*Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 3 (33)*
*ISSN 2522-9818 (print)*
*ISSN 2524-2296 (online)*

УДК 004.43:004.738.5:004.89 DOI: https://doi.org/10.30837/2522-9818.2025.3.137

О. Свєтлінський, І. Ревенчук

# ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ВІРТУАЛІЗАЦІЙНИХ РІШЕНЬ ДЛЯ РОЗГОРТАННЯ ЦИФРОВИХ ДВІЙНИКІВ У СИСТЕМАХ З ОБМЕЖЕНИМИ РЕСУРСАМИ

**Предметом дослідження** є засоби та підходи до віртуалізації в середовищах з обмеженими ресурсами, зокрема на одноплатних комп'ютерах, а також їх здатність забезпечувати якісну роботу цифрових двійників. **Мета статті** – оцінити ефективність віртуалізації для розгортання цифрових двійників на одноплатних комп'ютерах і розробити рекомендації щодо впровадження віртуалізації в середовищах з обмеженими обчислювальними ресурсами. У статті визначено такі **завдання**: аналіз сучасних апаратних рішень одноплатних комп'ютерів і наявних технологій віртуалізації; побудова експериментального середовища для практичного порівняння технологій віртуалізації на одноплатних комп'ютерах; реалізація прикладного навантаження, що імітує роботу цифрового двійника з реальними сенсорними показниками; дослідження ефективності мережної взаємодії між віртуальними середовищами та хост-системою; порівняння результатів за ключовими метриками. Упроваджено такі **методи**: технології віртуалізації *Docker*, *QEMU*, *Firecracker*; мережний моніторинг. **Досягнуті результати**: проведено порівняльний експеримент з використанням *Docker*, *QEMU* та *Firecracker* на базі *Raspberry Pi 4*; проаналізовано літературу та доступну документацію з окресленого питання; реалізовано базовий прототип цифрового двійника з під'єднанням фізичних сенсорних пристроїв, що збирають телеметричні показники в реальному часі; забезпечено поетапне автоматичне розгортання віртуальних середовищ із попередньо налаштованими компонентами для швидкого масштабування експерименту; проведено вимірювання мережних затримок, часу оброблення повідомлень та використання ресурсів платформи; визначено переваги та недоліки технологій віртуалізації в умовах обмежених обчислювальних ресурсів; оцінено практичну доцільність застосування кожного типу віртуалізації в умовах обмежених обчислювальних ресурсів. **Висновки**: *Docker* забезпечує найпростіше розгортання та найбільшу ефективність, але має меншу ізоляцію, якщо порівнювати з *QEMU* та *Firecracker*; *Firecracker* демонструє оптимальний баланс між продуктивністю, безпекою та споживанням ресурсів на одноплатних комп'ютерах; *QEMU* має найбільші накладні витрати, а тому й найменшу ефективність; використання легковагових гіпервізорів є доцільним у системах цифрових двійників на периферійних пристроях.

**Ключові слова:** віртуалізація; одноплатні комп'ютери; цифровий двійник; мережний моніторинг; *Raspberry Pi*.

## Вступ

Світ цифрових технологій перебуває в постійній трансформації у зв'язку зі стрімким зростанням потреб користувачів, Індустрії 4.0 та поступовим переходом багатьох секторів економіки до автоматизованих рішень. Водночас класичні підходи до розроблення та підтримання програмних систем поступово втрачають свою ефективність. Сучасні проєкти – це цілісні, багаторівневі архітектури з високим рівнем абстракції для її підтримання та управління, що вимагають нових інструментів для моделювання, тестування та масштабування з огляду на наявні проблеми.

Майже завжди, коли постає питання в масштабуванні, одним із найкращих рішень буде використання віртуалізації. Вона давно вийшла за межі хмарних центрів оброблення інформації, і нині все більше проникає на рівень периферійних обчислень, зокрема в пристрої одноплатних комп'ютерів (*SBC, single board computers*), наприклад *Raspberry Pi*. Віртуалізація дає змогу знизити вартість фізичного обладнання, підвищити гнучкість і швидкість розгортання нових вузлів та забезпечити кращу масштабованість систем, що є критично важливим у швидкозмінному середовищі Інтернету речей (*IoT, internet of things*).

Особливої актуальності набуває концепція *Digital Twin* – цифрового двійника, що допомагає створити віртуальне уявлення фізичного об'єкта чи системи, уможливлюючи моделювання, аналіз та оптимізацію без утручання в реальний процес. *Digital Twin* – це відповідь одразу на кілька ключових викликів сучасної інженерії – від зниження вартості розроблення до прискорення циклу тестування та розгортання систем.

Однак застосування віртуалізації в описаній сфері на одноплатних комп'ютерах залишається маловивченим. Обмеження в обчислювальних ресурсах для деяких чи навіть більшості проєктів може стати перепоною у використанні таких комп'ютерів

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)
Innovative technologies and scientific solutions for industries. 2025. No. 3 (33)

для віртуалізації. Отже, актуальною є проблема недостатнього вивчення можливостей створення цифрових двійників на *SBC* із застосуванням сучасних технологій віртуалізації, їх продуктивності, гнучкості та потенціалу масштабованості..

**Аналіз останніх досліджень і публікацій**

Одними з основних функцій пристроїв *IoT* є збирання, оброблення, зберігання та передача даних з фізичного світу у віртуальний. У масивних і автономних середовищах *IoT* виникають проблеми через обмежені ресурси. Для подолання цих проблем необхідна ефективна архітектура, наприклад віртуалізація, що допомагає оптимізувати обмежені ресурси внаслідок розподілу завдань та балансування навантаження. Зі зростанням кількості обсягів інформації особливу увагу потрібно приділяти ефективному управлінню даними: що і як зберігати й передавати, що безпосередньо впливає на оптимізацію ресурсів у великомасштабних системах *IoT* [1].

У дослідженні [2], присвяченому впровадженню віртуалізації на одноплатних комп'ютерах, продемонстровано, що *Raspberry Pi 4 Model B* може успішно працювати як хост віртуалізації разом із технологію віртуалізації *VMware ESXi 7* на *ARM*, незважаючи на обмеження апаратної та програмної сумісності. Хоча платформа поступається традиційним рішенням *x86* щодо продуктивності та швидкості зберігання, вона споживає вдев'ятеро менше енергії, що робить її привабливою для енергоефективних приграничних сценаріїв. Результати підтверджують можливість повної інтеграції такої інфраструктури *ARM64* у наявні корпоративні системи. Інші вчені в роботі [3] оцінили кілька одноплатних комп'ютерів і мікроконтролерів з огляду на під'єднання недорогих датчиків і надання функціональності. Автори розширили обмежені пристрої зовнішніми модулями та адаптерами для стандартизації експериментального середовища. Результати продемонстрували, що платформи *Raspberry Pi 4* та *AN 33 BLE* мають кращу підтримку спільноти, хоча й не найширші електронні можливості. Незважаючи на високе енергоспоживання плат *SBC*, це компенсується більшою інтеграцією функціональності, що робить їх придатними для гнучких рішень *IoT*. Хоча *Raspberry Pi 4B* не є найпотужнішим одноплатним комп'ютером на базі *ARM*, доступним на ринку, він виявився надійною та стабільною платформою, особливо у використанні з гіпервізором *KVM* (*kernel-based virtual machine*). Це було досліджено в статті [4], де один із ключових висновків полягає в тому, що обидві платформи здатні запускати віртуальні машини й мати високу працездатність завдяки застосуванню *KVM*.

Автори праці [5] переконують, що сучасні одноплатні комп'ютери досить продуктивні для запуску типових робочих навантажень. Завдяки своїй низькій вартості та енергоефективності кластери одноплатних комп'ютерів стають реальним інструментом для периферійних обчислень, зокрема рішення на базі *Raspberry Pi* дають змогу створювати недорогі, компактні та масштабовані системи, які можна розгортати поза традиційними центрами оброблення інформації. Автори дослідження [6] використовували *Docker Swarm* на одноплатних комп'ютерах у кластерах і дійшли висновків, що *SBC* забезпечують ефективну платформу для первинного оброблення інформації з метою мінімізації обсягу даних, що будуть надіслані до хмари чи інших серверів.

Щодо потенційної ефективності *MQTT* на *SBC* було проведено дослідження [7], у якому за допомогою розробленої системи моніторингу для сервера *MQTT* на базі брокера *Mosquitto* було проаналізовано продуктивність і характеристики комунікації *MQTT*. Система містила три компоненти: генератор навантаження з 11 *Raspberry Pi*, програма для моніторингу на сервері та вебпанель для візуалізації інформації в реальному часі. Експерименти підтвердили стабільну роботу системи, що свідчить про доцільність використання *Raspberry Pi* в середовищах із високим навантаженням по *MQTT*. Група вчених у межах дослідження [8] довела, що прототип на базі *Raspberry Pi 3* може реалістично виконувати складні робочі навантаження мережі з оброблення інформації з кількох датчиків, інтегруючись із *NMAP*, *MQTT*, *InfluxDB* та *MariaDB*. Система продемонструвала низьку затримку на оброблення показників і загальний відгук. Це є позитивним результатом для використання одноплатних комп'ютерів у контексті цифрового двійника.

