

Сопов О.О.

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Жаріков Е.В.

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ДЛЯ СИМУЛЯЦІЇ РОЗМІЩЕННЯ КОНТЕЙНЕРІВ У ХМАРНОМУ ЦЕНТРІ ОБРОБКИ ДАНИХ

Використання контейнерної віртуалізації для розгортання застосунків у хмарі набирає все більшого попиту, адже дозволяє робити це з меншою кількістю надлишкових ресурсів, та швидше, ніж класичний варіант віртуалізації – з використанням віртуальних машин. Перед хмарними провайдерами постає задача ефективного управління ресурсами, а саме ефективного розміщення контейнерів на обчислювальних вузлах. Така задача вимагає розроблення та застосування ефективних стратегій, спрямованих на зменшення кількості одночасно включених фізичних серверів, на зменшення часу порушення SLA, та на покращення інших цільових метрик хмарних обчислень. Вивчення нових підходів розміщення контейнерів із використанням виробничого середовища є дорогим і нестабільним, адже може заважати стабільній роботі вже існуючих хмарних послуг, що може призвести до порушення SLA. У статті розроблена математична модель трирівневої задачі розміщення контейнерного навантаження та інструментарій для симуляції розміщення контейнерів у хмарному центрі обробки даних (ЦОД). Запропоновані алгоритми роботи симулятора та наведено діаграми діяльності, класів та компонентів інструменту симуляції, та їх детальний опис. Запропонований у статті інструментальний засіб для симуляції розміщення контейнерів у хмарі дозволяє проаналізувати різні стратегії управління кластером фізичних машин і віртуальних машин на них, робити експерименти для вибору оптимальних стратегій та експериментувати, не використовуючи існуюче виробниче середовище. Запропонована архітектура дозволяє легко додавати нові стратегії керування шляхом наслідування від існуючих абстракцій. Реалізовано широкий спектр можливостей логування результатів експериментів, що дозволяє проаналізувати роботу стратегій для різних критеріїв оптимізації. Реалізовано можливість конфігурування параметрів ЦОД, та генерації запитів на розміщення контейнерів за заданим бета-розподілом. Проведено серію експериментів, які підтверджують правильність, стабільність та ідемпотентність розробленого інструменту, наведено результати проведення експериментів.

Ключові слова: контейнерна віртуалізація, хмарний ЦОД, розміщення контейнерів, оптимізація, архітектура, симулятор.

Постановка проблеми. У сучасному світі інформаційних технологій віртуалізація на основі контейнерів відіграє ключову роль у розробці та розгортанні програмного забезпечення, виходячи на передній план завдяки своїй гнучкості та ефективності. Провайдери хмарних послуг стикаються зі складним завданням оптимізації розміщення контейнерів, яке вимагає не тільки великої масштабованості, але й високої відповідності до змінюваних вимог користувачів та додатків.

Вивчення нових підходів розміщення контейнерів із використанням виробничого середовища є дорогим і нестабільним, адже може заважати стабільній роботі вже існуючих хмарних послуг, що може призвести до порушення SLA [1–3].

Симулятор розміщення контейнерів може зіграти важливу роль у зниженні ризиків, вартості та складності інфраструктури перш ніж рішення

можна буде розгорнути на реальній інфраструктурі [4]. Зосереджуючись на питаннях, пов'язаних із якістю певного компонента за різних сценаріїв, симулятори дозволяють інженерам відстежувати поведінку системи.

Отже, розробка ефективних інструментальних засобів для симуляції розміщення контейнерів є необхідною для хмарних ЦОД, адже дозволить експериментувати з новими стратегіями розміщення контейнерів, одночасно не впливаючи на виробниче середовище.

Інструмент, запропонований у цій статті, базується на симуляції різних сценаріїв розміщення контейнерів з використанням комплексних параметрів, таких як модель системи, вхідне робоче навантаження, політики управління ресурсами: фізичними серверами (ФС) та віртуальними машинами (ВМ), кількість і типи серверів, та типи

віртуальних машин у ЦОД. Дозволяє моделювати процеси створення нових контейнерів, а також отримувати всю основну статистику у будь-який момент часу симуляції: кількість порушень SLA (Service Level Agreement), кількість ФС та ВМ, використання ресурсів на кожному ФС та ВМ.

Аналіз останніх досліджень і публікацій.

Задача розміщення ресурсів у хмарному ЦОД представлена в численних дослідженнях, присвячених різним аспектам оптимізації ІТ-інфраструктури. Великої уваги серед науковців набрала тема розміщення ВМ на ФС, яка є однією із задач при розміщенні контейнерів.

У [4–7] автори аналізують велику кількість робіт, де представлені хмарні симулятори. Розглянуті симулятори створені для певних цілей і мають різні функції в кожному з них. Автори представляють всебічне дослідження основних хмарних симуляторів, висвітливши їх важливі функції та аналізують їх переваги та недоліки. Велика кількість симуляторів потребує більш гнучкого налаштування, можливості впливати на більшу кількість показників, більш обширне логування та кращі можливості експорту результатів симуляції. Проаналізовані симулятори не надають функціональності симуляції розміщення контейнерів. Більшість симуляторів беруть до уваги обмежену кількість параметрів ресурсної ємності, а саме, процесорний час та оперативну пам'ять.

