УДК 004.42,94

О. Сопов, Е. Жаріков

ПІДХІД ДО КОНСОЛІДАЦІЇ КОНТЕЙНЕРІВ У ХМАРНИХ ЦЕНТРАХ ОБРОБКИ ДАНИХ

Анотація. У сучасному інформаційному світі хмарні обчислення стають ключовим елементом інфраструктури для вирішення різноманітних завдань. Контейнерна віртуалізація зараз усе частіше замінює класичні віртуальні машини, адже використання контейнерів забезпечує легку ізоляцію, масштабованість і швидкість запуску застосунків, що дозволяє ефективно використовувати ресурси серверів і спрощує розгортання програм. За збільшенням кількості застосунків і їх ресурсної ємності, у хмарного ЦОД виникає потреба в ефективному розміщенні контейнерів для оптимізації витрат. У цій роботі наведене математичне представлення задачі розміщення контейнерів. Автори пропонують модифікацію Best Fit алгоритму, використовуючи прогнозування запитів клієнтів на розміщення контейнерів. Розроблений алгоритм дає можливість зменшити загальні витрати хмарного ЦОД на 5 % і загальну ціну порушення SLA на 30 %, проте збільшує вартість обслуговування ФС на 1 %. Результати дослідження свідчать про ефективність розробленого алгоритму в контексті оптимізації ресурсного управління у хмарних ЦОД.

Ключові слова: контейнерна віртуалізація, контейнер, хмарні обчислення, SLA, алгоритм прогнозування, симуляція.

Вступ

У сучасному світі інформаційних технологій хмарні обчислення стають необхідним елементом для вирішення різноманітних завдань: від бізнес-застосунків до наукових досліджень. Велика кількість організацій та індивідуальних користувачів обирають хмарні рішення для забезпечення високої доступності, масштабованості та зниження витрат на обладнання. Однак, зі зростанням обчислювальних вимог і розширенням ролі програмного забезпечення, виникає потреба в більш ефективному управлінні обчислювальними ресурсами.

У контексті цих змін вирішальним стає використання віртуалізації та контейнеризації, що дозволяє розподіляти ресурси та ізолювати різні додатки на спільних віртуальних машинах. Віртуалізація, що базується на розділенні фізичних серверів на віртуальні машини, давно вважається засобом оптимізації використання обчислювальних ресурсів [1]. Однак останнім часом усе більшою популярністю користується контейнерна віртуалізація, яка дає можливість створювати ізольовані середовища для застосунків із меншими витратами ресурсів на функціонування та зберігання. Одним із найпопулярніших рішень для контейнерної віртуалізації є Docker [2, 3].

 ISSN 1560-8956

Зі збільшенням кількості застосунків відповідно зростає кількість контейнерів, які мають бути розгорнуті у хмарі. Тому перед провайдером хмарних послуг постає задача ефективної консолідації контейнерів із метою мінімізації операційних витрат і збільшення обсягу надання послуг, використовуючи наявні ресурси.

Консолідація контейнерів із точки зору провайдера хмарних послуг розглядається як задача розміщення кількох контейнерів на одній віртуальній машині за умови розміщення кількох віртуальних машин на одному фізичному сервері [4]. Перед провайдерами хмарних послуг постає задача онлайн-консолідації контейнерів з огляду на тип надання послуг.

Під час консолідації важливо враховувати різноманітні фактори та вимоги, як-от обчислювальна потужність фізичних серверів і віртуальних машин на них, запити на ресурсну ємність контейнерів від клієнтів, зменшення використання електроенергії [5, 6] тощо.

Беручи до уваги онлайн-природу запитів на розміщення контейнерів, доцільно використовувати прогнозування майбутніх запитів клієнтів із метою вчасного ввімкнення необхідних ресурсів. У цій статті розглядається задача розміщення контейнерів і запропоновано новий алгоритм із прогнозування майбутніх запитів клієнтів із метою зменшення загальних витрат хмарного ЦОД водночас зі зменшенням часу порушення SLA.

