НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Навчально-науковий інститут телекомунікаційних систем

Кафедра телекомунікацій

До захисту допущено:

В.о.завідувача кафедри

_____ Сергій КРАВЧУК

«___»_____2026 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інженерія та програмування інфокомунікацій»

спеціальності 172 «Телекомунікації та радіотехніка»

на тему: «Побудова NAS (Network Attached Storage) на основі Raspberry Pi 5»

Виконав:
студент IV курсу, групи ТЗ-21
Кундель Олександр Леонідович _____

Керівник:
доцент кафедри ТК НН ІТС , к.т.н., доцент,
Авдєєнко Гліб Леонідович _____

Рецензент:
доцент кафедри ІТТ НН ІТС, к.т.н., доцент,
Новогрудська Ріна Леонідівна _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2026 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут телекомунікаційних систем
Кафедра телекомунікацій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Інженерія та програмування інфокомунікацій»

ЗАТВЕРДЖУЮ
В.о.завідувача кафедри
_____ Сергій КРАВЧУК
«___»_____2026 р.

ЗАВДАННЯ

на дипломну роботу студенту

Кунделю Олександру Леонідовичу

1. Тема роботи «Побудова NAS (Network Attached Storage) на основі Raspberry Pi-5», керівник роботи Авдєєнко Гліб Леонідович, к.т.н., затверджені наказом по університету від «26» травня 2026 р. № 1912-с.

2. Термін подання студентом роботи 10 червня 2026 р.

3. Вихідні дані до роботи: Одноплатний комп'ютер Raspberry Pi 5 (8 ГБ RAM); NVMe SSD 256 ГБ з платою M.2 HAT+; блок живлення USB-C 27 Вт; активне охолодження. ОС Raspberry Pi OS (64-bit). Гігабітна локальна мережа. Доступ до даних за протоколом SMB (Samba); віддалений доступ через VPN та SFTP. Вимоги: централізоване мережеве зберігання даних, оптимізація продуктивності, експериментальне дослідження швидкісних характеристик сховища.

4. Зміст роботи: 1) огляд технологій та платформ для побудови NAS; 2) програмно-апаратна реалізація NAS на базі Raspberry Pi 5; 3) оптимізація продуктивності NAS-системи; 4) експериментальні дослідження продуктивності NAS.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) презентація за результатами роботи: структурна схема NAS-системи; схема мережевої взаємодії та віддаленого доступу; графіки результатів

тестування продуктивності (PCIe Gen2/Gen3, Ethernet vs Wi-Fi, вплив розміру файлів); порівняння з комерційними рішеннями.

6. Дата видачі завдання 10.01.2026 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області та огляд джерел	10.01.2026 - 10.02.2026	Виконано
2	Вибір апаратного та програмного забезпечення	10.02.2026 - 01.03.2026	Виконано
3	Програмно-апаратна реалізація NAS	01.03.2026 - 31.03.2026	Виконано
4	Оптимізація продуктивності системи	01.04.2026 - 15.04.2026	Виконано
5	Експериментальні дослідження та аналіз результатів	15.04.2026 - 01.05.2026	Виконано
6	Оформлення пояснювальної записки	01.05.2026 - 10.06.2026	Виконано
7	Підготовка презентації до захисту	10.06.2026 - 15.06.2026	Виконано

Студент Олександр КУНДЕЛЬ

Керівник Гліб АВДЄЄНКО

РЕФЕРАТ

Обсяг пояснювальної записки становить 71 сторінку, наявні 16 рисунків, 6 таблиць, 3 додатки та 30 джерел за переліком посилань.

Метою дипломної роботи є розроблення, оптимізація та експериментальне дослідження продуктивного й економічно ефективного мережевого сховища даних (NAS) на базі одноплатного комп'ютера Raspberry Pi 5.

У роботі виконано порівняльний аналіз технологій побудови NAS (комерційні рішення та самостійне складання, файлові системи, мережеві протоколи) та обґрунтовано вибір апаратної і програмної платформи. Реалізовано повнофункціональне мережеве сховище із завантаженням операційної системи безпосередньо з NVMe-накопичувача, файловим сервісом Samba, приватною хмарою NextCloud та захищеним віддаленим доступом через VPN-мережу Tailscale і SFTP; виконано комплекс заходів із захисту сервера (автентифікація лише за криптографічним ключем, скорочення поверхні атаки, заборона гостьового доступу). Використано експериментальні методи дослідження: вимірювання продуктивності інструментами fio та iperf3 за схемою «до/після» з обробкою результатів методами математичної статистики (середнє, стандартне відхилення, коефіцієнт варіації).

Основні результати: активація режиму PCIe Gen 3 підвищила локальну швидкість накопичувача приблизно на 88 % (до 825-858 МБ/с); файловий обмін через Samba у дротовій гігабітній мережі досягає 110,8 МБ/с, тобто близько 93 % теоретичної межі каналу; експериментально доведено, що визначальним обмеженням продуктивності системи є пропускна здатність гігабітного мережевого інтерфейсу, а не накопичувач чи процесор. Кількісно оцінено вплив типу під'єднання клієнта (дротове з'єднання у 4-5 разів швидше за Wi-Fi) та розміру файлів на швидкість обміну. За співвідношенням «ціна/продуктивність» розроблена система (близько 216 EUR) не поступається комерційним NAS початкового рівня.

Результати роботи можуть використовуватися для побудови доступних мережевих сховищ для дому та малого офісу, а також у навчальному процесі; запропонована методика вимірювань придатна для дослідження подібних систем.

Ключові слова: МЕРЕЖЕВЕ СХОВИЩЕ ДАНИХ, NAS, RASPBERRY PI 5, NVME, PCIE, SAMBA, SMB, ПРОДУКТИВНІСТЬ, ОПТИМІЗАЦІЯ, ПРОПУСКНА ЗДАТНІСТЬ, ВІДДАЛЕНИЙ ДОСТУП, VPN.

ABSTRACT

The explanatory note contains 71 pages, 16 figures, 6 tables, 3 appendices and 30 references.

The purpose of this thesis is to develop, optimize and experimentally evaluate a productive and cost-effective network attached storage (NAS) based on the Raspberry Pi 5 single-board computer.

The thesis presents a comparative analysis of NAS technologies (commercial devices and DIY solutions, file systems, network protocols) and explains the choice of the hardware and software platform. A fully functional network storage was built: the operating system boots directly from an NVMe drive, file access is provided by the Samba service, and remote access is organized through the Tailscale VPN network, SFTP and the NextCloud private cloud. Security measures were also applied: authentication only with a cryptographic key, reduction of the attack surface and prohibition of guest access. Experimental methods were used in the work: performance measurements with the fio and iperf3 tools according to the "before/after" scheme, with statistical processing of the results (mean value, standard deviation, coefficient of variation).

Main results: activation of the PCIe Gen 3 mode increased the local drive speed by about 88 % (up to 825-858 MB/s); Samba file transfer in a wired Gigabit network reaches 110.8 MB/s, which is about 93 % of the theoretical limit of the channel; it was experimentally proven that the main performance limitation of the system is the bandwidth of the Gigabit network interface, and not the drive or the processor. The influence of the client connection type (a wired connection is 4-5 times faster than Wi-Fi) and of the file size on the transfer speed was also measured. By the price/performance ratio the developed system (about 216 EUR) is not worse than commercial entry-level NAS devices.

The results of the work can be used to build affordable network storage for home and small office, and also in the educational process; the proposed measurement methodology can be applied to similar systems.

Keywords: NETWORK ATTACHED STORAGE, NAS, RASPBERRY PI 5, NVME, PCIE, SAMBA, SMB, PERFORMANCE, OPTIMIZATION, THROUGHPUT, REMOTE ACCESS, VPN.

ЗМІСТ

ВСТУП 11
РОЗДІЛ 1 ОГЛЯД ТЕХНОЛОГІЙ ТА ПЛАТФОРМ ДЛЯ ПОБУДОВИ NAS 14
1.1 Порівняльний аналіз комерційних NAS та DIY-рішень 15
1.2 Аналіз файлових систем 16
1.3 Мережеві протоколи доступу до даних 18
1.4 Архітектурні особливості Raspberry Pi 5 19
Висновки 21
РОЗДІЛ 2 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ NAS 23
2.1 Обґрунтування вибору апаратного забезпечення 23
2.2 Конфігурація операційної системи та мережевого стека 26
2.3 Налаштування дискової підсистеми та міграція системи на NVMe 29
2.4 Розгортання та налаштування сервісу Samba 33
2.5 Організація віддаленого доступу та захисту 34
2.5.1 Захищений канал Tailscale 35
2.5.2 Передавання файлів через SFTP 36
2.5.3 Приватна хмара NextCloud 37
2.5.4 Посилення захисту SSH (hardening) 38
2.5.5 Скорочення поверхні атаки 39
Висновки 40
РОЗДІЛ 3 ОПТИМІЗАЦІЯ ПРОДУКТИВНОСТІ NAS-СИСТЕМИ 42
3.1 Оптимізація мережевого стека ядра Linux 42
3.2 Налаштування IRQ Affinity та CPU Governor 43
3.3 Активація PCIe Gen 3 та налаштування параметрів NVMe 45
3.4 Оптимізація параметрів Samba та аналіз «вузького місця» 47
Висновки 49
РОЗДІЛ 4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ПРОДУКТИВНОСТІ NAS 51
4.1 Розроблення методики тестування та опис стенду 51
4.2 Аналіз пропускної здатності мережі 53
4.3 Порівняльне тестування Samba: клієнт по Ethernet проти Wi-Fi 55
4.4 Дослідження впливу розміру та кількості файлів 56
4.5 Зведена таблиця результатів та порівняння з комерційними NAS 58
Висновки 61
ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ 62

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 65
ДОДАТКИ 68
Додаток А 68
Фрагмент журналу ядра (dmesg) зі збоєм NVMe-контролера через механізм APST 68
Додаток Б 69
Підсумкова конфігурація сервісу Samba (вивід testparm -s) 69
Додаток В 70
Сценарій вимірювання впливу розміру файлів на швидкість запису (PowerShell) 70

ПЕРЕЛІК СКОРОЧЕНЬ

ОС	операційна система
ПЗ	програмне забезпечення
ПК	персональний комп'ютер
ФС	файлова система
APST	(Autonomous Power State Transitions) - механізм автономного перемикання станів живлення NVMe-накопичувача
ARM	(Advanced RISC Machine) - архітектура процесорів зі скороченим набором команд
BDP	(Bandwidth-Delay Product) - добуток пропускної здатності каналу на затримку
CIFS	(Common Internet File System) - застаріла назва протоколу SMB
CPU	(Central Processing Unit) - центральний процесор
CV	(Coefficient of Variation) - коефіцієнт варіації
DIY	(Do It Yourself) - підхід самостійної побудови системи з окремих компонентів
DNS	(Domain Name System) - система доменних імен
EEPROM	(Electrically Erasable Programmable Read-Only Memory) - енергонезалежна пам'ять, що електрично перепрограмовується
FTP	(File Transfer Protocol) - протокол передавання файлів
GbE	(Gigabit Ethernet) - гігабітна мережа Ethernet
HDD	(Hard Disk Drive) - накопичувач на жорстких магнітних дисках
IOPS	(Input/Output Operations Per Second) - кількість операцій введення-виведення за секунду
IP	(Internet Protocol) - міжмережевий протокол
IRQ	(Interrupt Request) - запит апаратного переривання
LAN	(Local Area Network) - локальна обчислювальна мережа
NAS	(Network Attached Storage) - мережеве сховище даних
NAT	(Network Address Translation) - трансляція мережевих адрес
NFS	(Network File System) - мережева файлова система
NTP	(Network Time Protocol) - протокол синхронізації часу

NVMe	(Non-Volatile Memory Express) - інтерфейс доступу до твердотільних накопичувачів через шину PCIe
PCIe	(Peripheral Component Interconnect Express) - високошвидкісна послідовна шина введення-виведення
RAID	(Redundant Array of Independent Disks) - надлишковий масив незалежних дисків
RAM	(Random Access Memory) - оперативна пам'ять
RTT	(Round-Trip Time) - час проходження пакета до вузла та назад
SD	(Secure Digital) - формат карт флешпам'яті
SFTP	(SSH File Transfer Protocol) - протокол передавання файлів поверх SSH
SMB	(Server Message Block) - мережевий протокол спільного доступу до файлів
SSD	(Solid-State Drive) - твердотільний накопичувач
SSH	(Secure Shell) - протокол захищеного віддаленого доступу
TCP	(Transmission Control Protocol) - протокол керування передаванням даних
USB	(Universal Serial Bus) - універсальна послідовна шина
VPN	(Virtual Private Network) - віртуальна приватна мережа
WAN	(Wide Area Network) - глобальна обчислювальна мережа
Wi-Fi	технологія бездротових локальних мереж (стандарти IEEE 802.11)

ВСТУП

Актуальність теми. Обсяги цифрових даних, які створюють і зберігають користувачі, невпинно зростають: фото- та відеоархіви, резервні копії пристроїв, робочі документи, медіабібліотеки. Зберігання цих даних лише на окремих пристроях (ноутбуках, смартфонах, зовнішніх дисках) є ненадійним і незручним - дані роз'єднані, не захищені від втрати при поломці носія та недоступні одночасно з різних пристроїв. Поширеним рішенням стали хмарні сервіси, проте вони мають істотні недоліки: абонентська плата, залежність від інтернет-з'єднання та, найголовніше, втрата контролю над приватними даними, які зберігаються на чужих серверах.

Альтернативою, що поєднує зручність централізованого доступу з повним контролем над власними даними, є мережеве сховище (NAS, Network Attached Storage) - автономний пристрій, під'єднаний до локальної мережі. Готові комерційні NAS (Synology, QNAP) є зручними, але відносно дорогими та закритими в межах екосистеми виробника. Водночас поява доступних одноплатних комп'ютерів, зокрема Raspberry Pi 5, відкрила можливість самостійної побудови продуктивного NAS за значно нижчою ціною. Принциповим є те, що Raspberry Pi 5 - на відміну від попередніх поколінь - отримав апаратний інтерфейс PCIe, який уперше дозволяє під'єднати швидкий твердотільний накопичувач NVMe безпосередньо. Це робить платформу придатною для побудови саме продуктивного сховища, а не лише навчального макета. Таким чином, дослідження побудови та оптимізації NAS на базі Raspberry Pi 5 є актуальним як з практичного (доступне сховище для дому й малого офісу), так і з технічного погляду (виявлення реальних можливостей і обмежень нової платформи).

Мета роботи - розроблення, оптимізація та експериментальне дослідження продуктивного й економічно ефективного мережевого сховища даних (NAS) на базі одноплатного комп'ютера Raspberry Pi 5.

Задачі роботи. Для досягнення поставленої мети необхідно вирішити такі задачі:

1. виконати огляд та порівняльний аналіз технологій і платформ для побудови NAS - підходів до реалізації (комерційні рішення проти самостійного складання), файлових систем, мережевих протоколів доступу та архітектурних особливостей обраної апаратної платформи; 2. реалізувати програмно-апаратну частину NAS: обґрунтувати вибір апаратних компонентів, встановити й налаштувати операційну систему, дискову підсистему, файловий сервіс Samba та засоби віддаленого доступу, забезпечивши належний рівень безпеки; 3. виконати пошарову оптимізацію продуктивності системи (мережевий стек, дискова підсистема, параметри файлового сервісу) з експериментальною перевіркою ефекту кожної зміни та виявленням «вузького місця» системи; 4. провести комплексне експериментальне дослідження продуктивності сховища за єдиною методикою: визначити реальну швидкість файлового обміну, оцінити вплив типу під'єднання (дротового та бездротового) і характеру даних, а також порівняти отримане рішення з комерційними аналогами.

Об'єкт дослідження - процеси зберігання та передавання даних у мережевому сховищі.

Предмет дослідження - методи та засоби побудови й підвищення продуктивності NAS на базі одноплатного комп'ютера Raspberry Pi 5.

Методи дослідження. У роботі застосовано аналіз і систематизацію науково-технічних джерел та документації (для обґрунтування вибору технологій); методи системного проєктування (для побудови програмно-апаратної конфігурації); експериментальні методи - вимірювання продуктивності за допомогою спеціалізованих інструментів (`fio`, `iperf3`, штатні засоби копіювання) за схемою «до/після»; методи математичної статистики для обробки результатів вимірювань (обчислення середнього, стандартного відхилення та коефіцієнта варіації для оцінювання достовірності).

Наукова новизна та практичне значення. Практичну цінність роботи становить готова, відтворювана конфігурація продуктивного NAS на

доступній платформі вартістю близько 216 EUR, яку можна використати як для домашнього сховища, так і в навчальному процесі. Окремим результатом є експериментально обґрунтована методика вимірювання продуктивності та встановлений факт, що для розглянутої системи визначальним обмеженням продуктивності є не дискова підсистема чи процесор, а пропускна здатність гігабітного мережевого каналу. Це спростовує поширене уявлення про повільність накопичувача як головну ваду подібних рішень і чітко вказує напрям подальшого розвитку - перехід на швидші мережеві інтерфейси. Показано також, що активація режиму PCIe Gen 3 майже подвоює локальну швидкість накопичувача навіть на бюджетному SSD.

Структура та обсяг роботи. Робота складається зі вступу, чотирьох розділів, загальних висновків, списку використаних джерел і додатків.