Автори у роботах [8–9] використовують можливості симулятора Cloudsim [10], метою якого є надання узагальненої структури моделювання, яка дає змогу моделювати, симулювати та експериментувати з інфраструктурами хмарних обчислень, дозволяючи користувачам зосередитися на конкретних питаннях проектування системи, які вони хочуть дослідити, не турбуючись про низькорівневі параметри, пов'язані з хмарними інфраструктурами та службами. Cloudsim має можливість роботи із контейнерним навантаженням, проте така можливість є обмеженою: відсутня можливість динамічно змінювати стратегії керування ФС та ВМ, параметр ресурсної ємності IOPS (Input/Output Operations Per Second, кількість операцій вводу/виводу за секунду) не аналізується, можливості логування обмежені.

Набір інструментів GridSim [11] був розроблений авторами для вирішення проблеми оцінки продуктивності реальних великомасштабних розподілених середовищ повторюваним і контрольованим способом. Проте, симуляція розміщення контейнерів відсутня.

Постановка завдання. Метою дослідження є вирішення науково-практичної задачі створення

симулятора розміщення контейнерів у хмарному ЦОД задля розроблення та аналізу нових стратегій розміщення контейнерів із використанням середовища, близького до виробничого, що дозволить покращити існуючі рішення, шляхом: створення архітектури для симуляції не тільки розміщення віртуальних машин, а також і контейнерів; збільшення кількості параметрів, які аналізуються (CPU, RAM, IOPS, дисковий простір); покращення логування та методів експорту результатів експерименту.

У результаті, це дослідження спрямоване на розширення наукових та практичних знань у сфері управління хмарними ресурсами, використовуючи сучасні інструменти моделювання для вирішення конкретних задач оптимізації розміщення контейнерів. Такий підхід дозволить реалізувати більш ефективні, надійні та гнучкі хмарні сервіси відповідно до високих стандартів ІТ-індустрії.

Виклад основного матеріалу

Задача розміщення контейнерів

Розміщення контейнерів є однією з важливих задач управління ресурсами хмарних ЦОД, яка полягає в тому, щоб визначити оптимальний спосіб розподілу контейнерів між фізичними та віртуальними серверами. Контейнери – це легковагові ізольовані середовища для запуску різних застосунків у хмарі, які мають ряд переваг перед традиційними віртуальними машинами, такі як: швидше розгортання, менша витрата ресурсів, більша гнучкість та масштабованість. Однак, розміщення контейнерів також ставить перед дослідниками та практиками складні задачі оптимізації, пов'язані з балансуванням навантаження, енергоефективністю, надійністю та якістю обслуговування.

Провайдер хмарних послуг стикається з трьохрівневим завданням динамічної консолідації контейнерів, яке вимагає оперативного реагування на поточні запити клієнтів в онлайн-режимі. Завдання розміщення контейнера розділяється на дві ключові підзадачі: по-перше, вибір відповідної віртуальної машини для запуску контейнера, а у разі її відсутності – створення нової ВМ та її розміщення на доступному фізичному сервері. По-друге, якщо необхідний фізичний сервер не включений, слід його перевести у активний стан. У випадку відсутності доступних фізичних ресурсів, запит клієнта на послугу може бути відхилений, що призводить до порушення умов SLA.

Оптимізаційною задачею провайдера хмарних послуг є мінімізація використання ресурсів, що в свою чергу, досягається зменшенням часу порушення SLA, та зниження кількості одночасно активних ФС.

У структурі хмарного провайдера наявні фізичні сервери, позначені як $PM = \{PM_1, \dots, PM_n\}$, кожен з яких може перебувати в увімкненому або вимкненому стані. На i -му фізичному сервері можуть розташовуватися віртуальні машини, що позначаються як $VM = \{VM_{i,1}, \dots, VM_{i,m}\}$. В свою чергу, на j -ій віртуальній машині розміщуються контейнери, представлені як $C = \{C_{i,j,1}, \dots, C_{i,j,k}\}$. Ресурсна ємність кожного з цих компонентів описується в рівнянні 1:

$$resourceCapacity_i = (w_i^{dim1}, w_i^{dim2} \dots w_i^{dimn}), \quad (1)$$

де dim – ресурс, для якого визначається ресурсна ємність. Тому для PM , VM і C ресурсна ємність описується як (2–4):

$$PM_i = (PM_i^{CPU}, PM_i^{RAM}, PM_i^{DISK}, PM_i^{IOPS}), \quad (2)$$

$$VM_i = (VM_i^{CPU}, VM_i^{RAM}, VM_i^{DISK}, VM_i^{IOPS}), \quad (3)$$

$$C_i = (C_i^{CPU}, C_i^{RAM}, C_i^{DISK}, C_i^{IOPS}), \quad (4)$$

де $CPU, RAM, DISK, IOPS$ – ресурсна ємність за процесором, оперативною пам'яттю, дисковим простором та кількістю операцій вводу/виводу за секунду.

На рисунку 1 зображено трирівневу модель розміщення контейнерів у хмарному ЦОД. Контейнери розміщено на віртуальних машинах, які у свою чергу розміщено на фізичних серверах у ЦОД [12].