Аналіз попередніх досліджень

Проблема консолідації обчислювальних ресурсів представлена в численних дослідженнях, присвячених різним аспектам оптимізації використання інфраструктури в хмарних обчисленнях. Великої уваги набрала тема консолідації віртуальних машин (ВМ) на фізичних серверах (ФС), що є однією з підзадач консолідації контейнерів.

У роботі [7] автори розглядають проблему консолідації віртуальних машин як Bin Packing, описують різні методи та аналізують фактори розміщення ВМ у кожному методі із зазначенням відповідних переваг і недоліків.

Автори роботи [8] пропонують гібридний підхід до консолідації вір туальних машин, спрямований на мінімізацію споживання енергії та порушення SLA шляхом застосування модифікованої евристики Best Fit Decreasing до початкового розміщення віртуальної машини та застосування алгоритму Beam Search для керування міграціями ВМ.

У дослідженні [9] автори вивчають взаємозв’язки між енергоспоживанням, використанням ресурсів і продуктивністю консолідованих робочих навантажень. Автори моделюють проблему консолідації як модифіковану проблему пакування.

Автори [10] пропонують евристичний алгоритм для оптимізації консолідації віртуальних машин. Цей підхід базується на жадібному алгоритмі, який визначає розміщення віртуальних машин на вільних ресурсах фізичних серверів. Як наслідок, демонструється значний приріст ефективності використання обчислювальних ресурсів у порівнянні з традиційними методами.

У дослідженні [11] автори пропонують оптимізацію з використанням алгоритму імітованого відпалу для вирішення проблеми консолідації віртуальних машин. Проблема консолідації віртуальних машин представлена як розширення проблеми Bin Packing. Результати оцінювання показують, що з використанням розробленого алгоритму змодельований центр обробки даних (ЦОД) споживає майже таку ж кількість енергії, як і неоптимізований алгоритм, але розроблений алгоритм дозволяє зменшити порушення SLA.

За останні роки також привертає увагу проблема консолідації контейнерів. Так, у роботі [12] автори розглядають зв’язки між контейнером, віртуальною машиною та фізичним сервером, координуючи розміщення віртуальних сутностей на фізичних за допомогою алгоритму Best Fit.

Автори в [13] пропонують генетичний алгоритм із подвійним представленням хромосом для вирішення задачі розміщення контейнерів. Експерименти показують, що запропонований генетичний алгоритм дає можливість досягти значно вищої енергоефективності ЦОД у порівнянні з класичними.

У працях [14-16] також пропонуються різні методи оптимізації розміщення контейнерів на віртуальних машинах.

Проблема розміщення контейнерів на віртуальних машинах має вирішуватися динамічно, тобто в режимі онлайн, тому доцільно робити прогноз майбутніх запитів клієнтів задля вчасного ввімкнення віртуальної машини з необхідною конфігурацією або ввімкнення фізичного сервера, що дасть можливість тримати баланс між рівнем надання послуг і кількістю операційних витрат з огляду на провайдера хмарних послуг. У попередніх розглянутих працях відсутня оптимізація з використанням прогнозування.

Задача розміщення контейнерів

Перед провайдером хмарних послуг постає трирівнева задача динамічної консолідації контейнерів в онлайн-режимі, залежно від поточних запитів клієнтів. Задача розміщення контейнера складається з двох підзадач: знайти необхідну віртуальну машину та запустити контейнер на ній і, у разі відсутності віртуальної машини, створити її та розмістити на фізичному сервері. У разі відсутності ввімкненого фізичного сервера з достатньою кількістю ресурсів для розміщення віртуальної машини – увімкнути один із наявних вимкнених фізичних серверів. У разі відсутності фізичних ресурсів – відмовити в наданні послуг клієнту (із відповідним порушенням SLA).

Мінімізація часу порушення SLA та мінімізація кількості ввімкнених фізичних серверів є задачею оптимізації для провайдера хмарних послуг.

