситься до класу задач про пакування в ємності. Хмарні центри оброблення даних стають складнішими та масштабнішими, що зумовлює розроблення ефективніших алгоритмів консолідації віртуальних машин для максимізації використання ресурсів та мінімізації споживання електроенергії. Методи планування, які використовувались на ранніх етапах розвитку хмарних обчислень, простіші в реалізації, але погано працюють у динамічних сценаріях, оскільки вони не враховують складні обмеження та не підтримують багатоцільову оптимізацію. У цих умовах квантові обчислення надають можливість подолання комбінаторних проблем завдяки своїй здатності одночасно оброблювати багато станів.

Задача консолідації віртуальних машин сформульована у формі QUBO/Ising. Виконано порівняльний аналіз квантових фреймворків (SimulatedAnnealingSampler і QAOA), які є прямими класичними імітаціями двох основних архітектур квантових обчислень (annealing та gate-model відповідно) та класичних базових алгоритмів MBFD та CP-SAT у термінах масштабованості (розмір задачі), якості знайденого рішення (кількість активних фізичних серверів відносно точного оптимуму та коефіцієнт апроксимації відносно точного оптимуму), а також часу виконання алгоритму. Запропоновані модель ЦОД та гібридний квантово-класичний алгоритм розміщення віртуальних машин на фізичних серверах реалізовані мовою Python.

Результати статті отримані без використання квантового апаратного забезпечення, але розроблене програмне забезпечення моделювання може безпосередньо використовувати квантові процесори, що доступні за допомогою сервісу Amazon Braket.

Ключові слова: квантова симуляція, квантовий хмарний сервіс, квантовий розв'язувач, квантовий алгоритм наближеної оптимізації, CUBO, XQM, квантовий відпал.

1. Вступ. Останнім часом на ринку домінують мультихмарна та гібридна моделі розгортання хмарних послуг [1], а модель надання хмарних послуг «Інфраструктура як сервіс» (IaaS) залишається найпоширенішою в Україні [2]. Хмарними центрами обробки даних (ЦОД) зараз надається широкий спектр сучасних інформаційних послуг та застосунків. Сучасні ЦОД є складними системами надання високопродуктивних та відмовостійких ІТ-послуг користувачам і орендарям за допомогою концепції «комунальних обчислень» (англ. utility computing) [3].

Для управління продуктивністю, якістю послуг та споживанням електроенергії в хмарних ЦОД провайдери хмарних послуг вирішують три основних завдання для ефективного розподілу ресурсів:

- забезпечення виконання угоди про рівень обслуговування (англ. Service Level Agreement, SLA) між постачальником хмарних послуг та користувачем;
- зниження споживання електроенергії ЦОД;
- скорочення операційних витрат на управління послугами ЦОД.

Ефективне використання ресурсів хмарного ЦОД переважно досягається у спосіб розв'язання задачі консолідації та розміщення віртуальних машин (англ. Virtual Machine Consolidation / Placement), яка визначає, як розподіляються ресурси фізичних серверів (ФС) для запуску та роботи віртуальних машин (ВМ) з метою гарантування вимог до якості обслуговування (QoS) та зменшення загальної кількості використовуваних ФС. Провайдери

використовують ВМ для широкого кола задач: надання клієнтам згідно моделі IaaS, запуску контейнерів, а також реалізації безсерверних обчислень. ВМ з різними потребами в ресурсах зазвичай розміщуються на одному ФС з використанням техніки надлишкової підписки без урахування довгострокового використання, що може призвести до конкуренції за ресурси між локальними ВМ, низької якості обслуговування і, як наслідок, порушення SLA. Більше того, кількість ВМ та інтенсивність навантажень, що обслуговуються ВМ, постійно змінюються, що вимагає адаптивних методів для динамічного управління віртуальними машинами, починаючи із запиту на розміщення нової ВМ.

Для консолідації віртуальних машин широко застосовують відомі моделі і методи розв'язання задачі пакування в контейнери (англ. bin-packing problem, BPP) [4, 5]. Точні та формальні методи оптимізації не використовуються для консолідації великої кількості ВМ, як правило використовуються евристичні та наближені методи оптимізації з використанням променевого пошуку [6], імітації відпаду [7], навчання з підкріпленням [8], генетичних алгоритмів [9], та інші [10]. Але зі зростанням кількості ФС та ВМ точність запропонованих методів знижується, виникають складнощі з масштабованістю, а також зростає час пошуку рішення. Також ці методи пропонували кращу масштабованість, але часто вимагали налаштування значної кількості параметрів і більше ресурсів для реалізації методів при роботі для великих кластерів ФС. Методи машинного навчання, особливо навчання з підкріпленням та нейронні мережі, показали себе перспективними у прогнозуванні попиту на ресурси та оптимізації розподілу ресурсів між ФС. Однак ці методи вимагають спеціальних процедур навчання та часто не мають прозорості в процесах прийняття рішень. У цих умовах перспективним є застосування квантових та квантово-класичних алгоритмів розв'язання задачі консолідації ВМ в ЦОД [11].

Представлення задачі консолідації та розміщення віртуальних машин як задачі квадратичної бінарної оптимізації без обмежень (англ. Quadratic Unconstrained Binary Optimization, QUBO) та квадратичної моделі з обмеженнями (англ. Constrained Quadratic Model, CQM) є одним із сучасних способів формулювання прикладної NP-важкої задачі пакування, BPP, для виконання на базі квантових процесорів з використанням квантового алгоритму наближеної оптимізації (англ. Quantum Approximate Optimization Algorithm, QAOA [12], алгоритму варіаційного квантового розв'язувача власних значень (англ. Variational Quantum Eigensolver, VQE) [13] та апаратних засобів реалізації квантового відпаду D-Wave (англ. Annealing Quantum Computing) [14].

У статті [15] автори запропонували модель віртуального ЦОД у вигляді цілочисельного

лінійного програмування (англ. Integer Linear Programming, ILP) з метою консолідації віртуальних машин для одночасної мінімізації енергоспоживання мережевого обладнання та фізичних серверів. Розроблена модель у подальшому перетворена в еквівалентну двійкову модель, а потім крок за кроком у модель QUBO. Запропонована модель QUBO може бути використана для розв'язання задачі консолідації ВМ як на базі вентильних схем, так і на апаратних засобах реалізації квантового відпаду (англ. Quantum annealing processors).

Фреймворк квантового планування розміщення завдань, який використовує формулювання QUBO та квантові розв'язувачі, представлений у роботі [16]. Фреймворк забезпечує планування завдань у розподілених середовищах і поєднує квантову оптимізацію QAOA з традиційними фреймворками розподілених обчислень. Квантові алгоритми оптимізують призначення завдань серверам на основі задачі планування, сформульованій в моделі QUBO. Результати досліджень демонструють, значне покращення тривалості виконання завдань та підвищує ефективність використання ресурсів за рахунок використання квантового планування. Для імітації реальних навантажень використано трасування навантаження кластерів Google, а завдання описуються вимогами до процесора, пам'яті та часу виконання.

У роботі [17] пропонується нова двонаправлена система оптимізації та прогнозування для інтелектуального планування завдань у федеративних хмарних середовищах, яка називається квантово-натхненим адаптивним метаевристичним машинним навчанням (QI-AMHML), на основі квантового алгоритму оптимізації, який поєднує квантові амплітуди ймовірностей, динаміку пошуку та метод класичної оптимізації, що дозволяє приймати проактивні рішення щодо розміщення навантаження при змінних сценаріях та несправностях. Метод поєднує моделі стійкості, енергетичні моделі та обмеження балансування навантаження в єдину багатоцільову оптимізацію. Для оцінки роботи методу автори створили гібридний набір даних, об'єднавши трасування навантаження кластерів Google, Microsoft Azure та Alibaba Cloud із синтетичними подіями сплесків (навантаження) та несправностей. Застосування запропонованого методу дозволяє зменшити споживання енергії до 38,2%, зменшити середню затримку обслуговування на 33,5% та зберегти високі показники стійкості, що демонструє можливість квантово-натхненого пошуку, підкріпленого федеративними прогнозними моделями, забезпечити відмовостійкість та стійку продуктивність в реальних сучасних багатокластерних гібридних хмарах.