У першому розділі виконано огляд технологій та платформ для побудови NAS і обґрунтовано вибір рішень, реалізованих у роботі. У другому розділі описано програмно-апаратну реалізацію сховища: від вибору компонентів і встановлення операційної системи до налаштування файлового сервісу, віддаленого доступу та засобів безпеки. Третій розділ присвячено пошаровій оптимізації продуктивності з експериментальною перевіркою кожної зміни та виявленням «вузького місця» системи. У четвертому розділі наведено комплексне експериментальне дослідження продуктивності: вплив типу під'єднання, розміру файлів та порівняння з комерційними рішеннями. У загальних висновках узагальнено отримані результати.

РОЗДІЛ 1
ОГЛЯД ТЕХНОЛОГІЙ ТА ПЛАТФОРМ ДЛЯ ПОБУДОВИ NAS

Мережеве сховище даних (Network Attached Storage, NAS) - це спеціалізований пристрій, під'єднаний до локальної мережі та призначений для централізованого зберігання файлів і надання до них спільного доступу з різних клієнтських пристроїв. На відміну від звичайного зовнішнього диска, NAS працює автономно (без постійно ввімкненого комп'ютера), обслуговує кількох користувачів одночасно за мережевими протоколами та, як правило, надає додаткові сервіси - резервне копіювання, медіасервер, віддалений доступ, синхронізацію файлів. У цьому розділі розглянуто основні технології та платформи, що визначають побудову NAS: підходи до апаратної реалізації (готові комерційні рішення проти самостійного складання), файлові системи, мережеві протоколи доступу та архітектурні особливості обраної апаратної платформи - одноплатного комп'ютера Raspberry Pi 5. Огляд має на меті не лише систематизувати наявні рішення, а й обґрунтувати вибір, зроблений у практичній частині роботи.

Узагальнену структурну схему мережевого сховища наведено на рисунку 1.1: клієнтські пристрої звертаються через локальну мережу до сервера NAS, який засобами операційної системи та мережевих служб надає спільний доступ до даних, що зберігаються на накопичувачі.

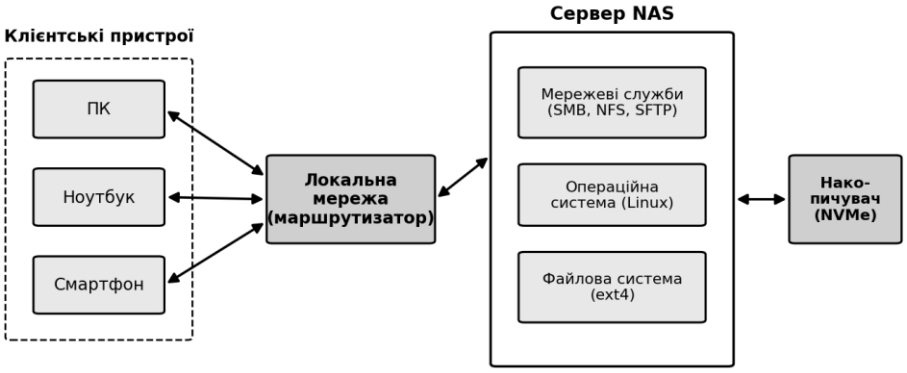


Рис. 1.1 - Узагальнена структурна схема NAS

1.1 Порівняльний аналіз комерційних NAS та DIY-рішень

За способом побудови всі NAS-рішення можна умовно поділити на дві групи: готові комерційні пристрої та системи, зібрані самостійно (DIY, Do It Yourself).

Готові комерційні NAS. Провідними виробниками цього сегмента є компанії Synology та QNAP. Їхні пристрої постачаються як завершений продукт: корпус із відсіками для дисків, власна операційна система (DSM у Synology, QTS у QNAP) з графічним вебінтерфейсом та екосистема готових застосунків. Сильні сторони такого підходу - простота введення в експлуатацію (налаштування «з коробки» за кілька кроків), апаратна підтримка кількох дисків з резервуванням даних (RAID-масиви), офіційна технічна підтримка та відлагоджене програмне забезпечення. Це робить комерційні NAS привабливими для користувачів, які цінують час і не бажають заглиблюватися в адміністрування.

Водночас цей підхід має й суттєві обмеження. По-перше, вартість: за орієнтовними роздрібними цінами станом на час виконання роботи навіть бюджетні дводискові пристрої на ARM-процесорі коштують близько 160-200 EUR без накопичувачів, моделі середнього рівня з процесором Intel - близько 280-320 EUR, а чотиривідсікові - від 400 до 650 EUR; до цих сум ще потрібно додати вартість самих дисків. По-друге, закритість екосистеми (vendor lock-in): користувач прив'язаний до програмного забезпечення виробника, можливості розширення обмежені офіційним каталогом застосунків, а апаратне оновлення (наприклад, заміна мережевого інтерфейсу) здебільшого неможливе. По-третє, у бюджетних моделях часто застосовують малопотужні процесори та обмежений обсяг оперативної пам'яті, що стримує продуктивність додаткових сервісів.

Самостійно зібрані (DIY) NAS. Альтернативою є побудова сховища на основі універсального обладнання - від повноцінного ПК до компактного одноплатного комп'ютера. Поява доступних одноплатних комп'ютерів, зокрема Raspberry Pi, зробила цей підхід особливо привабливим для

домашнього використання та малого офісу, про що свідчить і значна кількість присвяченої цій платформі літератури [1-5]. Переваги DIY-рішення - низька вартість, повний контроль над програмним і апаратним забезпеченням (можна встановити будь-яку ОС та будь-які сервіси), гнучкість конфігурації та цінний практичний досвід. Недоліки - потреба в самостійному складанні та налаштуванні, відсутність офіційної підтримки готового рішення й відповідальність за працездатність системи, яка лягає на користувача.

Обґрунтування вибору платформи. Для цієї роботи обрано DIY-підхід на базі Raspberry Pi 5. Такий вибір зумовлений сукупністю чинників: низькою вартістю (повний комплект разом із NVMe-накопичувачем коштує близько 216 EUR - порівнянно з ціною самого лише дводискового комерційного NAS середнього рівня, ще без дисків), повним контролем над системою, освітньою цінністю та - що принципово саме для Raspberry Pi 5 - появою апаратного інтерфейсу PCIe, який уперше для платформи Raspberry Pi дозволяє під'єднати швидкий накопичувач NVMe безпосередньо (детально архітектуру розглянуто в п. 1.4). Як показано в розділах 3 і 4, за ключовим для користувача показником - швидкістю обміну у гігабітній мережі - правильно зібране DIY-рішення не поступається комерційним пристроям того ж класу, оскільки всі вони обмежені однаковим гігабітним каналом.

1.2 Аналіз файлових систем

Файлова система визначає спосіб організації даних на накопичувачі та суттєво впливає на надійність, продуктивність і функціональні можливості сховища. Для Linux-систем, на яких базується більшість NAS, актуальними є чотири файлові системи: ext4, Btrfs, XFS та ZFS [6].

ext4 - четверте покоління розширеної файлової системи, стандартна для більшості дистрибутивів Linux. Це зріла, добре відлагоджена та надзвичайно стабільна система з журналюванням (journaling), яке захищає від пошкодження метаданих при раптовому вимкненні живлення. ext4

демонструє збалансовану продуктивність на широкому спектрі навантажень і має мінімальні вимоги до ресурсів. Її обмеження - відсутність вбудованих сучасних механізмів: контрольних сум даних (виявлення «тихого» псування), знімків стану (snapshots) та вбудованого керування томами.

Btrfs (B-tree File System) - сучасна файлова система з механізмом «копіювання при записі» (Copy-on-Write, CoW). Вона підтримує контрольні суми даних і метаданих, миттєві знімки стану, прозоре стиснення та вбудоване керування кількома пристроями. Це робить Btrfs привабливою для сценаріїв із підвищеними вимогами до цілісності даних. Платою за функціональність є вища складність і дещо нижча, ніж в ext4, продуктивність на окремих навантаженнях; історично деякі конфігурації Btrfs (зокрема певні рівні вбудованого RAID) тривалий час вважалися недостатньо надійними для промислового використання.

XFS - високопродуктивна журнальована файлова система, орієнтована на роботу з великими файлами та паралельний ввід-вивід завдяки механізму розподілу (allocation groups). XFS добре показує себе на потужних серверах із великими обсягами даних, проте має особливість: розмір тому можна лише збільшувати, але не зменшувати, що знижує гнучкість керування простором.

ZFS - комбінована файлова система й менеджер томів корпоративного класу. Вона забезпечує наскрізні контрольні суми, знімки, програмні RAID-масиви (RAID-Z), стиснення та механізми самовідновлення даних. ZFS вважається еталоном надійності, проте має дві істотні особливості, важливі для одноплатного комп'ютера: висока вимогливість до оперативної пам'яті (для ефективного кешування ARC рекомендують значні обсяги RAM) та позаядерний статус у Linux (через ліцензійну несумісність ZFS не входить до складу ядра й потребує окремого встановлення).

Обґрунтування вибору. Для розробленої системи обрано **ext4**. У сценарії одного накопичувача (без RAID-масиву) переваги Btrfs та ZFS у вигляді програмного резервування дисків не реалізуються, натомість вартими уваги залишаються стабільність, передбачуваність і низькі накладні витрати.

Висока вимогливість ZFS до пам'яті є небажаною для Raspberry Pi 5 з 8 ГБ RAM, частину якої вже споживають мережеві сервіси (Samba, NextCloud). ext4 ж є стандартною, найбільш перевіреною та найменш ресурсомісткою системою, що повністю відповідає завданню побудови надійного однодискового мережевого сховища. Незалежні порівняльні дослідження продуктивності файлових систем Linux підтверджують, що ext4 забезпечує збалансовані показники та одну з найнижчих затримок у типових файлових операціях [7].

1.3 Мережеві протоколи доступу до даних

Доступ клієнтів до файлів на NAS забезпечують мережеві протоколи спільного доступу. Найпоширеніші з них - SMB/CIFS, NFS та FTP/SFTP, кожен зі своєю сферою застосування.

SMB/CIFS (Server Message Block / Common Internet File System) - протокол спільного доступу до файлів, рідний для операційних систем Windows. У світі Linux його реалізує вільне програмне забезпечення Samba [8], що дозволяє Linux-серверу повноцінно обслуговувати клієнтів Windows, macOS, Linux, Android та iOS [9]. Сучасні версії протоколу (SMB2/SMB3) ефективні й безпечні, підтримують шифрування та автентифікацію. Завдяки універсальній підтримці на стороні клієнтів SMB є фактичним стандартом для гетерогенних мереж, де співіснують пристрої різних виробників.

NFS (Network File System) - протокол мережевого доступу до файлів, традиційний для Unix/Linux-середовищ. Він відзначається низькими накладними витратами та високою продуктивністю у власному середовищі, тісно інтегрується з моделлю прав доступу Unix. Проте за межами Unix-світу його застосування ускладнене: рідна підтримка у Windows обмежена, а класична модель автентифікації (на основі ідентифікаторів користувача) поступається SMB у зручності для домашніх змішаних мереж.

FTP/SFTP. FTP (File Transfer Protocol) - один із найстаріших протоколів передавання файлів; у первісному вигляді він небезпечний, бо

передає дані та паролі у відкритому вигляді. Сучасною захищеною альтернативою є **SFTP** (SSH File Transfer Protocol), що працює поверх захищеного з'єднання SSH і не потребує окремого серверного програмного забезпечення [10]. На відміну від SMB та NFS, які монтують мережеву теку як локальний диск для постійної роботи, SFTP орієнтований насамперед на разове захищене передавання файлів, зокрема через недовірені мережі.

Обґрунтування вибору. У практичній частині основним протоколом обрано **SMB (Samba)** - через універсальну підтримку всіма типами клієнтів, що робить сховище однаково доступним з ПК, смартфона чи планшета. Для захищеного віддаленого передавання файлів додатково задіяно **SFTP** (працює «з коробки» поверх SSH), а для безпечного доступу до всіх сервісів за межами домашньої мережі - технологію віртуальної приватної мережі (детально у розділі 2). Такий набір протоколів покриває як щоденну роботу в локальній мережі, так і безпечний віддалений доступ.

1.4 Архітектурні особливості Raspberry Pi 5

Raspberry Pi 5 - одноплатний комп'ютер, який за рахунок архітектурних нововведень уперше робить платформу Raspberry Pi по-справжньому придатною для побудови продуктивного NAS. Розглянемо ключові компоненти, що мають значення для цього застосування [11-14]. Загальну архітектуру платформи наведено на рисунку 1.2.

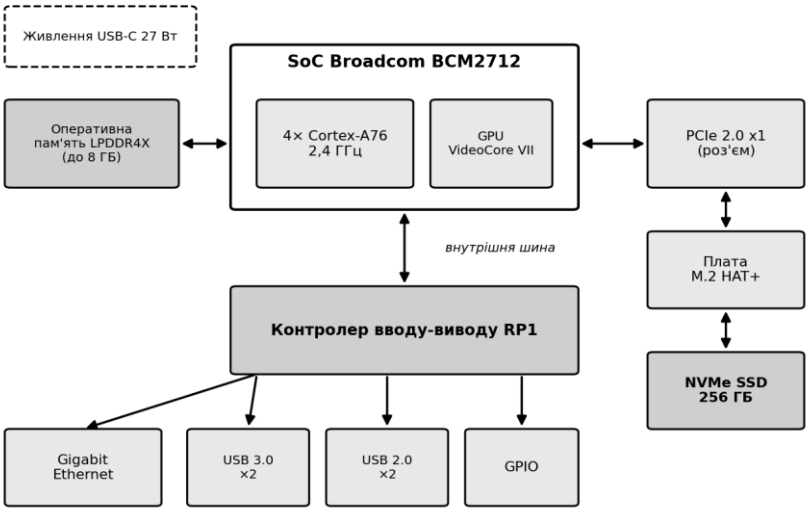


Рис. 1.2 - Структурна схема архітектури Raspberry Pi 5

Процесор та пам'ять. В основі Pi 5 лежить система на кристалі Broadcom BCM2712 з чотириядерним процесором Arm Cortex-A76 із тактовою частотою 2,4 ГГц - це значно потужніше за попереднє покоління. Обсяг оперативної пам'яті LPDDR4X сягає 8 ГБ, що дозволяє одночасно обслуговувати кілька мережевих сервісів (файловий сервер, вебзастосунки, бази даних) без браку ресурсів.

Контролер вводу-виводу RP1 та інтерфейс PCIe. Найважливішим нововведенням є власний південний міст - мікросхема RP1, розроблена Raspberry Pi, яка винесла периферію (USB 3.0, гігабітний Ethernet, інтерфейси GPIO) на окрему шину. Це усунуло вузьке місце попередніх моделей: на Raspberry Pi 4 усі USB 3.0-пристрої, зокрема накопичувач, поділяли єдиний контролер VL805, під'єднаний однією лінією PCIe Gen 2, а окремої зовнішньої шини PCIe не було взагалі, тому швидкий накопичувач доводилося під'єднувати через USB. На Pi 5 накопичувач (NVMe через зовнішню лінію PCIe) і периферія з мережею (через RP1) працюють незалежними шляхами й більше не конкурують за пропускну здатність. Найсуттєвіше для NAS - Pi 5 уперше виводить назовні інтерфейс **PCI Express** (одна лінія, офіційно Gen 2). Саме він через плату-адаптер M.2 HAT+ дозволяє під'єднати твердотільний накопичувач **NVMe** безпосередньо, а не через повільніший і менш надійний міст USB-SATA, як це доводилося робити на Raspberry Pi 4. Як показано в розділі 3, цей інтерфейс підтримує й режим Gen 3, що дає послідовну швидкість понад 800 МБ/с.

Мережа. Pi 5 оснащено вбудованим гігабітним адаптером Ethernet, під'єднаним до RP1 повноцінною смугою. Як експериментально підтверджено в розділі 4, цей інтерфейс досягає практичної межі гігабіта (близько 118 МБ/с), і саме він, а не накопичувач, є визначальним чинником швидкості мережевого обміну.

Живлення та тепловий режим. Зростання продуктивності супроводжується вищим енергоспоживанням. Для стабільної роботи з

периферією (зокрема з NVMe-накопичувачем) Pi 5 потребує блока живлення USB-C потужністю 27 Вт; за недостатнього живлення система обмежує струм на порти USB. Крім того, під навантаженням процесор помітно нагрівається, тому для цілодобової роботи NAS критично важливим є активне охолодження - без нього можливе зниження тактової частоти (тротлінг). У практичній частині застосовано офіційний активний кулер, що, як підтверджують вимірювання в розділі 4, повністю виключило тротлінг (температура не перевищувала ~50 °C під навантаженням).

Висновок щодо платформи. Сукупність цих особливостей - багатоядерний процесор, до 8 ГБ пам'яті, окремий контролер вводу-виводу та, головне, апаратний інтерфейс PCIe для NVMe - робить Raspberry Pi 5 повноцінною платформою для побудови продуктивного, компактного й енергоощадного NAS, чого були позбавлені попередні покоління.

Висновки

У розділі систематизовано технології та платформи для побудови мережевого сховища даних і обґрунтовано вибір, реалізований у практичній частині роботи.

1. **Підхід до реалізації.** Зіставлення готових комерційних NAS та самостійно зібраних рішень показало, що для завдань домашнього використання й малого офісу DIY-підхід на базі одноплатного комп'ютера виграє за вартістю, гнучкістю та контролем над системою, поступаючись комерційним пристроям лише у простоті початкового налаштування. Обрано платформу Raspberry Pi 5.

2. **Файлова система.** З-поміж ext4, Btrfs, XFS та ZFS для сценарію однодискового сховища на платформі з обмеженою пам'яттю обрано ext4 як найстабільнішу, найперевіренішу та найменш ресурсомістку систему; переваги CoW-систем (знімки, програмний RAID) у цьому сценарії не реалізуються.