На рис. 1 схематично зображено трирівневу модель контейнерної вір туалізації, у якій контейнери запускаються на віртуальних машинах, а віртуальні машини – на фізичних серверах.

 ISSN 1560-8956

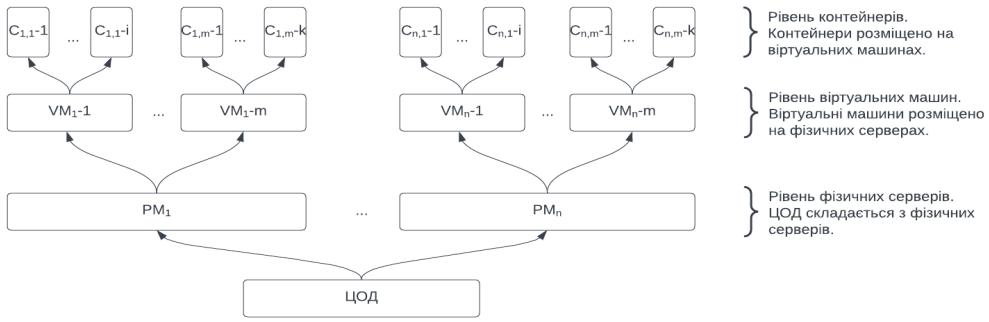

Рис. 1. Трирівнева модель контейнерної віртуалізації ЦОД

Нехай хмарний провайдер має в наявності фізичні сервери PM $= \{PM_1, \dots, PM_n\}$, які знаходяться в увімкненому або вимкненому стані. Віртуальна машина, розміщена на *i*-му фізичному сервері позначається як $VM = \{VM_{i,1}, \dots, VM_{i,m}\}$, а контейнер розміщений на *j*-й віртуальній машині позначається як $C = \{C_{i,j,1}, \dots, C_{i,j,k}\}$. Ресурсну ємність кожного з компонентів визначено в рівнянні 1.

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

де dim – ресурс, для якого визначається ресурсна ємність. Тому для *PM, VM* і *C* виділено ресурсні ємності, представлені рівняннями (2-4):

$$PM_i = \left(PM_i^{CPU}, PM_i^{RAM} \dots PM_i^{DISK}\right), \#(2)$$

$$VM_i = \left(VM_i^{CPU}, VM_i^{RAM} \dots VM_i^{DISK}\right), \#(3)$$

$$C_i = \left(C_i^{CPU}, C_i^{RAM} \dots C_i^{DISK}\right), \#(4)$$

де CPU, RAM і DISK – ресурсна ємність за процесором, оперативною пам’яттю та дисковим простором відповідно.

Нерівність (5) ілюструє обмеження кількості ВМ на *i*-му фізичному сервері. Для розміщення віртуальної машини на фізичному сервері справедливо:

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

де *N* – кількість віртуальних машин, *i* – номер фізичного сервера, $x_{i,j} = 1$, якщо віртуальну машину *j* розміщено на фізичному сервері *i*, інакше $x_{i,j} = 0$.

Нерівність (6) ілюструє обмеження кількості контейнерів на *j*-й ВМ. Для розміщення контейнера на віртуальній машині справедливо:

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

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 $y_{j,k} = 0$.

Нехай, ціна за 1 секунду роботи i-го фізичного сервера становить ω_i, а ціна 1 секунди порушення SLA – φ, а на момент часу t – вартість порушення SLA – $\varphi(t)$. Тоді для хмарного провайдера оптимізаційна задача приймає вигляд (7), яка полягає в мінімізації витрат на обслуговування та порушення SLA в момент часу t.

$$\sum_{i=1}^{N} \omega_i + \varphi(t) \to min \#(7)$$

Прогнозування запитів клієнтів

Розглянуті класичні методи, як-от Best Fit, First Fit та інші, добре вирішують задачу розміщення контейнерів на віртуальних машинах і розміщення віртуальних машин на фізичних серверах за умови, якщо всі ресурси наявні і до них можна швидко отримати доступ. Якщо запит клієнта на запуск контейнера не можна розмістити на наявних увімкнених ресурсах, то виникає необхідність увімкнути фізичний сервер із резерву, що, у свою чергу, займає час і спричинює порушення SLA.

