

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 флеш-масивів Pure Storage**  
**для зберігання даних Big Data”**

Виконав здобувач вищої освіти  
IV курсу, групи КН-22  
ОПП «Комп’ютерні науки»  
спеціальності 122 «Комп’ютерні науки»  
\_\_\_\_\_ Чабан О.О.  
« \_\_\_\_ » \_\_\_\_\_ 2026 р.

Керівник проекту  
кандидат технічних наук, доцент  
\_\_\_\_\_ Смірнова Т.В.  
« \_\_\_\_ » \_\_\_\_\_ 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

« 15 » січня 2026 року

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

*Чабану Олександру Олександровичу*

(прізвище, ім’я, по батькові)

1. Тема роботи *Програмне забезпечення системи флеш-масивів Pure Storage для зберігання даних Big Data*

2. Керівник роботи *Смірнова Тетяна Віталіївна, канд. техн. наук, доцент*

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 116-02 від 06.02.2026 року

3. Строк подання студентом роботи до захисту *23.05.2026 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи флеш-масивів Pure Storage для зберігання даних Big Data*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

*1. Призначення та область використання.*

*2. Перегляд аналогічних існуючих систем.*

*3. Опис і обґрунтування проектних рішень.*

*4. Етапи програмування системи.*

*5. Впровадження системи в промислову експлуатацію.*

*6. Висновки*

6. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

*Структурна схема системи* *1 аркуш*

*Функціональна схема системи* *1 аркуш*

*Діаграма процесів* *1 аркуш*

*Блок-схема алгоритму роботи додатку* *2 аркуша*

7. Дата видачі завдання « 15 » січня 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  
« 15 » січня 2026 р.

Підпис керівника

Смірнова Т.В.  
(прізвище та ініціали)

Завдання прийнято до виконання  
« 15 » січня 2026 р.

Підпис здобувача

Чабан О.О.  
(прізвище та ініціали)

## **АНОТАЦІЯ**

**Чабан О.О. Програмне забезпечення системи флеш-масивів Pure Storage для зберігання даних Big Data. 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 флеш-масивів Pure Storage для зберігання даних Big Data.

Метою розробки є програмне забезпечення системи флеш-масивів Pure Storage для зберігання даних Big Data.

Результат роботи – програмна реалізація системи флеш-масивів Pure Storage для зберігання даних Big Data.

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, Pure Storage, Big Data

## ABSTRACT

**Chaban O.O. Software for the Pure Storage flash array system for Big Data storage. 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 the Pure Storage flash array system for Big Data storage.

The purpose of the development is software for the Pure Storage flash array system for Big Data storage.

The result of the work is a software implementation of the Pure Storage flash array system for Big Data storage.

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 OS Windows 10/11.

The program was developed in the Python environment.

**Keywords:** computer science, Pure Storage, Big Data

## ЗМІСТ

|                                                                                                                                                                             |    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....                                                                                                               | 2  |
| ВСТУП.....                                                                                                                                                                  | 3  |
| 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....                                                                                                                                 | 4  |
| 1.1 Призначення системи.....                                                                                                                                                | 4  |
| 1.2 Область застосування.....                                                                                                                                               | 5  |
| 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....                                                                                                                                | 6  |
| 2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти..... | 6  |
| 2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....                                                                                            | 15 |
| 2.3 Розгорнута постановка завдання .....                                                                                                                                    | 20 |
| 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....                                                                                                                               | 21 |
| 3.1 Опис функціонування системи .....                                                                                                                                       | 21 |
| 3.2 Розробка структурної схеми.....                                                                                                                                         | 33 |
| 3.3 Розробка функціональної схеми .....                                                                                                                                     | 36 |
| 3.4 Розробка діаграми процесів.....                                                                                                                                         | 50 |
| 4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....                                                      | 52 |
| 4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....                                                                                                       | 52 |
| 4.2 Захист розробленого програмного забезпечення.....                                                                                                                       | 70 |
| 5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....                                                                                                                      | 72 |
| 6 ОСНОВНІ ВИСНОВКИ.....                                                                                                                                                     | 79 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....                                                                                                                                            | 81 |

|                 |                |                 |              |             |                                                                                              |                   |              |                |
|-----------------|----------------|-----------------|--------------|-------------|----------------------------------------------------------------------------------------------|-------------------|--------------|----------------|
|                 |                |                 |              |             | <b>ВКРБ-122.26.0033.00.00.ПЗ</b>                                                             |                   |              |                |
| <b>Вим.</b>     | <b>Арк.</b>    | <b>№ докум.</b> | <b>Підп.</b> | <b>Дата</b> | Програмне забезпечення системи<br>флеш-масивів Pure Storage для<br>зберігання даних Big Data | <b>Лім.</b>       | <b>Аркуш</b> | <b>Аркушів</b> |
| <b>Розроб.</b>  | Чабан О.О.     |                 |              |             |                                                                                              | <b>Б</b>          | 1            | 87             |
| <b>Перев.</b>   | Смірнова Т.В.  |                 |              |             |                                                                                              |                   |              |                |
| <b>Н.контр.</b> | Коваленко А.С. |                 |              |             |                                                                                              | <b>ЦНТУ КН-22</b> |              |                |
| <b>Затв.</b>    | Смірнов О.А.   |                 |              |             |                                                                                              |                   |              |                |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

|      |   |                                                                  |
|------|---|------------------------------------------------------------------|
| БД   | - | база даних                                                       |
| БЗ   | - | база знань                                                       |
| БРС  | - | блок розрахункових співвідношень                                 |
| ВКПК | - | вибір кількості пріоритетних користувачів                        |
| ІМА  | - | інтелектуальний мережний адміністратор                           |
| КОМ  | - | корпоративні обчислювальні мережі                                |
| ЛОМ  | - | локальні обчислювальні мережі                                    |
| МА   | - | мережне адміністрування                                          |
| ПЗ   | - | програмне забезпечення                                           |
| СЗД  | - | системи зберігання даних                                         |
| СУБД | - | система управління базою даних                                   |
| ЦОД  | - | центр обробки даних                                              |
| СМІР | - | протокол загальної керуючої інформації                           |
| DHCP | - | протокол динамічної конфігурації вузла                           |
| FMP  | - | HPOpenView Fault Management Platform                             |
| MIB  | - | база даних інформації керування                                  |
| OEMF | - | HPOpenView Element Management Framework                          |
| OER  | - | Optimized Edge Routing                                           |
| OV   | - | HPOpenView                                                       |
| Pf   | - | Performance Routing                                              |
| RMON | - | протокол моніторингу комп'ютерних мереж                          |
| SNMP | - | простий протокол керування мережею                               |
| WINS | - | служба зіставлення netbios-імен комп'ютерів з ір-адресами вузлів |

## ВСТУП

**Актуальність теми.** В останні роки спостерігається величезне зростання попиту на технічні досягнення. Потреба в комерційному сховищі даних зростає разом із щоденним розвитком бізнесу багатьох технологічних організацій, і масиви флеш-пам'яті зайняли центральне місце для багатьох підприємств. Крім того, обробка даних у режимі реального часу стає все більш необхідною для надання ефективних послуг у будь-який час, у будь-якому місці та в найкоротші терміни завдяки зростаючому включенню хмарних обчислень, Інтернету речей, машинного навчання, штучного інтелекту, мобільних пристроїв та когнітивних систем. Тому очікується, що найближчим часом цей зростаючий попит на обробку даних у режимі реального часу створить прибуткові бізнес-можливості, формуючи загальний прогноз ринку масивів на основі флеш-пам'яті та впливаючи на ширший ринок хмарних сховищ.

**Мета й завдання дослідження.** Метою роботи є програмне забезпечення системи флеш-масивів Pure Storage для зберігання даних Big Data.

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

- Огляд існуючих систем флеш-масивів Pure Storage для зберігання даних Big Data.
- Дослідження системи флеш-масивів Pure Storage для зберігання даних Big Data.
- Програмна реалізація системи флеш-масивів Pure Storage для зберігання даних Big Data.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі флеш-масивів Pure Storage для зберігання даних Big Data.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи флеш-масивів Pure Storage для зберігання даних Big Data, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 3    |

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Крім технічних інновацій рішення Pure Storage відрізняє революційна модель їхньої підтримки, що одержала назву Evergreen. Як затверджується, придбавши один раз масив, Pure Storage клієнт зможе його використовувати й через 20 років, при цьому він буде адекватну продуктивність і ємність. Замовникам більше немає необхідності регулярно мігрувати дані зі старих систем на нові. Наші технології будуються навколо даних, а не навколо контейнерів, на яких вони зберігаються. Такий підхід дозволяє скоротити вартість володіння на 61%.

Сервіс Evergreen передбачає два види підписки: Evergreen Silver і Evergreen Gold. «Золотий» пакет припускає можливість відновлення не тільки програмного, але й апаратного забезпечення – заміну контролерів з появою рішень наступного покоління (у середньому раз у три роки). Вартість трирічної підтримки срібного рівня становить менш 20% від вартості системи, а золотого не перевищує 40%. В Україні клієнти поки воліють більше дешеві сервісні контракти (на них доводиться більше 80%), оскільки розрахунок вартості володіння обмежується переважно 3-5 років.

Підписка Evergreen Gold виявляється вигідніше при розрахунку TCO за 10-літній період. У більшості вендорів підтримок обмежується 5 роками, оскільки за цей час масиви застарівають. Відповідно, у розрахунок вартості володіння за 10 років необхідно включати покупку нової СЗД. У випадку масивів Pure Storage їхній інтелект (контролери) буде оновлений три рази, у результаті наприкінці цього строку замовник буде мати набагато більше продуктивний масив, ніж споконвічно, а сумарно витратить менше.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 4    |

## 1.2 Область застосування

Очікується, що найближчим часом виникнуть значні можливості для зростання завдяки зростаючому попиту на креативні флеш-рішення для неструктурованих даних та великих даних, що підтримує ширші тенденції ринку флеш-масивів. Попит на нові системи зберігання даних для зберігання даних протягом тривалих періодів часу зріс, оскільки великі та неструктуровані дані стають все більш поширеними. Такі досягнення, як Non-Volatile Memory Express (NVMe), стимулювали ринок корпоративних повністю флеш-сховищ, забезпечуючи швидшу продуктивність та більшу щільність, ніж традиційні протоколи. Крім того, корпоративні системи зберігання даних починають впроваджувати нові технології пам'яті класу зберігання (SCM). SCM, ймовірно, будуть використовуватися як кешувальний шар повністю флеш-масивів. Ці нові розробки мають покращити продуктивність робочого навантаження, одночасно знижуючи витрати на зберігання даних, сприяючи аналізу галузі флеш-масивів, що розвивається, та розширюючи актуальність ринку інструментів для масивів.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи флеш-масивів Pure Storage для зберігання даних Big Data, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 5    |

## 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

До 2025 року понад 80% великих українських компаній перенесли основні аналітичні завдання на хмарні платформи, що спонукало коледжі оновлювати свої технологічні програми, щоб вони відповідали цим інструментам. Початкові посади в аналітиці та інженерії даних мали середню початкову заробітну плату понад 70 тисяч доларів у 2024 році, що мотивує учнів розвивати навички роботи з хмарними інструментами якомога раніше. Інструменти для роботи з великими даними продовжують швидко розвиватися, і багато хмарних платформ щороку додають нові функції.

У наведеному нижче списку наведено 20 найпопулярніших інструментів та технологій для роботи з великими даними, які варто опанувати у 2026 році, тобто тих, які впливають на те, як великі робочі навантаження переміщують, зберігають та обробляють файли в різних умовах. Кожен інструмент у цьому списку пов'язаний з навичками, яким навчають у коледжах прямо зараз, що допомагає учням створювати проекти, що відповідають тому, чого українські компанії хочуть від нових талантів.

#### 1. Apache Spark

Опитування серед українських технічних команд показують, що навички роботи зі Spark, SQL та хмарними сховищами даних відображаються в більшості списків стажувань для напрямків аналітики та інженерії. Завантажте публічний набір даних та виконайте базові перетворення, такі як фільтри, об'єднання та прості агрегації. Spark надає студентам швидкий спосіб роботи з великими робочими навантаженнями. Він виконує завдання в пам'яті, що скорочує час

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 6    |

очікування під час лабораторних робіт або невеликих проєктів. Багато українських компаній використовують Spark для пакетних завдань, потокових завдань та конвеєрів машинного навчання.

Ключові особливості інструменту:

- Працює з Python, що спрощує роботу для початківців.
- Виконує великі завдання з меншою кількістю рядків коду.
- Безкоштовне встановлення на ноутбук.
- Сильна спільнота та багато онлайн-довідників проєктів.

## 2. Apache Hadoop

Hadoop започаткував сучасну хвилю великих даних у технологічній галузі України понад десять років тому та залишається активним у корпоративних середовищах. Він допомагає розбивати великі набори даних на менші частини на багатьох машинах. Він запропонував ідею зберігання та обробки великих робочих навантажень у великих масштабах. Студенти можуть не запускати повні кластери вдома, але вивчення основних ідей додає впевненості для проходження практики.

Ключові особливості інструменту:

- Багато довгострокових корпоративних систем досі працюють на ньому.
- Викладає основні ідеї, такі як розподілене сховище та MapReduce.
- Допомагає студентам зрозуміти, як великі системи справляються з робочими навантаженнями.

## 3. Snowflake

Snowflake працює в хмарі та виконує складні аналітичні завдання без налаштування обладнання. Компаніям подобається він, оскільки швидко масштабується в години пік. Багато українських компаній використовують його для щоденної звітності, інформаційних панелей та бізнес-аналітики.

Ключові особливості інструменту:

- Працює повністю в хмарі, тому встановлення не потребує проблем.
- Використовує простий SQL, що зменшує криву навчання.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 7    |

- Справляється з великими робочими навантаженнями без ручного налаштування.
- Значне впровадження у фінансових, медичних, роздрібних та технологічних компаніях України.

#### 4. Databricks

Databricks надає єдине місце для роботи з блокнотами, великими робочими навантаженнями, завданнями SQL та машинним навчанням. Він працює в хмарі та легко пов'язаний зі Spark. Багато українських компаній обирають його, оскільки команди можуть писати код, виконувати завдання та ділитися результатами в одному робочому просторі. Це вселяє впевненість під час співбесід на стажування, де роботодавці очікують базових навичок роботи з хмарою та Spark.

Ключові особливості інструменту:

- Побудовано безпосередньо на Spark, що допомагає вам швидше розвиватися.
- Легко почати роботу з блокнотами, що підтримують SQL та Python.
- Широко використовується в українських компаніях для аналітики та роботи з конвеєрами.
- Потужний набір безкоштовних ресурсів спільноти для практики.

#### 5. Delta Lake

Університети України розширили кількість курсів з хмарних технологій та аналітики між 2022 і 2025 роками, оскільки компанії почали вимагати навички роботи з великими даними, орієнтовані на хмарні технології, а не застарілих інструментів. Delta Lake забезпечує стабільний спосіб зберігання великих файлів у хмарі, зберігаючи дані чистими та надійними. Це вирішує поширену проблему, коли файли змінюються або оновлюються в невідповідний час під час проєктів. Завдяки цьому можна відстежувати оновлення, скасовувати зміни та підтримувати порядок у наборах даних для занять або роботи з портфоліо.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 8    |

Ключові особливості інструменту:

- Зберігає чистоту наборів даних завдяки вбудованому відстеженню версій.
- Працює безперебійно з Spark та Databricks.
- Допомогає студентам створювати хмарні проекти, які наближені до реальних виробничих налаштувань.
- Підтримує великі робочі навантаження без складного запуску.