3. **Мережеві протоколи.** Основним протоколом доступу обрано SMB (Samba) - за універсальну підтримку всіма клієнтськими платформами; для захищеного віддаленого передавання файлів додатково задіяно SFTP.

4. **Апаратна платформа.** Архітектура Raspberry Pi 5 - окремий контролер вводу-виводу RP1 та, насамперед, апаратний інтерфейс PCIe для прямого під'єднання NVMe - уперше робить цю платформу придатною для побудови продуктивного NAS, що й підтверджено результатами практичної частини.

Обґрунтований у цьому розділі вибір технологій лягає в основу програмно-апаратної реалізації NAS, описаної в розділі 2.

РОЗДІЛ 2
ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ NAS

У цьому розділі описано практичну реалізацію мережевого сховища даних на базі одноплатного комп'ютера Raspberry Pi 5: обґрунтовано вибір апаратних компонентів, розглянуто встановлення та конфігурацію операційної системи, налаштування дискової підсистеми, розгортання файлового сервісу Samba та організацію захищеного віддаленого доступу. Реалізацію виконано покроково - кожен етап супроводжується відповідними командами та поясненнями прийнятих рішень. Результатом розділу є повністю працездатна та відтворювана конфігурація NAS, готова до подальшої оптимізації (розділ 3) та експериментальних досліджень (розділ 4).

2.1 Обґрунтування вибору апаратного забезпечення

Основою апаратної платформи обрано одноплатний комп'ютер **Raspberry Pi 5 (8 ГБ RAM)** на базі чотириядерного процесора ARM Cortex-A76 з тактовою частотою 2,4 ГГц. Вирішальним для вибору саме цієї моделі став зовнішній інтерфейс **PCIe 2.0 x1**, виведений на окремий роз'єм, що дає змогу підключити NVMe-накопичувач напряму, а не через USB-міст; детально архітектурні переваги Pi 5 для задач NAS розглянуто в п. 1.4 [11].

Накопичувач: NVMe SSD через PCIe замість USB та SD-карти. Для зберігання даних обрано NVMe SSD **Silicon Power A60** ємністю 256 ГБ (форм-фактор M.2 2280, у системі визначається як SPCC M.2 PCIe SSD), підключений через плату розширення **M.2 HAT+** (адаптер PCIe - M.2). Сам накопичувач має рідний інтерфейс PCIe Gen 3 x4, проте на Raspberry Pi 5 виведено лише одну лінію PCIe (x1), тому паспортні швидкості диска (читання до ~2100 МБ/с [15]) тут недосяжні - реальну пропускну здатність обмежує саме x1-лінк платформи. Такий вибір обґрунтований двома групами аргументів:

- *Пропускна здатність.* За власними вимірюваннями (детально - розділ 3) реальна швидкість NVMe-накопичувача через PCIe 2.0 x1 на Pi 5 становить близько 439 МБ/с на запис і 455 МБ/с на читання, тоді як підключення диска по USB 3.0 на Pi 5 обмежене приблизно 420 МБ/с [12]. При активації режиму PCIe Gen 3 (розглянуто в розділі 3) пропускна здатність зростає ще приблизно вдвічі. У будь-якому випадку накопичувач із запасом перекриває пропускну здатність гігабітного Ethernet (~125 МБ/с), тому саме мережа, а не диск, є потенційним «вузьким місцем» системи.
- *Тип накопичувача (SSD проти HDD).* Вибір типу накопичувача залежить від пріоритетів користувача. Жорсткі диски (HDD) доцільні там, де потрібні великі обсяги даних і багатодискові масиви з резервуванням (RAID), оскільки дають дешевший гігабайт. У цій роботі пріоритетом є швидкість обміну за помірного обсягу даних, а резервування рівня RAID не ставилося за мету, тому свідомо обрано твердотільний накопичувач (SSD). Додатково SSD значно надійніший за SD-карту як носій системи: флеш-пам'ять SD-карт має суттєво менший ресурс перезапису і за інтенсивного журналювання та роботи swap швидко зношується, тоді як NVMe SSD розраховані на роки безперервної експлуатації; саме тому офіційна документація Raspberry Pi рекомендує для постійних навантажень використовувати твердотільний накопичувач [16]. Для пристрою, що працює в режимі 24/7, надійність зберігання є критичним чинником.

Блок живлення. Використано офіційний блок живлення **USB-C 27 Вт (5 В / 5 А)**. Це не надлишковість, а технічна необхідність: лише при виявленні офіційного блока живлення на 27 Вт Raspberry Pi 5 автоматично піднімає ліміт струму на USB-портах із 600 мА до 1,6 А, що потрібне для стабільного живлення периферії [13]. Дешеві адаптери на 3 А спричиняють просідання напруги (undervoltage) у момент, коли NVMe-контролер виходить із режиму сну, а це призводить до раптових відключень диска та

пошкодження файлової системи. Коректність живлення підтверджено командою `vcgencmd get_throttled`, яка повернула значення `0x0` - ознак тротлінгу чи недостатньої напруги не виявлено.

Охолодження. Pi 5 розміщено в металевому корпусі з **активним охолодженням** (вентилятор). Для NAS, що працює цілодобово, це обов'язкова умова: без належного охолодження процесор під тривалим навантаженням досягає теплового порогу і знижує тактову частоту (throttling), що зменшує продуктивність; під інтенсивним мережевим навантаженням Raspberry Pi 5 відомий схильністю до значного нагріву [17]. Окрім продуктивності, нижча робоча температура подовжує термін служби напівпровідникових компонентів. У спокої температура процесора (за власними вимірюваннями) становить близько 47 °C, що залишає достатній запас до порогу тротлінгу (~85 °C).

Зведений перелік апаратних компонентів стенду наведено в таблиці 2.1, а зовнішній вигляд зібраного пристрою - на рис. 2.1.

Таблиця 2.1

Апаратне забезпечення NAS

Компонент	**Характеристика / призначення**
Raspberry Pi 5 (8 ГБ RAM)	Обчислювальне ядро, ARM Cortex-A76 4×2,4 ГГц
Silicon Power A60 NVMe SSD 256 ГБ	Накопичувач даних, M.2 2280, PCIe Gen3 x4 (на Pi - x1)
Плата M.2 HAT+	Адаптер PCIe - M.2 для підключення SSD (PCIe x1)
Блок живлення USB-C 27 Вт	Стабільне живлення Pi 5 та NVMe (5 В / 5 А)
Активне охолодження (мет. корпус)	Запобігання тепловому дроселюванню при 24/7
Gigabit Ethernet (RJ-45)	Основний мережевий канал, до 1000 Мбіт/с



а - вид згори (M.2 HAT+ та NVMe SSD)



б - вид спереду (Ethernet та USB)

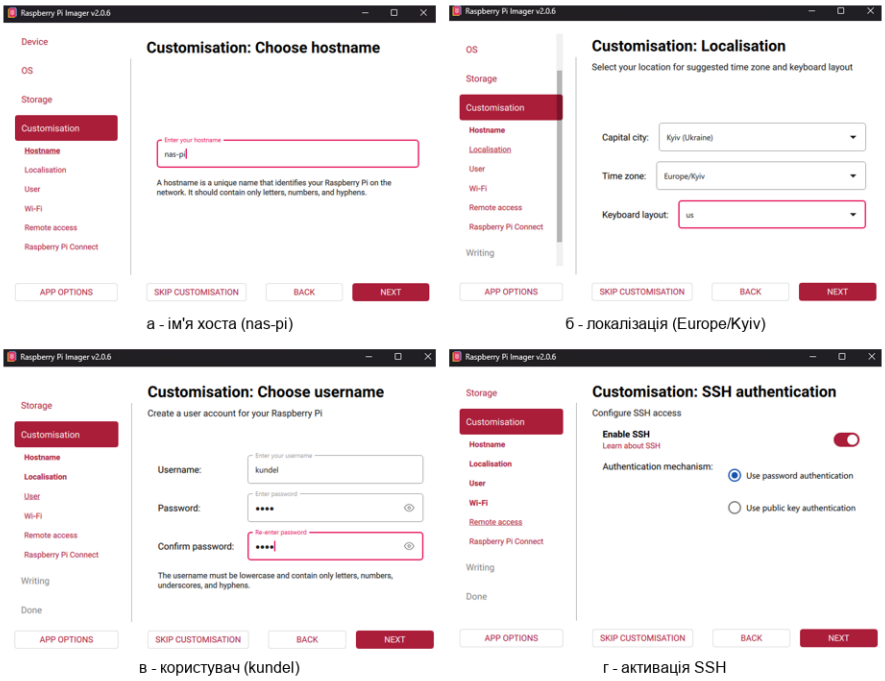
Рис. 2.1 - Зібраний апаратний стенд NAS у металевому корпусі: а - вид згори (плата M.2 HAT+ із встановленим NVMe-накопичувачем Silicon Power A60); б - вид спереду (порти Gigabit Ethernet та USB)

2.2 Конфігурація операційної системи та мережевого стека

Вибір операційної системи. Як ОС обрано **Raspberry Pi OS (64-bit)** на основі Debian - стандартний дистрибутив для цієї платформи. 64-розрядна версія потрібна, щоб система могла повноцінно задіяти всі 8 ГБ оперативної пам'яті та оптимально використати архітектуру ARM Cortex-A76. Збережено стандартне графічне середовище: це дає змогу за потреби адмініструвати пристрій локально, безпосередньо з підключеним монітором і клавіатурою (зручно, якщо віддалений доступ тимчасово недоступний, і простіше для менш досвідченого користувача), тоді як у штатному режимі сервер працює без монітора (headless) і керується по SSH.

Запис образу та попереднє налаштування. Образ записано на носій за допомогою офіційної утиліти **Raspberry Pi Imager** [18], яка дозволяє заздалегідь, ще до першого запуску, задати параметри системи (рис. 2.2):

- ім'я хоста (hostname) - `nas-pi`, завдяки чому пристрій доступний у локальній мережі за адресою `nas-pi.local`;
- обліковий запис користувача `kundel`;
- активацію служби SSH для віддаленого керування без монітора й клавіатури;
- часовий пояс `Europe/Kyiv` та розкладку клавіатури.



а - ім'я хоста (nas-pi)

б - локалізація (Europe/Kyiv)

в - користувач (kundel)

г - активація SSH

Рис. 2.2 - Попереднє налаштування системи в Raspberry Pi Imager (hostname, користувач, SSH, локалізація)

Завдяки попередньому налаштуванню Raspberry Pi 5 одразу після першого завантаження доступний по SSH, без потреби під'єднувати периферію.

Оновлення системи та прошивки. Після першого входу систему повністю оновлено, що, серед іншого, оновило ядро Linux до актуальної версії гілки `rpt-rpi-2712`:

```
sudo apt update && sudo apt full-upgrade -y
```

Лістинг 2.1 - Повне оновлення пакетів та ядра системи

Окремо оновлено мікропрограму завантажувача (EEPROM) - це важливо для стабільної роботи інтерфейсу PCIe з NVMe-накопичувачем:

```
sudo rpi-eeprom-update -a
sudo reboot
```

Лістинг 2.2 - Оновлення мікропрограми EEPROM

Перевірено коректність часового поясу та синхронізації часу (`timedatectl - Europe/Kyiv`, NTP активний - це необхідно для коректних відміток часу в системних журналах) та встановлено базовий набір службових утиліт:

```
sudo apt install htop nvme-cli gdisk -y
```

Лістинг 2.3 - Встановлення службових утиліт (моніторинг, керування NVMe, розмітка)

Налаштування мережевого стека. Архітектурно сервер NAS завжди підключений до маршрутизатора **кабелем по Gigabit Ethernet** (інтерфейс `eth0`) - це забезпечує стабільність і максимальну пропускну здатність, тому бездротовий інтерфейс на самому Pi не використовується (його вимкнено як рішення на користь стабільності та безпеки; тестування Wi-Fi у розділі 4 виконується з боку клієнта). Щоб адреса сервера не змінювалася після перезавантаження маршрутизатора, інтерфейсу `eth0` призначено **статичну IP-адресу** засобами NetworkManager (утиліта `nmcli` [19]):

```
sudo nmcli con mod "Wired connection 1" \
  ipv4.addresses 192.168.3.101/24 \
  ipv4.gateway 192.168.3.1 \
  ipv4.dns "192.168.3.1,8.8.8.8" \
  ipv4.method manual
sudo nmcli con up "Wired connection 1"
```

Лістинг 2.4 - Призначення статичної IP-адреси інтерфейсу eth0

Після цього сервер має фіксовану адресу **192.168.3.101** зі шлюзом 192.168.3.1 та двома DNS-серверами - локальним маршрутизатором і публічним Google DNS (8.8.8.8). Профіль зберігається в `/etc/NetworkManager/system-connections/` і переживає перезавантаження.

Узагальнену схему мережевої взаємодії побудованої системи наведено на рисунку 2.3. У локальній мережі (підмережа 192.168.3.0/24) сервер NAS під'єднано до маршрутизатора кабелем (Gigabit Ethernet, адреса 192.168.3.101), а клієнтські пристрої (ПК, ноутбук, смартфон) - кабелем або по Wi-Fi. Доступ до даних надається службами Samba (порт 445), SFTP (порт 22) та вебхмарою NextCloud (порт 80). Для віддаленого доступу (коли клієнт перебуває поза домашньою мережею) використовується захищений канал Tailscale (порт 41641), що буде розглянуто в п. 2.5.

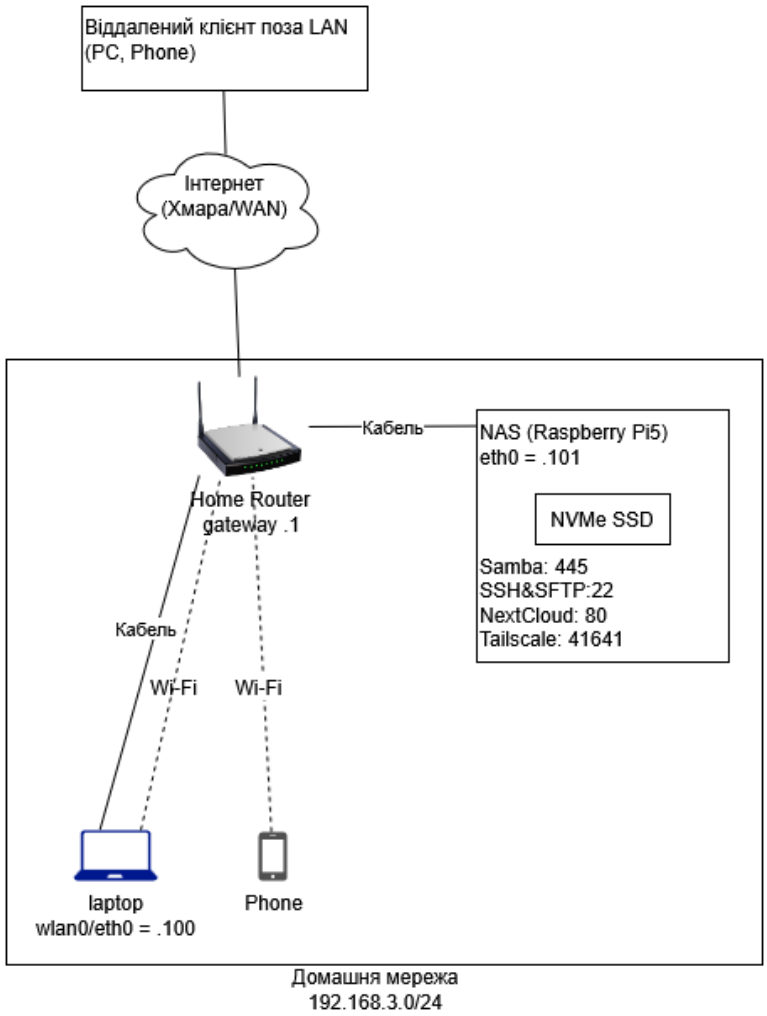


Рис. 2.3 - Структурна схема мережевої взаємодії NAS (локальний доступ по Ethernet/Wi-Fi та віддалений через VPN Tailscale)

2.3 Налаштування дискової підсистеми та міграція системи на NVMe

Архітектурне рішення: завантаження з NVMe. На відміну від класичної схеми, де операційна система розміщується на SD-карті, а NVMe слугує лише сховищем даних, у цій роботі обрано продуктивніший підхід - **повне перенесення ОС на NVMe** з подальшим завантаженням саме з нього; саме таку архітектуру рекомендують і сучасні практичні посібники з побудови NAS на Raspberry Pi 5 [20]. Обґрунтування вже наведено в п. 2.1: SD-карти мають суттєво менший ресурс перезапису і за інтенсивного серверного навантаження швидко зношуються, тоді як NVMe забезпечує і вищу швидкість (439-455 МБ/с за власними вимірюваннями), і кращу

цілісність даних під тривалим записом. Для NAS, де надійність зберігання є визначальною, перевага однозначно на боці NVMe.

Метод міграції. Систему спершу встановлено та налаштовано на SD-карту, після чого вже сконфігуровану ОС перенесено на NVMe методом клонування за допомогою утиліти `rpi-clone`. Альтернативою було б чисте перевстановлення ОС безпосередньо на NVMe, проте це вимагало б повторного налаштування всіх служб, прав доступу та конфігурацій. Клонування ж перенесло вже готову систему за ~2,5 хвилини зі збереженням усіх налаштувань, автоматично переписавши ідентифікатори розділів (PARTUUID) у `cmdline.txt` та `/etc/fstab`:

```
sudo rpi-clone nvme0n1
```

Лістинг 2.5 - Клонування налаштованої системи з SD-карти на NVMe

Порядок завантаження. Через утиліту `raspi-config` (Advanced Options → Boot Order) встановлено пріоритет завантаження `0xf146` - спочатку NVMe, потім USB, і лише як резерв (fallback) - SD-карта. Нове значення прошито в EEPROM. Після цього SD-карту вилучено, і пристрій працює повністю автономно з NVMe.