Обмеження розміщення VM на i -му фізичному сервері показано у нерівності (5). Для розміщення віртуальної машини на фізичному сервері справедливо:

$$\sum_{j=1}^N VM_{i,j} x_{i,j} \leq PM_i, \quad (5)$$

де N – кількість віртуальних машин,

i – номер фізичного сервера,

$x_{i,j} = 1$, якщо віртуальну машину j розміщено на фізичному сервері i , інакше $x_{i,j} = 0$.

Обмеження розміщення контейнера на j -й віртуальній машині показано у нерівності (6). Для розміщення контейнера на віртуальній машині справедливо:

$$\sum_{k=1}^K C_{i,j,k} y_{j,k} \leq VM_{i,j}, \quad (6)$$

де K – кількість контейнерів,

i – номер фізичного сервера,

j – номер віртуальної машини,

$y_{j,k} = 1$, якщо контейнер k розміщено на віртуальній машині j , інакше $y_{j,k} = 0$.

Архітектура та алгоритми інструментарію

Розглянемо розроблену архітектуру інструментальних засобів для симуляції динамічного розміщення контейнерів у хмарному ЦОД, а саме – діаграму класів, діаграму компонентів, та діаграму діяльності симуляції розміщення контейнерів. На рисунку 2 показано діаграму діяльності одного циклу симуляції розміщення контейнерів у хмарному ЦОД.

При запиті клієнта на розміщення контейнера з ресурсною ємністю C , симулятор виконуватиме наступні дії. Відповідно до (6), контейнер може бути розміщений на віртуальній машині лише за умови достатньої вільної ресурсної ємності $r(VM_i)$. На початку циклу симуляції обирається стратегія VMP по вибору віртуальної машини, на яку далі буде розміщено контейнер. Можливість вибору стратегії під час симуляції дозволяє тестувати динамічний вибір стратегій, з можливістю реагування на зовнішні чинники. За допомогою обраної стратегії VMP, обирається віртуальна машина, або приймається рішення про необхідність створення нової. Серед наявних розроблених стратегій є: First Fit, Best Fit, та Worst Fit [13]. Проте, архітектура інструменту легко масштабована, і можна додати нові стратегії