## 6. Apache Iceberg

Apache Iceberg надає студентам сучасний спосіб керування великими таблицями у хмарному сховищі. Він вирішує проблеми, з якими стикалися старі формати таблиць, такі як повільні оновлення та незручні розділи. Він підтримує чітку структуру таблиць, що сприяє безперебійній роботі як невеликих студентських лабораторій, так і величезних корпоративних конвеєрів. Багато українських компаній перейшли на Iceberg, оскільки він підтримує стабільне одночасне читання та запис без уповільнення.

Ключові особливості інструменту:

- Чисто обробляє оновлення, видалення та вставки.
- Працює з такими двигунами, як Spark, Fink та Trino.
- Допомогає створювати таблиці, що масштабуються від невеликих проектів до великих хмарних завдань.
- Значне зростання хмарних команд в Україні протягом 2024-2025 років.

## 7. Kafka

Більше половини українських додатків з високим трафіком використовують системи потокової передачі подій, такі як Kafka, для обробки живих дій та журналів. Kafka допомагає зрозуміти, як додатки фактично передають події. Українські компанії покладаються на нього для високошвидкісної передачі даних про кліки, дій додатків, оновлень датчиків та журналів. Коли людина вивчає Kafka, вона отримує чітке уявлення про те, як

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 9    |

сучасні системи обробляють безперервні вхідні події. Це забезпечує безпеку повідомлень, доки споживачі їх не прочитають.

Ключові особливості інструменту:

- Обробляє великі потоки подій зі стабільною продуктивністю.
- Працює з Spark, Flink та іншими двигунами.
- Допомогає учням зрозуміти моделі поведінки виробника та споживача.
- Активне використання у фінансових, роздрібних, медіа та технологічних компаніях України.

## 8. Google BigQuery

BigQuery – це повністю кероване сховище даних Google Cloud, яке легко обробляє набори даних петабайтного масштабу. Воно широко використовується в навчальних лабораторіях для навчання хмарній аналітиці, SQL, запитам та масштабованій обробці даних без необхідності турбуватися про інфраструктуру.

Ключові особливості інструменту:

- Безсерверна архітектура, не потрібна установка обладнання.
- Підтримує стандартний SQL та інтегрується з Python та R.
- Дозволяє аналізувати величезні набори даних за лічені секунди.
- Надає безкоштовний рівень для експериментів.

## 9. Airflow

Це інструмент оркестрації робочих процесів, який планує та керує конвеєрами даних. Він допомагає учням зрозуміти, як складні завдання з великими даними автоматизуються та контролюються в реальних середовищах.

Ключові особливості інструменту:

- Визначає робочі процеси як код за допомогою Python.
- Підтримує керування залежностями та повторні спроби.
- Інтегрується з хмарними платформами та основними інструментами обробки даних.
- Відкритий код з потужними ресурсами спільноти.

## 10. MongoDB

MongoDB – це NoSQL-база даних, розроблена для гнучкого зберігання документів. Вона ідеально підходить для дослідження сучасних архітектур великих даних, обробки напівструктурованих або неструктурованих наборів даних, поширених у веб– та мобільних додатках.

Ключові особливості інструменту:

- Зберігає документи у форматі JSON, що зручно для даних без схеми.
- Масштабується горизонтально на кількох серверах.
- Підтримує конвеєри агрегації та аналітику.
- Безкоштовна спільнотна версія доступна для практичних проектів.

## 11. Druid

Druid – це високопродуктивна аналітична база даних, оптимізована для запитів до великих наборів даних. Вона широко використовується для інтерактивних інформаційних панелей та аналітики часових рядів у сучасних додатках, що робить її дуже актуальною для освіти в галузі обробки даних. Її архітектура робить її ідеальною для дослідження запитів з низькою затримкою в сучасній освіті з великих даних.

Ключові особливості інструменту:

- Підтримує запити за менш ніж секунду для масивних наборів даних.
- Обробляє як потокові, так і пакетні дані.
- Інтегрується з інструментами візуалізації, такими як Superset та Tableau.
- Відкритий код зі зростаючим впровадженням у українських компаніях.

## 12. NiFi

NiFi – це інструмент інтеграції даних, розроблений для автоматизації потоку даних між системами. Він знайомить учнів зі створенням конвеєрів для обробки, перетворення та маршрутизації великих і різноманітних наборів даних. Він допомагає зрозуміти складні робочі процеси без тривалого кодування.

Ключові особливості інструменту:

- Інтерфейс перетягування спрощує створення робочого процесу.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 11   |

- Обробляє пакетну роботу та прийом даних.
- Підтримує безпечне та регульоване переміщення даних.
- Відкритий код з активною документацією та навчальними посібниками.

### 13. Hive

Hive надає інтерфейс, подібний до SQL, для запитів до великих наборів даних, що зберігаються в розподілених системах зберігання, таких як Hadoop та хмарні озера даних. Він поєднує традиційні навички роботи з реляційними базами даних з великими даними, що робить його чудовою платформою для вивчення масштабованої пакетної аналітики. Hive дозволяє експериментувати з масивними наборами даних, зберігаючи при цьому знайомі з SQL знання.

Ключові особливості інструменту:

- Використовує знайомий синтаксис SQL для великих даних.
- Бездоганно працює з Hadoop та Spark.
- Підтримує великомасштабну пакетну обробку.
- Сильна спільнота відкритого коду з освітніми ресурсами.

### 14. Розширений TensorFlow (TFX)

TFX – це платформа машинного навчання для створення готових до використання конвеєрів машинного навчання на великих наборах даних. Вона поєднує великі дані та штучний інтелект, надаючи досвід у робочих процесах машинного навчання, інтегрованих з інструментами для роботи з великими даними. Вона особливо корисна в хмарних середовищах, де набори даних є масивними та постійно оновлюються.

Ключові особливості інструменту:

- Автоматизує етапи робочого процесу машинного навчання: перевірку даних, трансформацію, навчання, розгортання.
- Працює з моделями TensorFlow та конвеєрами Spark.
- Масштаби для пакетних та потокових даних.
- Відкритий код з активними навчальними посібниками для навчання.

## 15. Trino

Trino (раніше PrestoSQL) – це розподілений механізм SQL-запитів для аналітики великих наборів даних у кількох системах зберігання. Він дозволяє експериментувати із запитом до кількох джерел даних без переміщення даних, що робить його ідеальним вибором для середовищ Lakehouse.

Ключові особливості інструменту:

- Виконує інтерактивні запити до величезних наборів даних.
- Працює з Hadoop, S3, реляційними базами даних та іншими.
- Підтримує ANSI SQL, що робить його знайомим для учнів.
- Відкритий код зі зростаючим впровадженням у підприємствах України.

## 16. Presto

Presto допомагає обробляти великі запити в багатьох системах зберігання даних без переміщення файлів. Він видає швидкі результати навіть у гігантських таблицях, що робить його корисним для хмарних навчальних лабораторій. Він виконує інтерактивні SQL-завдання, яких достатньо для практики на змішаних джерелах даних.

Ключові особливості інструменту:

- Виконує SQL у багатьох джерелах, таких як S3, Hive та MySQL.
- Добре працює для швидких, інтерактивних запитів.
- Підтримується великою, відкритою спільнотою.

## 17. ClickHouse

ClickHouse – це стовпчаста база даних, створена для дуже швидкої аналітики. Вона добре працює з інформаційними панелями, важкими звітами та робочими навантаженнями часових рядів. Її дизайн дозволяє великим запитам виконуватися швидко, навіть коли набір даних зростає.

Ключові особливості інструменту:

- Використовує сховище на основі стовпців для пришвидшення запитів.
- Обробляє робочі навантаження часових рядів та подій.
- Безкоштовний для роботи в хмарі або на локальних машинах.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 13   |

## 18. Flink

Flink обробляє потоки даних у реальному часі та стабільно обробляє події. Він навчає, як реальні додатки реагують на вхідні дії без затримки. Багато компаній покладаються на нього для поточних інформаційних панелей часу, перевірок на шахрайство та систем сповіщень.

Ключові особливості інструменту:

- Сильний у потоковому та пакетному обробці навантажень.
- Працює з Kafka та багатьма інструментами для конвеєрів.
- Виконує завдання з відстеженням стану без уповільнення.

## 19. Інструмент побудови даних (dbt)

Інструмент побудови даних (dbt) зосереджений на створенні чистих та надійних SQL-перетворень. Він навчає сильним звичкам, таким як контроль версій, тестування та модульне моделювання. Коледжі використовують його в хмарних курсах, щоб допомогти учням підготуватися до завдань робочого процесу. Багато команд в Україні покладаються на dbt, оскільки він чудово вписується в сучасні структури складів без складного кодування чи тривалого часу налаштування.

Ключові особливості інструменту:

- Перетворення на основі SQL з чіткою структурою.
- Вбудоване тестування для чистіших наборів даних.
- Працює з BigQuery, Snowflake, Redshift та іншими.

## 20. Hudi

Hudi допомагає керувати великими таблицями у хмарному сховищі за допомогою стабільних оновлень та видалень. Він підтримує як потокове, так і пакетне виконання завдань, що робить хмарні робочі процеси плавнішими. Він забезпечує надійний контроль на рівні записів без необхідності складного налаштування. Його дизайн допомагає упорядковувати старі та нові версії записів, що готує учнів до хмарних конвеєрів, які постійно оновлюються. Великі

технологічні команди використовують Hudi, оскільки він працює в Spark, Flink та Presto, зберігаючи порядок у таблицях навіть за інтенсивного використання.

Ключові особливості інструменту:

- Обробляє оновлення та видалення у великих масштабах.
- Працює з Spark, Flink та Presto.
- Зберігає порядок у великих таблицях завдяки вбудованому індексуванню.

### **Висновок**

На сайтах з пошуку роботи у 2025 році було розміщено понад 250 000 вакансій, пов'язаних з навичками роботи з хмарними технологіями та даними, що свідчить про явне зростання попиту на тих, хто знає ці інструменти. Ці двадцять інструментів пропонують надійний підхід для тих, хто планує працювати з великими наборами даних, хмарними системами або потоками живих подій. Кожен з них навчає різному аспекту світу великих даних, від швидких завдань SQL до потокових конвеєрів та великомасштабних систем зберігання даних.

## **2.2 Обґрунтування вибору засобів для побудови системи та мови програмування**

Python – це об'єктно-орієнтована мова програмування високого рівня загального призначення з відкритим кодом. Це визначення може бути важким для новачків, тому розглянемо кожну характеристику окремо, щоб зрозуміти, що вона означає:

- Відкритий вихідний код: це безкоштовно та доступно для подальших покращень, таких як додавання корисних функцій або виправлення помилок.
- Об'єктно-орієнтована: заснована не на функціях, але в об'єктах з певними атрибутами й методами.
- Високий рівень: зручний для людини, а не для комп'ютера.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 15   |

– Загальне призначення: можна використовувати для створення будь-яких програм.

Ця мова використовується в будь-якому програмному забезпеченні, про яке ви тільки можете подумати. Ви можете використовувати його для створення веб-сайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу та багато іншого. Також застосовується в науці про дані, аналізі даних, машинному навчанні, інженерії даних, веб-розробці, розробці програмного забезпечення та інших галузях.

### **Переваги та недоліки Python**

Переваги:

– Її легко читати, вчити та писати. Це мова програмування високого рівня з англійським синтаксисом. Це полегшує читання та розуміння коду. Її дійсно легко зрозуміти і вивчити, тому багато людей рекомендують Python новачкам. Вам потрібно менше рядків коду для виконання того ж завдання в порівнянні з іншими основними мовами, такими як C/C++ та Java.

– Підвищує продуктивність. Це дуже продуктивна мова. Завдяки її простоті розробники можуть зосередитися на розв'язанні проблеми. Їм не потрібно витрачати багато часу на розуміння синтаксису або поведінку мови програмування. Ви пишете менше коду та виконуєте більше завдань.

– Інтерпретована мова. Python мова, що інтерпретується, а це означає, що вона безпосередньо виконує код по рядку. Якщо сталася помилка, вона зупиняє подальше виконання та повідомляє про її виникнення. Вона показує лише одну помилку, навіть якщо у програмі їх кілька. Це спрощує налагодження.

– Динамічно типізована. Python не визначає тип змінної, доки ми не запустимо код. Вона автоматично надає тип даних, коли відбувається процес виконання. Фахівець може не турбуватися про оголошення змінних та типи даних.

– Безкоштовна та з відкритим вихідним кодом. Ця мова постачається під схваленою OSI ліцензією з відкритим вихідним кодом. Це робить його

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 16   |

безкоштовним для використання та розповсюдження. Ви можете завантажити вихідний код, змінити його та навіть розповсюджувати свою версію. Це корисно для організацій, які хочуть використати свою версію для розробки.

– Підтримка великих бібліотек. Стандартна бібліотека Python є величезною, ви можете знайти майже всі функції, необхідні для вашого завдання. Таким чином ви не залежите від зовнішніх бібліотек.

– Портативність. У багатьох мовах, таких як C/C++, потрібно змінити свій код, щоб запустити програму на різних платформах. З Python все інакше. Ви тільки пишете один раз і запускаєте її будь-де.

Недоліки:

– Низька швидкість. Вище ми обговорювали, що це інтерпретована мова з динамічною типізацією. Порядкове виконання коду часто призводить до повільного виконання. Динамічна природа Python також є причиною її низької швидкості, оскільки їй доводиться виконувати додаткову роботу при виконанні коду. Тому вона не підходить для цілей, де швидкість важливий аспект проєкту.

– Неефективна для пам'яті. Ця мова програмування використовує великий обсяг пам'яті, це може бути недоліком при створенні програм, коли віддають перевагу оптимізації пам'яті.

– Слабка у мобільних обчисленнях. Python зазвичай використовується у серверному програмуванні. Ми не бачимо – її на стороні клієнта або в мобільних програмах з таких причин: вона не заощаджує пам'ять і має повільну обчислювальну потужність у порівнянні з іншими мовами.

– Доступ до бази даних. Програмувати на цій мові легко, але коли ми взаємодіємо з базою даних, її не вистачає. Рівень доступу до бази даних у Python примітивний та недостатньо розвинений у порівнянні з іншими популярними технологіями.

– Помилки виконання. Це мова з динамічною типізацією, тому тип даних змінної може змінюватись у будь-який час. Змінна, що містить ціле число, у майбутньому може містити рядок, що може призвести до помилок виконання.

## Застосування Python:

– Для аналізу даних. Дані стали цінним активом у будь-якій сучасній галузі, і більшість компаній зацікавлені у збиранні, обробці та аналізі релевантних даних, щоб витягти з них цінну інформацію для бізнесу. І тут Python виходить за межі будь-якої конкуренції. Python особливо цінна тим, що крім великої стандартної бібліотеки надає величезний набір додаткових модулів, розроблених спеціально для аналітичних цілей. Найвідоміші бібліотеки Python для аналізу даних – це pandas і NumPy . Ці інструменти дозволяють робити з вашими даними майже все, наприклад, очищати і аналізувати їх, вивчати статистику або візуалізувати приховані тенденції у ваших даних.

– Для візуалізації даних. Візуалізація даних – це окрема частина аналізу даних, яка допомагає нам подавати інформацію, необроблену чи очищену, у більш змістовній формі. Тут Python знову входить у гру, пропонуючи широкий спектр інструментів візуалізації даних. Найпопулярніші з них – matplotlib і заснований на ній seaborn. Використовуючи їх, ми можемо створювати буквально всі види візуалізації: від найпростіших до складніших.

– Для машинного навчання. Машинне навчання (ML) є основою більшості завдань науки даних. Він є областю штучного інтелекту, пов'язаною з використанням алгоритмів, що дозволяють машинам вивчати закономірності та тенденції на основі історичних даних, щоб робити прогнози на основі невідомих даних. – Використовуючи методи ML, ми можемо створювати моделі, які можуть точно передбачити швидкість відтоку клієнтів компанії, оцінити ризик виникнення у людини певного захворювання, визначити оптимальне розташування автомобілів таксі й т.д. За допомогою Python ми можемо побудувати модель ML, використовуючи лише три рядки коду.