Підсумкова розмітка диска. У фінальному стані NVMe містить два розділи: невеликий завантажувальний розділ `/boot/firmware` та основний розділ із кореневою файловою системою, розширеною на весь доступний обсяг. Каталог даних `/mnt/nas` розміщується на кореневій ФС. Структуру підтверджено командами `lsblk` та `df`:

```
$ lsblk
NAME        SIZE MOUNTPOINT
nvme0n1   238.5G
├─nvme0n1p1 512M /boot/firmware
└─nvme0n1p2  238G /

$ df -h /
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme0n1p2  234G  8.1G  214G   4% /
```

Лістинг 2.6 - Підсумкова розмітка NVMe-накопичувача (root + завантажувальний розділ)

Кореневий розділ відформатовано у файлову систему **ext4**. Її обрано не випадково: для NAS з одним накопичувачем ext4 є оптимальним компромісом між швидкістю та простотою. За результатами порівняльних тестів файлових систем ext4 демонструє низьку затримку при роботі з метаданими та мінімально споживає ресурси CPU і RAM [7], що особливо важливо для одноплатного комп'ютера. Просунуті файлові системи Btrfs і ZFS у цьому сценарії надлишкові: механізм Copy-on-Write у Btrfs створює додаткове навантаження на процесор через постійне обчислення контрольних сум і деградує при зростанні кількості клієнтів, а ZFS потребує значного обсягу оперативної пам'яті для кешування, що конкурує з іншими службами (Samba, NextCloud). Для масивів із кількох дисків їхні переваги були б доречні, але для одного накопичувача - ні.

Каталог даних і права доступу. Дані користувача зберігаються в каталозі `/mnt/nas`. Оскільки після переходу на завантаження з NVMe цей каталог є частиною кореневої ФС (а не окремим розділом), для нього застосовано **bind mount** із захисними опціями монтування - це забороняє виконання файлів, інтерпретацію спецпристроїв та біт SUID у каталозі даних (захист на випадок завантаження зловмисного вмісту через спільний мережевий ресурс). Відповідний рядок у `/etc/fstab`:

```
/mnt/nas  /mnt/nas  none  bind,noexec,nodev,nosuid  0 0
```

Лістинг 2.7 - Bind mount каталогу даних із захисними опціями (noexec, nodev, nosuid)

Окрім того, на всю кореневу ФС діє опція `noatime`, яка вимикає запис часу останнього доступу до файлів - це зменшує кількість зайвих операцій запису і продовжує ресурс SSD. Власником каталогу `/mnt/nas` призначено користувача `kundel` із правами `770`.

Вирішення проблеми енергозбереження NVMe (APST). Під час налаштування виявлено характерну для бюджетних NVMe-накопичувачів несправність: приблизно через 50 секунд після завантаження контролер

диска переходив у режим енергозбереження і вже не відновлювався. У системному журналі (dmesg) при цьому з'являлися повідомлення (рис. 2.4):

```
nvme nvme0: controller is down; will reset: CSTS=0xffffffff, PCI_STATUS=0x10
nvme nvme0: Does your device have a faulty power saving mode enabled?
```

Лістинг 2.8 - Збій NVMe-контролера через механізм APST (фрагмент dmesg)

Наслідком був перехід файлової системи в режим «тільки читання» з помилками `I/O error` при будь-якій спробі запису та повна недоступність сховища. Причина - механізм **APST (Autonomous Power State Transition)**: накопичувач самостійно переходив у глибокий стан енергозбереження, з якого контролер не «прокидався». Це відома проблема сумісності окремих NVMe-накопичувачів із механізмом APST, типовим розв'язанням якої є обмеження енергостанів через параметр ядра `nvme_core.default_ps_max_latency_us` [21]. Проблему усунено забороною переходів NVMe у стани зниженого енергоспоживання - у єдиний рядок файлу `/boot/firmware/cmdline.txt` додано параметр ядра:

```
nvme_core.default_ps_max_latency_us=0
```

Лістинг 2.9 - Вимкнення енергозберігаючих станів NVMe в параметрах ядра

Після перезавантаження накопичувач працює стабільно, запис виконується без помилок. Розширений фрагмент журналу ядра з цим збоєм наведено в додатку А. Цей випадок є також показовою ілюстрацією нестабільності PCIe-лінка бюджетних SSD і безпосередньо пов'язаний із оптимізацією дискової підсистеми, розглянутою в розділі 3 (де, зокрема, виконано перехід на режим PCIe Gen 3).

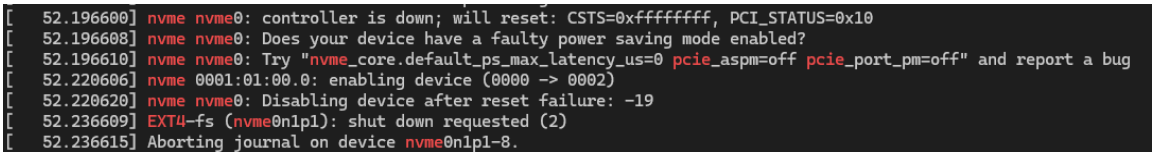


Рис. 2.4 - Повідомлення dmesg про збій NVMe-контролера до застосування виправлення

2.4 Розгортання та налаштування сервісу Samba

Для надання мережевого доступу до файлів обрано **Samba** - вільну реалізацію протоколу SMB/CIFS [8, 22]. Вибір на користь SMB обґрунтований кількома чинниками:

- SMB є стандартним протоколом файлового доступу у Windows-мережах: ресурс підключається як звичайний мережевий диск без встановлення додаткового ПЗ на боці клієнта;
- протокол кросплатформний - підтримується Windows, macOS, Linux та мобільними платформами;
- автентифікація за логіном і паролем забезпечує базовий рівень контролю доступу;
- усі провідні комерційні NAS-системи (Synology, QNAP, TrueNAS) використовують SMB як основний протокол [23].

Для порівняння, NFS вимагає окремого клієнта на Windows, а FTP/SFTP не дозволяють підключити ресурс як мережевий диск, тож для кінцевого користувача саме Samba є найзручнішим варіантом.

Встановлення. Samba (версія 4.22.8) встановлюється однією командою:

```
sudo apt install samba samba-common-bin -y
```

Лістинг 2.10 - Встановлення сервісу Samba

Права доступу. Власником каталогу спільного доступу призначено користувача `kundel`, права обмежено для власника й групи:

```
sudo chown -R kundel:kundel /mnt/nas
sudo chmod -R 770 /mnt/nas
```

Лістинг 2.11 - Налаштування прав доступу до каталогу спільного ресурсу

Конфігурація. Перед редагуванням створено резервну копію конфігураційного файлу (`/etc/samba/smb.conf.backup`). До файлу `/etc/samba/smb.conf` додано опис спільного ресурсу `[nas]` із обов'язковою автентифікацією:

```
[nas]
   path = /mnt/nas
```

```
read only = no
valid users = kundel
create mask = 0660
directory mask = 0770
```

Лістинг 2.12 - Опис спільного ресурсу [nas] у smb.conf

Параметр `valid users = kundel` обмежує доступ єдиним обліковим записом; маски `create mask = 0660` та `directory mask = 0770` задають права на новостворювані файли й каталоги. У глобальній секції також вимкнено застарілі компоненти заради безпеки (детальніше - у п. 2.5): додано `disable netbios = yes` та `smb ports = 445`, що залишає лише сучасний транспорт SMB2/3 через порт 445, а також закоментовано стандартну секцію `[homes]`, щоб на пристроях був видимий тільки ресурс `nas`. Коректність конфігурації перевірено утилітою `testparm`; повну підсумкову конфігурацію Samba наведено в додатку Б.

Обліковий запис Samba та запуск. Для користувача задано окремий пароль Samba, після чого службу перезапущено й додано в автозавантаження:

```
sudo smbpasswd -a kundel
sudo systemctl restart smbd
```

Лістинг 2.13 - Створення облікового запису Samba та перезапуск служби

Перевірка з боку клієнта. Працездатність спільного ресурсу перевірено з ПК під Windows підключенням мережевого диска (рис. 2.5). Читання та запис файлів виконуються коректно:

```
net use Z: \\192.168.3.101\nas /user:kundel
```

Лістинг 2.14 - Підключення мережевого диска з Windows

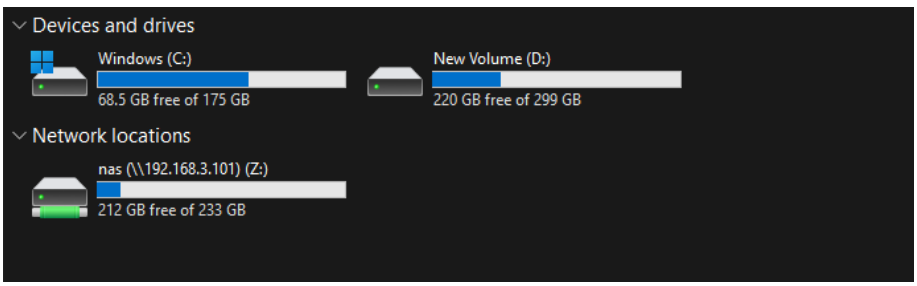


Рис. 2.5 - Спільний ресурс NAS, підключений як мережевий диск у Windows

2.5 Організація віддаленого доступу та захисту

Окрім доступу в межах локальної мережі, NAS налаштовано для безпечного використання ззовні. Віддалений доступ реалізовано як окремий функціональний блок із трьома способами підключення (VPN-мережа, SFTP та приватна хмара) і комплексом заходів із захисту сервера.

2.5.1 Захищений канал Tailscale

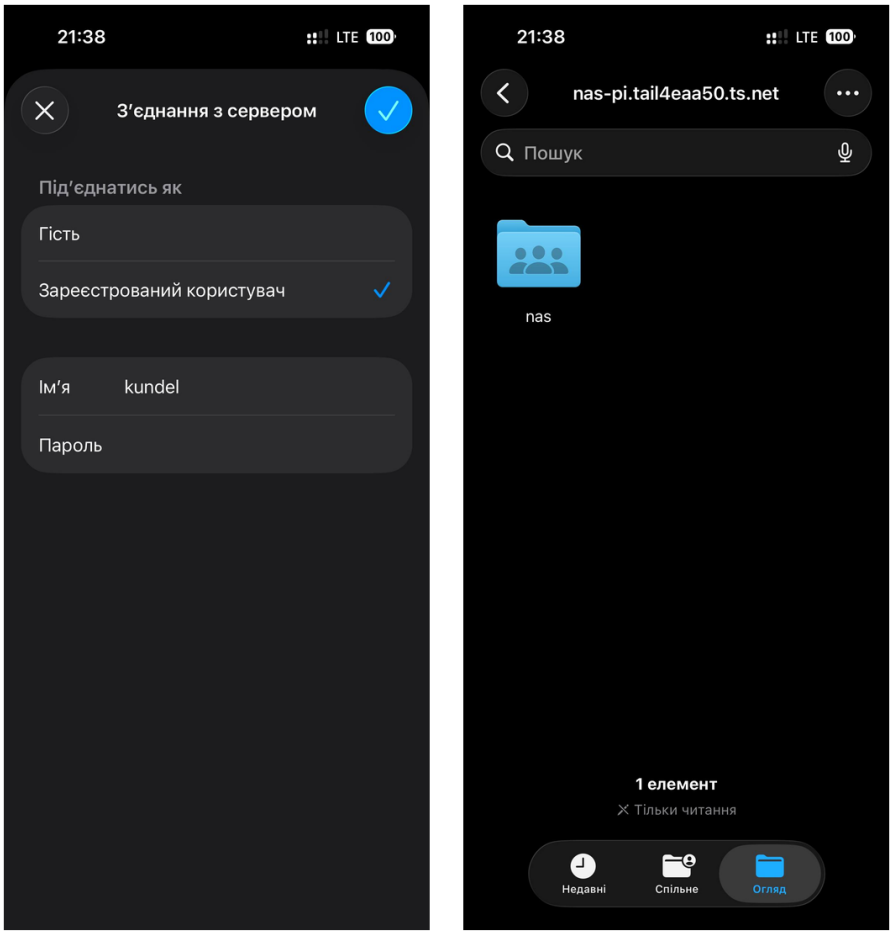
Для доступу до NAS із будь-якої зовнішньої мережі без прокидання портів на маршрутизаторі використано **Tailscale** - VPN-сервіс на базі протоколу **WireGuard**, що будує захищену однорангову (mesh) мережу. Пристрої з'єднуються напряму (peer-to-peer), а сервіс автоматично обходить NAT та брандмауери, тому **відкривати порти на роутері не потрібно**. Це принципово безпечніше за класичний port-forwarding: сервер залишається невидимим із публічного інтернету, що унеможливлює його сканування та зовнішні атаки. Додатково механізм **MagicDNS** надає пристрою постійне ім'я виду `nas-pi.tailnet.ts.net`, тож звертатися до NAS можна без знання IP-адреси [24].

Tailscale (версія 1.98.4) встановлено офіційним скриптом, після чого виконано вхід:

```
curl -fsSL https://tailscale.com/install.sh | sh
sudo tailscale up
```

Лістинг 2.15 - Встановлення та активація Tailscale

Сервер отримав адресу в приватній мережі Tailscale **100.92.140.23** та MagicDNS-ім'я `nas-pi.tail4eaa50.ts.net`. Перевірено доступ із зовнішньої мережі: SSH та мережева спільний ресурс Samba доступні через VPN із будь-якого підключення без додаткових налаштувань маршрутизатора. Зокрема, перевірено підключення до ресурсу SMB зі смартфона (iOS) по мобільному інтернету за MagicDNS-іменем (рис. 2.6).



а - автентифікація (kundel) б - відкритий ресурс nas

Рис. 2.6 - Підключення до спільного ресурсу NAS зі смартфона через Tailscale (MagicDNS) по мобільному інтернету: а - автентифікація користувача; б - відкритий ресурс nas

2.5.2 Передавання файлів через SFTP

Як простий захищений спосіб передавання файлів використано **SFTP** - протокол, що працює поверх SSH. Його перевага в тому, що він доступний одразу після ввімкнення SSH **без встановлення додаткового серверного ПЗ**, забезпечує повне шифрування переданих даних і команд та використовує ті самі облікові дані, що й SSH [10]. Працездатність перевірено з ПК під Windows (передавання файлів у обидва боки) та через графічний клієнт FileZilla (хост 192.168.3.101, порт 22, протокол SFTP; рис. 2.7):

```
sftp kundel@192.168.3.101
```

Лістинг 2.16 - Підключення до NAS за протоколом SFTP

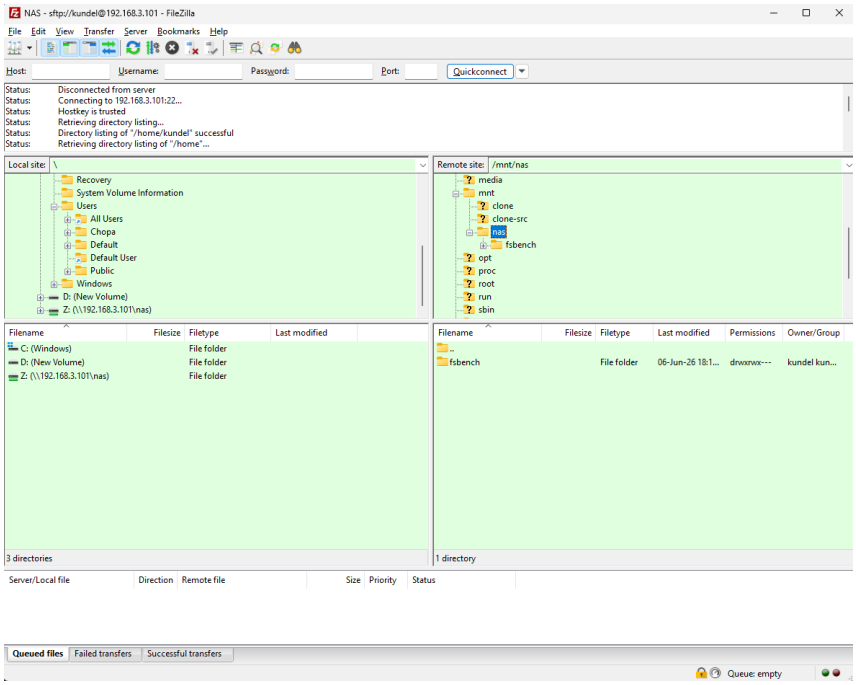


Рис. 2.7 - Підключення до NAS через клієнт FileZilla за протоколом SFTP

2.5.3 Приватна хмара NextCloud

Для зручного доступу до файлів через браузер і мобільні застосунки розгорнуто **NextCloud** - платформу приватного хмарного сховища (аналог iCloud чи OneDrive, але з повним контролем над власними даними) [25]. NextCloud (версія 32.0.9) встановлено як snap-пакет, який включає вебсервер, систему керування базами даних MySQL та кеш Redis. Адміністративний обліковий запис створено через консольну утиліту, а в перелік довірених доменів (`trusted_domains`) додано локальну, Tailscale- та MagicDNS-адреси сервера.

