УДК 004.75

Л. І. Рудь
О. В. Войцеховська

АНАЛІЗ ТА ПОРІВНЯННЯ ПРОДУКТИВНОСТІ ВІРТУАЛІЗАЦІЇ ТА КОНТЕЙНЕРИЗАЦІЇ ПРИ РОЗГОРТАННІ API НА WEB-СЕРВЕРІ

Вінницький національний технічний університет

Анотація

У даній роботі було розглянуто відмінності у продуктивності технологій віртуалізації та контейнеризації щодо розміщення API на веб-сервері, а саме продуктивності ЦП і фізичної пам'яті, часу відгуку та пропускної здатності. Досліджено дані про використання ресурсів, час відгуку та пропускну здатність під час використання вказаних технологій. Показано в яких випадках краще використовувати одну або іншу технологію.

Ключові слова: контейнеризація, віртуалізація, час відгуку, пропускна здатність.

Abstract

This paper examined performance differences between virtualization and containerization technologies for hosting APIs on a web server, namely CPU and physical memory performance, response time, and bandwidth. The data on the use of resources, response time and bandwidth during the use of the specified technologies were studied. It is shown in what cases it is better to use one or another technology.

Keywords: containerization, virtualization, response time, bandwidth.

Вступ

Контейнеризація та віртуалізація сьогодні є двома основними технологіями хмарних обчислень. Жодна з цих технологій не є новим винаходом, але вони не набули широкого застосування, поки не відновили популярність завдяки новим реалізаціям. Віртуалізація відновила популярність із заснуванням VMWare, а контейнеризація стала надзвичайно популярною за останнє десятиліття з розвитком програмної платформи Docker.

Метою даної роботи є порівняння продуктивності вказаних технологій при їх використанні для розміщення API та використання цими технологіями наданих апаратних ресурсів для обробки запитів HTTP.

Загальне представлення віртуалізації та контейнеризації

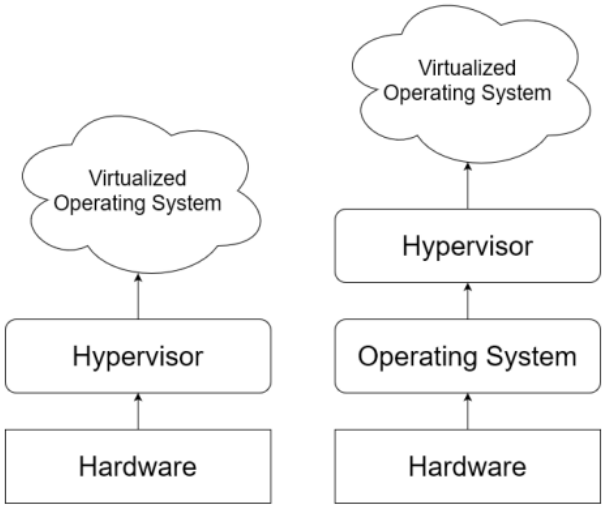
Віртуалізація створює абстрактний рівень над апаратним забезпеченням комп'ютера, який дозволяє використовувати апаратне забезпечення одного комп'ютера та розділяти його на кілька віртуальних комп'ютерів, відомих як віртуальні машини (ВМ), кожна з яких працює під керуванням власної операційної системи. Віртуальні машини можна використовувати, в разі потреби в різних або кількох однакових операційних систем на одному фізичному комп'ютері.

Контейнерізація дозволяє розміщувати програмний код у контейнери, для функціонування яких потрібні лише бібліотеки та залежності операційної системи хоста. Її можна використовувати для розміщення мікросервісів в ізольованому середовищі. Віртуалізація на основі контейнерів стала дуже популярною після появи платформи Docker [1] і є легкою альтернативою більш традиційній віртуалізації на основі гіпервізора.

Гіпервізором (Hypervisor) є програмне рішення, яке керує фізичними ресурсами однієї серверної платформи і може розподіляти їх між кількома ВМ. Він може створити безліч віртуальних машин на базі одного обладнання, розподіляючи його фізичні ресурси залежно від вказаних параметрів.

Використаємо гіпервізор типу 1, також відомий як гіпервізор Bare-metal, для порівняння технологій віртуалізації та контейнеризації. Як показано на рисунку 1, гіпервізор типу 1 розміщує свою віртуальну машину над апаратним забезпеченням, а не над операційною системою (ОС) хоста, як гіпервізор типу 2. Це дає гіпервізорам типу 1 можливість використовувати апаратне забезпечення

ефективніше, ніж гіпервізорам типу 2, які використовують апаратне забезпечення, змодельоване головною операційною системою.



а) б)

Рисунок 1 — Архітектура гіпервізора типу 1 (а) і гіпервізора типу 2 (б)

Архітектура контейнеризації, зображена на рисунку 2, є схожою на гіпервізор типу 2. Значна відмінність між ними полягає в тому, що механізм контейнерів може мати багато контейнерів, розміщених через ядро хост-ОС, тоді як гіпервізор типу 2 повинен створити повну гостьову ОС для кожної віртуальної машини, яку потрібно створити.