– Для розробки програмного забезпечення. Крім свого багатостороннього застосування в галузях науки про дані, Python використовується на кожному етапі розробки програмного забезпечення, включаючи контроль складання, автоматичну безперервну компіляцію, прототипування, відстеження помилок,

тестування та обслуговування програмного забезпечення. За допомогою цієї мови можемо створювати аудіо– або відеопрограми на основі методів штучного інтелекту, машинного навчання, API (інтерфейсів прикладного програмування), GUI (графічних інтерфейсів) або будь-якого іншого типу програмного забезпечення.

– Для веброзробки. У той час як для створення візуальної частини вебсайту ми переважно будемо використовувати такі мови, як HTML, CSS та JavaScript, для його невидимої частини ми часто вибираємо Python. Серед масштабних вебсайтів та програм, створених за допомогою цієї мови, варто згадати Google, Facebook, Instagram, YouTube, Dropbox та Reddit.

– Для автоматизації задач/скриптингу. Це відмінний інструмент для написання програм для автоматизації різних завдань, що повторюються. Цей процес називається скриптингом. Зокрема, можна робити скрипти для роботи з файлами та папками. Наприклад, можна створювати, перейменовувати, перетворювати, розділяти, об'єднувати або видаляти файли, перевіряти їх на наявність помилок. Ви також можете використовувати автоматизацію Python для пошуку та завантаження інформації з Інтернету, заповнення та надсилання онлайн-форм та надсилання регулярних повідомлень або електронних листів.

Яким фахівцям потрібно володіти Python:

- Фахівець з даних.
- Аналітик даних.
- Інженер даних.
- Інженер з машинного навчання.
- Журналіст даних.
- Архітектор даних.
- Повний стек веб-розробника.
- Backend-розробник.
- DevOps-інженер.
- Інженер-програміст.

### 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи флеш-масивів Pure Storage для зберігання даних Big Data.

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-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 20   |

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

**Попит на флеш-масиви зростає, оскільки вони перевершують традиційні методи зберігання даних.**

Завдяки своїм перевагам над жорсткими дисками, флеш-масиви стають дедалі популярнішими. Флеш-масиви набирають популярності, оскільки вони використовують менше серверів, менше енергії та менше часу на адміністрування сховища. Завдяки своєму крихітному розміру та зменшеним потребам у дисковому просторі та стійках, флеш-SSD спрощують керування пристроями. Ще однією перевагою пристрою є те, що він пропонує чудову швидкість передачі даних порівняно з жорсткими дисками. Крім того, зростаюче впровадження хмарних обчислень, мобільних пристроїв, Інтернету речей, когнітивних систем, штучного інтелекту та машинного навчання є одними з ключових рушійних сил, які, як очікується, стимулюватимуть зростання ринку флеш-масивів, впливаючи на загальне зростання ринку флеш-масивів.

Ці масиви користуються великим попитом для обробки даних у реальному часі та віртуальних операцій. Масиви на основі флеш-пам'яті допомагають забезпечити підвищену продуктивність бізнесу, оскільки віртуальні машини, віртуальні робочі столи та робочі навантаження баз даних потребують високої швидкості. Усі флеш-масиви перевершують своїх конкурентів завдяки вигідному коефіцієнту охолодження, який розсіює менше тепла та сприяє економії енергії. Як результат, вони більш адаптовані в центрах обробки даних, де їм потрібно значно менше енергії для кондиціонування повітря, що зміцнює загальне розширення ринку флеш-масивів, а також потребує моніторингу, що підтримується реєстратором даних температури та USB-реєстратором даних температури в контрольованих середовищах.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 21   |

## **Останні тенденції в індустрії флеш-масивів**

- Розширене шифрування покращує безпеку та захист даних, зміцнюючи конкурентний ландшафт ринку флеш-масивів.
- Штучний інтелект оптимізує продуктивність і дозволяє проводити прогнозне обслуговування, відображаючи постійний аналіз ринку флеш-масивів для всіх інтелектуальних робочих навантажень.
- Флеш-масиви тепер забезпечують більшу ємність для зберігання даних, що відповідає мінливій сегментації ринку флеш-масивів у різних моделях розгортання підприємств.

## **Висока вартість та технічні обмеження флеш-масивів перешкоджають розширенню ринку**

Флеш-масиви є дорогими порівняно зі звичайними системами зберігання даних. Через серйозні бюджетні обмеження це не підходить для малого та середнього бізнесу. Малий та середній бізнес потім використовує традиційні системи зберігання даних, що, як очікується, перешкоджатиме світовому ринку флеш-масивів протягом прогнозованого періоду, впливаючи на ширший прогноз ринку флеш-масивів. Більше того, через їхню нездатність забезпечити низьку затримку, гібридні флеш-масиви несумісні. Крім того, всі флеш-накопичувачі мають обмежувальну характеристику, яка включає їхню енергонезалежну конструкцію, яка призводить до втрати даних у разі відключення живлення, а також відновлення даних, що є складнішим, ніж зі звичайними дисками. Тому необхідне належне резервне копіювання даних у разі відключення живлення та подальшої втрати даних. Очікується, що всі ці фактори стримуватимуть розвиток ринку флеш-масивів протягом прогнозованого періоду, що відображається в поточній динаміці ринку флеш-масивів.

## **Нові продукти набувають значної популярності на ринку флеш-масивів**

Щоб залучити більше клієнтів, основні гравці на ринку флеш-масивів в першу чергу зосереджуються на розробці доступних рішень для зберігання даних

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-122.26.0033.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 22   |

на базі флеш-пам'яті. Вони також розширюють свій портфель продуктів шляхом випуску нових продуктів, злиттів та поглинань, щоб зміцнити свої позиції на світовому ринку, формуючи конкурентні структури на ринку флеш-масивів. Наприклад, у 2021 році компанія Karley, провідний постачальник апаратних та програмних технологій та рішень для ринків високопродуктивних, орієнтованих на дані обчислень, випустила плату прискорення інтелектуального зберігання K200-LP™ на базі процесора MPPA® DPU від Kalray. Ця карта зберігання забезпечує високу продуктивність, низьке енергоспоживання, економічну ефективність та відкритість для масштабування та відповідає попиту на доступність даних.

Знову ж таки, у червні 2022 року, Dell Technologies представила найновіше операційне середовище (OE) 5.2 для Dell Unity XT, яке забезпечує швидше відновлення даних, збільшує використання сховища, розширює топології аварійного відновлення тощо. Раніше, у грудні 2021 року, Pure Storage представила FlashArray//XL, високоемний 5U варіант FlashArray, здатний забезпечити хмарну модель для розгортання нових програм та інших переваг. Крім того, у травні 2022 року NetApp оголосила, що паралельна файлова система BeeGFS та повністю флеш-сховище NVMe NetApp EF600 схвалені для використання з NVIDIA DGX SuperPOD. Завдяки новій сертифікації ці варіанти використання можуть бути реалізовані швидше завдяки спрощенню інфраструктури для штучного інтелекту (ШІ) та високопродуктивних обчислень, що сприятиме розвитку тенденцій ринку флеш-масивів та інноваційних циклів.

### **Бурхливий розвиток ІТ-інфраструктури та присутність провідних гравців сприяють зростанню ринку**

Прогнозується, що ринок флеш-масивів зростатиме в результаті цього, що відповідає довгостроковим прогнозам ринку флеш-масивів та сталому зростанню ринку флеш-масивів. Крім того, очікується, що зростання віртуальних операцій для низки компаній, які запровадили нові практики, включаючи віддалену роботу та віртуальні офіси, стимулюватиме розширення ринку флеш-масивів,

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 23   |

підкріплюючи ширші міркування щодо перспектив ринку флеш-масивів.

### **Аналіз ринку флеш-масивів**

Очікується, що ринок повністю флеш-масивів зросте з 23,38 млрд доларів у 2025 році до 27,73 млрд доларів у 2026 році та, за прогнозами, досягне 65,09 млрд доларів до 2031 року зі середньорічним темпом зростання 18,62% протягом 2026-2031 років. Імпульс надається підприємствами, які переносять критично важливі бази даних, кластери навчання штучного інтелекту та конвеєри аналітики в режимі реального часу на флеш-системи, що забезпечують надійну затримку менше мілісекунди. Оператори гіпермасштабування масштабують нові центри обробки даних на графічних процесорах рекордними темпами, національні регулятори посилюють правила енергоефективності, а постачальники NAND-пам'яті продовжують знижувати вартість біта, усуваючи останні бар'єри, які стримували гібридні масиви. Паралельний прогрес у NVMe-over-Fabrics (NVMe-oF) та механізмах скорочення даних в масивах тепер дозволяє організаціям консолідувати більші набори даних, зберігаючи при цьому продуктивність, що ще більше схиляє бюджети до розгортання флеш-пам'яті. Постачальники, які поєднують спеціалізоване програмне забезпечення з ефективним обладнанням, набирають популярності, оскільки кінцеві користувачі надають пріоритет передбачуваній якості обслуговування під час виведення даних за допомогою штучного інтелекту, аналітики периферійних даних та сценаріїв кібервідновлення.[1]

За архітектурою, масштабовані масиви лідирували з часткою ринку All Flash Array у 2025 році, тоді як дезагреговані системи NVMe-oF розвиватимуться зі середньорічним темпом зростання 19,45% до 2031 року.

За ємністю, діапазон 20-100 ТБ становив 49,25% розгортань ринку флеш-масивів у 2025 році; системи ємністю понад 100 ТБ зростатимуть зі середньорічним темпом зростання 19,61% до 2031 року.

За протоколом інтерфейсу, Fibre Channel займав 38,22% ринку флеш-масивів у 2025 році, проте рішення NVMe демонструють найшвидше зростання зі

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 24   |

середньорічним темпом зростання 20,05%.

За галузями кінцевих користувачів, ІТ та телекомунікації зберегли 34,35% частки ринку повністю флеш-масивів у 2025 році; прогнозується, що охорона здоров'я та науки про життя зростатимуть зі середньорічним темпом зростання 20,54% до 2031 року.

За географічним розташуванням, на Північну Америку припадало 38,24% доходу ринку флеш-масивів у 2025 році, тоді як Азіатсько-Тихоокеанський регіон зростає зі середньорічним темпом зростання 20,30% до 2031 року.

### **Зростання кількості гіпермасштабованих та периферійних центрів обробки даних**

Оператори гіпермасштабних хмарних сервісів разом виділили понад 130 мільярдів доларів у 2025 році на капітальні витрати на кампуси, оптимізовані для штучного інтелекту, а на зберігання даних зараз припадає приблизно п'ята частина цих витрат.[2]Нові об'єкти потребують щільних рівнів флеш-пам'яті з низькою затримкою для живлення тисяч графічних процесорів, які обробляють параметри великих мовних моделей. Паралельно, розгортання на периферії в роздрібних торгових точках, вежах стільникового зв'язку та виробничих майданчиках віддає перевагу компактним флеш-системам, які витримують вібрацію та коливання температури, але при цьому забезпечують стабільну пропускну здатність. Постачальники реагують на це, пропонуючи уніфікований код контролера, який масштабується від дворозлових периферійних пристроїв до стійкових NVMe-oF-матриць в основних центрах обробки даних. Обмеження потужності, керовані комунальними службами, у великих мегаполісах ще більше прискорюють перехід від гібридних накопичувачів до флеш-пам'яті, яка забезпечує вищі показники IOPS на ват. Як результат, ринок повністю флеш-масивів отримує вигоду від синхронізованого попиту в централізованих хмарах та розподілених периферійних вузлах.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 25   |

## **Критично важливі для продуктивності робочі навантаження штучного інтелекту та аналітики в режимі реального часу**

Навчання моделей штучного інтелекту перенасичує канали сховища послідовними записами, тоді як робочі навантаження логічного висновку залежать від мікросекундних випадкових зчитувань. Профілі подвійного робочого навантаження розкривають обмеження затримки та глибини черги гібридних масивів, що підштовхує банки, страховики та платформи електронної комерції до флеш-пам'яті. Фінансові установи, що впроваджують алгоритмічні торгові механізми, повідомляють про відчутне збільшення доходів після перенесення сховищ даних про тік на повністю флеш-платформи, перевірені за тестами STAC. Лікарні використовують флеш-архіви PACS, щоб діагностичні механізми штучного інтелекту могли без затримки переглядати МРТ та КТ-скани, прискорюючи час встановлення діагнозу. Виробники запускають моделі прогнозного обслуговування на периферійних графічних процесорах, які в режимі реального часу обробляють телеметрію датчиків, робочий процес, який порушується, якщо сховище зупиняється. Конвергенція генерації з доповненим пошуком, графічних запитів та потокової аналітики позиціонує ринок повністю флеш-масивів як фундаментальний для конвеєрів даних наступного покоління.

### **Швидке зниження вартості NAND за біт**

Фабрики NAND тепер переходять на 232-шарові QLC-дизайни, що знижує вартість на біт та підвищує довговічність завдяки вдосконаленим алгоритмам виправлення помилок.[3] Середні ціни продажу корпоративних SSD-накопичувачів впали на 12% у 2024 році, що дозволило постачальникам встановлювати ціни на конфігурації з використанням флеш-пам'яті в межах середнього гібридного бюджету. Нижча вартість носіїв дозволяє виробникам оригінального обладнання (ОЕМ) об'єднувати більші необроблені ємності з вбудованою дедуплікацією та стисненням, забезпечуючи ефективні ємності, що перевищують ємності систем з обертовими дисками за аналогічною вартістю. Цінова еластичність розширює адресуєму базу для ІТ-інженерії відділів,

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-122.26.0033.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 26   |

аналітики периферійних даних та вторинного копіювання даних, збільшуючи швидкість впровадження на ринку масивів з використанням флеш-пам'яті.

### **Вимоги щодо енергоефективності для корпоративних ІТ-систем**

Переглянута Директива ЄС щодо енергоефективності та розширені стандарти України піднімають споживання енергії до трійки головних критеріїв покупки. Сучасні флеш-масиви споживають до 80% менше енергії, ніж аналогічні гібридні стійки, зменшуючи як експлуатаційні витрати, так і викиди за класифікацією Score-3. Великі постачальники хмарних технологій тепер повідомляють про показники ESG, які включають ват-години зберігання на корисний терабайт, мотивуючи закупівельні команди замінювати рівні з великою кількістю дисків на пули флеш-пам'яті. Національні комунальні підприємства в Ірландії, Сінгапурі та Вірджинії запроваджують квоти на доступ до живлення, які винагороджують центри обробки даних з вищою продуктивністю на кіловат, параметр, за якого процвітає флеш-пам'ять. Ця політична тенденція підтримує стійку траєкторію попиту на ринок повністю флеш-масивів.

### **Початкова премія за капітальні витрати порівняно з гібридними масивами**

Навіть з урахуванням падіння цін на SSD, базовий вузол, що використовує лише флеш-накопичувачі, все ще коштує на 40-60% дорожче, ніж гібридний альтернативний варіант з такою ж корисною ємністю. Фінансові директори у вертикалях, чутливих до ціни, часто надають пріоритет початковій вартості придбання над п'ятирічною сукупною вартістю володіння, що затримує проекти міграції флеш-накопичувачів. Малий та середній бізнес у Латинській Америці та частинах Південно-Східної Азії продовжує термін служби дискових масивів, додаючи оновлення кешу, а не переміщуючи робочі навантаження повністю. Економічна невизначеність, що виникає внаслідок інфляційних циклів, затягнула терміни затвердження закупівель, що ще більше пригнічує короткострокові переходи на ринок повністю флеш-масивів.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-122.26.0033.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 27   |

## Вузькі місця контролерів у масштабованих архітектурах

Більшість існуючих двоконтролерних конструкцій перенасичують потоки процесора, перш ніж NVMe-накопичувачі досягають повної пропускної здатності. Зі зростанням кількості SSD на шасі поглиблюються черги команд і зростає дисперсія затримки, що знижує детерміновану продуктивність, необхідну конвеєрам штучного інтелекту. Підприємства, які значні кошти інвестували в мережі зберігання даних Fibre Channel, вагаються замінювати їх, але поступове оновлення контролерів не усуває вузьке місце. Цей розрив мотивує витратити кошти на дезагреговані вузли NVMe-oF, які можуть розподіляти обробку вводу/виводу по всій мережі Ethernet, але нестача кваліфікованих кадрів у сучасних мережах зберігання даних перешкоджає швидкій міграції для багатьох IT-команд.