Під час розгортання виявлено несумісність, пов'язану зі специфікою архітектури Pi 5: служба Redis аварійно завершувалася з помилкою `<jemalloc>: Unsupported system page size`. Причина - стандартне ядро Pi 5 (`rpi-2712`) використовує розмір сторінки пам'яті 16 КБ, тоді як аллокатор jemalloc у складі snap-пакета зібрано під 4 КБ. Проблему усунено перемиканням на сумісне 4-кілобайтне ядро через `config.txt`:

```
kernel=kernel8.img
```

Лістинг 2.17 - Перемикання на 4-КБ ядро для сумісності з NextCloud (Redis/jemalloc)

Після цього всі служби NextCloud (Apache, MySQL, PHP-FPM, Redis) працюють штатно, а вебінтерфейс доступний за адресою сервера (рис. 2.8).

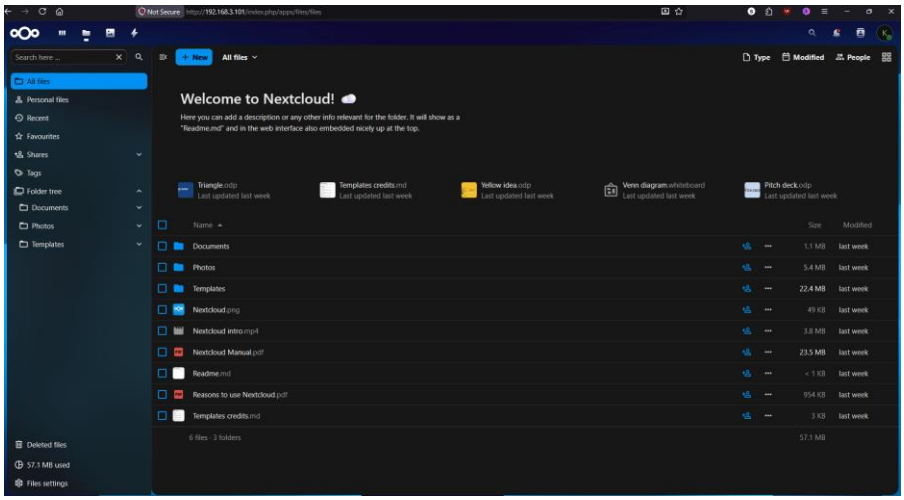


Рис. 2.8 - Вебінтерфейс приватної хмари NextCloud

2.5.4 Посилення захисту SSH (hardening)

Оскільки SSH є основним каналом керування сервером, його захист посилено переходом від парольної автентифікації до входу **виключно за криптографічним ключем**. Парольний вхід вразливий до атак перебором (brute-force), тоді як криптографічний ключ ed25519 має незрівнянно вищу стійкість, а повне вимкнення парольної автентифікації унеможливлює підбір облікових даних - найпоширеніший вектор автоматизованих атак на сервери, доступні в мережі [10].

На клієнтському ПК згенеровано ключ сучасного типу **ed25519**, а публічну частину додано до `~/.ssh/authorized_keys` на сервері. Оскільки утиліта Raspberry Pi Imager залишила drop-in `50-cloud-init.conf` з `PasswordAuthentication yes`, створено пріоритетний файл `/etc/ssh/sshd_config.d/00-hardening.conf` (префікс `00` гарантує зчитування раніше за `50`) з такими параметрами:

```
PasswordAuthentication no
KbdInteractiveAuthentication no
PubkeyAuthentication yes
DebianBanner no
```

Лістинг 2.18 - Drop-in конфігурація посилення захисту SSH

Перевірено: вхід за ключем виконується успішно, а спроба парольного входу відхиляється з повідомленням `Permission denied (publickey)`. Параметр `DebianBanner no` додатково приховує суфікс версії Debian у банері SSH від автоматичних сканерів.

2.5.5 Скорочення поверхні атаки

Щоб мінімізувати кількість потенційних точок входу, на сервері вимкнено невикористовувані мережеві служби:

- **rpcbind** (порт 111) - portmapper для RPC/NFS. Оскільки NFS не використовується (лише Samba), службу зупинено й замасковано. Це усуває класичну мішень для атак та вектор DDoS-ампліфікації.
- **NetBIOS** (порти 137/138/139, застарілий SMB1) - вимкнено службу `nmbd` та явно обмежено Samba сучасним транспортом (`smb ports = 445`).
- **Гостьовий (анонімний) доступ SMB** - заборонено повністю. У стандартній конфігурації Debian діє параметр `map to guest = bad user`, який перетворює невдалу спробу входу з неіснуючим іменем користувача на гостьовий сеанс. Цей рядок вилучено, а в секцію `[global]` додано явну заборону гостьових сеансів та анонімних підключень до службового ресурсу IPC$ (через які сканер міг би без автентифікації отримати перелік ресурсів сервера):

```
map to guest = never
restrict anonymous = 2
```

Лістинг 2.19 - Заборона гостьового та анонімного доступу SMB (smb.conf, секція [global])

Дієвість заборони перевірено з клієнта: спроба під'єднатися до сервера в режимі «гість» (без облікових даних) завершується помилкою автентифікації, тоді як вхід користувача `kundel` із паролем працює штатно.

Підсумковий стан відкритих портів перевірено командою `ss -tulnp` - у системі залишилися лише необхідні служби: SSH (22), Samba (445),

вебсервер NextCloud (80), Tailscale та локальний mDNS (avahi). Порти 111, 137, 138, 139 зникли.

```
$ ss -tulnp
tcp  LISTEN  0.0.0.0:22      (sshd)
tcp  LISTEN  0.0.0.0:445     (smbd)
tcp  LISTEN  0.0.0.0:80      (apache/nextcloud)
udp  LISTEN  *:5353          (avahi-daemon)
```

Лістинг 2.20 - Перелік відкритих мережевих портів після скорочення поверхні атаки

Щодо програмного брандмауера. Окремий міжмережевий екран (ufw) на цьому етапі не впроваджувався свідомо: захист від зовнішнього інтернету забезпечується самою архітектурою - на маршрутизаторі немає прокидання портів, а єдиний зовнішній доступ здійснюється через зашифрований тунель Tailscale. За потреби програмний брандмауер може бути доданий як ешелонований захист (defense-in-depth) для ізоляції від ненадійних пристроїв усередині локальної мережі.

Висновки

У розділі реалізовано повнофункціональне мережеве сховище даних на базі Raspberry Pi 5. Обґрунтовано вибір апаратної платформи: підключення NVMe SSD через PCIe 2.0 (M.2 HAT+) забезпечує вищу пропускну здатність, ніж USB 3.0, та значно кращу надійність зберігання порівняно зі SD-картою, а офіційний блок живлення 27 Вт і активне охолодження гарантують стабільну роботу в режимі 24/7.

Систему побудовано за продуктивнішою архітектурою, ніж класична: операційну систему повністю перенесено на NVMe (завантаження з NVMe), що усуває знос SD-карти як системного носія. Налаштовано дискову підсистему на ext4 із захисним bind mount каталогу даних, вирішено критичну проблему енергозбереження NVMe-контролера (APST) через параметр ядра. Розгорнуто файловий сервіс Samba з автентифікацією та обмеженням доступу, а також організовано три способи захищеного віддаленого доступу: VPN-мережу Tailscale (без прокидання портів), SFTP та

приватну хмару NextCloud. Окрему увагу приділено безпеці - виконано посилення SSH (вхід лише за ключем ed25519) та скорочено поверхню атаки вимкненням зайвих служб і повною забороною гостьового (анонімного) доступу SMB.

Отримана конфігурація є стабільною та відтворюваною (підтверджено серією контрольних перезавантажень) і слугує основою для оптимізації продуктивності, розглянутої в розділі 3.

РОЗДІЛ 3
ОПТИМІЗАЦІЯ ПРОДУКТИВНОСТІ NAS-СИСТЕМИ

У попередньому розділі побудовано працездатне мережеве сховище. Цей розділ присвячено підвищенню його продуктивності. Оптимізацію виконано пошарово - від мережевого стека ядра до дискової підсистеми та параметрів файлового сервісу, - причому ефект кожної зміни вимірювався експериментально за схемою «до/після». Такий підхід дозволяє не лише застосувати рекомендовані налаштування, а й об'єктивно встановити, які з них дають реальний приріст для конкретної конфігурації, а отже - виявити справжнє «вузьке місце» системи.

Методика вимірювань. Для оцінювання використано три інструменти: `fio` - для локальної швидкості NVMe (послідовний доступ блоком 1 МБ та випадковий 4 КБ), `iperf3` - для «чистої» пропускної здатності мережевого каналу, та копіювання файлу 4 ГБ засобами клієнта - для реальної швидкості Samba. Кожен замір повторювався, перший (прогрівний) прогін відкидався, а для решти обчислювалися середнє значення, стандартне відхилення σ та коефіцієнт варіації CV = σ/середнє. У всіх дискових і мережевих тестах CV не перевищував кількох відсотків, що свідчить про стабільність і відтворюваність результатів. Повний опис стенду та методики наведено в п. 4.1.

Усі мережеві вимірювання цього розділу виконано з клієнтом, підключеним до маршрутизатора **кабелем (Gigabit Ethernet)** - щоб бачити стелю гігабітного каналу, а не обмеження бездротового з'єднання.

3.1 Оптимізація мережевого стека ядра Linux

Базова пропускна здатність. Перш ніж змінювати параметри, зафіксовано відправну точку. Тест `iperf3` [26] показав, що мережевий канал працює на практичній стелі Gigabit Ethernet - **949 Мбіт/с** у напрямку «клієнт → NAS» і **945 Мбіт/с** у зворотному, причому лічильник повторних передач (Retr) дорівнював нулю, що вказує на ідеально чистий канал:

```
$ iperf3 -c 192.168.3.101 -t 60
[ ID] Interval        Transfer     Bitrate         Retr
[  5] 0.00-60.01 sec  6.64 GBytes  950 Mbits/sec    0    sender
[  5] 0.00-60.04 sec  6.63 GBytes  949 Mbits/sec         receiver
```

Лістинг 3.1 - Базова пропускна здатність каналу (iperf3, ≈949 Мбіт/с, Retr=0)

Реальна швидкість файлового обміну через Samba становила 110,8 МБ/с на запис (≈93 % від теоретичної межі гігабіта), тобто система вже близька до стелі каналу.

Налаштування TCP-буферів. Стандартні розміри буферів сокетів у Linux розраховані на універсальний сценарій. Для каналів з великим добутком «пропускна здатність × затримка» (BDP) їх збільшення дозволяє TCP тримати в польоті більше даних і повніше використовувати канал. До файлу `/etc/sysctl.d/99-nas.conf` додано збільшені до 16 МБ буфери та діапазони автотюнінгу TCP:

```
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
net.ipv4.tcp_window_scaling = 1
```

Лістинг 3.2 - Збільшення TCP-буферів ядра (sysctl)

Параметри застосовано командою `sudo sysctl --system`. Повторні вимірювання після цього **не виявили приросту** - швидкість Samba залишилася 110,8 МБ/с. Причина закономірна: у локальній мережі затримка вкрай мала (RTT < 1 мс за даними `ping`), тому добуток BDP становить лише близько 120 КБ, і стандартного вікна цілком достатньо для насичення гігабіта. Збільшені буфери дали б ефект на каналах із великою затримкою (наприклад, WAN) або вищою пропускною здатністю (10 Gbit/s), але на гігабітному LAN канал і так використовується повністю. Налаштування залишено в системі як коректне й нешкідливе, що стане в пригоді при майбутньому переході на швидші мережі.

3.2 Налаштування IRQ Affinity та CPU Governor

Розподіл переривань (IRQ Affinity). Обробка апаратних переривань мережевої карти та накопичувача навантажує процесор. Якщо всі

переривання обробляє одне ядро, воно може стати вузьким місцем («голодування CPU0»); рознесення переривань по ядрах є відомим прийомом оптимізації продуктивності серверів [27]. Аналіз `/proc/interrupts` виявив цікаву картину:

```
$ grep -iE "nvme|eth0" /proc/interrupts
 40:  547244       0       0       0   nvme0q1
 41:       0  626209       0       0   nvme0q2
 42:       0       0  645547       0   nvme0q3
 43:       0       0       0  705345   nvme0q4
112: 11438215       0       0       0   eth0
```

Лістинг 3.3 - Розподіл переривань NVMe та Ethernet за ядрами CPU

Накопичувач NVMe **вже оптимально розподілений ядром автоматично**: він використовує чотири апаратні черги (`nvme0q1-nvme0q4`), кожну з яких прив'язано до окремого ядра (CPU0-CPU3). Це механізм керованих (managed) переривань сучасного ядра Linux, тож ручне втручання тут не потрібне і не дасть покращення. Натомість усі переривання Ethernet (`eth0`, IRQ 112) оброблялися виключно ядром CPU0. Їх перенесено на окреме ядро CPU1:

```
echo 2 | sudo tee /proc/irq/112/smp_affinity
```

Лістинг 3.4 - Прив'язка переривань Ethernet до ядра CPU1

Повторні вимірювання **приросту не показали** - швидкість Samba залишилась незмінною. CPU0 не був вузьким місцем: чотири ядра Cortex-A76 на 2,4 ГГц із запасом справляються з потоком переривань гігабітного каналу, не насичуючись. Отже, для цієї системи ручний розподіл IRQ зайвий, а зміна до того ж не зберігається після перезавантаження, тому у фінальній конфігурації залишено типову (керовану ядром) поведінку.

Режим процесора (CPU Governor). За замовчуванням використовується регулятор `ondemand`, що динамічно змінює частоту залежно від навантаження. Для серверів іноді рекомендують режим `performance`, який фіксує частоту на максимумі й усуває мікрозатримки на «розгін» частоти:

```
echo performance | sudo tee /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
```

Лістинг 3.5 - Перемикання регулятора частоти CPU у режим performance

Частоту зафіксовано на 2,4 ГГц (підтверджено `vcgencmd measure_clock arm`). Однак на пропускну здатність великих файлів це **також не вплинуло** - і це логічно, адже система обмежена мережею, а не швидкодією CPU, який уже насичує гігабіт навіть у режимі `ondemand`. Режим `performance` потенційно корисний для зменшення затримок при роботі з великою кількістю дрібних файлів (це досліджується в п. 4.4), проте для безперервної роботи NAS режим `ondemand` є енергетично ощадливішим, тому як основний обрано саме його.

3.3 Активація PCIe Gen 3 та налаштування параметрів NVMe

Найбільший потенціал приросту виявився на рівні дискової підсистеми. Інтерфейс M.2 HAT+ на Raspberry Pi 5 за замовчуванням працює в режимі PCIe Gen 2 (5,0 ГТ/с), хоча і контролер накопичувача, і сам SoC підтримують Gen 3 (8,0 ГТ/с). Поточний режим підтверджено командою `lspci`:

```
$ lspci -vv | grep LnkSta
LnkSta: Speed 5GT/s (downgraded), Width x1 (downgraded)
```

Лістинг 3.6 - Початковий стан PCIe-лінка: Gen 2 (5 ГТ/с)

Базова швидкість (Gen 2). Вимірювання `fio` дало послідовну швидкість близько 439 МБ/с на запис і 455 МБ/с на читання - це відповідає теоретичній межі PCIe Gen 2 x1 (~500 МБ/с). Стабільність висока: коефіцієнт варіації менший за 1 %.

Активація Gen 3. Режим Gen 3 на зовнішньому роз'ємі Raspberry Pi 5 офіційно не сертифікований через можливі проблеми з цілісністю сигналу на гнучкому шлейфі, проте вмикається явним параметром і документований як експериментальна можливість [16]. До файлу `/boot/firmware/config.txt` додано рядок:

```
dtparam=pciex1_gen=3
```

Лістинг 3.7 - Активація режиму PCIe Gen 3

Контроль стабільності. Оскільки використовується бюджетний накопичувач (а в розділі 2 вже було зафіксовано його схильність до збоїв енергозбереження), після активації Gen 3 проведено ретельну перевірку. Лінк

успішно піднявся до 8,0 ГТ/с, а журнал ядра не містив жодних помилок контролера чи шини PCIe навіть під навантаженням:

```
$ lspci -vv | grep LnkSta
LnkSta: Speed 8GT/s, Width x1
$ dmesg | grep -iE "nvme.*(down|reset|error)|pcie.*error"
(порожній вивід - помилок немає)
```

Лістинг 3.8 - Підтвердження стабільного лінка PCIe Gen 3 (8 ГТ/с, без помилок)

Результат. Активація Gen 3 майже подвоїла локальну швидкість накопичувача, що узгоджується з результатами незалежних тестувань PCIe на Raspberry Pi 5 [28]. Зведені дані наведено в таблиці 3.1 та на рисунку 3.1.

