

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
« ____ » _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи зберігання та обробки
великих даних побудованої з використання протоколу NVMe”

КБГЗ - 2026

Виконав здобувач вищої освіти
IV курсу, групи КН-22
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Алмаєв О.Р.
« ____ » _____ 2026 р.

Керівник проекту
кандидат технічних наук
_____ Лисенко І.А.
« ____ » _____ 2026 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань . 12 “Інформаційні технології”
Спеціальність 122 “Комп’ютерні науки”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерні науки”

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 06 » лютого 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Алмасву Олександром Романовичу

(прізвище, ім'я, по батькові)

- Тема роботи Програмне забезпечення системи зберігання та обробки великих даних побудованої з використання протоколу NVMe
- Керівник роботи Лисенко Ірина Анатоліївна, канд. техн. наук
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 116-02 від 06.02.2026 року
- Строк подання студентом роботи до захисту 23.05.2026 р.
- Мета та завдання випускної кваліфікаційної роботи: Метою роботи є розробка програмного забезпечення системи зберігання та обробки великих даних побудованої з використання протоколу NVMe
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - Призначення та область використання.
 - Перегляд аналогічних існуючих систем.
 - Опис і обґрунтування проектних рішень.
 - Етапи програмування системи.
 - Впровадження системи в промислову експлуатацію.
 - Висновки
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

| | |
|--------------------------------------------|-----------------|
| <u>Структурна схема системи</u> | <u>1 аркуш</u> |
| <u>Функціональна схема системи</u> | <u>1 аркуш</u> |
| <u>Діаграма процесів</u> | <u>1 аркуш</u> |
| <u>Блок-схема алгоритму роботи додатку</u> | <u>2 аркуша</u> |

7. Дата видачі завдання « 06 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти | Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти | Примітка |
|-------|---------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|----------|
| 1. | Аналіз існуючих систем | 10.03.2026 р. | |
| 2. | Постановка задачі, оформлення ТЗ | 15.03.2026 р. | |
| 3. | Розробка моделі компонента | 20.03.2026 р. | |
| 4. | Розробка структур даних | 25.03.2026 р. | |
| 5. | Розробка алгоритмів зв'язку та відображення | 30.03.2026 р. | |
| 6. | Програмування алгоритмів | 10.04.2026 р. | |
| 7. | Оформлення ПЗ | 17.04.2026 р. | |
| 8. | Попередній захист роботи | 23.05.2026 р. | |
| | | | |
| | | | |
| | | | |
| | | | |

Дата видачі завдання
« 06 » лютого 2026 р.

Підпис керівника

Лисенко І.А.
(прізвище та ініціали)

Завдання прийнято до виконання
« 06 » лютого 2026 р.

Підпис здобувача

Алмаєв О.Р.
(прізвище та ініціали)

АНОТАЦІЯ

Алмаєв О.Р. Програмне забезпечення системи зберігання та обробки великих даних побудованої з використання протоколу NVMe. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи зберігання та обробки великих даних побудованої з використання протоколу NVMe.

Метою розробки є програмне забезпечення системи зберігання та обробки великих даних побудованої з використання протоколу NVMe.

Результат роботи – програмна реалізація системи зберігання та обробки великих даних побудованої з використання протоколу NVMe.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерні науки, зберігання та обробка великих даних

ABSTRACT

Almaiev O.R. Software for a big data storage and processing system built using the NVMe protocol. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed that is intended for a big data storage and processing system built using the NVMe protocol.

The purpose of the development is software for a big data storage and processing system built using the NVMe protocol.

The result of the work is a software implementation of a big data storage and processing system built using the NVMe protocol.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software are provided.

The program can be used on a PC with Windows 10/11 OS.

The program was developed in the Python environment.

Keywords: computer science, big data storage and processing

ЗМІСТ

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ | 2 |
| ВСТУП..... | 3 |
| 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ | 5 |
| 1.1 Призначення системи..... | 5 |
| 1.2 Область застосування..... | 9 |
| 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ | 10 |
| 2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти..... | 10 |
| 2.2 Обґрунтування вибору засобів для побудови системи та мови програмування..... | 21 |
| 2.3 Розгорнута постановка завдання | 22 |
| 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ | 23 |
| 3.1 Опис функціонування системи | 23 |
| 3.2 Розробка структурної схеми..... | 31 |
| 3.3 Розробка функціональної схеми | 38 |
| 3.4 Розробка діаграми процесів..... | 45 |
| 4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ..... | 47 |
| 4.1 Розробка блок-схем та опис алгоритмів функціонування системи..... | 47 |
| 4.2 Захист розробленого програмного забезпечення..... | 56 |
| 5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ | 58 |
| 6 ОСНОВНІ ВИСНОВКИ..... | 64 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 66 |

| | | | | | | | | |
|----------|----------------|----------|-------|------|--------------------------------------------------------------------------------------------------------------|------|-------|---------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | | | |
| Вим. | Арк. | № докум. | Підп. | Дата | | | | |
| Розроб. | Алмасв О.Р. | | | | Програмне забезпечення системи зберігання та обробки великих даних побудованої з використання протоколу NVMe | Літ. | Аркуш | Аркушів |
| Перев. | Лисенко І.А. | | | | | Б | 1 | 72 |
| Н.контр. | Коваленко А.С. | | | | ЦНТУ КН-22 | | | |
| Затв. | Смірнов О.А. | | | | | | | |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

| | | |
|--------|---|------------------------------------------|
| ІММ | – | імітаційна програмна модель |
| ЛОМ | – | локальні обчислювальні мережі |
| ПрЗд | – | пропускна здатність |
| ПМ | – | програмна модель |
| ТПП | – | таймер повторної передачі |
| ARTCP | – | удосконалений транспортний протокол |
| BER | – | імовірність бітової помилки |
| CR | – | запит на з'єднання |
| CBR | – | протокол передачі даних без підтверджень |
| FIFO | – | принцип first input first output |
| IP | – | адресний протокол |
| OSI | – | еталона модель мережної архітектури |
| RTT | – | час затрачуємий підтвердженням |
| TCP | – | протокол транспортного рівня |
| TCP/IP | – | протокол передачі даних |
| TPDU | – | повідомлення транспортного протоколу |
| UDP | – | протокол користувальницьких датаграм |
| WAN | – | територіальні мережі |

ВСТУП

Актуальність теми. В останні кілька років удосконалювання технологій зберігання даних відбувається семимільними кроками. Стрімко підвищуються ємність і продуктивність, оскільки популярні виробники виводять на ринки свої високопродуктивні NVMe SSD накопичувачі. Майже кожний виробник сховищ у світі ретельно вивчив і розробив енергонезалежні пристрої й NVMe стала однією з найбільш обговорюваних тем у галузі.

NVMe – специфікація на протоколи доступу до твердотільних накопичувачів (SSD), підключених по шині PCI Express, (від англ. Non-Volatile Memory Express), розроблена робочою групою з більше сотні компаній. Перевага стека NVMe – він забезпечує безпрецедентну швидкість і максимальну пропускну здатність для твердотільних накопичувачів. Звичайні SSD не забезпечують пропускну здатності, необхідної застосункам з особливо інтенсивним обміном даними, таких як масивні бази даних, для яких NVMe дає небувало низькі затримки й високі пікові швидкості.

Стандартні інтерфейси надзвичайно важливі для росту використання флеш-пам'яті. NVMe дає стандартний метод доступу до флеш-пам'яті, здатний використовувати можливості низької затримки й паралелізму інтерфейсів PCIe, сучасних процесорів, платформ і додатків. Скорочення затримок при роботі протоколу й збільшення паралелізму значно підвищують продуктивність операцій читання/запису SSD при одночасному контролі витрат на зберігання. Потенціал флеш-пам'яті до кінця не реалізований, і використання SSD у центрах обробки даних продовжує розширюватися.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи зберігання та обробки великих даних побудованої з використання протоколу NVMe.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 3 |

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем зберігання та обробки великих даних побудованої з використання протоколу NVMe.

– Дослідження системи зберігання та обробки великих даних побудованої з використання протоколу NVMe.

– Програмна реалізація системи зберігання та обробки великих даних побудованої з використання протоколу NVMe.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі зберігання та обробки великих даних побудованої з використання протоколу NVMe.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи зберігання та обробки великих даних побудованої з використання протоколу NVMe, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

КБПЗ-2026

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 4 |

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Хмарне сховище NVMe стосується хмарних сервісів зберігання даних, які використовують технологію енергонезалежної пам'яті Express (NVMe) для забезпечення надшвидкого доступу до даних з низькою затримкою. NVMe – це високопродуктивний інтерфейс та протокол зберігання, розроблений спеціально для твердотільних накопичувачів (SSD) та сучасних архітектур флеш-пам'яті. Інтеграція в хмарні середовища NVMe змінює спосіб зберігання, отримання та обробки даних, забезпечуючи виняткову продуктивність для програм, що потребують аналітики в режимі реального часу, високої пропускної здатності та масивного паралельного використання даних.

Хмарне сховище NVMe являє собою значний крок вперед порівняно з застарілими технологіями зберігання даних, такими як Serial ATA (SATA) або Serial Attached SCSI (SAS), які спочатку створювалися для обертових дисків. Ці старіші протоколи обмежені нижчою пропускною здатністю та вищою затримкою, що робить їх непридатними для робочих навантажень, що вимагають швидкості та масштабованості.

На відміну від цього, NVMe був спеціально розроблений для SSD-накопичувачів і працює через шину PCIe (Peripheral Component Interconnect Express), що забезпечує прямий зв'язок між носієм інформації та процесором. Це призводить до значного збільшення кількості операцій введення/виведення за секунду (IOPS), зменшення затримки та ефективнішого використання паралельних потоків обробки.

Огляд протоколу NVMe

Протокол NVMe спрощує та пришвидшує зв'язок між сховищем та обчисленнями, зменшуючи накладні витрати та оптимізуючи набори команд. Він

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 5 |

підтримує:

– Тисячі паралельних черг: на відміну від традиційних протоколів зберігання даних, які підтримують одну чергу команд, NVMe підтримує до 64 000 черг із 64 000 команд у кожній черзі.

– Структура команд з низькою затримкою: NVMe використовує менше циклів процесора для виконання операцій вводу/виводу.

– Прямий зв'язок PCIe: NVMe обходить багато традиційних вузьких місць, взаємодіючи безпосередньо з процесором через лінії PCIe.

– 2. Інтеграція хмарної інфраструктури

Хмарні рішення для зберігання даних NVMe розгортають SSD-накопичувачі NVMe у розподілених хмарних інфраструктурах. Їх часто групують за такими критеріями:

– NVMe-over-Fabrics (NVMe-oF): розширює можливості NVMe через мережеві тканини, такі як Ethernet, Fibre Channel або InfiniBand, забезпечуючи віддалений доступ до сховища NVMe з мінімальною затримкою.

– Гіперконвергентна інфраструктура (HCI): поєднує обчислювальні, сховищні та мережеві можливості в одній системі, де NVMe виконує роль високошвидкісного рівня зберігання даних.

– Зберігання як послуга (STaaS): пропонує сховище на базі NVMe через гнучкі моделі підписки, що усуває необхідність капіталовкладень в обладнання.

Основні переваги хмарного сховища NVMe:

– Надзвичайно низька затримка. NVMe значно зменшує затримку, оптимізуючи шляхи зв'язку та обходячи накладні витрати ядра. Затримка може знизитися з мілісекунд до мікросекунд, що важливо для чутливих до часу програм, таких як аналітика в реальному часі, торгові платформи та індексація баз даних.

– Висока пропускна здатність та кількість IOPS. Використовуючи паралельні черги та швидкі лінії PCIe, NVMe забезпечує неперевершену продуктивність IOPS. Він підтримує величезну кількість одночасних операцій читання та запису, що робить його ідеальним для: Аналіз великих даних.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 6 |

Штучний інтелект (ШІ) та машинне навчання (МН). Обробка медіа (рендеринг відео, транскодування). Електронна комерція та фінансові операції.

– Масштабованість. Хмарні рішення для зберігання даних NVMe розроблені для безперешкодного горизонтального масштабування. У міру зростання організацій сервіси на базі NVMe можуть обробляти більші обсяги даних та навантаження користувачів без зниження продуктивності.

– Енергетична та економічна ефективність. Твердотільні накопичувачі NVMe є енергоефективнішими, ніж механічні накопичувачі, що зменшує споживання енергії та вимоги до охолодження в центрах обробки даних. Хоча початкові витрати вищі, ніж у традиційних жорстких дисків, загальна вартість володіння (ТСО) зменшується завдяки кращій продуктивності, довговічності та зниженим потребам в інфраструктурі.