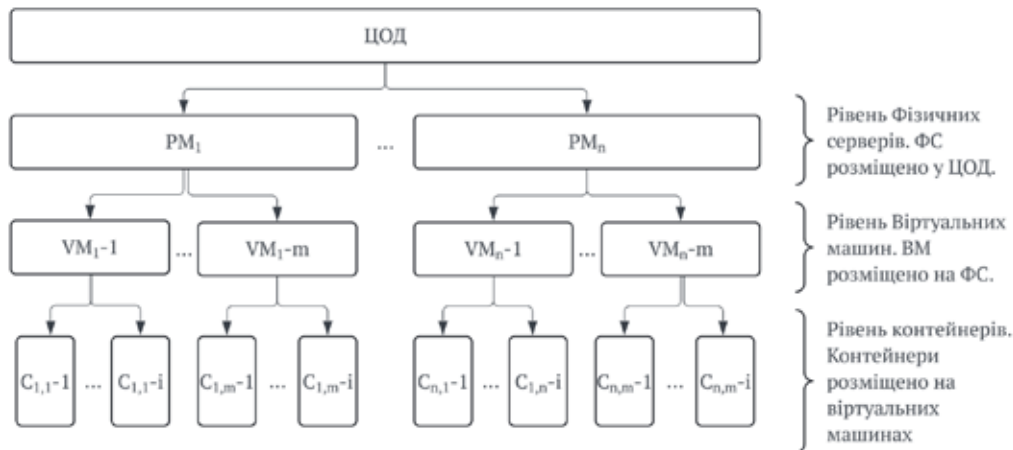


Рис. 1. Тривірнева модель розміщення контейнерів у хмарному ЦОД

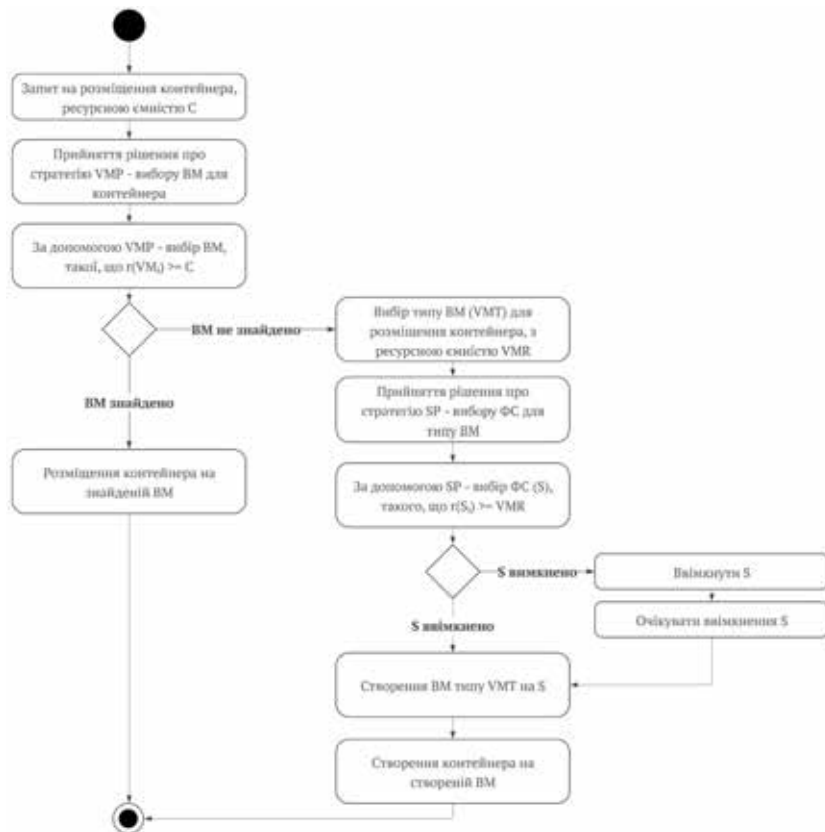


Рис. 2. Діаграма діяльності одного циклу симуляції розміщення контейнерів у хмарному ЦОД

за необхідності, такі, як, наприклад, розроблені авторами у роботах [14–15]. Якщо обрано існуючу віртуальну машину, то контейнер розміщується на ній та алгоритм завершує роботу.

Проте, якщо прийнято рішення про створення нової віртуальної машини, то обирається тип віртуальної машини, необхідної для розміщення контейнера та обирається ФС, на якому розмістити таку віртуальну машину. Вибір ФС відбувається за допомогою обраної стратегії розміщення серверів SP. Якщо знайдений сервер ввімкнено, то віртуальна машина розміщується на цьому сервері, а контейнер на віртуальній машині. Інакше – ФС вмикається (переходить з режиму резерву до робочого), і виконуються всі вище перелічені дії.

На рисунку 3 зображено діаграму класів інструменту симуляції розміщення контейнерів у хмарному ЦОД. Для наочності, невелику частину класів, таких як класи конфігурації, опущено. Ключовим тут є клас Simulator – ядро процесу симуляції. Основним параметром, який подається на вхід симулятора є об’єкт SimulatorConfig. На базі даної конфігурації збирається об’єкт класу CloudProvider, який відповідатиме на симульовані запити на розміщення контейнера. Будь-який запит на розміщення контейнера, в першу чергу, оброблюється у класі CloudProvider.

CloudProvider для розміщення контейнерів використовує VMSelectionStrategy та ServerSelectionStrategy. VMSelectionStrategy відповідає за вибір одного з підкласів VMSelector, в залежності від поточної обраної стратегії. VMSelector безпосередньо обирає необхідну віртуальну машину для розміщення контейнера або генерує запит на створення нової.

ServerSelectionStrategy відповідає за вибір одного з підкласів ServerSelector в залежності від поточної обраної стратегії. ServerSelector безпосередньо обирає необхідний сервер для розміщення віртуальної машини, і, за потреби, вмикає його.

Наявність VMSelectionStrategy та ServerSelectionStrategy дозволяє динамічно, в режимі реального часу, керувати стратегіями вибору віртуальних машин і серверів. Кожна подія симуляції логується у Reporter, що дозволяє дізнатись стан симуляції у будь-який момент часу, та дозволяє експортувати результати у необхідному форматі.

CloudProvider має доступ до DALService, який комунікує з базою даних, що дає можливість під час симуляції ретроспективно аналізувати оброблені запити для динамічної зміни стратегії керування.

На рисунку 4 показано діаграму компонентів інструменту симуляції розміщення контейнерів у хмарному ЦОД.