Однак для зменшення витрат на порушення SLA однією з обраних стратегій може бути ввімкнення всіх фізичних серверів, але це призведе до збільшення операційних витрат.

З огляду на роботу хмарного провайдера в онлайн-режимі, тобто майбутні запити клієнтів заздалегідь невідомі, і ймовірну сезонність запитів, прогнозування цих запитів вирішує проблему очікування клієнтом необхідних ресурсів інфраструктури за умови їх активної поточної недостачі. Із метою оптимізації запропоновано прогнозування навантаження для вчасного ввімкнення фізичних серверів із резерву та зменшення часу порушення SLA. Для прогнозування буде використано алгоритм, описаний компанією Meta у праці [17]. Для вибору поточної віртуальної машини або фізичного серверу буде використано алгоритм Best Fit.

Модифікація алгоритму полягає в прогнозуванні майбутніх запитів від клієнтів і відповідну підготовку кластера до цих запитів. Оскільки ввімкнення фізичних серверів і віртуальних машин займає час, то необхідно їх увімкнути заздалегідь із метою зменшення часу порушення SLA.

До життєвого циклу хмарного центру оброблення даних додається етап прогнозування: на кожному такті виконання роботи центру – спрогнозувати майбутні запити клієнтів і прийняти рішення, чи достатньо поточних ресурсів. Якщо ресурсів недостатньо, то слід увімкнути необхідні фізичні сервери та віртуальні машини. Реалізація алгоритму враховує прогнозування наступних п’яти запитів клієнтів і починає вмикати фізичні сервери та віртуальні машини так, щоб наступні п’ять контейнерів могли бути розміщені в кластері.

Запит клієнта на розміщення контейнера з ресурсною ємністю в час t визначено як $C_i = \left(C_i^{CPU}, C_i^{RAM}, C_i^{DISK}, t_i\right)$. Задача хмарного ЦОД полягає в розміщенні заданого

 ISSN 1560-8956

контейнера на віртуальній машині, де кількість вільних ресурсів більше або дорівнює ресурсній ємності заданого контейнера так, щоб виконувалася нерівність (6).

Прогнозування полягає у визначенні $C_{i+1}, C_{i+2}, \dots, C_{i+5}$ запитів користувачів на основі запитів $C_1, C_2, \dots, C_{i-1}$ за допомогою алгоритму, описаного у праці [17]. Для кожного ресурсу формується часовий ряд: $C_i^{CPU}(t_i) = (C_1^{CPU}, t_1)$, (C_2^{CPU}, t_2) ..., $C_i^{RAM}(t_i) = (C_1^{RAM}, t_1)$, (C_2^{RAM}, t_2) ... і $C_i^{DISK}(t_i) = (C_1^{DISK}, t_1)$, (C_2^{DISK}, t_2) ..., і за допомогою алгоритму прогнозуються $C_{i+1}^{CPU}(t_{i+1}), \dots, C_{i+5}^{CPU}(t_{i+5})$, $C_{i+1}^{RAM}(t_{i+1}), \dots, C_{i+5}^{RAM}(t_{i+5})$ і $C_{i+1}^{DISK}(t_{i+1}), \dots, C_{i+5}^{DISK}(t_{i+5})$, які, у свою чергу, формують $C_{i+1}, C_{i+2}, \dots, C_{i+5}$.

Прогнозування виконується лише за наявності щонайменше 10 запитів, тобто за умови $i \geq 10$. На основі отриманих прогнозованих запитів користувачів на розгортання контейнерів ЦОД, за допомогою Best Fit, обирає необхідні ВМ і ФС і вмикає їх, якщо поточних вільних ресурсів недостатньо.