Таблиця 3.1

Локальна швидкість NVMe до і після активації PCIe Gen 3

Тест (fio)	**PCIe Gen 2**	**PCIe Gen 3**	**Приріст**
Послідовний запис	439 МБ/с	825 МБ/с	+88 %
Послідовне читання	455 МБ/с	858 МБ/с	+89 %
Випадковий 4 КБ (читання)	60,8 МБ/с (14,8k IOPS)	76,4 МБ/с (18,7k IOPS)	+26 %
Випадковий 4 КБ (запис)	184 МБ/с (45,0k IOPS)	225 МБ/с (54,8k IOPS)	+22 %

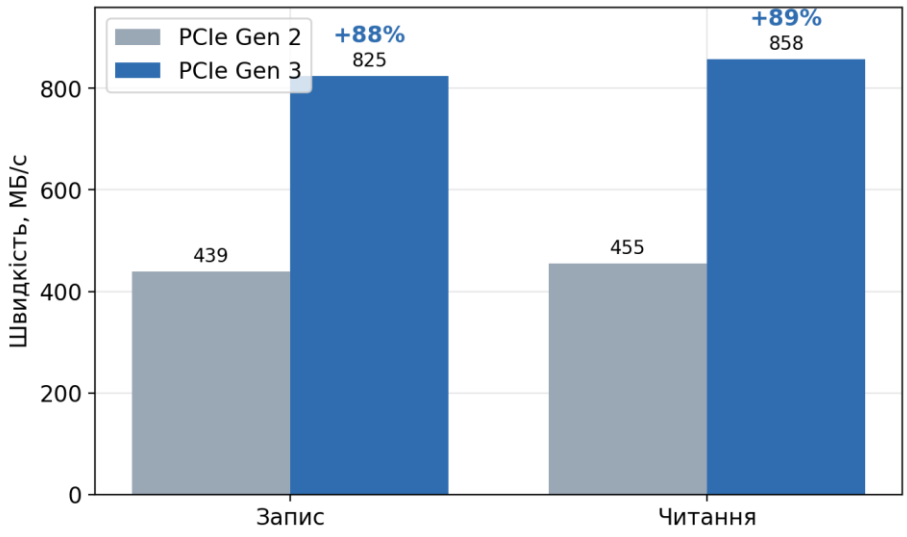


Рис. 3.1 - Локальна швидкість NVMe: PCIe Gen 2 → Gen 3 (послідовний доступ)

Послідовна швидкість зросла приблизно на 88 %, наблизившись до межі PCIe Gen 3 x1. Випадковий доступ дрібними блоками зріс помітно менше (+22-26 %), що закономірно: для операцій 4 КБ визначальною є затримка контролера, а не пропускна здатність шини. Важливо підкреслити, що цей приріст - **локальний**: він підвищує швидкодію дискових операцій на

самому сервері (зокрема для важких служб на кшталт NextCloud), але, як буде показано далі, не впливає на швидкість обміну через мережу. Оскільки режим Gen 3 виявився повністю стабільним, його залишено ввімкненим.

3.4 Оптимізація параметрів Samba та аналіз «вузького місця»

Налаштування Samba. До конфігурації сервісу додано рекомендовані параметри підвищення продуктивності [23]: вимкнення алгоритму Нейгла (`TCP_NODELAY`) для зниження затримки, режим «нульового копіювання» `use sendfile = yes` (ядро передає дані напряму з дискового кешу в мережевий сокет), асинхронний ввід-вивід та обмеження мінімальної версії протоколу до SMB2:

```
socket options = TCP_NODELAY IPTOS_LOWDELAY
use sendfile = yes
read raw = yes
write raw = yes
aio read size = 1
aio write size = 1
server min protocol = SMB2
```

Лістинг 3.9 - Параметри оптимізації Samba (/etc/samba/smb.conf)

Слід зауважити, що з рекомендованого набору свідомо вилучено явне задання розмірів буферів сокета (`SO_RCVBUF`/`SO_SNDBUF`): фіксований невеликий буфер перекрив би автотюнінг ядра, налаштований у п. 3.1, і на гігабіті міг би навпаки обмежити швидкість.

Повторні вимірювання показали, що **на пропускну здатність ці параметри не вплинули** - швидкість залишилась на рівні стелі каналу. Проте механізм `sendfile` зменшує навантаження на процесор під час передавання, тому ці налаштування залишено в системі як корисні.

Узагальнення: чому оптимізації не дали приросту. Результати всіх етапів зведено на рисунку 3.2. Пропускна здатність Samba (запис) залишалася практично незмінною - близько 110,8 МБ/с - незалежно від застосованих мережевих оптимізацій (буфери, IRQ, регулятор частоти, параметри Samba). Це не свідчить про марність оптимізацій загалом, а є

чітким діагнозом: **система обмежена пропускною здатністю гігабітного каналу**, який уже використовується майже повністю.

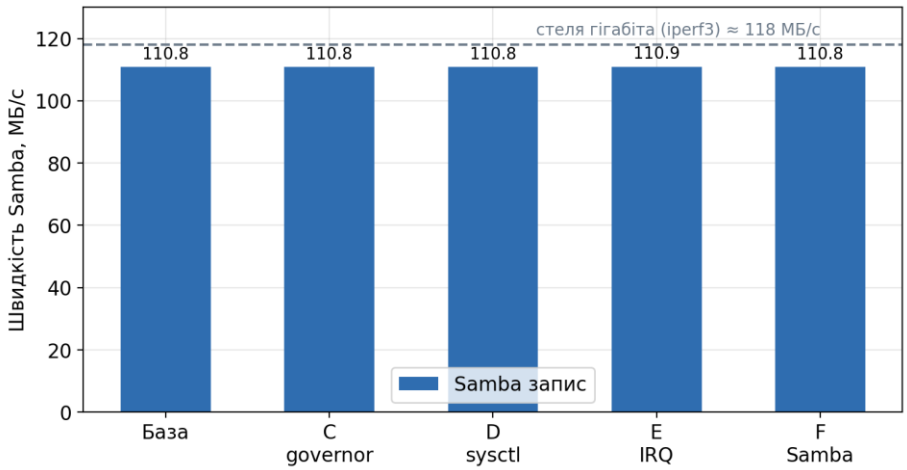


Рис. 3.2 - Пропускна здатність Samba за етапами оптимізації (результат стабільний - система обмежена каналом)

Цей висновок підтверджує прямий аналіз «вузького місця» (рис. 3.3). Зіставлення трьох вимірювань однозначно вказує на канал:

- локальна швидкість диска (NVMe Gen 3) - **858 МБ/с**;
- «чиста» пропускна здатність каналу (iperf3) - **118 МБ/с**;
- реальна швидкість Samba - **110,8 МБ/с**.

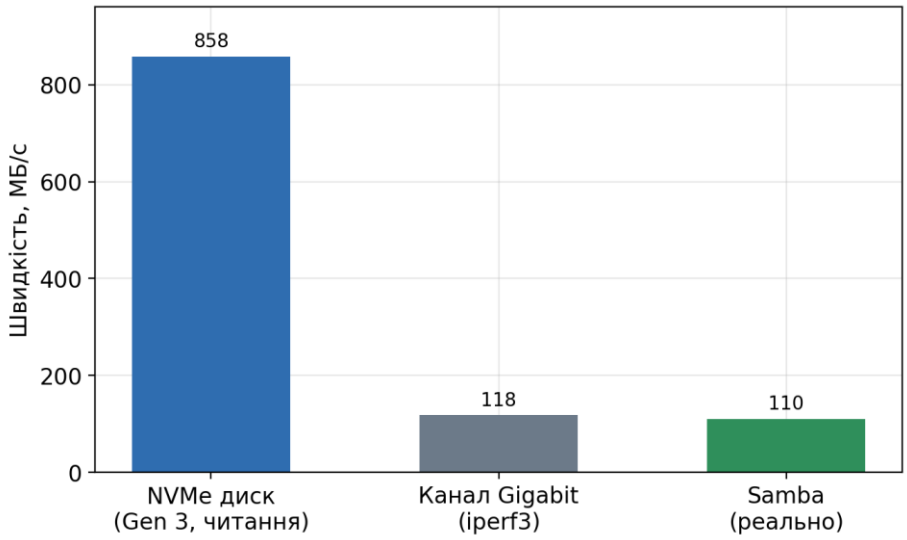


Рис. 3.3 - Аналіз «вузького місця»: диск приблизно у 7 разів швидший за мережу

Диск майже на порядок випереджає мережу, тому саме гігабітний канал, а не накопичувач чи процесор, визначає максимальну швидкість

файлового обміну. Реальна швидкість Samba (110,8 МБ/с) сягає близько 93 % теоретичної межі гігабіта, що є дуже хорошим показником - протокол SMB додає мінімальні накладні витрати.

Уточнення щодо швидкості читання. Окремо варто пояснити спостережену асиметрію: у тесті копіювання швидкість читання з NAS (≈93 МБ/с) виявлялася нижчою та нестабільнішою за швидкість запису (110,8 МБ/с). Додаткове вимірювання показало, що це **артефакт вимірювального клієнта, а не сервера**: при копіюванні з NAS клієнт одночасно записує отримані дані на власний локальний диск, який і обмежує швидкість. «Чисте» читання з NAS у пам'ять (без запису на диск клієнта) дало 108,7 МБ/с - практично стільки ж, скільки й запис. Отже, сам NAS симетричний і читає, і пише на рівні ~110 МБ/с (це підтверджує й симетричний результат iperf3 - 118 МБ/с в обидва боки). Цей приклад показує важливість відокремлення характеристик системи, що тестується, від обмежень вимірювального інструмента.

Висновки

У розділі виконано пошарову оптимізацію NAS-системи з експериментальною перевіркою ефекту кожної зміни. Отримані результати дозволяють зробити такі висновки.

1. **Дискова підсистема має значний резерв, який розкриває PCIe Gen 3.** Активація режиму Gen 3 підвищила локальну послідовну швидкість NVMe на ~88 % (з 439-455 до 825-858 МБ/с) і виявилася повністю стабільною навіть на бюджетному накопичувачі. Це найвагоміший приріст у розділі, корисний для локальних дискових операцій сервера.

2. **Система обмежена пропускною здатністю гігабітного каналу.** Жодна з мережевих оптимізацій (збільшення TCP-буферів, розподіл IRQ, режим процесора performance, тюнінг Samba) не підвищила швидкість мережевого обміну, оскільки канал уже використовується майже повністю

(Samba досягає ~93 % межі гігабіта). Це підтверджено прямим аналізом: накопичувач (858 МБ/с) приблизно в 7 разів швидший за мережу (118 МБ/с).

3. **Сучасне ядро виконує частину оптимізацій автоматично.** Зокрема, переривання NVMe вже оптимально розподілені по ядрах процесора без ручного втручання.

4. **Коректна методика дозволила відокремити характеристики NAS від обмежень клієнта** - зокрема, з'ясувати, що знижена швидкість читання у тесті була зумовлена локальним диском вимірювального ПК, а сам NAS симетричний (~110 МБ/с в обидва боки).

Таким чином, для подальшого зростання продуктивності цієї системи ключовим є не тонке налаштування програмного стека (його резерв вичерпано), а перехід на швидший мережевий інтерфейс. Детальне експериментальне дослідження продуктивності у різних умовах наведено в розділі 4.

РОЗДІЛ 4
ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ПРОДУКТИВНОСТІ NAS

У попередніх розділах побудовано та оптимізовано NAS-систему на базі Raspberry Pi 5. Цей розділ присвячено комплексному експериментальному дослідженню її продуктивності в умовах, наближених до реальної експлуатації. Мета - не лише отримати окремі показники швидкості, а й дати відповідь на практично важливі питання: яка реальна пропускна здатність сховища, наскільки вона залежить від типу під'єднання клієнта (дротове чи бездротове), як на неї впливає характер переданих даних (великі файли проти безлічі дрібних) і як отримана система співвідноситься з готовими комерційними рішеннями. Особливу увагу приділено методиці: щоб висновки були достовірними, кожне вимірювання виконувалося за єдиною статистично обґрунтованою процедурою.

4.1 Розроблення методики тестування та опис стенду

Випробувальний стенд. Усі вимірювання проведено на одній і тій самій конфігурації, що відповідає фінальному стану системи з розділів 2-3:

- **Сервер (NAS):** Raspberry Pi 5 (8 ГБ LPDDR4X), накопичувач SSD Silicon Power (SPCC) 256 ГБ у форматі M.2 2280, під'єднаний через плату M.2 HAT+ по інтерфейсу PCIe x1 (активовано режим Gen 3), активне охолодження (офіційний Active Cooler). ОС - Raspberry Pi OS (64-bit), ядро 6.18.29+rpt-rpi-v8. Сервер під'єднано до маршрутизатора кабелем (Gigabit Ethernet, інтерфейс eth0, статична адреса 192.168.3.101).
- **Клієнт:** ПК під керуванням Windows. Дротове під'єднання - через адаптер TP-Link UE300C (USB-C → Gigabit Ethernet) до того самого маршрутизатора; бездротове - через Wi-Fi того ж маршрутизатора. Це дозволяє коректно порівняти Ethernet і Wi-Fi за інших однакових умов.
- **Мережа:** гігабітний маршрутизатор Huawei WS5200, категорійний кабель; сервер і клієнт в одній локальній підмережі 192.168.3.0/24.

Схему випробувального стенду наведено на рисунку 4.1. Сервер NAS постійно під'єднано до маршрутизатора кабелем (Gigabit Ethernet), а клієнтський ПК під'єднується до того самого маршрутизатора двома способами почергово - кабелем (через адаптер USB-C → GbE) та по Wi-Fi, що дозволяє коректно порівняти обидва типи з'єднання за інших однакових умов.

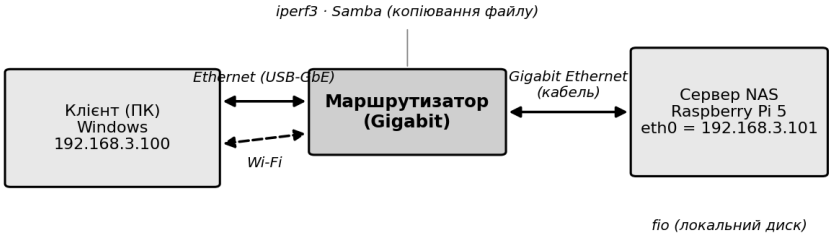


Рис. 4.1 - Схема випробувального стенду

На момент випробувань на сервері діяла фінальна конфігурація після оптимізації: PCIe Gen 3 увімкнено, параметри Samba та TCP-буфери ядра застосовано, регулятор частоти процесора - `ondemand`.

Інструменти вимірювання. Для різних рівнів системи використано спеціалізовані інструменти:

- `fio` - еталонний інструмент для локальної швидкості накопичувача (послідовний доступ блоком 1 МБ, випадковий доступ блоком 4 КБ із вимірюванням IOPS та затримки);
- `iperf3` [26] - для «чистої» пропускної здатності мережевого каналу без участі дискової підсистеми та файлового протоколу;
- штатне копіювання файлів засобами клієнта (`Measure-Command { Copy-Item … }`) та `robocopy` - для реальної швидкості файлового обміну через Samba, тобто для показника, який безпосередньо відчуває користувач.

Методика забезпечення достовірності. Щоб результати були відтворюваними, а не випадковими, дотримано таких правил.

1. **Повторюваність і статистика.** Кожен замір виконувався п'ять разів (`N = 5`). Перший, прогрівний (warm-up), прогін відкидався, а за рештою

чотирьох обчислювалися середнє значення, стандартне відхилення σ і коефіцієнт варіації CV = σ/середнє. Критерієм стабільності вважали CV < 3 %; у такому разі результат визнавали достовірним. 2. **Відкидання прогріву.** Перший прогін майже завжди аномальний (розподіл пам'яті, прогрів кешів, розгін вікна TCP), тому його завжди виключали зі статистики. 3. **Чисте середовище.** Під час вимірювань на сервері не виконувалося фонове навантаження (синхронізація NextCloud, паралельне копіювання тощо); стан спокою контролювали через `htop`. Усі тести одного етапу проводили в одному сеансі за сталих умов. 4. **Контроль нагріву.** Після кожного важкого тесту перевіряли температуру та стан тротлінгу (`vcgencmd measure_temp`, `vcgencmd get_throttled`). У всіх вимірюваннях `get_throttled` дорівнював `0x0`, температура не перевищувала ~50 °C - отже, активне охолодження виключило скидання частоти, і результати не спотворені перегрівом. 5. **Підтвердження усталеного режиму.** На базовому етапі мережеві тести виконано на двох розмірах файлу (1 ГБ і 4 ГБ): збіг швидкостей у межах σ підтвердив, що навіть на меншому файлі передавання встигає вийти на усталений режим, тобто вимірюється стала пропускна здатність, а не початкова фаза «розгону» з'єднання. 6. **Уникнення спотворення кешем.** Для тесту читання через Samba використовувався файл 4 ГБ - більший за обсяг кешу клієнта, щоб операційна система не віддавала дані з оперативної пам'яті замість мережі. Для `dd` той самий захист забезпечували прапорці `direct` та скидання кешу сторінок.

Окремо наголосимо на принципі **відокремлення системи, що тестується, від вимірювального інструмента**. Як буде показано в п. 4.3, занижена швидкість читання, яку спершу спостерігали, виявилася обмеженням локального диска клієнтського ПК, а не сервера. Урахування цього ефекту є обов'язковою умовою коректної інтерпретації результатів.

4.2 Аналіз пропускної здатності мережі

Перш ніж вимірювати швидкість файлового обміну, доцільно визначити «стелю» - максимальну пропускну здатність самого каналу, без впливу диска та протоколу SMB. Для цього використано `iperf3`, який генерує синтетичний TCP-потік безпосередньо між клієнтом і сервером.

Дротове під'єднання (Gigabit Ethernet). Канал працював на практичній межі гігабіта - **949 Мбіт/с** у напрямку «клієнт → NAS» і **945 Мбіт/с** у зворотному. Лічильник повторних передач (Retr) дорівнював нулю, а коефіцієнт варіації між прогонами був близький до нуля, що свідчить про ідеально чистий і стабільний канал:

```
$ iperf3 -c 192.168.3.101 -t 60
[ ID] Interval        Transfer     Bitrate         Retr
[  5] 0.00-60.01 sec  6.64 GBytes  950 Mbits/sec    0    sender
[  5] 0.00-60.04 sec  6.63 GBytes  949 Mbits/sec         receiver
```

Лістинг 4.1 - Пропускна здатність каналу по Ethernet (iperf3, ≈949 Мбіт/с, Retr = 0)

Отримане значення ≈949 Мбіт/с (близько 118 МБ/с) відповідає теоретичній межі Gigabit Ethernet з урахуванням службових витрат протоколів (заголовки Ethernet, IP, TCP). Симетричність результату (майже однакова швидкість в обидва боки) вказує, що канал не має «вузького» напрямку і саме він задаватиме верхню межу для файлового обміну.