Квантово-асистований гібридний метод для підвищення ефективності планування завдань на платформах хмарних обчислень представлений у [18]. Він використовує QAOA для отримання майже оптимальних відображень завдань у ресурси шляхом

розв'язання комбінаторних задач планування, а VQE – для уточнення моделі споживання енергії за допомогою методів варіаційної оптимізації схем. Метод дозволяє зменшити споживання енергії на 23,6%, тривалість виконання завдань на 17,4% та збільшити використання ресурсів на 17,7%. Результати підтверджують, що квантово-класична гібридна оптимізація може бути застосована у високопродуктивних інфраструктурах хмарних обчислень.

Для практичного застосування квантових і гібридних квантово-класичних алгоритмів [19] на ринку існує декілька провайдерів послуг в галузі квантових обчислень, таких як Amazon Braket [20], Rigetti Quantum Cloud Services [21], IBM Quantum [22], Google Quantum AI [23], Azure Quantum [24], D-Wave [14], Strangeworks [25] та інші. Але основним викликом для практичного застосування вентильних схем залишається наявність **NISQ-обмежень** (англ. Noisy Intermediate-Scale Quantum), таких як шум, декогеренція, помилки зчитування, обмежена кількість кубітів та помилки квантових вентилів. Для зменшення впливу NISQ-обмежень в алгоритмах VQE та QAOA застосовують методи пом'якшення помилок (англ. Error Mitigation): метод екстраполяції до нульового шуму (англ. ZNE), віртуальне очищення результатів обчислень без використання повноцінної квантової корекції помилок, оптимізація архітектури анзапу для зменшення кількості вентилів.

У статті автори розробили адаптовану модель ЦОД для задачі розміщення віртуальних машин на фізичних серверах з метою зниження електроспоживання за рахунок зменшення кількості активних ФС, та провели порівняльний аналіз чотирьох класів розв'язувачів цієї задачі в єдиній QUBO/CQM-постановці: класичної евристики (MBFD), точного класичного розв'язувача обмежень (CP-SAT), локальної імітації квантового відпалу (SimulatedAnnealingSampler) та гейт-модельного варіаційного квантового алгоритму (QAOA на локальному симуляторі). Такий дизайн дозволяє отримати змістовне порівняння класичних і квантово-натхнених підходів на етапі локальної симуляції, з метою підготовки до використання реального квантового обладнання.

Отже, для розв'язання задачі розміщення віртуальних машин ЦОД з метою зменшення електроспоживання у статті пропонується спрощена модель ЦОД та порівняльний аналіз алгоритмів розміщення віртуальних машин на основі класичних і квантово-натхнених розв'язувачів, що дозволяє виявити межі застосовності кожного підходу та дослідити залежність показників якості управління ресурсами ЦОД від параметрів квантової моделі.

2. Постановка задачі розміщення віртуальних машин на фізичних серверах. При моделюванні хмарного ЦОД зазвичай враховують такі ресурси в якості входних параметрів: завантаженість центрального процесора (ЦП), доступну оперативну пам'ять, пропускну здатність

мережевого інтерфейсу фізичного сервера та продуктивність роботи системи збереження даних (СЗД).

Чим більше ресурсів враховується в моделі ЦОД під час оцінки роботи методів управління ресурсами, тим ефективніше працюють методи керування розміщенням та консолідацією віртуальних машин ЦОД. Багато фреймворків і програмних засобів моделювання враховують лише робоче навантаження центрального процесора та обсяг використаної оперативної пам'яті як входні дані для змодельованої ВМ [26, 27]. У літературі представлені також інструменти моделювання ЦОД, які використовують у якості входних даних для кожної змодельованої ВМ навантаження на чотири ресурси: ЦП, оперативну пам'ять, мережу та СЗД [28].

Процес розміщення ВМ розглянуто як підзадачу загальної проблеми консолідації ВМ, яка зазвичай розглядається як задача пакування в контейнери, де N предметів (віртуальних машин) з різними розмірами в багатовимірному просторі ресурсів потрібно розмістити в M контейнерах (фізичних серверах) для одночасного досягнення кількох цілей. У повноцінних схемах управління ресурсами хмарних ЦОД розглядають дві цілі: мінімізацію споживання електроенергії та мінімізацію порушень SLA. Задача пакування ускладнюється зміною властивостей предметів, запитами на розгортання нових предметів та перепризначенням існуючих предметів в інші контейнери. Оскільки клієнти запитують послуги та подають заявки для обробки однієї або кількох ВМ, навантаження в хмарному ЦОД може значно змінюватися з часом. Це вимагає розв'язання оптимізаційної задачі для розміщення нових ВМ та періодичного перерозподілу активних ВМ в асинхронному режимі або після визначеної кількості кроків. Іншими обмеженнями при консолідації віртуальних машин є обмеження на кількість одночасних міграцій ВМ на один ФС, обмеження використання ресурсів ФС, а також обмеження кількості (або часу) порушень SLA [29].

На відміну від загальної задачі консолідації ВМ, задача розміщення ВМ є її підзадачею і не включає перерозподіл ВМ по ФС при перевищенні попиту на ресурси ФС з боку локальних ВМ. Задачу розміщення ВМ, як правило розв'язують у трьох випадках:

1. При виборі фізичних серверів для групи ВМ, які заплановані для створення і запуску;
2. При виборі фізичного серверу для однієї/кожної ВМ, яка запланована для створення і запуску;
3. При сервісному обслуговуванні набору ФС потрібно розподілити вже працюючі ВМ по інших ФС кластеру.

Отже, необхідно розробити модель ЦОД та квантово-класичний метод керування його ресурсами для розв'язання задачі розміщення групи віртуальних машин на фізичних серверах з метою зниження електроспоживання ЦОД за рахунок зменшення кількості активних ФС. Для застосування квантових

методів та алгоритмів необхідно адаптувати модель ЦОД з урахуванням розмірів ВМ та ФС, але без врахування міграцій ВМ і у подальших публікаціях розвинути квантово-класичний метод керування з урахуванням динаміки процесів та навантажень в середині ВМ.

3. Методика дослідження, моделі та засоби моделювання. Для розроблення програмного забезпечення симуляції і моделювання розміщення (консолідації) віртуальних машин на фізичних серверах хмарного ЦОД необхідно розробити модель хмарного ЦОД, розробити або реалізувати існуючі алгоритми керування процесом консолідації та вибрати засоби або платформи моделювання.

3.1. Адаптована модель хмарного ЦОД. Модель хмарного ЦОД складається з множини фізичних серверів $P = \{1, \dots, M\}$, де M – кількість ФС, та множини віртуальних машин $V = \{1, \dots, N\}$, де N – кількість ВМ, що підлягають розміщенню. Модель представлена у вигляді квадратичної моделі з обмеженнями, CQM, та адаптована з [6] та [30] для статичного розміщення ВМ без урахування міграцій ВМ між ФС.