У наступному дослідженні [9] увагу зосереджено на застосуванні цифрових двійників в управлінні сільськогосподарськими водними ресурсами, що є яскравим прикладом необхідності розвитку напряму граничних обчислень. Використання *SBC* є корисним не тільки в розподілених інфраструктурах для складних енергоефективних рішень у сільськогосподарському секторі, а й у ширшому контексті граничних

*Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 3 (33)*
*ISSN 2522-9818 (print)*
*ISSN 2524-2296 (online)*

обчислень. Окрім цього, кіберфізичні системи (*CPS, cyber-physical systems*) також отримують переваги від застосування цифрового двійника. Відповідно до дослідження [10] цифрові двійники відіграють ключову роль у розвитку Індустрії 4.0, поєднуючи цифровий та фізичний світи, забезпечуючи динамічне моделювання та інтеграцію інформації реального світу. Це робить технологію надзвичайно актуальним інструментом для сучасних систем *CPS* у промисловості, медицині та міському плануванні].

## Мета роботи й завдання

З поданого вище аналізу можна визначити, що використання віртуалізації на одноплатних комп'ютерах досліджено, але більшість авторів зосереджують увагу на окремих аспектах, таких як запуск однієї віртуальної машини, загальне енергоспоживання або базова продуктивність. Як правило, у поданих роботах не розглядається повне "прикладне навантаження", що відповідало б реальному використанню цифрового двійника з потоками інформації, взаємодією датчиків, моделюванням, запуском декількох віртуальних машин на одній платформі.

Крім того, ефект поступового зниження ефективності роботи *SBC* залежно від запуску додаткових віртуальних машин залишається значною мірою невивченим. Це є критичним фактором у разі, коли один фізичний пристрій має емулювати одночасно декілька цифрових двійників або працювати в багатокомпонентній системі.

У зв'язку з цим виникає потреба в комплексному дослідженні, у межах якого необхідно не лише вивчити технічну можливість запуску віртуальних машин на одноплатних комп'ютерах, а й оцінити їх ефективність в умовах навантажень, наближених до реальних.

Мета дослідження – аналіз ефективності створення цифрових двійників на одноплатних комп'ютерах із використанням віртуалізації, а також формулювання рекомендацій щодо доцільного впровадження цієї технології в умовах обмежених ресурсів.

Отже, для досягнення мети необхідно виконати такі завдання:

– проаналізувати сучасні апаратні рішення одноплатних комп'ютерів щодо підтримки технологій віртуалізації;

– розглянути та порівняти наявні технології віртуалізації, звертаючи увагу на їх вимоги, сумісність і продуктивність;

– дослідити наявні середовища виконання, придатні для віртуалізації на *SBC*;

– визначити критерії оцінювання ефективності віртуалізації потреб цифрових двійників;

– сформулювати й реалізувати прикладне навантаження, що імітуватиме роботу цифрового двійника в типовому сценарії;

– створити експериментальну систему, яка дасть змогу вимірювати продуктивність зі змінною кількістю віртуальних машин;

– розробити рекомендації щодо оптимального впровадження віртуалізації на *SBC* у контексті створення цифрового двійника.

## Матеріали й методи

### *Аналіз апаратного забезпечення*

На ринку існує чимало одноплатних комп'ютерів із різними характеристиками та призначенням. Саме тому особливу увагу необхідно приділити вибору платформи, щоб визначити найбільш ефективну для завдання. Вона має відповідати низці вимог: достатній обсяг оперативної пам'яті, підтримка апаратної віртуалізації, наприклад *ARM Virtualization Extensions*, розширені можливості введення-виведення та низьке енергоспоживання. Розглянемо основні сучасні рішення.

*Raspberry Pi 4 Model B* [11] є однією з найпопулярніших платформ одноплатних комп'ютерів. Вона має чотириядерний процесор, варіації до 8 ГБ оперативної пам'яті, підтримку *USB 3.0, Gigabit Ethernet, microSD* або *USB SSD* для зберігання інформації. Завдяки високій популярності та доступності, наявності безлічі готових образів ОС та підтримці 64-бітної архітектури ця модель стала фактичним стандартом для експериментів.

*Raspberry Pi 5* [12] – наступна ітерація в лінійці *Raspberry Pi*, має варіації до 16 ГБ оперативної пам'яті, підтримку *PCIe 2.0*. Ця модель забезпечує потенційну продуктивність удвічі та більше разів вищу, ніж *Pi 4* в реальних завданнях, зокрема у віртуалізації. З першого погляду це робить її набагато привабливішою для сценаріїв дослідження *Digital Twin*, особливо коли на одному пристрої планується запускати кілька віртуальних машин. Однак важливо зважати на деякі технічні особливості, наприклад необхідність активного охолодження. Процесор *Pi 5* має більш високу теплову проєктну потужність (*TDP, thermal design power*), і без вентилятора або великого радіатора він починає тротліти (*throttle*) навіть в умовах середнього

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)
Innovative technologies and scientific solutions for industries. 2025. No. 3 (33)

навантаження. Отже, для стабільної роботи в завданнях віртуалізації охолодження є не рекомендацією, а вимогою. Крім цього, навіть якщо *Raspberry Pi 5* вже декілька років на ринку, *Raspberry Pi 4* має більш стабільну екосистему, оскільки вона підтримується з 2019 року. Звичайно, зі збільшеною потужністю висуваються й підвищені вимоги до живлення. *Pi 4* є більш енергоефективним, оскільки використовує блок живлення 5В/3А порівняно з 5В/5А у *Pi 5*.

*Radxa Rock 5B* [13] – потужний одноплатний комп'ютер на базі восьмиядерного процесора, що підтримує до 32 ГБ оперативної пам'яті, має вбудований *eMMC*, слот для *NVMe SSD, HDMI 2.1*. Отже, *Rock 5B* є високопродуктивною альтернативою лінійці *Raspberry Pi* для завдань ШІ, віртуалізації, оброблення медіа, або навіть ігор, але ціна значно вища за інші варіанти.

*Libre Computer Board AML-S905X-CC* [14] – бюджетна альтернатива з меншими можливостями. Має чотириядерний процесор, 2 ГБ оперативної пам'яті, є дешевшим варіантом, якщо порівнювати з *Pi 4* та *Pi 5*, але віртуалізація тут можлива лише в базовому вигляді.

*Orange Pi 5* [15] – більш потужний аналог *Raspberry Pi 5*, але від іншого виробника. Наявна підтримка до 16 ГБ оперативної пам'яті, *eMMC, M.2 NVMe SSD* з восьмиядерним процесором. Цю модель можна ефективно використовувати в периферійних проєктах, бо також підтримує віртуалізацію на апаратному рівні. *Orange Pi 5* має значно вищу ціну, ніж *Raspberry Pi*, і за обчислювальними можливостями наближена до *Radxa Rock 5B*.

Порівняння потенційних платформ для проведення експерименту подано в табл. 1.

**Таблиця 1.** *Характеристики одноплатних комп'ютерів*

| Платформа | Процесор | Оперативна пам'ять | Підтримка апаратної віртуалізації | Живлення | Охолодження | Орієнтовна ціна |
|---|---|---|---|---|---|---|
| **Raspberry Pi 4 Model B** | 4 ядра | до 8 ГБ, LPDDR4 | так | 5V/3A | пасивне, опціонально активне | $35–75 |
| **Raspberry Pi 5** | 4 ядра | до 16 ГБ, LPDDR4X | так | 5V/5A | обов'язково активне | $50–120 |
| **Radxa Rock 5B** | 8 ядер | до 32 ГБ, LPDDR4X | так | 5V/5A | обов'язково активне | $130–270 |
| **Orange Pi 5** | 8 ядер | до 16 ГБ, LPDDR4X | так | 5V/4A | рекомендовано активне | $85–130 |
| **Libre AML -S905X-CC** | 4 ядра | до 2 ГБ, DDR3 | ні | 5V/2A | пасивне | $20–30 |

Унаслідок аналізу платформ обрано модель *Raspberry Pi 4 Model B* з 4 ГБ оперативної пам'яті. Цей вибір зумовлений низкою факторів. Модель з 4 ГБ оперативної пам'яті забезпечує достатньо ресурсів для запуску кількох віртуалізованих середовищ, зберігаючи водночас помірну вартість. Окрім цього, ця модель доступна на ринку з 2019 року, що дало змогу розробникам досягти стабільного середовища підтримки як у спільноті користувачів, так і серед дистрибутивів операційних систем, драйверів та інструментів віртуалізації. На відміну від *Raspberry Pi 5*, четверта модель може працювати тривалий час навіть за умови пасивного охолодження. І хоча *Pi 4* поступається *Pi 5* за обчислювальною потужністю, у завданнях, де основне навантаження – мережний обмін, зберігання інформації, взаємодія з датчиками, продуктивність *Pi 4* залишається більш ніж достатньою, а завдяки своїй низькій вартості та широкій доступності *Pi 4* допомагає легко масштабувати експерименти або розгортати аналогічні системи в інших середовищах.