### Проблеми дезагрегації, домінування масштабування

Масштабовані масиви забезпечили 47,12% доходу у 2025 році, що робить їх найбільшим внеском у ринок повністю флеш-масивів. Знайомий двоконтролерний дизайн інтегрується з існуючою оркестрацією SAN, що пояснює їхню стійкість. Однак та сама архітектура має труднощі з використанням дедалі щільніших NVMe-накопичувачів, що підштовхує покупців, критично важливих для продуктивності, до масштабованих або повністю дезагрованих NVMe-oF конструкцій. Постачальники зменшують ризики, додаючи обчислювальні полиці без урахування стану та шари кешування, але ці модифікації не можуть усунути фундаментальні обмеження контролерів.

На противагу цьому, прогнозується, що дезагреговані кластери NVMe-oF демонструватимуть середньорічний темп зростання (CAGR) на рівні 19,45% до 2031 року, що є найшвидшим показником на ринку повністю флеш-масивів. Дезагрегація усуває вузькі місця на одному шасі, переносячи логіку сховища на вузли, підключені до Ethernet, які масштабуються лінійно. Pure Storage та кілька програмно-визначених учасників демонструють петабайтні ферми графічних процесорів, побудовані на цій моделі, посиляючись на стабільні показники

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 28   |

пропускної здатності ліній під час контрольних точок моделі штучного інтелекту. Впровадження все ще залежить від того, чи опанують команди центрів обробки даних RDMA, контроль перевантажень та багатошляховість через високошвидкісний Ethernet, створюючи дефіцит навичок, який гальмує розгортання в найближчій перспективі.

### **Системи високої щільності прискорюють зростання**

Діапазон 20-100 ТБ містив 49,25% розгортань у 2025 році, що є ідеальним варіантом для консолідації баз даних та усунення розростання віртуальних машин. Цей діапазон поєднує ціну, потужність стійки та вікна перебудови, що робить його стандартним для циклів оновлення середніх підприємств. Флеш-пам'ять QLC у поєднанні з агресивним вбудованим стисненням забезпечують ефективні ємності, які перевершують попередні гібридні моделі, не змушуючи операторів використовувати екзотичні конструкції шасі.

Прогнозується, що системи обсягом понад 100 ТБ зростатимуть зі середньорічним темпом зростання (CAGR) на 19,61% до 2031 року, випереджаючи загальний ринок флеш-масивів. Гіперскейлери надають перевагу цим щільним полицям для живлення кластерів графічних процесорів, де локальність знижує витрати на трафік зі сходу на захід. Фінансові торговельні центри також тяжіють до високощільних збірок, консолідуючи історію тиків та тестуючи бібліотеки на окремих пулах з наднизькою затримкою. Модуль DirectFlash від Micron на 150 ТБ, розроблений спільно з Pure Storage, є прикладом прагнення постачальників до багатопетабайтних стійок, які обіцяють менше кабелів, менше ват та безкомпромісну пропускну здатність.

### **NVMe руйнує лідерство в Fibre Channel**

Fibre Channel отримав 38,22% доходу за 2025 рік, що підкреслює його закріплений статус у критично важливих середовищах, які цінують детерміновану затримку. Сумісність з існуючими адаптерами хост-шини та матрицями комутаторів робить FC консервативним шляхом оновлення для основних баз даних ERP та розвантаження мейнфреймів. Ця інерція захищає FC

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-122.26.0033.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 29   |

від негайного витіснення, навіть попри те, що дорожня карта протоколу відстає від нових функцій NVMe, таких як багатошляхові простори імен.

Прогнозується, що інтерфейси NVMe, що охоплюють PCIe-direct, NVMe-oF RoCE та NVMe-over-TCP, зростатимуть зі середньорічним темпом зростання (CAGR) на 20,05%. Нові інсталяції рідко починаються на SAS або SATA, і кілька банків зі списку Fortune 100 визначають інтерфейси NVMe-native для кожної покупки Tier 0. Інтеграція NetApp ONTAP з Google Distributed Cloud демонструє міжсайтову реплікацію через NVMe-oF, яка відповідає суворим правилам суверенітету даних. Напрямок руху вказує на Ethernet-фабрики, що переносять конвергентний обчислювальний та сховищний трафік, що переміщує капітал зі спеціалізованих SAN-комутаторів на більш економічні 100 GbE та 400 GbE фабрики типу "leaf-spine".

### **Охорона здоров'я виходить за межі лідерства в ІТ**

ІТ та телекомунікації зберегли 34,35% доходу за 2025 рік, що відображає постійний перехід у секторі віртуальних машин з локальних кластерів на стійки колокації, що надають перевагу флеш-пам'яті для змішаних робочих навантажень. Однак зростання сповільнюється зі зрілістю хмарної міграції, звільняючи місце для охорони здоров'я та наук про життя, які, як очікується, продемонструють середньорічний темп зростання 20,54%. Архів діагностичної візуалізації, конвеєри геноміки та аналітика записів пацієнтів вимагають високого обсягу вводу-виводу та дотримання політик незмінних знімків.

Лікарні, які впроваджують системи радіології на основі штучного інтелекту, виконують логічний висновок на основі бібліотек зображень петабайтного масштабу та не можуть прийняти затримку обертового диска. Фармацевтичні лабораторії, що проводять високопродуктивне секвенування генів, використовують флеш-масиви для обробки даних, які постачають дані алгоритмам виклику баз даних у режимі реального часу, скорочуючи цикли виявлення. Урядові та державні установи йдуть слідом, купуючи повністю флеш-вузли для податкової аналітики, робочих навантажень перепису та безпечних

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 30   |

хмарних середовищ, які забороняють фізичне знищення дисків. Серед кінцевих користувачів показники сталого розвитку визначають перевагу флеш-пам'яті, оскільки бюджети на енергоспоживання залишаються обмеженими, незважаючи на вищі обчислювальні вимоги.

Північна Америка генерувала 38,24% доходу у 2025 році, завдяки українським гіперскейлерам, що розширюють кластери штучного інтелекту, та торговим фірмам з Уолл-стріт, які модернізують озера ринкових даних. Регіон щороку виділяє понад 50 мільярдів доларів на інфраструктуру штучного інтелекту, а рівні повністю флеш-пам'яті лежать в основі кожного графічного модуля. Суворі правила SEC та FINRA також вимагають дотримання принципу «одноразова запис – багаторазове читання», якому сучасні флеш-масиви відповідають завдяки незмінним фреймворкам знімків.

Азіатсько-Тихоокеанський регіон є територією, що розвивається найшвидше, зі середньорічним темпом зростання 20,30% до 2031 року, оскільки заводи автоматизують контроль якості, а національні стратегії штучного інтелекту сприяють поширенню суверенних центрів обробки даних. Китай направляє субсидії місцевим брендам сховищ даних, щоб компенсувати експортний контроль, тоді як Японія та Південна Корея модернізують периферійні вузли 5G за допомогою полиць компактної флеш-пам'яті, які обробляють відеоаналітику на місці. Індійські центри ІТ-послуг розгортають стійки, що повністю базуються на флеш-пам'яті, всередині кампусів колокації наступного покоління, щоб скоротити час реагування додатків для глобальних клієнтів. Сінгапур та Гонконг об'єднують свої фінансові райони з регіональними кластерами флеш-пам'яті для аварійного відновлення, щоб дотримуватися правил швидшого клірингу та розрахунків. Європа балансує енергетичну політику з цифровими інноваціями, спонукаючи підприємства до сховищ, що зменшують ват-годинний слід. Німеччина фінансує пілотні лінії Industry 4.0, які вбудовують системи зору на основі штучного інтелекту, що потребують швидкого відновлення даних, тоді як Франція інвестує в суверенні хмарні регіони, оснащені

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 31   |

повністю флеш-магістральними рівнями. Північні країни залучають гіпермасштабних учасників до кампусів, що працюють на гідроенергетиці, забезпечуючи новий попит на флеш-лотки високої щільності. Тим часом, великі нафтові компанії Близького Сходу впроваджують флеш-масиви для платформ сейсмічного моделювання, а африканські телекомунікаційні компанії розгортають платформи мобільних грошей на основі флеш-пам'яті, які потребують цілодобової доступності, незважаючи на регіональну волатильність енергопостачання. У сукупності географічна диверсифікація підтримує стійкий шлях зростання ринку повністю флеш-масивів.

### **Конкурентний ландшафт**

Концентрація в галузі перебуває на помірному рівні, оскільки Pure Storage, Dell Technologies, NetApp, Hewlett Packard Enterprise, IBM та Huawei конкурують з більш гнучкими проривними компаніями. Pure Storage поєднує власні модулі DirectFlash з програмними модулями, які тісно інтегруються з платформами графічних процесорів NVIDIA, позиціонуючи себе як спеціаліста з інфраструктури штучного інтелекту. Dell оновлює свою лінійку PowerStore підтримкою NVMe-oF для захисту величезної встановленої бази, тоді як NetApp залучає покупців гібридної хмари через власні зв'язки з постачальниками гіпермасштабованих послуг.

Нові учасники, такі як VAST Data, Solidigm та Silk, роблять акцент на дезаггредованих просторах імен та програмно-визначених архітектурах. Їхні вузли, що масштабуються в міру зростання, обходять вузькі місця контролерів та вбудовують алгоритми багаторівневого розподілу, які автоматично розміщують холодні блоки на дешевших QLC. Швидкозростаючі сегменти, такі як кібервідновлення та периферійний штучний інтелект, створюють можливості для створення білого простору, де лояльність до бренду менш вкорінена, що дозволяє конкурентам конвертувати покупців, які вперше здійснюють покупку.

Стратегічні альянси формують диференціацію більше, ніж характеристики обладнання. Серія FlashSystem C від IBM інтегрує вбудовані механізми штучного

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 32   |

інтелекту для керування розміщенням даних. Hitachi Vantara розширює свій портфель VSP One наскрізною телеметрією, яка допомагає клієнтам звітувати про показники викидів вуглецю на рівні масиву.[4]Такі партнерства, як Nutanix та Pure Storage, поєднують гіперконвергентні обчислення із зовнішніми масивами NVMe-oF для задоволення потреб клієнтів, які шукають гнучкості ліцензування, у сфері віртуалізації. Поєднання тісної інтеграції з хмарними сервісами, показників енергоспоживання та оптимізації, орієнтованої на штучний інтелект, визначатиме зміщення часток на ринку повністю флеш-масивів протягом прогнозованого періоду.

### 3.2 Розробка структурної схеми

Практичні тактики зберігання даних можуть зробити відновлення швидшим та надійнішим у 2026 році. Флеш-масив Pure Storage забезпечує продуктивність NVMe та вбудоване скорочення обсягу даних, що робить часті знімки, миттєве відновлення, ефективну реплікацію та швидше тестування відновлення практичними та економічно ефективними для багатьох команд.

Тепер перейдемо до практичних дій, які ви можете вжити. Дев'ять тактик, наведених нижче, написані таким чином, що кожна починається з простого пояснення переваг, а потім йдуть короткі, практичні нотатки, які ви можете використовувати або обговорити зі своєю командою зі зберігання даних та постачальниками. Читайте їх по порядку або перейдіть до пунктів, які відповідають вашим найнагальнішим потребам у відновленні.

#### Чому зберігання даних важливе для RTO та RPO у 2026 році

Простіше кажучи, RTO (цільовий час відновлення) – це те, як швидко системи повинні повернутися до роботи після проблеми, а RPO (цільова точка відновлення) – це обсяг даних, який ви можете собі дозволити втратити. Обидва показники – це бізнес-показники, а не лише ІТ-цілі, оскільки повільне відновлення або втрата великих даних можуть перервати роботу клієнтів і

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 33   |

підірвати довіру. Рівень сховища є центром обох цілей, оскільки він створює та зберігає знімки та репліки, на яких базується відновлення. Коли сховище може швидко робити часті знімки та ефективно їх зберігати, у вас є новіші точки відновлення, і ви можете відновлюватися швидше. Це безпосередньо покращує RPO та RTO, не змушуючи вносити ризиковані зміни до програм.

Часто флеш-масив для зберігання даних допомагає поєднувати продуктивність NVMe з вбудованим скороченням обсягу даних. Це означає, що знімки споживають менше місця, а монтування або клони виконуються швидше, тому команди можуть зберігати більше точок відновлення та проводити реалістичніші тести відновлення. Коротше кажучи, краще сховище дозволяє частіше практикувати відновлення та досягати швидшого й надійнішого відновлення.

Коли щось йде не так, певні можливості сховища мають вирішальне значення для тривалого простою та швидкого відновлення. Нижче наведено чотири функції Pure Storage, які клієнти часто називають тими, що скорочують час відновлення та зменшують невизначеність.

### **Незмінні знімки безпечного режиму**

Безпечний режим (SafeMode) дозволяє зберегти знімки екрана справді недоторканими протягом встановленого періоду, щоб ви знали, що точка відновлення не була змінена. Ця чіткість важлива під час розслідування інциденту або відновлення після зараження програмою-вимагачем, оскільки ви можете вибрати знімок екрана та бути впевненими, що він містить необхідні вам оригінальні дані. На практиці це може заощадити години на розслідуванні, які в іншому випадку були б витрачені на перевірку цілісності резервної копії.

У безпечному режимі знімки позначаються як захищені від несанкціонованого доступу та забезпечується їх зберігання, а інтерфейс керування відображає цей стан. Ви повинні бачити чітко позначені захищені знімки, деталі зберігання, прикріплені до кожного моменту часу, та журнали аудиту, які записують будь-які спроби зміни або видалення знімків. Ці видимі

індикатори дозволяють легше довіряти знімку як надійному джерелу відновлення під час інциденту.

### **Синхронна реплікація ActiveCluster**

ActiveCluster дозволяє двом сайтам діяти як однорангові, щоб певні програми могли продовжувати працювати, навіть якщо один сайт вийшов з ладу. Для команд, які не можуть терпіти втрату даних, така симетрична реплікація означає, що транзакції зберігаються в обох місцях, а відновлення після збою відбувається за допомогою контрольованого перемикання, а не тривалого відновлення. Вигодою є менша втрата даних і менше часу, витраченого на відновлення служб після інциденту на сайті.

ActiveCluster – потужний інструмент, але він залежить від швидкої та надійної мережі й ретельного налаштування сайту. Шукайте документацію постачальників і приклади роботи клієнтів, які показують, як він поведився в реальних розгортаннях, щоб ви могли оцінити, чи працюватиме ваша мережа та поєднання програм належним чином.

### **Зменшення чистоти даних для практичного зберігання**

Скорочення обсягу даних Purity допомагає вам зберігати набагато більше знімків, не купуючи пропорційний обсяг необробленої пам'яті. Цей простий ефект дозволяє командам зберігати ближчі точки відновлення та довше їх зберігати, що безпосередньо зменшує потенційну втрату даних і надає більше можливостей під час вибору точки відновлення. Для багатьох компаній краще скорочення обсягу даних змінює план відновлення з теоретичного на практичний.