Параметри моделі:

- C_i^k – ємність ФС i за ресурсом k ($k \in \{CPU, RAM\}$);
- d_j^k – потреба ВМ j за ресурсом k ;
- $T_k = 0.90$ – поріг допустимого завантаження ФС за ресурсом k (максимальна частка ємності, яку дозволено використовувати);
- $D_k = 0.85$ – бажаний рівень завантаження ФС за ресурсом k (цільовий показник для оцінки якості консолідації);
- профілі ФС R630, R740, R830, R940 (конфігурації Dell PowerEdge за [6]) визначають конкретні значення C_i^{CPU} , C_i^{RAM} для кожного типу сервера в експериментальному наборі.
- $x_{j,i} \in \{0, 1\}$ – змінна розміщення ВМ j на ФС i в результаті оптимізації;
- $y_i \in \{0, 1\}$ – змінна активності ФС i (містить хоча б одну ВМ) в результаті оптимізації.

Цільова функція: мінімізація кількості активних ФС для зменшення енергоспоживання ЦОД, $\min \sum_{i=1}^M y_i$.

Обмеження:

- кожна ВМ розміщується рівно на одному ФС:

$$\sum_{i=1}^M x_{j,i} = 1, \forall j \in \{1, \dots, N\} \quad (1)$$

- сумарне навантаження ВМ j на ФС i не перевищує допустимий поріг T_k від ємності ФС, для кожного ресурсу k :

$$\sum_{j=1}^N d_j^k \cdot x_{j,i} \leq T_k \cdot C_i^k \cdot y_i, \forall i \in \{1, \dots, M\}, \forall k; \quad (2)$$

- зв'язок між x і y – ВМ може бути розміщена лише на активному ФС (забезпечується автоматично обмеженням (2), оскільки права частина дорівнює нулю при $y_i = 0$).

Показник $D_k = 0.85$ використовується не як жорстке обмеження, а як значення для оцінки якості розміщення ВМ постфактум: розміщення вважається «якісним», якщо фактичне середнє завантаження активних ФС наближається до D_k , не перевищуючи поріг T_k (компроміс між щільністю консолідації та запасом на пікові навантаження).

Представлена модель та цільова функція є спільною відправною точкою для всіх методів, що порівнюються в розділі 4 статті: MBFD і CP-SAT розв'язують її безпосередньо в обмеженій формі, тоді як для квантово-натхнених методів (SimulatedAnnealingSampler, QAOA) вона попередньо перетворюється у безумовну форму BQM/QUBO та Ising (штрафні коефіцієнти λ_1 , λ_2 представлені в розділах 3.2 і 4.4 статті) шляхом додавання обмежень (1)-(2) як штрафних доданків до цільової функції.

3.2. Розроблення алгоритму QAOA для розміщення ВМ. Квантовий алгоритм наближеної оптимізації – це варіаційний гібридний квантово-класичний алгоритм комбінаторної оптимізації який широко застосовується для розв'язання NP-складних задач в різних галузях [12]. Задача формулюється як мінімізація гамільтоніана вартості H_C (Ising-форма отримана зі штрафного перетворення CQM $\rightarrow$ BQM $\rightarrow$ Ising). Квантова схема з p шарів чергує:

cost-унітарний оператор $\exp(-iyH_C)$, що кодує структуру задачі (обмеження ємності серверів, правило «одна ВМ – один сервер»),

mixer-унітарний оператор $\exp(-i\beta H_M)$, $H_M = \sum X_i$, що забезпечує дослідження простору станів.

$2p$ класичних параметрів ($\gamma_1 \dots \gamma_p, \beta_1 \dots \beta_p$) підбираються класичним оптимізатором COBYLA [31] за принципом мінімізації очікуваної енергії, виміряної на квантовому симуляторі Qiskit Aer. Після збіжності виконується фінальний прогін з оптимальними кутами, і серед виміряних бітових рядків обирається той, який представляє мінімальну енергію.

Для оцінки якості рішень QAOA використовується відома метрика – коефіцієнт апроксимації r :

$$r = \frac{C_{opt}}{C_{QAOA}},$$

де C_{opt} – точне значення цільової функції (мінімальна кількість активних ФС, отримана через розв'язувач ExactCQMSolver на малих задачах), C_{QAOA} – значення, знайдене розв'язувачем QAOA. $r = 1.0$ означає, що розв'язувач QAOA знайшов оптимальне рішення; $r < 1$ означає, що знайдено субоптимальне рішення. Ця метрика застосовується лише до задач достатньо малого розміру ($\leq 18 - 20$ бінарних

змінних), де можна обчислити точне оптимальне значення. Псевдокод алгоритму розміщення віртуальних машин представлений в Лістингу 1.

Algorithm: QAOA для задачі розміщення віртуальних машин ($CQM \rightarrow QAOA$)

Input:

V : множина віртуальних машин (ВМ) з вимогами до ресурсів (CPU, RAM, Bandwidth)

P : множина фізичних серверів (ФС) з їхніми ємностями

T : поріг завантаження фізичних серверів (Threshold)

p : кількість шарів (depth) у схемі QAOA

$penalty$: ваговий коефіцієнт штрафів за порушення обмежень

MAX_{ITER} : максимальна кількість ітерацій оптимізатора

$SHOTS$: кількість запусків квантової схеми

Output:

$assignment$: словник призначень $\{VM: PM\}$

num_{active} : кількість задіяних фізичних серверів

$feasible$: прапорець/статус допустимості отриманого розв'язку

r : коефіцієнт наближення (approximation ratio).

Лістинг 1. Псевдокод алгоритму розміщення віртуальних машин

1: // --- Етап 1: реалізація математичної моделі ---

2: $CQM \leftarrow \text{BuildCQM}(V, P, T)$ // Мінімізація активних ФС за умов (1), (2), та
// обмеження зв'язків/трафіку між ФС

3: $BQM \leftarrow \text{CqmToBqm}(CQM, penalty)$ // Перетворення обмежень у штрафні функції

4: $(h, J, offset) \leftarrow BQM.\text{ToIsing}()$ // Перехід до моделі Ізінга: $s_i \in \{-1, +1\}$

5: $n \leftarrow \text{length}(h)$ // Кількість необхідних кубітів

6: if $n > MAX_{QUBITS}$ then

7: raise Error("Розмірність задачі перевищує ліміт кубітів для симуляції")

8: end if

9: // --- Етап 2: Варіаційна оптимізація QAOA ---

10: $(gamma, beta) \leftarrow \text{RandomInit}(size = 2 \cdot p)$

11: Function CostFunction($gamma, beta$):

12: $qc \leftarrow \text{BuildQAOACircuit}(h, J, n, gamma, beta)$

13: $counts \leftarrow \text{AerSimulator.run}(qc, shots = SHOTS).get_counts()$

14: $E \leftarrow \text{Compute ExpectationEnergy}(counts, h, J, offset)$

15: return E

16: $(gamma_{opt}, beta_{opt}) \leftarrow \text{COBYLA_Optimizer.minimize}(\text{CostFunction}(gamma, beta), maxiter = MAX_{ITER})$

17: // --- Етап 3: Вибірка найкращого стану та декодування ---

18: $qc_{opt} \leftarrow \text{BuildQAOACircuit}(h, J, n, gamma_{opt}, beta_{opt})$

19: $final_counts \leftarrow \text{AerSimulator.run}(qc_{opt}, shots = SHOTS).get_counts()$

20: $x_{star} \leftarrow \text{GetStateWithMinimumEnergy}(final_counts, h, J, offset)$

21: $cqm_sample \leftarrow \text{InvertToBinary}(x_{star})$ // Перетворення $\{-1, +1\} \rightarrow \{0, 1\}$ у змінні CQM

22: if $CQM.\text{CheckFeasible}(cqm_sample)$ then

23: $feasible \leftarrow \text{True}$

24: $assignment \leftarrow \text{ExtractAssignments}(cqm_sample, V, P)$ // $x_{v,p} = 1$

25: $num_{active} \leftarrow \text{CountActiveServers}(cqm_sample, P)$ // $y_p = 1$