### *Аналіз середовищ виконання для віртуалізованих систем*

У контексті створення цифрових двійників на базі *Raspberry Pi* та вивчення віртуалізації ключовим моментом є вибір відповідного віртуалізованого середовища, що самостійно запускатиметься всередині віртуальних машин чи контейнерів. Основні вимоги до такого середовища:

– мінімальне споживання ресурсів;

– можливість автоматизації та налаштування без графічного інтерфейсу;

– наявність мережних інструментів і базових сервісів, зокрема *MQTT, Node-RED* тощо;

– сумісність із архітектурою *ARM*.

Отже, маємо критерії щодо вибору найкращого середовища для віртуалізованих систем.

ISSN 2522-9818 (print)
*Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 3 (33)* ISSN 2524-2296 (online)

Зауважимо, що використовувати *Windows* для домашніх комп'ютерів на *SBC* не є вигідною альтернативою з погляду ефективності. І хоча існує *Windows 10/11 IoT Core* для *ARM*, це все ж таки дуже обмежена версія зі своєю специфікою та не найпрозорішою підтримкою технологій віртуалізації. Емуляція ж середовища, що запустило б дистрибутиви, робить практичне тестування продуктивності платформи неможливим.

*MacOS* – це закрита платформа, що за межами офіційного обладнання від *Apple* не підтримує жодних версій *ARM* та технічно неможлива для легального використання чи досліджень на *Raspberry Pi*. Отже, єдиною альтернативою, що можна раціонально розглядати, є застосування дистрибутивів *Linux*.

Розглянемо деякі доступні варіанти віртуалізованого середовища на *ARM*.

*Debian* – класичний вибір дистрибутиву *Linux*. Він стабільний, з багатою підтримкою архітектури *ARM*, має чи не найбільший репозиторій пакетів, але займає багато місця на диску між інших дистрибутивів. Загальне використання – системи, де важливі стабільність і стандартизація, що можуть собі дозволити такий обсяг дистрибутиву.

*Ubuntu Server* – похідний дистрибутив від *Debian*, що також має активну спільноту, надмірну документацію та офіційну підтримку *ARM*, проте, на відміну від *Debian*, містить більшу кількість задач, що працюють на фоні, та потребує більш потужну платформу, що не позитивно впливатиме на ефективність роботи віртуалізованих середовищ.

*Arch Linux ARM* – значно легкий і гнучкий, на відміну від попередніх дистрибутивів, але вимагає чимало часу на налаштування та має недолік з нестабільністю бази пакетів, а це може дуже негативно позначитися на автоматизованому розгортанні віртуальних машин.

*Alpine Linux* – дистрибутив, що відрізняється від інших вкрай малим розміром образу, дуже низьким споживанням оперативної пам'яті та стабільною підтримкою архітектури *ARM64*. Він не має недоліків з автоматизованим розгортанням, як *Arch Linux*, і має суттєвий репозиторій пакетів, що робить налагодження практичного навантаження дуже простим. Цей дистрибутив є найбільш підхожий варіант для завдань дослідження.

Завдяки окресленим характеристикам *Alpine Linux* обраний як основне середовище виконання у віртуалізованих системах, що дає змогу зосередитися на архітектурі цифрового двійника, не витрачаючи ресурсів на непотрібну інфраструктуру.

### *Аналіз технологій віртуалізації*

Вибір технології віртуалізації – один із найважливіших аспектів у дослідженні продуктивності віртуалізації на *SBC*. Він визначає, наскільки ефективно можна реалізувати ізольовані середовища для моделювання, аналізу й тестування систем. Однак на платформах *SBC* цей вибір ускладнений обмеженнями апаратної архітектури *ARM64*, нестачею ресурсів, а також частковою або експериментальною підтримкою деяких інструментів.

*Docker* [16] на платформах *ARM*, зокрема *Raspberry Pi*, посідає особливе місце завдяки своїй легкості, швидкому запуску контейнерів та відносній простоті використання. Ця технологія не створює повноцінних віртуальних машин, а замість цього застосовує так звані контейнери (*Docker containers*), у цьому разі процеси користувача поділяються на рівні ядра. Це означає високу продуктивність та низьке споживання пам'яті, але знижений рівень ізоляції, що спричиняє потенційні проблеми з безпекою.

*QEMU* дає змогу створювати й запускати повноцінні віртуальні машини, що є ізольованими операційними системами, які мають власне ядро та диск. Емулятор має відкриту кодову базу, що дає змогу додавати, редагувати та вилучати функціонал за побажанням користувача [17]. У разі, коли хост-система (*host system*) підтримує апаратну віртуалізацію, *KVM* допомагає значно прискорити *QEMU* внаслідок використання гіпервізора (*hypervisor*) на рівні ядра. У поєднанні з *KVM* віртуальні машини потенційно можуть майже не відрізнятися від реальних машин із виділеним апаратним забезпеченням.

*Firecracker* [18] – гіпервізор, що створює так звані мікровіртуальні машини (*micro VM*). Він розроблений з огляду на безсерверні обчислення (*serverless computing*) та забезпечує надзвичайно швидкий запуск із мінімальним використанням ресурсів. Хоча підтримка *ARM64* в *Firecracker* поки не так розвинена, як *×86_64*, вона вже дає змогу створювати ізольовані середовища, що можуть запускатися дуже швидко. Це робить її перспективною альтернативою для систем, які потребують масштабування в реальному часі, наприклад для оброблення подій датчиків на великій кількості паралельних цифрових двійників.

Крім описаних вище технологій, існують інші, що з певних причин менш ефективні для використання

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2025. No. 3 (33)

на *Raspberry Pi*. Наприклад, *KubeVirt* запускає віртуальні машини в середовищі *Kubernetes*, але потребує складної інфраструктури. *KubeVirt* орієнтується саме на хмарні обчислення. *Xen* – відносно легкий гіпервізор – активно застосовувався в системах, що вбудовувалися в минулому, але тепер менш поширений на платформах з архітектурою *ARM64*.

Тому вибір технології віртуалізації для *Digital Twin* на *SBC* залежить від компромісу між продуктивністю, ізоляцією, швидкістю запуску, сумісністю та масштабованістю. *Docker* надає легке контейнерне середовище, тоді як *QEMU* з *KVM* забезпечує повну віртуалізацію. *Firecracker*, також з використанням *KVM*, незважаючи на свою експериментальність на *ARM64*, відкриває перспективу мікровіртуальної архітектури для цифрових двійників. Разом ці інструменти дають змогу обрати рішення відповідно до конкретного завдання, що стане предметом подальших досліджень.

### *Аналіз ефективності віртуалізації*

Аналіз рівня ефективності різних технологій віртуалізації на *Raspberry Pi*, що є обчислювально обмеженою платформою, вимагає використання чітко визначених і відтворюваних метрик для отримання точних експериментальних показників. У дослідженні обрано низку ключових критеріїв, що дають змогу кількісно оцінити як продуктивність самої віртуалізованої системи, так і ступінь навантаження на апаратне забезпечення.

Час запуску (*startup time*) – це період від запуску ініціалізації віртуального середовища до повної робочої готовності, що передбачає запуск операційної системи, служб усередині віртуального середовища, виклик користувацьких скриптів. Цей критерій є критично важливим, коли віртуальні середовища мають запускатися динамічно, наприклад, у процесі масштабування або у відповідь на зовнішні події. У разі цифрового двійника це дасть змогу миттєво додавати нові сенсори чи їх групи, не потерпаючи від тривалого запуску середовища.

Використання оперативної пам’яті – ступінь застосування оперативної пам’яті хост-системи віртуальними середовищами. Знаючи максимальний обсяг пам’яті, критерій допомагає оцінити, скільки середовищ може бути запущено одночасно на обмеженій платформі без перевантаження пам’яті, що негативно вплине на швидкодію всієї системи. Хоча більшість гіпервізорів дають змогу та рекомендують виставляти незмінні обмеження на пам’ять, критерій може бути корисним для знаходження та аналізу витоків пам’яті у процесі застосування віртуалізованих середовищ.