Проведення експерименту й аналіз результатів

У цьому розділі буде проведено експеримент із запропонованим алгоритмом із прогнозуванням запитів клієнтів, описаним у попередньому розділі, а результати експериментів (кумулятивна ціна для хмарного ЦОД, ціна обслуговування фізичних серверів і ціна порушення SLA) порівняні з алгоритмом Best Fit.

Конфігурація експерименту

Серію із 6 симуляцій проведено для гетерогенного хмарного ЦОД, який має в наявності типи фізичних серверів із характеристиками, зазначеними в табл. 1.

Таблиця 1. **Характеристики фізичних серверів ЦОД**

Тип ФС	Оперативна пам’ять, ГБ	Кількість ядер процесора	Об’єм пам’яті, ГБ	Час старту, с
ФС – 1	10	5	500 ГБ	30
ФС – 2	20	10	1 ТБ	60
ФС – 3	40	20	4 ТБ	120

Для ЦОД визначено віртуальні машини з характеристиками, зазначеними в табл. 2.

Таблиця 2. **Характеристики віртуальних машин ЦОД**

Тим ВМ	Оперативна пам’ять, ГБ	Кількість ядер процесора	Об’єм пам’яті, ГБ	Час старту, с
ВМ – 1	2	1	100 ГБ	5
ВМ – 2	4	2	200 ГБ	10
ВМ – 3	8	4	400 ГБ	20
ВМ – 4	20	10	0.7 ТБ	30
ВМ – 5	30	15	1 ТБ	40

У кожному експерименті випадково генеруються запити на розміщення контейнерів, що мають ресурсну ємність від 256 МБ оперативної пам’яті, 0,5 ядра процесора та 10 ГБ фізичної пам’яті до 25 ГБ оперативної пам’яті, 12 ядер CPU та 1 ТБ фізичної пам’яті.

Ціна ресурсів: 1 умовна одиниця за 1 МБ оперативної пам’яті та 2 умовні одиниці за 0,001 ядра процесора. Ціна 1 секунди порушення SLA – 100 умовних одиниць. Порушення SLA починається на 5-у секунду після розміщення запиту на розміщення контейнера, якщо контейнер не було розміщено. Усі значення на графіках поділено на 1000 для репрезентативності.

Кумулятивну ціну обслуговування для хмарного ЦОД показано на рис. 2. Кумулятивна ціна розраховується для суми цін на обслуговування ФС і порушення SLA.