26: else

27: $feasible \leftarrow \text{False}$

28: $assignment \leftarrow \text{None}$

29: $num_{active} \leftarrow \text{infinity}$

30: end if

31: // --- Етап 4: Розрахунок метрик ефективності ---

32: if $\text{IsSmallInstance}(V, P)$ and $feasible$ then

33: $C_{opt} \leftarrow \text{ExactCQMSolver}(CQM).get_optimal_active_servers()$

34: $r \leftarrow \frac{C_{opt}}{num_{active}}$ // Коефіцієнт апроксимації

35: else

36: $r \leftarrow \text{None}$

37: end if

38: return ($assignment, num_{active}, feasible, r$)

Таблиця 1

Бібліотеки та їх версії, використані для локальної симуляції

Компонент	Бібліотека	Версія	Призначення
Гейт-модельна симуляція QAOA	qiskit, qiskit-aer	2.5.2 / 0.17.2	Побудова та виконання квантових схем на StatevectorSimulator
Класична оптимізація параметрів	scipy (COBYLA)	1.17.1	Зовнішній цикл підбору (γ, β)
CQM/BQM/Ising-формулювання	dimod, dwave-ocean-sdk	0.12.22 / 9.4.0	Побудова обмеженої моделі, конвертація CQM→BQM→Ising
Локальна імітація відпалу	dimod.SimulatedAnnealingSampler	входить у dimod 0.12.22	annealing-парадигма (Розділ 4.4)
Точний перебір (ground truth)	dimod.ExactCQMSolver	входить у dimod 0.12.22	Еталонні розв'язки для approximation ratio (Розділ 4.3)
Класичний базовий рівень	ortools (CP-SAT)	9.15.6755	Точний/near-exact класичний розв'язувач (Розділ 4.2)
Хмарний доступ до квантового процесора (англ. Quantum Processing Unit, QPU)	amazon-braket-sdk	1.126.0	Верифікація на реальному обладнанні через Amazon Braket

3.3. Засоби моделювання. Розроблене програмне забезпечення симуляції може бути використано як для локальної симуляції, так і для обчислень за допомогою сервісу Amazon Braket. Обчислення для локальної симуляції CQM та QAOA виконано у хмарному контейнерному обчислювальному середовищі на базі віртуальної машини Ubuntu 24.04 LTS, Python 3.12.3, 1 vCPU Intel Xeon @ 2.10 GHz, 4 ГБ RAM. У табл. 1 представлені використані бібліотеки та їх версії.

Оскільки CQM/BQM/QAOA-симуляції обмежені кількістю бінарних змінних (≤ 18 -20 кубітів через експоненційну складність моделювання вектора стану), а не обчислювальною потужністю, характеристики середовища (1 vCPU, 4 ГБ RAM) не є вузьким місцем для задач такого розміру. Для перевірки методу на реальному квантовому обладнанні (Rigetti, IQM, IonQ) обчислення виконуються не локально, а на боці провайдера через Amazon Braket, локальна конфігурація впливає лише на етап реалізації моделі задачі та надсилання запиту.

Вхідними даними є: штучний набір віртуальних машин і фізичних серверів, який згенеровано генератором з розподілом розмірів small/medium/large у пропорції 40%/40%/20%, а також профілі фізичних серверів R630/R740/R830/R940, як представлено в [30]. *Діапазон розмірів задачі:* від двох ФС та чотирьох ВМ для точної перевірки відповідності, представленої в розділі 4.3, до 32-х фізичних серверів для порівняння MBFD та CP-SAT, як представлено в розділі 4.2. Для локальних квантових методів (SimulatedAnnealingSampler, QAOA) розмір обмежено до 5-ти ФС та 12-ти ВМ і приблизно 18-20 кубітів відповідно через обчислювальні обмеження локальної симуляції. Під розміром задачі будемо розуміти початкову кількість ФС та ВМ до оптимізації.

Вихідні дані. Для кожного запуску квантової схеми зберігаються для аналізу такі дані: кількість активних ФС, час виконання, статус допустимості рішення (feasible/infeasible), коефіцієнт апроксимації (де застосовно), а для QAOA, додатково, траєкторія збіжності параметрів (γ, β) за ітераціями оптимізатора та фінальний розподіл вимірювань (measurement counts).

4. Результати. Запропонована модель ЦОД та алгоритм розміщення ВМ на ФС реалізовані мовою Python. Результати отримані без використання квантового апаратного забезпечення, але розроблене програмне забезпечення моделювання може безпосередньо використовувати квантові процесори, що доступні за допомогою сервісу Amazon Braket.

4.1. Опис порівняльного дослідження. Для комплексної оцінки застосовності квантових та квантово-натхнених методів до задачі розміщення ВМ використано єдину CQM-модель, представлену в розділі 3.1 статті, як базу для застосування чотирьох принципово різних класів розв'язувачів:

- **MBFD** (англ. Modified Best-Fit Decreasing) – класична жадібна евристика, що слугує швидким базовим рівнем для порівняння без гарантій отримання оптимального рішення;
- **CP-SAT** (Google OR-Tools) – точний (англ. near-exact) класичний розв'язувач з обмеженнями, що дає верхню межу якості класичних методів;
- **SimulatedAnnealingSampler** (з використанням D-Wave Ocean SDK, який використаний для локального запуску без хмарного доступу) – класична імітація квантового відпалу, представник annealing-парадигми квантових обчислень;
- **QAOA** (Qiskit Aer, локальний симулятор з моделюванням вектора стану) – представник

вентильної (англ. gate-model) парадигми квантових обчислень.

Такий підхід дозволяє провести порівняльний аналіз квантових фреймворків (SimulatedAnnealingSampler і QAOA), які є прямими класичними імітаціями двох основних архітектур квантових обчислень (annealing та gate-model відповідно) та розв'язують ту саму QUBO/Ising-форму задачі. Їх якість роботи та масштабованість оцінюються відносно класичних базових алгоритмів (MBFD та CP-SAT), а також точного розв'язку ExactQMSolver на малих кількостях ФС і ВМ, як показано в розділі 4.3 статті. Це забезпечує змістовне порівняння вже на етапі локальної симуляції, без доступу до реального квантового обладнання. Проаналізувати результати моделювання, отримані на реальних QPU (Rigetti Cephelus, IQM Garnet, IonQ Forte), плануються у наступній статті завдяки отриманим дослідницьким кредитам від AWS.

У цій статті порівняння проведено за трьома показниками: (а) якість рішення – кількість активних ФС відносно точного оптимуму та коефіцієнт апроксимації відносно точного оптимуму, (б) час виконання, (в) межа масштабованості (максимальний розмір задачі, при якому метод надійно дає допустиме рішення).

4.2. Порівняння класичних алгоритмів розміщення ВМ. На базовому (класичному) рівні порівняно два алгоритми: евристичний Modified Best-Fit Decreasing (MBFD) та точний/евристичний розв'язувач Google OR-Tools CP-SAT, що використовують ту саму математичну модель обмежень (призначення ВМ, обмеження ємності по CPU/RAM, поріг завантаження $T=0.9$), що й квантова CQM-модель. Це забезпечує коректність подальшого порівняння результатів, отриманих різними розв'язувачами. Експерименти проведено у діапазоні від 4-х до 32-х фізичних серверів (до оптимізації) із трьома класами розміру (small/medium/large у пропорції 40%/40%/20% з профілями R630/R740/R830/R940 з [30]). Отримані залежності часу виконання та кількості вивільнених ФС від розміру задачі показують очікувану картину: MBFD виконується практично миттєво на всіх розмірах задачі за рахунок жадібної евристики, тоді як CP-SAT забезпечує кращу якість консолідації та розміщення ВМ (для розміщення тієї ж кількості ВМ використовується менше активних ФС) ціною суттєво більших витрат часу на виконання обчислень для великих розмірів задачі (багато ФС і ВМ). Результати порівняння алгоритмів MBFD та CP-SAT для різної кількості фізичних серверів наведено в табл. 2 та проілюстровано на рис. 1.