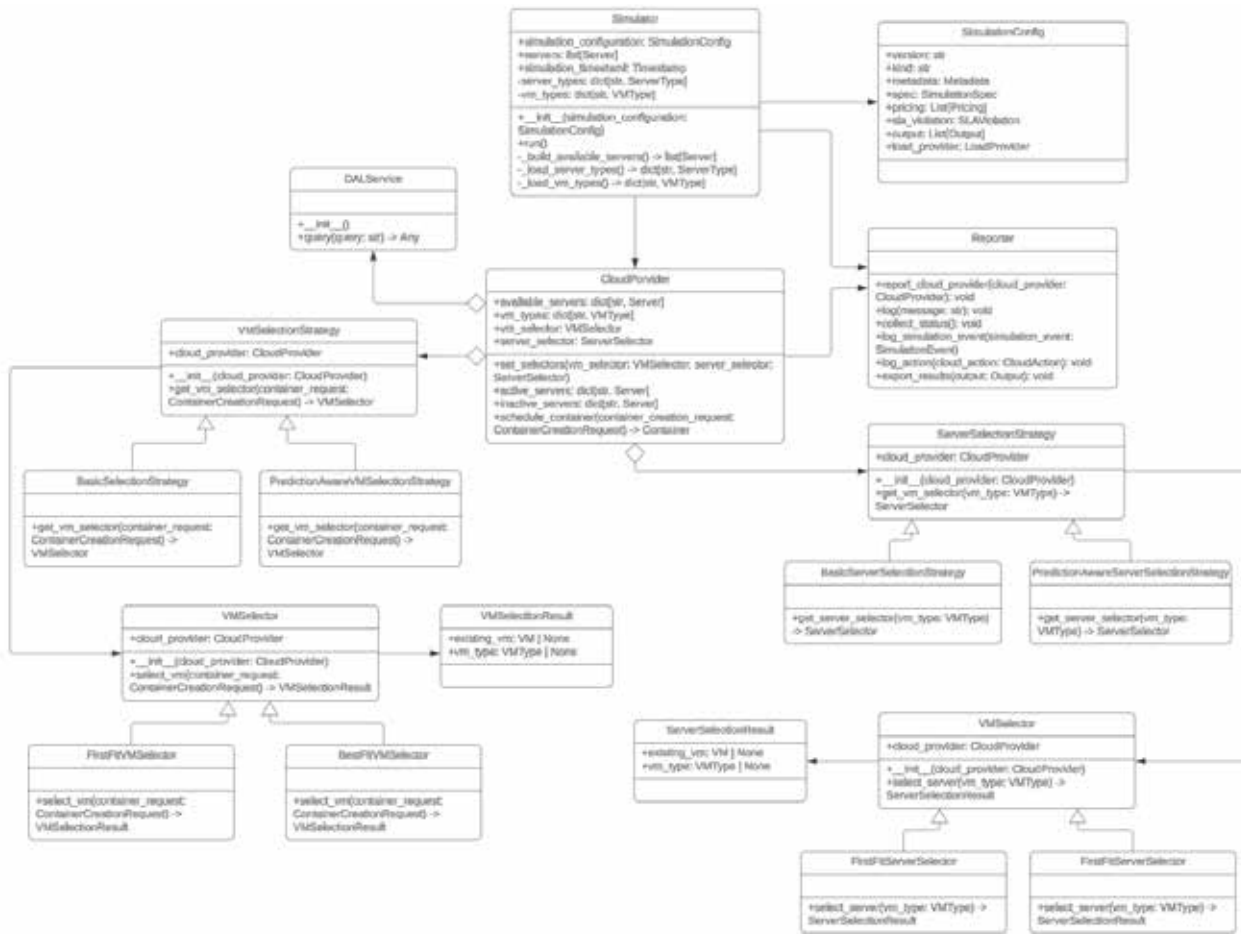


Рис. 3. Діаграма класів інструменту симуляції розміщення контейнерів у хмарному ЦОД

Пакет Simulation містить основну логіку симуляції, реалізацію покрокового виконання симуляції. Пакет Configuration містить логіку десеріалізації файлів конфігурації та, безпосередньо, моделі конфігурації. Пакет CloudProvider містить всю логіку розміщення контейнерів у хмарному ЦОД, логіку вибору VM та ФС. Пакет DAL містить логіку роботи з базою даних.

Проведення експериментів

Реалізацію інструментальних засобів для симуляції динамічного розміщення контейнерів у хмарному ЦОД виконано за допомогою мови програмування Python [16]. Конфігурацію експерименту задано за допомогою Yaml файлу, зображеного на рисунку 5.

Навантаження для експерименту генерується випадковим чином у бета-розподілі. Файл конфігурації для генерації навантаження випадковим чином показано на рисунку 6.

Для запуску інструменту розроблено інтерфейс командного рядка, результат роботи якого показано на рисунку 7.

На випадково згенерованому (за бета-розподілом) навантаженні проведено серію із 10 експе-

риментів. Експериментально підтверджено, що результати експерименту на однаковому наборі даних однакові, що свідчить про стабільність інструментарію та його ідемпотентність. Розроблений інструмент можливо використовувати для порівняння різних стратегій розміщення контейнерів у хмарному ЦОД.

Розроблений інструментарій має широкі можливості логування: кожен крок симуляції логується, і статус симуляції відображається у консолі та файлі логу. Наприклад, на рисунку 8 зображено стан симуляції після декількох її кроків.

Результати експерименту логуються у файли журналу у форматі JSON, що дає змогу візуалізувати такі дані у будь-якому зручному інструменті, наприклад, за допомогою MS Excel, matplotlib, тощо. Логується інформація про стан всіх серверів, віртуальних машин, контейнерів, логуються запити на розміщення контейнерів та фактичний час їх розгортання. На рисунку 9 показано навантаження на сервери по CPU. Візуалізацію виконано шляхом візуалізації даних, які експортовані інструментом.

