Сопов Олексій Олександрович

аспірант кафедри ІПІ

Національний технічний університет України

"Київський політехнічний інститут імені Ігоря Сікорського"

ORCID: 0000-0003-0389-3070

**ІНСТРУМЕНТАРІЙ ДЛЯ СИМУЛЯЦІЇ РОЗМІЩЕННЯ КОНТЕЙНЕРІВ У ХМАРНОМУ ЦЕНТРІ ОБРОБКИ ДАНИХ**

Хмарні центри обробки даних (ЦОД) стали важливим компонентом для сучасного інформаційного світу. В останні роки контейнерна віртуалізація отримала великий популярність і стала ефективною альтернативою класичній віртуалізації. Однією з основних переваг контейнерної віртуалізації є легкість контейнерів порівняно з віртуальними машинами. Контейнери дозволяють ізолювати додатки та їх залежності, при цьому вони використовують менше ресурсів та швидше запускаються [1].

Перед хмарним ЦОД постає трирівнева задача розміщення контейнерів: розмістити декілька контейнерів на віртуальній машині, а декілька віртуальних машин на фізичному сервері. Хмарний ЦОД має на меті надати послугу по розміщенню контейнерів із мінімізацією витрат на обслуговування фізичних машин та порушення SLA [2].

У праці [3] автори проводять порівняльний аналіз 33 інструментів для симуляції хмарних середовищ. Цей аналіз дозволяє порівняти відомі симулятори з точки зору підтримуваної моделі, архітектури та функцій високого рівня. У працях [4-6] автори пропонують і інші стимуляційні моделі. Проте, більша частина розглянутих рішень не включає в себе симуляцію розгортання контейнерів у хмарі, або, наявний функціонал погано описано для кінцевого розробника. Тож, інструментарій для розміщення контейнерів у хмарному ЦОД потребує більш широкого представлення у науковому колі, покращення гнучкості та покращення досвіду використання з точки зору розробника, або науковця.

Задачами інструментарію симуляції розміщення контейнеру у хмарному ЦОД є:

* симуляція запитів від користувачів на розміщення контейнерів із заданою ресурсною ємністю у час *t*: $resourceCapacity=\left\{CPU, RAM, DISK, t\right\}$;
* створення симуляційного гетерогенного кластеру ЦОД, що включає в себе різні типи фізичних серверів із різними ресурсними ємностями, використовуючи дані, надані користувачем;
* формування різних типів віртуальних машин, використовуючи дані надані користувачем;
* проведення симуляції розміщення контейнерів;
* логування і візуалізація результатів, що дозволить оцінити результат роботи алгоритмів;
* надати можливість додання нових алгоритмів і модифікації існуючих.

Задля виконання задач запропоновано структуру класів інструментарію для симуляції розміщення контейнерів у хмарному ЦОД, показану на рисунку 1.



Рисунок 1 – Діаграма класів інструментарію для симуляції

Життєвий цикл симулятора полягає у наступному:

* генеруються випадкові запити на розміщення контейнерів із заданою ресурсною ємністю у час t;
* ініціалізується кластер ЦОД із різними заданими типами фізичних серверів;
* ініціалізуються типи віртуальних машин;
* ініціалізуються обʼєкти класів Simulator, ExperimentState, CloudProvider, VMController, VMSelectionStrategy, VMCreationStrategy, PMController, PMSelectionStrategy, PMCreationStrategy
* розпочинається цикл симуляції, допоки не будуть оброблені всі згенеровані запити, із кроком 1 секунда.
* якщо до кластера надійшов запит на симуляцію: викликати метод deploy\_container для CloudProvider.

Логіка методу deploy\_container гнучко налаштовується шляхом, або, перевизначення, або використання інших стратегій вибору/створення віртуальних машин, вибору/створення фізичних машин.

 Клас CloudProvider є основним для симуляції і використовує VMController та PMController для відповідної роботи із віртуальними машинами, або фізичними серверами. Якщо надійшов запит на розміщення контейнера – обрати віртуальну машину і розмістити контейнер на ній за умови достатньої кількості ресурсів. Інакше – створити нову віртуальну машину і розмістити її на одному з увімкнених фізичних серверів із достатньою кількістю ресурсів. Якщо таких фізичних серверів не знайдено – створити новий.

 Симулятор реалізовано мовою програмування Python. Розроблений симулятор дозволяє проводити експерименти по розміщенню контейнерів швидко і якісно, не витрачаючи на це велику кількість грошей, як-от, при тестах в реальних умовах. Використання такого симулятору значно пришвидшить впровадження нових алгоритмів управління, що позитивно впливає на хмарні центри обробки даних.

**Література**

1. Sturm R., Pollard C., Craig J. Managing Containerized Applications. *Application Performance Management (APM) in the Digital Enterprise*. 2017. С. 177–185. URL: <https://doi.org/10.1016/b978-0-12-804018-8.00013-9>.
2. A Framework and Algorithm for Energy Efficient Container Consolidation in Cloud Data Centers / S. F. Piraghaj та ін. *2015 IEEE International Conference on Data Science and Data Intensive Systems (DSDIS)*, м. Sydney, Australia, 11–13 груд. 2015 р. 2015. URL: <https://doi.org/10.1109/dsdis.2015.67>.
3. Mansouri N., Ghafari R., Zade B. M. H. Cloud computing simulators: A comprehensive review. *Simulation Modelling Practice and Theory*. 2020. Т. 104. С. 102144. URL: <https://doi.org/10.1016/j.simpat.2020.102144> (дата звернення: 11.09.2023).
4. A CloudSim-Extension for Simulating Distributed Functions-as-a-Service / H. Jeon та ін. *2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, м. Gold Coast, Australia, 5–7 груд. 2019 р. 2019. URL: <https://doi.org/10.1109/pdcat46702.2019.00076>.
5. Fakhfakh F., Kacem H. H., Kacem A. H. Simulation tools for cloud computing: A survey and comparative study. *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, м. Wuhan, China, 24–26 трав. 2017 р. 2017. URL: <https://doi.org/10.1109/icis.2017.7959997>.
6. Cai Z., Li Q., Li X. ElasticSim: A Toolkit for Simulating Workflows with Cloud Resource Runtime Auto-Scaling and Stochastic Task Execution Times. *Journal of Grid Computing*. 2016. Т. 15, № 2. С. 257–272. URL: https://doi.org/10.1007/s10723-016-9390-y.