Запросіть приклади показників скорочення для робочих навантажень, подібних до вашого, і, якщо можливо, запустіть короткий пробний період або демонстрацію зі зразком ваших даних. Спостереження за тим, як ваші фактичні набори даних стискаються та дедуплікуються, – це найшвидший спосіб встановити реалістичні очікування щодо збереження та відновлення.

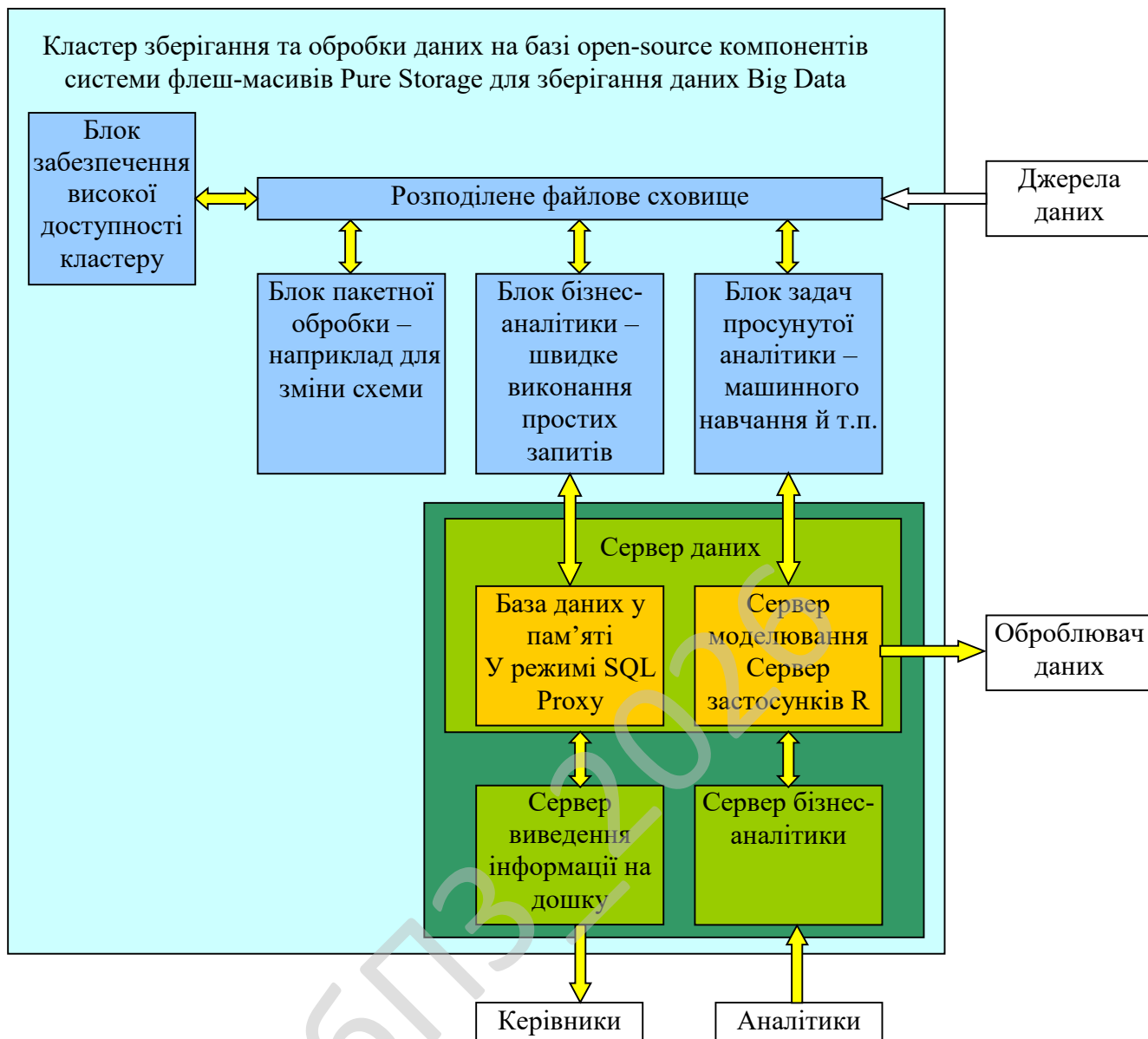


Рисунок 3.1 – Структурна схема системи

### CloudSnap та Protect для чистих копій поза масивом

CloudSnap та Protect дозволяють копіювати знімки масиву в хмарне сховище, зберігаючи при цьому деталі, що пояснюють вміст кожного знімка. Це дозволяє безпечно тестувати відновлення або проводити судово-медичні перевірки в окремому середовищі без ризику для виробничих систем. Для більшості команд перевага очевидна: надійна, ізольована копія, яку можна використовувати для тестування відповідності або відновлення.

Експорт у хмару має зберігати метадані знімків та працювати з постачальником хмарних послуг, якого ви плануєте використовувати. Перевірте, які хмарні цілі підтримуються, як метадані та контекст відновлення переносяться з експортом, а також які практичні кроки для отримання та монтування знімка з хмари. Постачальники часто публікують документацію або приклади запуску, які показують повний процес експорту та відновлення; огляд цих матеріалів допомагає вам оцінити необхідний час та зусилля.

### 3.3 Розробка функціональної схеми

Скористайтеся міццю й швидкістю системи флеш-масивів Pure Storage для зберігання даних Big Data для рішення самих складних бізнес-завдань. Системи зберігання забезпечують продуктивність сучасних твердотільних накопичувачів, а також простоту, універсальність і економічність для будь-яких цілей, будь те консолідація навантажень загального призначення, розгортання нової інфраструктури віртуальних робочих столів, великомасштабні OLTP-системи або приватна хмара на базі флеш-технологій. Навіщо іти на компроміс, здобуваючи менш комплексні рішення? Ці масиви мають всі необхідні функції, щоб ви змогли швидше приступитися до роботи із флеш-технологіями й раніше одержати результати.

Комплексна система на базі флеш-технологій допомагає вам досягти неможливого й вивести свій бізнес на новий рівень ефективності й процвітання. Орієнтована на продуктивність конфігурація "активний/активний" підтримує високе число операцій вводу-виводу в секунду в процесі горизонтального й вертикального масштабування й гарантує, що всі інвестиції в систему зберігання даних прямо підуть на прискорення роботи застосунків.

#### Динамічні інтелектуальні функції

Системи зберігання даних на базі флеш-технологій спеціально створені для адаптації до постійних змін і росту. Вони стабільно підтримують

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 37   |

навантаження в умовах непередбаченого розвитку середовища. Більше не потрібно відмовлятися від бізнес-можливостей через неадаптуєму інфраструктуру зберігання даних. Тепер, як ніколи раніше, ви зможете експериментувати з новими ідеями й міняти стратегії по ходу справи.

### **Ефективність і вигода**

Інноваційні функції автоматичної оптимізації, включаючи інтелектуальні способи дедуплікації й стиску, допомагають скоротити витрати. Повсюдно застосовувані "тонкі методи" і інтуїтивні засоби керування дозволяють використовувати ресурси по максимуму, а сама головне – вся розширена функції ПЗ включені в системах флеш-масивів Pure Storage для зберігання даних Big Data за замовчуванням. Не потрібно купувати й підтримувати ніякі додаткові ліцензії!

### **Об'єднання продуктивності і ємності до 10 масивів**

Убудований "гіпервізор сховища" дозволяє плавно переносити навантаження між масивами, вносячи зміни в систему зберігання без порушення роботи серверів. Перевірена на підприємствах операційна система флеш-масивів Pure Storage для зберігання даних Big Data автоматизує завдання керування, що займають багато часу й піддані помилкам, включаючи зіставлення LUN із серверами й виділення ресурсів RAID. Підвищені швидкість розгортання й адаптуємость конфігурацій, щоб успішно конкурувати на сучасних динамічних ринках.

### **Просте перенесення даних**

Перенесення даних усередині федерації виконується швидко й прозоро для керуючих систем. Потрібно перенести тім в інший масив? Функція динамічного переносу допоможе легко це зробити, не перериваючи виконання навантажень і зберігаючи зв'язок між моментальними копіями й даними реплікації протягом усього переносу.

## Майстри балансування навантаження

Вхідна в продукт утиліта дає поточні рекомендації з балансування навантаження, пропонуючи найкращі варіанти первісного розміщення даних і потім згодом рекомендуючи нові можливості оптимізації.

## Постійно доступна система зберігання даних

Live Volume також підтримує безперервне виконання навантажень у часонепередбувальних простоїв і катастрофічних збоїв. Автоматичне перемикавання при відмові між синхронізованими томами в локальних і віддалених масивах без переривання роботи допомагає захистити найважливіші бізнес-процеси.

## Цінність масивів на флеш-дисках – усе включено!

Тільки в системах флеш-масивів Pure Storage для зберігання даних Big Data всі розширені функції включені за замовчуванням, що забезпечує максимальну цінність і перспективність серед всіх продуктів лінійки системи флеш-масивів Pure Storage для зберігання даних Big Data. Всі функції ліцензовані для масиву повної ємності. Ви одержуєте наступне:

- Інтелектуальні методи дедуплікації й стиску.
- Динамічний перенос усередині федерації.
- Live Volume з автоматичним перемиканням при відмові.
- Технологія Data Progression.
- Розподіл даних по рівнях RAID.
- Динамічні контролери.
- Тонкі моментальні копії.
- Thin Clones ("Тонке" клонування).
- Реплікація (одновузлова, багатовузлова, "один до багатьох").
- Remote Instant Replay (синхронна/асинхронна реплікація в IP-мережах).
- Підтримка мереж (SAN) і мережних систем (NAS) зберігання даних (блокове й файлове зберігання з використанням одного пула через опціональний NAS-пристрій FS8600 з горизонтальним масштабуванням).

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 39   |

- Мережі з підтримкою декількох протоколів.
- Власні засоби відновлення застосунків.
- Веб-інтерфейси на основі HTML – Unisphere for системи флеш-масивів

Pure Storage для зберігання даних Big Data, Unisphere Central.

- Хмарна аналітика й моніторинг – CloudIQ.
- Інтерфейс клієнтського застосунку.
- Підтримка дисків із самошифруванням.
- Розрахунки між підрозділами.
- Маркування множини мереж VLAN.
- Якість обслуговування.
- Підтримка віртуальних томів VVOL.

### **Надійне рішення корпоративного класу**

І звичайно, якість системи флеш-масивів Pure Storage для зберігання даних Big Data підкріплено репутацією лідера ринку масивів на флеш-дисках і лідера в секторі систем зберігання в цілому. Продукти системи флеш-масивів Pure Storage для зберігання даних Big Data повсюдно використовуються в центрах обробки даних по усьому світі. Ви можете скористатися перевагами настільки великого досвіду в технологіях і бізнесі!

### **Можливість збільшення ємності**

Підтримувані шасі розширення:

– системи флеш-масивів Pure Storage для зберігання даних Big Data SC420F: 24 відсіку для накопичувачів 2,5" (SAS 12 Гбіт/с).

Максимальна кількість накопичувачів:

– SC7020F: 606 накопичувачів на кожний масив; більше у федеративних системах.

– SC5020F: 222 на кожний масив (з опціональними полками розширення); більше у федеративних системах.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-122.26.0033.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 40   |

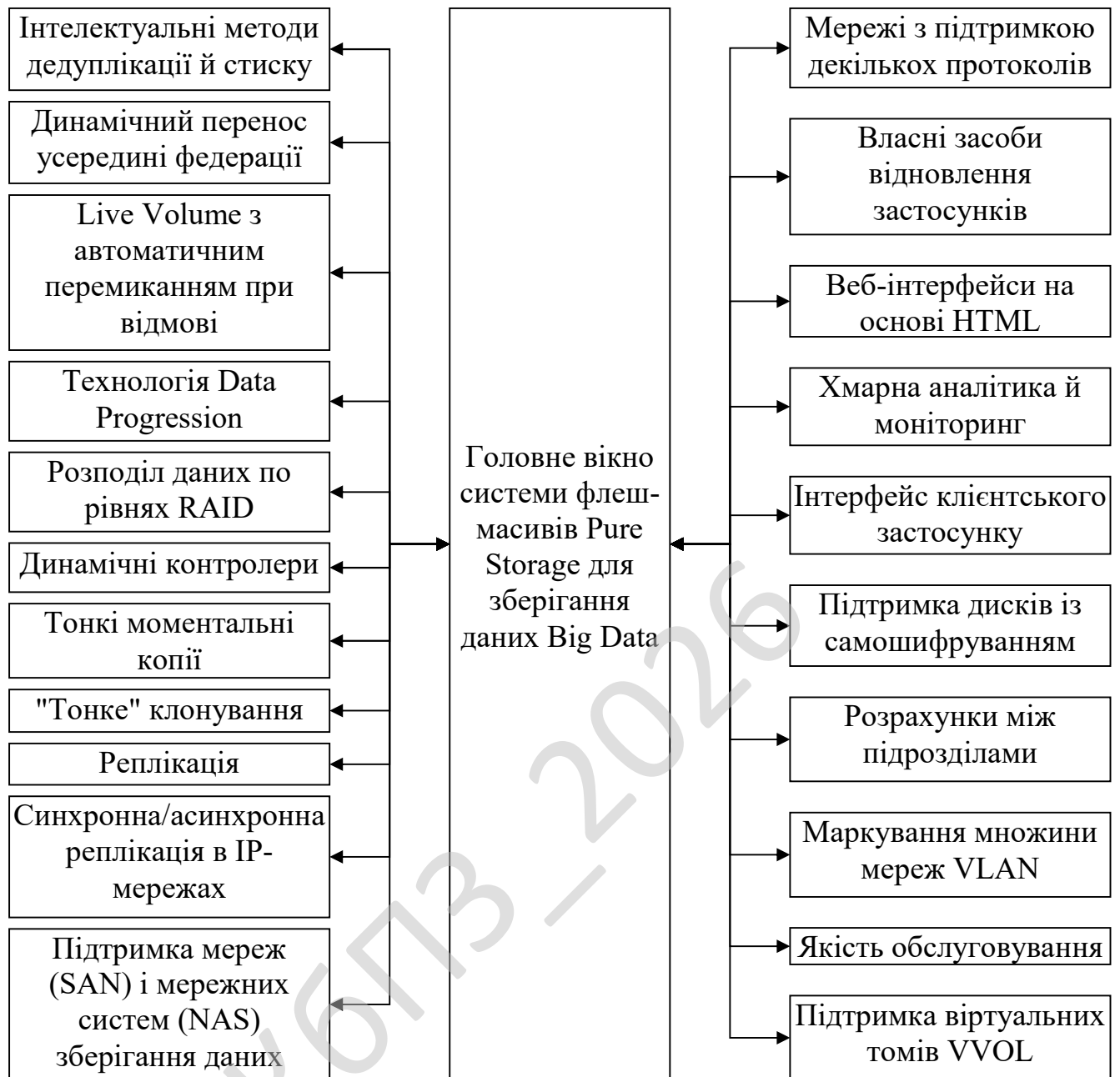


Рисунок 3.2 – Функціональна схема системи

Максимальна неформатована ємність:

- SC7020F: 4 Пбайт на кожний масив; більше у федеративних системах.
- SC5020F: 2,16 Пбайт на кожний масив; більше у федеративних системах.

Максимальна ємність без стиску даних (мережна система зберігання даних):

- 4 Пбайт у єдиному просторі імен (потрібно NAS-пристрій FS8600).
- До 20 Пбайт у глобальному просторі імен (потрібно FS8600 з декількома масивами).

Носії для зберігання даних:

- Твердотільні накопичувачі: Write Intensive і Read Intensive (можна використовувати спільно в одній системі), твердотільні накопичувачі із самошифруванням.

### Оптимізація даних

– **Метод автоматичного багаторівневого розміщення даних** Заснований на політиках перенесення даних залежно від їхнього використання в реальному часі; набудовується розмір, що, сторінки від 512 Кбайт до 4 Мбайт.