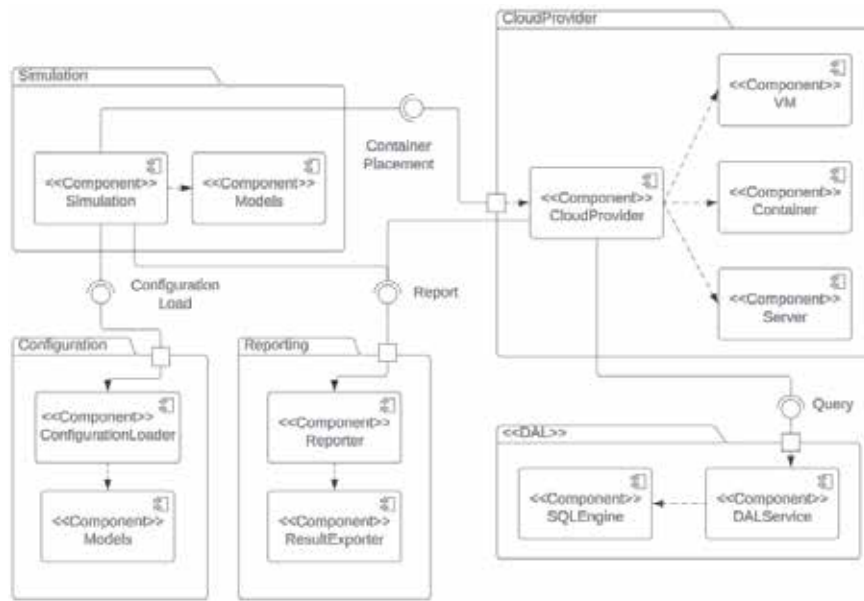


Рис. 4. Діаграма компонентів інструменту симуляції розміщення контейнерів у хмарному ЦОД

```

version: v1
kind: Simulation
metadata:
  name: experiment-1
spec:
  available:
    - server_type: x-large
      quantity: 10
    - server_type: x-medium
      quantity: 5
  server_types:
    - name: x-medium
      seconds_to_up: 20
      resource_capacity:
        cpu: 3
        memory: 16384
        storage: 256000
        iops: 40000
    - name: x-large
      seconds_to_up: 50
      resource_capacity:
        cpu: 24
        memory: 131072
        storage: 1024000
        iops: 60000
  vm_types:
    - name: x-small
      seconds_to_up: 5
      resource_capacity:
        cpu: 1.5
        memory: 8192
        storage: 126000
        iops: 15000
    - name: x-medium
      seconds_to_up: 5
      resource_capacity:
        cpu: 8
        memory: 43690
        storage: 341333
        iops: 20000
    - name: x-large
      seconds_to_up: 5
      resource_capacity:
        cpu: 24
        memory: 65536
        storage: 512000
        iops: 30000
  vm_selection_strategy: BasicVMSelectionStrategy
  server_selection_strategy: BasicServerSelectionStrategy
  pricing:
    - cpu: 0.01
      memory: 0.0001
      storage: 0.00001
      iops: 0.000001
  sla_violation:
    cost: 0.5
  output:
    type: json
    path: results/result.json
  load_file:
    type: csv
    path: load/output.csv
    
```

Рис. 5. Файл конфігурації експерименту

```

version: v1
kind: LoadGeneration
type: random_ranged
distribution:
  type: beta_distribution
  alpha: 2
  beta: 5
output:
  type: csv
  path: load/output.csv
time_between_requests:
  low: 1
  high: 1000
requests: 130
ranges:
  cpu:
    low: 0.5
    high: 3
  memory:
    low: 256
    high: 3096
  storage:
    low: 1024
    high: 10240
  iops:
    low: 100
    high: 2000
    
```

Рис. 6. Файл конфігурації навантаження

Usage: main.py [OPTIONS] COMMAND [ARGS]...

```

Options
--help Show this message and exit.

Commands
generate-load Generate load based on the configuration file.
simulate Run the simulation based on the load configuration file.
    
```

Рис. 7. Інтерфейс командного рядка інструменту

STEP: 4

reporter.py:107
reporter.py:107

Cloud Provider Status

Server	VM	Container	CPU	Memory	Disk	IOPS	State
959eb861-ecd9-44a5-8d...	7f600602-d1ff-4354-98...		8.0/24.0	43690.0/131072.0	341333.0/1024000.0	20000.0/60000.0	Status.ACTIVE
			5.7/8.0	5598.94/43690.0	24650.69/341333.0	2700.31/20000.0	Status.ACTIVE
		ab77970c-1ea7-4066-95...	1.75	972.97	6007.46	526.27	Status.ACTIVE
		84088920-fe1c-4a0a-a3...	0.76	1455.22	4215.18	513.88	Status.ACTIVE
		49160c6d-cd79-420e-b9...	0.99	642.73	4375.55	382.9	Status.ACTIVE
		690679c3-4b5f-4805-93...	1.31	1286.43	5672.49	242.33	Status.ACTIVE
		0c231d16-4714-4154-9e...	0.89	1241.59	4380.01	1042.93	Status.ACTIVE
7e137d7d-3319-46bf-ba...			0.0/24.0	0.0/131072.0	0.0/1024000.0	0.0/60000.0	Status.INACTIVE
974872e2-7b9c-4809-89...			0.0/24.0	0.0/131072.0	0.0/1024000.0	0.0/60000.0	Status.INACTIVE