Бездротове під'єднання (Wi-Fi). Для порівняння той самий тест проведено з клієнтом, під'єднаним по Wi-Fi через той самий маршрутизатор. Пропускна здатність виявилася суттєво нижчою та помітно асиметричною - **184 Мбіт/с** на передавання (клієнт → NAS) і **217 Мбіт/с** на приймання. Зведені дані наведено в таблиці 4.1.

Таблиця 4.1

Пропускна здатність каналу (iperf3): Ethernet проти Wi-Fi

Напрям	**Ethernet**	**Wi-Fi**
Клієнт → NAS (upload)	949 Мбіт/с (≈118 МБ/с)	184 Мбіт/с (≈23 МБ/с)
NAS → клієнт (download)	945 Мбіт/с (≈118 МБ/с)	217 Мбіт/с (≈27 МБ/с)
Повторні передачі (Retr)	0	поодинокі

Дротовий канал виявився приблизно у п'ять разів швидшим за бездротовий і значно стабільнішим. Бездротова пропускна здатність

залежить від багатьох змінних факторів (відстань до точки доступу, завантаженість радіоефіру, ширина каналу), тоді як Ethernet дає сталий результат, близький до апаратної межі.

4.3 Порівняльне тестування Samba: клієнт по Ethernet проти Wi-Fi

Визначивши стелю каналу, перейдемо до головного показника - реальної швидкості файлового обміну через Samba, яку відчуває користувач при копіюванні файлів на мережевий диск і з нього. Вимірювання виконано копіюванням файлу 4 ГБ у кожному напрямку (п'ять прогонів, прогрівний відкинуто).

Дротове під'єднання. Швидкість запису на NAS склала **110,8 МБ/с** при коефіцієнті варіації 0,4 % - результат стабільний і становить близько 93 % теоретичної межі гігабіта.

Швидкість читання спочатку видавалася нижчою (близько 93-96 МБ/с) і помітно нестабільною. Додаткове дослідження показало, що це **артефакт вимірювального клієнта, а не сервера**: при копіюванні файлу з NAS клієнтський ПК одночасно записує отримані дані на власний локальний диск, і саме цей диск стає обмежувальним фактором. Щоб виключити його вплив, проведено «чисте» читання з NAS у пам'ять (через `FileStream`, без запису на локальний диск клієнта):

```
# читання файлу з мережевого диска у пам'ять, без збереження на диск ПК
$f = [System.IO.File]::OpenRead("Z:\testfile.bin")
$buf = New-Object byte[] 8MB
while ($f.Read($buf, 0, $buf.Length) -gt 0) { }
```

Лістинг 4.2 - «Чисте» читання з NAS у пам'ять для виключення впливу диска клієнта

Цей тест дав **108,7 МБ/с** - практично стільки ж, скільки й запис. Отже, сам NAS симетричний: він і читає, і пише на рівні ~110 МБ/с, що узгоджується із симетричним результатом `iperf3` (118 МБ/с в обидва боки). Спостережене раніше «провисання» читання повністю пояснюється обмеженням диска клієнта.

Бездротове під'єднання. Той самий тест по Wi-Fi дав значно нижчі швидкості - **23,1 МБ/с** на запис і **26,5 МБ/с** на читання. Показовим є зіставлення цих значень із «чистою» пропускною здатністю каналу з п. 4.2: швидкість Samba по Wi-Fi (185 / 212 Мбіт/с в перерахунку) практично збігається з результатом `iperf3` (184 / 217 Мбіт/с), тобто сягає 98-100 % пропускної здатності каналу. Це означає, що протокол SMB майже не додає накладних витрат, а вузьким місцем є саме бездротовий канал.

Зведене порівняння наведено на рисунку 4.2.

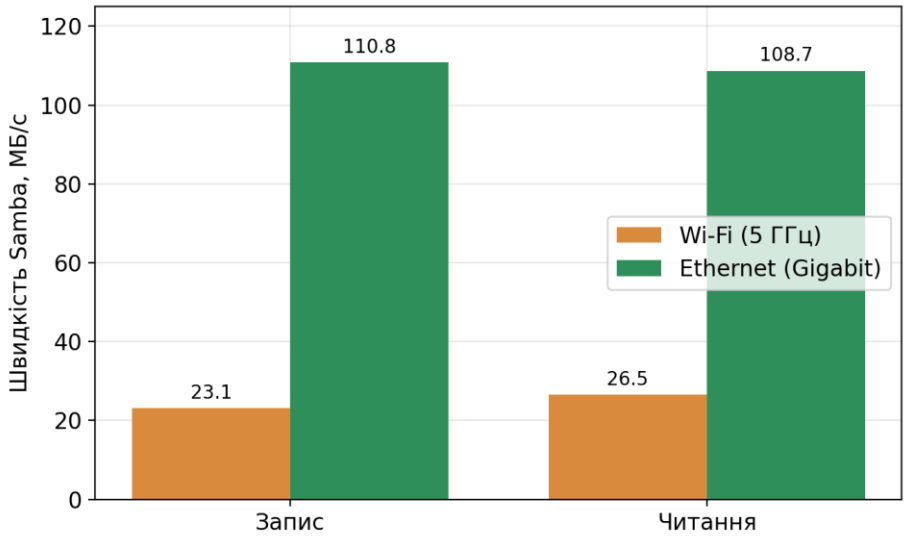


Рис. 4.2 - Швидкість Samba та пропускна здатність каналу: Ethernet проти Wi-Fi

Висновок. Дротове під'єднання забезпечує приблизно у 4-5 разів вищу та значно стабільнішу швидкість, ніж Wi-Fi (≈110 МБ/с проти ≈23-27 МБ/с). Для NAS, що працює з великими обсягами даних (резервні копії, медіабібліотека), дротове під'єднання сервера є не просто бажаним, а визначальним для продуктивності. Це підтверджує прийняте в розділі 2 архітектурне рішення - тримати сам сервер постійно на кабелі, а бездротовий доступ забезпечувати з боку клієнтських пристроїв там, де він прийнятний за швидкістю (перегляд, доступ до документів).

4.4 Дослідження впливу розміру та кількості файлів

Попередні тести використовували один великий файл, що відображає сценарій передавання відеоархівів чи образів дисків. Проте на практиці часто

доводиться копіювати багато дрібних файлів (фотоколекції, проєкти з вихідним кодом, документи). Щоб дослідити цей вплив, проведено серію вимірювань швидкості запису через Samba для наборів файлів різного розміру, але однакового сумарного обсягу (~1 ГБ). Копіювання виконувалося утилітою `robocopy` (три прогони, прогрівний відкинуто); повний текст сценарію вимірювання наведено в додатку В. Результати наведено в таблиці 4.2 та на рисунку 4.3.

Таблиця 4.2

Швидкість запису через Samba залежно від розміру файлів (сумарно ~1 ГБ, Ethernet)

Набір файлів	**Розмір одного файла**	**Швидкість, МБ/с**	**Частка від стелі**
16 × 64 МБ	64 МБ	111,2	~100 %
1 × 1024 МБ	1 ГБ	96,0	~87 %
256 × 4 МБ	4 МБ	85,6	~77 %
1024 × 1 МБ	1 МБ	35,2	~32 %
4096 × 256 КБ	256 КБ	4,7	~4 %

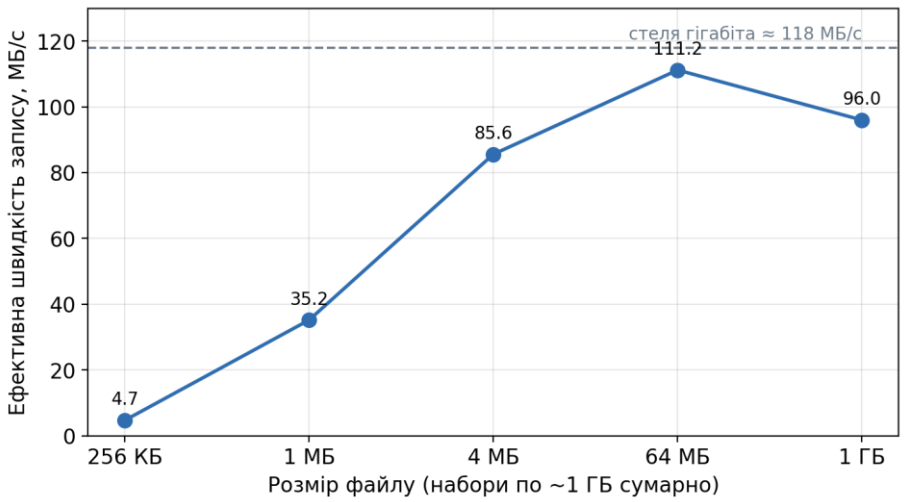


Рис. 4.3 - Вплив розміру файлів на швидкість запису через Samba

Залежність виражена надзвичайно різко. На великих файлах (64 МБ і більше) швидкість виходить на усталений рівень ~110 МБ/с - фактично на стелю гігабітного каналу. Дещо нижче значення для одиночного файла 1 ГБ (96,0 МБ/с) пояснюється особливістю вимірювання, а не системи: для одного файла завершальне скидання кешу запису повністю входить у виміряний час,

тоді як у серії файлів воно перекривається передаванням наступного файла; зі зростанням обсягу цей внесок зменшується - копіювання файла 4 ГБ (п. 4.3) дало 110,8 МБ/с. Зі зменшенням розміру файлів швидкість обвалюється: для файлів 1 МБ вона падає до 35 МБ/с, а для файлів 256 КБ - до 4,7 МБ/с, тобто приблизно у 23 рази повільніше за максимум (стелю каналу).

Причина - **накладні витрати на кожен окремий файл**, які не залежать від його розміру: відкриття та закриття сеансу SMB, передавання метаданих, створення запису у файловій системі, а головне - затримки на обмін підтвердженнями (round-trip) між клієнтом і сервером на кожну операцію. Для великого файла ці витрати незначні на тлі обсягу даних, але для тисяч дрібних файлів вони повністю домінують: канал простоює, очікуючи на службові обміни, замість того щоб передавати корисні дані.

Практичний висновок. Реальна швидкість роботи з NAS визначається не лише пропускною здатністю каналу, а й характером даних. Для ефективного передавання великої кількості дрібних файлів їх доцільно попередньо об'єднувати в архів (один великий файл копіюється на порядок швидше за тисячі дрібних того ж сумарного обсягу). Цей результат також пояснює, чому для сценаріїв із дрібними файлами латентність системи важливіша за «пікову» пропускну здатність.

4.5 Зведена таблиця результатів та порівняння з комерційними NAS

Зведення результатів. У таблиці 4.3 узагальнено ключові показники продуктивності розробленої системи, отримані в розділах 3 і 4.

Таблиця 4.3

Зведені показники продуктивності розробленого NAS

Показник	**Значення**	**Примітка**
Локальна швидкість NVMe (читання, Gen 3)	858 МБ/с	fio, послідовний доступ
Локальна швидкість NVMe (запис, Gen 3)	825 МБ/с	fio, послідовний доступ
Пропускна здатність каналу (Ethernet)	≈118 МБ/с	iperf3, 949 Мбіт/с
Samba, запис (Ethernet, великі файли)	110,8 МБ/с	~93 % межі гігабіта

Продовження таблиці 4.3

Samba, читання (Ethernet, великі файли)	108,7 МБ/с	симетрично запису
Samba, запис/читання (Wi-Fi)	23,1 / 26,5 МБ/с	обмежено каналом Wi-Fi
Samba, дрібні файли (256 КБ)	4,7 МБ/с	домінують накладні витрати
Температура під навантаженням	≤50 °C	без тротлінгу (0x0)

Отримана система забезпечує стабільну швидкість файлового обміну близько 110 МБ/с - практично на межі гігабітного каналу, з мінімальними накладними витратами протоколу. Локальна швидкість накопичувача (понад 800 МБ/с) має значний запас і реалізується у внутрішніх операціях сервера (служби NextCloud, індексація, робота з базою даних).

Порівняння з комерційними NAS. Коректне порівняння варто будувати не на абсолютних «паспортних» цифрах, а на чинниках, що визначають реальну продуктивність. Більшість комерційних NAS початкового рівня для дому й малого офісу оснащені гігабітними мережевими інтерфейсами (1GbE). Так, бюджетна дводискова модель QNAP TS-233 має один порт 1GbE, а Synology DS224+ - два порти 1GbE з можливістю агрегації каналів [29, 30]. Як експериментально доведено в розділі 3 і підтверджено в п. 4.2-4.3, саме мережевий канал, а не накопичувач, є вузьким місцем для файлового обміну. Тому при роботі з великими файлами через одне клієнтське з'єднання будь-який такий NAS - і комерційний, і розроблений у цій роботі - упирається в ту саму стелю близько 110 МБ/с. Моделі з двома портами 1GbE (як DS224+) можуть перевищити цю межу (до ~200 МБ/с) лише за рахунок агрегації обох каналів за одночасної роботи кількох клієнтів, проте одне з'єднання все одно лишається обмеженим ~110 МБ/с; розроблену систему можна масштабувати так само - під'єднанням USB-адаптера 2,5GbE. За цим, найважливішим для користувача показником (швидкість одного з'єднання у гігабітній мережі), розроблена система не поступається готовим рішенням свого класу.

Зіставлення за іншими характеристиками наведено в таблиці 4.4.

Таблиця 4.4

Порівняння розробленого NAS із типовим комерційним рішенням початкового рівня

Характеристика	**Розроблений NAS (Raspberry Pi 5)**	**Типовий комерційний 2-дисковий NAS (1GbE)**
Швидкість Samba (великі файли, одне з'єднання 1GbE)	~110 МБ/с	~110 МБ/с (та сама стеля гігабіта)
Тип основного накопичувача	NVMe SSD (PCIe Gen 3)	переважно 2,5″/3,5″ HDD
Локальна швидкість диска	>800 МБ/с	~150-250 МБ/с (HDD)
Енергоспоживання	низьке (одноплатний комп'ютер)	помірне (диски + контролер)
Гнучкість ПЗ	повний контроль (Linux, будь-які сервіси)	переважно екосистема виробника
Орієнтовна вартість	~216 EUR (з накопичувачем)	~180-330 EUR (без дисків)

Висновок порівняння. Розроблена система демонструє сприятливе співвідношення «ціна/продуктивність». За показником, який безпосередньо відчуває користувач у локальній мережі (швидкість обміну великими файлами ~110 МБ/с), вона еквівалентна комерційним рішенням початкового рівня, оскільки всі вони обмежені однаковим гігабітним каналом. При цьому розроблений NAS використовує швидший твердотільний накопичувач (що дає перевагу у внутрішніх операціях і латентності) та забезпечує повний контроль над програмним забезпеченням за порівнянної загальної вартості - ще й з урахуванням того, що в розроблену систему вже входить накопичувач, тоді як комерційні рішення продаються без дисків. Слід чесно зазначити й обмеження: комерційні рішення виграють у простоті налаштування «з коробки», готовій екосистемі застосунків та апаратній підтримці кількох дисків із RAID. Подальше підвищення пропускної здатності розробленої системи лежить не в площині програмних оптимізацій (їх резерв вичерпано, що показано в розділі 3), а в переході на швидший мережевий інтерфейс (2,5GbE та вище), що зніме обмеження поточного гігабітного каналу.

Висновки

У розділі проведено комплексне експериментальне дослідження продуктивності розробленого NAS за єдиною статистично обґрунтованою методикою (п'ять прогонів, відкидання прогріву, контроль стабільності за коефіцієнтом варіації, контроль нагріву). Основні результати такі.

1. **Визначено реальну продуктивність системи.** При дротовому під'єднанні NAS забезпечує стабільну симетричну швидкість файлового обміну ~110 МБ/с (запис 110,8, читання 108,7 МБ/с), що становить близько 93 % теоретичної межі гігабіта при мінімальних накладних витратах протоколу SMB.

2. **Кількісно оцінено вплив типу під'єднання.** Дротове з'єднання приблизно у 4-5 разів швидше та значно стабільніше за Wi-Fi (≈110 проти ≈23-27 МБ/с), що обґрунтовує постійне під'єднання сервера кабелем.

3. **Виявлено критичний вплив характеру даних.** Швидкість обміну різко падає зі зменшенням розміру файлів - від ~110 МБ/с на великих файлах до 4,7 МБ/с на файлах 256 КБ (у ~23 рази), через накладні витрати на кожну файлову операцію. Практична рекомендація - об'єднувати набори дрібних файлів в архів.

4. **Підтверджено конкурентоспроможність рішення.** За ключовим для користувача показником (швидкість обміну у гігабітній мережі) розроблений NAS не поступається комерційним рішенням початкового рівня, маючи при цьому швидший накопичувач, більшу гнучкість і нижчу вартість.

5. **Підтверджено коректність методики.** Зокрема, вдалося відокремити характеристики самого NAS від обмежень вимірювального клієнта, що дало змогу довести симетричність швидкості читання й запису сервера.

Отримані результати підтверджують, що поставлену мету досягнуто: побудовано працездатний, продуктивний і економічно ефективний NAS, продуктивність якого об'єктивно виміряно й проаналізовано, а її обмеження - чітко локалізовано (гігабітний мережевий канал).

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

У дипломній роботі вирішено актуальну практичну задачу - розроблено, оптимізовано та експериментально досліджено продуктивне й економічно ефективне мережеве сховище даних (NAS) на базі одноплатного комп'ютера Raspberry Pi 5. Поставлену мету досягнуто, усі визначені задачі виконано. За результатами роботи можна зробити такі висновки.