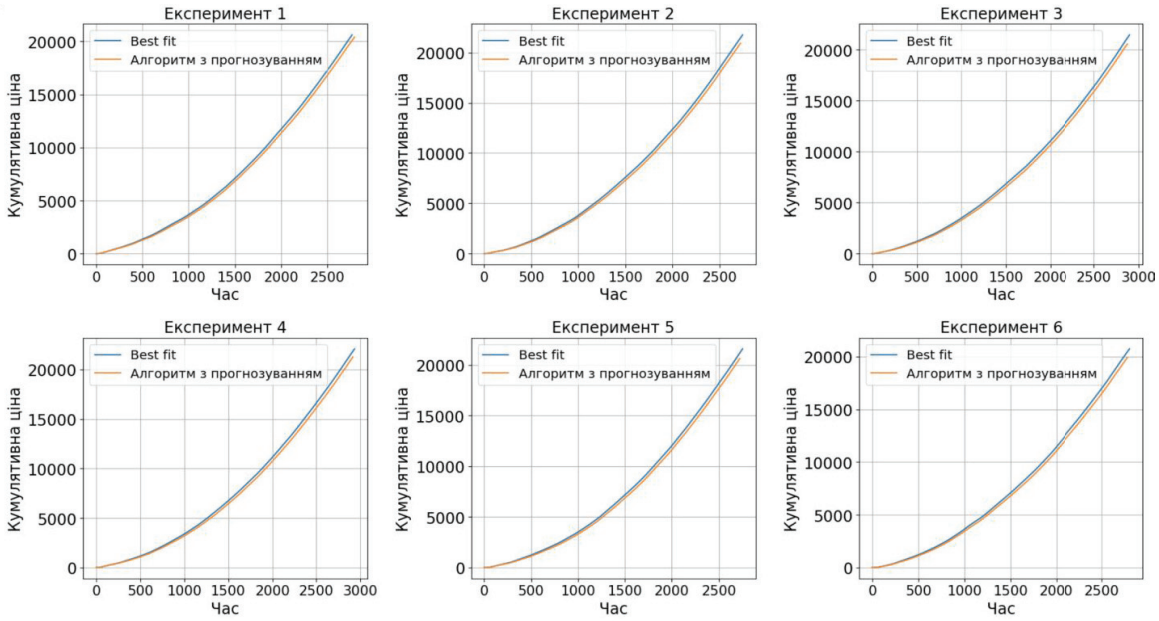

Рис. 2. Кумулятивна ціна для хмарного ЦОД

На графіках видно, що кумулятивна ціна за умови використання алгоритму з прогнозуванням у середньому на 5 % менша, ніж за умови використання алгоритму Best Fit. Це пов’язано зі значним зменшенням сумарної ціни за порушення SLA. Як показано на рис. 3, ціна за порушення SLA в середньому зменшується на 30 %.

Проте через те що ФС і ВМ вмикаються для прогнозованих запитів і деякий час не використовуються, а чекають на запити, ціна на обслуговування ФС зростає в середньому на 1 %. Ціну на обслуговування ФС показано на рис. 4.

Необхідність тримати ФС увімкненим під час використання алгоритму з прогнозуванням збільшує загальну ціну на обслуговування ФС, проте незначно зменшує час проведення експерименту, адже всі згенеровані запити на розміщення оброблюються швидше.