Час проходження *ping*-пакета (*RTT, round trip time*) – це середній час, необхідний для надсилання *ICMP*-пакета та отримання відповіді. Критерій допомагає оцінити базову якість мережної взаємодії віртуального середовища, зокрема затримки, які могли бути спричинені нагромадженням віртуальними мережними інтерфейсами або додатковою маршрутизацією. Це важливо для цифрових двійників, яким необхідно отримувати та передавати інформацію в режимі реального часу.

*RTT* для повідомлень *MQTT* – час від надсилання повідомлення *MQTT* до моменту, коли копія повертається відправнику назад. На відміну він часу проходження *ping*-пакета, цей критерій відтворює не тільки затримку мережі, але й здатність віртуального середовища швидко обробляти події в протоколі, що часто використовується для Інтернету речей. Це особливо важливо в системах цифрових двійників у процесі передачі телеметрії.

Затримка доступу до диска (*disk delay*) – це час виконання типових операцій читання та запису у віртуальне файлове сховище незалежно від архітектури віртуального середовища. Продуктивність диска суттєво впливає на системи, що потребують роботи з базами даних. Продуктивність диска суттєво впливає на ефективність "цифрових двійників", через їх активну роботу з базами даних.

Використання центрального процесора – середній рівень його застосування під час виконання типових завдань чи в режимі очікування. Він показує, наскільки ефективно гіпервізор розподіляє ресурси й чи не витрачає він занадто багато обчислювальної потужності на існування. Простий критерій, який може швидко, але не дуже точно, визначити ступінь результативності певної технології віртуалізації.

Температура центрального процесора – це температура чипа під час роботи системи. Як і з використанням центрального процесора – це простий, але показовий критерій, що може допомогти звернути увагу на потенційні проблеми ефективності роботи цього елемента. Питання охолодження не є суттєвим, зокрема для *Raspberry Pi 4*, але загалом *SBC* можуть значно перегріватися, що допомагає відслідковувати цей критерій.

*Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 3 (33)*
*ISSN 2522-9818 (print)*
*ISSN 2524-2296 (online)*

Кожен із наведених вище критеріїв обраний з огляду на особливості реального застосування цифрових двійників, тобто систем, які мають бути автономними, реагувати в реальному часі, забезпечувати ізоляцію, але водночас легко масштабуватися та, якщо можливо, не сильно завантажувати загальну систему. Комплексне використання цих метрик дає змогу об'єктивно порівнювати різні підходи до віртуалізації та формувати рекомендації щодо їх практичного впровадження.

## Проведення експерименту

Для порівняльного аналізу ефективності віртуалізації з використанням різних технологій віртуалізації в контексті цифрових двійників на базі одноплатного комп'ютера необхідно було реалізувати експериментальне середовище, що передбачає як підготовку апаратного забезпечення, так і програмну автоматизацію проведення дослідження з додатковим первинним аналізом результатів.

Як було зазначено вище, обрана платформа – це *Raspberry Pi 4 Model B* з 4 ГБ оперативної пам'яті. На *SD*-карту обсягом 32 ГБ був записаний 64-бітний дистрибутив *Debian* для архітектури *ARM*, попередньо налаштований для використання *KVM*. Без увімкнення функціоналу *KVM* всі вимірювання будуть заздалегідь далекі від оптимальних значень.

До платформи було під'єднано зовнішні сенсори, а саме п'ять пристроїв: *GY-302*, *GY-87*, *DHT22*, *MQ-7* та мікросхема АЦП *MCP3008*. Загалом комп'ютер мав доступ до шести сенсорів різної направленості: сенсор освітленості *BH1750FVI*, гіроскоп *MPU6050*, компас *QMC5883L*, сенсор атмосферного тиску *BMP180*, сенсор вологості та температури *DHT22*, датчик газу *MQ-7*. Для постійного зчитування значень сенсорів та відправлення цих показників за допомогою *MQTT* було розроблено скрипт мовою програмування *Python*, що збирає телеметричні показники з різною для кожного сенсора періодичністю.

Зв'язок із датчиками найчастіше виконувався через шину *I2C*, деякі використовували спеціальні протоколи через *GPIO* та частина сенсорів була під'єднана до шини *SPI*.

Для кожної з досліджуваних технологій віртуалізації, а саме *QEMU*, *Firecracker*, *Docker*, створено окремі середовища, максимально схожі одне на одне. Були спроби скопіювати середовище *QEMU* для роботи на *Firecracker*, але це виявилося нетривіальним і виходить за межі дослідження. У кожному з випадків образ містив дистрибутив *Alpine Linux* з мінімальним набором системних утиліт, що потім за потреби розширювалися. Для всіх середовищ був обмежений розмір диска до 1 ГБ, а обсяг оперативної пам'яті становив максимум до 256 МБ. Що менше місця займає образ без втрат у швидкості, то більше паралельно працюючих віртуальних машин може бути.

Як практичне навантаження було реалізовано просту цифрову подвійну комунікацію: значення, надходячи з фізичних пристроїв, опиняються в межах хост-истеми та надсилаються віртуальною локальною мережею через протокол *MQTT* усім запущеним віртуальним машинам чи контейнерам. У віртуальному середовищі за допомогою технології *Node-RED* такі повідомлення обробляються та зберігаються в базі даних тимчасових рядів *InfluxDB*. Ця група технологій імітує оброблення телеметричної інформації пристроїв *IoT* у віртуальному контексті, щоб отримати експериментальні показники, максимально наближені до реальних.

Для мережної взаємодії між віртуальними середовищами та хост-системою використано два підходи: інтерфейси *TAP* для *QEMU* та *Firecracker*, і мережний міст у разі *Docker*. Кожній віртуальній машині призначався окремий інтерфейс, що забезпечує пряме під'єднання до мережного простору хост-системи, даючи змогу обмінюватися повідомленнями *MQTT*, *ICMP*-пакетами та збирати метрики затримки.

Для того, щоб запобігти людській помилці та неточності у виконанні випробування, розроблено групу скриптів *Bash*, які могли бути запущені експериментатором одноразово, і система самостійно виконувала повний цикл:

– початок роботи скрипта *Python* для отримання показників із сенсорів;

– запуск ініціалізації набору віртуальних середовищ;

– старт сервісів *Node-RED* та *InfluxDB* у віртуалізованих середовищах;

– збір системних метрик за описаними вище критеріями;

– зберігання отриманої інформації у файлах формату *.csv*;

– завершення роботи скриптів і віртуалізованих середовищ.

Кожен експеримент проводився кілька разів з метою досягнення статистично стабільних результатів.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)
Innovative technologies and scientific solutions for industries. 2025. No. 3 (33)

## Результати дослідження

Для проведення експерименту застосовано *Raspberry Pi 4 Model B*, переваги якого вже описані в розділі про аналіз апаратного забезпечення. За допомогою з'єднувальних дротів було під'єднано всі потрібні пристрої, що використовують шини *I2C, SPI* та *GPIO*. Як носій пам'яті обрано карту *MediaRange MicroSDHC Class 10* на 32 ГБ. Вона не змінювалася між експериментами над різними технологіями віртуалізації, тому будь-які недоліки карти пам'яті позначилися на всіх експериментальних показниках, що не заважає їх порівнювати в межах дослідження. Сама програмна база операційної системи *Pi 4* також залишалась незмінною, тому отримана інформація максимально відповідає вимогам порівняльного аналізу між обраними технологіями. Результат установки зображено на рис. 1. Ліворуч розташований *Raspberry Pi 4*, що видно із символу малини на друкованій платі, а праворуч, на макетній платі, подано всі периферійні пристрої.

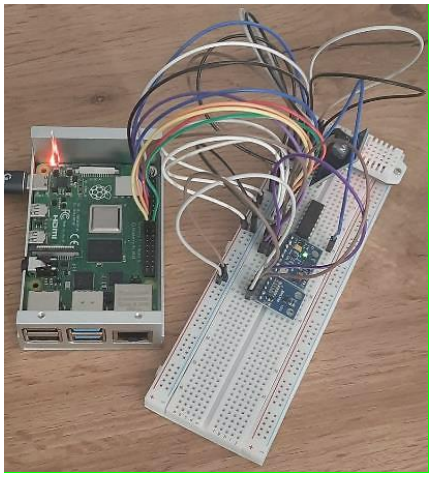

**Рис. 1.** Експериментальна апаратна установка з *Raspberry Pi 4* та пристроями *GY-302, GY-87, DHT22, MQ-7, MCP3008*

Загалом проведено чотири запуски експерименту для кожної з технології віртуалізації. Показники були усереднені з метою запобігання отримання некоректних значень або впливу на експеримент зовнішніх факторів. Підготовлено вісім віртуальних середовищ, але під час проведення експериментів платформа показувала сильну нестабільність, екстремальне погіршення продуктивності та надвисокі показники температури процесора в умовах восьми одночасно запущених віртуальних машин *QEMU*. З огляду на це було прийнято рішення обійтися сімома середовищами та порівнювати інформацію саме з них, щоб залишати *Pi 4* працездатним.