Таблиця 2

Якісні показники роботи алгоритмів MBFD та CP-SAT в залежності від розміру задачі

M (кількість ФС)	N (кількість ВМ)	MBFD: кількість активних ФС	MBFD: час роботи алгоритму, мс	CP-SAT: кількість активних ФС	CP-SAT: час роботи алгоритму, мс	Оптимум CP-SAT
4	12	2	0.00	1	110.5	так
8	24	3	0.00	1	22.8	так
12	36	4	0.00	2	30.3	так
16	48	6	0.10	2	53.9	так
20	60	7	0.10	3	90.8	так
24	72	8	0.10	3	135.3	так
28	84	9	0.10	4	202.3	так
32	96	11	0.20	4	248.6	так

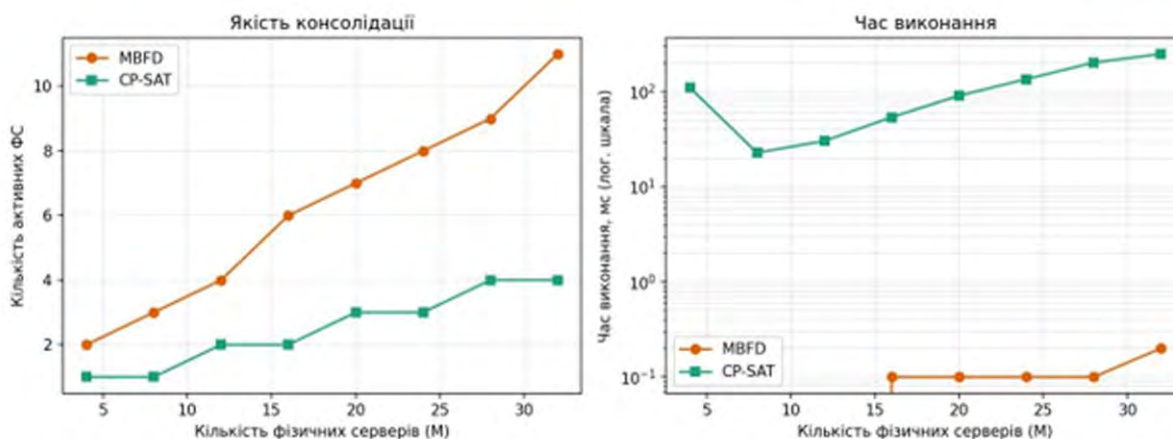


Рис. 1. Якісні результати роботи класичних алгоритмів MBFD та CP-SAT

4.3. Перевірка коректності CQM-моделі.

Перш ніж застосовувати наближені методи (симуляцію відпалу, QAOA), формулювання CQM-моделі перевірено точним перебором усіх можливих комбінацій розміщення через `dimod.ExactCQMSolver`. Через експоненційну складність (2^N , де N – кількість двійкових змінних) перевірка обмежена дуже малими розмірами задачі – до 2-3 ФС і 4-8 ВМ. На тестовій конфігурації (2 сервери, 4 ВМ, 10 бінарних змінних) розв'язувач підтвердив коректність моделі: з 1024 можливих комбінацій 18 виявились допустимими, а оптимальне рішення (1 активний сервер) збіглося з очікуваним результатом консолідації. Ця перевірка слугує еталоном (англ. ground truth) для обчислення коефіцієнту апроксимації в наступних розділах.

4.4. Емпіричний аналіз масштабованості локальної симуляції. Локальна симуляція без хмарного доступу реалізована через конвертацію CQM у BQM (`dimod.cqm_to_bqm`, штрафна форма обмежень) з подальшим розв'язанням через `SimulatedAnnealingSampler`. Тестування виявило чітку межу масштабованості цього підходу: на задачах розміром 4-5 ФС та 10-12 ВМ алгоритм надійно знаходить допустимі рішення (до 33 з 300 прогонів дають допустимий розв'язок навіть при помірній кількості параметрів – 300 читань, 3000 розгорток (англ. sweeps). Однак вже при кількості ФС 6-7 та 15-17 ВМ жодна комбінація параметрів ($penalty_{strength}$ від 20 до 300, до 500 читань, до 5000 розгортки) не дала жодного допустимого рішення. Цей результат не є результатом неправильного налаштування моделі, а системна властивість підходу, що базується на штрафах (англ. penalty-based) без адаптивного підбору штрафів, властивого

хмарним гібридним розв'язувачем, наприклад LeapHybridCQMSampler від D-Wave. Отже, локальна класична симуляція непридатна для використання у масштабах промислового ЦОД, про що свідчать результати, отримані квантовим/гібридним розв'язувачами.

4.5. QAOA на локальному симуляторі. Гейт-модельний підхід до розв'язання задачі розміщення ВМ на ФС на базі тієї ж CQM-моделі реалізовано з використанням алгоритму квантової наближеної оптимізації на симуляторі Qiskit Aer, з класичним оптимізатором COBYLA для підбору варіаційних параметрів (γ, β) . Через експоненційну складність симуляції з моделюванням вектора стану (2^N станів) експерименти обмежені задачами з використанням приблизно 18-20 кубітів.

На тестовій конфігурації (6 віртуальних машин з навантаженням [3, 4, 2, 5, 3, 4] умовних одиниць, 3 фізичні сервери ємністю [12, 12, 12] умовних одиниць, штрафні коефіцієнти $\lambda_1 = \lambda_2 = 6.0$, 21 бінарна змінна) отримано коефіцієнти апроксимації $r = 0.667, 1.0$ та 0.667 для глибини схеми $p = 1, 2$ та 3 відповідно (табл. 3). При глибині схеми $p = 2$ отримано коефіцієнт апроксимації $r = 1.0$, тобто QAOA знайшов те саме оптимальне рішення, що й точний перебір. Оптимальне рішення $C_{opt} = 2$, знайдене розв'язувачем QAOA лише при $p = 2$ точним перебором 2^{21} комбінацій, вимагає активації лише двох з трьох доступних ФС (рис. 2). Отже, результат не покращується монотонно зі зростанням глибини схеми, що обумовлено відомою властивістю QAOA, пов'язаною з чутливістю класичного оптимізатора COBYLA до ландшафту цільової функції, а не із самою глибиною схеми.

Таблиця 3

Якісні показники роботи алгоритму QAOA

Глибина схеми, p	Еталонна реалізація, з допомогою CP-SAT (кількість активних ФС)	Реалізація з допомогою QAOA (кількість активних ФС)	коефіцієнт апроксимації, r	допустимих зі 8000 запусків	час оптимізації, с
1	2	3	0.667	3	18.56
2	2	2	1.0	2	37.56
3	2	3	0.667	6	68.51

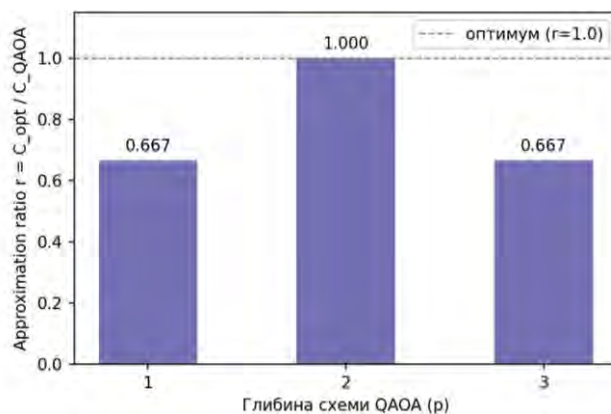


Рис. 2. Якісні показники роботи алгоритму QAOA в залежності від глибини схеми

4.6. Доступ до хмарних квантових ресурсів: обмеження та поточний статус. Аналіз доступності квантового апаратного забезпечення показав, що під час написання цієї статті прямий доступ до трьох основних постачальників хмарних квантових обчислень – D-Wave Leap, IBM Quantum Platform та Microsoft Azure Quantum (для партнерських систем Rigetti, IonQ, Pasqal, Quantinuum), – заблокований для користувачів з України, відповідно до їх офіційної документації щодо географічних обмежень. Amazon Braket [20] виявився єдиним доступним квантовим сервісом для дослідників з України. Завдяки програмі AWS Cloud Credits for Research та AWS Academy авторами отримано згоду від AWS на дослідницькі кредити у розмірі \$2500 для подальшого застосування при дослідженні методів оптимізації розміщення і консолідації ВМ на ФС завдяки доступу до гейт-модельних QPU трьох провайдерів: Rigetti (Cerberus-1-108Q, 108 кубітів), IQM (Garnet, 20 кубітів) та IonQ (Forte-Enterprise-1, 36 кубітів).