 ISSN 1560-8956

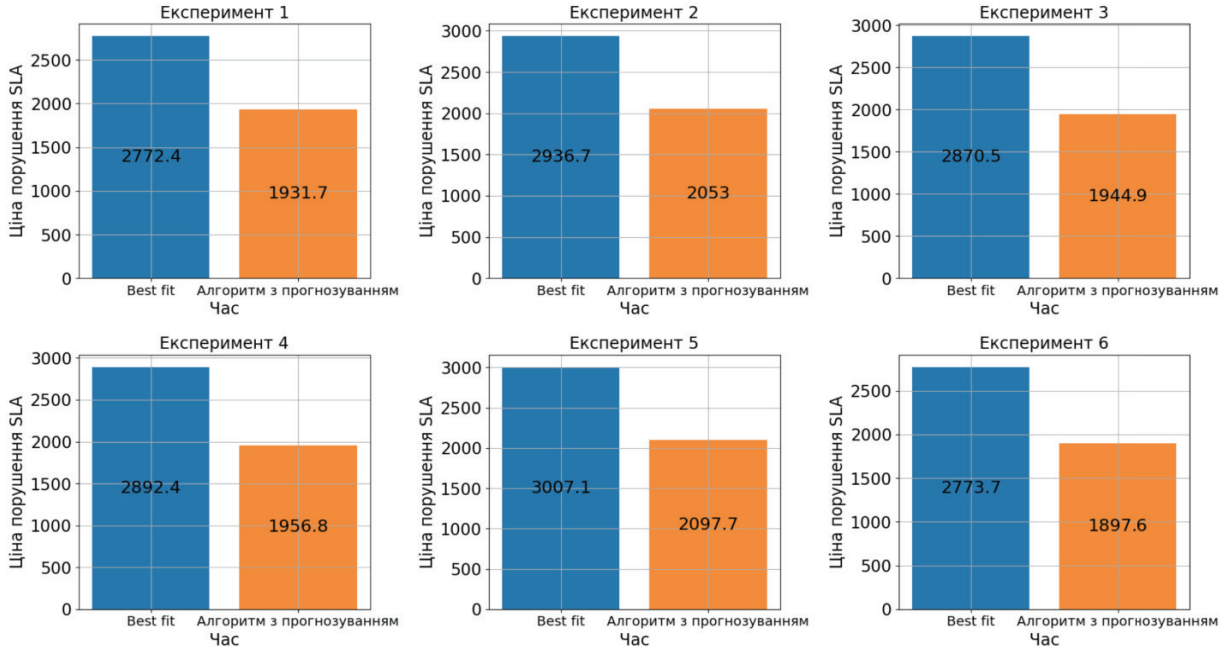

Рис. 3. Ціна порушення SLA

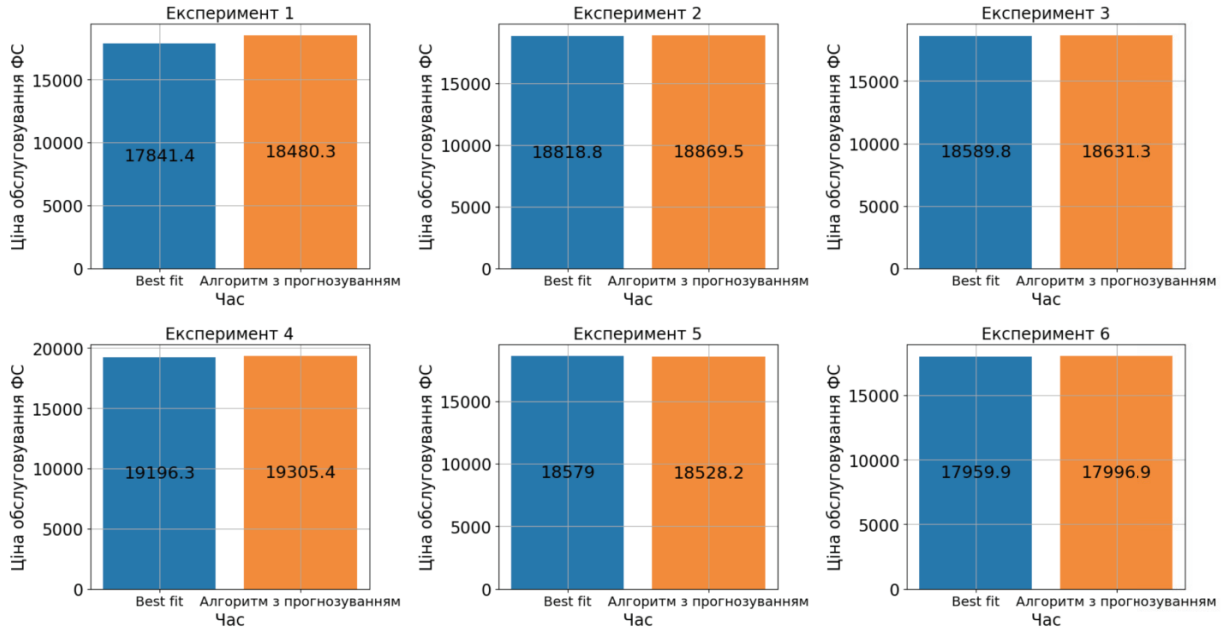

Рис. 4. Ціна обслуговування ФС

Висновки

У цій роботі описано модель, із використанням якої можна проводити експерименти по розміщенню контейнерів у хмарному ЦОД без використання реальних

фізичних ресурсів. Розроблений алгоритм із прогнозуванням навантаження дає можливість зменшити загальну ціну на обслуговування, що включає ціну на обслуговування ФС і ціну порушення SLA в середньому на 5 %, переважно завдяки зменшенню ціни порушення SLA на 30 %.