– Сумісність із сучасними робочими навантаженнями. Хмарні додатки, оркестрація на основі Kubernetes та контейнери значно вииграють від високошвидкісної та стабільної продуктивності, яку забезпечує NVMe. Розробники та операційні команди можуть створювати та розгортати швидші та надійніші сервіси.

Типові випадки використання:

– Високопродуктивні бази даних. Хмарне сховище NVMe ідеально підходить для транзакційних баз даних (OLTP), аналітики (OLAP) та гібридних моделей. Такі системи, як MySQL, PostgreSQL, Oracle та Microsoft SQL Server, отримують вигоду від швидшої обробки запитів та скорочення часу індексації.

– Штучний інтелект та машинне навчання. Робочі навантаження навчання та логічного висновку штучного інтелекту вимагають швидкого отримання та зберігання даних. NVMe пришвидшує цей процес, особливо під час розпізнавання зображень, обробки природної мови та великомасштабних наборів даних.

– Аналітика в режимі реального часу. Такі галузі, як фінанси, логістика та кібербезпека, покладаються на обробку даних у режимі реального часу. NVMe-сховище забезпечує швидке отримання, перетворення та аналіз поточкових даних.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 7 |

– Медіа та створення контенту. Редагування відео, 3D-рендеринг, розробка ігор та зображення високої роздільної здатності вимагають високої пропускної здатності. NVMe підтримує безперебійні робочі процеси з контентом та швидший час рендерингу.

– Віртуалізація та контейнери. Хмарним середовищам, що використовують віртуальні машини (VM) та контейнери, потрібне швидке сховище для скорочення часу завантаження, пришвидшення міграцій та підтримки масштабування в режимі реального часу. Хмарне сховище NVMe забезпечує плавність та ефективність цих операцій.

– Платформи електронної комерції. Системи онлайн-торгівлі вимагають швидкого завантаження, миттєвого оформлення замовлення та відстеження запасів у режимі реального часу. Низьколатентний доступ NVMe допомагає зменшити кількість покинутих кошиків та підвищити задоволеність користувачів.

Проблеми та міркування:

– Управління витратами. Твердотільні накопичувачі NVMe дорожчі за гігабайт, ніж жорсткі диски. Хоча переваги в продуктивності часто виправдовують інвестиції, організації повинні ретельно оцінювати вимоги до робочого навантаження та вибрати відповідний рівень сховища.

– Складність розгортання. Керування пристроями NVMe, особливо в середовищах NVMe-oF, може призвести до архітектурної складності. Для розгортання та керування може знадобитися оновлений драйвер, оптимізовані мережеві шляхи та спеціалізовані навички.

– Довговічність та резервування даних. Швидке сховище не усуває потреби в резервуванні та захисті. Постачальники хмарних послуг повинні поєднувати продуктивність NVMe з надійними стратегіями реплікації, кодування стирання та резервного копіювання, щоб забезпечити цілісність даних.

– Сумісність. Застарілі системи або програми можуть не повністю використовувати потенціал NVMe. Для використання можливостей NVMe можуть знадобитися стратегії міграції та оптимізації.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 8 |

1.2 Область застосування

Областю застосування є центр зберігання даних. Дата центр і ЦОД – усе це рідкі поняття для українських компаній. І якщо серед ІТ-шників його роль питань не викликає, то починаючи від менеджерів до власників компаній інших областей виникає маса питань. Питання дуже різноманітні – від того, що таке Дата-центр, як улаштований процес його роботи й властиво навіщо він потрібний. Так що ж це таке Дата-центр? Звичайно можна звернутися до Вікіпедії й прочитати, що: Дата-центр (від англ. data center), або центр (зберігання й) обробки даних (ЦОД/ЦЗОД) – це спеціалізований будинок для розміщення (хостинга) серверного й мережного встаткування. Це ясно для згаданих вище всіх тих же ІТ-шників.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи зберігання та обробки великих даних побудованої з використання протоколу NVMe, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 9 |

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

Intel DC P3600 і P3700

Ми познайомимося з парою накопичувачів Intel, що підтримують NVMe.

Першим з них є DC P3600 ємністю 800 ГБ. Виконано він у вже звичному форм-факторі – низькопрофільна карта розширення PCIe x4. На відміну від багатьох інших накопичувачів з інтерфейсом PCIe, вивчених нами раніше й представляючих собою карту формату M.2 у комплекті з адаптером-переходником, виглядає він як закінчене рішення й постачене досить значним радіатором (про що пари слів скажемо нижче).

А от DC P3700 на 400 ГБ саме має виконання в «вінчестерному» стилі, про яке говорилося вище. Треба помітити, з його підключенням нам довелося попотіти: поки на ринку надлишку кошиків і кабелів, необхідних для підключення цих пристроїв, не спостерігається.

На що варто звернути увагу? Цифри споживаного струму по лінії +12 В вселяють повагу: як бачите, накопичувач вимагає майже 30 Вт. Насправді, не всі так страшно: на наклейці наведена загальна інформація для всієї лінійки, і навіть старша модель (ємністю 2 ТБ) споживає 25/11 Вт при записі/читанні даних, а нашим героям досить 12/9 Вт. Втім, і це чимало. Для порівняння – багато вінчестерів для ноутбуків цілком обмежуються п'ятьома ватами, тобто такий твердотільний накопичувач у разі прожорливіший. Масові ж SSD у цей час нерідко укладаються в десяті частки вата, так що розрив з ними ще більше. Це й пояснює наявність масивних радіаторів у всього сімейства, а також відразу дає

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 10 |

зрозуміти, що в портативних комп'ютерах подібним до пристроїв робити нема чого.

Що ж стосується внутрішнього пристрою, то воно у всіх трьох лінійках (є ще й P3500) приблизно однакове – відрізняються тільки мікросхеми флеш-пам'яті, що дає небагато різні швидкісні показники й досить що помітно розрізняється витривалість: для наших героїв це 4,38 і 7,3 PBW відповідно. Ціною же пристрою не тільки розрізняються між собою, але й сильно відрізняються від накопичувачів споживчого сегмента – мова йде про 1, 5-2 доларах за гігабайт, тобто в парі раз більше, ніж для найшвидших «користувальницьких» SSD, що, втім, для знайомих з корпоративним сегментом не ново – за більше високу надійність прийнято платити.

А основою всіх трьох згаданих «корпоративних» лінійок і більше дешевого Intel 750 (націленого на ринок персональних комп'ютерів і робочих станцій) є власний контролер Intel CN29AE41AB0. Він підтримує вісімнадцять каналів для підключення флеша, у той час як «побутові» контролери обмежуються вісьма, а те й чотирма. Високий паралелізм (яким, як уже були сказано вище, саме NVMe-пристрої здатні розпорядитися щонайкраще) є запорукою високої продуктивності, але є в нього й побічний ефект: максимальної швидкодії можна чекати лише від старших (і найдорожчих!) модифікацій. А от 400 і навіть 800 ГБ, настільки значних на тлі найбільш продаваних у роздріб SSD – обмаль, що ми нижче й побачимо.

Практичне експрес-тестування

Зштовхнувшись із певними проблемами сумісності, про що було сказано вище, ми прийшли до виводу, що протестувати ці моделі по нашій стандартній методиці не вийде. Втім, ми не занадто розбудувалися: найближчим часом будуть готові результати «побутової» моделі Intel 750, з якої проблем виявилось істотно менше. Тому (раз вуж у нас матеріал сьогодні більше теоретичний) ми вирішили коротенько подивитися, чого можна й чого не можна очікувати від впровадження NVMe.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 11 |

Для тестування P3600 і P3700 нами був зібраний стенд на базі платформи LGA 2011-3, оскільки для даного сегмента такі накопичувачі в основному й рекомендуються. Відповідно, нам довелося використовувати досить потужний процесор Core i 7-5960X, але на результати дискових тестів (особливо тих, які ми використовуємо сьогодні) це не впливає. Обсяг пам'яті про всякий випадок постаралися обмежити, взявши всього один модуль DDR 4-2133 ємністю 4 ГБ. А для порівняння ми приведемо результати Kingston Hyper Predator і OCZ RevoDrive 350, саме недавно протестованих в «стандартних» умовах.

Отже, що стосується послідовного читання, те отут ми відразу зштовхнулися з тим, що при такій кількості каналів контролера «повільними» виявляються навіть моделі ємністю 400 ГБ! При 800 ГБ потенційні можливості PCIe 3.0 розкриваються більш повно. А при записі ми знову сильно впираємося не в інтерфейс, а в сам флеш із усіма що впливають: продуктивність обох моделей Intel виявилася більше низькою, ніж у банального Kingston Predator, не підтримуючого NVMe і працюючого всього лише з PCIe 2.0. Втім, подиву це в нас не викликало – зрозуміло, що властиво NVMe при таких типах навантаження нічого дати не може, а потенційні можливості швидкісних шин треба ще зуміти утилізувати.

А що з випадковим читанням блоками по 4ДО, що саме обіцяли прискорити? Отут – показово: дійсно стало помітно швидше, навіть якщо порівнювати наших героїв з OCZ RevoDrive, по суті своєї масивом, що є, RAID0 із чотирьох SSD. Predator же ще повільніше – у його випадку відставання від DC P3600 більш ніж дворазове (формально більше швидкому P3700 продовжує заважати низька ємність флеш-пам'яті).

На записі RevoDrive намагався не відстати, але й у нього це не вийшло. Що ще більш цікаво – результати «однопотокової» запису: прискорити її за допомогою RAID-масивів і подібних хитрувань неможливо, а от NVMe радикально міняє положення справ. Фактично, ці два пристрої Intel виявилися першими з попавших до нас у руки, що демонструють швидкості

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 12 |

«однопотокової» запису вище 100 МБ/с. Забігаючи вперед, повідомимо, що Intel 750 у цьому плані не гірше – і навіть краще: на ньому вдалося одержати майже 300 МБ/с. Ну а всі AHCI SSD (не тільки представлені на діаграмі, але та інші моделі) по цьому показнику «бовтаються» у районі 50-60 МБ/с, так що можна говорити про революційну зміну ситуації.

Втім, сама по собі революція – лише в досить вузькій області. З погляду простого користувача, що працює за комп'ютером, різниці між різними сучасними SSD ні, незалежно від інтерфейсів або протоколів роботи.

Виводи

Як бачимо, NVMe не панацея, що було зрозуміло апріорі – пропускну здатність шини ця технологія не збільшує, а зменшення затримок по масштабі не витримує ніякого порівняння з тим, що було отримано при переході від механіки до флеш-пам'яті. З іншого боку, технологія не тягне за собою «вантаж сумісності» з тією самою механікою й взагалі орієнтована саме на напівпровідникові носії інформації, причому не обов'язково на флеш-пам'ять – просто інші технології поки ще не вийшли за стіни лабораторій. Однак підготуватися до їхньої появи можна й задалегідь – головне, що потенційно NVMe підійде й для них. А якщо при цьому будуть знижені власні затримки пам'яті, виходить, застосування подібних технологій більш ніж виправдано, оскільки вони краще традиційного SATA.

Деякий ефект є й зараз. Чи є сенс за ним ганятися? На наш погляд, поки зарано робити це масово, оскільки він не занадто значний. Більшості користувачів як і раніше прекрасно підійдуть звичайні SATA-накопичувачі, благо вони мають і кращу в класі сумісність. А от тим, хто бажає одержати саме-найкраще, без інтерфейсу PCIe однаково не обійтися, комп'ютер у них уже, як правило, «самий-найкращий» і сучасний, так що проблем із сумісністю бути не повинне, і тоді покупка топового накопичувача з підтримкою NVMe може виявитися цілком виправданою. Головне – не розраховувати на те, що це принципово змінить «якість» роботи комп'ютера, звичайно.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 13 |

OCZ RevoDrive 350 480 ГБ

RevoDrive 350 – прямий спадкоємець RevoDrive3 X2, так що ця історія тягнеться ще до часів ери M.2. Так – уже в ті роки деякі виробники замислювалися про те, що комусь може не вистачити пропускнуї здатності інтерфейсу SATA, так що розробляли спеціалізовані пристрої, що підключаються безпосередньо до PCIe. Великого поширення на масовому ринку вони не одержали, оскільки їм не вистачало тої самої універсальності, у якої багато було сказано вище: підходили тільки до повнорозмірним настільних комп'ютерів, вимагали установки спеціальних драйверів. Але, у загальному й цілому, завдання одержання високої продуктивності вирішували. Нехай і трохи нетрадиційним образом. Зокрема в RevoDrive 350 використовується спеціалізований контролер ICT-0262, до якого підключені чотири розпаяних на платі SSD на базі контролерів SandForce SF-2281 у режимі RAID0 по 120 ГБ (у даній моделі) кожний, а з комп'ютером пристрій з'єднується за допомогою інтерфейсу PCIe 2.0 x8. Властиво, така складність і приводить до того, що пристрій виходить таки громіздким (а попередні моделі були зовсім двошартовими), але дуже швидким. Хоча б через RAID-режим, та й інтерфейс, як бачимо, удвічі «ширше» максимального для M.2. Втім, останній може використовувати й чотири лінії PCIe 3.0 хоча б теоретично, а ICT-0262 – пристрій PCIe 2.0, так що на практиці можна одержати й рівність пропускнуї здатності. Але такий от «комбінований бутерброд» доступний досить давно, а швидкі M.2-накопичувачі тільки-тільки очікуються. Загалом, не взяти для порівняння цей пристрій ми просто не могли.