– **Структура автоматичного багаторівневого розміщення.** До 2 рівнів твердотільних накопичувачів (Write Intensive і Read Intensive)

– **Налаштування багаторівневого розміщення.** Користувальницькі профілі, можливість "закріплення" томів на будь-якому рівні

– **Підтримка RAID.** RAID 0, 1, 5, 6, RAID 10 і RAID 10 DM (подвійне дзеркалювання); можливі будь-які комбінації рівнів RAID в одному масиві

– **Розподіл даних по рівнях RAID.** Автоматичне надання ресурсів і динамічний перерозподіл даних по декількох рівнях RAID на одному рівні; не потрібне попереднє виділення груп RAID

– **Динамічне виділення ресурсів.** За замовчуванням активно на всіх томах, працює з повною продуктивністю для всіх функцій

– **Тонкі моментальні копії.** Зберігаються тільки зміни записів, моментальні копії автоматично переносяться в більше економічну підсистему зберігання даних

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 42   |

– **Інтелектуальні методи дедуплікації й стиску.** Можливість вибору для кожного тому. Також доступний варіант тільки зі стиском

### **Загальні відомості про корпус**

Якщо не зазначене інше, технічні характеристики відносяться до обох моделей масивів SC7020F і SC5020F.

### **Формат корпусу**

"Усе в одному" (дубльовані контролери, внутрішні відсіки дисководів, мережні порти й порти розширення).

### **Розмір стійки**

3U.

### **Контролери**

Два на корпус із можливістю гарячого перемикання (конфігурація "активний-активний").

### **Число процесорів на контролер**

- SC7020F: два процесори Intel® Xeon® E 5-2628 v3, 2,5 ГГц, 8 ядер.
- SC5020F: один процесор Intel® Xeon® E 5-2630 v3, 2,4 ГГц, 8 ядер.

### **Ємність внутрішнього сховища**

30 відсіків для накопичувачів 2,5".

### **Системна пам'ять**

- SC7020F: 256 Гбайт на кожний масив (128 Гбайт на кожний контролер).
- SC5020F: 128 Гбайт на кожний масив (64 Гбайт на кожний контролер).

### **Операційна система**

Dell Storage Center OS (SCOS) 7.2 або більше пізньої версії.

### **Мережа й розширення підсистеми вводу-виводу**

### **Мережні протоколи зовнішнього інтерфейсу**

FC, iSCSI (підтримка одночасно декількох протоколів).

### **Максимальне число портів FC 32 Гбіт/с**

- SC7020F: 24 на кожний масив (SFP+).
- SC5020F: 8 на кожний масив (SFP+).

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-122.26.0033.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 43   |

**Максимальне число портів FC 8/16 Гбіт/с**

- SC7020F: 24 на кожний масив (SFP+).
- SC5020F: 8 на кожний масив (SFP+).

**Максимальне число портів iSCSI 100 Гбіт/с**

- SC7020F: 16 QSFP28 на кожний масив.
- SC5020F: 8 QSFP28 на кожний масив.

**Максимальне число портів iSCSI 25 Гбіт/с**

- SC7020F: 16 SFP28 на кожний масив.
- SC5020F: 8 SFP28 на кожний масив.

**Максимальне число портів iSCSI 10 Гбіт/с**

- SC7020F: 32 SFP+ (10 Гбіт/с) або Base-T на кожний масив.
- SC5020F: 16 SFP+ (10 Гбіт/с) або Base-T на кожний масив.

**Порти керування**

2 на кожний масив (BASE-T 1 Гбіт/с).

**Внутрішні протоколи розширення**

SAS 12 Гбіт/с.

**Максимальне число внутрішніх портів розширення**

- SC7020: 24 на кожний масив.
- SC5020: 8 на кожний масив.
- До 16 корпусів розширення на кожний масив.

**Функціональність****Конфігурації масиву**

Тільки повні флеш-масиви.

**Формат зберігання даних**

Блоковий (SAN) і/або файловий (NAS) із загального пула з опціональною системою FS8600.

**Максимальна кількість хостів SAN: 500.**

**Максимальна кількість портів ініціатора: 1 000.**

**Максимальна розмір LUN: 500 Тбайт.**

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-122.26.0033.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 44   |

**Максимальна кількість LUN: 2 000.**

**Максимальне число моментальних копій:**

– SC7020F: 16 384.

– SC5020F: 8 192.

**Максимально операцій вводу-виводу в секунду:**

– SC7020F: 1 200 000.

– SC5020F: 1 025 000.

**Максимально операцій вводу-виводу в секунду (із затримкою менш 1 мс):**

– SC7020F: 1 050 000.

– SC5020F: 818 000.

**Максимально операцій вводу-виводу в секунду (80% читання, 20% запис):**

– SC7020F: 346 000.

– SC5020F: 330 000.

**Максимальна пропускна здатність (читання):**

– SC7020F: 29 000 Мбайт/с.

– SC5020F: 19 000 Мбайт/с.

**Максимальна пропускна здатність (запис):**

– SC7020F: 14 000 Мбайт/с.

– SC5020F: 9 500 Мбайт/с.

**Мобільність і перенесення даних**

**Реплікація**

– Реплікація з іншими масивами серії SC.

– Синхронна або асинхронна реплікація по протоколах FC або iSCSI.

– Взаємозв'язок мети/джерела може мати тип "один до багатьох" або "багато до одного".

– Підтримує всі служби даних системи флеш-масивів Pure Storage для зберігання даних Big Data на томах джерела й мети.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 45   |

- Зміна типу реплікації й топологій на вимогу.
- Підтримка крос-платформної реплікації з масивами серії PS/EqualLogic (у будь-якому напрямку).

#### **Мобільність томів**

Вхідна до складу базового продукту функція Live Migrate забезпечує прозорий для вузла перенесення даних між масивами.

#### **Федеративні системи з декількома масивами**

- Вхідна до складу базового продукту функція Live Migrate забезпечує прозорий для вузла перенесення даних між масивами.
- Моментальні копії зберігаються під час переносу.

#### **Thin Import ("Тонкий" імпорт)**

Ефективно використовує простір перенесення даних з масивів серій PS (EqualLogic) і MD3 без порушення роботи.

#### **Thin Clones ("Тонке" клонування)**

- Клонування окремих томів з нульовою дедуплікацією даних.
- Клони підтримують незалежні моментальні копії й реплікацію.
- Оптимально підходять для інфраструктури віртуальних робочих столів, тестування й розробки, а також інших застосунків, яким потрібні дискретні екземпляри загальних даних.
- Більше ефективний спосіб, чим дедуплікація для копій баз даних.

#### **Захист даних, аварійне відновлення, безпека**

##### **Безперервність бізнесу**

- Компонент Live Volume забезпечує двостороннє автоматичне перемикання й відновлення при відмові.
- Безперервна робота, відновлення після збоїв, запобігання збоїв.
- Компонент Live Volume Managed Replication підтримує варіанти реплікації з додатковим третім вузлом.

– Нульові значення цільового часу відновлення/цільової крапки відновлення з набудовується соглашением, що, про рівень обслуговування при перемиканні вузла при відмові для кожного тому.

- Підтримка технології кластера VMware Metro Stretch.
- VMware Site Recovery Manager.

### **Розподілене резервування**

Не потрібно виділення дисків гарячого резерву, всі диски доступні для читання й запису.

### **Тонкі моментальні копії**

Зберігаються тільки зміни записів, моментальні знімки автоматично переносяться на більше економічну підсистему зберігання даних.

### **Погодженість із додатками**

- Replay Manager: погоджені з додатками моментальні копії в середовищах Microsoft або VMware.
- AppSync: погоджені з додатками моментальні копії в середовищах Microsoft, VMware і Oracle.

### **Шифрування неактивних даних**

- Підтримка дисків із самошифруванням.
- Повне шифрування дисків (FDE) за стандартом AES-256.
- Накопичувачі сертифіковані у відповідності зі стандартом FIPS 140-2 рівні 2.
- Для FIPS 140-2 рівнів 1, 2 і 3 доступні варіанти із сервером керування ключами (KMS).

### **Підтримка зовнішнього диспетчера ключів**

- SafeNet KeySecure k460, SafeNet KeySecure k250, SafeNet KeySecure k150v компанії Gemalto.
- Thales EMS 200.

### **Умови експлуатації**

**Температура при експлуатації:** від 10 до 35 °C (від 50 до 95 °F).

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-122.26.0033.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 47   |

**Температура при зберіганні:** від -40 до 65 °C (від -40 до 149 °F).

**Вологість при експлуатації (без конденсації):** від 10 до 80% з максимальною крапкою роси 29 °C (84,2 °F).

**Вологість при зберіганні (без конденсації):** від 5 до 95% з максимальною крапкою роси 33 °C (91 °F).

**Тип рознімання живлення:** NEMA 5-15/CS22.2, n°42.

**Живлення**

**Живлення/потужність:**

– SC7020F: два джерела живлення потужністю 1 485 Вт із можливістю гарячого перемикавання.

– SC5020F: два джерела живлення з можливістю гарячого перемикавання (варіанти потужністю 1 378 або 1 485 Вт).

**Максимальна вихідна потужність:**

– SC7020F: 1 485 Вт.

– SC5020F: 1 378 Вт (варіант БП 1 378 Вт) або 1 485 Вт (варіант БП 1 485 Вт).

**Максимальна вхідна потужність:**

– SC7020F: 1 688 Вт.

– SC5020F: 1 584 Вт (варіант БП 1 378 Вт) або 1 688 Вт (варіант БП 1 485 Вт).

**Максимальна вхідний струм:**

– SC7020F: 8,8 А.

– SC5020F: 16 А (варіант БП 1 378 Вт) або 8,8 А (варіант БП 1 485 Вт).

**Максимальна пусковий струм:**

55 А в плин 10 мс або менш.

**Номінальний робочий діапазон вхідної напруги:**

– SC7020F: 200-240 У змінного струму.

– SC5020F: 100-240 У змінного струму (варіант БП 1 378 Вт) або 200-240В змінного струму (варіант БП 1 485 Вт).

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 48   |

**Номинальна вхідна частота: 50/60 Гц**

**Максимальна тепловиділення/тепловіддача:**

– SC7020F: 5 760 БТЕ/год.

– SC5020F: 5 770 БТЕ/год (варіант БП 1 378 Вт) або 5 760 БТЕ/год (варіант БП 1 485 Вт).

### **Керування**

#### **Інтерфейси керування**

– Інтерфейс на основі браузера (HTML 5).

– Unisphere for системи флеш-масивів Pure Storage для зберігання даних Big Data (менеджер компонентів).

– Unisphere Central (керування декількома масивами).

– CloudIQ (хмарна аналітика й моніторинг).

– Клієнтський застосунок.

– Міжплатформене керування декількома масивами, вузлами й об'єктами.

#### **Об'єднання у федерацію**

Створення великих систем з декількома масивами за допомогою уніфікованого керування із плавним переносом навантаження між масивами за допомогою функції Live Migrate. Додавання масивів будь-якого типу без переривання роботи, ефективне використання їхньої сукупної ємності й продуктивності. Утиліта Volume Advisor контролює кілька масивів, пропонуючи варіанти оптимального розміщення даних і балансування навантаження. Переміщення томів не впливає на моментальні копії або захист даних реплікації.

#### **Підтримка сценаріїв**

– Microsoft PowerShell API.

– RESTful API.

#### **Підтримка серверних ОС**

– Microsoft® Windows Server®, Oracle® Solaris, HP®-UX, Oracle Linux, IBM® AIX®, Novell® NetWare, SLES, Apple, HPTru64, VMware®, Citrix® XenServer®, RedHat®.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-122.26.0033.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 49   |

## Інтеграція сторонніх застосунків

VMware, Microsoft, IBM, OpenStack, Symantec, Commvault, Foglight, Docker

### Сумісність із масивами серії PS

- Двостороння реплікація.
- Керування щоденними завданнями через єдиний інтерфейс.
- Thin Import ("Тонкий" імпорт): ефективно використовує простір перенесення даних без порушення роботи з масивів серії PS.

## 3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей.

Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграми потоків даних містять чотири типи елементів:

- Зовнішні по відношенню до системи сутності.
- Потоки даних між елементами трьох попередніх типів.
- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Сховища даних (репозиторії).



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-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 52   |

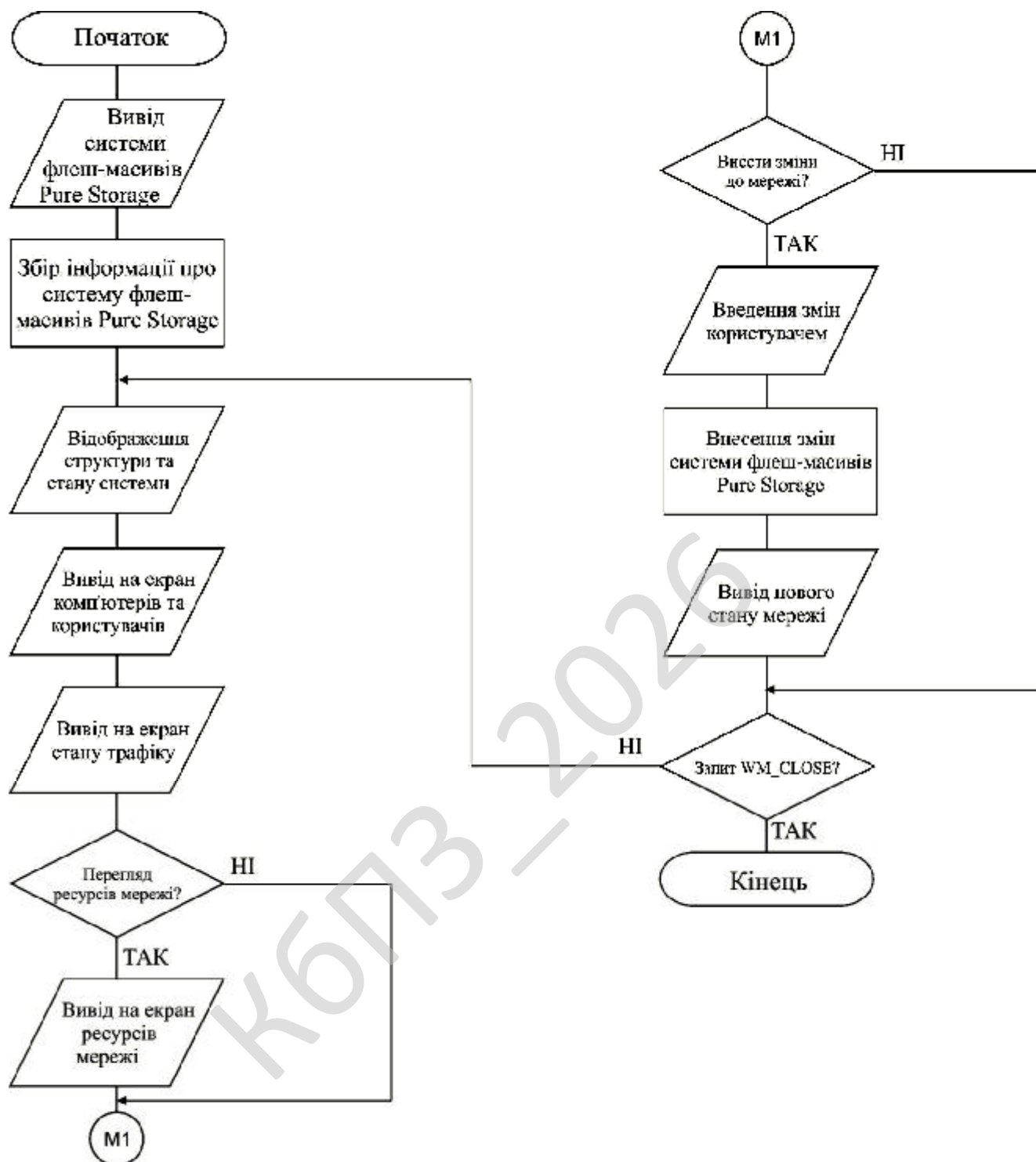


Рисунок 4.1 – Блок-схема основної програми

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених

мережних програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

- рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 54   |



Рисунок 4.2 – Блок-схема роботи підпрограми

- прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;
- рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

- модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;
- модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Для того, щоб людина, яка працює в Інтернеті, могла переглянути ту чи іншу сторінку, на її комп'ютері повинно бути встановлено відповідне програмне забезпечення. Програми для перегляду веб-сторінок називаються браузером.