1. **Обґрунтовано вибір платформи та базових технологій.** Порівняльний аналіз показав, що для домашнього використання та малого офісу самостійно зібране сховище на базі Raspberry Pi 5 виграє за вартістю, гнучкістю та контролем над системою, поступаючись комерційним пристроям лише у простоті початкового налаштування. Для однодискового сховища на платформі з обмеженою оперативною пам'яттю обґрунтовано вибір файлової системи ext4 (як найстабільнішої та найменш ресурсомісткої), основного протоколу доступу SMB (за універсальну підтримку всіма клієнтськими платформами) та протоколу SFTP для захищеного віддаленого передавання файлів.

2. **Реалізовано повнофункціональний NAS.** Побудовано працездатну систему із завантаженням операційної системи безпосередньо з NVMe-накопичувача, файловим сервісом Samba, хмарною платформою NextCloud та захищеним віддаленим доступом через VPN-мережу (Tailscale) і SFTP. Окрему увагу приділено безпеці: реалізовано вхід лише за криптографічним ключем, скорочено поверхню атаки шляхом вимкнення зайвих мережевих служб і застарілих протоколів та заборонено гостьовий (анонімний) доступ до файлового сервісу.

3. **Виконано пошарову оптимізацію продуктивності з експериментальною перевіркою.** Встановлено, що активація режиму PCIe Gen 3 підвищує локальну послідовну швидкість накопичувача приблизно на 88 % (з 439-455 до 825-858 МБ/с) і є повністю стабільною навіть на бюджетному SSD. Це найвагоміший приріст, отриманий у роботі.

4. **Експериментально виявлено «вузьке місце» системи.** Доведено, що жодна з мережевих оптимізацій (збільшення TCP-буферів, розподіл переривань, режим процесора, тюнінг Samba) не підвищує швидкість мережевого обміну, оскільки система обмежена пропускною здатністю гігабітного каналу. Реальна швидкість Samba (110,8 МБ/с) сягає близько 93 % теоретичної межі гігабіта, тоді як накопичувач приблизно у сім разів швидший за мережу. Отже, визначальним обмеженням є саме мережевий інтерфейс, а не диск чи процесор.

5. **Кількісно оцінено вплив умов експлуатації.** Експериментально показано, що дротове під'єднання приблизно у 4-5 разів швидше та значно стабільніше за Wi-Fi (≈110 проти ≈23-27 МБ/с), а швидкість обміну різко падає зі зменшенням розміру файлів (від ≈110 МБ/с на великих файлах до 4,7 МБ/с на файлах 256 КБ) через накладні витрати на кожну файлову операцію. Підтверджено також важливість коректної методики вимірювань - зокрема, відокремлення характеристик самого сховища від обмежень вимірювального клієнта.

6. **Підтверджено конкурентоспроможність рішення.** За ключовим для користувача показником - швидкістю обміну у гігабітній мережі - розроблений NAS не поступається комерційним рішенням початкового рівня, які обмежені тим самим гігабітним каналом, маючи при цьому швидший накопичувач, більшу гнучкість і нижчу вартість (близько 216 EUR).

Практичне значення. Отримана конфігурація є завершеним, відтворюваним рішенням, придатним для безпосереднього використання як домашнє чи офісне сховище, а також у навчальному процесі. Запропонована методика експериментального оцінювання продуктивності може застосовуватися для дослідження інших подібних систем.

Напрями подальшого розвитку. Оскільки доведено, що резерв програмних оптимізацій вичерпано, а обмежувальним чинником є мережа, найперспективнішим напрямом підвищення продуктивності є перехід на швидший мережевий інтерфейс (2,5GbE та вище), здатний розкрити наявний

запас швидкості дискової підсистеми. Додатковими напрямами є впровадження апаратного резервування даних (RAID на кількох накопичувачах) та розширення набору сервісів сховища.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Campbell C. Raspberry Pi 4: The Complete User Guide for Beginners and Pro to Master the New Raspberry Pi 4 with Tips and Tricks for DIY Innovative Projects. Independently published, 2021. 84 p.

2. Smith C. Programming Raspberry Pi 3 & 4 For Beginners. Churchgate Publishing House, 2019. 226 p.

3. Pratt L. Raspberry Pi: 3 in 1 - Essential Beginners Guide + Tips and Tricks + Advanced Guide to Learn About the Realms of Raspberry Pi. Independently published, 2021. 329 p. ISBN 979-8714828621.

4. Tech Hybrid. Raspberry Pi Guide: Beginner's Guide to Raspberry Pi. Hybird Tech, 2023. 90 p.

5. Peregrino E. Programming Raspberry Pi in 30 Days: Learn how to build amazing Raspberry Pi projects using Python with ease. BPB Publications, 2023. 398 p. ISBN 978-93-55513-717.

6. Linux File Systems: ext4 vs Btrfs vs ZFS // CBT Nuggets : вебсайт. URL: https://www.cbtnuggets.com/blog/technology/system-admin/linux-file-systems-ext4-vs-btrfs-vs-zfs (дата звернення: 21.01.2026).

7. Benchmarking Linux Filesystems: ZFS, XFS, Btrfs vs. ext4 // LinuxCommunity.io : вебсайт. URL: https://linuxcommunity.io/t/benchmarking-linux-filesystems-zfs-xfs-btrfs-vs-ext4/6405 (дата звернення: 27.01.2026).

8. Samba Documentation / The Samba Team : вебсайт. URL: https://www.samba.org/samba/docs/ (дата звернення: 04.02.2026).

9. Monk S. Raspberry Pi Cookbook: Software and Hardware Problems and Solutions. 4th ed. Sebastopol : O'Reilly Media, 2023. 621 p. ISBN 978-1-098-13092-3.

10. Remote access // Raspberry Pi Documentation : вебсайт. URL: https://www.raspberrypi.com/documentation/computers/remote-access.html (дата звернення: 06.02.2026).

11. Raspberry Pi 5 : Product Brief / Raspberry Pi Ltd. URL: https://cdn-shop.adafruit.com/product-files/6447/raspberry-pi-5-product-brief.pdf (дата звернення: 12.02.2026).

12. Raspberry Pi 5 Review - Hands-On with the Most Powerful Raspberry Pi Yet // Hackster.io : вебсайт. URL: https://www.hackster.io/news/raspberry-pi-5-review-hands-on-with-the-most-powerful-raspberry-pi-yet-57efaf61b10f (дата звернення: 16.02.2026).

13. 5 Surprising Features of Raspberry Pi 5 // PiCockpit : вебсайт. URL: https://picockpit.com/raspberry-pi/5-surprising-features-of-raspberry-pi-5/ (дата звернення: 18.02.2026).

14. McGinnis M. Raspberry Pi 5 Projects for Newbies: A Simplified Guide to Creating Projects with the Raspberry Pi 5. Independently published, 2024. 258 p.

15. P34A60 M.2 PCIe Gen3x4 NVMe SSD : Specifications // Silicon Power Computer & Communications Inc. : вебсайт. URL: https://www.silicon-power.com/web/product-P34A60 (дата звернення: 20.02.2026).

16. Configuration // Raspberry Pi Documentation : вебсайт. URL: https://www.raspberrypi.com/documentation/computers/configuration.html (дата звернення: 03.03.2026).

17. Brejcha J. One step closer to 10 Gigabit Ethernet on Raspberry Pi 5 - and it is hot : блог. URL: https://www.jiribrejcha.net/2024/04/one-step-closer-to-10-gigabit-ethernet-on-raspberry-pi-5-and-it-is-hot/ (дата звернення: 05.03.2026).

18. Getting started // Raspberry Pi Documentation : вебсайт. URL: https://www.raspberrypi.com/documentation/computers/getting-started.html (дата звернення: 06.03.2026).

19. nmcli : NetworkManager Reference Manual // NetworkManager : вебсайт. URL: https://networkmanager.dev/docs/api/latest/nmcli.html (дата звернення: 11.03.2026).

20. Geerling J. The ULTIMATE Raspberry Pi 5 NAS : відео // YouTube. URL: https://www.youtube.com/watch?v=l30sADfDiM8 (дата звернення: 14.03.2026).

21. Solid state drive/NVMe // ArchWiki : вебсайт. URL: https://wiki.archlinux.org/title/Solid_state_drive/NVMe (дата звернення: 18.03.2026).

22. How to build a Raspberry Pi NAS // Raspberry Pi : вебсайт. URL: https://www.raspberrypi.com/tutorials/nas-box-raspberry-pi-tutorial/ (дата звернення: 20.03.2026).

23. Optimizing Samba File Sharing Performance on Raspberry Pi Devices for Efficient and Lightweight Network Storage Solutions // International Journal of Novel Research and Engineering & Future Management (IJNREFM). 2024. Vol. 2, Issue 6. URL: https://ijnrefm.com/wp-content/uploads/IJNREFM_V2_issue6_122.pdf (дата звернення: 24.03.2026).

24. How Tailscale works // Tailscale Inc. : вебсайт. URL: https://tailscale.com/blog/how-tailscale-works (дата звернення: 26.03.2026).

25. Nextcloud - self-hosted productivity platform // Nextcloud GmbH : вебсайт. URL: https://nextcloud.com/ (дата звернення: 28.03.2026).

26. iperf3 / ESnet, Lawrence Berkeley National Laboratory : вебсайт. URL: https://software.es.net/iperf/ (дата звернення: 03.04.2026).

27. Interrupts and IRQ Tuning // Red Hat Enterprise Linux Performance Tuning Guide : вебсайт. URL: https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/6/html/performance_tuning_guide/s-cpu-irq (дата звернення: 07.04.2026).

28. Meyer B. PCIe Performance on Raspberry Pi 5 : блог. URL: https://iambillmeyer.com/posts/2025-01-02-raspberrypi5-pcie-performance/ (дата звернення: 09.04.2026).

29. DiskStation DS224+ : Specifications // Synology Inc. : вебсайт. URL: https://www.synology.com/en-global/products/DS224+ (дата звернення: 17.04.2026).

30. TS-233 : Specifications // QNAP Systems, Inc. : вебсайт. URL: https://www.qnap.com/en/product/ts-233 (дата звернення: 20.04.2026).

ДОДАТКИ

Додаток А

Фрагмент журналу ядра (dmesg) зі збоєм NVMe-контролера через механізм APST

Фрагмент зафіксовано на етапі первинного налаштування дискової підсистеми (NVMe використовувався як окремий розділ даних, до міграції операційної системи на NVMe). Несуттєві для аналізу записи журналу пропущено (позначено [...]).

```
[    0.000000] Linux version 6.18.29+rpt-rpi-2712 (serge@raspberrypi.com)
               (aarch64-linux-gnu-gcc-14 (Debian 14.2.0-19) 14.2.0 ...
[...]
[    0.496597] brcm-pcie 1000110000.pcie: link up, 5.0 GT/s PCIe x1 (!SSC)
[    0.496619] pci 0001:01:00.0: [1f99:1202] type 00 class 0x010802 PCIe Endpoint
[    0.496789] pci 0001:01:00.0: 4.000 Gb/s available PCIe bandwidth, limited by
               5.0 GT/s PCIe x1 link at 0001:00:00.0 (capable of 31.504 Gb/s
               with 8.0 GT/s PCIe x4 link)
[    0.497291] nvme nvme0: pci function 0001:01:00.0
[    0.504344] nvme nvme0: 4/0/0 default/read/poll queues
[    0.506667]  nvme0n1: p1
[...]
[    4.008914] EXT4-fs (nvme0n1p1): mounted filesystem d3dc98a6-b5fb-4308-8970-
               465fe8dffef2 r/w with ordered data mode. Quota mode: none.
[...]
[   52.196600] nvme nvme0: controller is down; will reset: CSTS=0xffffffff,
               PCI_STATUS=0x10
[   52.196608] nvme nvme0: Does your device have a faulty power saving mode
               enabled?
[   52.196610] nvme nvme0: Try "nvme_core.default_ps_max_latency_us=0
               pcie_aspm=off pcie_port_pm=off" and report a bug
[   52.220606] nvme 0001:01:00.0: enabling device (0000 -> 0002)
[   52.220620] nvme nvme0: Disabling device after reset failure: -19
[   52.236609] EXT4-fs (nvme0n1p1): shut down requested (2)
[   52.236615] Aborting journal on device nvme0n1p1-8.
[   52.236623] Buffer I/O error on dev nvme0n1p1, logical block 30965760,
               lost sync page write
[   52.236627] JBD2: I/O error when updating journal superblock for nvme0n1p1-8.
[...]
[ 1163.359271] EXT4-fs warning (device nvme0n1p1): htree_dirblock_to_tree:1051:
               inode #2: lblock 0: comm smbd[192.168.3.: error -5 reading
               directory block
[...]
[ 1385.088672] nvme nvme0: Identify namespace failed (-5)
```

Додаток Б

Підсумкова конфігурація сервісу Samba (вивід testparm -s)

Утиліта testparm виводить результуючу конфігурацію Samba (з урахуванням файлу tuning.conf) без коментарів та параметрів зі стандартними значеннями.

```
# Global parameters
[global]
        disable netbios = Yes
        log file = /var/log/samba/log.%m
        logging = file
        max log size = 1000
        obey pam restrictions = Yes
        pam password change = Yes
        panic action = /usr/share/samba/panic-action %d
        passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s*\s
                      password:* %n\n *password\supdated\ssuccessfully* .
        passwd program = /usr/bin/passwd %u
        restrict anonymous = 2
        server min protocol = SMB2
        server role = standalone server
        smb ports = 445
        socket options = TCP_NODELAY IPTOS_LOWDELAY
        unix password sync = Yes
        idmap config * : backend = tdb
        include = /etc/samba/tuning.conf
        use sendfile = Yes

[printers]
        browseable = No
        comment = All Printers
        create mask = 0700
        path = /var/tmp
        printable = Yes

[print$]
        comment = Printer Drivers
        path = /var/lib/samba/printers

[nas]
        comment = NAS Storage on Raspberry Pi 5
        create mask = 0660
        directory mask = 0770
        path = /mnt/nas
        read only = No
        valid users = kundel
```

Додаток В

Сценарій вимірювання впливу розміру файлів на швидкість запису (PowerShell)

Сценарій формує набори файлів однакового сумарного обсягу (~1 ГБ), копіює кожен набір на мережевий диск утилітою robocopy із вимірюванням часу та обчислює середню швидкість (перший, прогрівний, прогін відкидається).

```
# bench_filesize.ps1 - file-size / file-count impact on Samba write speed
# Write to NAS (PC -> NAS), each set ~1 GB total, folder copy via robocopy.
# Run on PC in PowerShell (Z: mapped to \\192.168.3.101\nas).

$nasRoot   = "Z:\fsbench"
$localRoot = "C:\bench\fsbench"
$repeats   = 3   # runs per set (1st = warm-up, dropped from average)

# Sets: name; file size (bytes); count (each set ~1024 MB total)
$sets = @(
  @{ n="1x1024MB";   size=1073741824; count=1 },
  @{ n="16x64MB";    size=67108864;   count=16 },
  @{ n="256x4MB";    size=4194304;    count=256 },
  @{ n="1024x1MB";   size=1048576;    count=1024 },
  @{ n="4096x256KB"; size=262144;     count=4096 }
)

# 1 MB random block to fill files (avoid compression/sparse)
$chunk = New-Object byte[] (1MB)
(New-Object Random).NextBytes($chunk)
function New-TestFile($path, $size) {
  $fs = [System.IO.File]::Create($path)
  $w = 0
  while ($w -lt $size) {
    $n = [Math]::Min($chunk.Length, $size - $w)
    $fs.Write($chunk,0,$n); $w += $n
  }
  $fs.Close()
}

Write-Host "Preparing file sets..." -ForegroundColor Cyan
Remove-Item $localRoot -Recurse -Force -ErrorAction SilentlyContinue
foreach ($s in $sets) {
  $dir = Join-Path $localRoot $s.n
  New-Item -ItemType Directory -Force $dir | Out-Null
  for ($i=0; $i -lt $s.count; $i++) {
    New-TestFile (Join-Path $dir "f$i.bin") $s.size
  }
  Write-Host ("  {0} - ready" -f $s.n)
}

$results = @()
Write-Host "`nMeasuring (write to NAS)..." -ForegroundColor Cyan
foreach ($s in $sets) {
  $src = Join-Path $localRoot $s.n
```

```
  $dst = Join-Path $nasRoot   $s.n
  $totalMB = ($s.size * $s.count) / 1MB
  $speeds = @()
  for ($r=1; $r -le $repeats; $r++) {
    Remove-Item $dst -Recurse -Force -ErrorAction SilentlyContinue
    $t = Measure-Command {
      robocopy $src $dst /E /NFL /NDL /NJH /NJS /NC /NP /NS /R:0 /W:0 | Out-Null
    }
    $mbps = [math]::Round($totalMB / $t.TotalSeconds, 1)
    $speeds += $mbps
  }
  Remove-Item $dst -Recurse -Force -ErrorAction SilentlyContinue
  $avg = if ($speeds.Count -gt 1) {
    [math]::Round((($speeds | Select-Object -Skip 1) |
      Measure-Object -Average).Average,1)
  } else { $speeds[0] }
  $results += [pscustomobject]@{ Set=$s.n; AllRuns=($speeds -join ", "); AvgMBps=$avg
}
  Write-Host ("  {0,-12}  ->  {1} MB/s" -f $s.n, $avg) -ForegroundColor Green
}

Write-Host "`n=== RESULTS 4.4 ===" -ForegroundColor Yellow
$results | Format-Table -AutoSize
Remove-Item $localRoot -Recurse -Force -ErrorAction SilentlyContinue
Write-Host "Temp files removed." -ForegroundColor Cyan
```