Накопичувач тестувався із прошиванням версії 2.50.

Plextor M6e 256 ГБ

Цей накопичувач ми вже теж вивчали докладно й от він-те саме повністю відповідає описаному на початку роботи: карта формату M.2 2280 з додатковим адаптером, що дозволяє встановлювати його в слот PCIe x4 і ширше. Формат слоту обраний таким лише з міркувань сумісності – пристрій використовує контролер Marvell 88SS 9183-BNP2 з інтерфейсом PCIe 2.0 x2, тобто теоретичні

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 14 |

можливості стандарту тут розкриваються не повністю. Зате 1 ГБ/с – це набагато швидше, ніж 600 МБ/с в SATA 3.0, а вийшло компактно й недорого:)

Під час тестування використовувалося прошивання версії 1.01.

Plextor M6e Black Edition 256 ГБ

Через кілька місяців після появи M6e у виконанні карт розширення PCIe і M.2, компанія випустила спеціальну версію першого накопичувача – M6e Black Edition. Варто відзначити, що відбулося це одночасно з анонсом M7e, що буде використовувати вже чотири, а не дві лінії PCIe, так що зміст відновлення споконвічно не простежувався. Зрештою, принципової різниці між технічними характеристиками «звичайної» і «чорної» редакцій ні, та й бути не може, оскільки обидві являють собою той самий накопичувач формату M.2 на платі-переходнику. Елементна база однакова, прошивання теж ті самі, практичне дослідження теж підтвердило ідентичність пристроїв.

Так у чому ж різниця і є чи вона взагалі? Насправді є, причому відмінності не тільки зовнішні, але й внутрішні. З першими все просто – M6e Black Edition у куди більшому ступені схожий на закінчений роздрібний продукт, ніж попередник, по якому відразу було видно, що це накопичувач формату M.2 у комплекті із простеньким переходником. В «чорній» редакції ж внутрішній пристрій хоча б приховано гарним радіатором. Виконує той, звичайно, більшою мірою декоративні функції, оскільки «перегріти» накопичувач і без нього в нормальних умовах експлуатації було неможливо (а в ненормальній й радіатор не допоможе), але принаймні це красиво.

Змінився й сам адаптер, причому стосується це не тільки кольору текстоліту. Наприклад, простенькі світлодіодні індикатори раніше були розташовані на задній заглибці слоту, так що були ледве більш ніж марні. У новій моделі характер підсвічування змінився, а сама вона переїхала на верх карти, що особливо оцінять власники корпусів із прозорими кришками. Та й всі інші при бажанні можуть поспростерігати за «життям» накопичувача, оскільки на платі з'явилося роз'єм для підключення корпусного індикатора дискової

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 15 |

активності. При наявності різних накопичувачів (і Мбе, і традиційних SATA) прийде, щоправда, вибирати, на кого саме буде реагувати світлодіод, але вибір – краще, ніж нічого.

Але не варто думати, що всі відмінності лише декоративні – є ще одне, котре може в деяких випадках виявитися корисним, і зв'язано воно із системою живлення. Справа в тому, що накопичувачі формату М.2 від своїх попередників успадкували тільки одну напругу живлення – 3,3 В. У системах, споконвічно розрахованих на їхнє використання, проблем ні, але більшість десктопних карт розширення воліє 12 В, тому до якості живлення на низькій напрузі виробники системних плат нерідко ставляться абияк. Що, у загальному-те, звичайно проблем не викликає, однак старша модифікація Мбе, ємністю 512 ГБ, уже десь на грані виникнення потенційних проблем – адже їй потрібен струм силою 3,5 А, що не так уже мало. А в Black Edition проблема виправлена, оскільки адаптер постачений власним перетворювачем – харчує накопичувач, як і покладено, трьома вольтами, але сам споживає 12. Причому навіть не обов'язково зі слоту розширення – на платі передбачене й типовий SATA-роз'єм живлення, що дозволяє підключити накопичувач безпосередньо до блока живлення, заодно й знизивши навантаження на системну плату.

Загалом, такі от зміни. Не можна сказати, що принципові й необхідні, так що на практиці більшість користувачів ніякої різниці між цими двома пристроями не помітять: як уже було сказано, вони побудовані на однаковій елементній базі, використовують однакові прошивання, можна однаковим образом використовувати додаткове кешовані за допомогою утиліти PlexTurbo 2.0 і т.п. Однак з погляду завершеності й пропрацьованості конструкції оновлена модифікація явно краще. І, що теж немаловажно, гарніше.

Під час тестування використовувалося прошивання версії 1.05, доступна й для звичайних Мбе. Але останній ми перешивати й повторно тестувати не стали: сам-те накопичувач на платі-адаптері той же, так що одержали б і ідентичні

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 16 |

результати. А так у нас є можливість зрівняти дві версії прошивань і подивитися, що змінилося в плані продуктивності.

Kingston Hyper Predator 480 ГБ

Найбільше Predator нагадує оригінальний Мбе, відрізняючись від нього лише в деталях – закінченості Black Edition не спостерігається. Однак для пристрою, встановлюваного усередину комп'ютера, це й неважливо – головне, як воно працювати буде. А на високі швидкісні характеристики можна настроїтися споконвічно: тут застосований контролер Marvell 88SS9293, що має інтерфейс PCIe 2.0 x4 (зважаючи на все, він же буде використовуватися й в Plextor M7e), тобто теоретична пропускна здатність інтерфейсу становить уже не 1, а 2 ГБ/с. OCZ RevoDrive 350, нагадаємо, по цьому параметрі ще краще (знову вдвічі), але отут уже важливим моментом є те, наскільки такі швидкості зможе «переварити» «начинка» пристрою й у яких сценаріях. Помітимо, що сама компанія говорить лише про 1400 МБ/с при читанні даних і 1000 МБ/с при записі, що менше теоретичної межі, але приблизно вдвічі більше, ніж в Мбе.

Залишається тільки згадати, що компанія постачила пристрій кеш-пам'яттю типу DDR 3-1600 ємністю 1 ГБ, ну а чипи властиво флеша тут звичні по багатьох інших пристроях – «поліпшені» 19-нанометрові Toshiba A19 Toggle Mode 2.0. Недоліком їх з погляду продуктивності є ємність кристалів 128 Гбіт, але в пристрої із сумарною ємністю 480 ГБ (тобто з додатковим резервуванням) цим ефектом можна зневажити: кристалів, як правило, уже вистачає, щоб повністю заповнити всі канали контролера.

Накопичувач тестувався із прошиванням версії OC34L5TA.

Методика тестування

Методика докладно описана в окремій статті. Там можна познайомитися з використовуваним апаратним і програмним забезпеченням.

Час доступу

Якщо хто й виділяється (та й то не завжди), то це OCZ RevoDrive 350, де контролер – зовсім не родич іншим випробуваним. Між ними ж розкид значень

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 17 |

на тім же рівні, що забезпечує й проста зміна прошивання. У загальному ж і цілому – було б про що говорити: всі швидкі порівняно з вінчестерами (які бувають повільними, дуже повільними, і дуже-дуже повільними).

Послідовні операції

При таких навантаженнях інтерфейс має значення – властиво, при невеликій пропускнув він взагалі визначальний. От вище – не завжди. Якщо Plextor M6e (без різниці – якої модифікації) чесно вперся приблизно в 800 МБ/с, то Kingston Hyper Predator приблизно вдвічі швидше. Це приблизно відповідає швидкості інтерфейсу (x2 проти x4) і навіть перевершує заявлені значення. А от OCZ RevoDrive 350 «вичавити» максимум зі своїх восьми ліній PCIe не може, практично навіть уступаючи накопичувачу Kingston.

При записі ж він відіграється, перевалюючи на більших блоках навіть за 1400 МБ/с. «Прості» PCIe-накопичувачі працюють повільніше, але в точній відповідності із заявленими характеристиками.

Випадковий доступ

Відновлення прошивання істотно поліпшило показники Plextor M6e, фактично вивівши його на лідируючі позиції. Втім, у випадку Kingston можливий і реванш – з аналогічної причини. Але в загальному й цілому на аналогічні результати здатні й накопичувачі з SATA-Інтерфейсом, тому звертати пильну увагу на цих шаблонах ми не будемо.

Тим більше, що, на відміну від IOMeter, AS SSD ніяких змін зовсім не зауважує.

А те, що контролери Marvell стиском даних не займаються, було відомо вже давно. От і для самого нового контролера компанії це підтвердилося – єдина новина, та й та відносна.

Продуктивність у додатках

Загальний бал в PCMark у всіх приблизно однаково високий. Це ми вже не раз відзначали. Тим більше, компанія додала у свій час у тестовий пакет більше

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 18 |

низькорівневий варіант цих трас, але нас він по зрозумілих причинах не дуже цікавить – низькорівневих тестів і без PCMark 7 досить.

Єдина траса, на якій різниця між випробуваними досягає 10% – запуск додатків. Але й отут результати краще в середні навіть у самого повільного пристрою, яким виявився Kingston Hyper Predator.

З результатів «робітників» шаблонів NASPT нам, як звичайно, цікавий тільки ContentCreation (інші мабуть, можна вже й забирати з тестового набору): оскільки він буває більшою кількістю операцій і читання, і запису, так ще й малими блоками, продуктивність у ньому звичайно виявляється багато меншої, чим теоретично досяжна, але дуже залежної від інтерфейсу підключення. Відзначимо, що для SATA-пристроїв типові швидкості в 60-80 МБ/с, а от PCIe-накопичувачі демонструють куди більше високі. Що ж – те, що їх має сенс використовувати в робочих станціях), цілком логічно. Тепер от і тестами підтверджено.

Робота з більшими файлами

Тут є де розгорнутися інтерфейсу, хоча «вичавити» з нього максимум у районі PCIe x4 і вище вже непросто. Помітимо при цьому, що одержати високі швидкості на пристроях, типу RevoDrive (так само як і із самостійно зібраних RAID-масивів з SSD), можна лише при багатопоточній навантаженню, а от більше «прості» накопичувачі на PCIe-контролерах Marvell демонструють гарні результати й при одному потоці даних.

А отут накопичувач Kingston обігнав OCZ незалежно від кількості потоків. Втім, показово, що другий знову сильно знижує швидкість в однопотоковому режимі й отут уже може конкурувати тільки з SSD Plextor, постачених усього лише двома лініями інтерфейсу.

При читанні одночасно із записом знову з'ясовується, що «прості» сучасний пристрій легко може виступати на рівні з «рекордистами» минулого.

| | | | | | | |
|------|------|-----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № док.ум. | Підпис | Дата | | 19 |

Загальний середній бал

І показово виглядає загальний бал – чотири лінії PCIe при досить простому контролері дозволяють при усередненні результатів по всій сукупності тестів виступати на рівні накопичувачів куди більше складної організації. При цьому, природно, куди більше дорогих. Та й досить громіздких, що не дозволяє використовувати їх повсюдно.

З докладними ж результатами всіх тестів можна познайомитися, скачавши таблицю у форматі Microsoft Excel.

Виводи

Ідея одержати пристрій швидше інших завжди була популярна й серед виробників, і серед користувачів. Останнім, щоправда, воно не завжди було потрібно на практиці, але раз вуж людина готова платити за свої бажання – чому б і немає. Була б пропозиція.

І воно з'явилося досить давно – як тільки було з'ясовано, що стандартні дискові інтерфейси для твердотільних накопичувачів є вузьким місцем. Правда, «знаходження» широкого інтерфейсу виявилось лише частиною справи – виробникам довелося займатися підвищенням швидкості самих пристроїв, щоб повністю скористатися новими можливостями. А маючи в наявності «звичайні» чипи флеш-пам'яті й «звичайні» же SATA-контролери для неї, вирішити завдання можна було тільки багатоплатними накопичувачами із двох/чотирьохканальним внутрішнім RAID0.

