

УДК 004.451:004.415.2

## **ВІРТУАЛІЗАЦІЯ ТА КОНТЕЙНЕРИЗАЦІЯ В АРХІТЕКТУРІ ПРОГРАМНИХ СИСТЕМ У ПРОЕКТІ «PROJEKT B»**

Степаненко А.В., Побіженко І.О.

e-mail: artem.stepanenko@nure.ua, iryna.pobizhenko@nure.ua

Харківський національний університет радіоелектроніки, каф. ПІ  
м. Харків, Україна

In today's software systems, the rapid development of technologies requires effective approaches to managing computing resources. Virtualization and containerization have become key elements of software architecture[1], providing flexibility, scalability and optimal use of resources. Virtualization allows you to create independent virtual environments on a single physical server, which significantly reduces infrastructure costs and increases the efficiency of hardware use. It provides process isolation, which increases system security and stability. Thanks to virtualization, you can quickly deploy new servers, balance loads and create backups to ensure business continuity.

У сучасних програмних системах стрімкий розвиток технологій вимагає ефективних підходів до управління обчислювальними ресурсами. Віртуалізація та контейнеризація стали ключовими елементами архітектури програмного забезпечення [1], забезпечуючи гнучкість, масштабованість та оптимальне використання ресурсів. Віртуалізація дозволяє створювати незалежні віртуальні середовища на одному фізичному сервері, що значно знижує витрати на інфраструктуру та підвищує ефективність використання апаратних засобів. Вона забезпечує ізоляцію процесів, що підвищує безпеку та стабільність роботи системи. Завдяки віртуалізації можна швидко розгортати нові сервери, балансувати навантаження та створювати резервні копії для забезпечення безперервності бізнес-процесів. Являючись прикладом, програмний додаток «Project B» використовує зазначені технології для отримання конкретних ефективних покращень із управлінням.

Оптимізація інформаційних систем вимагає нових сучасних рішень [4], що забезпечують гнучкість, ефективність управління ресурсами та спрощують розгортання програмних компонентів. Використання AWS для віртуалізації, контейнеризації та оркестрації дозволяє автоматизувати масштабування, підвищити відмовостійкість системи та оптимізувати використання обчислювальних потужностей

Контейнери віртуалізації – це відносно нова технологія, яка ідеально підходить для додатків «Платформа як послуга» (PaaS) і «Програмне забезпечення як послуга» (SaaS) [1]. Вони забезпечують майже миттєве надання та модифікацію «на льоту», забезпечуючи майже рідну продуктивність програми, тобто накладні витрати на віртуалізацію програми, що працює всередині контейнера, незначні. Контейнери

віртуалізації забезпечують у 10 разів більшу щільність завдань на вузол порівняно з віртуальними. Програмний додаток «Project B» відповідає усім критеріям для того щоб віднести його до PaaS додатків. Для реалізації застосунку із можливістю внесення швидких змін з мінімальними вкладеними ресурсами, слід використовувати контейнеризацію.

Контейнери Linux забезпечили ефективний метод розбиття складних монолітних систем на тонкокомпоновані частини, які виконуються окремо як ізольовані служби [2]. Кожен компонент програми може масштабуватися незалежно. Програма просто стає набором служб, що працюють у контейнерах. Створений на основі контейнерів Linux, Docker згодом зробив революцію в технології контейнерів, покращивши доступність, зручність використання та надавши готові образи контейнерів. Частково завдяки успіху Docker контейнери стали дуже популярними з 2013 року.

Сьогодні контейнерні технології стали невід'ємною частиною розробки та розгортання програмного забезпечення, особливо в хмарних середовищах. Завдяки своїй гнучкості та ефективності контейнери широко використовуються в архітектурі мікросервісів, де кожен сервіс працює незалежно та взаємодіє через стандартизовані API. Розвиток контейнерних технологій продовжується, і сьогодні вони відіграють ключову роль у DevOps, безперервній інтеграції та розгортанні (CI/CD), а також у створенні хмарних та серверлесс-рішень.