Алгоритм із прогнозування доцільно використовувати за умови значної, у порівнянні з ціною обслуговування ФС, ціною порушення SLA, адже розроблений алгоритм передбачає ввімкнення ФС і ВМ заздалегідь. У цьому разі хмарний провайдер витрачає ресурси на підтримку серверів у режимі готовності, але зменшує час порушення SLA.

Використання розробленого алгоритму має бути компромісним рішенням для кожного хмарного провайдера, залежно від вартості порушення SLA та вартості обслуговування ФС. Отже, алгоритм із прогнозуванням дозволяє стабільно зменшувати ціну за порушення SLA, проте водночас збільшує ціну на обслуговування ФС.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Telenyk S., Zharikov E., Rolik O. Architecture and conceptual bases of cloud IT infrastructure management. *Advances in intelligent systems and computing*. Cham, 2016. С. 41–62. URL: https://doi.org/10.1007/978-3-319-45991-2_4.

2. Rad B. B., Bhatti H. J., Ahmadi M. An introduction to docker and analysis of its performance. *IJCSNS international journal of computer science and network security*. 2017. Т. 17, № 3. С. 228–235.

3. Docker: accelerated container application development. (2023). URL: https://www.docker.com/.

4. Hussein M. K., Mousa M. H., Alqarni M. A. A placement architecture for a container as a service (CaaS) in a cloud environment. *Journal of Cloud Computing*. 2019. Т. 8, № 1. URL: https://doi.org/10.1186/s13677-019-0131-1.

5. Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities / S. H. H. Madni та ін. *Journal of Network and Computer Applications*. 2016. Т. 68. С. 173–200. URL: https://doi.org/10.1016/j.jnca.2016.04.016 .

6. Beloglazov A., Abawajy J., Buyya R. Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. *Future Generation Computer Systems*. 2012. Т. 28, № 5. С. 755–768. URL: https://doi.org/10.1016/j.future.2011.04.017.

7. 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.

8. Cloud Resource Management with a Hybrid Virtual Machine Consolidation Approach / E. Zharikov та ін. *2019 IEEE International Conference on Advanced Trends in*

Information Theory (ATIT), м. Kyiv, Ukraine, 18–20 груд. 2019 р. 2019. URL: https://doi.org/10.1109/atit49449.2019.9030459.

9. Zhao F., Kansal A., Srikantaiah S. Energy-Aware Consolidation for Cloud Computing. *Cluster Computing*. 2008. Т. 12, № 10-10.

10. Beloglazov A., Buyya R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. *Concurrency and Computation: Practice and Experience*. 2011. Т. 24, № 13. С. 1397–1420. URL: https://doi.org/10.1002/cpe.1867.

11. 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)*, м. Lviv, Ukraine, 5–8 верес. 2017 р. 2017. URL: https://doi.org/10.1109/stc-csit.2017.8098750.

12. Container-VM-PM Architecture: A Novel Architecture for Docker Container Placement / R. Zhang та ін. *Lecture Notes in Computer Science*. Cham, 2018. С. 128–140. URL: https://doi.org/10.1007/978-3-319-94295-7_9.

13. Tan B., Ma H., Mei Y. Novel Genetic Algorithm with Dual Chromosome Representation for Resource Allocation in Container-Based Clouds. *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, м. Milan, Italy, 8–13 лип. 2019 р. 2019. URL: https://doi.org/10.1109/cloud.2019.00078.

14. Alwabel A. A Novel Container Placement Mechanism Based on Whale Optimization Algorithm for CaaS Clouds. *Electronics*. 2023. Т. 12, № 15. С. 3369. URL: https://doi.org/10.3390/electronics12153369.

15. A Genetic Algorithm-Based Energy-Efficient Container Placement Strategy in CaaS / R. Zhang та ін. *IEEE Access*. 2019. Т. 7. С. 121360–121373. URL: https://doi.org/10.1109/access.2019.2937553.

16. 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.

17. J Taylor S., Letham B. Forecasting at scale. URL: https://doi.org/10.7287/peerj.preprints.3190v2.