Пройшло кілька років, і от тепер з'ясовується, що аналогічний рівень швидкості можна досягти й більше простим способом. Так завжди буває: сучасна приміська електричка їздить швидше, ніж паровози-рекордисти столітньої давнини:) Просто для цього довелося зробити підходящі контролери, але розробляти їх однаково було потрібно, оскільки ринок уже перестали влаштовувати й фізичні габарити накопичувачів традиційного типу. Загалом, у сучасних пристроїв висока не тільки продуктивність, але й універсальність. Старі PCIe-накопичувачі можна було використовувати лише в десктопах, причому

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 20 |

повнорозмірних, а нові M.2 – де завгодно: десктопи, ноутбуки, планшети, міні-ПК... В NUC третього покоління, до речі, замість mSATA установлений слот M.2 саме з інтерфейсом PCIe x4, тобто в цій крихітці можна спокійно використовувати той же Kingston Predator і одержувати швидкості, які раніше досягалися лише за допомогою RAID-масивів.

Така от сувора хода історії. Яка, втім, поки не скасовує звичайні SATA SSD – завдяки більшим розмірам їхньої друкованої плати, накопичувачі великої ємності простіше випускати саме в такому форматі. І дешевше. А от ходові 128, 256 або 512 ГБ згодом можуть залишитися тільки в «картковому» виконанні – так навіть більш універсально.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – це потужна мова програмування, яка проста у вивченні. Він має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис і динамічна типізація Python разом з його інтерпретованим характером роблять його ідеальною мовою для створення сценаріїв і швидкої розробки додатків у багатьох сферах на більшості платформ.

Інтерпретатор Python і обширна стандартна бібліотека доступні у вихідному або двійковому вигляді для всіх основних платформ на веб-сайті Python <https://www.python.org/> і можуть вільно поширюватися. Цей же сайт також містить дистрибутиви та вказівники на багато безкоштовних сторонніх модулів Python, програм і інструментів, а також додаткову документацію.

Інтерпретатор Python легко розширюється за допомогою нових функцій і типів даних, реалізованих у C або C++ (або інших мовах, які можна викликати з C). Python також підходить як мова розширення для налаштовуваних програм.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 21 |

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи зберігання та обробки великих даних побудованої з використання протоколу NVMe.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 22 |

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Протокол Non-Volatile Memory Express (NVMe)

Ми протестували вже далеко не один твердотільний накопичувач із інтерфейсом PCIe, але всі попадавши до нас у руки SSD старанно емулювали «традиційні» дискові пристрої. Що, втім, не заважало їм демонструвати високу продуктивність, тому що, по-перше, SSD по визначенню швидше «механіки», а по-друге (як наслідок), для них уже актуальне питання пропускну́ї здатності інтерфейсу підключення, у чому кілька ліній PCIe однозначно перевершують SATA. Однак чи потрібна масиву комірок пам'яті здатність «прикидатися» апріорі повільним блоковим пристроєм, типу вінчестера або навіть накопичувача на оптичних дисках? Очевидно, що продуктивності проміжні рівні ієрархії не додають: чим більш прямий шлях, тим вище швидкості. Звичайно, з погляду сумісності «стандартні» інтерфейси переважніше, але адже PCIe цю саму сумісність обмежує споконвічно. Тому як тільки мова зайшла про використання цього інтерфейсу, виробники відразу ж задумалися й про відповідний програмний прошарок: щоб у ній не було нічого зайвого для SSD, зате враховувалися всі їхні особливості. Так з'явився інтерфейс NVMe: Non-Volatile Memory Express.

Зараз поступово починається і його просування в масовий сектор: PCIe-накопичувачі нарешті одержали стандарт універсального втілення, придатного для широкого класу комп'ютерів, у вигляді M.2. Однак є сумніву, що на цьому шляху все буде гладко, про що ми сьогодні й поговоримо. А також оцінимо плюси й мінуси NVMe на прикладі пари пристроїв корпоративного класу, де ця технологія застосовується вже майже рік, благо там і проблем із впровадженням менше.

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 23 |

Для початку влаштуємо невеликий екскурс у часи появи перших персональних комп'ютерів. Незважаючи на те, що й сьогоднішні «монстри» деякою мірою сумісні з тодішніми «мастодонтами», підхід до конструкції останніх був зовсім іншим, ніж прийнятий зараз. Наприклад, оригінальна IBM PC не тільки не забезпечувалася вінчестером, але й могла працювати навіть без дисководу – з побутовим магнітофоном. Природно, стандартизувати тоді інтерфейс для підключення швидких накопичувачів не були потреби, та й накопичувачів таких у масовому виконанні особливо не водилося, тому що коштували вони занадто дорого (це зараз флеш-пам'ять продається за ціною менш долара за гігабайт, а колись кілобайт на жорсткому диску коштував дорожче). Втім, уже в PC XT вінчестер з'явився, а необхідний для нього контролер, як і слід було сподіватися, установлювався в роз'єм шини розширення на системній платі.

Якийсь час жорсткі диски в персональних комп'ютерах використовували інтерфейс ST-506/412, де вся керуюча логіка була винесена в контролер комп'ютера, а плата електроніки самого вінчестера містила тільки модулі аналогової обробки й керування шпіндельним двигуном, позиціонером і комутатором головок. Знову ж, час було таке – квапитися було особливо нікуди, і багато дисків підключати теж, так що всіх улаштувало підключення не більше двох вінчестерів, для чого доводилося використовувати аж три кабелі (два у випадку одного накопичувача). Швидкість передачі даних від вінчестера до комп'ютера по інтерфейсі була навіть нижче здатностей тодішніх тихоходів. Загалом, таке от рішення, проте яке працювало й відносно недороге.

Однак продуктивність комп'ютерів росла, що давало їм можливість вирішувати усе більше широке коло завдань, а це, у свою чергу, не могло не позначитися на периферійному встаткуванні: хотілося підключати більше число більше різноманітних пристроїв, що працюють із більше високою швидкістю. ST-506/412 такі потреби «не тяг». Результатом чого були спроби розробити поліпшений інтерфейс жорстких дисків (ESDI) або портувати на персоналки звичний по робочих станціях інтерфейс SCSI, але все скінчилося тим, що

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 24 |

збільшилася пропускна здатність інтерфейсу до 150 МБ/с із можливістю подальшого росту.

У принципі, ще в часи паралельного ATA пропускна здатність стала трохи надлишковою для тодішніх пристроїв, основними з яких продовжували залишатися вінчестери й оптичні накопичувачі. Властиво, немудро, що розроблювачі спантеличилися й іншими питаннями. Зокрема, реальну продуктивність дискових пристроїв можна підвищити шляхом грамотної оптимізації запитів – щоб мінімізувати переміщення головок. Але для того, щоб оптимізувати звертання до диска, треба не поспішати їх виконувати, а створювати черга команд, що потім і з. Такі механізми з'явилися ще до впровадження SATA, але TCQ стандартом не стала, а от NCQ – давно вже стандартна функція для всіх SATA-контролерів. Черга команд звичайно має невелику глибину, оскільки все, що нам потрібно у випадку вінчестера – оптимізувати послідовність «перебору» циліндрів, і черга ця одна, оскільки ніякого внутрішнього паралелізму в масових «механічних» накопичувачах просто немає. Він може з'явитися тільки при використанні декількох дискових пристроїв і об'єднанні їх в RAID-масив, чим і користуються відповідні контролери – причому чипсетні це роблять рідко: хоча продуктивність читання даних у дводисковому масиві RAID1, наприклад, можна подвоїти, використовуючи чергування запитів до дисків, чипсети Intel це не використовують. І навіть залучення RAID-функціональності в чипсетних дискових контролерах не є масовим, а вуж більше складні RAID-контролери й зовсім зустрічаються лише в специфічних нішах. Властиво, саме із цієї причини (більшість комп'ютерів забезпечуються лише одним-двома «послідовними однозадачними» накопичувачами) ніякий паралелізм стандартам підключення дискових пристроїв був не потрібний. А більші «внутрішні» затримки при доступі до інформації, викликані роботою механіки, дозволяли не занадто піклуватися й про накладні витрати на забезпечення роботи інтерфейсу. Так AHCI і працює. І, у загальному-те, довгий час протокол улаштував усіх. Однак прихід твердотільних накопичувачів у масовий сегмент змусив виробників задуматися...

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 26 |

Non-Volatile Memory Express

Для механічних накопичувачів середні затримки при виконанні команд становили порядку 2 мс, а використання флеш-пам'яті дозволяє скоротити цей час приблизно в 20 разів – до 100 мкс. 25-30 мкс із вини драйверів і контролерів просто губляться на тлі першого значення, але на тлі другого вони вже дуже помітні. «Усунення» з ланцюжка SATA-контролера шляхом емуляції його функцій властиво контролером твердотілого накопичувача в «AHCI PCIe SSD» заощаджує більше 10 мкс, а припинення емуляції AHCI і перехід на спеціалізований програмний інтерфейс і зовсім скорочують накладні витрати до п'яти мікросекунд. Повторимося: основним винуватцем затримок продовжує залишатися властиво флеш-пам'ять, але сумарно затримки скорочуються в півтора разу. Відповідно, очікувати такого ж підривного ефекту, як при відмові від механіки, не коштує (все-таки 20 разів – це не 1,5 рази, але й одержати «на порожнім місці» полуторакратне зменшення затримок теж добре. Тим більше що сумарний ефект виявляється кумулятивним – до нього додається й загальне збільшення пропускної здатності інтерфейсу, властиве SSD з підключенням до PCIe.

На чому ще можна вигадати? На збільшенні паралелізму при обробці команд. Як уже було сказано вище, вінчестери обробляють команди послідовно, оскільки блок головок у кожний момент часу може перебувати тільки над одним конкретним циліндром. Усе, що дає черга команд – можливість оптимізувати переміщення головок читання/запису, що теж чимало, однак слабо застосовно до SSD, які, по суті своєї, є паралельними пристроями. Навіть найпростіші контролери підтримують чотири канали флеш-пам'яті, при «традиційному» підході поєднуються в один для повної емуляції AHCI. Але якщо емуляція AHCI нам не потрібна, те всі канали можна вважати незалежними накопичувачами. Відповідно, завданням драйвера й контролера є «розкидування» запитів по каналах, а далі кожний з них буде обробляти свої команди найбільш ефективним образом. Тобто черг повинне бути трохи й, бажано, максимально глибоких – це дозволить вичавити із флеш-пам'яті всі (наприклад, поки одні блоки будуть

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 27 |

перезаписуватися, в ідеалі з інших можна читати дані, не чекаючи закінчення тривалих операцій). Але AHCI, де черга команд усього одна на кожний пристрій, так ще й глибиною в 32 команди, нам тут нічим не допоможе. «Рідні» же для NVMe можливості куди серйозніше: до 64К (65536 в «некомп'ютерних одиницях») черг і до 64К команд у кожній. Зрозуміло, сьогодні такий простір поки не використовується, але запас на майбутнє вселяє оптимізм:)

Звичайно, є в стандарті й інші оптимізації, але головне вже зрозуміло: на відміну від AHCI, що розроблявся для максимальної сумісності з різними пристроями, що вже існували на той момент, NVMe є спеціалізованою розробкою, багато в чому націлений у майбутнє. І не коштує зацикливатися тільки на флеш-пам'яті: є підозри, що деякі можливості протоколу розкриються при освоєнні нових стандартів енергонезалежної пам'яті, особливо – з низькими власними затримками. Але поки різноманіття твердотільних накопичувачів на ринку ні, так що NVMe використовується саме із пристроями на флеші. Для механічних накопичувачів інтерфейс NVMe, мабуть, не потрібний. А є чи зміст переводити на нього хоча б всі SSD? У перспективі – можливо.

Підводні камені сумісності

Повернемося до протоколу AHCI і згадаємо, щоби про нього знаємо. По-перше, з'явився він дуже давно, так що безпосередньо підтримується не тільки сучасними, але й застарілими операційними системами. По-друге, властиво «підтримка» у його випадку досить проста: в ідеалі системі взагалі не потрібно вдаватися особливо функціонування конкретного пристрою, оскільки всі інкапсулюється на рівні контролера. Для комп'ютера такий твердотільний накопичувач із інтерфейсом PCIe – це всього лише черговий дисковий контролер, працювати з якими «учили» ще DOS і BIOS. Відповідно, навіть дуже стара система повинна легко «переварювати» його додавання, благо й на апаратному рівні ніяких особливостей немає.