У межах усіх експериментів для кожного почергового запуску віртуального середовища виміряно час від запуску ініціалізації середовища до виконання скрипта користувача із середини запущеного середовища. Час вимірювався способом збереження часової мітки у двох випадках, а повідомлення до хост-системи було реалізовано за допомогою *ping*-пакета, надісланого віртуалізованим середовищем (див. табл. 2).

**Таблиця 2.** *Час почергового запуску віртуальних середовищ*

| Кількість запущених віртуальних середовищ | Час запуску Docker (с) | Час запуску Firecracker (с) | Час запуску QEMU (с) |
|---|---|---|---|
| **1** | 0.761 | 3.176 | 11.944 |
| **2** | 0.84 | 3.246 | 11.786 |
| **3** | 0.94 | 3.34 | 12.313 |
| **4** | 1.088 | 3.455 | 16.731 |
| **5** | 1.507 | 3.652 | 20.784 |
| **6** | 1.642 | 3.733 | 27.01 |
| **7** | 1.779 | 3.948 | 44.393 |

З отриманих експериментальних показників можемо спостерігати вагому різницю між кожною технологією. Поки *Docker* і *Firecracker* запускаються впродовж декількох секунд чи навіть швидше, *QEMU* відносно тривалий час проходить ініціалізацію. Частково це можна пояснити рівнем емуляції та безпеки, що забезпечує *QEMU*, втрачаючи здатність швидко розгортати нові віртуальні машини.

Окрім цього, помітна загальна тенденція в надлишковому часі запуску в кожній технології приблизно після четвертого запущеного середовища. Імовірно, це пов'язано з тим, що платформа – це чотириядерний процесор.

Щоб зменшити короткострокові коливання, які були сильно помітні під час первинного огляду, покращити їх візуальну інтерпретацію та сформулювати висновки про загальні тенденції, для всіх наступних експериментальних показників застосовано їх згладжування за допомогою процесу експоненціального рухомого середнього (*exponential moving average*). Водночас було збережено часову динаміку:

$$EMA_t = \alpha \cdot x_t + (1-\alpha) \cdot EMA_{t-1}, \quad (1)$$

де $EMA_t$ – експоненціальна рухома середня в момент часу $t$;

$\alpha$ – коефіцієнт згладжування;

*Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 3 (33)*
*ISSN 2522-9818 (print)*
*ISSN 2524-2296 (online)*

$x_t$ – поточний елемент показників;

$EMA_{t-1}$ – попередня експоненціальна рухома середня.

Експериментально визначено оптимальне значення коефіцієнта згладжування. У всіх лінійних графіках воно дорівнює приблизно 0.18.

Обрана модель платформи мала 4 ГБ оперативної пам'яті, що спочатку здавалося дуже недостатньо для такого типу експерименту, але *Firecracker* та *QEMU* досить ефективно виконували завдання цифрового двійника з виділеними 256 МБ оперативної пам'яті на кожну з машин (див. рис. 2).

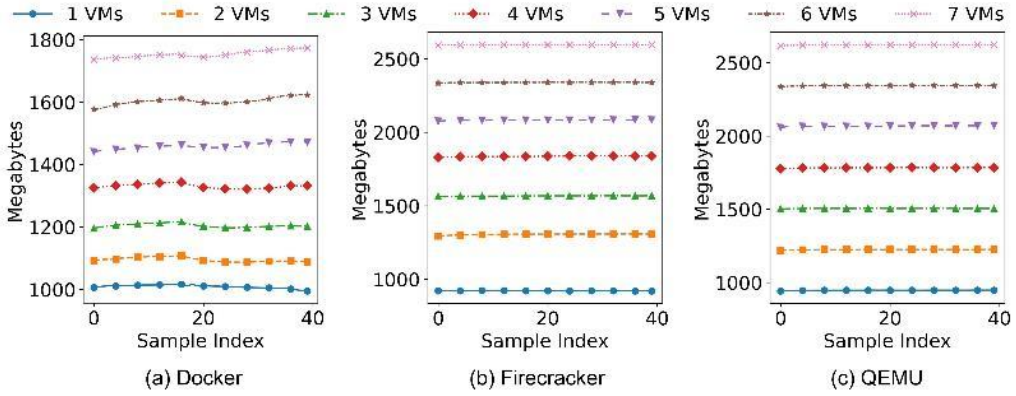


**Рис. 2.** Порівняння використання оперативної пам'яті під час почергового запуску віртуальних середовищ

Отже, спостерігаємо паритет між двома технологіями, а *Docker* зі свого боку динамічно підвищував використання пам'яті від навантаження, що виявилося суттєво менше за інші варіанти. Приблизно 100 МБ за контейнер, тому теоретично можна було б запустити до 30 контейнерів або 12 віртуальних машин, зважаючи тільки на оперативну пам'ять.

Наступним критерієм, за яким проведено експеримент, є час проходження *ping*-пакета (див. рис. 3).

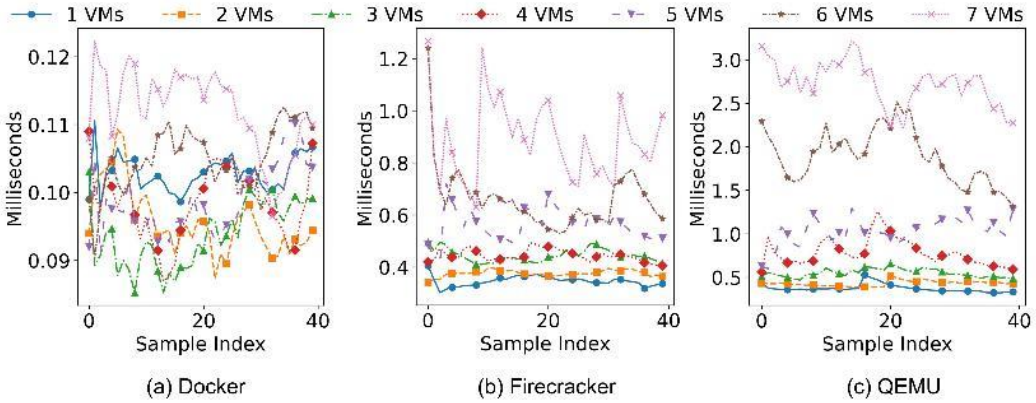


**Рис. 3.** Порівняння середнього часу проходження *ping*-пакета внутрішньою мережею між хост-системою та віртуалізованими середовищами

Вимірювання проведено в масштабі мікросекунд, але перетворені в мілісекунди для кращої візуалізації. *Docker* продемонстрував майже незалежність часу проходження пакета від кількості запущених контейнерів. Завжди значення коливаються приблизно зі швидкістю 0.1 мілісекунда, що є найкращим показником серед альтернатив. *Firecracker* та *QEMU* мали гірші результати за цим критерієм за умови 0.6 мілісекунди та 1.5 мілісекунди в середньому відповідно до альтернатив. Стабільне та наднизькі значення проходження *ping*-пакета *Docker* вказує на мінімальні мережні витрати, типові для контейнеризації, яка не потребує повної мережної віртуалізації. У *Firecracker* і *QEMU* додаткові віртуальні машини спричиняють затримки, що зростають. Це вказує на вищу вартість мережної ізоляції.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)
Innovative technologies and scientific solutions for industries. 2025. No. 3 (33)

Більш затратною за ресурсами операцією є проходження *MQTT*-повідомлення, що також дає змогу оцінити стан мережі в процесі віртуалізації, але в іншому контексті. Було застосовано схожі інструменти вимірювання часу між цим і попереднім критерієм через їх функціональну схожість. Досягнуті результати дещо віддзеркалюють показники від *ping*-пакетів, але додають новий погляд на ефективність віртуалізації (див. рис. 4).

З отриманих показників видно, що *Docker* демонструє кращу продуктивність і стабільність, що повторює результати *ping*-пакетів. *Firecracker* повільніше обробляє повідомлення, і кожне почергове середовище сильніше погіршує швидкість оброблення, але все ж таки залишається досить стабільним, у межах 100–200 мілісекунд. Це відповідає пріоритету технології – забезпечити ізоляцію, подібну до віртуальної машини, з продуктивністю, подібною до контейнера. *QEMU* сильно сповільнюється за умови шести-семи запущених машин, але в разі чотирьох машин тримається майже стабільно, як і інші альтернативи. Знову ж таки, це може бути пов'язано з чотириядерним процесором платформи. Результати *QEMU* вказують на значні витрати на віртуалізацію мережі, пов'язану з емуляцією мережного інтерфейсу.