Рис. 8. Можливості логування інформації

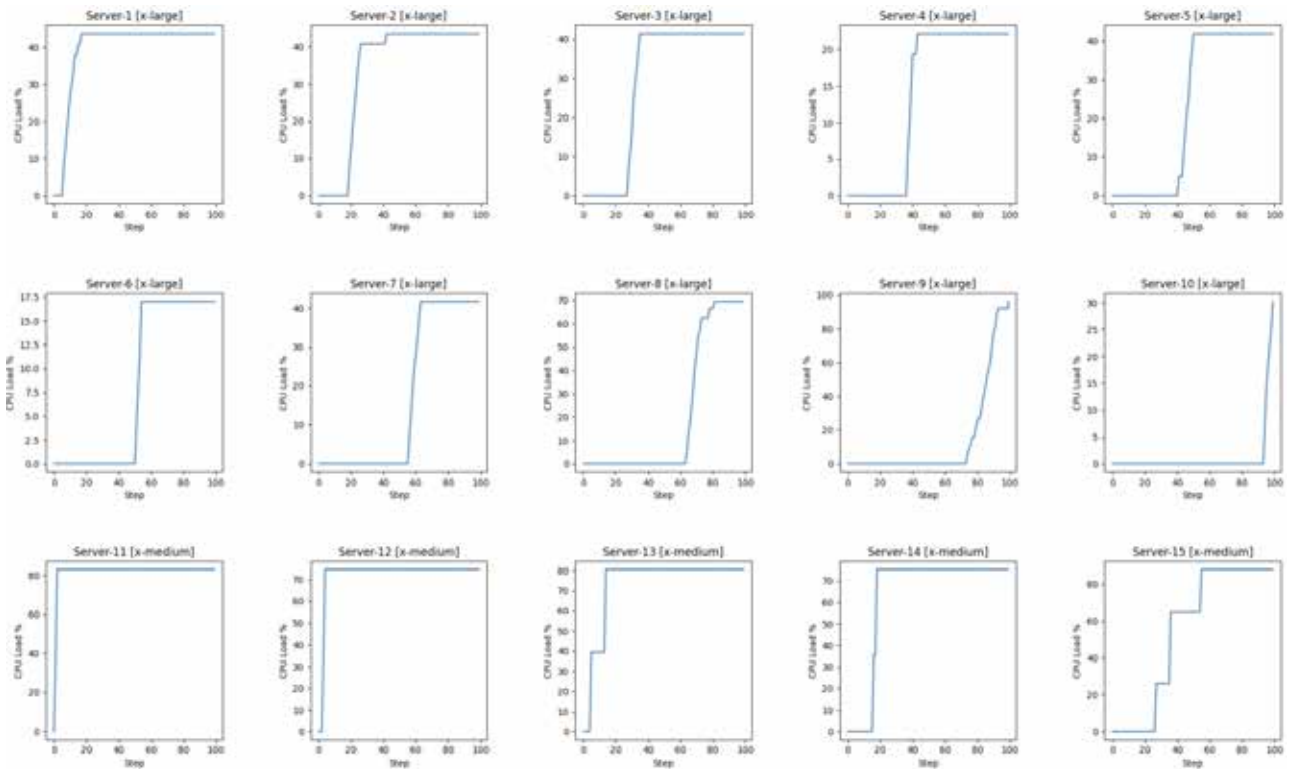


Рис. 9. Графік використання CPU на кожному з фізичних серверів

Під час експерименту обрано WorstFit стратегію, яка залишалась сталою протягом всього часу експерименту. Використання такої стратегії дозволяє виконувати енергоефективне управління хмарним ЦОД. З графіків видно, що сервери включаються почергово. Надлишкові сервери заздалегідь не включено, що дозволяє зменшити використання електроенергії.

Завдяки експортованим даним є можливість варіативно візуалізувати дані та проаналізувати їх під багатьма доступними розрізами.

Висновки. Використання симуляторів для розміщення контейнерів є критично важливим, оскільки використання симулятора дозволяє знизити витрати, ефективно використовувати ресурси, зменшувати складність експериментів та ризики перед впровадженням нових рішень у реальну IT-інфраструктуру.

Аналіз показав, що наявні хмарні симулятори мають обмежену функціональність роботи із контейнерами, не враховують всі необхідні параметри ресурсної ємності та мають обмежені можливості логування та експорту результатів.

У статті запропоновано новий інструмент-симулятор розміщення контейнерів у хмарному ЦОД, який дозволяє моделювати розміщення контейнерів з урахуванням різних сценаріїв та параметрів ресурсної ємності, таких як CPU, RAM, IOPS та дисковий простір. Інструмент підтримує динамічне налаштування стратегій розміщення

контейнерів, що дозволяє адаптуватися до змінних умов.

Детально описано архітектуру розробленого інструменту, включаючи діаграми класів, компонентів та діяльності. Розроблена архітектура дозволяє легко масштабувати рішення для експериментів з будь-якими стратегіями розміщення як VM, так і ФС.

Проведено серію експериментів, які підтвердили стабільність та ідемпотентність роботи інструменту. Результати експериментів показали можливість використання інструменту для порівняння різних стратегій розміщення контейнерів та ефективного управління ресурсами хмарного ЦОД.

Інструмент забезпечує детальне логування всіх етапів симуляції та підтримує експорт результатів у зручних форматах для подальшого аналізу та візуалізації, що дозволяє глибше розуміти процеси та оптимізувати управління ресурсами хмарного ЦОД.

Розроблений інструментальний засіб дозволяє аналізувати нові підходи розміщення контейнерів із використанням середовища, близького до виробничого.

Подальші дослідження необхідно зосередити на розробленні ефективних методів керування контейнерами у хмарному ЦОД для трирівневої моделі розміщення контейнерів, додаванні можливостей керування міграціями у розроблений інструментальний засіб, а також інтеграцію з OpenStack.