Що в цьому плані міняє NVMe? Так практично все: адже основною метою його розробки є скасування проміжних рівнів – саме звичних і перевірених часом. Зокрема, прийде забути про завантаження в Legacy-режимі – тільки UEFI.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 28 |

Також можливі проблеми із прошиваннями деяких системних плат, на які ми наткнулися, і навіть із чипсетами – Intel, наприклад, гарантує підтримку NVMe для чипсетів «дев'ятого» сімейства, а от з попередніми можливі варіанти. Крім того, буде потрібно спеціальний драйвер. Підтримка NVMe, втім, уже убудована в Windows 8.1, Windows Server 2012 R2 і Linux починаючи з версії ядра 3.10, але цими системами список використовуваних споживачами не обмежується. А драйверна підтримка більше старих систем залишається на розсуд виробника й може бути обмеженою: приміром, тільки x 64-версії Windows 7, але не x86. Зокрема, із цим ми зіткнулися, одержавши дві SSD корпоративні серії Intel – у підсумку виявилось неможливо протестувати їх по «стандартній методиці» для накопичувачів, тому що на звичайно використовуваному стенді із платою на базі чипсета Z77 вони взагалі не заробили (а якби навіть і визначилися, легше б від цього не стало: ми використовуємо Windows 7 x86, а вона на той момент Intel не підтримувалася). До речі, є й ще одна проблема із впровадженням стандарту: по-перше, контролери з підтримкою NVMe розраховані в основному на PCIe 3.0, а по-друге, раз вуж мова заходить про мінімізацію затримок, максимальна продуктивність досягається при підключенні пристрою безпосередньо до процесора без проміжних хопів у вигляді чипсета. Що із цього треба? Те, що весь потенціал технології NVMe без проблем може бути розкритий тільки на системах з LGA 2011-3 (або, у крайньому випадку, попередньою версією цього сокета), оскільки процесори для цієї платформи в надлишку постачені лініями PCIe 3.0. Що ж стосується масової LGA1150, то там прийдеться «урізати» по пропускній здатності шини відеокарту: у наявності всього 16 ліній PCIe 3.0, а використання SSD в «чипсетних» слотах взагалі протипоказано.

Роз'єм SFF-8639: новий шлях до багатодискових конфігурацій

Проблеми сумісності на цьому не закінчуються. Тобто при бажанні встановити в систему один накопичувач – все просто: досить знайти відповідний слот PCIe (що, втім, як уже було сказано, не завжди просто), і все. Але адже ступінь паралелізму можна збільшити одночасним використанням декількох

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 29 |

визначати тип підключеного пристрою й вибрати відповідний режим роботи (SATA або PCIe), а от у випадку SFF-8639 вся відповідальність за правильність підключення покладає на збирача.

А як тільки роз'єм на пристрої в нас є, і кабелі теж, далі все просто: інший кінець кабелю встромляється в цілком звичайну низькопрофільну плату PCIe. Звичайною справою є використання одного кабелю на парі портів, так що плати можуть бути різні: слот PCIe x8 дозволить підключити саме пари накопичувачів одним кабелем, слот x16 – відповідно, два кабелі й чотири пристрої. У принципі, можна чекати швидкої появи відповідних рознімань безпосередньо на серверних материнських платах, що зробить карти-райзери непотрібними й дозволить використовувати накопичувачі й у компактних серверах.

Переваги перед безпосереднім використанням PCIe-слотів (властивим «побутовим» пристроям) очевидні: компактність (особливо при відмові від райзерів), більше ощадливе використання ліній PCIe (якщо є тільки один слот x16, можна або встановити в нього один SSD «на карті», або підключити до нього чотири накопичувачі), можливість використання стандартних кошиків з гарячою заміною пристроїв. Але, повторимося, застосування SFF-8639 «у побуті» розроблювачами не планується – орієнтація взята на сервери й багатодискові конфігурації. Там, де потрібний одиночний накопичувач, стандартним як і раніше буде залишатися використання «звичайних» карт розширення або (у перспективі) M.2.

3.2 Розробка структурної схеми

Звичайні протоколи SAS призначені для механічних жорстких дисків. Швидкий розвиток флеш-носіїв і постійно зростаюча потреба в поліпшеній продуктивності різних систем позначили недоліки SAS-протоколів. Протоколи визнані слабкою ланкою в розробці СЗД на базі флеш-пам'яти. У порівнянні з жорсткими дисками SAS пропускна здатність і максимальна кількість операцій

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 31 |

уведення-виводу в секунду (IOPS) на SSD значно покращилися, а в деяких випадках затримка була знижена до наднизького рівня 100 мкс. Ємність SSD продовжує рости. Виникає питання: як вирішити проблему слабкої ланки – протоколу передачі? Організація NVM Express розробила стандарт протоколу NVMe, видаливши планувальник уведення-виводу й складні рівні протоколу SCSI у системах жорстких дисків SAS, замінивши їх на легкий стек NVMe.

NVMe дає величезний ріст ефективності, оскільки для кожної операції уведення-виводу потрібно менш половини циклу процесора SAS. Інакше кажучи, NVMe дає у два рази більше IOPS, чим SAS при аналогічному циклі процесора. Недоліки SAS очевидні:

– Більше взаємодій, (потрібне перетворення між PCIe і складним стеком SAS/SCSI, а витрати на обробку досить високі), це утрудняє одержання більшої кількості IOPS. Диски NVMe можуть прямо підключатися до інтерфейсів PCIe, спрощуючи взаємодію протоколів для підвищення продуктивності.

– Протокол SAS підтримує одну чергу, а NVMe – до 65536 (64K) черг, кожна із глибиною до 65536 команд. Множинні черги дозволяють досягти високого рівня паралелізму при обробці, для відповідності високої продуктивності багатоканальних мікросхем з вибором кристала, властивої SSD. Механізми блокування повинні бути додані до одиночних черг SAS у багатоядерних середовищах, тоді як NVMe відразу підтримує механізми без блокування, щоб максимально використовувати переваги паралелізму для настроювання декількох ядер при одночасному зниженні накладних витрат.

NVMe, безумовно, більше ефективний протокол, але відносно новий. Протокол SAS перевірений багаторічною практикою, є досить зрілим і, відповідно, має деякі переваги:

– Власна підтримка багатопортових і двопортових дискових технологій. Протокол стабільний і надійний у багатоядерних системах, у те час як двопортова технологія PCIe як і раніше має потребу в доробці.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 32 |

– Гарна підтримка операцій «гарячої» заміни. Механізми уведення-виводу не дозволяють несправностям поширюватися по всій системі. Додавання функції «гарячої заміни» в PCIe технологічно складно, що означає необхідність нагромадження інформації, перш ніж буде забезпечена висока надійність протоколу.

– Мережа SAS гнучка й легко масштабується. Особливості мережного розгортання були повністю враховані із самого початку розробки протоколу SAS. Опора протоколу на обмін маршрутами спрощує наступне розширення. Навпроти, масштабованість PCIe досить обмежена, коли мова йде про глибину каскаду й загальну кількість підтримуваних пристроїв.

Системи зберігання SSD на основі SAS використовуються у звичайних високопродуктивних середовищах, у те час як системи зберігання на базі NVMe широко використовуються в сценаріях, чутливих до затримок. SAS буде використовуватися ще довгий час, адже NVMe має певні особливості обслуговування.

Аналітики прогнозують, що поставки SSD перевищать обсяги SAS в 2019 році, і цей розрив помітно збільшиться із часом. PCIe SSD є основним напрямком розвитку, для якого найкраще підходить протокол NVMe. Прискорювані темпи розгортання ЦОД і хмарних рішень також підвищує привабливість NVMe SSD для сценаріїв гібридних додатків. Для високошвидкісних операцій з кеш-пам'яттю використовуються нові носії SCM, такі як 3D XPoint. Надвисокі вимоги до продуктивності можуть бути реалізовані тільки за рахунок використання ліній PCIe, що стимулює подальшу розробку й використання стека NVMe. Крім використання в локальних SSD-модулях PCIe, організація NVM Express випустила в червні 2016 року специфікацію NVMe over Fabrics, що дозволило перенести стек на протоколи RDMA/FC і тим самим забезпечити високопродуктивне рішення для віддаленого доступу до твердотільних накопичувачів і вирішити проблему спільного використання міжсайтових і локальних ресурсів SSD.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 33 |

NVMe – це новий протокол, заснований на технології PCIe. Незважаючи на більші успіхи, досягнуті з погляду продуктивності, є ряд проблем, пов'язаних з конструкцією дисків і систем.

NVMe SSD являє собою тип жорсткого диска, тому він повинен мати можливості швидкої заміни без переривання обслуговування. Однак PCIe не має надійні можливості «гарячої» заміни, особливо коли мова заходить про раптову «гарячій» установці PCIe. У результаті деякі налаштування можуть привести до переривання обслуговування й помилок системи. Крім того, СЗД корпоративного класу вимагають високої надійності й, отже, дубльованої схеми контролерів як мінімум, у той час як реалізація двопортової технології на основі PCIe у конструкції SSD представляє певні складності. Деякі з дисків, що випускаються, SSD не мають убудованих двопортових PCIe (наприклад, ряд SSD NVMe від Intel, які використовують внутрішній PCIe-комутатор для реалізації дубльованих портів. Використання таких спрощених схем захисту неминуче приведе до зниження надійності системи.

Ще одним викликом для NVMe СЗД є складність проведення оптимізації на системному рівні. Деякі думають, що використання системи NVMe як простий накопичувач NVMe SSD дає величезні переваги в продуктивності системи. Однак не всі так просто. Архітектура системи уведення-виводу вимагає доробки, щоб зробити її більше підходящою для паралельної обробки в багатоядерних конфігураціях. Така доробка дозволить повністю використовувати переваги NVMe.

Система зберігання та обробки великих даних побудованої з використання протоколу NVMe, реалізована в даній роботі побудована на встаткуванні Huawei. Huawei початку вивчати флеш-технології в 2005 році. Дванадцять років накопиченого досвіду й безперервних інвестицій дозволили компанії одержати численні основні патенти в цій області, і більшість із них ставляться до твердотільних накопичувачів. Сьогодні Huawei є єдиним самостійним виробником операційних систем зберігання даних, контролерів і твердотільних

накопичувачів, що дає їй єдину у своєму роді можливість здійснювати наскрізну оптимізацію флеш-технологій. Клієнти можуть насолоджуватися відмінними можливостями послуг обробки даних, привнесеними революцією в області флеш-технологій, використовуючи комплексні продукти й рішення, надавані Huawei, не говорячи вже про власної відкритий екосистемі компанії. Завдяки цілковитому портфелю продуктів і рішень у галузі й участі у відкритій екосистемі, Huawei дозволяє клієнтам використовувати переваги технологій послуг обробки, які стали можливими завдяки революції в області флеш-пам'яті.

SSD-накопичувачі Huawei підтримують двопортову технологію. Два порти PCIe 3.0x2 незалежні друг від друга. Це дозволяє усунути взаємні перешкоди, забезпечуючи при цьому поліпшену працездатність у процесі відновлення системи й обробки виключень. Дублюванням контролера також забезпечується підвищена надійність системи.

Обробка виключень у конфігураціях PCIe досить складна, особливо коли справа доходить до забезпечення надійності системи при несподіваному додаванні пристрою. Huawei уже багато років інвестує в дослідження PCIe і має максимально повне подання про керування каналами, обробці виключень і можливостях «гарячої заміни» для підвищення надійності системи, тому що заміна диска може виконуватися без переривання обслуговування й помилок.

При виконанні «гарячої» заміни NVMe SSD переводить PCIe у несправний стан. SSD можуть бути витягнуті в будь-який час і будь-який спосіб у системі OceanStor Dorado V3 Huawei, що є явною перевагою для систем NVMe.

У конструкції диска PCIe передбачаються наступні варіанти використання:

– При стандартному процесі обробки помилки AER будуть передаватися доти, поки процес уведення-виводу не буде зупинений на обслуговується стороні, що, що може привести до шквалу помилок і, у свою чергу, сильно ускладнити процедуру обробки. Huawei відключає AER на диску PCIe, після чого

відключаються процеси вводу-виводу. У результаті поширення шквалу помилок зупиняється.

– Якщо операція виймання виконується не досить швидко, то канал відновлюється й помилки поширюються у зворотному напрямку. Для цього випадку в системі Huawei передбачене швидке відключення каналів диска PCIe у процесі обробки.

– У випадку виявлення несподіваного «гарячого» додавання пристрою запуск функції DPC (відключення нижчестоящого порту) може бути ініційований іншими критичними помилками, хоча помилка «Несподіване відключення» не виникає. Відключення каналу може пройти швидше, і повернення пакетів, відправлених на верхній рівень, відбувається набагато швидше, ніж при стандартних процесах обробки.