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є

важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

Для запиту до веб-сервера клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма, назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит, відповідь пересилається по мережі клієнтові.

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка ПЗ. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів ПЗ, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Для роботи з системою користувач використовує стандартне програмне забезпечення – звичайний браузер. Це позбавляє його необхідності завантажувати та інсталиувати спеціальні програми (хоча інколи така необхідність все-таки виникає).

Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних. Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 57   |

обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

Найчастіше веб-сервер і серверні модулі проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі.

На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв PHP, а для управління даними – СУБД MySQL. Таким чином, зв'язку PHP-MySQL слід розглядати як стандартний інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі. Водночас як засоби управління даними, так і middleware-засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Delphi.

Взагалі, технології створення розподілених, зокрема веб-програм, стрімко розвиваються. Слід згадати про технології EJB (Enterprise Java Beans), CORBA, а також про .NET – порівняно нову ініціативу компанії Microsoft. Для зберігання даних та їх передачі часто використовується так звана розширювана мова розмітки XML (Extensible Markup Language).

Незважаючи на те що я працював над ПЗ один в реалізації програми я використовував підходи пришвидшення розробки на основі методологій Agile – Extreme Programming.

Екстремальне програмування (Extreme Programming, далі XP) це методологія розробки програмного забезпечення, найпопулярніша серед так званих гнучких методологій. Має на меті поліпшення якості програмного забезпечення та чутливість до змін у вимогах замовників. Як вид гнучких методологій, XP радить часті "випуски" програми у коротких циклах розробки, що має на меті поліпшити продуктивність праці та покращити можливості виконання вимог замовника що змінюються. Авторами даної методології є Кент Бек, Ворд Каннінгем, Мартін Фаулер та інші.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 58   |

Інші елементи екстремального програмування включають в собі: парне програмування, проведення обширної перевірки сирцевого коду, модульне тестування всього коду, уникання створення функціональності до того як вона дійсно необхідна, простота та ясність коду, очікування на зміну вимог замовників з плином часу та коли вимоги до продукту стають ясніші, досить часте спілкування із замовником та між самими програмістами.

Назва методології походить від ідеї застосувати корисні методи і практики розробки програмного забезпечення, піднявши їх до "екстремальних" рівнів.

Критики ХР зауважують на потенційні недоліки цієї методології – нестабільні вимоги, незадокументовані компроміси конфліктів користувачів, відсутність загального документу дизайну програми.

Технологія екстремального програмування була розроблена Кентом Беком, Уардом Каннінгхемом та Роном Джеффріесом під час роботи над Chrysler Comprehensive Compensation System (C3). У 1996 Кент Бек став лідером проекту і почав вдосконалювати методи розробки, що застосовувалися в роботі над проектом. Свій метод він виклав у книзі «Extreme Programming Explained», котру було видано у жовтні 1999. Після купівлі Крайслера компанією Даймлер-Бенц проект C3 було скасовано у лютому 2000.

Хоча саме екстремальне програмування є відносно новим, багато її практик вже існували і використовувались протягом певного часу; однак, методологія підносить "найкращі практики" до екстремального рівня. Для прикладу, практика по плануванню і написанню тестів перед написанням кожної маленької частини коду було використано раніше в проекті НАСА "Меркурій". Для зменшення часу на розробку ПЗ деякі формальні документи тестування (такі як приймальне тестування) писались паралельно (або й раніше) з написанням самого ПЗ. Незалежна група тестування НАСА може писати процедури тестування базуючись на формальних вимогах до продукту до того як програмне забезпечення розроблене та інтегроване в систему. В ХР ця концепція піднесена до "екстремального рівня" завдяки написанню автоматичних тестів які

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 59   |

перевіряють поведінку навіть малих частинок коду, а не тільки значних функціональних частин ПЗ.

Посібник Extreme Programming Explained: Embrace Change описує Екстремальне Програмування, як:

- Спроба примирити гуманність і продуктивність.
- Механізм для соціальної зміни.
- Шлях до удосконалення.
- Стиль розвитку.

Дисципліна розробки програмного забезпечення.

Головною метою Екстремального Програмування є скорочення вартості неочікуваних змін. У традиційних методах розробки (на кшталт SSADM) вимоги до розвитку системи визначаються на початку роботи над проектом, і часто виправляються пізніше. Це означає, що вартість проекту через зміни буде більшою за заплановану (традиційна особливість для програмного забезпечення, що проектується).

ХР використовується для скорочення вартості змін, завдяки представленню простих значень, принципів і методів. При використанні екстремального програмування, проект повинен стати гнучкішим щодо змін.

Extreme Programming Explained описує екстремальне програмування як дисципліну розробки програмного забезпечення яка змушує людей створювати високоякісне ПЗ якомога швидше.

ХР намагається зменшити ціну зміни вимог до ПЗ завдяки малим циклам розробки, а не одним довгим циклом. Екстремальне програмування сприймає зміни до вимог як звичайні, неминучі та бажані аспекти розробки ПЗ, і ці зміни мають бути очікуваним. Основна ідея полягає в тому що неможливо розробити самодостатній пакет вимог до ПЗ, зміни в вимогах – неминучі.

Екстремальне програмування також вводить набір практик та принципів на основі методології гнучкої розробки програмного забезпечення.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 60   |

Екстремальне програмування описує чотири базові активності що виконуються при розробці програмного забезпечення: написання коду, тестування, слухання та дизайн.

Написання коду. Прихильники ХР заявляють що єдиним дійсно важливим результатом розробки ПЗ є код: без готового коду нема продукту.

Тестування. Методологія екстремального програмування заявляє, що якщо дрібне тестування може перевірити незначну частину функціональності, то багато дрібних тестів можуть перевірити набагато більше частинок і продукт в цілому.

Основні прийоми ХР. Дванадцять основних прийомів екстремального програмування (за першим виданням книги Extreme programming explained) можуть бути об'єднані в чотири групи:

1. Короткий цикл зворотного зв'язку (Fine scale feedback).
  - 1.1. Розробка через тестування (Test driven development).
  - 1.2 Гра в планування (Planning game).
  - 1.3. Замовник завжди поруч (Whole team, Onsite customer).
  - 1.4 Парне програмування (Pair programming).
2. Безперервний, а не пакетний процес.
  - 2.1 Безперервна інтеграція (Continuous Integration).
  - 2.2 Рефакторинг (Design Improvement, Refactor).
  - 2.3 Часті невеликі релізи (Small Releases).
3. Розуміння, що поділяється всіма учасниками.
  - 3.1 Простота (Simple design).
  - 3.2 Метафора системи (System metaphor).
  - 3.3 Колективне володіння кодом (Collective code ownership) або обраними шаблонами проектування (Collective patterns ownership).
  - 3.4 Стандарт кодування (Coding standard or Coding conventions).
4. Соціальна захищеність програміста (Programmer welfare), а саме 40 годинний робочий тиждень (Sustainable pace, Forty hour week).

Redmine – вільне серверне ПЗ для управління проектами та відстежування помилок. До системи входить календар-планувальник та діаграми Ганта для візуального представлення ходу робіт за проектом та строків виконання. Redmine написано на мові Ruby і є ПЗ розробленим з використанням відомого веб-фреймворку Ruby on Rails, що означає легкість в розгортанні системи та її адаптації під конкретні вимоги. Для кожного проекту можна вести свої вікі та форуми.

Функціональні можливості:

- Ведення декількох проектів.
- Гнучка система доступу з використанням ролей.
- Система відстеження помилок.
- Діаграми Ганта та календар.
- Ведення новин проекту, документів та управління файлами.
- Сповіщення про зміни за допомогою RSS-потоків та електронної пошти.
- Власна Wiki для кожного проекту.
- Форуми для кожного проекту.
- Облік часових витрат.
- Налаштування власних (custom) полів для задач, затрат часу, проектів та користувачів.
- Легка інтеграція із системами керування версіями (SVN, CVS, Git, Mercurial, Bazaar и Darcs).
- Створення записів про помилки на основі отриманих листів
- Підтримка LDAP автентифікації.
- Можливість самореєстрації нових користувачів.
- Багатомовний інтерфейс (у тому числі українська мова).
- Підтримка СКБД: MySQL, PostgreSQL, SQLite.

Діаграма Ганта (*Gantt chart*, також стрічкова діаграма, графік Ганта) – це популярний тип діаграм, який використовується для ілюстрації плану, графіка

робіт за будь-яким проектом. Є одним з методів планування та управління проектами.

Діаграма Ганта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

Діаграма може використовуватися для представлення поточного стану виконання робіт: частина прямокутника, що відповідає завданню, заштриховується, відзначаючи відсоток виконання завдання; показується вертикальна лінія, що відповідає моменту «сьогодні».

Часто діаграма Ганта використовується спільно з таблицею зі списком робіт, рядки якої відповідають окремо взятій задачі, зображеній на діаграмі, а стовпці містять додаткову інформацію про задачу.

Система відстеження помилок Багтрекер – прикладна програма для допомоги розробникам програмного забезпечення (програмістам, тестувальникам тощо) враховувати і контролювати помилки, знайдені у програмах, питання щодо функціональності, рішення та оновлення, побажання користувачів, а також стежити за процесом їх виконання.

Кожному, хто розробляв програмні продукти, добре знайоме співвідношення «20/80» – останні 20 % роботи тривають 80 % часу.

Як це не парадоксально, але нічого дивного в цій пропорції немає, адже саме на завершальній стадії починається тестування проекту, коли виявляються помилки, і що більший проект, то більше буде знайдено помилок.

Водночас досить часто виявляється, що більшість цих помилок були відомі та могли бути виправлені з меншими витратами на попередніх стадіях роботи, але не були вчасно описані, а потім загубилися серед інших важливих завдань.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 63   |

Отже, система відстеження помилок у найпростішому варіанті – це процес, що включає в себе виявлення помилки, її опис, виправлення і перевірку цього виправлення, тобто процес «стеження» за багом протягом всього як його життєвого циклу, так і життєвого циклу розробки в цілому.

Сукупність інформації про дефект. Головний компонент такої системи – база даних, що містить відомості про виявлені дефекти. Ці відомості можуть включати в себе:

- номер (ідентифікатор) дефекту;
- хто повідомив про дефект;
- дата і час виявлення дефекту;
- версія продукту, в якій виявлено дефект;
- серйозність (критичність) дефекту та пріоритет рішення;
- опис кроків для відтворення дефекту (неправильної поведінки програми);
- відповідальний за усунення дефекту;
- обговорення можливих рішень та їх наслідків;
- поточний стан виправлення дефекту;
- версії продукту, в якій дефект виправлений.

Крім того, розвинені системи надають можливість прикріплювати файли, які допомагають описати проблему, наприклад, дамп пам'яті або скріншот.

Використання. Основна перевага систем відстеження помилок полягає в забезпеченні чітких централізованих оглядів, запитів на розробку (включаючи помилки і виправлення) та їх стан. У корпоративному середовищі, системи відстеження помилок можуть бути використані для генерації звітів по продуктивності програмістів виправлення помилок. Однак, це може іноді приводити до неточних результатів, тому що різні помилки можуть мати різні ступені пріоритету та серйозності, що пов'язано з складністю їх фіксації.

Життєвий цикл дефекту. Як правило, система відстеження помилок використовує той чи інший варіант «життєвого циклу» помилки, стадія якого визначається поточним станом помилки.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 64   |

Типовий життєвий цикл дефекту:

1. Новий – дефект зареєстрований тестувальником.
2. Призначений – призначений відповідальний за виправлення дефекту.
3. Дозволений – дефект переходить назад у сферу відповідальності тестувальника. Як правило, супроводжується резолюцією, наприклад:

– Виправлено (виправлення включені у версію таку-то).

- Дубль (повторює дефект, що вже знаходиться в роботі).
- Не виправлено (працює відповідно до специфікації, має занадто низький пріоритет, виправлення відкладено до наступної версії тощо).

– «В мене все працює» (запит додаткової інформації про умови, в яких дефект проявляється).

4. Далі тестувальник проводить перевірку виправлення, залежно від чого дефект або знову переходить у стан «Призначений» (якщо він описаний як виправлений, але не виправлений), або у стан «Закрито».

5. Відкрито повторно – дефект знайдено знову в іншій версії.

Система може надавати адміністраторові можливість налаштування користувачі, які можуть переглядати і редагувати помилки залежно від їх стану, переводити їх в інший стан або видаляти.

У корпоративному середовищі, система відстеження помилок може використовуватися для отримання звітів, що показують продуктивність програмістів при виправленні помилок. Однак, часто такий підхід не дає достатньо точних результатів через те, що різні помилки мають різну ступінь серйозності та складності. При цьому серйозність проблеми прямо не стосується складності її усунення.

**Microsoft SQL Server** – комерційна система керування базами даних, що розповсюджується корпорацією Microsoft.

Мова, що використовується для запитів – Transact-SQL, створена спільно Microsoft та Sybase.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 65   |

Transact-SQL є реалізацією стандарту ANSI/ISO щодо структурованої мови запитів SQL із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних.

### **Походження**

Базовий код MS SQL Server (до версії 7.0) ґрунтувався на коді Sybase SQL Server. Це дозволило Microsoft вийти на ринок баз даних для підприємств, де конкурували Oracle, IBM, і, пізніше, сама Sybase. Microsoft, Sybase і Ashton-Tate спочатку об'єдналися для створення і випуску на ринок першої версії програми, що отримала назву SQL Server 1.0 для OS/2 (близько 1989 року), яка фактично була еквівалентом Sybase SQL Server 3.0 для Unix, VMS та ін. Microsoft SQL Server 4.2 був випущений у 1992 році та входив до складу операційної системи Microsoft OS/2 версії 1.3.

Офіційний реліз Microsoft SQL Server версії 4.21 для ОС Windows NT відбувся одночасно з релізом самої Windows NT (версії 3.1). Microsoft SQL Server 6.0 був першою версією SQL Server, створеною виключно для архітектури NT і без участі в процесі розробки Sybase.

До того часу, як вийшла на ринок ОС Windows NT, Sybase і Microsoft розійшлися та створювали вже власні моделі цього програмного продукту. Microsoft намагалася отримати виняткові права на всі версії SQL Server для Windows. Пізніше Sybase змінила назву свого продукту на Adaptive Server Enterprise щоб уникнути плутанини з Microsoft SQL Server. До 1994 року Microsoft отримала від Sybase три повідомлення про авторські права як натяк на походження Microsoft SQL Server.

Після розділення компанії зробили декілька самостійних релізів програм. SQL Server 7.0 був першим сервером баз даних зі справжнім графічним інтерфейсом адміністрування. Для усунення претензій з боку Sybase у порушенні авторських прав, весь успадкований код в сьомій версії був переписаний. Це

забезпечило також й успіх SQL Server 2000, який був першою редакцією, орієнтованою на архітектуру IA-64.