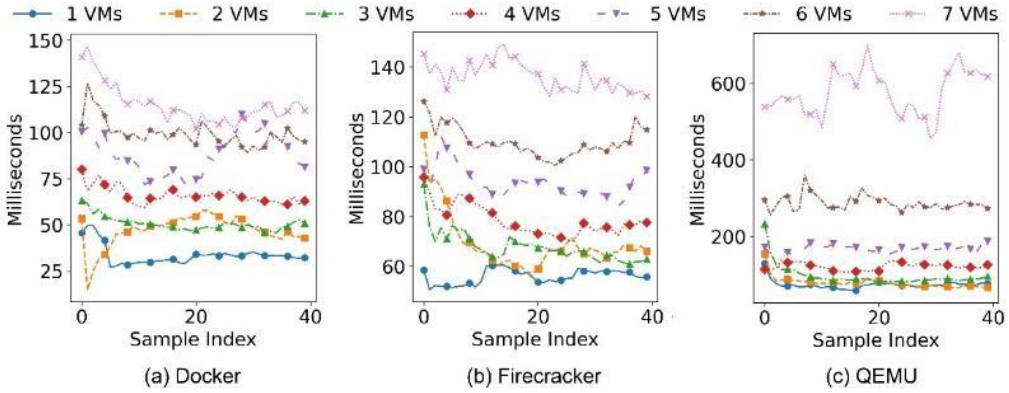


**Рис. 4.** Порівняння середнього часу проходження *MQTT*-повідомлення від хост-системи до кожного з віртуалізованих середовищ і у зворотному напрямку

Затримка доступу до диска відіграє важливу роль у порівнянні технологій віртуалізації. У контексті цифрового двійника необхідно завжди зберігати отримані показники, вчасно оновлювати налаштування. За умови значних навантажень це може суттєво вплинути на швидкодію окремого цифрового двійника, тому отримані показники відіграють важливу роль у загальному порівнянні технологій (див. рис. 5).

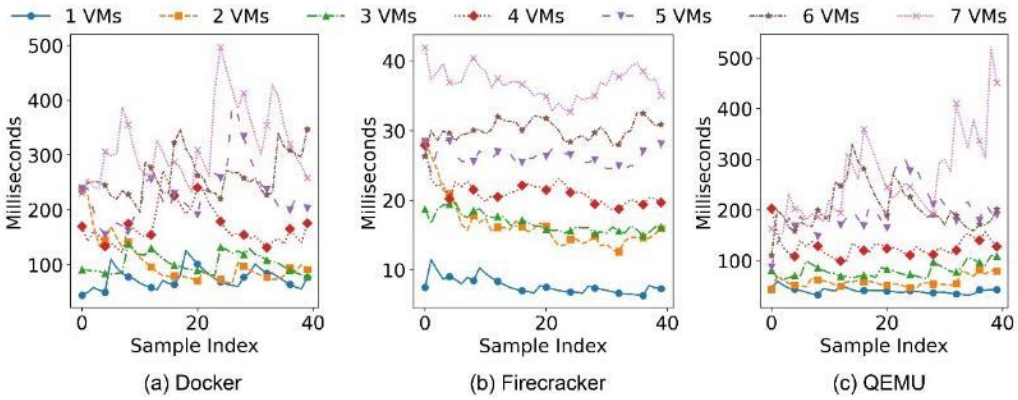


**Рис. 5.** Порівняння середньої затримки доступу до диска між усіма запущеними віртуальними середовищами під час проведення експерименту

*Firecracker* демонструє найкращу продуктивність диска, що пов'язано з використанням мінімалістичного дизайну, що означає менше емуляції. *Firecracker* використовує віртуальні пристрої безпосередньо,

*Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 3 (33)*
*ISSN 2522-9818 (print)*
*ISSN 2524-2296 (online)*

а не повну апаратну емуляцію, як *QEMU*. *Docker*, хоч і швидко обробляє повідомлення, але працює на файловій системі хост-системи й часто використовує накладну файлову систему (*overlay filesystem*), що створює додаткові витрати під час кожного запису / читання. Крім цього, наявні недоліки в разі значної кількості контейнерів, бо вони конкурують за одні й ті самі ресурси. Через те, що *Docker* не має ізоляції введення-виведення на рівні ядра, як це можливо у *Firecracker*, спостерігаємо сильну різницю в отриманих показниках. *QEMU* загалом схожий на *Firecracker* за показниками, але через надмірну емуляцію в межах віртуалізації на одноплатному комп'ютері все ж таки поступається останньому. *QEMU* так чи інакше створює високі накладні витрати на операції введення-виведення, а можливість кешувати дані, що відсутня у *Docker*, робить його кращим за цим критерієм, ніж контейнери.

Загальним критерієм, що тільки побічно інформує про ефективність віртуалізації, є навантаження центрального процесора. За декілька хвилин експерименту для кожної кількості віртуальних середовищ відсоток навантаження зміг стабілізуватися (див. рис. 6).

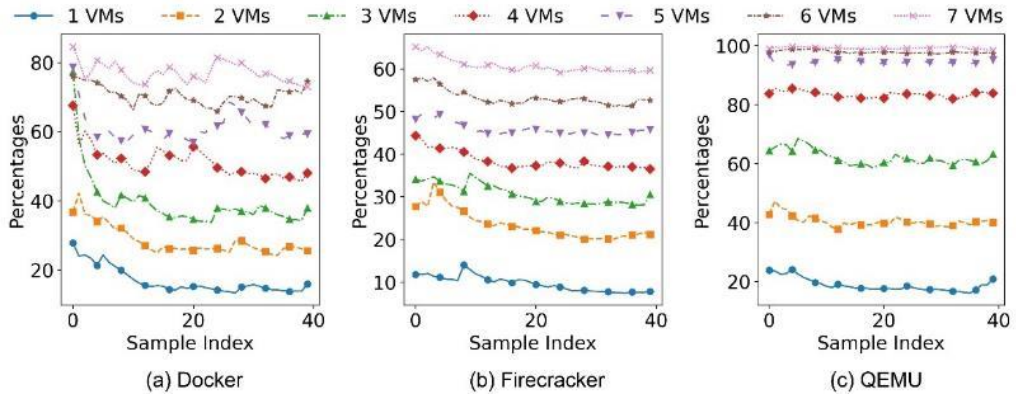


**Рис. 6.** Порівняння завантаженості центрального процесора під час проведення експерименту з віртуалізації цифрових двійників

Відразу спостерігаємо недостатню потужність у *Pi 4* для підтримки семи віртуальних машин. Уже на п'яти машинах платформа була навантажена майже на максимум. Саме ці показники переконали відмовитися від восьми віртуальних середовищ. Максимальна ізоляція та емуляція девайсів і ресурсів спричиняє надмірні витрати часу центрального процесора. *Docker* демонструє вищі пікові навантаження, якщо порівнювати з *Firecracker*. Це можна пояснити прямим доступом контейнерів до ресурсів хост-системи та відсутністю ізоляції. Хоча *Firecracker* показує кращі результати щодо завантаження процесора, але беручи до уваги інші критерії, не можна сказати, що *Firecracker* виконує задачі швидше за *Docker*.

Схожа ситуація і з температурою центрального процесора, що прямо залежить від його навантаження. Кожний наступний цифровий двійник сильніше навантажує систему, що призводить до більшого нагрівання чипа (див. рис. 7).

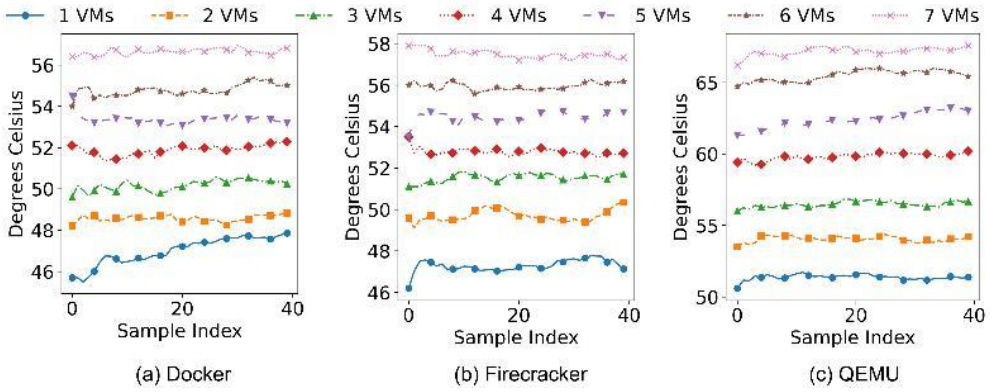


**Рис. 7.** Порівняння температури центрального процесора платформи під час навантаження віртуалізованих цифрових двійників

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2025. No. 3 (33)

*Docker* та *Firecracker* майже однаково нагрівають процесор, тоді як *QEMU*, дійшовши до максимального навантаження, розігріває платформу до 70ºС. Це не максимально допустима температура, що становить 85ºС, але, зрештою, значно вища за альтернативи.