– Навіть при повільному вийманні дисків функція швидкого роз'єднання лінії зв'язку допоможе забезпечити надійність системи.

У розробці систем NVMe компанії Huawei допомагає багатий досвід роботи з жорсткими дисками, в основі якого лежать всебічний моніторинг працездатності, прогнозування життєвого циклу, моніторинг несправностей апаратного й програмного забезпечення, а також засобу відновлення й ізоляції носіїв, що працюють у цих оточеннях. Все це допомагає згладити процес розгортання NVMe. Запатентована функція включення/вимикання живлення через роз'єм P3 забезпечує ефективне рішення для усунення апаратних несправностей і більшості програмних збоїв на SSD, що ефективно знижує частоту відмов.

Для використання високої продуктивності NVMe необхідна оптимізація на рівні системи. Huawei розробила набір механізмів визначення черговості операцій вводу-виводу. Ці механізми повністю видаляють м'ютекси (блокування) на рівні диска із трактів уведення-виводу, щоб уникнути конфліктів при блокуванні черги. Даний підхід істотно поліпшує продуктивність обробки на стороні сховища. Приклад схеми визначення черговості операцій уведення-

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 36 |

виводу наведений нижче; показані кілька потоків і черг (така схема дозволяє домогтися максимальної продуктивності).

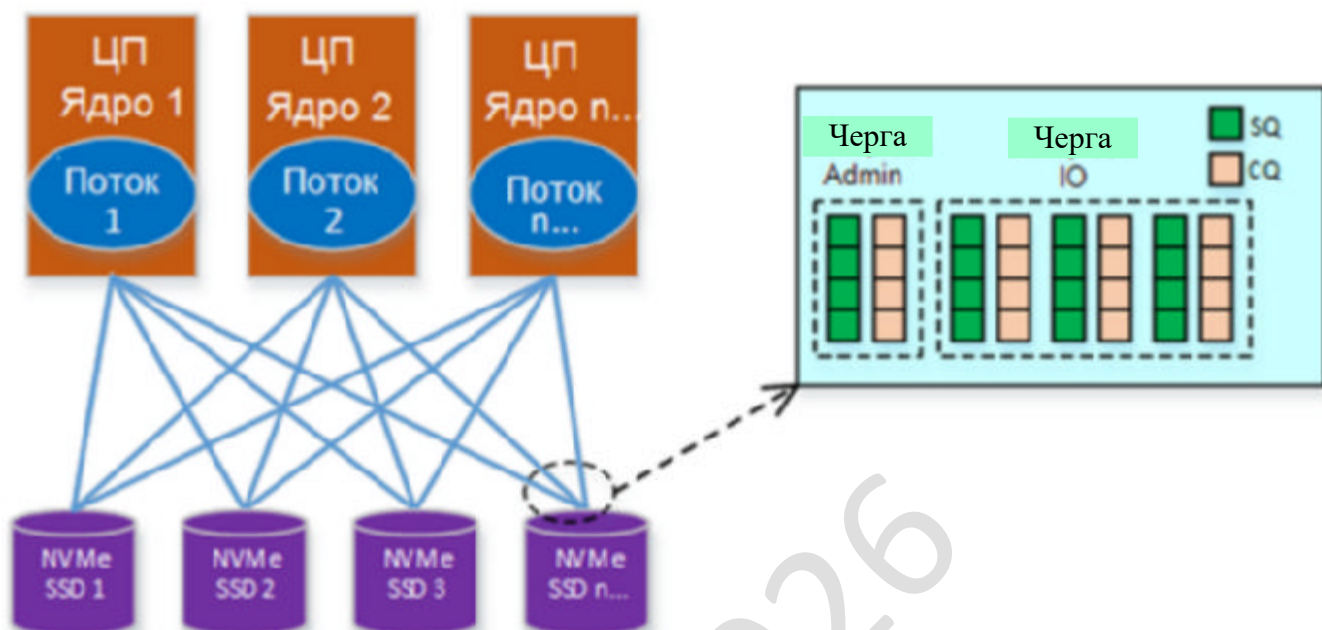


Рисунок 3.1 – Структурна схема системи

Система зберігання та обробки великих даних побудованої з використання протоколу NVMe побудована з використанням Huawei OceanStor Dorado V3 на базі флеш-накопичувачів призначена для підтримки критично важливих важко навантажених додатків організацій будь-якого масштабу. Масиви забезпечують продуктивність до 4 мільйонів IOPS при збереженні прогнозованого й стабільного рівня затримки в 0,5 мс. Організації прискорюють роботу своїх основних процесів при забезпеченні повної надійності системи. OceanStor Dorado V3 працює за протоколом NVMe, а недавно розроблена апаратна архітектура підтримує двопортові SSD-накопичувачі NVMe і можливості «гарячої» заміни в непередбачених ситуаціях. Недавно розроблена паралельна програмна архітектура забезпечує максимальні експлуатаційні переваги NVMe.

Huawei повністю готова до All-flash NVMe і продовжує вкладати кошти в розвиток систем зберігання на базі NVMe. Huawei представила новий носій SCM

для ще більшої продуктивності. Випуск специфікації NVMe over Fabrics забезпечує глобальний спільний доступ до ресурсів SSD, а надання зовнішніх інтерфейсів NVMe допомагає оптимізувати апаратну й програмну архітектуру системи, підвищуючи конкурентоспроможність сховищ All-flash у новій ері.

3.3 Розробка функціональної схеми

Для реалізації системи зберігання та обробки великих даних побудованої з використання протоколу NVMe необхідно створити імітаційну модель. Для організації моделі мережі за найпростішою схемою потрібен набір з 14 об'єктів (рисунок 3.7). На рисунку 3.7 наведена функціональна схема розробленої системи, з рисунка 3.7 бачимо, що для реалізації імітаційної моделі встановлюється два хоста та два маршрутизатора системи зберігання та обробки великих даних побудованої з використання протоколу NVMe, через які йде трафік. Зафарбована область позначає топологічні елементи мережі (канали й маршрутизатори системи зберігання та обробки великих даних побудованої з використання протоколу NVMe).

На функціональній схемі вказані усі елементи, які необхідні для передачі пакета у мережі системи зберігання та обробки великих даних побудованої з використання протоколу NVMe з використанням транспортного протоколу ARTCP. До них відносяться:

- об'єкти протоколу ARTCP та CBR;
- хости;
- симплексні канали передачі даних;
- інтерфейси по яким відбувається з'єднання між маршрутизатора системи зберігання та обробки великих даних побудованої з використання протоколу NVMe;
- маршрутизатори системи зберігання та обробки великих даних побудованої з використання протоколу NVMe;

– комутуюче поле у маршрутизаторі системи зберігання та обробки великих даних побудованої з використання протоколу NVMe (таблиця маршрутизації).

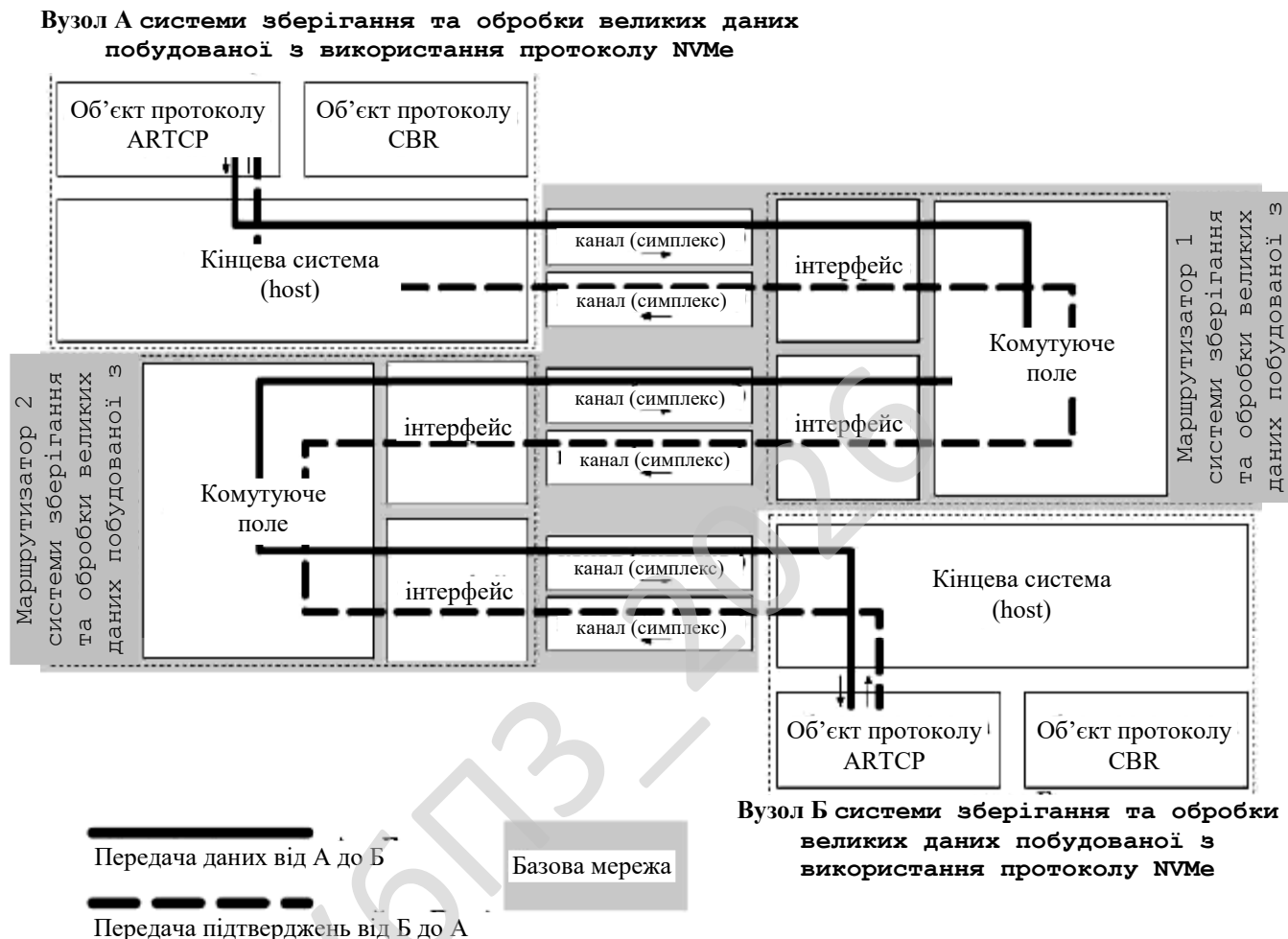


Рисунок 3.2 – Функціональна схема системи

У системі зберігання та обробки великих даних побудованої з використання протоколу NVMe, використовується протокол ARTCP. Наведемо його опис.

Використовуваний у даній роботі, удосконалений протокол Adaptive Rate Transmission Control Protocol (ARTCP), позичає деякі механізми від протоколу TCP. В ARTCP повністю переглянутий алгоритм керування потоком, що і

відрізняє його від TCP. Разом з тим, запропонований протокол може забезпечувати сумісність із TCP.

Аспекти новизни протоколу ARTCP

ARTCP відрізняється від стандартного TCP тим, що сегменти відправляються в мережу системи зберігання та обробки великих даних побудованої з використання протоколу NVMe не у вигляді сплеску в межах вікна, а розділені часовими проміжками, тривалість яких визначається поточним значенням швидкості. Швидкість потоку регулюється не розміром змінного вікна, а значенням швидкості, зміною якої здійснюється адаптація алгоритму відповідно до умов. Механізм ковзного вікна в ARTCP застосовується тільки для запобігання переповнення буферів одержувача. На розмір вікна в ARTCP не накладено ніяких обмежень у жодному з режимів роботи. Вікно обмежене лише наявністю буферного простору в одержувача, тому для одержувача рекомендується встановлювати вікно на максимальний розмір.

У випадку, коли одержувач обмежений у буферному просторі, ARTCP буде поводитися як звичайний протокол TCP, обмежений вікном. Таким чином, протокол ARTCP сполучить у собі метод ковзного вікна для регулювання керування потоком від краю до краю й метод контролю швидкості для підстроювання під ПрЗд проміжних вузлів з'єднання.

Другою найважливішою відмінністю ARTCP є те, що як сигнал про стан перевантаження або наявності додаткових ресурсів у мережі використовуються темпоральні характеристики потоку – вимір шпаруватості потоку в одержувача й зміни часу RTT. Втрата пакета ніяк не відбивається на роботі ARTCP крім здійснення ретрансляції загубленого пакета механізмом корекції помилок. Як і в TCP об втрати пакета повідомляють дві можливих події: спрацьовування ТПП або послідовне одержання двох підтверджень тих самих даних.