Протягом подальших шести років корпорація Microsoft працювала над вдосконаленням вже існуючої версії SQL Server 2000 доки не збудувала зручнішу систему Microsoft SQL Server 2005. Були вдосконалені продуктивність, клієнтські інструменти інтегрованого середовища розробки, а також у декількох додаткових системах, що встановлюються разом із SQL Server 2005. Змінено: інструментарій процесів керування сховищами даних (SQL Server Integration Services або SSIS), сервер звітів, сервер OLAP та інтелектуального аналізу даних (Analysis Services), а також декілька технологій повідомлень, особливо Service Broker та Notification Services.

### **SQL Server 2012+**

SQL Server 2012 включає низку вдосконалень для роботи з критичними бізнес-застосунками і бізнес-аналітикою як в традиційних дата-центрах, так і в приватних, публічних і гібридних хмарних середовищах. Серед нових можливостей SQL Server 2012 виділяються SQL Server AlwaysOn (рішення підтримки високого рівня доступності даних та аварійного відновлення), xVelocity (технологія збільшення продуктивності сховищ даних та програм бізнес-аналітики), нові рішення в області візуалізації PowerPivot і PowerView для створення звітів і аналітичних програм з Excel і SharePoint, покращені інструменти для інтеграції даних і управління ними, включаючи SQL Server Data Quality Services і Master Data Services, нова семантична модель бізнес-аналітики та інструмент для адміністраторів баз даних і розробників застосунків SQL Server Data Tools.

Також Microsoft зробила значні інвестиції в області Big Data, а саме в інтеграцію SQL Server і популярних інструментів для бізнес-аналітики з неструктурованою інформацією.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-122.26.0033.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 67   |

Microsoft робить SQL Server доступним у різноманітних варіантах, які різняться наборами властивостей в залежності від цілей кінцевого користувача. Це такі редакції як:

SQL Server Compact Edition (SQL CE) – Компактний механізм бази даних. Завдяки малому обсягу (2 Мб для DLL) має зменшені властивості у порівнянні з іншими варіантами. Розмір бази даних обмежений 4 Гб і не може використовуватися як служба Windows.

SQL Server Express Edition – Раніше відомий під назвою MSDE (Microsoft SQL Server Desktop Engine), Microsoft SQL Server Express є вільно поширюваною версією SQL Server. Дана версія має деякі технічні обмеження, також відсутні графічні інструменти адміністрування. Такі обмеження роблять її непридатною для розгортання великих баз даних. В основному вона використовується у застосунках, при проектуванні, або для самостійного вивчення. Розмір бази даних обмежений 4 Гб, розмір пам'яті, що може бути адресованою-1 Гб, підтримує лише один процесор.

SQL Server Workgroup Edition – SQL Server Workgroup Edition включає функціональність ядра бази даних, але не включає додаткові сервіси.

SQL Server Standard Edition – SQL Server Standard edition включає механізми ядра бази даних, а також автономні сервіси. Відрізняється від варіанту Enterprise Edition тим, що підтримує менше активних вузлів та не включає деякі функції збільшення продуктивності.

SQL Server Enterprise Edition – SQL Server Enterprise Edition-це повнофункціональна версія SQL Server

SQL Server Developer Edition – SQL Server Developer Edition включає ті самі функції, що й SQL Server Enterprise Edition, але містить обмеження щодо використання його лише для розробки та тестування. Його ліцензія не дозволяє використання як виробничого сервера.

## Функціональність

Microsoft SQL Server як мову запитів використовує версію SQL, що отримала назву Transact-SQL (скорочено T-SQL), яка є реалізацією SQL-92 (стандарт ISO для SQL) з багатьма розширеннями. T-SQL дозволяє використовувати додатковий синтаксис процедур, що зберігаються і забезпечує підтримку транзакцій (взаємодія бази даних з керуючим застосунком).

Microsoft SQL Server та Sybase ASE для взаємодії з мережею використовують протокол рівня застосунка під назвою Tabular Data Stream (TDS, протокол передачі табличних даних).

Microsoft SQL Server також підтримує Open Database Connectivity (ODBC)-інтерфейс взаємодії застосунків з СУБД. Версія SQL Server 2005 надає можливість підключення користувачів через веб-сервер-сервіси, що використовують протокол SOAP. Це дозволяє клієнтським програмам, не призначеним для Windows, кроссплатформенно з'єднуватися з SQL Server. Microsoft також випустила сертифікований драйвер JDBC, що дозволяє застосункам під керування Java (таким як BEA і IBM Websphere) з'єднуватися з Microsoft SQL Server 2000 і 2005.

SQL Server підтримує дзеркалювання та кластеризацію баз даних. Кластер серверу SQL-це сукупність однаково конфігурованих серверів; така схема допомагає розподілити робоче навантаження між декількома серверами. Усі сервери мають одне віртуальне ім'я, а дані розподіляються за IP-адресами машин кластеру протягом робочого циклу. Також у разі відмови або збою на одному з серверів кластеру доступне автоматичне перенесення навантаження на інший сервер.

SQL Server підтримує надлишкове дублювання даних за трьома сценаріями:

1. Знімок: Виконується «знімок» бази даних, який сервер відправляє одержувачам.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 69   |

2. Історія змін: Всі зміни бази даних безперервно передаються користувачам.

3. Синхронізація з іншими серверами: Бази даних декількох серверів синхронізуються між собою. Зміни усіх баз даних відбуваються незалежно на кожному сервері, а під час синхронізації відбувається звірка даних. Дублювання такого типу передбачає можливість вирішення протиріч між базами даних.

SQL Server 2005 має вбудовану підтримку .NET Framework. Завдяки цьому, процедури бази даних, що зберігаються, можуть бути написані на будь-якій мові платформи .NET з використанням повного набору бібліотек, доступних для .NET Framework. На відміну від інших процесів, .NET Framework виділяє додаткову пам'ять і будує засоби керування SQL Server, не використовуючи вбудовані засоби Windows. Це підвищує продуктивність порівняно із загальними алгоритмами Windows, оскільки алгоритми розподілу ресурсів спеціально налагоджені для використання у структурах SQL Server.

#### 4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм RC5, який являє собою блоковий шифр із безліччю параметрів: розміром блоку, розміром ключа й числом раундів. В алгоритмі RC5 передбачені три операції: XOR, додавання й циклічні зрушення. На більшості процесорів операції циклічного зрушення виконуються за постійний час, змінні циклічні зрушення являють собою нелінійну функцію. Циклічні зрушення залежать як від ключа, так і від даних.

В RC5 використовується блок змінної довжини, але в приводиться прикладі, що буде розглянутий, 64-бітовий блок даних. Шифрування використовує  $2r+2$  залежних від ключа 32-бітових слів –  $S_0, S_1, S_2, \dots, S_{2r+1}$  – де  $r$  – число раундів. Для шифрування спочатку потрібно розділити блок відкритого тексту на два 32-бітових слова:  $A$  и  $B$ . (При впакуванні байтів у слова в алгоритмі

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 70   |

RC5 дотримується угода про прямий порядок (little-endian) байтів: перший байт займає молодші біти регістра  $A$  й т. ін.) Потім:

$$A = A + S_0, B = B + S_0$$

Для  $i$  від 1 до  $r$ :

$$A = ((A \oplus B) \lll B) + S_{2i}, B = ((B \oplus A) \lll A) + S_{2i+1}$$

Вихід перебуває в регістрах  $A$  и  $B$ .

Розшифрування теж нескладно. Потрібно розбити блок відкритого тексту на два слова,  $A$  й  $B$ , а потім:

Для  $i$  від  $r$  до 1 із кроком -1:

$$B = ((B - S_{2i+1}) \ggg A) \oplus A, A = ((A - S_{2i}) \ggg B) \oplus B$$

$$B = B - S_i, A = A - S_0$$

Символом « $\ggg$ » позначене циклічне зрушення вправо. Звичайно ж, всі додавання й вирахування виконуються по модулю  $2^{32}$ .

Створення масиву ключів складніше, але теж прямолінійно. Спочатку байти ключа копіюються в масив  $L$  із 32-бітових слів, доповнюючи при необхідності заключне слово нулями. Потім масив  $S$  ініціалізується за допомогою лінійного конгруентного генератора по модулю  $2^{32}$ :

$$S_0 = P$$

Для  $i$  від 1 до  $2(r+1) - 1$ :

$$S_i = (S_{i-1} + Q) \bmod 2^{32}$$

де  $P = 0xb7e15163$  і  $Q = 0x9e3779b9$ .

Нарешті, потрібно підставити  $L$  в  $S$ :

$$i = j = 0, A = B = 0$$

Виконати  $3n$  раз (де  $n$  – максимум від  $2(r+1)$  і  $c$ ):

$$A = S_i = (S_i + A + B) \lll 3$$

$$B = L_i = (L_i + A + B) \lll (A + B)$$

$$i = (i + 1) \bmod 2(r+1)$$

$$j = (j + 1) \bmod c$$

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у бакалаврської дипломної роботі система флеш-масивів Pure Storage для зберігання даних Big Data. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи: Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші; Верхнього меню; Розділу виведення результату роботи системи; Розділу обрання групи; Функції представлені у графічному вигляді (іконки); Функціональних кнопок ПЗ.

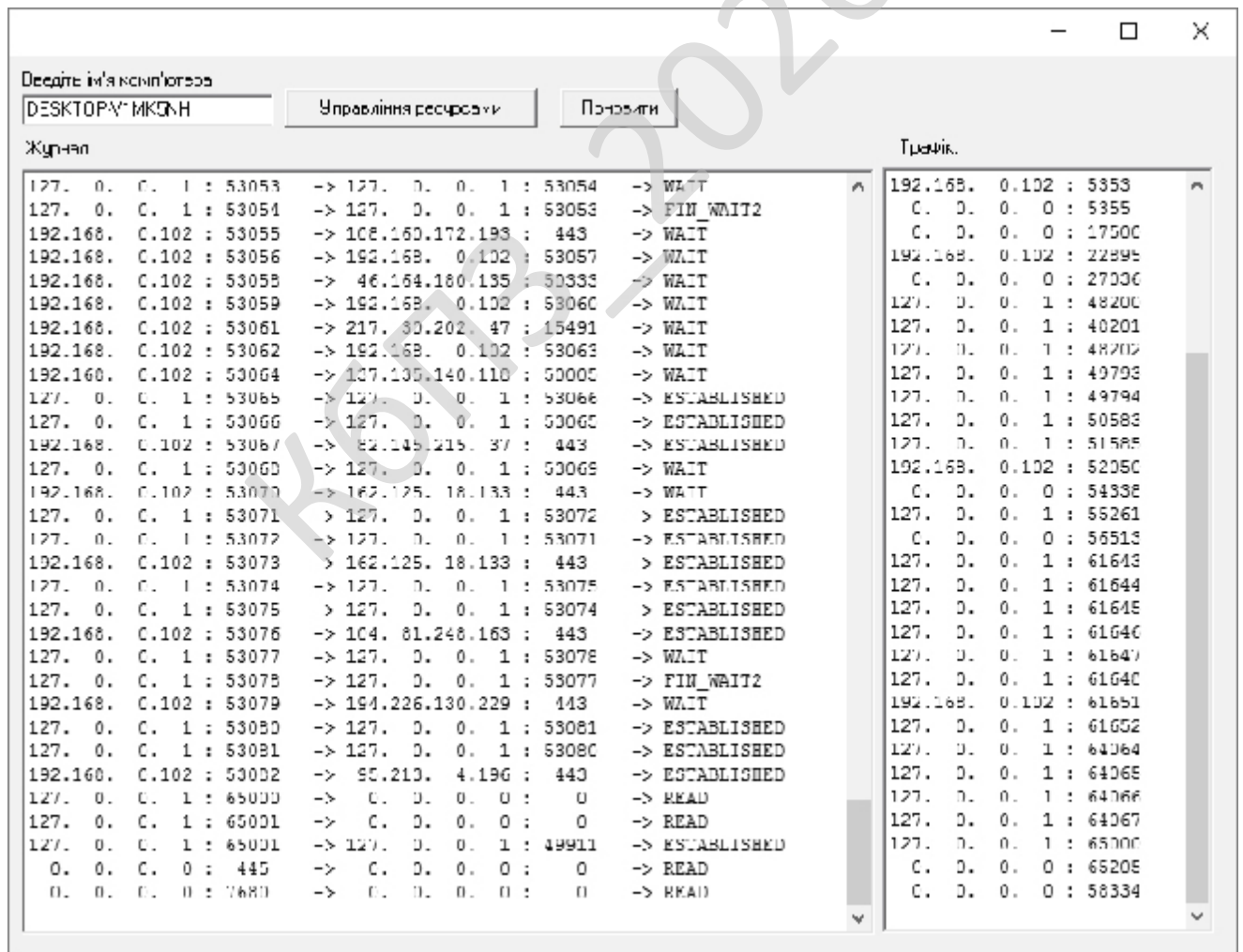


Рисунок 5.1 – Головне вікно ПЗ

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.

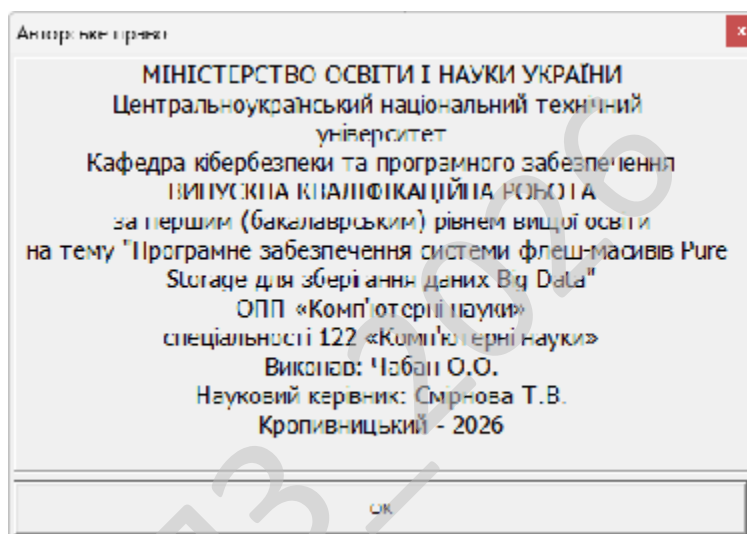


Рисунок 5.2 – Авторське право

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

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 73   |

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.
- Оновлення.
- 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.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 74   |

в ІТ 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).
- 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.

Проводилось тестування чорної скриньки.

Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

- Як виконуються функції програми.
- Як приймаються вихідні дані.
- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме  $10^{10}$ . Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

Програмний виріб тут розглядається як «чорна скринька», чію поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

- Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ІТс).

- Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його ОТ).

Будь-який спосіб тестування «чорної скриньки» повинен:

- Виявити такі вхідні дані, які з високою ймовірністю належать набору ІТс.
- Сформулювати такі очікувані результати, які з високою ймовірністю є елементами набору ОТ.

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

- Некоректних чи відсутніх функцій.
- Помилки інтерфейсу.
- Помилки у зовнішніх структурах даних або в доступі до зовнішньої бази даних.
- Помилки характеристик (необхідна ємність пам'яті і т.д.).
- Помилки ініціалізації та завершення.

Обрано умови розповсюдження – Freeware. Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів.

Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення.

Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються.

Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

## 6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи флеш-масивів Pure Storage для зберігання даних Big Data.

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

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем флеш-масивів Pure Storage для зберігання даних Big Data.

- Досліджена система флеш-масивів Pure Storage для зберігання даних Big Data.

- На основі отриманих результатів досліджень створена програмна реалізація системи флеш-масивів Pure Storage для зберігання даних Big Data.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання флеш-масивів Pure Storage для зберігання даних Big Data.

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 системи флеш-масивів Pure Storage для зберігання даних Big Data. 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 RC5.

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-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 80   |

## 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-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 81   |

10. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем ІР-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 користувацького 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.

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., Serhiienko 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.

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-122.26.0033.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 86   |

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.