D-Wave квантові відпалювачі були виключені з переліку обладнання Amazon Braket ще у 2022 році, тому лише розглянутий у цій статті QAOA (гейт-модельний алгоритм, представлений у розділі 3.2) може бути застосований для експериментальної перевірки на реальному квантовому обладнанні за допомогою сервісу Amazon Braket. Annealing-парадигма (SimulatedAnnealingSampler) залишається предметом виключно класичної локальної симуляції, представленої у розділі 4.4.

Усі результати, представлені в цій статті, отримані виключно шляхом локальної класичної симуляції, включно з симуляцією квантового відпалу (SimulatedAnnealingSampler) та гейт-модельної QAOA-схеми (Qiskit Aer, StatevectorSimulator). Перевірка розробленого програмного забезпечення на реальному квантовому обладнанні за допомогою сервісу Amazon Braket запланована як предмет окремої наступної публікації після завершення узгодження прав доступу.

5. Висновки. У статті розроблено адаптовану модель ЦОД для задачі розміщення віртуальних машин на фізичних серверах з метою мінімізації кількості активних ФС, та проведено порівняльний аналіз чотирьох класів розв'язувачів цієї задачі в єдиній QUBO/CQM-постановці: класичної жадібної евристики (MBFD), точного класичного розв'язувача обмежень (CP-SAT), локальної імітації квантового відпалу (SimulatedAnnealingSampler) та гейт-модельного варіаційного квантового алгоритму QAOA на локальному симуляторі Qiskit Aer.

Основні отримані результати:

- коректність CQM-моделі підтверджена точним перебором (ExactCQMSolver) для невеликих розмірів задач (2 ФС, 4 ВМ), що дає можливість використовувати модель як еталон для подальших порівнянь;
- MBFD і CP-SAT демонструють очікуваний компроміс: MBFD виконується практично миттєво на всіх розмірах задачі ціною гіршої

якості консолідації, CP-SAT дозволяє отримати кращі розв'язки ціною суттєво більшого часу обчислень для великих розмірів задачі (до 32 ФС);

- локальна імітація квантового відпалу має чітку емпіричну межу масштабованості, оскільки допустимі рішення отримуються для розміру задачі до 5 ФС та 12 ВМ; жодна комбінація параметрів (penalty_strength, reads, sweeps) не дала допустимого рішення для задачі з 6-7 ФС. Це не артефакт налаштування, а системна властивість підходу, що базується на штрафних, без адаптивного підбору штрафів, притаманного хмарним гібридним розв'язувачам.
- QAOA на локальному симуляторі демонструє немонотонну залежність якості рішення від глибини схеми: коефіцієнт апроксимації становив 0.667, 1.0 та 0.667 для $p = 1, 2$ та 3 відповідно (6 ВМ, 3 ФС, 21 бінарна змінна). Оптимальне значення досягається не при максимальній глибині схеми, а залежить від збіжності класичного оптимізатора COBYLA, тобто розв'язувач може знайти оптимальне значення у межах обчислювально доступного розміру задачі ($\leq 18-20$ кубітів).

Поточні проблеми в схемах управління ресурсами хмарних ЦОД вимагають розроблення та вдосконалення методів консолідації ВМ з урахуванням гетерогенності ресурсів, «живої» міграції ВМ, параметрів мережі та СЗД, платформ розгортання, моделей навантаження та прогнозування використання ресурсів. Окремим напрямом подальших досліджень є запуск розробленого методу QAOA на реальному квантовому обладнанні (з урахуванням апаратного шуму, спотворення коефіцієнта апроксимації залежно від глибини схеми p та кількості кубітів) з оформленням результатів цих експериментів у вигляді окремої публікації.

Література

1. Flexera 2026 State of the Cloud Report: The convergence of cloud and value. URL: <https://www.flexera.com/blog/finops/flexera-2026-state-of-the-cloud-report-the-convergence-of-cloud-and-value/> (дата звернення: 30.05.2026).
2. Хмарний ринок України у 2025 році. URL: <https://delo.ua/news-companies/xmarnii-rinok-ukrayini-u-2025-roci-451838/> (дата звернення: 05.06.2026).
3. Barroso L. A., Hölzle U., Ranganathan P. The Datacenter as a Computer: Designing Warehouse-Scale Machines. 2026. 347 p. DOI: <https://doi.org/10.1007/978-3-031-99489-0>.
4. Rozekhani S. M., Mahan F., Pedrycz W. VM consolidation steps in cloud computing: A perspective review // Simulation Modelling Practice and Theory. 2024. Vol. 138. Article 103034. DOI: <https://doi.org/10.1016/j.simpat.2024.103034>.