Таким чином, у порівнянні зі своїм попередником, ARTCP має наступні переваги:

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 40 |

1. ARTCP не потрібно доводити мережу до стану перевантаження, щоб визначити доступну частку ПрЗд, тому виключені втрати пакетів пов'язані із цим процесом.

2. ARTCP істотно знижує вимоги до міжмережних пристроїв системи зберігання та обробки великих даних побудованої з використання протоколу NVMe. По-перше, для нормального функціонування даного протоколу потрібен менший об'єм буферного простору, ніж для TCP, оскільки режим передачі є згладженим. По-друге, ARTCP не вимагає й не залежить від наявності яких або механізмів диспетчеризації або керування чергами, таких як RED або WFQ.

3. ARTCP не інтерпретує втрату пакета як ознаку перевантаження мережі, використовуючи замість цього темпоральні характеристики потоку. Тому ARTCP повинен особливо ефективно працювати в системах бездротового зв'язку, там де використання TCP неефективно.

4. На відміну від TCP новий протокол не покладається цілком на потік підтверджень у зворотному напрямку для синхронізації процесу передачі. У зв'язку із цим можлива реалізація ARTCP з меншою частотою підтверджень, що не обмежувала б швидкість в асиметричних системах.

Евристика в основі алгоритму ARTCP

Аналіз робіт в області транспортних протоколів, дозволив укласти, що недоліки протоколу TCP досить істотні і є наслідком самого алгоритму, що лежить в основі протоколу TCP. Тому модифікація TCP без заміни його основних алгоритмів не може привести до істотного поліпшення характеристик протоколу.

Тому в цій роботі було вирішено створити вдосконалений алгоритм транспортного протоколу, що залишається, однак, повністю сумісним з архітектурою TCP/IP.

Для того щоб усунути недоліки, властиві TCP, необхідно було знайти спосіб одержання інформації про стан мережі, відмінний від застосування в цих цілях втрат сегментів. Найбільше добре на роль індикатора стану мережі підходять часові характеристики потоку: час RTT і міжсегментні інтервали. З

використанням міжсегментних інтервалів можна також визначити частку ПрЗд каналу. Для цього потрібно запам'ятовувати міжсегментні інтервали потоку у відправника й вимірювати їх в одержувача. Порівняння значень інтервалів характеризує стан мережі, а мінімальне значення вимірюваних інтервалів в одержувача дозволяє визначити доступну частку ПрЗд.

Таким чином, виходить наступна схема: установка швидкості потоку відправником за допомогою ретельної диспетчеризації сегментів, вимір швидкості прибуття потоку в одержувача й передача цієї інформації відправникові разом з іншою контрольною інформацією. Різниця старого й нового значень швидкості відправлення потоку ARTCP на кожному кроці задається випадковою змінною, однак, при наявності сигналу про перевантаження мережі ймовірність зниження швидкості перевищує ймовірність її збільшення на кожному новому кроці.

Параметри й змінні

Нехай τ часовий інтервал між послідовними трансляціями пакетів. Задача функції диспетчеризації сегментів у тому, щоб затримувати відправлення чергового сегмента на час τ після початку передачі попереднього сегмента. Позначимо всі змінні, відносні до відправника індексом s , і r – відносні до одержувача. Отже, τ часовий інтервал між моментами початку відправлення в мережу системи зберігання та обробки великих даних побудованої з використання протоколу NVMe сегмента $i+1$ і $i-20$, а τ_r – інтервал між послідовно прибулими до одержувача сегментами.

Нехай швидкість каналу зв'язку, безпосередньо до якого підключений відправник – R_{ls} , тоді час який потрібен на відправлення одного сегмента (з моменту початку передачі до моменту її закінчення) $t_{ls} = S / R_{ls}$, де s – розмір переданого сегмента.

Очевидно, що максимально можлива швидкість потоку:

$$R_s^{\max} = R_{ls} = S / \tau_s^{\min}. \quad (3.1)$$

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 42 |

Мінімальне значення міжсегментного інтервалу в цьому випадку буде τ_s^{\min} , коли пакети відправляються в мережу системи зберігання та обробки великих даних побудованої з використання протоколу NVMe без затримок з максимальною швидкістю каналного рівня. Шляхом зміни τ у межах $[\tau^{\min}, \infty)$, ARTCP може контролювати швидкість потоку в межах $[R_s, 0)$. Ілюстрація принципу керування швидкістю наведена на рисунку 3.1.

Затримка відправлення готових сегментів виробляється за допомогою системного таймера. Наприклад, для повного використання ПрЗд каналу в 512 Кб/с при розмірі сегмента в 1000 байт кожний сегмент необхідно відправляти із затримкою $\tau = 0.015625$ с, щоб швидкість потоку склала 64 пакета/с.

У таблиці 3.1 наведений список параметрів і змінних використовуваних алгоритмом керування потоком протоколу ARTCP.

Формат повідомлення

Формат повідомлень використовуваних ARTCP може в точності збігатися з форматом пакета TCP. Стандарт TCP [4] передбачає наявність додаткових полів у заголовку сегмента між стандартним заголовком і полем даних. ARTCP може передавати додаткову інформацію в цих полях, що буде гарантувати сумісність із TCP. Усього протокол ARTCP вимагає використання лише двох нових полів: значення попереднього порядкового номера "PS" у напрямку від відправника до одержувача й значення шпаруватості "TI" у напрямку від одержувача до відправника. Значення "TI" можна передавати у вигляді опції часової мітки [6], а значення "TI" вимагає поля, що дозволяє помістити порядковий номер сегмента.

Таблиця 3.1 – Список параметрів і змінних

| | |
|----------|------------------------------------------------------------------------------------------------------------------------|
| s | Розмір кадру каналного рівня утримуючий сегмент |
| τ_s | Часовий проміжок між послідовними трансляціями сегментів (від першого біта до першого біта) |
| τ_R | Часовий проміжок, обмірюваний між послідовними моментами прибуттями сегментів (від останнього біта до останнього біта) |
| $R_s(t)$ | Швидкість потоку, установлювана відправником |



Рисунок 3.3 – Діаграма взаємодії процесів

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо реалізацію бакалаврської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень.

Було створено блок-схеми роботи системи. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується.

Блок-схеми показують весь процес роботи системи з підсистемами та частково доказують правильність вибраних проектних рішень. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає високого рівня декомпозиції задач на класи.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підсистеми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 47 |



Рисунок 4.1 – Блок-схема основної програми

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

Було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу); Діаграма класів.

Діаграма діяльності. Це візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів.

Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Діаграма прецедентів це діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 49 |

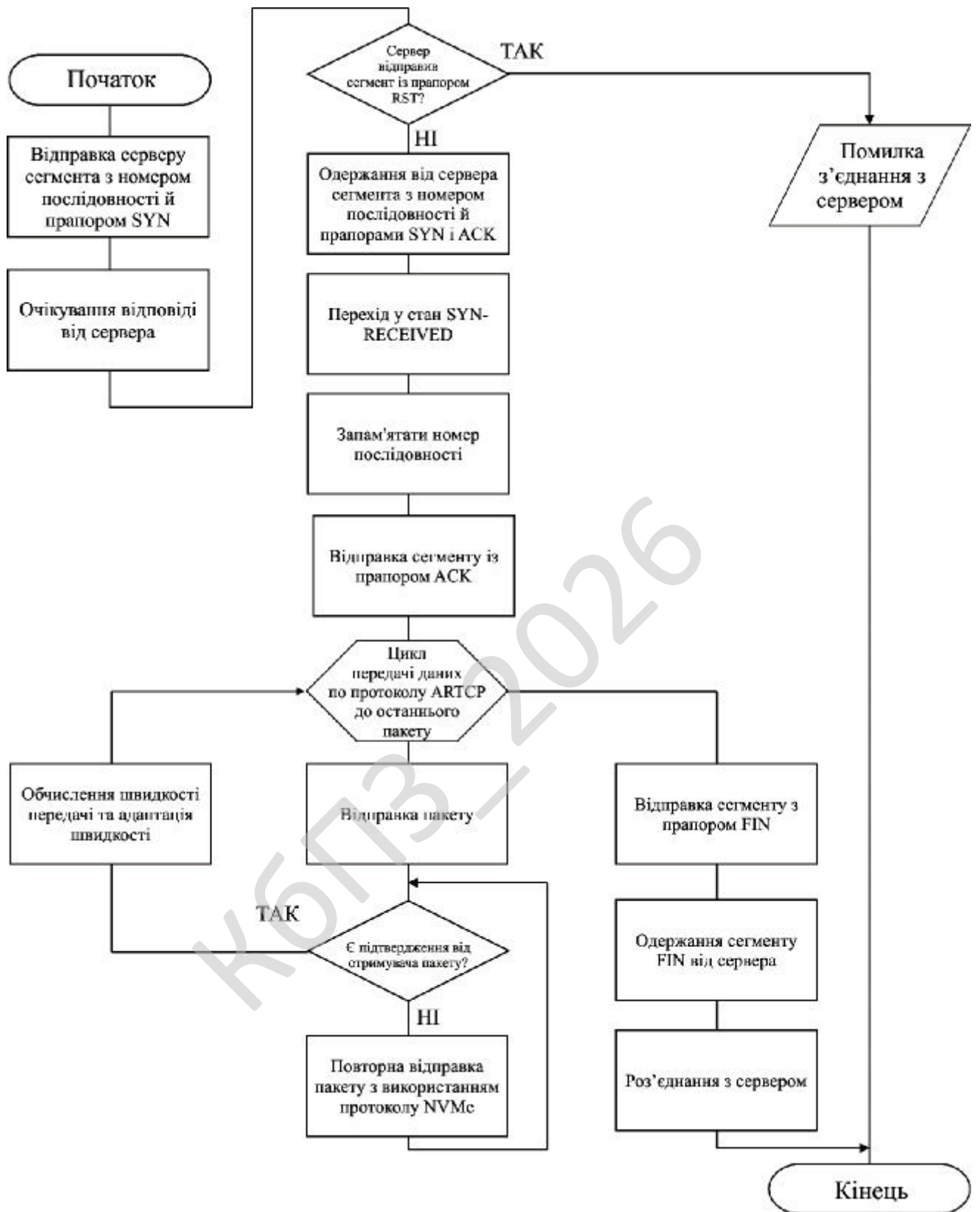


Рисунок 4.2 – Блок-схема роботи підпрограми

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проєктована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором.

При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

- асоціації (association relationship);
- включення (include relationship);
- розширення (extend relationship);
- узагальнення (generalization relationship).

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме — за допомогою відношень включення, розширення і узагальнення.

Відношення асоціації – одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується при побудові всіх графічних моделей систем у формі канонічних діаграм.

Включення (include) у мові UML – це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності (dependency) є таке відношення між двома елементами

моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

Відношення розширення (extend) визначає взаємозв'язок базового варіанта використання з іншим варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов.

Діаграма класів це статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення.

Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.

Діаграма класів (class diagram) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

В UML існують наступні типи зв'язків які використовуються у діаграмі класів: Асоціації; Агрегація; Композиція.

Асоціації це якщо між двома класами визначена асоціація, то можна переміщатися від об'єктів одного класу до об'єктів іншого. Цілком припустимі випадки, коли обидва кінці асоціації відносяться до одного і того ж класу. Це означає, що з об'єктом деякого класу дозволено зв'язати інші об'єкти з того ж класу. Асоціація, що зв'язує два класи, називається бінарної. Можна, хоча це рідко буває необхідним, створювати асоціації, що зв'язують відразу кілька класів. Графічно асоціація зображується у вигляді лінії, що з'єднує клас сам з собою або з іншими класами.

Асоціації може бути присвоєно ім'я, яке описує природу відносини. Зазвичай ім'я асоціації не вказується, якщо тільки ви не хочете явно задати для неї рольові імена або у вашій моделі настільки багато асоціацій, що виникає

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 52 |

Композиція має жорстку залежність часу існування екземплярів класу контейнера та примірників містяться класів. Якщо контейнер буде знищений, то весь його вміст буде також знищено. Графічно представляється як і агрегація, але з зафарбовані ромбиком.

Jira – була використана комерційна система відслідковування помилок, призначена для організації взаємодії з користувачами, хоча в деяких випадках використовується і для управління проектами. Розроблено компанією Atlassian, є одним з двох її основних продуктів (поряд з вікі-системою Confluence). Має веб-інтерфейс.

Назва системи отримано шляхом усічення слова «Gojira» – Японського імені монстра Годзилла, що, в свою чергу, є відсиланням до назви конкуруючого продукту – Bugzilla; створювалася в якості заміни Bugzilla і багато в чому повторює її архітектуру. Система дозволяє працювати з декількома проектами. Для кожного з проектів створює і веде схеми безпеки і схеми оповіщення.