## Висновки

Результати дослідження продемонстрували суттєві розбіжності в ефективності використання обчислювальних ресурсів, у стабільності та затримках передачі інформації в умовах упровадження обраних технологій віртуалізації *Docker*, *Firecracker* та *QEMU* і застосування цифрового двійника на одноплатному комп'ютері *Raspberry Pi 4 Model B*.

У межах поставленого завдання, а саме віртуалізації легкого й базового цифрового двійника, *Docker* виявився найкращим варіантом для сценаріїв із високими вимогами до продуктивності та мінімальних затримок. Його легка архітектура дала змогу досягти найменших затримок у мережі та обміну повідомленнями *MQTT*, а також помірного навантаження на процесор. Однак це супроводжувалося вищим навантаженням на центральний процесор, що свідчить про інтенсивну взаємодію з хост-системою.

Визначено, що *Firecracker* має більше накладних витрат, ніж *Docker*, але також залишається дуже вдалим рішенням, що поєднує високу ефективність доступу до диска, швидкий час запуску. Незважаючи на дещо вищу мережну затримку, ця технологія забезпечує ізоляцію та стабільність, що є вкрай важливим для безпеки хост-системи та інших віртуалізованих середовищ.

Найменш ефективною технологією в межах проведеного експерименту є *QEMU*. Попри всі переваги повної віртуалізації, обрана платформа не може впоратися з таким навантаженням на процесор, тому масштабування до рівня *Firecracker* чи *Docker* не є можливим. Використання *QEMU* призвело до високих температур чипа та значних затримок як у процесі мережної комунікації, так і під час роботи з диском. У застосуванні лише декількох віртуальних машин *QEMU* є гарною альтернативою, особливо коли ізоляція та емуляція є важливими компонентами.

Отже, вибір технології віртуалізації на одноплатному комп'ютері залежно від потреб насамперед буде на користь *Docker* або *Firecracker*. Перший кращий щодо налагоджування та функціоналу за більш прості проєкти чи системи, а *Firecracker* ефективний, коли ізоляція та підтримка саме окремого диска мають значення в прийнятті рішень. *QEMU* можна застосовувати в окремих випадках, які малоймовірно будуть дотичними до контексту цифрового двійника.

## Перспективи подальшого дослідження

У статті проаналізовано ефективність віртуалізації на одноплатних комп'ютерах у контексті цифрових двійників, однак є низка напрямів, що заслуговує на подальше опрацювання та розширення.

По-перше, хоча було обрано *Raspberry Pi 4* як платформу для проведення експериментів, існують і інші альтернативи, що можуть бути більш придатними платформами за інших критеріїв, наприклад *Raspberry Pi 5*. Нова апаратна платформа пропонує значно вищу потужність обчислень завдяки кращому процесору. Проте в цьому разі вона потребує більш складного охолодження та є менш стабільною, що пояснюється її новизною. Порівняльний експеримент на *Pi 5* виявив би не тільки різницю в продуктивності, а й у стабільності, масштабованості та температурних обмеженнях, що суттєво розширило б межі дослідження в контексті цифрових двійників на *SBC*.

По-друге, з обраних критеріїв порівняння видно, що було зосереджено увагу на стані всієї платформи та майже всі показники отримані ззовні віртуалізованого середовища, а не зсередини. Один із напрямів подальшого розвитку міг бути присвячений глибшому вивченню саме внутрішнього стану віртуалізованих середовищ. Перехід від моделі "чорної скриньки" до "білої скриньки", де збираються внутрішні метрики віртуалізованої ОС, дало б змогу краще зрозуміти вплив шару віртуалізації на продуктивність машини.

По-третє, було б цілком доцільно винести частину зібраних телеметричних показників із сенсорів за межі експериментальної платформи, наприклад на окремий мікроконтролер, як-от *ESP32*, який передаватиме всі значення по *MQTT* у віртуальні середовища. Отже, експериментальна платформа цілком зосереджена на віртуалізації цифрових двійників, і такий розподіл завдань є іншою загальноприйнятою альтернативою в розподілених системах *IoT*, де інформація надходить у хмару чи прилеглі сервери з окремих пристроїв.

По-четверте, практичне навантаження може бути ускладнене та розширене. Замість запропонованої

*Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 3 (33)*
*ISSN 2522-9818 (print)*
*ISSN 2524-2296 (online)*

пари технологій *Node-RED* та *InfluxDB*, можна моделювати більш складні цифрові двійники, що передбачатимуть аналіз аномалій, візуалізацію, навчання моделей машинного навчання, передачу команд по *MQTT* безпосередньо до сенсорів тощо. Якщо ускладнювати практичне навантаження, можна все ближче й ближче відтворювати реальні приклади з *IoT*, але водночас зберігати передбачуваність і контрольованість експериментальної системи.

Усі окреслені напрями не тільки розширюють прикладну цінність експерименту, але й відкривають перспективи подальших наукових досліджень у сфері периферійних обчислень, ефективної віртуалізації та цифрових двійників у системах *IoT*.

**Список літератури**

1. Saniya Z., Roohie N. M. Resource management in pervasive Internet of Things: A survey. *Journal of King Saud University Computer and Information Sciences*. 2021. Vol. 33. № 8. P. 921–935. DOI: https://doi.org/10.1016/j.jksuci.2018.08.014
2. Lambropoulos G., Mitropoulos S., Douligeris C., Maglaras L. Implementing Virtualization on Single-Board Computers: A Case Study on Edge Computing. *Computers*. 2024. Vol. 13. №. 2. 54 p. DOI: https://doi.org/10.3390/computers13020054
3. Álvarez J.L., Mozo J.D., Durán E. Analysis of Single Board Architectures Integrating Sensors Technologies. *Sensors*. 2021. Vol. 21. 6303 p. DOI: https://doi.org/10.3390/s21186303
4. Gamess E., Parajuli M., Shah S. Performance evaluation of the KVM hypervisor running on ARM-based single-board computers. *International Journal of Computer Networks & Communications (IJCNC)*. 2023. Vol. 15. №. 2. P. 147–164. DOI: 10.5121/ijcnc.2023.15208
5. Johnston J. S., Basford J. P., Perkins S. C., Herry H., Tso F. P., Pezaros D., Mullins D. R., Yoneki E., Cox J. S., Singer J. Commodity single board computer clusters and their applications. *Future Generation Computer Systems*. 2018. Vol. 89. P. 201–212. DOI: https://doi.org/10.1016/j.future.2018.06.048
6. Penchalaiah N., Al-Humaimeedy A.S., Mashael M., Babu C.J., Khan O.I., Aldhyani T.H.H. Clustered Single-Board Devices with Docker Container Big Stream Processing Architecture. *Computers, Materials and Continua*. 2022. Vol. 73. №. 3. P. 5349--5365. DOI: https://doi.org/10.32604/cmc.2022.029639
7. Hwang K., Jung I.H., Lee J.M. Monitoring of MQTT-based Messaging Server. *Webology*. 2022. Vol. 19. №. 1. P. 4724–4735. DOI: 10.14704/WEB/V19I1/WEB19316
8. Alam T., Rokonuzzaman M., Sarker S., Azim Z., Das T., Hossain M.I. Internet of Things-based Home Automation with Network Mapper and MQTT Protocol. *Computers and Electrical Engineering*. 2024. Vol. 120. 109807 p. DOI: https://doi.org/10.1016/j.compeleceng.2024.109807
9. Ahsen R., Di Bitonto P., Novielli P., Magarelli M., Romano D., Diacono D., Monaco A., Amoroso N., Bellotti R., Tangaro S. Harnessing. Digital Twins for Sustainable Agricultural Water Management: A Systematic Review. *Applied Sciences*. 2025. Vol. 15. 4228 p. DOI: https://doi.org/10.3390/app15084228
10. Vidyalakshmi G., Gopikrishnan S., Wadii B., Anis K., Gautam S. Digital Twins and Cyber-Physical Systems: A New Frontier in Computer Modeling. *CMES – Computer Modeling in Engineering and Sciences*. 2025. Vol. 143. №. 1. P. 51–113. DOI: https://doi.org/10.32604/cmes.2025.057788
11. Raspberry Pi 4 Official Website. URL: https://www.raspberrypi.com/products/raspberry-pi-4-model-b (дата звернення: 13.09.2024)
12. Raspberry Pi 5. The everything computer. Optimised. URL: https://www.raspberrypi.com/products/raspberry-pi-5/ (дата звернення: 14.09.2024)
13. Radxa ROCK 5B. URL: https://radxa.com/products/rock5/5b (дата звернення: 01.10.2024)
14. Le Potato AML-S905X-CC. URL: https://libre.computer/products/aml-s905x-cc (дата звернення: 01.10.2024)
15. Orange Pi 5. URL: http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/Orange-Pi-5.html (дата звернення: 01.10.2024)
16. Docker Homepage. URL: https://www.docker.com/ (дата звернення: 06.01.2025)
17. Oracle Virtual Box vs VMware Workstation vs QEMU. URL: https://www.starwindsoftware.com/blog/type-2-hypervisors-comparison-virtualbox-vmware-qemu/ (дата звернення: 25.11.2024)
18. Getting started with Firecracker on Raspberry Pi. URL: https://dev.to/l1x/getting-started-with-firecracker-on-raspberry-pi-1pbc (дата звернення: 10.02.2025)