За допомогою контейнерів буде розподілено клієнтську частину, серверну частину і базу даних у додатку «Project B». Контейнери дозволяють розділити додаток на незалежні частини із потрібними технологіями для запуску певного функціоналу. Це дозволяє конкретно розподілити бізнес-логіку та реалізує вискоелективне середовище.

Контейнери представляють наступну еволюцію в технології розгортання, пропонуючи легку портативну альтернативу віртуальним машинам[3]. Контейнери спільно використовують ядро хост-ОС і середовище виконання, що дозволяє їм бути більш ефективними та гнучкими, ніж віртуальні машини. На відміну від віртуальних машин, які вимагають окремих екземплярів ОС, контейнери використовують віртуалізацію на рівні ОС для спільного використання основного ядра ОС, що призводить до швидшого часу запуску та менших накладних ресурсів. Контейнери є легковаговими, оскільки всі вони використовують спільне ядро ОС, що зменшує обсяг пам'яті та обчислювальних ресурсів, необхідних для їх роботи.

Завдяки своїй легковаговій природі контейнери дозволяють запускати більше інстансів додатків на одному фізичному сервері, що підвищує ефективність використання ресурсів. Вони також забезпечують уніфіковане середовище виконання, що мінімізує проблеми сумісності між розробницьким, тестовим і продуктивним середовищами.

Контейнеризація надає легкий та ефективний спосіб розгортання програмних компонентів. Контейнери дозволяють запускати додатки у стандартизованих ізольованих середовищах, що забезпечує їхню портативність між різними платформами та серверами. Це особливо важливо для розподілених систем та мікросервісної архітектури, де кожен сервіс може бути незалежно оновлений або масштабований без впливу на інші частини системи.

Віртуальні машини (VM) працюють на рівні гіпервізора, що дозволяє запускати кілька операційних систем на одному фізичному сервері. Вони забезпечують повну ізоляцію середовища, але споживають більше ресурсів через необхідність емуляції апаратного забезпечення. Контейнери, навпаки, працюють на рівні операційної системи та використовують спільне ядро, що робить їх легшими та швидшими у запуску й масштабуванні.

І контейнери, і віртуальні машини відіграють вирішальну роль у розгортанні та виконанні програм. Хоча вони мають певну схожість у своїй здатності ізолювати та інкапсулювати робочі навантаження, контейнери та віртуальні машини значно відрізняються за своєю архітектурою, використанням ресурсів і характеристиками розгортання.

Сучасні інструменти для управління контейнерами, такі як Docker і Kubernetes, значно спрощують процеси розгортання, моніторингу та автоматизації програмних продуктів. Kubernetes, наприклад, забезпечує автоматичне масштабування, балансування навантаження та самовідновлення сервісів у разі їх відмови, що робить його незамінним інструментом для великих розподілених систем.

Окрім цього, Kubernetes дозволяє інтегруватися з хмарними платформами, такими як AWS, Google Cloud і Microsoft Azure, що розширює можливості управління кластерами та покращує гнучкість розгортання. Завдяки цьому контейнери стали ключовою технологією у DevOps-процесах, дозволяючи швидко розгортати, оновлювати та масштабувати застосунки без зупинки їхньої роботи.

#### Список використаних джерел:

1. Білова Т. Г., Побіженко І. О., Ярута В. О. Формування ієрархічної структури показників моніторингу складної інформаційної системи. Системи обробки інформації. 2015. Вип. 12. С. 78-80. URL: [http://nbuv.gov.ua/UJRN/soi\\_2015\\_12\\_19](http://nbuv.gov.ua/UJRN/soi_2015_12_19) (дата звернення: 05.03.2025).
2. Altan Z. Applications and Approaches to Object-oriented Software Design / Z. Altan. – 2019. – pp. 95-110.
3. Grzech A. Information Systems Architecture and Technology: Proceedings of 36th International Conference on Information Systems Architecture and Technology / A. Grzech, J. Świątek, L. Borzemski, Z. Wilimowska. – 2016. – 77 p.
4. Kumar J. System Design on AWS / J. Kumar, M. Singh. – 2025.