До версії 3.13.5 (включно) розрізнялися редакції Enterprise, Professional і Standard, після – Залишилася тільки редакція Enterprise (для великих організацій).

Система заснована на Java EE і працює на кількох популярних системах управління базами даних і операційних системах.

Основний елемент обліку в системі – завдання (ticket або issue). Завдання містить назву проекту, тему, тип, пріоритет, компоненти і зміст. Завдання може бути розширена додатковими полями (також і нові призначені для користувача поля можуть бути визначені), додатками (наприклад – Фотографіями, скріншотами) або коментарями. Завдання може редагуватися або просто змінювати статус, наприклад, з «відкритий» в «закритий». Які переходи між станами можливі, визначається через настроюється потік операцій. Будь-які зміни в задачі записуються в журнал.

Jira має велику кількість можливостей конфігурації: для кожної програми може бути визначений окремий тип завдання з власним workflow, набором статусів, одним або декількома видами уявлення (screens). Крім того, за

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 54 |

допомогою так званих «схем» можна визначити для кожного індивідуального Jira-проекту власні права доступу, поведінку і видимість полів і багато іншого.

Завдяки універсальному підходу можна пристосувати Jira для багатьох непрофільних завдань, наприклад, керування вимогами, керування ризиками, аж до реалізації невеликої системи бронювання, автоматизації процесу рекрутингу.

Для інтеграції з зовнішніми системами підтримує інтерфейси SOAP, XML-RPC і REST. Поставляється із засобами інтеграції з такими системами управління версіями, як Subversion, CVS, Git, Clearcase, Team Foundation Server, Mercurial і Perforce.

Існують доповнення, що дозволяють вбудувати Jira в інтегровані середовища розробки, в тому числі Eclipse і IntelliJ IDEA. Перекладена багатьма мовами, включаючи українську, англійську, японську, німецьку, французьку, іспанську.

Для сторонніх розробників надаються кошти розробки розширень системи – плагінів. Розробники розширень можуть викладати плагіни для продажу на спеціальний розділ сайту Atlassian.

Є комерційним продуктом, який може бути ліцензований для роботи на локальному сервері або доступний в якості віддаленого додатки. Ціноутворення залежить від максимального числа користувачів, при цьому близько \$50 за користувача для локального і \$7 на місяць за користувача для віддаленого доступу є типовими цінами.

Для академічних і комерційних клієнтів доступний повний вихідний код під ліцензією розробника.

Для проектів з відкритим вихідним кодом Atlassian надає спеціальну безкоштовну ліцензію при дотриманні наступних правил:

- проект використовує ліцензії, схвалені Open Source Initiative;
- Вихідний код проекту доступний для скачування;
- у проекту є публічно доступна веб-сайт;
- програмне забезпечення від Atlassian є на веб-сайті проекту.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 55 |

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у бакалаврської дипломної роботі програмне забезпечення системи зберігання та обробки великих даних побудованої з використання протоколу NVMe. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Функціональних кнопок ПЗ.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Верхнього меню: Файл; Генерація пакету; Параметри; Довідка.
- Розділу обрання групи.
- Розділу виведення результату роботи системи.
- Функції представлені у графічному вигляді (іконки).

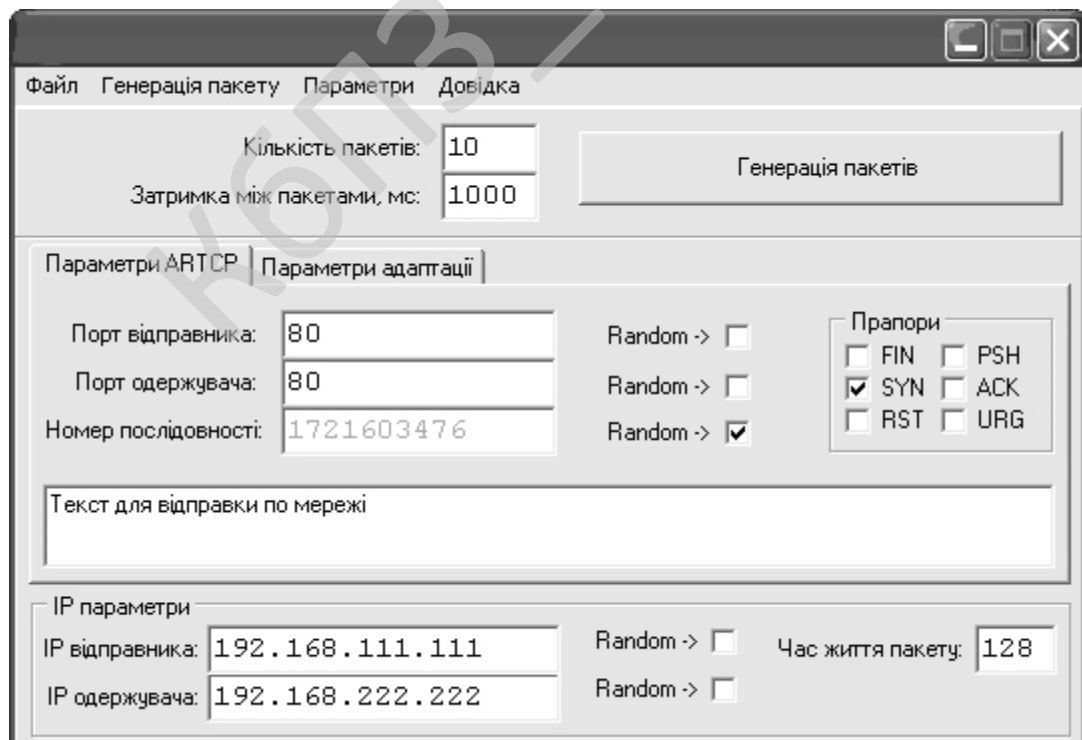


Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

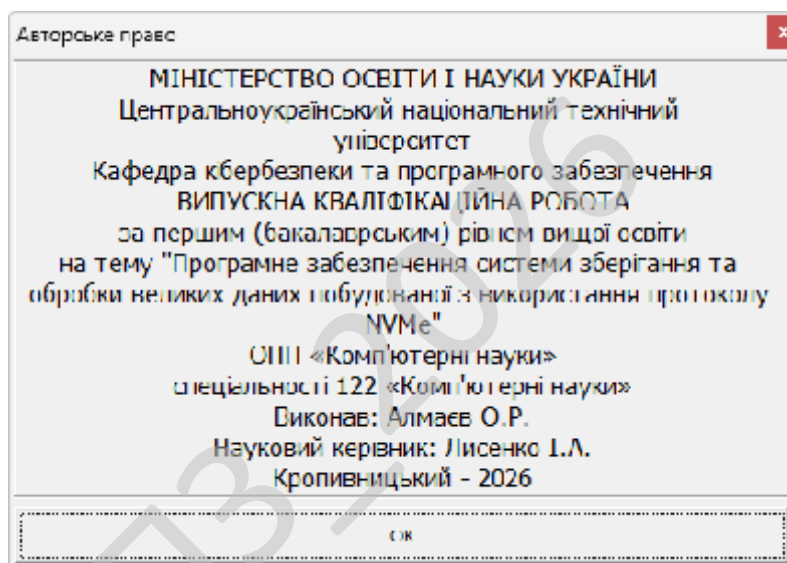


Рисунок 5.2 – Авторське право

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку

виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

- Випуск.
- Встановлення та активація.
- Деактивація.
- Адаптація.
- Обновлення.
- Вмонтування.
- Відстежування версій.
- Видалення.
- Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 60 |

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в IT рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

- Гарантується перевірка всіх незалежних маршрутів програми.
- Знаходяться гілки True, False для всіх логічних рішень.
- Виконуються всі цикли (у межах їхніх кордонів та діапазонів).

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 61 |

– Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

– Кількість незалежних маршрутів може бути дуже велика.

– Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.

– У програмі можуть бути пропущені деякі маршрути.

– Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

– Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.

– Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

– При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

– Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

Обрано умови розповсюдження – proprietary software.

Програмне забезпечення, на яке зберігаються як немайнові, так і майнові авторські права. Отримавши або придбавши таке програмне забезпечення, користувач отримує обмежені права користування ним: може бути заборонено або закрито доступ до коду (вивчення), внесення змін, тиражування, розповсюдження та перепродаж. Програмне забезпечення вважається власницьким, якщо наявне хоча б одне з перелічених обмежень.

Найчастіше основним методом захисту майнових прав на власницьке ПЗ, поза ліцензійною угодою, власник обирає закриття сирцевого коду, захищаючи свій продукт від модифікації і вбудовуючи системи обмеження користування

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 62 |

через авторизацію. Таке програмне забезпечення називається закритим. Проте, код власницького продукту може бути і відкритим, але власник може обмежити права користувача умовами користувацької ліцензії.

Власницьке програмне забезпечення та комерційне програмне забезпечення не є синонімами — власницьким може бути і безплатне (тобто, некомерційне) програмне забезпечення.

На противагу власницькому ПЗ існує вільне програмне забезпечення, автори і власники якого дозволяють вивчати, модифікувати і поширювати свій продукт. Саме визначення власницького програмного забезпечення виникло в результаті діяльності громадського руху вільного програмного забезпечення (представленого Фондом вільного програмного забезпечення та іншими організаціями) і осмислення умов свободи користування програмами. Визначенням власницького програмного забезпечення є не невідповідність хоча б одній з базових умов вільного програмного забезпечення. Сама назва власницьке ПЗ підкреслює визначальне значення власника у способі використання і можливостях розвитку цього програмного забезпечення.

КБПЗ-2026

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | VKPB-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 63 |

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи зберігання та обробки великих даних побудованої з використання протоколу NVMe.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем зберігання та обробки великих даних побудованої з використання протоколу NVMe.

– Досліджена система зберігання та обробки великих даних побудованої з використання протоколу NVMe.

– На основі отриманих результатів досліджень створена програмна реалізація системи зберігання та обробки великих даних побудованої з використання протоколу NVMe.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання зберігання та обробки великих даних побудованої з використання протоколу NVMe.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 64 |

системи зберігання та обробки великих даних побудованої з використання протоколу NVMe. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм AES.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 65 |

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 1». Cisco Press. 2020. – 848 p.
2. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 2 Premium Edition eBook and Practice Test». Cisco Press. 2020. – 624 p.
3. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.
4. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
5. Ramon Nastase «Computer Networking: The Beginner’s guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
6. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
7. Вінтенко Б., Смірнов О., Миронець І., Смірнова Т., Смірнов С. «Імітаційна модель шляхів вхідних даних комп’ютерної інтелектуальної системи підтримки оператора енергоблоку АЕС». *Комбінаторні конфігурації та їхні застосування: Матеріали XXVII Міжнародного науково-практичного семінару, присвяченого 125-річчю Національного університету «Запорізька політехніка» (Запоріжжя-Кропивницький-Київ, 4-6 червня 2025 р.)*. Запоріжжя: НУ «Запорізька політехніка», 2025. С.82-91.
8. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Drieiev, O., Smirnov, O., Dorenskyi, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». *International Review on Modelling and Simulations* 18 (1), 2025. pp. 32-42.
9. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 66 |

10. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

11. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

12. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

13. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

14. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

15. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

16. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings* Volume 3156, 2022, Pages 390-399.

17. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 67 |

18. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

19. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

20. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

21. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.

22. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

23. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

24. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable*

Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

25. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.

26. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

27. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 366-379.

28. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 633-645.

29. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». *International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019*; Odessa; Ukraine; 9-13 September 2019. P.22-28.

30. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

31. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of*

Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

32. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

33. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.

34. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019*, P. 395-399.

35. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

36. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 347-352.

37. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», *CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019*, Pages 618-629.

38. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», *Telecommunications and Radio Engineering*. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 70 |

39. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95

40. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки*. №4. С. 103-110. 2020.

41. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», *Кібербезпека: освіта, наука, техніка*. № 3(7). С. 43-62. 2020.

42. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

43. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у *Кібербезпека та інформаційні технології: монографія*. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

44. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». *Центральноукраїнський науковий вісник. Технічні науки*. № 2(33). с. 161-172, 2019.

45. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

46. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 71 |

D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

47. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

48. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. Центральнуукраїнський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.

49. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральнуукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

50. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

51. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

52. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

53. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Алгоритми формування безлічі маршрутів передачі метаданих у антивірусні хмарні системи. Збірник наукових праць "Системи обробки інформації". - Випуск 5 (142). - Х.: ХУПС - 2016. - С. 148-152.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-122.26.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 72 |