5. Lambert S., Ostapenko V., Lefèvre L., Caron E., Orgerie A. C., Fichel B., Grivel R. Revisiting virtual machine consolidation to save resources and energy in heterogeneous production cloud infrastructures // *The International Journal of High Performance Computing Applications*. 2026. Vol. 40, No. 3. P. 383–403. DOI: <https://doi.org/10.1177/10943420251399694>.
6. Zharikov E., Telenyk S. Performance Analysis of a Dynamic Virtual Machine Management Method Based on the Power-Aware Integral Estimation // *Electronics*. 2021. Vol. 10, No. 21. Article 2581. DOI: <https://doi.org/10.3390/electronics10212581>.
7. Telenyk S., Zharikov E., Rolik O. Consolidation of virtual machines using simulated annealing algorithm // 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT). IEEE, 2017. Vol. 1. P. 117–121.
8. El-Taani I., Boukala M. C., Bouzeffrane S., Amrous A. I. Efficient VM consolidation: deep reinforcement learning approach based PM workload awareness // *Annals of Telecommunications*. 2026. Vol. 81, No. 3. P. 191–210. DOI: <https://doi.org/10.1007/s12243-025-01100-5>.
9. Panwar S. S., Rauthan M. M. S., Barthwal V., Mehra N., Bhatt C., Rawat R. S. Efficient Energy-Aware Genetic approach for Placing Virtual Machines in cloud data centers // *Procedia Computer Science*. 2026. Vol. 283. P. 1113–1122.
10. Rana N., Jeribi F., Khan Z., Alrawagfeh W., Ben Dhaou I., Haseebuddin M., Uddin M. A systematic literature review on contemporary and future trends in virtual machine scheduling techniques in cloud and multi-access computing // *Frontiers in Computer Science*. 2024. Vol. 6. Article 1288552. DOI: <https://doi.org/10.3389/fcomp.2024.1288552>.
11. Jiang Y., Wang X., Liang Z., Li Y., Morstyn T., Lopes P. L., Brandsema M. J. QAOA-Driven PMU placement optimization with graph learning-based parameter initialization refinement // *Quantum Machine Intelligence*. 2026. Vol. 8, No. 1. Article 32. DOI: <https://doi.org/10.1007/s42484-026-00354-z>.
12. Farhi E., Goldstone J., Gutmann S. A quantum approximate optimization algorithm : arXiv preprint arXiv:1411.4028. 2014. URL: <https://arxiv.org/pdf/1411.4028> (дата звернення: 10.06.2026).
13. Peruzzo A., McClean J., Shadbolt P., Yung M. H., Zhou X. Q., Love P. J., O'Brien J. L. A variational eigenvalue solver on a photonic quantum processor // *Nature Communications*. 2014. Vol. 5, No. 1. Article 4213. DOI: <https://doi.org/10.1038/ncomms5213>.
14. Tackle Complex Problems with Annealing Quantum Computing. URL: <https://www.dwavequantum.com/solutions-and-products/systems/annealing-quantum-computing/> (дата звернення: 21.06.2026).
15. Amoretti M., Ferrari D., Manzalini A. Classical and Quantum Solvers for Joint Network/Servers Power Optimization : arXiv preprint arXiv:2205.01165. 2022. URL: <https://arxiv.org/pdf/2205.01165> (дата звернення: 27.06.2026).
16. Patil M., Mhetre S. S. Optimizing Distributed Systems With Quantum Computing // *International Journal of Advances in Signal and Image Sciences*. 2025. P. 683–700.
17. Divya N., Kiranbabu M. N. V., Babu G. C. Quantum-Inspired Adaptive Meta-Heuristic–Machine Learning framework for resilient and energy-efficient task scheduling in multi-cloud ecosystems // *Scientific Reports*. 2026. Vol. 16, No. 1. Article 21212. DOI: <https://doi.org/10.1038/s41598-026-43125-3>.
18. Jayaraman S., Premkumar R., Raguraman H., Visuwasam L. M. M. A Quantum-Assisted Framework for Energy-Efficient Cloud Task Scheduling Using the Quantum Approximate Optimization Algorithm and Variational Quantum Eigensolver // 2025 International Conference on Computing Technologies & Data Communication (ICCTDC). IEEE, 2025. P. 1–6.
19. Christeson T., Ullah M. H., Arabnya A., Khodaei A., Fan R. Hybrid Quantum-Classical Optimization of the Resource Scheduling Problem // *Next Research*. 2026. Article 101350. DOI: <https://doi.org/10.1016/j.nexres.2026.101350>.
20. Amazon Braket Getting Started. URL: <https://aws.amazon.com/ru/braket/getting-started/> (дата звернення: 29.06.2026).
21. Welcome to Quantum Cloud Services. URL: <https://docs.rigetti.com/qcs/> (дата звернення: 30.06.2026).
22. IBM Quantum Platform. URL: <https://quantum.cloud.ibm.com/> (дата звернення: 30.06.2026).
23. Cirq, an open source framework for programming quantum computers. URL: <https://quantumai.google/cirq> (дата звернення: 01.07.2026).
24. Azure Quantum Computing | Microsoft Azure. URL: <https://azure.microsoft.com/en-us/solutions/quantum-computing/#features> (дата звернення: 01.07.2026).
25. Strangeworks homepage. URL: <https://strangeworks.com/> (дата звернення: 02.07.2026).
26. Goyal S., Awasthi L. K., Garg V., Kumar G. Multi-resource aware virtual machine consolidation approach for modern cloud data centers // *Computing*. 2025. Vol. 107, No. 11. P. 1–32. DOI: <https://doi.org/10.1007/s00607-025-01563-3>.
27. Goyal S., Awasthi L. K. Adaptive multi-objective virtual machine consolidation for energy-efficient cloud data centers // *Journal of Grid Computing*. 2025. Vol. 23, No. 2. Article 21. DOI: <https://doi.org/10.1007/s10723-025-09808-3>.
28. Жаріков Е. В., Теленик С. Ф. Спеціалізоване програмне забезпечення для моделювання динамічної консолідації віртуальних машин // *Проблеми програмування*. 2022. № 1. С. 3–12.
29. Singh J., Walia N. K. A comprehensive review of cloud computing virtual machine consolidation // *IEEE Access*. 2023. Vol. 11. P. 106190–106209. DOI: <https://doi.org/10.1109/ACCESS.2023.3314613>.
30. Zharikov E., Telenyk S., Rolik O., Serdiuk Y. Cloud resource management with a hybrid virtual machine consolidation approach // 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT). IEEE, 2019. P. 289–294. DOI: <https://doi.org/10.1109/ATIT49449.2019.9030459>.
31. Powell M. J. A direct search optimization method that models the objective and constraint functions by linear interpolation // *Advances in Optimization and Numerical Analysis*. Dordrecht : Springer Netherlands, 1994. P. 51–67. DOI: https://doi.org/10.1007/978-94-015-8330-5_4.