*ISSN 2522-9818 (print)*
*ISSN 2524-2296 (online)* *Innovative technologies and scientific solutions for industries. 2025. No. 3 (33)*

**References**

1. Saniya, Z., Roohie, N. (2021), "Resource management in pervasive Internet of Things: A survey", *Journal of King Saud University* Computer and Information Sciences, No. 33(8), P. 921–935. DOI: https://doi.org/10.1016/j.jksuci.2018.08.014
2. Lambropoulos, G., Mitropoulos, S., Douligeris, C., Maglaras, L. (2024), "Implementing Virtualization on Single-Board Computers: A Case Study on Edge Computing", *Computers*, No. 13(2), 54 p. DOI: https://doi.org/10.3390/computers13020054
3. Álvarez, J., Mozo, J., Durán, E. (2021), "Analysis of Single Board Architectures Integrating Sensors Technologies", *Sensors*, No. 21, 6303 p. DOI: https://doi.org/10.3390/s21186303
4. Gamess, E., Parajuli, M., Shah, S. (2023), "Performance evaluation of the KVM hypervisor running on ARM-based single-board computers", *International Journal of Computer Networks & Communications (IJCNC)*, No. 15(2), P. 147–164. DOI: 10.5121/ijcnc.2023.15208
5. Johnston, J., Basford, J., Perkins, S., Herry, H., Tso, F., Pezaros, D., Mullins, D., Yoneki, E., Cox, J., Singer, J. (2018), "Commodity single board computer clusters and their applications", *Future Generation Computer Systems*, No. 89, P. 201–212. DOI: https://doi.org/10.1016/j.future.2018.06.048
6. Penchalaiah, N., Al-Humaimeedy, A., Mashael, M., Babu, C., Khan, O., Aldhyani, T. (2022), "Clustered Single-Board Devices with Docker Container Big Stream Processing Architecture", *Computers*, Materials and Continua, No. 73(3), P. 5349–5365. DOI: https://doi.org/10.32604/cmc.2022.029639
7. Hwang, K., Jung, I., Lee, J. (2022), "Monitoring of MQTT-based Messaging Server", *Webology*, No. 19(1), P. 4724–4735. DOI: 10.14704/WEB/V19I1/WEB19316
8. Alam, T., Rokonuzzaman, M., Sarker, S., Azim, Z., Das, T., Hossain, M. (2024), "Internet of Things-based Home Automation with Network Mapper and MQTT Protocol", *Computers and Electrical Engineering*, No. 120. 109807 p. DOI: https://doi.org/10.1016/j.compeleceng.2024.109807
9. Ahsen, R., Di Bitonto, P., Novielli, P., Magarelli, M., Romano, D., Diacono, D., Monaco, A., Amoroso, N., Bellotti, R., Tangaro, S. (2025), "Digital Twins for Sustainable Agricultural Water Management: A Systematic Review", *Applied Sciences*, No. 15, 4228 p. DOI: https://doi.org/10.3390/app15084228
10. Vidyalakshmi, G., Gopikrishnan, S., Wadii, B., Anis, K., Gautam, S. (2025), "Digital Twins and Cyber-Physical Systems: A New Frontier in Computer Modeling", *CMES - Computer Modeling in Engineering and Sciences*, No. 143(1), P. 51–113. DOI: https://doi.org/10.32604/cmes.2025.057788
11. Raspberry Pi 4 Official Website, available at: https://www.raspberrypi.com/products/raspberry-pi-4-model-b (last accessed: 13.09.2024)
12. Raspberry Pi 5. The everything computer. Optimised, available at: https://www.raspberrypi.com/products/raspberry-pi-5/ (last accessed: 14.09.2024)
13. Radxa ROCK 5B, available at: https://radxa.com/products/rock5/5b (last accessed: 01.10.2024)
14. Le Potato AML-S905X-CC, available at: https://libre.computer/products/aml-s905x-cc (last accessed: 01.10.2024)
15. Orange Pi 5, available at: http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/Orange-Pi-5.html (last accessed: 01.10.2024)
16. Docker Homepage, available at: https://www.docker.com/ (last accessed: 06.01.2025)
17. Oracle Virtual Box vs VMware Workstation vs QEMU, available at: https://www.starwindsoftware.com/blog/type-2-hypervisors-comparison-virtualbox-vmware-qemu/ (last accessed: 25.11.2024)
18. Getting started with Firecracker on Raspberry Pi, available at: https://dev.to/l1x/getting-started-with-firecracker-on-raspberry-pi-1pbc (last accessed: 10.02.2025)

*Надійшла (Received) 22.05.2025*

*Відомості про авторів / About the Authors*

**Свєтлінський Олег Андрійович** – аспірант, Харківський національний університет радіоелектроніки, кафедра програмної інженерії, Харків, Україна; e-mail: oleh.svietlinskyi@nure.ua; ORCID ID: https://orcid.org/0009-0003-4221-9297

**Ревенчук Ілона Анатоліївна** – кандидат технічних наук, доцент, Харківський національний університет радіоелектроніки, доцент кафедри програмної інженерії, Харків, Україна; e-mail: ilona.revenchuk@nure.ua; ORCID ID: https://orcid.org/0000-0002-5188-9538

*Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 3 (33)*
*ISSN 2522-9818 (print)*
*ISSN 2524-2296 (online)*

**Svietlinskyi Oleh** – Postgraduate, Kharkiv National University of Radio Electronics, Department of Software Engineering, Kharkiv, Ukraine.

**Revenchuk Ilona** – PhD (Engineering Sciences), Associate Professor, Kharkiv National University of Radio Electronics, Associate Professor at the Department of Software Engineering, Kharkiv, Ukraine.

# RESEARCH ON THE EFFICIENCY OF VIRTUALIZATION SOLUTIONS FOR DEPLOYING DIGITAL TWINS IN SYSTEMS WITH LIMITED RESOURCES

The **subject matter** of the article is the means and approaches to virtualization in environments with limited resources, in particular on single-board computers, as well as their ability to ensure the effective operation of digital twins. The **goal** of the work is to evaluate the effectiveness of virtualization for deploying digital twins on single-board computers and develop recommendations for virtualization usage in environments with limited computing resources. The following **tasks** were solved in the article: analysis of modern hardware solutions for single-board computers and existing virtualization technologies; construction of an experimental environment for practical comparison of virtualization technologies on single-board computers; implementation of an application load that simulates the operation of a digital twin with real sensor data; study of the efficiency of network interaction between virtual environments and the host system; comparison of results by key metrics. The following **methods** used are: virtualization technologies Docker, QEMU, Firecracker; network monitoring; thorough analysis of literature and available documentation on the topic; comparative analysis of the achieved results. The following **results** were obtained: a comparative experiment using Docker, QEMU and Firecracker based on Raspberry Pi 4 was carried out; thorough analysis of literature and available documentation on the topic was conducted; a basic prototype of a digital twin with the connection of physical sensor devices that collect telemetry data in real time was implemented; phased automatic deployment of virtual environments with pre-configured components was provided for rapid scaling of the experiment; network delays, message processing time and platform resource usage were measured; the advantages and limitations of virtualization technologies in conditions of limited computing resources were determined; the practical feasibility of using each type of virtualization in conditions of limited computing resources was assessed. **Conclusions**: Docker provides the easiest deployment and highest efficiency, but has less isolation compared to QEMU and Firecracker; Firecracker demonstrates the optimal balance between performance, security, and resource consumption on single-board computers; QEMU has the highest overhead, and therefore the lowest efficiency; the use of lightweight hypervisors is advisable in digital twin systems on peripheral devices.

**Keywords**: virtualization; single-board computers; digital twin; network monitoring; Raspberry Pi.

*Бібліографічні описи / Bibliographic descriptions*

Свєтлінський О.А., Ревенчук І.А. Дослідження ефективності віртуалізаційних рішень для розгортання цифрових двійників у системах з обмеженими ресурсами. *Сучасний стан наукових досліджень та технологій в промисловості*. 2025. № 3 (33). С. 137–151. DOI: https://doi.org/10.30837/2522-9818.2025.3.137

Svietlinskyi, O, Revenchuk, I. (2025), "Research on the efficiency of virtualization solutions for deploying digital twins in systems with limited resources", *Innovative Technologies and Scientific Solutions for Industries*, No. 3 (33), P. 137–151. DOI: https://doi.org/10.30837/2522-9818.2025.3.137