Список літератури:

1. Liao W., Draper J. Cloud Computing and Docker Containerization: A Survey. *Proceedings of the West Virginia Academy of Science*. 2019. Т. 91. № 1. URL: <https://doi.org/10.55632/pwvas.v91i1.505>.
2. Pusapati S. S. V. Containerization. *Advances in Systems Analysis, Software Engineering, and High Performance Computing*. 2024. С. 98–122. URL: <https://doi.org/10.4018/979-8-3693-1682-5.ch006>.
3. Raghav Goel and Dr. Bhoomi Gupta. Introduction to Containerization. *International Journal for Modern Trends in Science and Technology*. 2020. Т. 6. № 12. P. 294–300. URL: <https://doi.org/10.46501/ijmtst061255>.
4. Mansouri N., Ghafari R., Zade B. M. H. Cloud computing simulators: A comprehensive review. *Simulation Modelling Practice and Theory*. 2020. Т. 104. P. 102–144. URL: <https://doi.org/10.1016/j.simpat.2020.102144>.
5. Building a cloud on earth: A study of cloud computing data center simulators / M. Abu Sharkh та ін. *Computer Networks*. 2016. Т. 108. С. 78–96. URL: <https://doi.org/10.1016/j.comnet.2016.06.037>.
6. Ahmed A., Sabyasachi A. S. Cloud computing simulators: A detailed survey and future direction. *2014 IEEE International Advance Computing Conference (IACC)*. Gurgaon, India, 21–22 feb. 2014. URL: <https://doi.org/10.1109/iadcc.2014.6779436>.
7. Modeling and simulation of cloud computing: A review / W. Zhao та ін. *2012 IEEE Asia Pacific Cloud Computing Congress (APCloudCC)*. Shenzhen, China, 14–17 nov. 2012. URL: <https://doi.org/10.1109/apcloudcc.2012.6486505>.
8. Ismail A. Energy-driven cloud simulation: existing surveys, simulation supports, impacts and challenges. *Cluster Computing*. 2020. Т. 23/ № 4. P. 3039–3055. URL: <https://doi.org/10.1007/s10586-020-03068-4>.
9. Buyya R., Ranjan R., Calheiros R. N. Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. *Simulation (HPCS)*. Leipzig, Germany, 21–24 june. 2009. URL: <https://doi.org/10.1109/hpcsim.2009.5192685>.
10. CloudSim: A Framework For Modeling And Simulation Of Cloud Computing Infrastructures And Services. URL: <https://github.com/Cloudslab/cloudsim>.

11. Buyya R., Murshed M. GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing. *Concurrency and Computation: Practice and Experience*. 2002. Т. 14, № 13–15. Р. 1175–1220. URL: <https://doi.org/10.1002/cpe.710>.

12. Сопов О., Жаріков Е. Підхід до консолідації контейнерів у хмарних центрах обробки даних. *Адаптивні системи автоматичного управління*. 2023. Т. 2. № 43. С. 154–163. URL: <https://doi.org/10.20535/1560-8956.43.2023.292267>.

13. S K., Nair M. K. Bin packing algorithms for virtual machine placement in cloud computing: a review. *International Journal of Electrical and Computer Engineering (IJECE)*. 2019. Т. 9. № 1. С. 512. URL: <https://doi.org/10.11591/ijece.v9i1.pp512-524>.

14. Katal A., Choudhury T., Dahiya S. Energy optimized container placement for cloud data centers: a meta-heuristic approach. *The Journal of Supercomputing*. 2023. URL: <https://doi.org/10.1007/s11227-023-05462-2>.

15. 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)*. Lviv, Ukraine, 5–8 sept. 2017. URL: <https://doi.org/10.1109/stc-csit.2017>.

16. Мова програмування Python. URL: <https://www.python.org/>.

Sopov O.O., Zharikov E.V. INSTRUMENTAL TOOLS FOR CONTAINER PLACEMENT SIMULATION IN CLOUD DATA CENTERS

The use of container virtualization for deploying applications in the cloud is gaining increasing demand as it allows deployment with fewer redundant resources and faster than the classical virtualization option using virtual machines. Cloud providers face the challenge of efficient resource management, specifically the effective placement of containers on computing nodes. This task requires the development and application of efficient algorithms aimed at reducing the number of simultaneously active physical servers, minimizing SLA violation times, and improving other target metrics of cloud computing. Studying new container placement approaches using a production environment is costly and unstable, as it can interfere with the stable operation of existing cloud services, potentially leading to SLA violations. The article develops a mathematical model for a three-level container load placement problem and tools for simulating container placement in a cloud data center (CDC). The proposed algorithms for the simulator are presented along with activity diagrams, class and component diagrams of the simulation tool, and their detailed descriptions. The proposed tool for simulating container placement in the cloud allows analyzing different strategies for managing a cluster of physical and virtual machines on them, conducting experiments to select optimal strategies, and experimenting without using the existing production environment. The proposed architecture allows for easy addition of new management strategies by inheriting from existing abstractions. A wide range of capabilities for logging experiment results is implemented, enabling the analysis of strategy performance for various optimization criteria. The ability to configure CDC parameters and generate container placement requests according to a given beta distribution is also implemented. A series of experiments confirming the correctness, stability, and idempotence of the developed tool was conducted, and the results of these experiments are presented.

Key words: container virtualization, cloud data center, container placement, optimization, architecture, simulator.