References

- Flexera 2026 State of the Cloud Report: The convergence of cloud and value. URL: <https://www.flexera.com/blog/finops/flexera-2026-state-of-the-cloud-report-the-convergence-of-cloud-and-value/> (дата звернення: 30.05.2026).
- Khmarnyi ryнок Ukrainy u 2025 rotsi. URL: <https://delo.ua/news-companies/xmarinii-rynok-ukrayini-u-2025-roci-451838/> (дата звернення: 05.06.2026).
- Barroso L. A., Hölzle U., Ranganathan P. The Datacenter as a Computer: Designing Warehouse-Scale Machines. 2026. 347 p. DOI: <https://doi.org/10.1007/978-3-031-99489-0>.
- Rozezhkhan S. M., Mahan F., Pedrycz W. VM consolidation steps in cloud computing: A perspective review // Simulation Modelling Practice and Theory. 2024. Vol. 138. Article 103034. DOI: <https://doi.org/10.1016/j.simpat.2024.103034>.
- Lambert S., Ostapenko V., Lefèvre L., Caron E., Orgerie A. C., Fichel B., Grivel R. Revisiting virtual machine consolidation to save resources and energy in heterogeneous production cloud infrastructures // The International Journal of High Performance Computing Applications. 2026. Vol. 40, No. 3. P. 383–403. DOI: <https://doi.org/10.1177/10943420251399694>.
- Zharikov E., Telenyk S. Performance Analysis of a Dynamic Virtual Machine Management Method Based on the Power-Aware Integral Estimation // Electronics. 2021. Vol. 10, No. 21. Article 2581. DOI: <https://doi.org/10.3390/electronics10212581>.
- Telenyk S., Zharikov E., Rolik O. Consolidation of virtual machines using simulated annealing algorithm // 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT). IEEE, 2017. Vol. 1. P. 117–121.
- El-Taani I., Boukala M. C., Bouzeffrane S., Amrous A. I. Efficient VM consolidation: deep reinforcement learning approach based PM workload awareness // Annals of Telecommunications. 2026. Vol. 81, No. 3. P. 191–210. DOI: <https://doi.org/10.1007/s12243-025-01100-5>.
- Panwar S. S., Rauthan M. M. S., Barthwal V., Mehra N., Bhatt C., Rawat R. S. Efficient Energy-Aware Genetic approach for Placing Virtual Machines in cloud data centers // Procedia Computer Science. 2026. Vol. 283. P. 1113–1122.
- Rana N., Jeribi F., Khan Z., Alrawagfeh W., Ben Dhaou I., Haseebuddin M., Uddin M. A systematic literature review on contemporary and future trends in virtual machine scheduling techniques in cloud and multi-access computing // Frontiers in Computer Science. 2024. Vol. 6. Article 1288552. DOI: <https://doi.org/10.3389/fcomp.2024.1288552>.
- Jiang Y., Wang X., Liang Z., Li Y., Morstyn T., Lopes P. L., Brandsema M. J. QAOA-Driven PMU placement optimization with graph learning-based parameter initialization refinement // Quantum Machine Intelligence. 2026. Vol. 8, No. 1. Article 32. DOI: <https://doi.org/10.1007/s42484-026-00354-z>.
- Farhi E., Goldstone J., Gutmann S. A quantum approximate optimization algorithm : arXiv preprint arXiv:1411.4028. 2014. URL: <https://arxiv.org/pdf/1411.4028> (дата звернення: 10.06.2026).
- Peruzzo A., McClean J., Shadbolt P., Yung M. H., Zhou X. Q., Love P. J., O'Brien J. L. A variational eigenvalue solver on a photonic quantum processor // Nature Communications. 2014. Vol. 5, No. 1. Article 4213. DOI: <https://doi.org/10.1038/ncomms5213>.
- Tackle Complex Problems with Annealing Quantum Computing. URL: <https://www.dwavequantum.com/solutions-and-products/systems/annealing-quantum-computing/> (дата звернення: 21.06.2026).
- Amoretti M., Ferrari D., Manzanini A. Classical and Quantum Solvers for Joint Network/Servers Power Optimization : arXiv preprint arXiv:2205.01165. 2022. URL: <https://arxiv.org/pdf/2205.01165> (дата звернення: 27.06.2026).
- Patil M., Mhetre S. S. Optimizing Distributed Systems With Quantum Computing // International Journal of Advances in Signal and Image Sciences. 2025. P. 683–700.
- Divya N., Kiranbabu M. N. V., Babu G. C. Quantum-Inspired Adaptive Meta-Heuristic-Machine Learning framework for resilient and energy-efficient task scheduling in multi-cloud ecosystems // Scientific Reports. 2026. Vol. 16, No. 1. Article 21212. DOI: <https://doi.org/10.1038/s41598-026-43125-3>.
- Jayaraman S., Premkumar R., Raguraman H., Visuwasam L. M. M. A Quantum-Assisted Framework for Energy-Efficient Cloud Task Scheduling Using the Quantum Approximate Optimization Algorithm and Variational Quantum Eigensolver // 2025 International Conference on Computing Technologies & Data Communication (ICCTDC). IEEE, 2025. P. 1–6.
- Christeson T., Ullah M. H., Arabnya A., Khodaei A., Fan R. Hybrid Quantum-Classical Optimization of the Resource Scheduling Problem // Next Research. 2026. Article 101350. DOI: <https://doi.org/10.1016/j.nexres.2026.101350>.
- Amazon Braket Getting Started. URL: <https://aws.amazon.com/ru/braket/getting-started/> (дата звернення: 29.06.2026).
- Welcome to Quantum Cloud Services. URL: <https://docs.rigetti.com/qcs/> (дата звернення: 30.06.2026).
- IBM Quantum Platform. URL: <https://quantum.cloud.ibm.com/> (дата звернення: 30.06.2026).
- Cirq, an open source framework for programming quantum computers. URL: <https://quantumai.google/cirq> (дата звернення: 01.07.2026).
- Azure Quantum Computing | Microsoft Azure. URL: <https://azure.microsoft.com/en-us/solutions/quantum-computing/#features> (дата звернення: 01.07.2026).
- Strangeworks homepage. URL: <https://strangeworks.com/> (дата звернення: 02.07.2026).
- Goyal S., Awasthi L. K., Garg V., Kumar G. Multi-resource aware virtual machine consolidation approach for modern cloud data centers // Computing. 2025. Vol. 107, No. 11. P. 1–32. DOI: <https://doi.org/10.1007/s00607-025-01563-3>.
- Goyal S., Awasthi L. K. Adaptive multi-objective virtual machine consolidation for energy-efficient cloud data centers // Journal of Grid Computing. 2025. Vol. 23, No. 2. Article 21. DOI: <https://doi.org/10.1007/s10723-025-09808-3>.
- Zharikov E. V., Telenyk S. F. Spetsializovane prohramne zabezpechennia dlia modeliuvannia dynamichnoi konsolidatsii virtualnykh mashyn // Problemy prohramuvannia. 2022. № 1. S. 3–12.
- Singh J., Walia N. K. A comprehensive review of cloud computing virtual machine consolidation // IEEE Access.

2023. Vol. 11. P. 106190–106209. DOI: <https://doi.org/10.1109/ACCESS.2023.3314613>.
30. Zharikov E., Telenyk S., Rolik O., Serdiuk Y. Cloud resource management with a hybrid virtual machine consolidation approach // 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT). IEEE, 2019. P. 289–294. DOI: <https://doi.org/10.1109/ATIT49449.2019.9030459>.
31. Powell M. J. A direct search optimization method that models the objective and constraint functions by linear interpolation // Advances in Optimization and Numerical Analysis. Dordrecht : Springer Netherlands, 1994. P. 51–67. DOI: https://doi.org/10.1007/978-94-015-8330-5_4.

Zharikov E.V., Kuzemskyi M.V., Polupan Yu.V., Zarichkovyi O.A., Lishchuk K.I. Model and method for virtual machine consolidation in a cloud data center using quantum algorithms.

The application of quantum and quantum-classical algorithms for solving a wide range of optimization problems is gaining increasing prevalence. This article proposes the application of quantum-classical algorithms for solving the problem of virtual machine allocation across physical servers in a cloud data center, which belongs to the class of bin packing problems. Cloud data centers are becoming increasingly complex and large-scale, necessitating the development of more efficient virtual machine consolidation algorithms to maximize resource utilization and minimize power consumption. Scheduling methods employed in the early stages of cloud computing development are simpler to implement but perform poorly in dynamic scenarios, as they do not account for complex constraints and do not support multi-objective optimization. Under these conditions, quantum computing provides the capability to overcome combinatorial challenges due to its ability to process multiple states simultaneously.

The virtual machine consolidation problem is formulated in QUBO/Ising form. A comparative analysis of quantum frameworks (SimulatedAnnealingSampler and QAOA), which are direct classical emulations of two fundamental quantum computing architectures (annealing and gate-model, respectively), and classical baseline algorithms MBFD and CP-SAT has been performed in terms of scalability (problem size), solution quality (number of active physical servers relative to the exact optimum and approximation ratio relative to the exact optimum), as well as algorithm execution time. The proposed data center model and hybrid quantum-classical algorithm for virtual machine placement on physical servers are implemented in Python.

The results presented in this article were obtained without the use of quantum hardware; however, the developed simulation software can directly utilize quantum processors available through the Amazon Braket service.

Keywords: *quantum simulation, quantum cloud service, quantum solver, quantum approximate optimization algorithm, CUBO, XQM, quantum annealing*

Жаріков Едуард В'ячеславович – д-р техн. наук, професор, завідувач кафедри інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

ORCID: <https://orcid.org/0000-0003-1811-9336>

e-mail: eduard.zharikov-fiot@iitl.kpi.ua

Куземський Максим Володимирович – студент кафедри інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

ORCID: <https://orcid.org/0009-0002-4954-7042>

e-mail: kuzemskyi.m.v.-ip51@edu.kpi.ua

Полупан Юлія Вікторівна – к.т.н., доцент, доцент кафедри інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

ORCID: <https://orcid.org/0009-0000-0243-824X>

e-mail: juliyapolupan@i.ua

Зарічковий Олександр Анатолійович – Ph. D., асистент кафедри інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

ORCID: <https://orcid.org/0000-0002-4132-6424>

e-mail: alexander.zarichkovyi@gmail.com

Ліщук Катерина Ігорівна – к.т.н., доцент, доцент кафедри інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

ORCID: <https://orcid.org/0000-0002-9902-0065>

e-mail: lishchuk_kpi@ukr.net

Дата першого надходження статті 05.07.2026.

Дата прийняття статті до друку після рецензування 26.08.2025.

Дата публікації 30.09.2026.



Стаття з відкритим доступом,
відповідно до умов ліцензії

[Creative Commons \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)