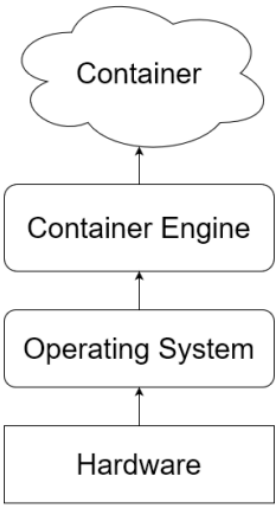


Рисунок 2 — Архітектура контейнеризації

Аналіз результатів дослідження

Відмінності у використанні ресурсів у технологіях віртуалізації та контейнерізації стосуються продуктивності щодо використання ЦП і фізичної пам’яті. Проведено дослідження з метою з’ясування яка з технологій використовує надане обладнання більш ефективно.

Для експерименту з обома технологіями було створено сервер Apache, який розміщував Django REST API, який у випадку успішного доступу відповідав би кодом статусу HTTP 200.

Для контейнеризації використовувалася технологія Docker. Було створено Dockerfile на основі образу Ubuntu 18.04 з усіма необхідними залежностями. Для технології віртуалізації використовувався Hyper-V, який розміщував віртуальну машину сервера Ubuntu 18.04. Середовища були обмежені двома ядрами центрального процесора (ЦП) і чотирма гігабайтами фізичної пам’яті. Випробування проводилися на обох технологіях при 20, 40, 60, 80, 90 і 100 % від максимального навантаження.

Результати вимірювань для обох технологій наведено в таблицях 1 — 2.

Таблиця 1 — Результати тестів для контейнеризації

Docker						
Вимірювання	**20%**	**40%**	**60%**	**80%**	**90%**	**100%**
Кількість запитів, мс	26000	52000	78000	104000	117000	130000
Середній час відповіді, мс	8.16	8.15	8.21	8.42	8.76	8.23
Пропускна здатність, запит/сек	1503.21	1540.11	1538.77	1505.69	1460.23	1544.80
Використання ЦП, %	22.63	24.89	25.46	25.73	24.85	24.91
Використання оперативної пам’яті, %	10.07	14.3	14.97	15.14	15.26	14.16

Таблиця 2 — Результати тестів для віртуалізації

Hyper-V						
Вимірювання	**20%**	**40%**	**60%**	**80%**	**90%**	**100%**
Кількість запитів, мс	26000	52000	78000	104000	117000	130000
Середній час відповіді, мс	5.03	4.97	4.96	4.96	4.93	4.5
Пропускна здатність, запит/сек	1503.21	1540.11	1538.77	1505.69	1460.23	1544.80
Використання ЦП, %	22.63	24.89	25.46	25.73	24.85	24.91
Використання оперативної пам’яті, %	10.07	14.3	14.97	15.14	15.26	14.16

Аналізуючи завантаження ЦП, середній час відповіді і пропускну здатність, можна зробити висновок, що Docker має низьке використання ЦП порівняно з віртуальною машиною, тому що він просто не отримує достатньо запитів, щоб працювати належним чином. Ця гіпотеза в основному базується на дослідженні [2], і полягає в тому, що Docker є кращим вибором, коли мова йде про «сиру» продуктивність ЦП порівняно з Hyper-V, оскільки Docker міг ефективніше використовувати використовуваний ЦП. Проте загальне використання ЦП було нижчим, що пояснено в дослідженні [3], яке показує, що Docker має гіршу мережеву продуктивність, ніж хост-ОС. Оскільки Hyper-V — це чистий гіпервізор, продуктивність мережі має бути такою ж, як у хост-ОС. Ці два дослідження показують, що Docker справді має перевагу в продуктивності процесора, але не має достатнього рівня використання мережі.

Під час розгортання програми, яка використовує HTTP-запити в певній формі, важливими є трафік та дії, які виконує сервер. Якщо вказана програма матиме відносно низький трафік, але вирішуватиме завдання, що вимагають високої продуктивності ЦП, тоді рішення Docker може стати правильним виходом. З іншого боку, якщо очікується, що програма матиме дуже високий трафік, але навантаження на процесор не є особливо важливим, тоді кращим вибором може бути віртуальна машина.

Висновки

У даній роботі було зібрано дані про використання ресурсів, час відгуку та пропускну здатність під час використання технологій віртуалізації та контейнеризації. Дані також порівнювались з відповідними дослідженнями, щоб підтвердити їх. Результати експерименту показали, що віртуалізація має переваги над контейнеризацією в усіх вимірюваних аспектах. Можна зробити висновок, що контейнеризація має вузьке місце у вибраній реалізації, яка перешкоджає продуктивності мережі контейнера, що призводить до того, що контейнер не може обробляти таку кількість запитів HTTP, яку обробляє віртуалізоване середовище.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Docker [Електронний ресурс] — Режим доступу до ресурсу: https://www.docker.com
2. Z. Li, Comparison between common virtualization solutions: Vmware workstation, hyper-v and docker, in 2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC). IEEE, 2021, pp. 701–707.
3. A. Kovács, Comparison of diferent linux containers, in 2017 40th International Conference on Telecommunications and Signal Processing (TSP), 2017, pp. 47– 51.

Рудь Людмила Ігорівна – студентка групи 1КІ-21м, факультет інформаційних технологій та комп'ютерної інженерії, e-mail: liuda.rud498@gmail.com

Войцеховська Олена Валеріївна – кандидат технічних наук, доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, Вінниця.

Rud Liudmila Igorevna – student of the group 1KI-21m, faculty of information technologies and computer engineering, e-mail: liuda.rud498@gmail.com

Voytsekhovska Olena V. — PhD, Assistant Professor of the Computer Techniques Department, Vinnytsia National Technical University, Vinnytsia.